1500
1500 ICL

Series 1500

the transaction system

Programmer's
Reference
Manual

1500 1500 ICL

Series 1500

Programmer's Reference Manual

ICL endeavours to ensure that the information in this document is correct and fairly stated, but does not accept liability for any error or omission.

The development of ICL products and services is continuous and published information may not be up-to-date. Any particular issue of a product may contain part only of the facilities described in this document or may contain facilities not described here. It is important to check the current position with ICL.

Specifications and statements as to performance in this document are ICL estimates intended for general guidance. They may require adjustment in particular circumstances and are therefore not formal offers or undertakings.

Statements in this document are not part of a contract or program product licence save insofar as they are incorporated into a contract or licence by express reference. Issue of this document does not entitle the recipient to access to or use of the products described, and such access or use may be subject to separate contracts or licences.

From 1 October 1976, International Computers Limited has acquired certain of the computer activities previously carried on by the Singer Company.

The products and services now offered by ICL may differ in detail from those formerly offered by your Singer supplier. Readers should therefore consult their local ICL office to confirm the current availability and support status of any products or services referenced in this publication.

Technical Publication 7851

© International Computers Limited 1978

First Edition 1978

ICL will be pleased to receive readers' views on the contents and organization, etc. of this publication. Please write to:

ICL
Operations Division
ICL Plaza
Route 12 North
Utica, New York 13502
Attn: Manager, Product Support Engineering

Distributed by:

ICL, Inc.
Operations Division
Technical Publications Dept.
Cosby Manor Road
Utica, N.Y. 13502
U.S.A.

The reader should be aware that the term "intelligent terminal" is used throughout this manual when referring to the Models 1501 through 1504 and the Model 1580. Since the writing of this manual, the formal name of these units has been changed to "transaction system" and, therefore, "transaction system" appears on the cover.

This manual is intended to be used by ICL programmers and data processing specialists but is also available to the customer. The purpose of this manual is to introduce a programmer to the Series 1500 equipment, describe the characteristics of that equipment from the programmer's point of view, and provide programming examples in sufficient detail to allow the reader to learn how to program in the Assembler System language. This manual is intended to complement the Assembler System and CDE/BTL Reference Manuals, which concentrate on operating instructions for those programming systems.

The Programmer's Manual is made up of the following sections:

a. Section 1, Equipment Description, which is a brief description of each unit of equipment from a programmer's point of view. The relationship between various models of terminals is also described in order to familiarize a programmer with the relationship between the terminals and the instruction sets, memory, storage, and displays available.

b. Section 2, Instruction Descriptions, which outlines the format of each machine instruction in the instruction set, illustrates its coding, and describes its function. Processor architecture and basic operating principles are described in detail so that the description of instructions can be related to the internal organization of the equipment.

c. Section 3, Programming Techniques. This section outlines the basic techniques involved in programming the Series 1500 equipment. It includes such information as the format of tape records, format of display areas, memory available, instructions on how to program communications, the use of the background routines, etc. All major equipment functions and peripheral equipment are covered.

d. Appendix A, Memory Map, which shows the location of backgrounds in memory. Also shown in the memory map are the memory locations reserved for special uses.

e. Appendix B, Standard I/O Library 1 Instructions. This appendix is a reproduction of instructions originating in the Series 1500 Software Development Group and is available in only those manuals distributed within ICL; it is not included in manuals provided to customers.

The ways in which the sections of the manual are used will undoubtedly be as varied as the readers themselves. However, it is expected that a novice programmer would first familiarize himself with the characteristics of each unit available. Instruction descriptions should be read next to determine the power and flexibility of the instruction set. Some readers may be interested in processor architecture, while others have little interest in that detailed information.

In the programming techniques section, all readers are expected to use the general information and the descriptions of background routines and tables, those being essential to programming the processor.

The remainder of Section 3 is subdivided into programming techniques for specific units, and it is expected that the reader would choose only those in which he had immediate interest.

After reading this manual, the reader would normally refer to either the Assembler System or CDE/BTL for instructions on programming in either of those languages. As required, the reader would return to the Programmer's Reference Manual to look up detailed information and study the examples.

Most manual numbers referenced within this publication have been converted to a new 4-digit numbering system. Listed below are all publications that bear the new numbers.

New Number	Replaces Old Number
7804	22-6504-06
7805	.91015
7827	.91020
7835	.91024
7841	.91017
7843	.91014
7844	.91001
7845	.91028
7847	.91002

New Number	Replaces Old Number
7852	91021
7855	91008 thru 91012
7856	91035
7858	91019
7860	22-6504-01
7861	22-6504-14
7862	22-6504-02
7863	22-6504-03
7864	22-6504-04
7866	22-6504-07
7867	22-6504-08
7868	22-6504-09
7869	22-6504-11

SECTION		PAGE
1	EQUIPMENT DESCRIPTION.	1-1
	1-1. General.	1-1
	Compatible Instruction Sets.	1-1
	Transparent Keyboard	1-4
	Program-Controlled Character Sets.	1-4
	Serial I/O Channel	1-4
	Cartridge Tapes.	1-5
	Standard Tape Handling	1-6
	Disc Pack Handling	1-6
	Communications Capability.	1-6
	Environmental Requirements	1-7
	1-2. The Intelligent Terminals.	1-7
	Model 1501 Intelligent Terminal.	1-7
	Model 1501-40/43 Intelligent Terminal.	1-12
	Model 1502 Intelligent Terminal.	1-13
	Model 1503 Intelligent Terminal.	1-16
	Model 1504 Intelligent Terminal.	1-17
	Model 1580 Intelligent Terminal.	1-20
	1-3. Tape Drives.	1-23
	Model 1511 and 1512 Tape Units	1-25
	Model 1513 and 1514 Tape Units	1-25
	1-4. Printers	1-27
	Model 1525 Printer	1-27
	Model 1551 Printer	1-28
	Model 1553, 1554, 1555, and 1556 Printers.	1-29
	1-5. Disc Drives.	1-30
	1-6. Communications Units	1-31
	Model 1534 Asynchronous Communications Adapter	1-32
	Model 1534-1 and 1534-2 Asynchronous Communications Adapters	1-33
	Model 1535 Binary Synchronous Communications (BSC) Adapter.	1-33
	Model 1539 Interruptible Communications Controller	1-33
	Model 1536 Intelligent Communications Adapter.	1-34

CONTENTS (Continued)

SECTION		PAGE
1	1-7. Other Peripheral Units	1-35
	Model 1531 Card Reader	1-36
	Model 1517 Paper Tape Reader	1-36
	Model 1590 MDTS Poller	1-36
	Model 1538 Automatic Dial Unit	1-37
	1-8. An Integrated System	1-37
2	INSTRUCTION DESCRIPTIONS	2-1
	2-1. General	2-1
	2-2. Processor Architecture	2-3
	Word Formats	2-4
	Memory Addresses	2-4
	Instruction Addressing	2-9
	Use of the Stack Pointer	2-10
	Data Addressing	2-12
	Instruction Cycles	2-14
	Instruction Classes	2-14
	Initial Conditions and Program Loading	2-20
	Interrupt Handling	2-22
	Stall and Skip Conditions	2-22
	Use of the U and V Bits	2-23
	Processor Status Word	2-24
	Stack Pointer	2-25
	Terminal Identification Plug	2-26
	B Register	2-26
	Memory Data Register	2-27
	Memory Address Register	2-27
	Adder	2-28
	Memory Input and Output Multiplexers	2-29
	Address Net	2-29
	2-3. C4 Processor Instruction Set	2-31
	TLJ - Test Literal and Jump	2-31
	TMJ - Test Mask and Jump	2-35
	TLX - Test Literal and Exit	2-35
	TMX - Test Mask and Exit	2-36
	BRU - Branch Unconditional	2-36
	BRE - Branch on Equal	2-36
	BRH - Branch on High	2-36
	BRL - Branch on Low	2-37
	SBU - Stack and Branch Unconditional	2-37
	SBE - Stack and Branch on Equal	2-37
	SBH - Stack and Branch on High	2-37

CONTENTS (Continued)

SECTION		PAGE
2	SBL - Stack and Branch on Low.	2-37
	EXB - Exit and Branch.	2-37
	EXU - Exit Unconditional	2-38
	SAM - Select Auxiliary Memory.	2-38
	SMS - Set Memory Section	2-38
	SMC - Set Memory Control	2-38
	SSC - Set Memory Section and Control	2-38
	SAC - Set Arithmetic Condition	2-38
	LSW - Load Sense Switches.	2-39
	EMP - Enable Memory Parity	2-39
	LPS - Load Processor Status.	2-39
	DPI - Disable Processor Interrupt.	2-39
	EPI - Enable Processor Interrupt	2-39
	CPI - Clear Processor Interrupt.	2-39
	TRM - Load Terminal Identification	2-39
	LDA - Load Accumulator	2-40
	LDX - Load Index Register.	2-40
	LIA - Load Instruction Address	2-40
	STA - Store Accumulator.	2-40
	ADA - Add to Accumulator	2-40
	ADX - Add to Index Register.	2-40
	SUA - Subtract from Accumulator.	2-40
	SUX - Subtract from Index Register	2-40
	ANA - Logical AND to Accumulator	2-41
	SAN - Shift and Logical AND to Accumulator	2-41
	ERA - Exclusive OR to Accumulator.	2-41
	SER - Shift and Exclusive OR to Accumulator.	2-41
	IRA - Inclusive OR to Accumulator.	2-41
	SIR - Shift and Inclusive OR to Accumulator.	2-41
	CPA - Compare Accumulator.	2-41
	CPX - Compare Index Register	2-42
2-4.	C4 Virtual Processor Instruction Set	2-42
	CSL - Card Select (Select Auxiliary Memory).	2-42
	EPN/EPO/EPC - Enter Processor New/Old/Conditional.	2-44
	REL - Processor Release.	2-44
	SCF/RCF - Set/Reset Condition Flag	2-45
	DCS - Disc Card Select	2-45
	DDC - Device Demand Control.	2-45
	RVP - Read VP State.	2-45
2-5.	C5 Processor Instruction Set	2-46
2-6.	Standard C4 Input/Output Instructions.	2-47
	Cartridge Tape Instructions.	2-48
	Acquisition of the General Status Byte	2-49

CONTENTS (Continued)

SECTION			PAGE
2		Keyboard Instructions.	2-49
		Small CRT Display Instructions	2-50
		Serial Input/Output Channel Instructions	2-51
		Dual I/O Channel Instructions.	2-52
		Communications Interface Instructions.	2-52
		Parallel I/O Channel Instructions.	2-56
	2-7.	Special C4 Virtual Processor Input/Output Instructions . .	2-57
		DES Controller Instructions.	2-58
		Quadruple Asynchronous Communications Channel Instructions	2-59
		Local Asynchronous Communication Channel Instructions. . .	2-60
		Utility Channel Instructions	2-60
	2-8.	C5 Input/Output Instructions	2-62
		I/O Controller Function Instructions	2-64
		Keyboard and Interrupt Control Instructions.	2-65
		Large CRT Instructions	2-66
		Communication Channel Instructions	2-68
		Interruptible Communications Controller Instructions . . .	2-68
3	PROGRAMMING TECHNIQUES		3-1
	3-1.	General.	3-1
	3-2.	Relationship of Machine Instructions to the Assembler System	3-1
	3-3.	Organization of Storage.	3-5
		Memory Pages	3-5
		Format of a Standard Cartridge Tape Record	3-7
		Format of a Disc Sector.	3-8
	3-4.	Series 1500 Code	3-9
	3-5.	Background Routines and Tables	3-11
		Bootstrap and Loader Routine (BTS)	3-15
		Standard Dot Pattern (SDP)	3-16
		Standard Keyboard Translation Table (STT).	3-16
		Standard Mini-Tape Read and Write Function (MRW)	3-16
		Page Select Read and Write Functions	3-17
		Mini-Tape I/O Control (IOC).	3-18
		Maxi-Tape Functions.	3-20
		Matrix Printer Function (MXP).	3-21
		Line Printer Function (LNP).	3-22

CONTENTS (Continued)

SECTION			PAGE
3		Printer Translation Table	3-23
		Typewriter Print Function (TYP)	3-23
		Typewriter Single Character Print Function (P1S)	3-24
		Printer Translation Table TT1 (1500-to-Typewriter)	3-24
	3-6.	Small CRT Display	3-24
		Display Format	3-25
		Instructions	3-27
		Underscore	3-30
		Alternate Character Set	3-32
		Programming Aids	3-32
	3-7.	Large CRT Display	3-35
		Display Format	3-36
		Character Matrix	3-37
		The Cursor	3-38
		LCRT I/O Instructions	3-39
		Modes of Operation	3-42
		C4 Mode	3-43
		Options	3-45
		Programming Aids	3-45
	3-8.	Cartridge Tape Drives	3-53
		Recording Method	3-53
		Record Format	3-53
		Cartridge Tape Instructions	3-54
		Tape Unit Selection	3-56
		A Write Operation	3-56
		Read Check and Rewrite Operations	3-59
		A Read Operation	3-60
		Status Byte	3-61
		High-Speed Search	3-62
		Programming Examples	3-65
	3-9.	Serial Input/Output Channel	3-67
		Organization of Data on the SIO Line	3-69
		SIO Channel Instructions	3-72
		SIO Channel Status Byte	3-75
		Sequence of Operation in the Master Mode	3-76
		Sequence of Operation in the Slave Mode	3-77
		Dual Serial Input/Output Channels	3-78
	3-10.	Communications Adapters	3-79
		Typical Equipment Arrangement	3-80
		Line Arrangements	3-80
		Forms of Data Transferred	3-81
		Line Protocol	3-82
		Transmission Codes	3-82

CONTENTS (Continued)

SECTION		PAGE
3	Models of Communications Adapters	3-82
	Relationship to the Intelligent Terminals	3-82
	Model 1534 Asynchronous Communications Adapter	3-85
	Model 1534 Programming Techniques	3-89
	Model 1534-1 and 1534-2 Asynchronous Communications Adapters	3-99
	Model 1534-1 and 1534-2 Programming Techniques	3-99
	Model 1535 Binary Synchronous Communications (BSC) Adapter	3-106
	Model 1535 Programming Techniques	3-111
	Timer	3-116
	Model 1539 Interruptible Communications Controller	3-123
	Model 1539 Programming Techniques	3-132
	3-11. Printers	3-150
	Model 1525/1526 Printer	3-150
	Model 1525 Instructions	3-150
	Model 1525 Status Word	3-151
	Model 1525 Programming Methods	3-151
	Model 1551 S/D Printer	3-157
	Model 1551 Instructions	3-157
	Model 1551 Status Word	3-158
	Model 1551 Programming Methods	3-158
	Form Control Tape	3-163
	Model 1553-1556 Printers	3-164
	Model 1553-1556 Instructions	3-165
	Model 1553-1556 Status Word	3-166
	Model 1553-1556 Programming Methods	3-166
	3-12. Maxi-Tape Units	3-170
	Tape Formats	3-170
	Record Length	3-174
	Maxi-Tape Instructions	3-175
	Maxi-Tape Status Word	3-176
	Programming Methods	3-178
	3-13. Disc Units	3-189
	Disc Format	3-189
	Access to Memory	3-190
	Disc Instructions	3-191
	Disc Status Word	3-195
	Preparation of the Parameter Table	3-195
	Handling Disc Controller Interrupts	3-197
	Programming Methods	3-199
APPENDIX A		A-1
APPENDIX B		B-1

ILLUSTRATIONS

FIGURE		PAGE
1-1	Keyboard Layout.	1-5
1-2	Tape Cartridge	1-6
1-3	A Series 1500 Disc Pack.	1-7
1-4	Model 1501 Intelligent Terminal.	1-8
1-5	Model 1501-40 Intelligent Terminal	1-13
1-6	Model 1502 Intelligent Terminal.	1-14
1-7	Model 1503 Intelligent Terminal.	1-17
1-8	Model 1504 Intelligent Terminal.	1-18
1-9	Model 1580 Intelligent Terminal.	1-21
1-10	Model 1533 Dual Drive.	1-24
1-11	Model 1511 or 1512 Tape Unit	1-26
1-12	Model 1513 or 1514 Tape Unit	1-26
1-13	Model 1525 Printer	1-28
1-14	Model 1551 Printer	1-29
1-15	Model 1553, 1554, 1555, or 1556 Printer.	1-30
1-16	Model 1534-2 Communications Adapter.	1-32
1-17	Model 1536 Intelligent Communications Adapter.	1-34
1-18	An Integrated System	1-38
1-19	A Clustered System	1-41
1-20	Integrated System Showing the Use of the 1504, 1536, and 1580.	1-43
2-1	Block Diagram of Typical Intelligent Terminal.	2-1
2-2	Basic C4 Processor Block Diagram	2-5
2-3	Typical Instruction Format	2-7
2-4	Organization of Memory	2-7
2-5	Memory Address Format.	2-8
2-6	Sample Address	2-8
2-7	Contents of Stack Pointer Addresses.	2-9
2-8	Typical Location of Instructions in Memory	2-10
2-9	Use of the Stack Pointer	2-11
2-10	Format of an Instruction Using the Indexed Address Mode.	2-13
2-11	Functions Performed During Each Instruction Cycle.	2-16
2-12	Format of EXI Instruction in C5 Processor Instruction Set.	2-46
2-13	Small CRT Display Function Codes	2-51
2-14	C5 I/O Instruction Format.	2-63
3-1	Format and Description of the ADA Instruction in Machine Language	3-2
3-2	Examples of Instructions in an Assembler System Source Listing Printout.	3-4
3-3	Format of a Cartridge Tape Record.	3-7
3-4	Disc Sector Format	3-8
3-5	Tri-Octal Grouping of a Data Byte.	3-9
3-6	Format of Small CRT Display Area	3-26
3-7	Formation of the Character A	3-28
3-8	Format of IWR for Small CRT Instructions	3-29

ILLUSTRATIONS (Continued)

FIGURE		PAGE
3-9	Memory Address Format.	3-29
3-10	Display Sequence in Normal Mode.	3-31
3-11	Display Sequence in Interleaved Mode	3-31
3-12	Memory Allocation vs. Display Area	3-37
3-13	LCRT Character Matrix.	3-38
3-14	Format of LCRT Instructions.	3-39
3-15	Small CRT Formats.	3-44
3-16	Series 1500 Code to USASCII Translation.	3-47
3-17	Weighted USASCII Code Generator.	3-48
3-17A	Large CRT Sample Program 1	3-52
3-17B	Large CRT Sample Program 2	3-52
3-18	Format of Cartridge Tape Instructions.	3-54
3-19	Length of Record with Respect to Head Positions.	3-57
3-20	Format of Tape Status Byte	3-61
3-21	Use of the End-of-Text Records	3-64
3-21A	Tape Write Sample Program.	3-66
3-21B	Retry Recovery Sample Program.	3-66
3-21C	High-Speed Move Sample Program	3-67
3-22	Address Sequence	3-69
3-23	Typical Exchange Between an Intelligent Terminal (Master) and Magnetic Tape Unit (Slave)	3-70
3-24	Serial I/O Channel Byte Formats.	3-71
3-25	Typical Equipment Required	3-80
3-26	Models of Communications Adapters.	3-83
3-27	Asynchronous Serial Communications Format.	3-85
3-28	Format of Model 1534 Control Word.	3-86
3-29	Format of a Synchronous Transmission	3-107
3-30	Use of the DLE Character	3-116
3-31	SDLC Transmission Format	3-124
3-32	Control Word Format.	3-127
3-33	Use of the DLE Character in the Transparent Mode	3-137
3-34	Example of a Printed Character	3-152
3-35	Portion of the CT1 Translation Table	3-162
3-36	Section of the Model 1551 Format Control Tape.	3-164
3-37	Model 1553-1556 Form Control Tape.	3-167
3-38	Tape Formats	3-171
3-39	Record Spacing	3-173
3-40	Disc Sector Format	3-190
3-41	Format of Disc Controller Instructions	3-191
3-42	Format of Parameter Bytes.	3-192
3-43	Typical Parameter Table.	3-193

TABLES

TABLE		PAGE
1-1	Basic Instruction Set.	1-2
1-2	Instruction Set Additions for Model 1504	1-4
1-3	Maxi-Tape Drive Characteristics.	1-25
1-4	Printer Speeds	1-27
2-1	Cycles Required by Each Instruction.	2-15
2-2	Tape Loader ROM Program.	2-21
2-3	C4 Processor Instruction Set	2-32
2-4	Additional C4 Virtual Processor Instructions	2-43
3-1	Program Segments in Standard I/O Library 1	3-12
3-2	Value of the Small CRT IWR	3-34
3-3	SIO Channel Instructions	3-73
3-4	Baud Selection Control Words	3-90
3-5	Model 1534 Routines.	3-97
3-6	Model 1534-1 and 1534-2 Routines	3-104
3-7	Model 1535 Sample Routines	3-121
3-8	Example of Model 1539 Transmit and Receive Routines.	3-145
3-9	Model 1539 Interrupt Handler Routine	3-147
3-10	Examples of Model 1525 Routines.	3-153
3-11	Examples of Model 1551 Routines.	3-159
3-12	Sample Maxi-Tape Routines.	3-179
3-13	Disc Programming Routines.	3-200

1-1. GENERAL.

The Series 1500 is a line of general-purpose data processing units, including three basic types of equipment:

a. Intelligent terminals, which are small computers capable of performing most data processing operations.

b. Standard peripheral equipment, consisting of the common magnetic tape drives, printers, and disc drives.

c. Special equipment, which includes communications adapters and other units required to adapt the intelligent terminals to a variety of applications. An example of these units is the MDTS poller, which is a unit that accepts inputs from electronic cash registers and forwards them to an intelligent terminal for processing.

The purpose of this section is to provide an overall physical and functional description of each unit. Also included with the description of each unit is a discussion of the features of interest to a programmer, such as instruction sets and memory capacity. Before each unit is discussed individually, however, some of the features common to most units are described.

Compatible Instruction Sets.

One important feature of the Series 1500 Intelligent Terminals is that the basic instruction set, with minor variations, can be operated by all six terminals (Models 1501, 1501-40/43, 1502, 1503, 1504, and 1580). This means that a program prepared for one terminal, such as the Model 1501, can be executed by a Model 1503 with little or no change. Of course, the equipment required by the program, such as storage and display, must be compatible with the terminal capabilities.

The basic instruction set that can be operated by the processors in the Model 1501, 1501-40/43, 1502, and 1503 is listed in table 1-1. With the exceptions noted, these instructions can also be executed by the 1504 and 1580. Table 1-2 lists the additions made to the basic processor instruction set to operate the Model 1504.

Table 1-1. Basic Instruction Set

Mnemonic	Description
TLJ	Test Literal and Jump
TMJ	Test Mask and Jump
TLX	Test Literal and Exit
TMX	Test Mask and Exit
BRU	Branch Unconditional
BRE	Branch on Equal
BRH	Branch on High
BRL	Branch on Low
SBU	Stack and Branch Unconditional
SBE	Stack and Branch on Equal
SBH	Stack and Branch on High
SBL	Stack and Branch on Low
EXB	Exit and Branch
EBI°	Exit, Branch, and Enable Interrupt
EXU	Exit Unconditional
EXI°	Exit and Enable Interrupt
SAM**	Select Auxiliary Memory
SMS	Set Memory Section
SMC	Set Memory Control
SSC	Set Memory Section and Control
SAC	Set Arithmetic Condition
LSW or EMP*	Load Sense Switches or Enable Memory Parity
LPS	Load Processor Status
DPI□	Disable Processor Interrupt

Table 1-1. Basic Instruction Set (Continued)

Mnemonic	Description
EPI□	Enable Processor Interrupt
CPI**	Clear Processor Interrupt
TRM**	Load Terminal Identification
LDA	Load Accumulator
LDX	Load Index Register
LIAx	Load Instruction Address
STA	Store Accumulator
ADA	Add to Accumulator
ADX	Add to Index Register
SUA	Subtract from Accumulator
SUX	Subtract from Index Register
ANA	Logical AND to Accumulator
SAN	Shift and Logical AND to Accumulator
ERA	Exclusive OR to Accumulator
SER	Shift and Exclusive OR to Accumulator
IRA	Inclusive OR to Accumulator
SIR	Shift and Inclusive OR to Accumulator
CPA	Compare Accumulator
CPX	Compare Index Register

*Executed as LSW only on 1501 terminals equipped with non-parity memories; as EMP on all other terminals.
**Not applicable to 1504 and 1580.
○ 1580 only.
□ Not applicable to 1504.
x Not applicable to 1580.

Table 1-2. Instruction Set Additions for Model 1504

Mnemonic	Description
CSL	Card Select
EPN/EPO/EPC	Enter Processor New/Old/Conditional (Processor Program Exchange)
REL	Processor Release
SCF/RCF	Set/Reset Condition Flag
DCS	Disc Card Select
DDC	Device Demand Control
RVP	Read VP State

Transparent Keyboard.

The same keyboard is used in all the Series 1500 terminals, although its housing differs from one terminal to another. The term "transparent" means that when each key is depressed it produces a code to indicate its physical position rather than a code representing the characters inscribed on the keytops. A program can assign any meaning to any key; thus, no special control panels are required to operate the system.

A photograph of the keyboard is placed above an outline drawing of the keyboard in figure 1-1. Here, the code produced by each key is shown. For example, the F and ; key produces the 042 code in the lower case mode and the 142 code in the shifted, or upper case mode.

Program-Controlled Character Sets.

Held in the main memory of each intelligent terminal are the dot patterns that produce characters on the screen of the small display. Since these patterns are loaded into memory with the application programs, they can be changed to represent any character set required. The character set for the large display screen is stored in a replaceable read-only memory, so the character set for the large display can also be changed easily.

Serial I/O Channel.

The serial input/output channel is a single cable that connects the peripheral equipment to an intelligent terminal. Since the line is small and easily connected and disconnected at each unit, it greatly simplifies the cabling required for a Series 1500 system.

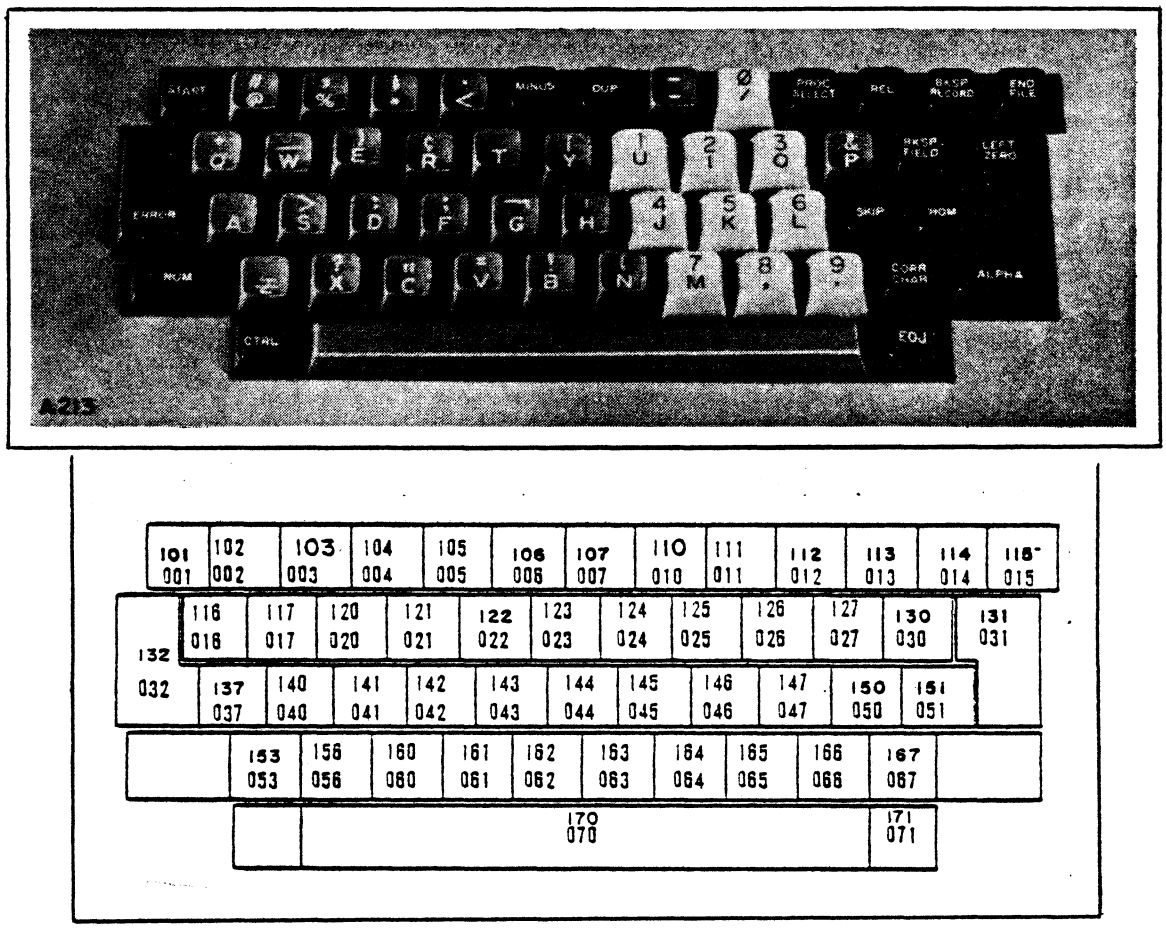


Figure 1-1. Keyboard Layout

Made possible by the fact that nearly all peripheral units are very slow when compared to internal computer speeds, the serial I/O channel allows the processor to exchange data with one unit at a time in serial form with ample time between transfers. Data transfer on the cable is at a speed of 58,800 characters per second. Sixty-four device addresses can be used, and devices can be located at distances up to 1000 feet away from the terminal. A cable amplifier unit is available for longer connections.

Cartridge Tapes.

A key feature of the Series 1500 terminals is the cartridge tape, which is shown in figure 1-2. Cartridge tapes are used for initial program loading, and for data input and outputs during operation.

The tape cartridges are approximately two inches in diameter, and they enclose the tape storage reel to protect it from handling contamination. When a tape is inserted into a cartridge drive, the drive threads and handles it automatically under processor control. When use is finished, the drive unthreads the cartridge, and the cartridge is simply lifted out.

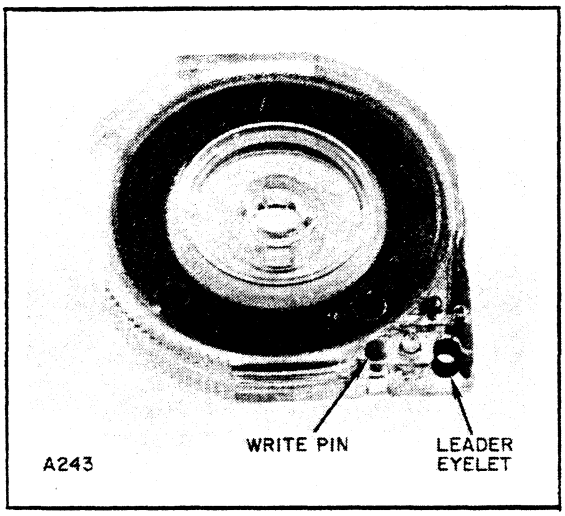


Figure 1-2. Tape Cartridge

Cartridge tape drives employ the same sensing systems used on larger high performance tape drives. They can detect the presence of a tape cartridge, determine whether it is rewound or in use, sense whether it is write-protected or not, and sense its reflective end-of-tape marker. All of the tape drive's status conditions are available to the programmer.

Standard Tape Handling.

Available as peripheral units are four models of tape drives that handle standard computer tape reels. This allows the Series 1500 terminals to process data from and prepare data for larger computer systems. The tape drives that handle the standard computer reels are called "maxi-tapes" and the cartridge tapes unique to the Series 1500 are called "mini-tapes".

Disc Pack Handling.

There are two models of disc drives available with the Series 1500 terminals. One of these drives accommodates a removable disc pack of the 5440 type widely used in the data processing industry. Of course, the data formats must be compatible, but the removable disc pack allows the Series 1500 equipment to exchange discs with many computer systems. A disc pack is shown in figure 1-3.

Communications Capability.

One highly important function of data processing systems is the transfer of data to remote locations over telephone or teletype lines. When equipped with a communications adapter available in the Series 1500 line and a MODEM (modulator-demodulator) from any one of many manufacturers, a Series 1500 terminal is capable of sending and receiving data over long distances by most of the commonly-used methods.

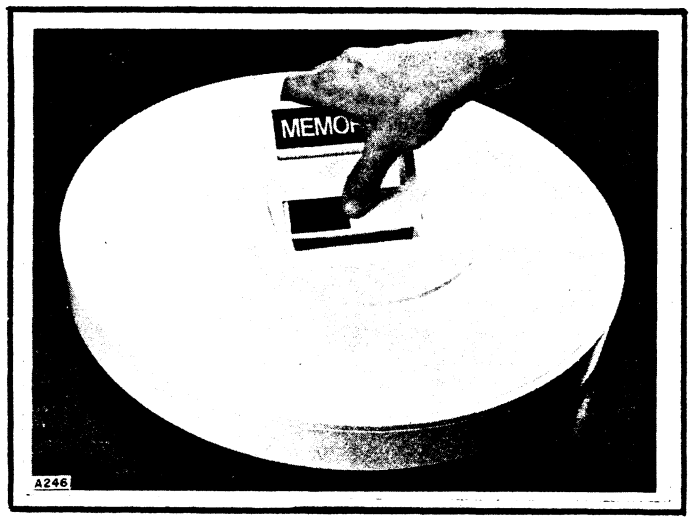


Figure 1-3. A Series 1500 Disc Pack

Environmental Requirements.

The Series 1500 equipment is designed to operate in a normal office environment, with no outside source of cooling. Fans within the units provide all the cooling required.

1-2. THE INTELLIGENT TERMINALS.

In the data processing industry, the term "intelligent terminal" is most frequently used to describe a unit that pre-processes operator entries before they are forwarded to a central computer system and, in turn, develops a response to the operator. This is one possible application of the Series 1500 terminals. However, these terminals are sufficiently powerful and versatile to be used as complete data processing systems.

There are six intelligent terminals in the Series 1500 line: the 1501, 1501-40/43, 1502, 1503, 1504, and 1580. Most have common features and can execute the same basic instruction set. In addition, most can operate the standard peripheral units and be equipped with the same options. In the following paragraphs, the terminals are described in order by model number.

Model 1501 Intelligent Terminal.

The Model 1501 Intelligent Terminal shown in figure 1-4 is a small computer that includes a processor, a memory, and an input/output controller. Two tape drives, which are operated by the computer, are mounted in the unit. Magnetic tape cartridges loaded on these drives provide programs and data for the computer and store data produced by the computer.

A display screen and keyboard are housed in the terminal. These components allow the operator to control the data processing for which the terminal is intended. The keyboard allows the operator to enter data and control operation of the computer, while the screen displays data, questions, and instructions to assist the operator.

After power is applied to the terminal, the first program is loaded into the processor's memory from the tape cartridge on the second tape drive. This is accomplished when the operator pushes the PROGRAM LOAD switch, which causes the processor to perform a short loading program. From this point on, the operation of the processor is under control of the program read from the second tape drive.

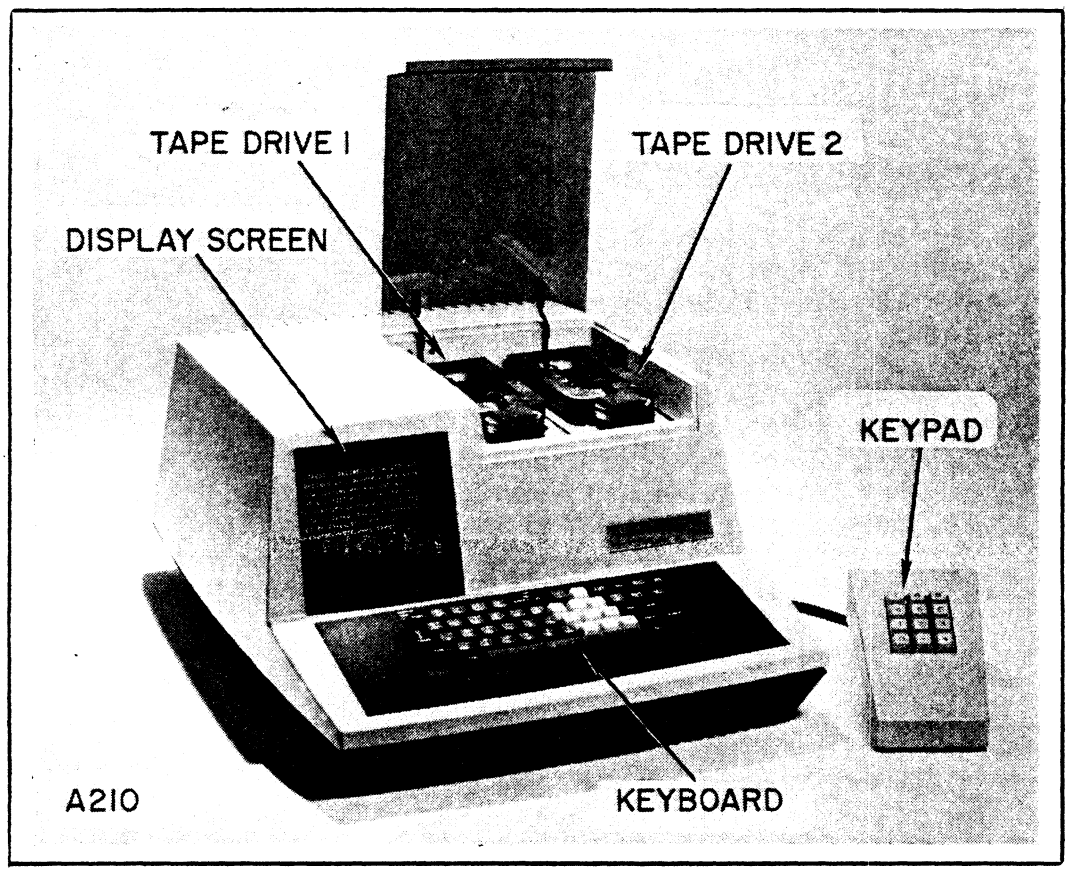


Figure 1-4. Model 1501 Intelligent Terminal

Controls

Aside from the keyboard, there are only three switches on the Model 1501. The first is the power on/off switch at the right rear corner at the base of the terminal. Located in a well beneath the display screen are the following two switches:

a. PROGRAM LOAD switch, which when pushed to the PROGRAM LOAD position (toward the display), causes the program on tape drive 2 to be loaded into the processor memory. This switch has three positions: PROGRAM LOAD, off, and program interrupt. The interrupt position, which is not labeled and is activated when the switch is pulled away from the display, causes the processor program to jump to a specific memory location to perform an interrupt servicing routine, if one is stored there. The center position of the switch is the off position.

b. SYSTEM RESET switch, which when depressed also causes the processor program to jump to a specific memory location and execute the program located there. The action that the processor takes when the SYSTEM RESET switch is used depends upon the program currently stored in the memory.

Cartridge Tape Drives and Tapes

Two cartridge tape units are standard in the 1501. Each holds one cartridge like that shown in figure 1-2.

The tape drives have a variety of uses in system operation. Examples are: (1) initially loading a program into the system, (2) providing input data, and (3) collecting output data.

The data is recorded serially on a single channel on the tape at a density of 1600 bits per inch. Capacity of the 100-foot cartridge is up to 900 records with 128 data bytes each, but the number of bytes in a record and the number of records depends upon the application programs. Reading and writing speed is 10 inches per second, and the tape is driven at 40 inches per second during search and rewind operations.

Loading the cartridge is a simple procedure that is fully explained in the Operator Instructions Manual. Once the cartridge is placed in the loader, the loader must be pushed down and the locking lever moved to the lock position. Of course, this procedure is reversed in order to unload a tape.

Normally, the processor rewinds the tape before instructions are given to remove the cartridge. If, however, the tape is to be removed before it has been rewound by the processor, it is necessary to depress the red rewind pushbutton on the tape drive. Once this button has been depressed, the tape will be rewound and can be unloaded.

Located on the outer case of the cartridge is a two-position hole for the write pin. If the write pin is inserted into the position on top of the cartridge, the tape cannot be written on. If the write pin is inserted into the position on the bottom of the cartridge, the tape may be written on; this is called the "write-enable" position. Figure 1-2 shows the bottom of the cartridge with the write pin inserted in the "write-enable" position.

The Keyboard and Display

Housed in the 1501 are a keyboard and a display screen. The display is the primary means of machine-to-operator communication. It simplifies the operator's duties by providing instructions when needed. Audible tones inform the operator when an error is encountered, and the display informs the operator as to the nature of the error.

Up to 8 rows of 32 characters each can be shown on the screen. Each character is formed on a 5-by-7 dot matrix. In the standard character set 64 different characters are available, including the alphabet, numerals, and punctuation marks. However, the processor program can create any character set on the display, and there is a 128-character set available.

The characters are a series of dot patterns stored in the main memory. This means that the same hardware can be used world-wide to display local character sets, with dot patterns selected according to local needs.

Series 1500 Intelligent Terminals, including the Model 1501, are equipped with a standard "keypunch arranged" keyboard. A fully-programmable translation table held in the main memory allows easy adaptation to such keyboard arrangements as teletype or typewriter. Letters such as the German á or ö, or the Spanish ñ, may be easily substituted for other characters, and the keytops are easily changed to produce any character set.

The keyboard consists of a 52-key alphanumeric keyboard, and a keypad is optional. A speaker is housed below the keyboard assembly to provide audible cues to the operator.

The keyboard is a solid-state device, using no mechanical switches or contacts for any function. Each key generates a unique code identifying its physical location to the active program.

Organization of the Mini-Computer

Series 1500 terminals are called "intelligent terminals" because their operation is controlled by a versatile and powerful mini-computer. In the Model 1501, this computer consists of a processor, a memory, and an input/output controller. Each of these components is made up of printed circuit boards.

The processor has 41 different instructions in its instruction repertoire, and there are many input/output instructions that are executed by the I/O controller or peripheral units. The format of all instructions and data in the terminal is organized into bytes (8 bits). Instruction execution time ranges from three to six microseconds. Memory capacity is variable, starting with a minimum of 4,096 bytes and expandable to 8,192 bytes and 16,384 bytes.

The I/O controller is provided to handle data transfer between most I/O devices and the processor. It controls the operation of the cartridge tape drives, the display, the keyboard, and the serial input/output channel.

All peripheral equipment operates on the serial I/O channel. This channel is a single coaxial cable terminated at both ends. Up to 64 units can communicate with one another over this link, with the intelligent terminals acting as masters and the peripheral units, such as printers and tape units, acting as slaves. However, a terminal can also be programmed to operate as the slave of another terminal.

The terminal can also communicate over telephone lines to other terminals or a central computer site. The only change required to the terminal is the addition of a communications adapter and an external MODEM (modulator-demodulator), which connects to the telephone lines.

Models and Options Available

There is only one basic model of the 1501. However, several options are available that will increase the capabilities of the basic unit. These options are as follows:

a. Additional Cartridge Tape Drives. Two tape drives are standard. Normally, the program is loaded from the tape cartridge in drive 2, and the cartridge on drive 1 is used as an input, output, or scratch tape. Additional cartridge tape drives are available in pairs in the Model 1533 Dual Drive. This unit is a small desk-top unit that can be placed on a table alongside the 1501. Up to three Model 1533's can be used with each 1501. Note that the Model 1533 operates in parallel with the two standard tape drives and does not operate over the serial I/O channel.

b. Communications Adapters. In order to transfer data over communications lines, either to other Series 1500 systems or to central processing equipment, communications adapters must be added to the 1501. Of course, a MODEM is required to convert the adapter's output to the carriers required for the transmission.

Two types of communications adapters are available: (1) asynchronous and (2) BSC, which means binary synchronous communications. The adapter chosen depends upon the transmission techniques that will be used to communicate with the remote unit. The 1501 can accommodate one adapter internally, which can be installed at the factory or in the field.

Also available is an asynchronous communications adapter, designated the Model 1534-1 or Model 1534-2, that is a free-standing unit. This unit, which operates over the serial I/O channel, connects the Model 1501 to one communications line.

c. Dual Serial I/O Channel Option. The standard Model 1501 serial I/O channel provides for communication between the 1501 and up to 64 peripheral units and remote intelligent terminals. Available as an option are dual serial I/O channels. This feature allows the user to choose one of two serial I/O channels to be active at one time. The choice of which serial I/O channel is active is controlled by the program. Both channels have identical characteristics.

d. Booster-Isolator Option. Normally, the serial I/O channel cable can be up to 1000 feet long, which is sufficient for most applications. When the SIO cable must extend further than 1000 feet, a booster-isolator (Model 1537) must be inserted in the line.

Model 1501-40/43 Intelligent Terminal.

As shown in figure 1-5, the Model 1501-40/43 consists of a Model 1501 placed on a desk-like enclosure that houses a disc drive. The disc drive can be either a Model 1540, which houses a single disc that is not removable, or a Model 1543, which houses both a fixed disc and a removable disc pack. The terminal model number is simply a contraction of the basic 1501 and the disc unit number.

The Model 1501 used in this configuration is nearly identical to the standard 1501 described earlier. There are three basic differences:

a. Only one tape drive is standard; the second is optional.

b. Memory capacity is a standard 16,384 bytes.

c. A disc controller, mounted in the desk portion, operates with the processor, I/O controller, and memory of the 1501.

The disc controller coordinates the exchange of information between the disc and the processor and memory. It accepts instructions from the processor and carries out the exchange of information called for by the instruction.

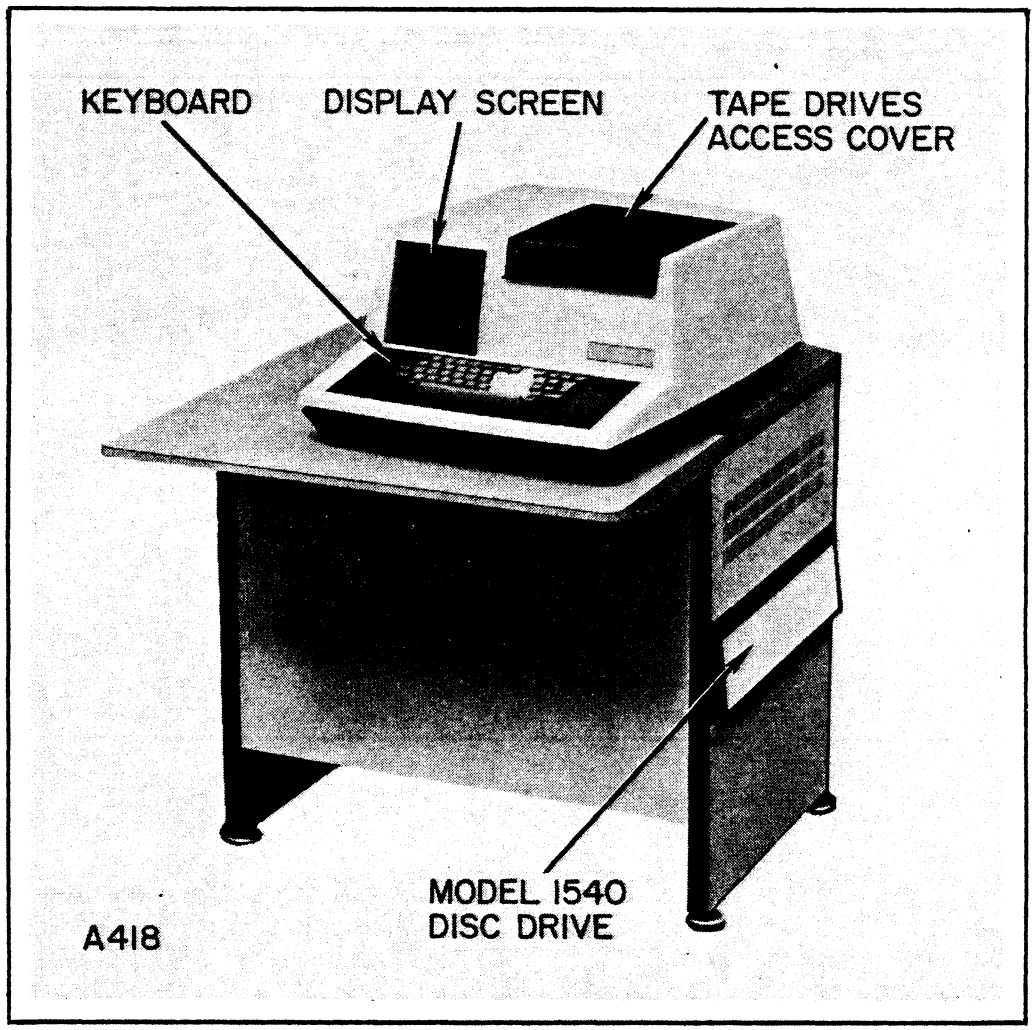


Figure 1-5. Model 1501-40 Intelligent Terminal

The disc drives available were mentioned above. Each disc has a storage capacity of 2.5 million bytes, and the details of each disc drive are given later in this section.

Model 1502 Intelligent Terminal.

Although the 1502, which is shown in figure 1-6, appears to be an entirely different terminal, it is very similar to the 1501 in capability. The processor, I/O controller, and memory that make up the mini-computer are housed in the desk-like enclosure. They are identical to the components in the 1501.

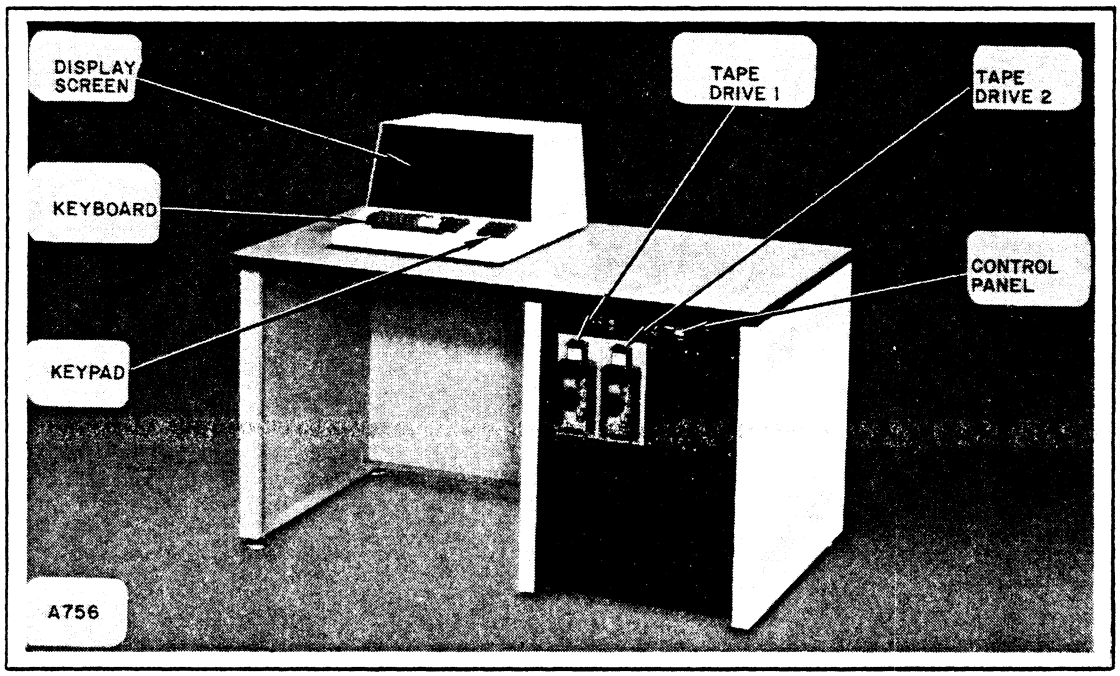


Figure 1-6. Model 1502 Intelligent Terminal

Two tape drives, which are operated by the computer, are mounted vertically in the front of the unit. Magnetic tape cartridges loaded on these drives provide programs and data for the computer and store data produced by the computer, as they do for the 1501.

To summarize the functional differences between the 1501 and 1502:

a. The 1502 has an optional large display screen.

b. The 1502 has a standard memory capacity of 16,384 bytes, and this is expandable to 65K.

c. Two tape drives are standard in the 1502, and two more are optional. The 1502 cannot operate the additional drives available in the Model 1533's.

A display screen and keyboard are housed in a separate chassis on top of the desk portion of the terminal. These components allow the operator to control the data processing for which the terminal is intended. The keyboard allows the operator to enter data and control operation of the computer, while the screen displays data, questions, and instructions to assist the operator. Unlike the 1501, however, either a small or large display screen is available with the 1502.

Controls

Located on a control panel above the tape drives are three controls used to apply power and start the processor. They are:

a. POWER on-off switch and FAULT indicator. The POWER pushbutton is a push-on/push-off switch that applies AC power to the terminal, and the FAULT indicator lights to indicate a drop in AC line voltage.

b. PROGRAM LOAD switch which, when pushed up to the PROGRAM LOAD position, causes the program on tape drive 2 to be loaded into the processor memory. This switch has three positions: PROGRAM LOAD, off (the center position), and program interrupt. The interrupt (down) position, which is not labeled, causes the processor program to jump to a specific memory location to perform an interrupt servicing routine, if one is stored there.

c. SYSTEM RESET switch, when depressed, which also causes the processor program to jump to a specific memory location and execute the program located there. The action that the processor takes when the SYSTEM RESET switch is used depends upon the program currently stored in the memory.

The Keyboard and Display

Combined in an assembly that rests on the desk top are a keyboard, a keypad, and a display screen. The display screen is available in two sizes. Up to 8 rows of 32 characters each can be shown on the small screen. Each character is formed on a 5-by-7 dot matrix and is selected by the processor. In the standard character set 64 different characters are available, including the alphabet, numerals, and punctuation marks. However, the processor program can create any character set on the display.

The large display screen is also available in the 1502. This unit provides up to 24 rows of 80 characters each. In addition, it offers a mode of operation in which it emulates the small screen display and thus allows programs and displays prepared for the small unit to be used. Characters are formed in the same manner as on the small screen, and a similar flexibility in character sets is available.

Series 1500 Intelligent Terminals, including the Model 1502, are equipped with a standard "keypunch arranged" keyboard, as shown and described earlier in the 1501 description. The keyboard consists of a 52-key alphanumeric keyboard, and a 13-key keypad. A speaker is housed below the keyboard assembly to provide audible cues to the operator.

Cartridge Tape Drives and Tapes

Standard in the Model 1502 are two cartridge tape units. Another two units can be installed at the factory or in the field to bring the total to four mechanically-independent tape units. The drives are identical to those used in the 1501.

Memory Capacity

Standard in the 1502 is a memory with a storage capacity of 16,384 bytes. (Each byte is made up of 8 bits, and all data within the 1502 is handled in bytes.) A memory module in the 1502 has slots for four memory boards. In the standard 1502, only one of these slots is filled.

Options Available

There are several options available for the Model 1502 that will increase the capabilities of the basic unit. These options are as follows:

a. Additional Cartridge Tape Drives. Two tape drives are standard in the Model 1502. Normally, the program is loaded from the tape cartridge in drive 2, and the cartridge on drive 1 is used as an input, output, or scratch tape. One or two optional tape drives can be installed alongside the standard drives, either at the factory or in the field. This brings the total number of cartridge tape drives that can be operated by the Model 1502 to four.

b. Increased Memory Capacity. Many applications of the 1502 can be made more efficient if the memory available is increased; therefore, additional memory is available as an option. Since the memory is divided into groups of 16,384 bytes of storage capacity, the memory can be increased to 32K or 65K from the standard 16K. This option is also either factory or field installed.

c. Display Options. Either a large or small display is available, as discussed earlier in this description.

d. Other options are the same as those available with the 1501.

Model 1503 Intelligent Terminal.

The Model 1503 is identical to the Model 1502 except that it includes a disc controller and can operate up to four disc drives. These units may be either Model 1540 or Model 1543. The first sits on the right side of the desk, as shown in figure 1-7, and the remaining three occupy a separate cabinet standing alongside the 1503. The disc controller and the disc drives are the same as those used by the 1501-40/43 described earlier in this section.

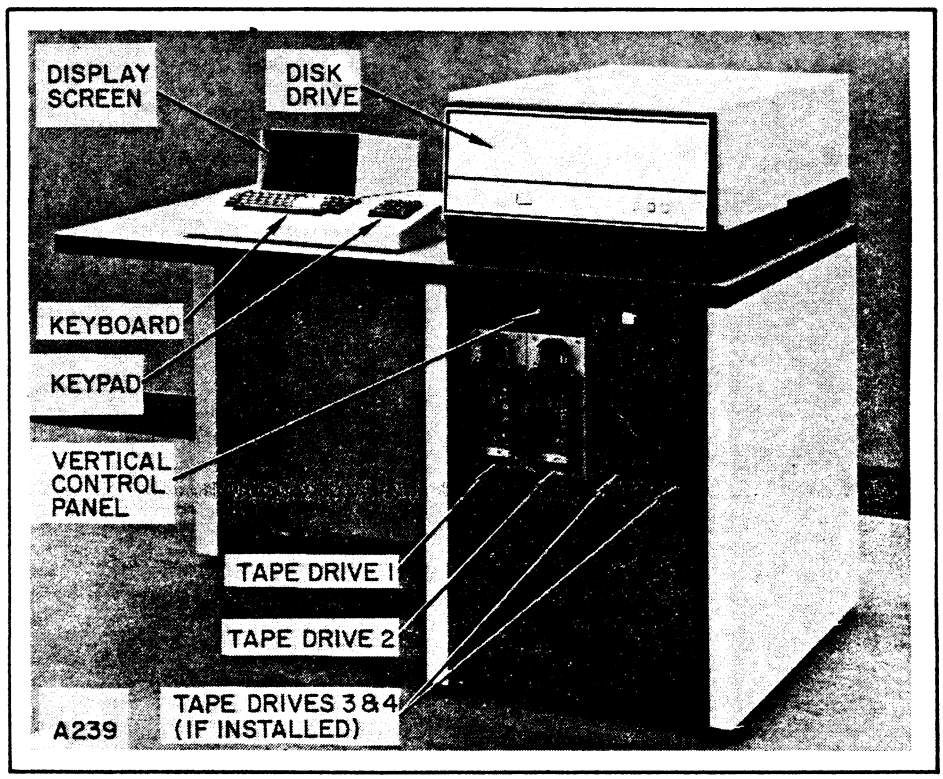


Figure 1-7. Model 1503 Intelligent Terminal

When either disc unit is added to convert a Model 1502 to a Model 1503, a disc controller is installed in the main chassis to coordinate the exchange of information between the disc and the processor and memory. The disc controller accepts instructions from the processor and carries out the exchange of information called for by the instruction.

As mentioned above, there are two disc units available, the Model 1543 and the Model 1540. The Model 1543 houses two discs, one removable and one fixed, while the Model 1540 has only a single, fixed disc. Each disc is capable of storing about 2.5 million bytes of data. Up to three additional drives of either model can be added.

Model 1504 Intelligent Terminal.

In outward appearance, the Model 1504, which is shown in figure 1-8, closely resembles the 1502 and 1503. Internally, however, the design is totally different.

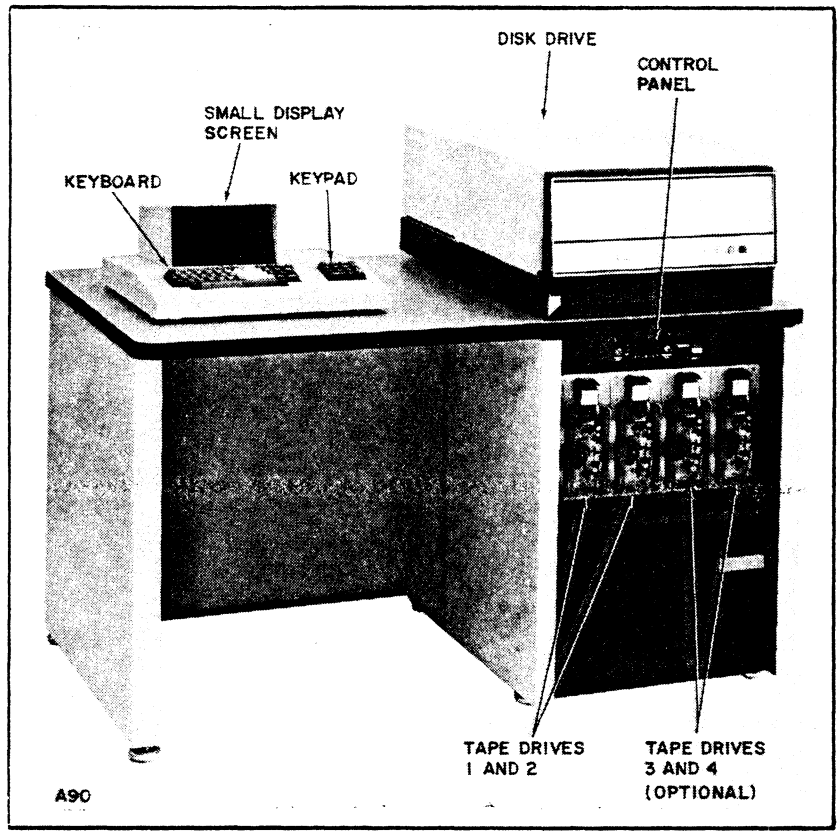


Figure 1-8. Model 1504 Intelligent Terminal

The Model 1504 is a multi-processor; that is, the processor can handle several different tasks nearly simultaneously. Since the input/output devices are much slower than the processor, there could be idle time while the processor waited for the I/O data transfer to take place. In the Model 1504, the processor takes advantage of this time by performing other tasks and returning to the I/O device only when it requires attention.

The 1504 is "downward" instruction compatible with the 1501, 1502, and 1503 terminals; that is, it can execute programs written for these terminals. It also uses the same cartridge tapes, serial I/O channel, and keyboard and display. Options include two additional tape drives, a large display screen, and up to four disc drives.

A key option in the 1504 is an input/output system with the direct execute sequence (DES) feature. The DES I/O channels control up to 64 low-speed input/output devices and can gain immediate access to the processor.

Controls

The control panel on the 1504 is similar to that used for the 1502 and 1503. It has the SYSTEM RESET, PROGRAM LOAD, and POWER switches, which perform the same functions as the 1502/1503 controls; however, the control panel used in the 1504 also houses three rows of indicators. These indicators are intended for the use of the field service representatives or programmer. Their purpose is to indicate which of four computer programs is selected and which section of the memory is being used.

Organization of the Mini-Computer

Operation of the Model 1504 Intelligent Terminal is controlled by a computer consisting of a general-purpose processor, memory, and I/O controller. It is a stored-program unit with an instruction repertoire that includes the standard Series 1500 instruction set plus 11 additional instructions for processor control. Standard memory capacity is 32K, and memory capacity is expandable up to 65K.

An I/O controller is provided to handle data transfer between several I/O devices and the processor. This unit operates the cartridge tape drives, the CRT and keyboard assembly, and a serial input/output channel.

When the disc unit is added to the Model 1504, a disc controller is installed in the main chassis to coordinate the exchange of information between the disc and the processor and memory. The disc controller accepts instructions from the processor and carries out the exchange of information called for by the instruction.

Available as an option is the direct execute sequence (DES) controller. This group of circuits operates low-speed communications channels. When a channel requires attention, it can gain immediate access to the processor, hence the name "direct execute" is given to this option.

The Model 1504's processor and memory are organized as if they were four separate functional processors. Each is called a virtual processor (VP), is dedicated to a specific task, and has the following priority:

a. VP1 has the highest priority and can be assigned to the user's primary data processing operation.

b. VP2 is assigned to operate the disc units and has the second highest priority.

c. VP3 is the channel background processor and the DES channel processor. When used as the background processor, it has the third highest priority. However, when used as a DES channel processor, it has the highest priority of all processors, including VP1.

d. VP0 is called the host processor. It is active only when no other processor is in use, and it performs the "housekeeping" routines.

Models and Options Available

There is only one basic model of the 1504. Available as options are many of the same options available with the 1502/1503, as follows:

a. Two additional cartridge tape drives.

b. A small or large display screen.

c. Communications adapters.

d. Model 1540 and 1543 disc units.

e. A dual serial I/O channel.

f. Additional memory capacity.

Mentioned earlier in this description was the DES option. This option consists of a DES controller and up to five channel boards, which can be used in any combination. The channel boards available are:

a. The utility option, which has 8 channels.

b. The quadruple asynchronous communications channel option, which has four channels and can operate with a MODEM.

c. The 16-channel local asynchronous communications option, which cannot operate with a MODEM.

All of these options can be used in any combination, provided that the total number of channels does not exceed 64 and that there is only one utility option.

Model 1580 Intelligent Terminal.

The 1580 consists of a mini-computer, a large display screen, and a keyboard, as shown in figure 1-9. It was originally designed to operate with a Model 1504 Intelligent Terminal and communicate with the 1504 over an asynchronous communications link. While it can be programmed to perform a wide variety of tasks, it is not intended to operate independently. Normally, it operates with a host computer, connected by the aforementioned communications link.

Within the terminal are the processor, memory, input/output controller, and a CRT controller. Optional communications or input/output interfaces can be installed to increase the capabilities of the 1580.

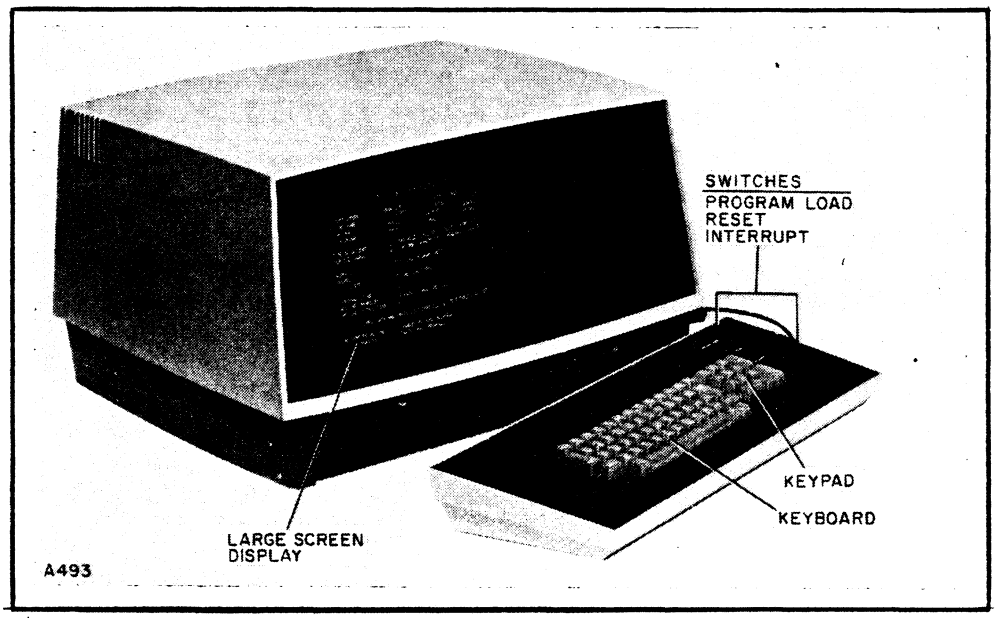


Figure 1-9. Model 1580 Intelligent Terminal

The CRT display and keyboard allow the operator to control the data processing for which the terminal is intended. The keyboard allows the operator to enter data and control operation of the computer, while the CRT displays data, cues, questions, and instructions to assist the operator.

The keyboard assembly houses the PROGRAM LOAD, RESET, and INTERRUPT switches. After power is supplied to the terminal, the application program is loaded into the processor memory when the PROGRAM LOAD switch is depressed. The source of the program depends upon the application of the 1580; however, the program must come from the host processor, whether local or remote.

As previously mentioned, the terminal can communicate with other local terminals. With the addition of an external MODEM (modulator-demodulator), the terminal also has the capability to transmit and receive over telephone lines. This option provides the capability for information to be communicated to a central computer.

Notice that the Model 1580 does not include cartridge tape drives; none are available as an option. In addition, the 1580 cannot accommodate a disc controller and operate the disc units.

Another major feature of the Series 1500, the serial I/O channel, is also not available. This prevents the 1580 from operating any of the normal Series 1500 peripheral equipment.

A typical application is several Model 1580 terminals operating with one Model 1504. The Model 1580 would operate as a data entry pre-processor for a Model 1504 Intelligent Terminal.

The benefits of using the Model 1580 and Model 1504 in this configuration are as follows: Data can be processed by a host computer at a far greater speed than data can be transmitted or entered by an operator. In addition, it is not practical to interrupt the host computer every time data is available, transactions are to be recorded, or some processing is required, particularly when the host is operating with a large number of data entry terminals. Therefore, it is economically expedient to add a pre-processor to handle the direct operator inputs. This is the function of the Model 1580. It cues the operator as to the data required, displays the data, and presents questions and instructions to assist the operator. The data entered is then checked and processed within the 1580 before being sent to the host.

This processing network is heavily dependent upon the data communications between the host and the 1580. Programs that control operation of the 1580 are held by the host and transferred over the communications lines to the 1580 when the operator depresses the PROGRAM LOAD switch.

Controls

The Model 1580 Intelligent Terminal has relatively few operator controls aside from the keyboard and keypad. The additional controls and indicators are as follows:

a. The power on/off switch is located at the rear of the unit, in the bottom right corner.

b. The PROGRAM LOAD switch, located in the upper right corner of the keyboard assembly, causes the programs to be loaded into the processor memory.

c. The RESET switch, located in the upper right corner of the keyboard assembly, causes the program to jump to a specific memory location. The action the processor then takes is determined by the program stored in memory.

d. The INTERRUPT switch, located in the upper right corner of the keyboard assembly, causes the program to jump to a specific memory location to perform an interrupt servicing routine.

e. The power on indicator is a red dot visible on the right side of the screen when power is applied.

Organization of the Mini-Computer

Operation of the terminal is controlled by a computer contained within the terminal. This computer consists of a processor, a memory, and an input/output controller. The processor is a stored program, general-purpose machine with a repertoire of 38 instructions, most of which are the same as the instructions used in other Series 1500 terminals. The 1580 memory is available with a capacity of 8,192 bytes or a maximum of 16,384 bytes.

The input/output controller coordinates and handles the data transfer between the processor and the keyboard and I/O devices. Included in the controller is an interrupt network that handles interrupts from several sources and coordinates interrupt servicing with the processor. In addition, the I/O controller contains a read-only memory that determines the source from which the programs are loaded into the 1580.

The CRT controller produces the visual display on the 12-inch CRT housed in the terminal. It generates up to 24 lines of 80 characters each on the display. A character ROM provides the dot patterns for the characters displayed.

Models and Options Available

There are two models of the 1580 available: the 1580 and the 1580-S. The 1580-S is a simplified version in which one fan is deleted and some circuits are combined, but both models perform the same functions.

The only option available is an asynchronous communications channel and related cables. Two such channels can be installed in the 1580 and one in the 1580-S. (One channel is standard in the 1580-S.) Each channel can be connected so as to conduct either local or remote asynchronous communications. Of course, a MODEM must be added when remote communications capability is required.

1-3. TAPE DRIVES.

The desk-mounted terminals (1502, 1503, and 1504) are each capable of housing four cartridge tape drives. This number is expected to be sufficient for all normal applications of the terminals. Since the 1501 can house only two drives, however, it may be necessary to add the Model 1533 Dual Drive to operate with the 1501.

The Model 1533, which is shown in figure 1-10, houses two cartridge tape drives, identical to those used in the terminals. These drives are operated by the I/O controller in the 1501, which is capable of operating a total of eight drives, including the two housed in the 1501.

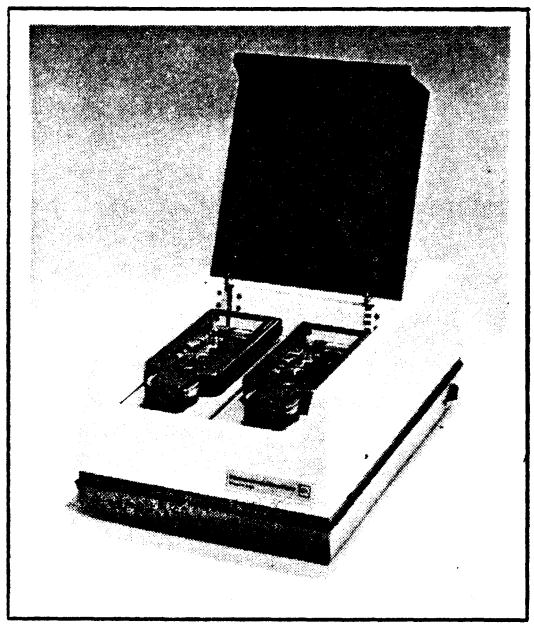


Figure 1-10. Model 1533 Dual Drive

The tape cartridges used with the Series 1500 terminals are capable of holding data files; however, they are exclusive to the Series 1500 equipment and cannot be used directly by other computer systems. In addition, it may be desirable to have a large data file recorded on one reel of tape rather than on several tape cartridges. In order to allow the Series 1500 terminals to handle standard tape reels, the "maxi-tapes" are available as peripheral equipment. These units, which are operated over the serial I/O channel, include Models 1511, 1512, 1513, and 1514.

All four models use 1/2-inch tape on standard reels. The basic differences are listed in table 1-3, and include the following:

a. Number of recording tracks (7 or 9).

b. Recording density (556, 800, or 1600 bits per inch).

c. Maximum reel size (7 or 10.5 inches).

d. Data encoding method (NRZI or phase encoding).

Table 1-3. Maxi-Tape Drive Characteristics

Characteristic	Model			
	I511	1512	1513	1514
Maximum Reel Size Inches/Centimeters	7/18	7/18	10.5/27	10.5/27
Reel Capacity	600 ft.	600 ft.	2400 ft.	2400 ft.
Number of Read/Write Tracks	7	9	9	9
Data Read/Write Density Bits Per Inch	556	800	800	1600
Record Size		(Program Controlled)		
Average Data Transfer Rate Characters Per Second	6950	10,000	10,000	20,000
Read/Write Speed (ips)	12.5	12.5	12.5	12.5
Encoding Method	NRZI	NRZI	NRZI	P.E.
Rewind Speed Inches Per Second	50	50	150	150

Model 1511 and 1512 Tape Units.

The Model 1511 Tape Unit, which is shown in figure 1-11, uses half-inch magnetic tape mounted on reels of 7 inches (18 centimeters) or smaller diameter. Its recording density is 556 bits per inch, and it is available with odd or even parity, although odd parity is standard. The recording format is seven tracks, with data encoded by the NRZI method. The 1511 is a compact unit that can be used on any desk or table top. Physically, the Model 1512 is the same as the Model 1511, but it records in a density of 800 bits per inch in a 9-track format.

Model 1513 and 1514 Tape Units.

A Model 1513 Tape Unit accommodates tape reels up to 10.5 inches (27 centimeters) in diameter. It has a recording density of 800 bits per inch in a 9-track format with NRZI encoding. The unit, shown in figure 1-12, is suitable for desk or table top use. Its ability to handle the largest standard tape reels distinguishes it from the smaller Model 1512.

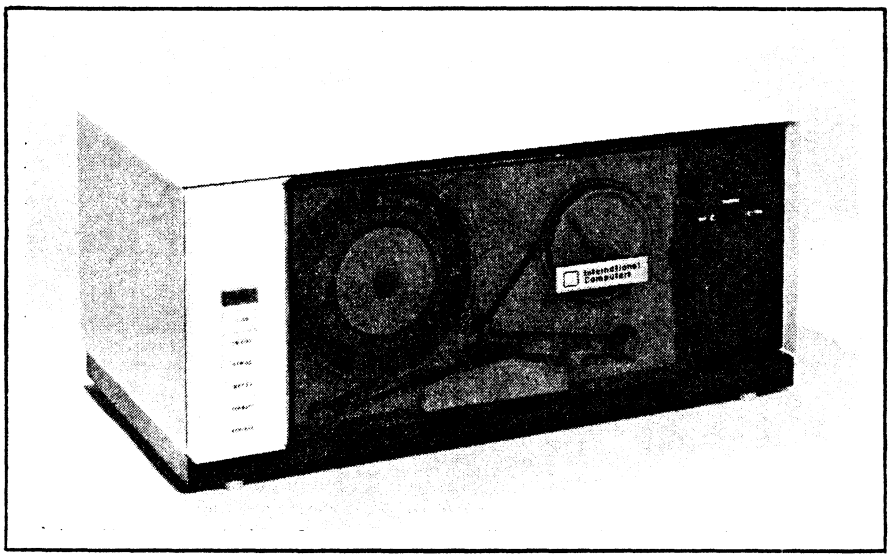


Figure 1-11. Model 1511 or 1512 Tape Unit

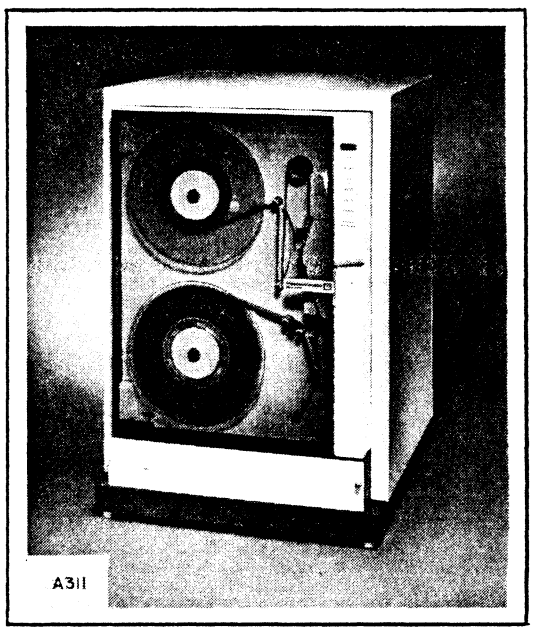


Figure 1-12. Model 1513 or 1514 Tape Unit

The Model 1514 is about the same size as the 1513, and it accepts tape reels up to 10.5 inches (27 centimeters) in diameter. However, the 1514 recording density is 1600 BPI. The phase encoding technique is employed in a 9-track format.

1-4. PRINTERS.

Characteristics of the seven Series 1500 printers available are listed in table 1-4. As in the case of the tape drives, the intelligent terminals operate the printers over their serial input/output channels.

Table 1-4. Printer Speeds

Printer Model	1525	1551S	1551D	1553	1554	1555	1556
Speed (Lines per minute)	30 CPS*	60	125	125	200	300	400

*Characters per second

Model 1525 Printer.

The Model 1525 Printer is intended to be used where the volume of printing is low to provide economical printing of reports, forms, and communications messages. (A photograph of the Model 1525 Printer is shown in figure 1-13.)

The printer uses a dot matrix technique, and achieves a printing speed of 30 characters per second. Lines may be up to 132 characters wide. Horizontal spacing is 10 characters per inch, and vertical spacing is 6 lines per inch. Printing and form control functions are under control of the intelligent terminal; there is no form format tape provided in the printer itself.

A unique feature of the 1525 is its ability to handle low-cost journal roll paper (as used for teletype) as well as pin-fed forms. A friction platen accepts continuous journal roll paper in varying widths from 7 to 15 inches. Continuous pin-feed paper is also accepted in up to six-part copies ranging from 8.5 to 15.5 inches.

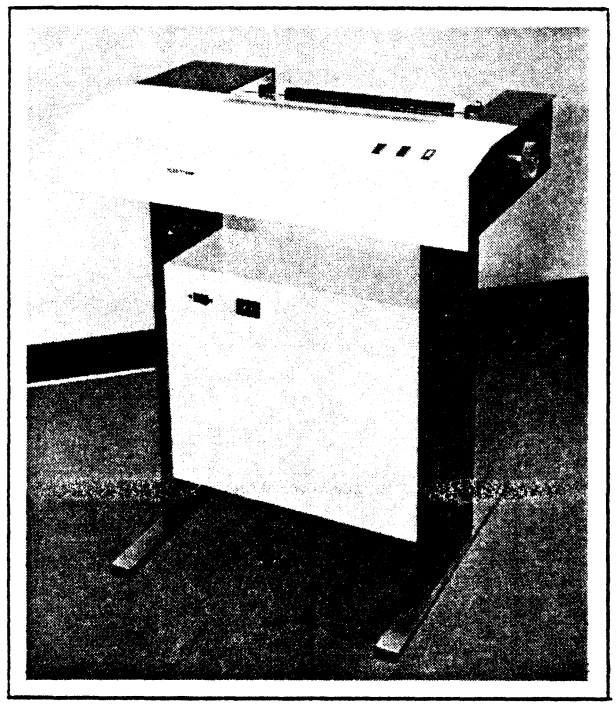


Figure 1-13. Model 1525 Printer

Model 1551 Printer.

A photograph of the Model 1551 Printer appears in figure 1-14. Two models of this printer are available: a Model 1551S employs a single print head and prints 60 lines per minute, while the 1551D prints 125 lines per minute using dual print heads.

A Model 1551 printer employs a dot matrix printing technique and forms each character on a matrix of 5 by 7 dots as the print head crosses the carriage. Both models are capable of printing up to 132 columns per line, with horizontal spacing of 10 characters per inch and vertical spacing of 6 lines per inch.

The vertical spacing is controlled by a two-channel vertical format tape in the printer, which operates the paper feeding mechanism. In turn, commands from the intelligent terminal determine when the vertical format tape is used.

The 1551 is easily adjusted to accommodate various sizes and thicknesses of forms. It can accommodate pin-fed forms up to 14.87 inches (38 centimeters) wide. A paper stacker collects printed forms behind the carriage, so that tables, boxes, or other special provisions are not required.

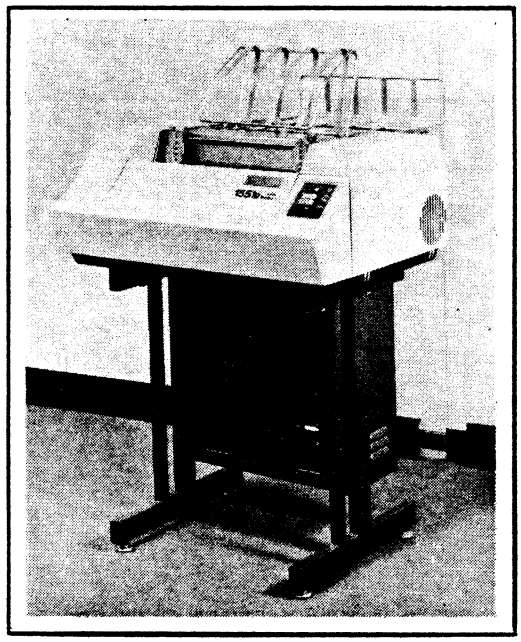


Figure 1-14. Model 1551 Printer

Model 1553, 1554, 1555, and 1556 Printers.

These line printers, which are shown in figure 1-15, offer printing speeds from 125 lines per minute to 400 lines per minute. They are identical in appearance and provide the printouts on one-to-six-part, fan-folded paper.

All four printers print on a line-by-line basis, with the line consisting of up to 132 columns, horizontally spaced at 10 characters per inch. The vertical line spacing is 6 lines per inch, with an 8-line-per-inch option. Vertical spacing is controlled by the standard 2-channel, or optional 12-channel, vertical format unit, which is connected to the paper feeding mechanism. The standard 64 USASCII character set is used, with an optional numeric OCRA/OCRB character type within this set.

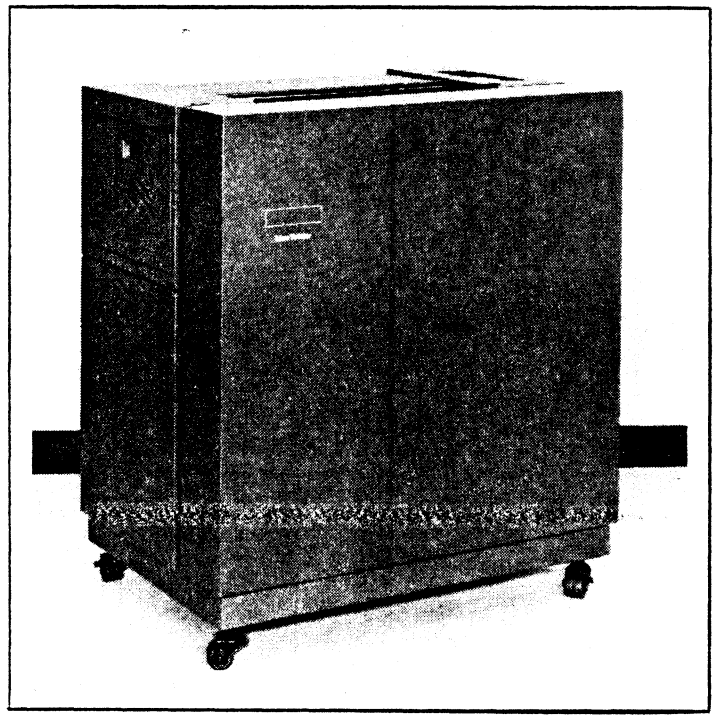


Figure 1-15. Model 1553, 1554, 1555, or 1556 Printer

1-5. DISC DRIVES.

There are two disc drives available with the Series 1500 Intelligent Terminals. They are the Model 1540, which houses one non-removable disc, and the 1543, which houses both a non-removable disc and a replaceable disc pack.

The disc pack is a type 5440. It is a top-loading, wide slot, with 24 sectors.

The disc in the Model 1540 is about 14 inches in diameter with a capacity of 2.5 million bytes and an average access time of 90 milliseconds. It is organized into tracks and sectors for addressing purposes. There are 203 data tracks on each side of the disc, a total of 406 tracks. Each track is divided into 24 sectors, with each sector containing 256 bytes of data.

The disc is under control of the program stored in the terminal. The processor issues a command to the disc controller; the controller, in turn, executes the command to locate the proper track and sector. During the time period that the controller is locating the sector, the processor continues to execute other non-disc instructions.

Upon locating the sector, the disc controller interrupts the processor, at which point the processor either accepts the data or sends the data out. When reading from the disc or writing onto the disc, the data will always go into or come from a buffer area in memory. The minimum memory size of a terminal using a disc is 16K, with 8K being occupied by the Data Base Manager Program, which is a program that handles the organization of disc data.

As mentioned above, the Model 1543 houses two discs. The only difference in organization of the discs themselves is that the discs in the Model 1543 have 408 tracks on each disc.

The Model 1540 Disc Drive has no external controls. Power is applied when the terminal is turned on, and the disc starts automatically. Since the Model 1543 has a replaceable disc pack, it does include operating controls, which are as follows:

a. LOAD/RUN Switch . . . . . Determines mode of operation of disc unit. The LOAD mode allows a disc pack to be loaded; the RUN mode allows the computer to use the disc.

b. LOAD Indicator. . . . . Indicates disc is in the LOAD mode. The holding clamps may be moved and a disc pack may be loaded or unloaded.

c. READY Indicator . . . . . Indicates disc is in the RUN mode and ready for operation by the computer.

d. CHECK Indicator . . . . . Indicates a low voltage problem in the disc unit.

e. PROTECT Switch/Indicator. . Indicates that the disc cannot be written on. The PROTECT switch/indicator is depressed to change the write protect condition.

1-6. COMMUNICATIONS UNITS.

The Series 1500 line includes six communications adapters: Models 1534, 1534-1, 1534-2, 1535, 1536, and 1539. These adapters permit the Series 1500 Intelligent Terminals to operate with data communications equipment, thus enabling the terminals to communicate over long distances. Also required are MODEMS (modulator-demodulators) that actually send and receive over the long distance lines.

Three of these communications adapters (1534, 1535, and 1539) are printed circuit boards. They are installed internally in an intelligent terminal and cabled to the communications equipment. Most of the terminals can accommodate one such board.

When an optional adapter is installed in an intelligent terminal, that terminal can operate one communications line. There are, however, applications of the terminals in which several lines must be operated by one terminal or in which several terminals require access to one communications line. In this case, the Model 1534 Asynchronous Communications Adapter is combined with other circuits and housed in a free-standing unit. This unit operates one communications line. However, the intelligent terminals can communicate with the adapter over their serial input/output channels, and this allows many adapters to operate with each terminal. When the adapter is packaged in a free-standing unit, it is known as the 1534-1 or 1534-2. (See figure 1-16.)

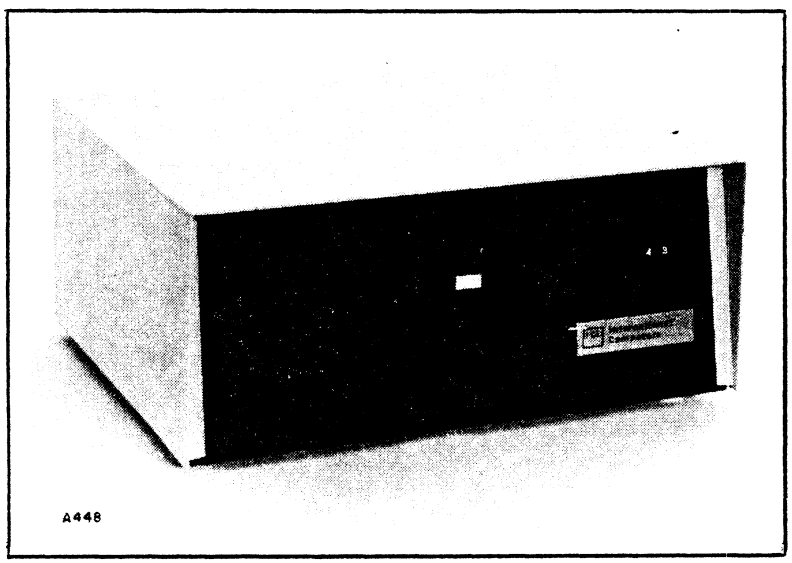


Figure 1-16. Model 1534-2 Communications Adapter

Model 1534 Asynchronous Communications Adapter.

This option provides an asynchronous communications channel between one Series 1500 Intelligent Terminal and a MODEM. A supervisory (secondary) channel is available for control operations.

Asynchronous communication is a start/stop form of communication requiring one synchronizing "start" bit and at least one "stop" bit to accompany each data word that is transmitted. Data flow between the 1534 adapter and the MODEM is in serial form, and transmissions can be conducted in the half-duplex or full-duplex mode.

The 1534 adapter complies with the EIA RS-232-C and CCITT V24 standards. It is available with a communications cable to connect a type 202 MODEM or an RS-232-C MODEM.

Model 1534-1 and 1534-2 Asynchronous Communications Adapters.

For installations that require more than one asynchronous communications line, the Model 1534-1 connects to the Series 1500 serial input/output cable. Otherwise, it operates identically to the Model 1534.

Where a Series 1500 system is used in a TELEX application, the Model 1534-2 adapter is required. It connects to the Series 1500 serial input/output cable and, except for the current loop capability, is identical to the Model 1534-1. The current converter in the 1534-2 is designed to operate over relatively short distances at low speeds.

Model 1535 Binary Synchronous Communications (BSC) Adapter.

This option provides a synchronous communications channel between one Series 1500 Intelligent Terminal and a MODEM. A supervisory (secondary) channel is available for control operations.

Synchronous communications is a form of continuous communication in which the sending and receiving units maintain synchronization for the duration of the transmission. Data flow between the 1535 adapter and the MODEM is in serial form.

If the MODEM generates an external clock, the operation is completely timed externally. If the MODEM does not generate a clock, then the 1535 generates, under program control, either a 600 baud, 900 baud, 1200 baud, or 1800 baud clock.

The 1535 adapter complies with the EIA RS-232-C and CCITT V24 standards. Its standard communication cable is compatible with a Type 202 MODEM.

Model 1539 Interruptible Communications Controller.

This option is capable of any one of three forms of communications: (1) asynchronous, (2) BSC, and (3) SDLC, which means synchronized data link communications. Both the BSC and SDLC modes are a form of synchronous communication.

Data flow between the 1539 and the MODEM is in serial form. In the asynchronous mode, the data has the same form as the 1534. In the BSC mode, characters of 5 to 8 bits can be used and a transparency feature is provided. In the SDLC mode, the 1539 uses 8 bits per character and is capable of using NRZ form with MODEMS that have no clock.

The 1539 meets the RS-232-C and CCITT V24 standards, and is thus capable of operating with RS-232-C or Type 202 MODEMS. Transmission rates that can be selected by the program range from 50 to 19,200 baud.

Model 1536 Intelligent Communications Adapter.

This free-standing unit is called an "intelligent" communications adapter because it includes a processor, memory, and I/O controller to manage communications. This unit is primarily a communications buffer. A photograph of the Model 1536 is shown in figure 1-17.

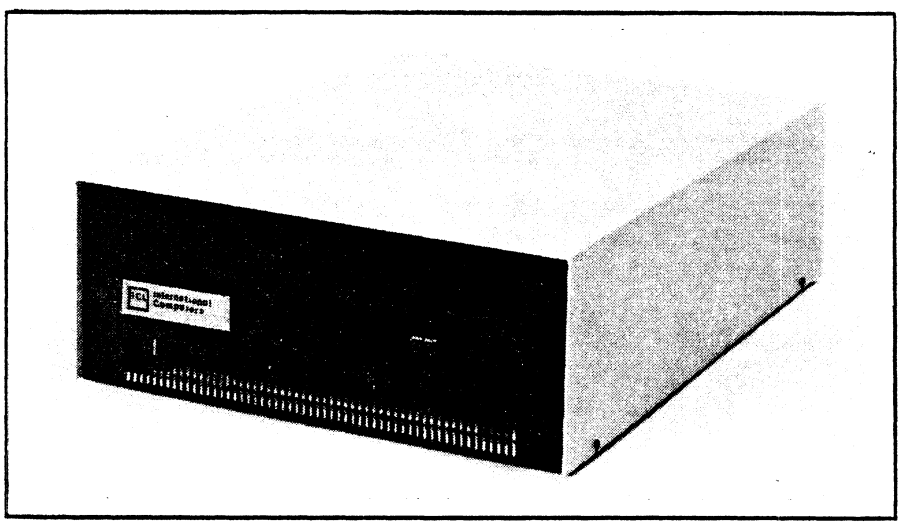


Figure 1-17. Model 1536 Intelligent Communications Adapter

Installed in the 1536 is the Model 1539 Interruptible Communications Controller (ICC) described earlier. The 1536 uses the ICC to conduct long distance communications. Also available in the 1536 is an asynchronous communications channel similar to the Model 1534 described above. Normally, the 1536 uses this link to communicate with a host processor, such as the Model 1504 Intelligent Terminal.

The 1536 Intelligent Communications Adapter is designed to buffer communications data and transfer it in blocks to or from a host processor. This data transfer is accomplished via an asynchronous communications link. The 1536 can also transmit over the asynchronous communications link through a MODEM to a remote host processor. This link is programmed to conform to standard line disciplines at rates ranging from 600 to 19,200 baud.

As mentioned above, the Model 1536 includes a Model 1539 ICC, which allows communication with other data processing equipment over long-distance communications lines. The 1539 can communicate either half-duplex or full-duplex in any one of three modes: asynchronous, BSC, or SDLC.

The processor, memory, and I/O controller used in the 1536 are similar to those used in the intelligent terminals. The processor is a stored program, general-purpose machine with a repertoire of 38 instructions, most of which are the same as the instructions used in the Series 1500 terminals. The memory capacity is 8,192 bytes, and is expandable to 16,384 bytes.

The input/output controller coordinates and handles the data transfer between communications devices and the processor. In addition, it contains a read-only memory that determines the source from which the programs are loaded into the 1536.

A dual 1536 is also available. It contains two processors, memories, and communications adapters, as well as the associated cabling and switches.

1-7. OTHER PERIPHERAL UNITS.

The remaining peripheral units available are as follows:

a. Model 1531 Card Reader. The 1531 is a desk-top size reader that operates at 300 cards per minute. It accepts the standard data processing card and transfers data to an intelligent terminal over the serial I/O channel.

b. Model 1517 Paper Tape Reader. This unit is offered to allow an intelligent terminal to process data encoded in 5- to 8-channel, punched paper tapes. It reads up to 300 characters per second and operates on the serial I/O channel.

c. Model 1590 MDTS Poller. MDTS means Modular Data Transaction System, which is an electronic cash register. Each 1590 accepts transaction data from up to 12 cash registers and transfers it over the serial I/O channel to an intelligent terminal where it is processed.

d. Model 1538 Automatic Dial Unit. This unit also operates over the serial I/O channel. When connected to the appropriate communications unit, such as the Bell 801 Auto-Call, it allows an intelligent terminal to call a telephone number. Normally, this would be a data line to another data processing system.

Model 1531 Card Reader.

The Model 1531 Card Reader expands the capability of the Series 1500 Intelligent Terminals to include the reading of standard data processing cards. It is adapted to read 80-column, 96-column, and 51-column cards at a speed of 300 cards per minute. The input hopper and output stacker have a capacity of 1000 cards each.

As mentioned earlier, the 1531 is connected to the terminal via the serial I/O channel, and the functions of the reader are controlled by the terminal. When a read command is sent to the reader, a card is read and the data stored in a buffer in the reader. The data is then transferred, under program control, a byte at a time, to the terminal. Both Hollerith and 96-column card codes are converted to USASCII in the reader, relieving the programmer of the task.

Model 1517 Paper Tape Reader.

Also available is the Model 1517 Paper Tape Reader, which operates over the serial I/O channel like most peripherals. The 1517 is capable of reading 5, 6, 7, or 8-channel paper tape, at 300 characters per second. Tape width can be 11/16, 7/8, or 1 inch (1.75, 2.2, or 2.5 cm), and the thickness ranges from 0.0025 to 0.0040 inches (.0635 to .1016 mm).

Control of the reader is accomplished by the terminal. When a read command is sent to the reader, one character code is read from tape; the number of bits in the code depends on the number of channels on tape. The data is sent from the reader to the terminal, where the program must perform the translation and parity checking. The program is also responsible for handling the status and error recovery.

Model 1590 MDTS Poller.

The Model 1590 MDTS Poller provides the means by which the Series 1500 Intelligent Terminals can collect data from the Series 900 Modular Data Transaction System (MDTS) electronic cash registers. Once the data is gathered by the intelligent terminal, it can be processed and recorded on magnetic tape, printed, or transmitted to a central data collection point over telephone lines.

Two models of the MDTS Poller are available: The Model 1590 and the Model 1590-1. The Model 1590 is capable of operating with a maximum of six MDTS units and the 1590-1 is capable of handling 12 MDTS units. Both the 1590 and 1590-1, operating in the slave mode, are connected to the serial I/O channel of an intelligent terminal.

The terminal controls the 1590 operation. It scans the inputs from the cash registers by supplying a port selection address to the 1590. If an MDTS service request is active at the addressed port, the 1590 automatically acknowledges the request, accepts the incoming message, and

stores it in a buffer. When the end of the MDTS message is detected, the 1590 sends a status word to the intelligent terminal.

The terminal responds by sending an unload command to the 1590, which forwards the MDTS message to the terminal. A return message, that is, a message from the terminal to the MDTS, is then loaded into the 1590 and sent to the MDTS.

The 1590 is connected to an MDTS terminal by a twisted pair, two-wire cable. The input transmission rate (MDTS-to-1590) is 3600 baud, while the output transmission rate (1590-to-MDTS) is 120 baud.

Model 1538 Automatic Dial Unit.

The Model 1538 Automatic Dial Unit expands the capability of the Series 1500 Intelligent Terminals to include automatic dialing. In normal applications, the host processor or terminal will be equipped with auto answering. However, auto answering is not a requirement.

The auto dial feature gives the terminal the capability of establishing a telephone connection, and then through a Model 1534 or 1535 Communications Adapter, transmitting data while unattended by the operator. Normally, this procedure will be utilized at night, when the telephone rates are less expensive.

The Model 1538 is strictly a dialing feature; no data is transferred through it. Therefore, a Model 1534 or 1535 Communications Adapter is also required to complete the communications link. The communications adapter used must be internally installed in the terminal and not the units that operate on the serial I/O channel.

1-8. AN INTEGRATED SYSTEM.

Up to this point, the features of each unit and the functions performed by each unit have been discussed. Next to be described is how the units operate together in an integrated system.

Since the Model 1501 can perform most of the functions that any of the terminals can perform, it has been chosen as the basic terminal for the purposes of this discussion. The additional features of the larger terminals are then described.

Figure 1-18 is a diagram in which one unit of each model in the Series 1500 line is shown. The Model 1501 is shown on the left. From there, one serial I/O channel cable is shown connected to one of each kind of unit that can operate over this channel.

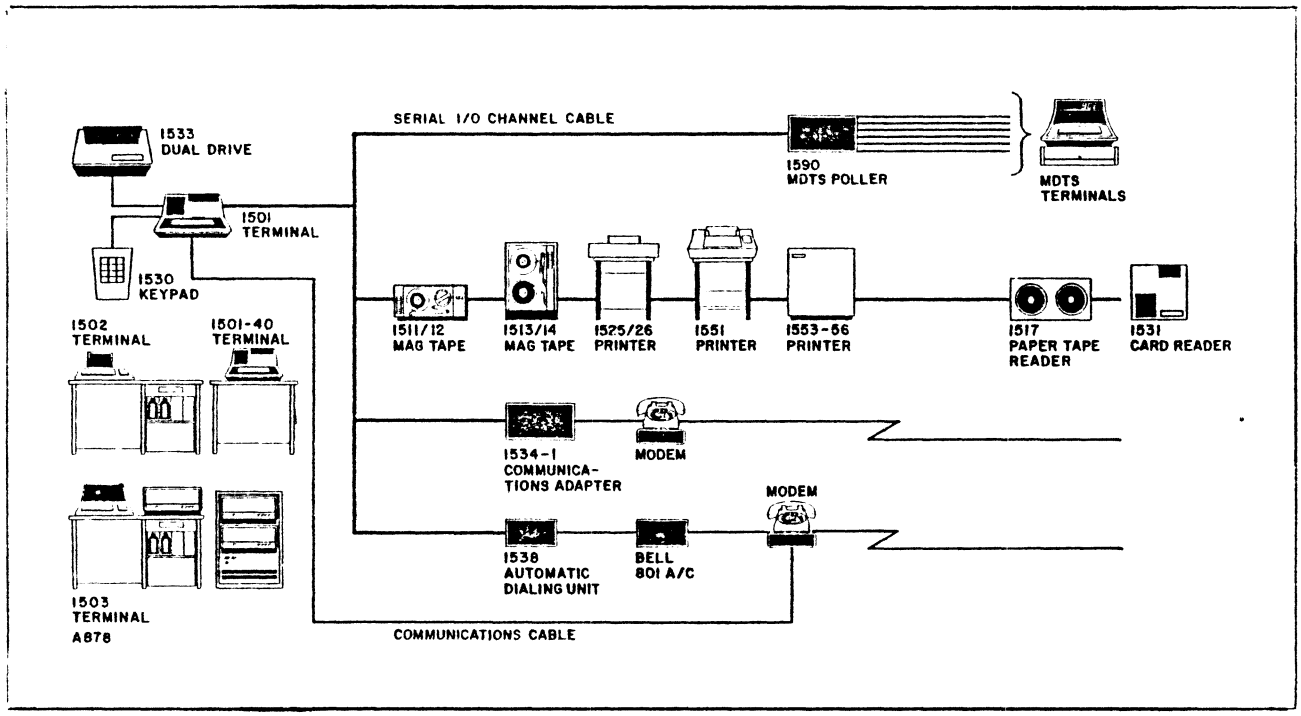


Figure 1-18. An Integrated System

As described earlier, this channel is a coaxial cable available in a variety of lengths. Ten or 15 feet is a typical length. In the figure, one cable would connect the 1501 to the 1511/1512 tape unit at a "T" connector in the back of the tape unit. Then, another cable would couple the 1511/1512 to the 1513/1514, etc. At the last unit in the physical string, a terminator is placed on the "T" connector.

Although the units are connected in series physically, all have equal access to the line, and they are selected by a 2-digit address. An internal plug or an external thumbwheel switch establishes the unit address.

Shown in a line across the center of the figure are the common peripheral units: tapes, printers, and readers. Their function is obvious to those familiar with data processing. The only feature that is worth noting here is the fact that each peripheral unit includes a serial I/O channel adapter. These adapters are nearly identical to one another and they establish the unit address.

Each peripheral unit operating on the serial I/O channel operates in the slave mode, and the intelligent terminal operates in the master mode. A typical exchange on this channel is as follows:

a. The terminal clears the line, sends a synchronization pattern and an address to all units, and then stops transmitting.

b. All units synchronize, but only one recognizes its address.

c. The selected unit sends a synchronization pattern, its own address, and its status to all units on the line.

d. All units synchronize, but only the master, the intelligent terminal, accepts the information.

e. After examining the address and status returned to it by the selected peripheral unit, the terminal determines what action is to be taken.

f. Assuming that the terminal is to print information, it sends the synchronization pattern and the command to print. It is not necessary to send the address because only the unit previously selected is active.

g. After receiving the print command, the printer responds by transmitting the synchronization pattern and its status back to the terminal.

h. Recognizing the status as being "ready to print", the terminal sends the sync pattern and one character to be printed.

i. After receiving the character to be printed, the printer again responds with the sync pattern and status.

j. The exchange outlined in steps h and i continues until the terminal is finished printing and whatever commands are required are sent to the printer.

k. If the terminal is to select another peripheral unit, it must repeat step a; only one unit can be active at any one time.

Communications over longer distances than the serial I/O channel can cover are handled by telephone, teletype, or special data transmission lines. A communications adapter can be plugged into the 1501 and connected by a cable to a MODEM. The MODEM then operates the phone lines.

Four arrangements are possible:

a. A communications adapter board is plugged into the 1501 and cabled to a MODEM. Then either a direct connection to a leased line is made or a telephone is used to dial the line only when data is to be transferred.

b. An asynchronous communications adapter is combined with a serial I/O channel adapter to become the Model 1534-1, so that the 1501 can operate the communications line over its serial I/O channel. This allows the 1501 to operate several communications lines, since many adapters can operate on the serial I/O channel. Again, the adapter is cabled to a MODEM, and either a direct or dialed connection is made.

c. Each of the preceding possibilities has the disadvantage of requiring operator intervention to dial a number or the disadvantage of the high cost of a dedicated phone line. The Model 1538 Automatic Dialing Unit is available to overcome this problem. This unit operates over the serial I/O channel and allows the 1501 to dial a telephone number. The data is then transferred from the 1501 through its internally-installed communications adapter and through a MODEM to the called number.

d. An asynchronous communications adapter is combined with the serial I/O channel adapter just as in the 1534-1, but added are the circuits required to drive a teletype line instead of a MODEM. This adapter is the 1534-2, and it allows the 1501 to communicate in a Telex network.

Electronic cash registers are now widely used by retail stores, and the 1501 has three ways in which it can handle data from these units. Assuming that the registers and the 1501 are close enough together that the serial I/O channel can be used, a Model 1590 MDTS Poller can gather data from the cash registers and forward the data to the 1501 over the serial I/O channel.

Not shown in the figure are the Series 800 store and forward units. These units, which are located near the Series 900 cash registers, record the transactions on tape cartridges and forward the data to a 1501 over a communications line. Of course, the tape cartridges could also be mailed to the 1501 site for processing.

The primary advantage of substituting one of the larger terminals for a 1501 is the availability of additional memory, disc units, and a large screen display. When a disc-equipped terminal is used, it greatly increases the storage capacity of the system. Programs that take advantage of the disc storage and access time thus become available to the user.

A disc-equipped terminal can serve as the core of a clustered system of terminals, with the smaller terminals acting as satellites. A description of a clustered system is given in the following paragraphs.

A small clustered system is shown in figure 1-19. Here, a Model 1503 Intelligent Terminal is serving as the host to four remote terminals - two 1501's and two 1502's. Operating in the 1503 is a program that manages the disc and handles requests for disc access from the remote terminals. At present, 16 remote terminals can be operated with each host.

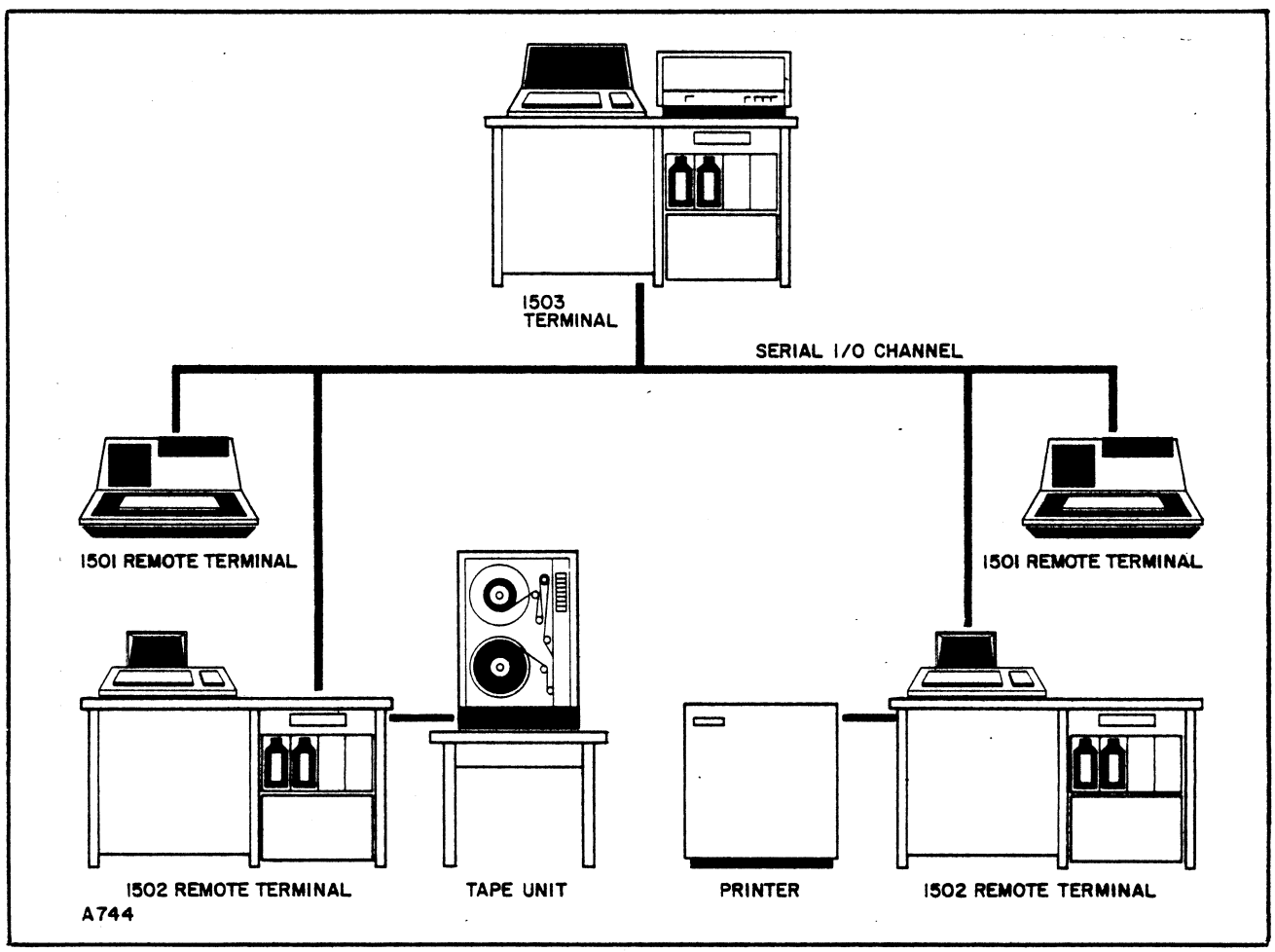


Figure 1-19. A Clustered System

The remote terminals are each identified by an address plug, and each operates as a slave unit on the serial I/O channel. The host terminal contacts each remote terminal in turn and processes its request for disc access. In general, the sequence described earlier for the exchange between a master and a slave on the I/O channel is used for the exchange between the host and remote terminals.

Note that two of the remote terminals shown are also operating peripheral equipment. This is one application of the dual serial I/O option. The remote terminals operate their peripheral equipment over their primary I/O channel and switch to the secondary channel when it is necessary to communicate with the host terminal.

Not mentioned up to this point in the description of an integrated system were the Model 1504 and Model 1580 Intelligent Terminals and the Model 1536 Intelligent Communications Adapter. These three units were originally designed to operate together in a system such as the one shown in figure 1-20.

The Model 1504 is intended to act as the host to a group of Model 1580's. However, the 1504 also has the capability to operate all of the peripheral and communications equipment shown in figure 1-18 over its serial I/O channel and communications lines and could thus substitute for either the 1501, 1501-40/43, or 1502/1503 terminals.

The major difference between the 1504 and the other terminals is that the 1504 can operate up to 64 asynchronous communications channels. It is necessary to refer to the description of the 1504 to determine the possible allocation of these channels but, for the purposes of this discussion, they can be categorized as local or long-distance channels. The long-distance channels use telephone lines and thus require a MODEM.

A number of 1580's act as satellites of the 1504, gaining access by interrupting the host terminal. The 1580 acquires its program from the 1504 and has no other source. Note that the 1580 is interconnected with the 1504 over either a local or long-distance communications channel.

It was mentioned above that the 1504 could operate communications channels just as the other terminals can. If, however, the anticipated communications load is expected to exceed the rate at which the 1504 can process the data, or if external processing is desired for other reasons, the Model 1536 is available to buffer and process data.

In addition to communications adapters, the 1536 is equipped with a processor and memory. (They are identical to those used in the 1580.) This allows the 1536 to exchange data with a remote system and perform such functions as format conversion, code conversion, and buffering. The 1536 exchanges data with the 1504 over either a local or long-distance communications line.

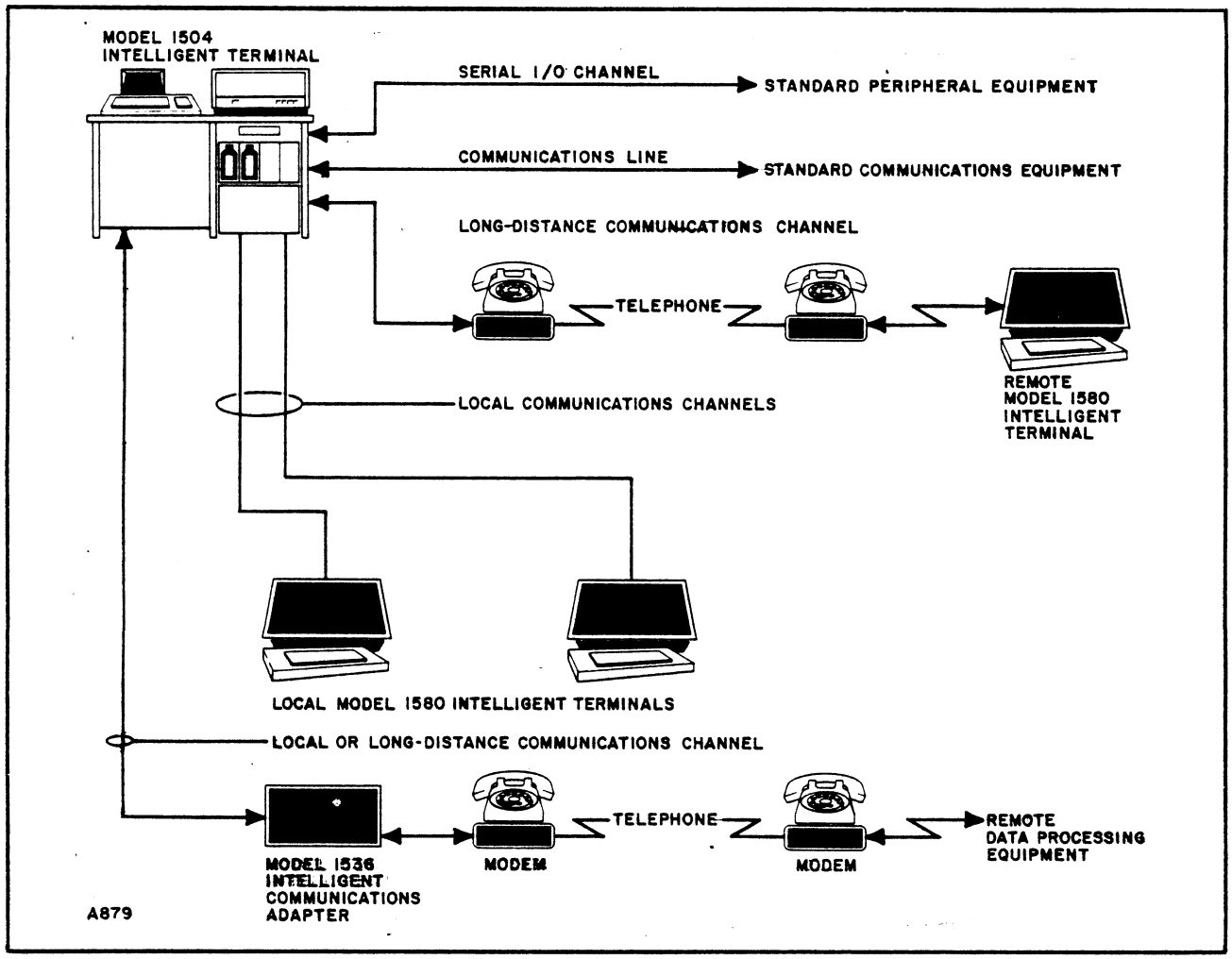


Figure 1-20. Integrated System Showing the Use of the 1504, 1536, and 1580

2-1. GENERAL.

The Series 1500 Intelligent Terminals make up a family of mini-computers and, with some exceptions, all execute the same basic set of instructions. In addition, the standard peripheral units and I/O instructions can be used with most terminals. A block diagram showing the basic components of a typical terminal appears in figure 2-1.

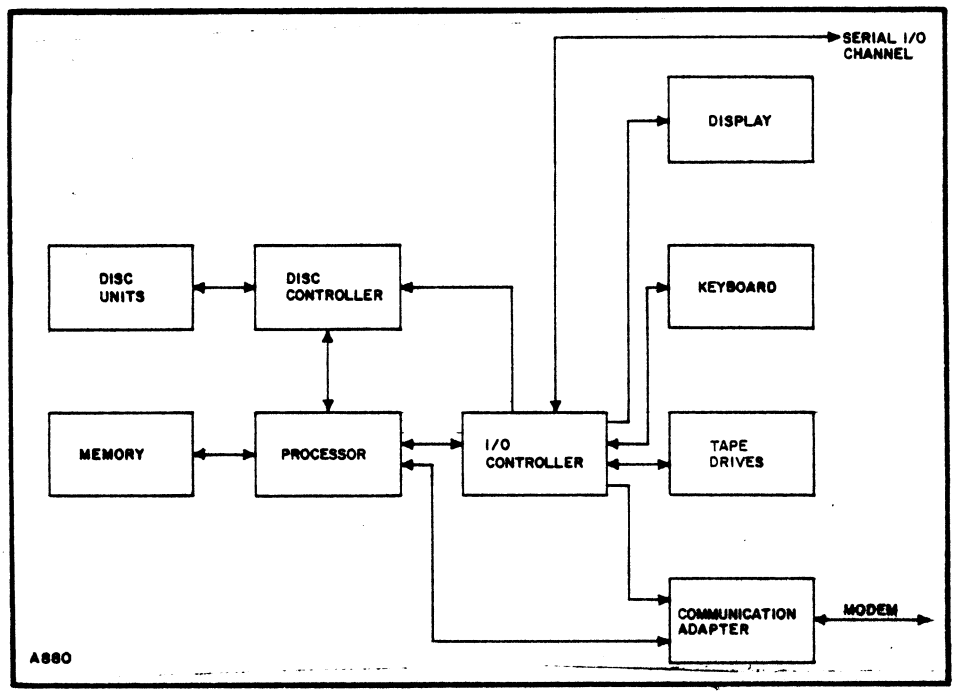


Figure 2-1. Block Diagram of Typical Intelligent Terminal

Although there are six intelligent terminals (1501, 1501-40/43, 1502, 1503, 1504, and 1580) and one intelligent communications adapter (the 1536) in the Series 1500 line, there are only three basic processors used, as follows:

a. The C4 processor is the original unit designed for the Series 1500 terminals. In the latest version, it has a repertoire of 41 instructions plus the I/O instructions executed by the I/O controller, disc controller, or peripheral units.

   The C4 processor is used in four terminals: the 1501, 1501-40/43, 1502, and 1503. It consists of one large printed circuit board and is capable of operating with a variety of memory sizes.

b. The C4VP is the second processor available, and it appears only in the Model 1504 Intelligent Terminal. ("VP" means virtual processor.) This unit has a repertoire of about 50 instructions, the basic C4 set plus special purpose instructions necessary to carry out the virtual processor concept.

   In the 1504 application, the C4VP is assigned to operate with specific sections of memory to perform specific tasks, and an interrupt network causes the processor to switch memory sections. This allows the one physical processor to appear as if it were several different processors; hence, the term "virtual processor" is given to this application.

   Although it can operate the basic C4 instruction set, the architecture of the C4VP is significantly different from that of the C4. Made up of two large printed circuit boards, the C4VP requires at least 32K of memory and can utilize as much as 65K.

c. The C5 is the newest processor in the line, and it appears in only two products: (1) the Model 1580 Intelligent Terminal, and (2) the Model 1536 Intelligent Communications Adapter. Although the architecture of the C5 processor is different from both the C4 and C4VP, the instruction set for the C5 is nearly identical to the C4 set. The minor differences are explained later in this section.

   The C5 processor occupies one large printed circuit board. In its present applications (1536 and 1580), it uses either an 8K or 16K memory; however, its addressing capability can accommodate up to 65K.

As is the case with nearly all computers, the Series 1500 terminals have assembler and compiler programs available to convert programmer's statements into machine language. The source statements are given in what is called "assembler language", and are converted to machine language by an Assembler System made up of a family of six programs.

Also available as a programming tool are two high-level languages, called Complex Data Entry (CDE) and Business Transaction Language (BTL). The instruction sets for both languages are very similar to one another, with the BTL being the larger set.

Both the CDE and BTL systems provide instruction interpreters for each high-level instruction. The interpreter converts the high-level instruction, such as DWT-Disc Write, into the machine language instructions necessary to carry out that operation.

Use of the Assembler System is described in publication number 91021, and the CDE and BTL systems are covered in publications 91002 and 91001, respectively. This section of the Programmer's Reference Manual describes each terminal's instruction repertoire in machine language and relates it to the Assembler System.

2-2. PROCESSOR ARCHITECTURE.

Organization and capability of the processor is of prime importance to a programmer; thus, the details of the processor and related units are covered in this manual. The purpose of the following paragraphs is to describe the overall organization and use of the processor.

Within each of the Series 1500 Intelligent Terminals is housed a general-purpose mini-computer. Regardless of whether it uses the C4, C4VP, or C5 processor, it is made up of three basic components: (1) memory, (2) an input/output controller, and (3) the processor. In the disc-equipped terminals, a disc controller is added to handle the disc data transfer.

The basic processor is a small, stored program unit with a repertoire of about 40 instructions. This number depends upon whether the C4, C4VP, or C5 processor is being considered, but it does not include the input/output instructions, which are executed by the I/O controller, the disc controller, or the peripheral units.

A block diagram of the basic C4 processor is shown in figure 2-2. Those familiar with the organization of computers will recognize most of the elements, which are:

a. An operation register and decoder. These circuits hold and decode each instruction read from memory.

b. A memory data register, which holds data read from memory.

c. A B-register, which holds one of the operands involved in many of the operations.

d. An adder, which performs the actual arithmetic or logical operation.

e. A memory address register, which holds the address of a memory location to be used.

f. An accumulator, which normally holds one of the operands involved in an arithmetic operation and also holds the result when the operation is completed. The accumulator also serves as a primary input/output register for the processor and, for that reason, is physically located in the I/O controller.

Other elements normally used in general-purpose computers are:

a. Index registers, which hold a value used to modify a memory address. This processor has index registers, but they are memory locations rather than hardware registers. Seven memory locations in each memory section are reserved to be used as index registers.

b. An instruction address register, which holds the address of the memory location in which the next instruction to be executed can be found. Again, memory locations substitute for hardware registers. Thirty-two memory locations are reserved to hold instruction addresses. A complex scheme in which a counter, called the stack pointer, designates the memory location to be read to acquire the address of the next instruction to be executed is used in this processor.

Word Formats.

The processor is organized to handle both instructions and data in 8-bit bytes. The bit positions in each byte are labeled 0 through 7, in order by the power of two that the position represents.

A typical instruction format is shown in figure 2-3. Note that each instruction is made up of two parts: (1) the IWR, instruction word-right, and (2) the IWL, instruction word-left. Memory addresses are also made up of two bytes, left and right, while data is handled as a single byte.

Memory Addresses.

Because the operation of the processor is very closely related to the memory, a brief description of the memory addressing scheme is presented here before operation of the processor is discussed. A typical memory in the Series 1500 Intelligent Terminals is solid-state with a capacity of 16,384 bytes, and this memory is organized into groups of 256 bytes each; each group is called a "page". Shown in figure 2-4 are the 77 (octal) pages that make up a 16,384-byte memory.

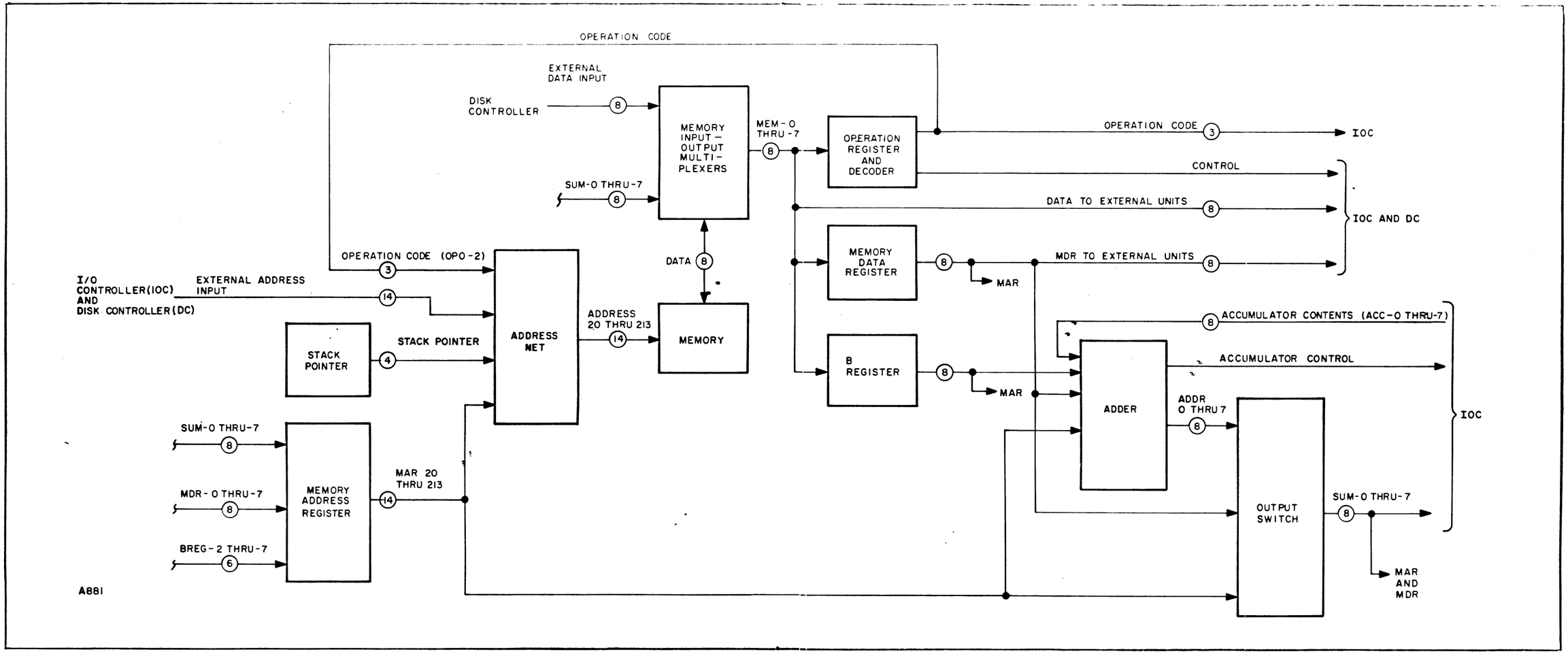


Figure 2-2. Basic C4 Processor Block Diagram

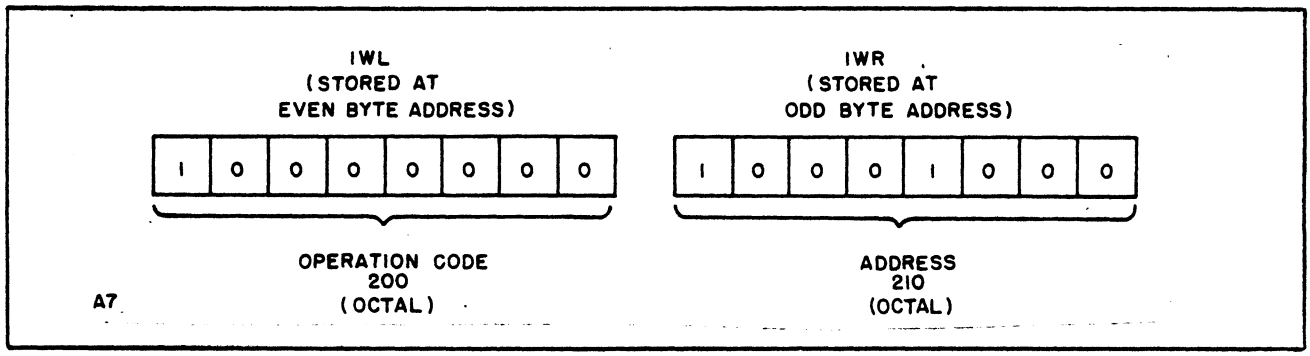


Figure 2-3. Typical Instruction Format

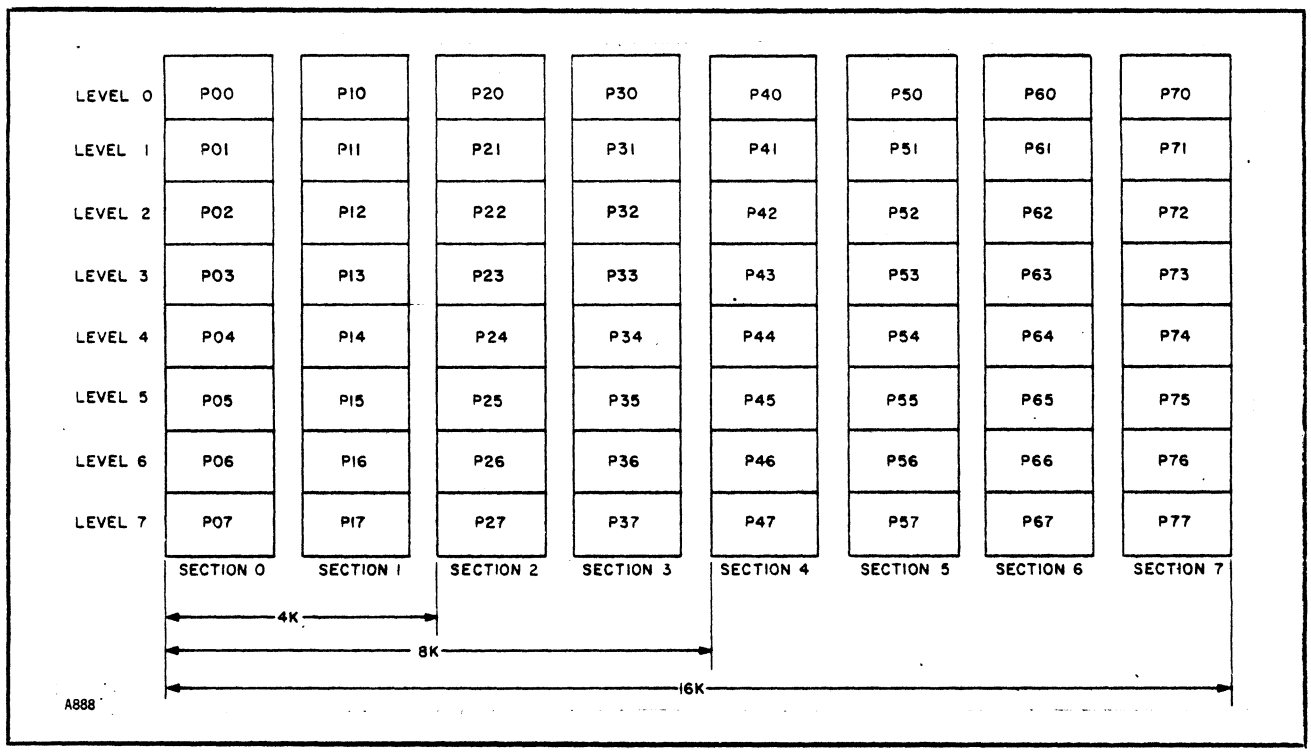


Figure 2-4. Organization of Memory

An address of 14 bits is required to address a memory of 16,384 locations. Since the processor is organized to handle 8-bit bytes, this requires two bytes, as shown in figure 2-5. Note that the section bits and level bits together are called the page number; they choose a 256-byte page. The eight low-order bits are called "byte" number; they select one of the 256 bytes in the page. For example, the address format to select byte 320 (octal) of page 16 (called P16-320) would appear as shown in figure 2-6. Note that both the page number and byte number are expressed in octal in this case for ease of reading the sample address.

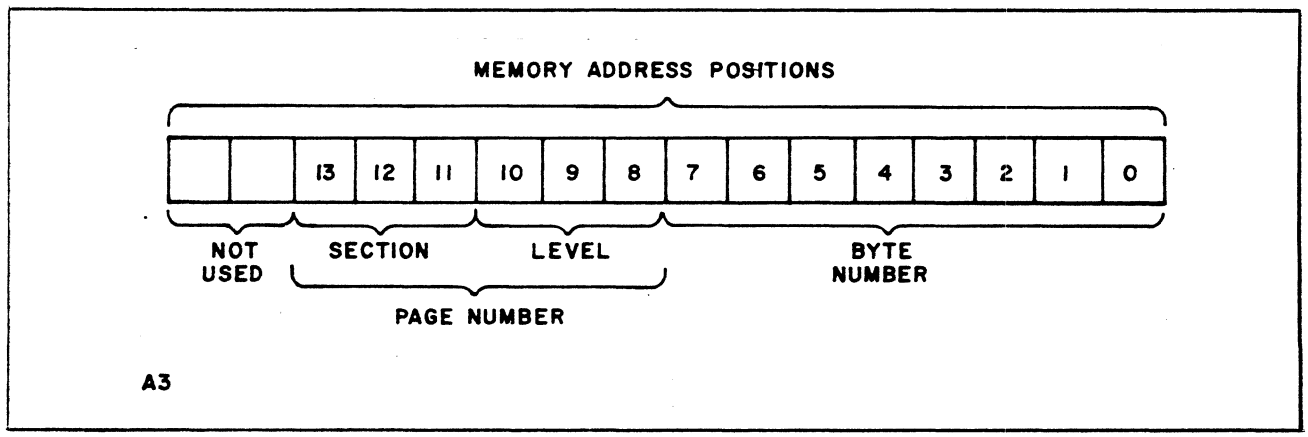


Figure 2-5. Memory Address Format

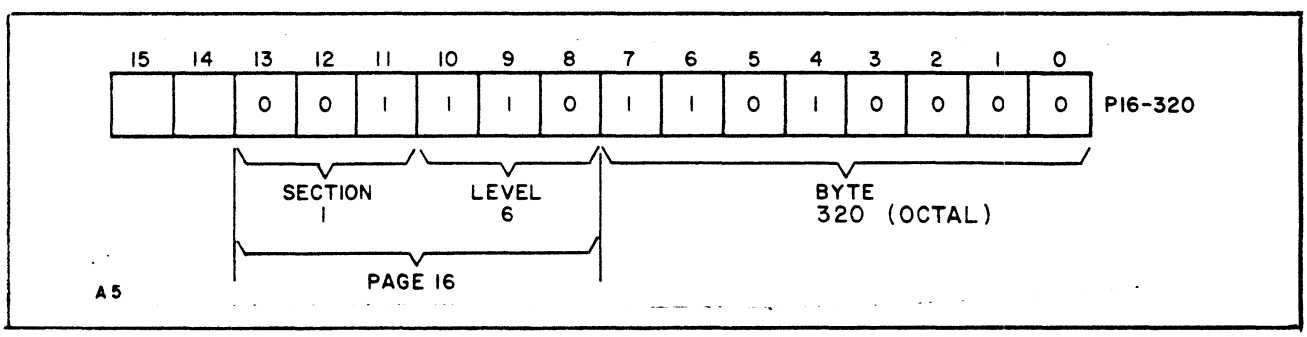


Figure 2-6. Sample Address

Instruction Addressing.

Operation of the processor is controlled by a program made up of many instructions. This program is stored in memory, from which it is read and executed one instruction at a time. The address of the instruction currently being executed is held in two memory locations, called stack pointer addresses. There are 32 of these locations, which means that the addresses of 16 different instructions can be held in the "stack".

The contents of two stack pointer addresses are shown in figure 2-7. Note that the first location holds the address of a byte within a page and that the second location holds the page number in which an instruction is stored. In order to execute that instruction, the processor must read the contents of a pair of stack pointer addresses. (The setting of the stack pointer determines which pair of addresses is used.) Once the processor has read the instruction address from the memory location specified by the stack pointer, it then uses this address to read the instruction itself from memory.

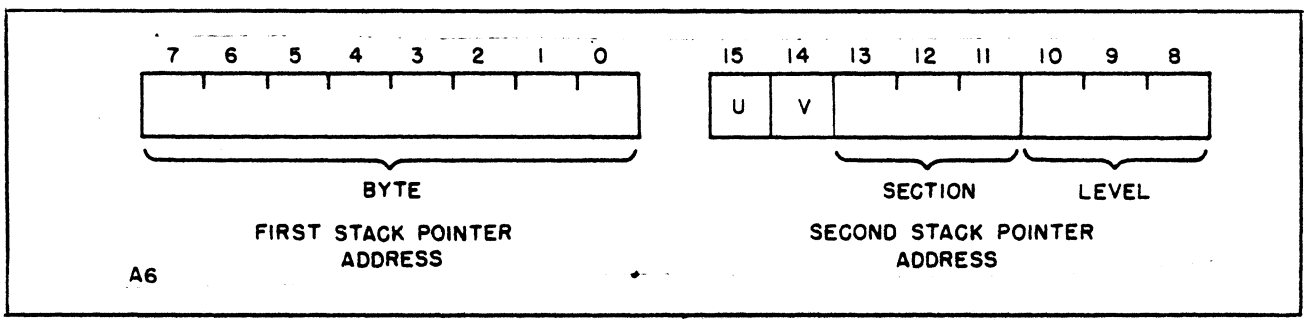


Figure 2-7. Contents of Stack Pointer Addresses

Instructions in machine language and their corresponding addresses are written in the following format:

Address of Instruction	Instruction
P03-100	.200-377
P03-102	.260-001
P03-104	.230-007

The first instruction above (200-377) is located at octal locations 100 and 101 of page 3. P03-100 is the address of the IWL, the 200 byte. The IWR (377) is located at the address of P03-101. These instructions would be stored in page 3 of memory as shown in figure 2-8. Notice that each instruction must begin at an even address in order to be read and executed properly.

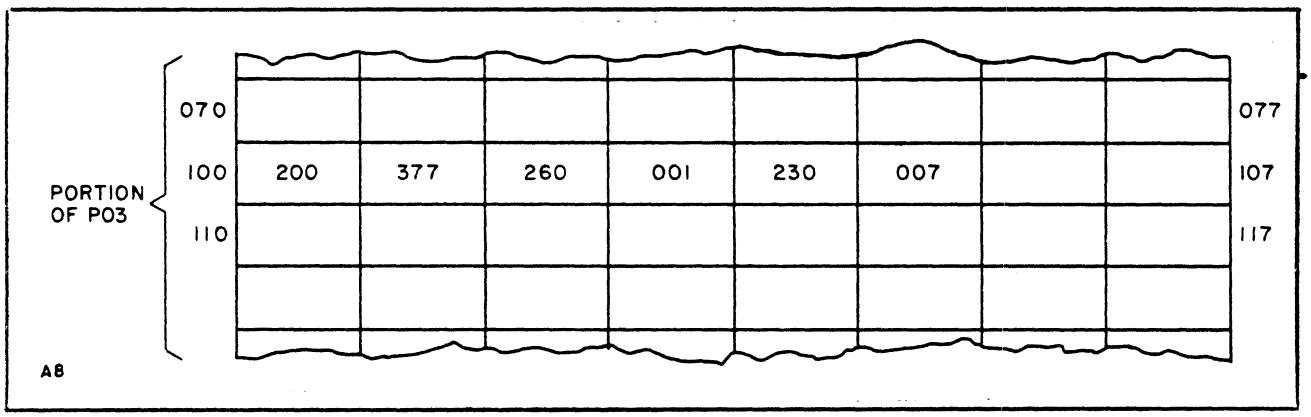


Figure 2-8. Typical Location of Instructions in Memory

Use of the Stack Pointer.

As shown in the preceding discussion, the address of the current instruction consists of two bytes. It is called the instruction address word left and right (IAWL and IAWR). The IAW in an active program is held in two specific memory locations, initially P00-040 and P00-041. These two locations are the first two stack pointer addresses.

When instructions stored in sequential addresses are executed, the IAW is read from P00-040 and 041, incremented by two during execution of the instruction, and returned to P00-040 and 041. However, there are many cases in which an instruction transfers to a non-sequential memory location to acquire the next instruction. Non-sequential execution is encountered in programs that contain "Test" and "Branch" instructions. These instructions cause the program to jump to a non-sequential address if a certain condition is met.

Figure 2-9 illustrates the principles discussed above. Steps A and B illustrate the execution of instructions stored in sequential addresses. Step A is instruction 200-000 (Load Accumulator with 000), located at address P01-050. This instruction address is being held in the first stack pointer address P00-040 and P00-041. (The IAWR is in P00-040; the IAWL is in P00-041.)

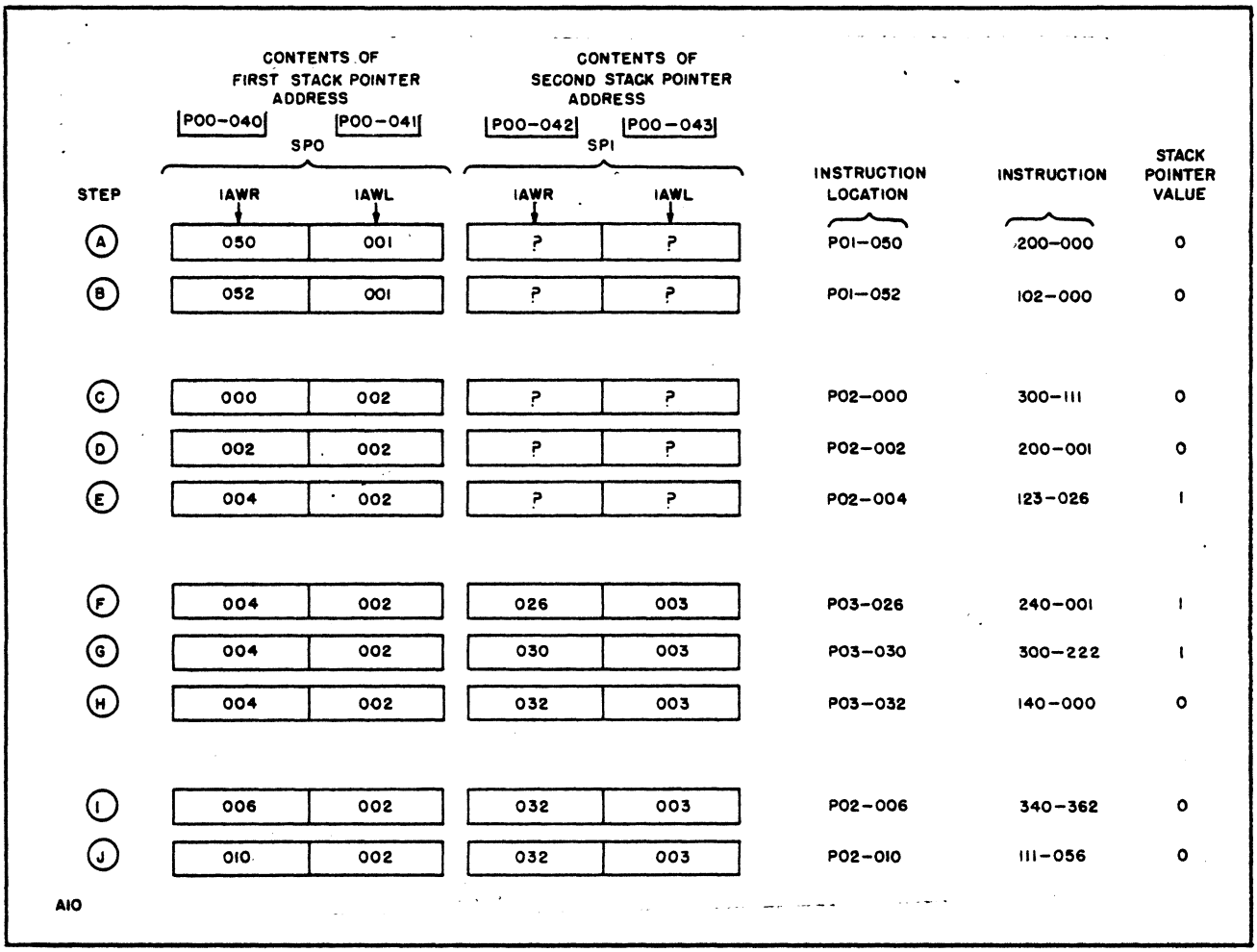


Figure 2-9. Use of the Stack Pointer

While the Load Accumulator instruction is being executed, the IAWR (050) is incremented by two and inserted back into P00-040. Now the address of the current instruction is P01-052. This is shown in Step B.

The instruction stored in address P01-052 is a branch instruction. When branch instructions are executed, they put the new branch address into the stack pointer address (P00-040 and P00-041), as shown in Step C. Note that branch instruction 102-000 produces address P02-000 in the stack pointer address locations. This means that the instruction held in P02-000 is executed next.

Steps D and E again illustrate sequential instruction addressing; however, step E is a stack and branch instruction. In addition to putting a new IAW into the stack pointer addresses, this instruction steps the stack pointer from SP0 to SP1 to cause the pointer to "point" to the second pair of addresses. This changes the location of the stack pointer addresses to P00-042 and 043. The old stack pointer addresses (SP0 in the figure) are not changed, and they are left containing the address of the stack and branch instruction executed in step E. The purpose of storing the address of the instruction at which the branch took place is to allow the program to return to this point later. The new, and now the current, stack pointer addresses (SP1 in the figure) contain the new "branch to" address.

Steps F, G, and H illustrate sequential execution using the new stack pointer addresses (SP1) to hold the instruction address words. Step H is an exit instruction. This instruction does not contain any "branch to" address. It simply steps the stack pointer down one, returning to stack pointer addresses P00-040 and 041 to get instruction address words. The IAW retrieved is 004-002. This is incremented by two to obtain instruction address P02-006, in step I, and thus the program has returned to the point at which the branch took place (the next instruction address after the instruction in step E). The stack pointer addresses continue to be P00-040 and 041 unless another stack and branch instruction is executed, and the contents of the SP1 locations do not change.

As previously mentioned, the purpose of the stack and branch, then exit, instructions is to preserve the "come from" address and allow the program to return to this address by executing the exit instruction. This allows the sharing of subroutines by different areas of the main program. A maximum of 15 consecutive stack and branch operations are possible before returning to the original starting point.

Data Addressing.

Data bytes can be addressed by an instruction in any one of three different addressing modes: immediate, direct, or indexed. In the immediate addressing mode, the IWR is used as the data byte, and no further memory access is required by the current instruction. Immediate addressing provides a means of entering data into the accumulator by specifying the data in the IWR of the instruction in which it is used. For example, instruction 200-311 causes an octal 311 to be loaded into the accumulator.

In the direct addressing mode, the IWR is used as the address of the data byte. Since only the eight bits of the IWR are available for an address, only 256 (377 octal) selections are possible. Essentially, a byte within a page can be selected in the direct addressing mode but the page itself cannot be chosen.

In this mode, the page number is established by two conditions. First, the section of memory (0 through 7, the most significant digit of the page number) can be selected by a Set Memory Section instruction previously executed. Second, the level of memory (0 through 7, the least significant digit of the page number) is not supplied. Therefore, it is effectively zero. Thus, instructions executed in the direct addressing mode can select addresses in pages 00, 10, 20, 30, 40, 50, 60, and 70, with the most significant digit of the page number having been previously established.

The indexed addressing mode provides a method of addressing data stored anywhere within memory. An indexed address is composed of a byte address contained in a specified index register in memory plus a page address contained in the IWR. The high-order six bits of the IWR (2-7) specify the page within memory and the index register specifies the location within that page. Bits 0-2 of the IWL select the index register to be used.

An example of an instruction executed in the indexed address mode is 213-050 (Load Accumulator), the format of which is shown in figure 2-10. This instruction causes the contents of the location in P12 specified by the contents of index register 3 to be loaded into the accumulator. If index register 3 contained 145 (octal), for example, the contents of P12-145 would be loaded into the accumulator. The two low-order bits of the IWR determine what changes the instruction makes in the contents of the selected index register, as follows:

00 - Leave unchanged

10 - Increment by 1

11 - Decrement by 1

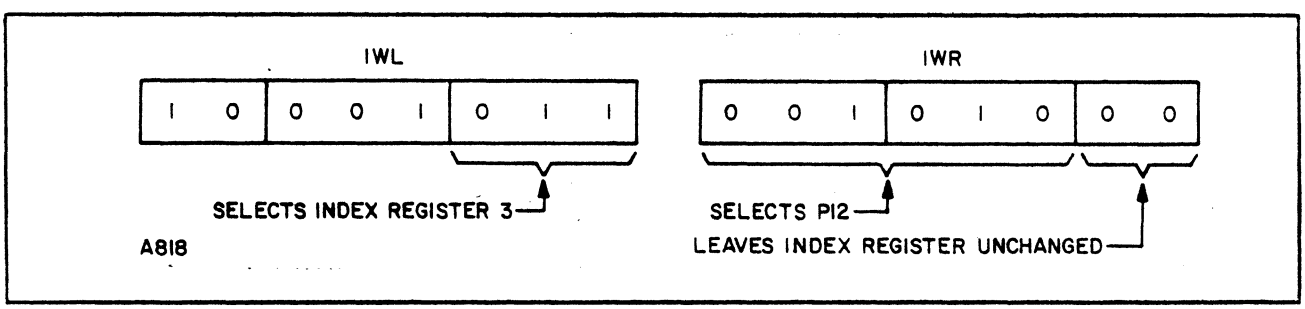


Figure 2-10. Format of an Instruction Using the Indexed Address Mode

Instruction Cycles.

The basic timing unit for the processor is the machine cycle, which is one microsecond long.

The next timing division in the processor is the instruction cycle, which is the time required to acquire an instruction and execute it. Depending upon the type of instruction, an instruction cycle is made up of three to six machine cycles. Thus, an instruction may require between three and six microseconds.

As stated above, a maximum of six machine cycles are required to perform an instruction. These cycles are designated I (instruction) or E (execution) as follows: I1, I2, I3, I4, E2, and E3. In general, they serve the following purposes: Cycles I1 and I2 are devoted to forming the address from which the next instruction will be read. Then, I3 and I4 are the intervals during which the instruction is read and E2 and E3 are the intervals during which it is executed. This simplified explanation is not true for all classes of instructions, however, and further explanation is provided in the following paragraphs.

Instruction Classes.

Instructions can be divided into four general classes:

Class 0, Jump Instructions

Class 1, Branch and I/O Instructions

Class 2, Transfer and Arithmetic Instructions

Class 3, Boolean and Compare Instructions

The class to which an instruction belongs is indicated by the two most significant bits of the left byte of the instruction word, which are 0, 1, 2, or 3, corresponding to the class of instructions. Instruction cycles I1 through I4 are common to all instruction classes, while execution cycles E2 and E3 operate only during Class 2 and Class 3 instructions. Table 2-1 shows the cycles required for each class of instruction.

Table 2-1. Cycles Required by Each Instruction

IWL	TYPE	REQUIRED CYCLES
0NN 10N 11N 12N 13N 14N 15N 16N 17N	Neither Direct, Indexed, Nor Immediate Address	I1, I2, I3, I4
20N 22N 24N 26N 30N 32N 34N 36N	Immediate Address	I1, I2, I3, I4, E3
21N 23N 25N 27N 31N 33N 35N 37N	Direct or Indexed Address	I1, I2, I3, I4, E2, E3

The number of machine cycles in an instruction depends upon the instruction and the conditions that exist when the instruction is executed. Since the IWL, which contains the operation code, is read during cycle I3, the following discussion defines the starting cycle as I3. Figure 2-11 shows the functions performed during each cycle.

At the beginning of cycle I3, the address of the instruction to be executed has been formed in the memory address register and is applied to the address net shown in the processor block diagram. The least significant position of the address net is controlled directly by the I3 and I4 signals. During I3, it is held at zero to read an even address and during I4 it is switched to a one to read the odd address.

I1	1ST HALF	1. Using the stack pointer as an address, read the IAWL. 2. Load the IAWL in the MDR (memory data register)
	2ND HALF	1. Add 2 to IAWL (if no branch) 2. Restore the IAWL 3. Load MAR (memory address register) with IAWL or branch address
I2	1ST HALF	1. Using the stack pointer as an address, read the IAWR. 2. Load the IAWR in the MDR
	2ND HALF	1. Add carry to IAWR (if carried from IAWL) 2. Restore the IAWR 3. Load MAR with IAWR or branch address
I3	1ST HALF	1. Using the MAR as an address, read the IWL 2. Load the IWL into the operation register, B register, and MDR.
	2ND HALF	1. Using the I/O controller address, read the CRT character code; or using the disk controller address, read or write. 2. Transfer character code to I/O controller or transfer data to disk controller.

A398 (Sheet 1)

Figure 2-11. Functions Performed During Each Instruction Cycle (Sheet 1 or 2)

I4	1ST HALF	1. Using the MAR as an address, read the IWR 2. Load the IWR in the MDR and/or B register
	2ND HALF	1. Using the I/O controller address, read the dot pattern for the CRT character position. 2. Transfer dot pattern to the I/O controller.
E2	1ST HALF	1. If the indexed address mode is selected, use the index register number and section number as an address, read the contents of an index register. Load the contents of the index register into the MDR. 2. If the direct address mode is selected, no memory access is required.
	2ND HALF	1. If the indexed mode is selected, modify the contents of the index register and restore the contents of the index register. Load MAR with composite address from the B register and the MDR. 2. If the direct address mode is selected, load the MAR with the IWR without another memory access cycle.
E3	1ST HALF	1. If necessary, read the contents of the location specified by MAR. 2. Load the contents into MDR.
	2ND HALF	1. Execute instruction 2. If necessary, store the results in the location specified by MAR.

A398 (Sheet 2)

Figure 2-11. Functions Performed During Each Instruction Cycle (Sheet 2 of 2)

During the first half of I3, the IWL is read. This byte is then loaded into the operation register, the B register, and the memory data register. At this point, the operation code is decoded and instruction execution can begin.

During the second half of I3, the I/O controller gains access to memory. The I/O controller reads either character codes or dot patterns for the CRT display. In terminals equipped with a disc, the disc controller can override the I/O controller and gain memory access during the second half of I3 to either read or write data.

At the beginning of cycle I4, the processor again uses the contents of the memory address register to select a memory location and read the contents. As previously mentioned, the I4 signal applied to the address net switches the low-order position to a one; thus, the IWR is read from an odd address. The IWR is loaded into the MDR, replacing the previously loaded IWL. Thus, at the end of cycle I4, the IWL is in the operation register, while the IWR is in the memory data register.

During the second half of cycle I4, the I/O controller again addresses memory. Again, it reads either the character code or the dot pattern required to produce the characters on the CRT display.

The next step taken depends upon the class of instruction. Class 0 and Class 1 instructions require only cycles I3, I4, I1, and I2 to perform their function, whereas Class 2 and Class 3 instructions also require cycles E2 and E3.

Assume that the current instruction is a Class 2 or 3 instruction. Once this instruction is decoded, the processor determines from its operation code whether it is to be executed in the immediate, direct, or indexed addressing mode. Operation in the indexed mode is discussed first.

When the indexed address mode is selected, the index register number specified in the IWL and the section number held in the section register are gated through the address net to select an index register and read its contents during cycle E2. The data read is loaded into the memory data register. During the second half of cycle E2, the contents of the MDR are modified as specified by the two index register control bits. The modified number is then written back into the same index register.

At the end of cycle E2, the contents of the MDR (contents of the index register) are transferred to the low-order positions of the MAR, and bits 2-7 of the B register (the page number portion of an address) are loaded in the high-order positions of the MAR. Together, these portions form a composite address of the memory location to be used by this instruction.

If the direct address mode is selected, the memory access cycle during the first half of cycle E2 is not required. The MDR still holds the IWR and, during the second half of E2, the IWR is transferred to the MAR.

For Class 2 or 3 instructions operating in the immediate address mode, cycle E2 is skipped. Cycle E3 begins immediately after cycle I4.

If a memory access cycle is required during cycle E3 of the Class 2 and 3 instructions, the MAR contents are used to address memory as required to execute the instruction. Note that some of the Class 2 and 3 instructions do not require further memory access for their execution.

After an instruction (regardless of class) is executed, cycle I1 must be performed to acquire the next instruction. However, the disc controller is again allowed to gain memory access to either read or write before cycle I1 begins. Since this is done before the cycle counter is moved to I1, the counter is at zero and the cycle devoted to disc access is designated I0.

During cycle I1, the contents of the stack pointer are gated through the address net. This reads the IAWL from memory. The action taken during the second half of I1 depends upon whether or not the instruction is to execute a branch. Assuming that no branch is to be taken, the IAWL is increased by two, to specify the address of the current instruction, and returned to the stack pointer address from which it was read. The modified IAWL is then loaded into the low-order positions of the MAR to become part of the address from which the next instruction will be read.

Still assuming that no branch is to be taken, the processor must determine if cycle I2 is necessary. There are four different conditions that cause cycle I2 to be performed. These conditions are:

a. A branch is to be taken. The results of the branch instructions executed during I3 and I4 are examined and determine whether or not a branch is to be taken.

b. The instruction is in the 150 through 157 group.

c. The current instruction used either the direct or indexed address mode. This means that the page number held in the high-order section of the MAR was changed and, therefore, it is necessary to read the IAWR to reacquire the page number.

d. A carry resulted when two was added to the IAWL in cycle I1. If a carry resulted, the IAWR has been stepped into the next page and, therefore, the IAWR must be read and modified.

If cycle I2 is not required, the processor moves to cycle I3 and reads the IWL from the location specified by the contents of the MAR. This begins a new instruction.

If, however, cycle I2 is performed, the following action takes place: The IAWR is read from the odd stack pointer address. If a carry must be added, this is done and the updated IAWR is restored to the location from which it was read. The IAWR is also loaded into the high-order positions of the MAR and used to read the next instruction.

The next case to be considered is what must be done if the current instruction produces a branch condition. In this case, both cycles I1 and I2 must be performed, but the action taken during these cycles depends upon which instruction is being executed. Basically, an instruction address is formed to substitute for the IAWL and IAWR. This new address is loaded into the MAR and the next instruction read from the new address during I3 and I4.

Initial Conditions and Program Loading.

When power is first applied, the output of the POWER ON switch clears many circuits and holds them in the reset condition. So, the processor is inactive even though power is applied.

The next step taken is the operation of the PROGRAM LOAD switch. This allows the processor to operate. The program load signal selects tape drive 2, starts the tape, and begins reading. It also clears the stack pointer, thus establishing the first location in the address stack as the source of the IAW.

During cycles I1 and I2, the stack pointer addresses locations 040 and 041, respectively, but the contents of these locations are not used. Instead, an IAWL of 001 is formed at the adder, transferred to the MAR, and also stored in location 040. The least significant bit being a one has no effect because it is not used at the MAR. When the contents of 041 are read, the initial cycle conditions force a 001 into the IAWR. This makes the composite instruction address P01-000, which addresses a ROM (read-only memory).

When the PROGRAM LOAD switch was depressed, the ROM load conditions were established. Now, the cycle counter moves the I3 and allows the ROM to be addressed and the first IWL read.

The cycle counter then moves to I4 in the normal manner and addresses and reads the IWR from the ROM. The first instruction is executed. Then the remaining seven instructions of the ROM program are read and executed normally. They read the first record from the tape in drive 2 and branch to the first instruction read in. This instruction steps the stack pointer, which clears the ROM loading conditions and ends the loading sequence.

The ROM holds an eight-instruction program. This program reads 128 bytes from tape drive 2, stops the tape, and then branches to the first instruction read from the tape. From this point, the program read from the tape controls the processor.

The program stored in the ROM is shown in table 2-2. The instruction addresses are held in the instruction address stack locations 40/41, and the processor runs through the normal instruction cycles for each instruction read from the ROM. Thus, the instruction addresses are stepped in the normal way.

Table 2-2. Tape Loader ROM Program

Instruction Location	Instruction	Description
P01-000	201-030	LDX, load index register 1 with 030
P01-002	170-007	Transfer byte, unconditional
P01-004	231-002	STA, Store byte in page 00-030, increment index register 1
P01-006	341-230	CPX, Compare index register 1 to literal 230
P01-010	111-003	BRL, Branch to P01-002 if record not read
P01-012	170-016	Read status to reset read
P01-014	170-005	Stop tape
P01-016	100-030	BRU, branch to page 00-030

The first instruction in the ROM program loads index register 1 with an octal 30. The second instruction is a Transfer Byte instruction that is sent to the I/O controller for execution. Since the right byte is 007 rather than 207, the processor stalls (remains at the current instruction) until the byte is transferred into the accumulator.

When the byte is available in the accumulator, the third instruction (231-002) stores it in location P00-030 and increments the index register by 1, bringing it to 31. The fourth instruction compares the contents of index register 1 to the literal 230. If the index register has not reached 230 yet, the next instruction branches back to continue reading.

When 128 bytes have been read, the index register equals 230, and the branch is no longer performed. The reset read and stop tape instructions then terminate the reading operation, and the last instruction branches to P00-030 to acquire the next instruction; this being the first instruction loaded from the tape.

Interrupt Handling.

The processor is provided with two methods of interrupting normal instruction execution: one is the program interrupt position of the PROGRAM LOAD switch; the other is an external interrupt signal input for use by external devices. At present, only the disc controller and the Model 1539 Communications Adapter interrupt the processor.

When either the switch is pushed or an external interrupt signal occurs, the processor records the interrupt and synchronizes the interrupt signal with the processor timing. If a second interrupt signal occurs before the first interrupt is synchronized, the interrupt overflow indicator is set. This becomes bit 6 in the processor status word.

If no inhibiting conditions exist, the interrupt causes the automatic stack and branch operation. This transfers program control to location 000 of page P03, which contains the first instruction of the interrupt service routine. Four conditions can inhibit the stack and branch operation. They are:

a. The tape is moving.

b. A keyboard character is available to be transferred to the accumulator.

c. A Disable Interrupt instruction has been executed.

d. A different memory section is being used. (A Set Memory Section instruction was executed.) This prevents an interrupt from taking place until the IAWR has been restored to the stack pointer address and thus prevents the page portion from being lost.

To summarize the automatic stack and branch (ASB) operation:

a. The stack pointer is stepped up by one.

b. During I1 and I2, the IAW of P03-000 is forced through the adder, stored in the new stack pointer addresses, and loaded into the MAR.

c. During I3 and I4, the next instruction is read from P03-000. This is the first instruction of the interrupt servicing routine.

Stall and Skip Conditions.

In order to exchange data with the peripheral equipment operated by the I/O controller, the processor sends a Transfer Byte instruction to the controller. This instruction can be executed in either of two modes: stall or skip. If the IWR is 007, the instruction is executed in the stall mode; 207 in the IWR chooses the skip mode.

The Transfer Byte instruction is decoded and executed in the I/O controller. If the peripheral unit is not ready, the controller returns the stall signal to the processor. The action taken by the processor then depends upon the instruction mode selected, as follows:

a. If the stall mode is selected, the processor remains at the current instruction until the peripheral unit is ready.

b. If the skip mode is selected, the processor skips the next instruction and continues with the program.

The stall signal from the I/O controller is applied to the adder. There, it prevents the adder from adding two to the IAWL during cycle I1. This means that the same address is returned to the current stack pointer location and the processor executes the Transfer Byte instruction over and over until the I/O controller removes the stall signal.

If the IWR is 207-Skip on Busy, the skip signal is sent to the adder. During I1, the cycle in which the IAWL is being modified, the skip signal applied to the adder causes the IAWL to be increased by four instead of the usual two. This causes the processor to skip the next instruction and proceed.

When the terminal is equipped with a large screen display, there are two other conditions that can cause the processor to stall. They are the Clear Screen and Exit C4 Mode instructions used to control the large display. The use and effect of these two instructions are discussed later in this section.

Use of the U and V Bits.

Two control bits, "U" and "V", determine which memory page and section that the direct address, the indexed address, or the branch address affects. These two control bits are established by the SMC and SSC instructions. When either of these instructions is executed, bits 7 and 6 of the IWR become the U and V bits, respectively.

Each time that the IAWR is restored to a stack pointer address during cycle I2, the U and V bits occupy positions 7 and 6, respectively, of the IAWR. When the IAWR is read during the next I2 cycle, the U and V bits are used to control the next instruction.

One function of the U bit is to determine which section of memory is to be used in indexed address instructions. As mentioned in the overall processor description, addresses 1 through 7 in each memory section are reserved for use as index registers. Whenever the U bit is a zero, the index registers in section 0 are to be used. Whenever the U bit is set to a one, the index registers to be used are those in the same section as the current instruction.

The second function of the U bit is to determine which level and section the direct address affects. When the U bit is a zero, the IWR (address) of all instructions executed in the direct address mode applies to level 0 of section 0. Conversely, when the U bit is a one, the direct address applies to level 0 of the same section as the current instruction.

To summarize the function of the U bit:

a. When it is a zero, the section 0 index registers are to be used and direct addresses apply to level 0 of section 0 (page 00).

b. When it is a one, the index registers in the same section as the instruction are to be used, and the direct addresses apply to level 0 of the same section as the instruction (page 00, 10, 20, etc.).

The V bit performs a similar function for branching operations. If the V bit is a one, any branch, stack and branch, or exit and branch instruction with page 0 specified in the branch addresses causes the branch to occur within the current page. If any page other than 0 is specified in the branch address, the V bit control is inactive, and a normal branch occurs. For example, assume that the V bit is a one and a branch instruction located in page 5 specified a branch to page 0, location AAA. The resulting branch will be, to page 5, location AAA. If the branch address specified was page 6, location BBB, then the resulting branch will be to page 6, location BBB. If the branch instruction is located in any page other than page 0 and a branch to page 0 is desired, the V bit must be left at zero.

To summarize the use of the V bit:

a. If it is a zero, a branch instruction located in any page other than page 0 can branch to page 0.

b. If it is a one, a branch instruction with page 0 specified in the branch address will branch within the same page as the instruction.

Processor Status Word.

Available for examination at any time is a processor status word, which is loaded into the accumulator when an LPS-Load Processor Status instruction is executed. This 8-bit word is made up of the following information:

a. Bits 0 through 3 hold the current setting of the stack pointer.

b. Bits 4 and 5 hold the status of the condition register. Bit 4 indicates that the accumulator contents are greater than the quantity against which they are being compared. This is the "high" condition. Bit 5 indicates the "equal" condition. If both bits are zero, it indicates the "low" condition, which means that the accumulator contents are less than the quantity against which they are being compared.

c. Bit 6 indicates an interrupt overflow, which was discussed in the description of interrupt handling.

d. Bit 7 indicates that the program interrupt switch was activated to cause the interrupt.

Stack Pointer.

The stack pointer is a 4-bit counter that can be incremented or decremented by certain instructions. It produces a 4-bit address, SP 1-4, that is applied to address net positions 1-4, respectively. During machine cycles I1 and I2, the stack pointer output addresses memory to acquire the next instruction address.

Address net bit position 0 is a zero during cycle I1, a one during cycle I2; position 5 of the net is a one during both of these cycles. This causes the stack pointer to address even locations 40 through 76 (octal) during I1 and odd locations 41 through 77 during I2. These 32 locations, 40 through 77 (octal), constitute the instruction address stack.

The purpose of the instruction address stack is to provide a means of returning to the point at which the program branched to a subroutine. When the processor is started, the stack pointer begins at zero. This addresses locations 40 and 41 to acquire the IAWL and IAWR, respectively. Each time that an instruction is executed, the IAW read from these locations is stepped by two and returned to memory. Thus, when the contents of locations 40 and 41 are read to acquire the IAWL and IAWR for the next instruction, the IAWL and IAWR address the next instruction in sequential order. If a branch instruction is executed, the address normally returned to the stack pointer locations is replaced by the "branch to" address, and instructions are again executed in sequence, starting at the "branch to" address.

There are, however, eight instructions that change the stack pointer addresses to new locations. They are:

a. TLX - Test Literal and Exit.

b. TMX - Test Mask and Exit.

c. SBU - Stack and Branch Unconditional.

d. SBE - Stack and Branch on Equal

e. SBH - Stack and Branch on High.

f. SBL - Stack and Branch on Low.

g. EXB - Exit and Branch.

h. EXU - Exit Unconditional

These instructions can increase or decrease the stack pointer so as to cause the stack pointer to "point" to different locations from which the IAWL and IAWR are to be read. The stack instructions step the counter up, and the exit instructions step the counter down.

The conditions required for a step up are:

a. One of the four stack instructions or an automatic stack and branch operation is being executed, and

b. If a conditional stack instruction (SBE, SBL, SBH) is being executed, the condition being tested is met.

The conditions required for a step down condition are:

a. One of the exit instructions is being executed, and

b. If a conditional exit instruction (TLX and TMX) is being executed, the condition being tested is met.

Terminal Identification Plug.

The purpose of the terminal identification plug is to allow a program to distinguish one processor from all others. For example, a program may allow only specific processors to have access to payroll records. In this case, the program would read the program interlock code to determine if this processor was to be granted access to those records.

Identification is provided by an 8-bit plug that occupies position K1 on the processor board. The purpose of this plug is to assign a unique number to the processor in which it is installed.

From the plug, the 8-bit code is available at all times except when instruction 155, Load Processor Status, gates the status word and blocks the identification code. Instruction 157, Load Terminal Identification, places the code in the accumulator.

B Register.

The B register is an 8-bit register. It is loaded at the end of the first half of I3 and again at the end of the first half of I4.

During the first half of cycle I3, the processor reads the IWL from memory. This is loaded into the B register. Thus, the B register holds the IWL at the end of cycle I3.

During the first half of cycle I4, the processor reads the IWR from memory. If the current instruction is a Class 2 or 3 instruction, the IWR is loaded into the B register. Note that in Class 2 and 3 instructions the IWR is always a literal or an address.

Memory Data Register.

The memory data register is an 8-bit register. Its purpose is to hold information read from memory until the processor can accept and act on that information.

During the first half of each machine cycle, information is read from memory and appears at the input of the MDR. This information is loaded into the MDR, as follows:

a. During cycle I1, the IAWL is loaded in the MDR if a branch is not to be taken.

b. During cycle I2, the IAWR is loaded in the MDR.

c. During cycles I3 and I4, the IWL and IWR, respectively, are loaded in the MDR.

d. During cycle E2, the contents of an index register are loaded in the MDR provided that an index register is specified. If no index register is specified, the MDR is left unchanged during cycle E2.

e. During cycle E3, the contents of a memory location are loaded into the MDR for any instruction that requires that data be read from memory for its execution.

Memory Address Register.

The memory address register (MAR) is a 16-bit register. Its purpose is to hold a memory address that the processor applies to memory. The MAR is divided into two 8-bit sections, a low-order section and a high-order section. Since only 14 bits are required for the memory address, the two high-order bit positions of the 16-bit register are not used.

The address loaded into the MAR originates in the memory data register (positions 0-7), the B register (positions 2-7), or the adder output switch (positions 0-7). Applied to the low-order section are bits 0-7 from the MDR and positions 0-7 from the adder. Only during instruction cycle E2, when the MDR holds the contents of an index register or the direct address, is the MDR 0-7 input allowed to reach the MAR. During all other instruction cycles, the 0-7 lines from the adder reach the low-order section of the MAR.

The inputs to the high-order section of the MAR are B register positions 2-7, and the 0-5 lines from the adder. During instruction cycle I2, the 0-5 positions are gated to the high-order section of the MAR. This brings the modified page portion of an instruction address into the MAR.

During instruction cycle E2, if an index register has been selected by the current instruction, bits 2-7 from the B register are gated to the high-order section of the MAR. This is the page part of a composite address. The other part (the byte number) is read from an index register.

To summarize the inputs to the MAR:

a. The adder output positions 0-7 are gated to the low-order portion in all cases except cycle E2. During E2, the output of the MDR, which holds the direct address or the contents of an index register, is gated to positions 0-7 of the MAR.

b. The adder output positions 0-5 are gated to the high-order portion during cycle I2. In this case, the data on these lines is a modified page portion of an instruction address. During cycle E2 of instructions using the indexed address mode, the page portion of a composite address reaches the MAR from the B register. These are bits 2-7 of the IWR.

The MAR outputs are labeled according to their weight, 20 for 2^0, the low-order bit, through 213 for 2^{13}, the high-order bit. At the address net, the MAR output is gated through bit positions 1-13 during the first half of I3 and I4. Bit 0 is a zero during I3 and one during I4, thereby addressing the two successive bytes of the instruction word. If direct or indexed addressing is used, the MAR is also gated through bit positions 0-13 during E3.

Adder.

The adder can perform either logical or arithmetic functions, depending upon the mode selected and the function chosen within that mode. Adder controls select both the mode and function, based upon the instruction being executed.

Two eight-bit quantities can be applied to the adder to be acted upon. The selection and whether or not a carry-in signal is to be applied to the adder is based upon the instruction being executed.

The adder forms an 8-bit quantity at its output. Also produced are the carry-out and the A-EQ-B (A equals B) signal.

The adder mode is selected by the adder controls. When the arithmetic mode is selected, carries between adder stages are enabled. Conversely, when the logic mode is selected, carries are disabled.

Memory Input and Output Multiplexers.

The purpose of the input multiplexer shown in the processor block diagram is to choose either the information on the SUM-0 through SUM-7 lines from the output switch or the information on a second set of input lines to be stored in memory. The second group of lines is an external input from the disc controller.

Enabling or disabling the input multiplexers is a write signal. This signal is active whenever information is to be stored. This occurs when the IAW is restored, when the index register contents are to be stored, when an external input is to be stored, and when any instruction requiring data to be stored in memory is executed.

The selection line to the input multiplexers is the write signal from the disc controller. It chooses the disc data to reach memory and blocks the adder input.

Two sets of eight lines are routed to the memory output multiplexer, although only one set is shown in the processor block diagram. The first is the memory output. The second set of lines carries one of the following: (1) the output of the program-loading ROM, (2) external data, (3) a status word (consisting of SP1-4 and four status bits), or (4) an 8-bit word produced by the terminal identification plug.

Address Net.

The address net is a set of 14 gates. Their purpose is to select a 14-bit address and apply it to memory. Four bits from the address net (0-3) are also routed to the ROM in order to read the loading program. Note that positions 0-7 select the byte locations within a page and positions 8-13 select the page number.

There are four basic sources of the address gated through the net to memory. They are the stack pointer, the memory address register, an external input address, and the IWL that selects an index register. The conditions under which each of these inputs are accepted are described in the following paragraphs.

Inputs from the Stack Pointer

Stack pointer output bits SP-1 through SP-4 are routed to four of the five low-order positions of the net. During both cycles I1 and I2, the stack pointer contents are accepted if no external entry is present. The four-bit output of the stack pointer occupies positions 1-4 of the net, and bit 5 is a fixed one to establish the lowest address as 40. Bit 0 is directly controlled by the I2 signal. During I1, bit 0 is a zero, causing the address to be even; during I2, bit 0 is a one, causing the address to be odd.

During the first half of cycle I1 and cycle I2, the stack pointer input selects memory locations from which instruction addresses are read. During the second half of each of these cycles, the stack pointer input selects the memory location in which an updated instruction address is to be restored. This places the instruction address in the location specified by the stack pointer.

Note that the lowest stack pointer address that can be used is 40 and the highest is 77. These locations are always assigned to be used for stack pointer addresses.

Inputs from the Memory Address Register

A description of how an address is formed in the memory address register was presented earlier in this section. Once this address is formed, it is available to the address net.

During instruction cycles I3 and I4, the objective is to read the instruction from memory. Note that the address was acquired during cycles I1 and I2 and is held in the memory address register. The MAR inputs are accepted during the first half of I3 to read the left half of the instruction and during the first half of I4 to read the right half of the instruction. Again, the low-order bit is not accepted from the address source; it is left at zero during I3 and switched directly to a one during I4.

External Address Inputs

The second half of cycle I3 and I4 are devoted to reading memory to refresh the CRT display, and the disc controller can gain access during cycles I0 and I3. If an external access request is present, the external address is accepted during the second half of these cycles. Note that only a 12-bit address is accepted; bits 12 and 13 are left at zero. This limits the pages to which the I/O controller and disc controller can gain access.

Selection of an Index Register

During instruction cycle E2 of instructions operating in the indexed address mode, the objective is to acquire the contents of an index register. The index register to be addressed is specified by bits 0-2 of the IWL, which are held in the operation register and applied to positions 0-2 of the address net. These addresses may be in any section, and a control bit (U) discussed earlier determines whether the section is 0 or another. During E2, the index register number is gated through positions 0-2 of the net, and if the U bit is set, the section number from the index section register is gated through positions 11-13.

Selection of a Data Address

During instruction cycle E3, the purpose of the address net is to select a memory address to acquire data to be operated on or store a result. The source of the address depends upon whether direct or indexed addressing is used.

If indexed addressing is used, a composite address made up of the IWR and the contents of an index register (read during cycle E2) is formed in MAR. This is gated through the address net during E3 to read or store data in memory.

If direct addressing is used, the IWR specifies the byte number within a page. This is held in the MAR. The section number is supplied by the section register. Since the level number is not supplied, the level is effectively zero. Thus, in the direct addressing mode, the address net accepts an address from the MAR and section register to select an address in page 00, 10, 20, 30, 40, 50, 60, or 70.

2-3. C4 PROCESSOR INSTRUCTION SET.

The C4 processor can execute 41 distinct machine language instructions. These are summarized in table 2-3. Note that each instruction is made up of two bytes: (1) the left instruction word (IWL), which holds the operation code, and (2) the right instruction word (IWR), which holds an address or an operand. Note also that the instructions are divided into four classes, the class (0, 1, 2, or 3) being determined by the high-order octal digit of the operation code. In the following paragraphs, the function performed by each instruction is described. It may be necessary to refer to the block diagram of the processor given earlier in this section to fully understand the data flow and results produced by each instruction. Note also that further descriptions and examples of use appear in the Assembler System manual.

TLJ - Test Literal and Jump.

The accumulator is compared to the literal, and the result is indicated in the condition register. If they are equal, a jump forward (+) or a jump backward (-) up to 15 instruction locations (decimal) is performed. If, however, they are not equal, the next sequential instruction is executed. (Timing: 3 microseconds if the jump is not performed, 4 microseconds if the jump is performed.)

Table 2-3. C4 Processor Instruction Set

Instruction	Format
	IWL: 7 6 5 4 3 2 1 0 — IWR: 7 6 5 4 3 2 1 0
CLASS 0:	
TLJ - Test Literal and Jump (0NN)	0 0 0 | JUMP COUNT | * | LITERAL
*0-Jump Forward, 1-Jump Backward	
TMJ - Test Mask and Jump (0NN)	0 0 1 | JUMP COUNT | * | MASK
TLX - Test Literal and Exit (000)	0 0 0 0 0 0 0 0 | LITERAL
TMX - Test Mask and Exit (040)	0 0 1 0 0 0 0 0 | MASK
CLASS 1:	
BRU - Branch Unconditional (10N)	0 1 0 0 0 | LEVEL | ADDRESS | 0
BRE - Branch on Equal (10N)	0 1 0 0 0 | LEVEL | ADDRESS | 1
BRH - Branch on High (11N)	0 1 0 0 1 | LEVEL | ADDRESS | 0
BRL - Branch on Low (11N)	0 1 0 0 1 | LEVEL | ADDRESS | 1
SBU - Stack and Branch Unconditional (12N)	0 1 0 1 0 | LEVEL | ADDRESS | 0
SBE - Stack and Branch on Equal (12N)	0 1 0 1 0 | LEVEL | ADDRESS | 1
SBH - Stack and Branch on High (13N)	0 1 0 1 1 | LEVEL | ADDRESS | 0
SBL - Stack and Branch on Low (13N)	0 1 0 1 1 | LEVEL | ADDRESS | 1
EXB - Exit and Branch (16N)	0 1 1 1 0 | LEVEL | ADDRESS | 0
EXU - Exit Unconditional (140)	0 1 1 0 0 0 0 0 | 0 0 0 0 0 0 0 0
SAM - Select Auxiliary Memory (14N)	0 1 1 0 0 | BLOCK | 0 0 0 0 0 0 0 0
SMS - Set Memory Section (150)	0 1 1 0 1 0 0 0 | 0 0 | SECTION | 0 0 0

Table 2-3. C4 Processor Instruction Set (Continued)

Instruction	Format
	IWL: 7 6 5 4 3 2 1 0 — IWR: 7 6 5 4 3 2 1 0
SMC - Set Memory Control (151)	0 1 1 0 1 0 0 1 | Y | 0 | 0 0 0 0 0 0
SSC - Set Memory Section and Control (152) . . .	0 1 1 0 1 0 1 0 | Y | 0 | SECTION | 0 0 0
Y = 0 - Reset U = 1 - Set U	
SAC - Set Arithmetic Condition (153)	0 1 1 0 1 0 1 1 | 0 0 0 0 0 0 0 0
LSW - Load Sense Switches (154) (or) EMP - Enable Memory Parity	0 1 1 0 1 1 0 0 | 0 0 0 0 0 0 0 0
LPS - Load Processor Status (155).	0 1 1 0 1 1 0 1 | 0 0 0 0 0 0 0 0
DPI - Disable Processor Interrupt (156).	0 1 1 0 1 1 1 0 | 0 0 0 0 0 0 0 0
EPI - Enable Processor Interrupt (156)	0 1 1 0 1 1 1 0 | 0 0 0 0 0 0 0 1
CPI - Clear Processor Interrupt (156).	0 1 1 0 1 1 1 0 | 0 0 0 0 0 0 1 0
TRM - Load Terminal Identification (157)	0 1 1 0 1 1 1 1 | 0 0 0 0 0 0 0 0
LDA - Load Accumulator (200) . . . . . (Immediate Address)	1 0 0 0 0 0 0 0 | LITERAL
LDA - Load Accumulator (210) . . . . . (Direct Address)	1 0 0 0 1 0 0 0 | ADDRESS
LDA - Load Accumulator (21N) . . . . . (Indexed Address)	1 0 0 0 1 | IR | PAGE ADDRESS | Y | Y
YY - Index Register Control	
LDX - Load Index Register (20N).	1 0 0 0 0 | IR | LITERAL
LIA - Load Instruction Address (22X)	1 0 0 1 0 | IR | LITERAL
STA - Store Accumulator (230). . . . . (Direct Address)	1 0 0 1 1 0 0 0 | ADDRESS

Table 2-3. C4 Processor Instruction Set (Continued)

Instruction	Format
	IWL: 7 6 5 4 3 2 1 0 — IWR: 7 6 5 4 3 2 1 0
STA - Store Accumulator (23X). . . . . (Indexed Address)	1 0 0 1 1 | IR | PAGE ADDRESS | Y | Y
ADA - Add to Accumulator (240) . . . . . (Immediate Address)	1 0 1 0 0 0 0 0 | LITERAL
ADA - Add to Accumulator (250) . . . . . (Direct Address)	1 0 1 0 1 0 0 0 | ADDRESS
ADA - Add to Accumulator (25N) . . . . . (Indexed Address)	1 0 1 0 1 | IR | PAGE ADDRESS | Y | Y
ADX - Add to Index Register (24N).	1 0 1 0 0 | IR | LITERAL
SUA - Subtract from Accumulator (260). . . . . (Immediate Address)	1 0 1 1 0 0 0 0 | LITERAL
SUA - Subtract from Accumulator (270). . . . . (Direct Address)	1 0 1 1 1 0 0 0 | ADDRESS
SUA - Subtract from Accumulator (27N). . . . . (Indexed Address)	1 0 1 1 1 | IR | PAGE ADDRESS | Y | Y
SUX - Subtract from Index Register (26N)	1 0 1 1 0 | IR | LITERAL
CLASS 3:	
ANA - Logical AND to Accumulator (300) . . . . . (Immediate Address)	1 1 0 0 0 0 0 0 | LITERAL
ANA - Logical AND to Accumulator (310) . . . . . (Direct Address)	1 1 0 0 1 0 0 0 | ADDRESS
ANA - Logical AND to Accumulator (31N) . . . . . (Indexed Address)	1 1 0 0 1 | IR | PAGE ADDRESS | Y | Y
SAN - Shift and Logical AND to Accumulator (30N)	1 1 0 0 0 | SHIFT COUNT | LITERAL
ERA - Exclusive OR to Accumulator (320). . . . . (Immediate Address)	1 1 0 1 0 0 0 0 | LITERAL
ERA - Exclusive OR to Accumulator (330). . . . . (Direct Address)	1 1 0 1 1 0 0 0 | ADDRESS
ERA - Exclusive OR to Accumulator (33N). . . . . (Indexed Address)	1 1 0 1 1 | IR | PAGE ADDRESS | Y | Y

Table 2-3. C4 Processor Instruction Set (Continued)

Instruction	Format
	IWL: 7 6 5 4 3 2 1 0 — IWR: 7 6 5 4 3 2 1 0
SER - Shift and Exclusive OR to Accumulator (32N)	1 1 0 1 1 | SHIFT COUNT | LITERAL
IRA - Inclusive OR to Accumulator (360). (Immediate Address)	1 1 1 1 0 0 0 0 | LITERAL
IRA - Inclusive OR to Accumulator (370). (Direct Address)	1 1 1 1 1 0 0 0 | ADDRESS
IRA - Inclusive OR to Accumulator (37N). (Indexed Address)	1 1 1 1 1 | IR | PAGE ADDRESS | Y Y
SIR - Shift and Inclusive OR to Accumulator (36N)	1 1 1 1 0 | SHIFT COUNT | LITERAL
CPA - Compare Accumulator (340). (Immediate Address)	1 1 1 0 0 0 0 0 | LITERAL
CPA - Compare Accumulator (350). (Direct Address)	1 1 1 0 1 0 0 0 | ADDRESS
CPA - Compare Accumulator (35N). (Indexed Address)	1 1 1 0 1 | IR | PAGE ADDRESS | Y Y
CPX - Compare Index Register (34N)	1 1 1 0 0 | IR | LITERAL

TMJ - Test Mask and Jump.

The IWR is used as a mask to designate the accumulator bits to be tested. When the selected bits are all ones, the condition register is set to equal. When the mask bit is zero, the corresponding accumulator bit is ignored. If the condition register is set to equal, a jump forward (+) or jump backward (-) up to 15 instruction locations is performed. If the condition register is not set to equal, the next sequential instruction is executed. (Timing: 3 microseconds if the jump is not performed, 4 microseconds if the jump is performed.)

TLX - Test Literal and Exit.

The accumulator is compared to the literal, and the result is indicated in the condition register. If they are not equal, the next sequential instruction is performed. If they are equal, a condition exit is performed to complete the return linkage established by the stack and branch instruction executed last. The stack pointer is decremented to the preceding level, which contains the address of the last stack and branch

instruction executed. This address is then incremented by 2 to establish the address of the instruction following the stack and branch instruction. This is the instruction address where the processing continues. (Timing: 3 microseconds if the jump is not performed, 4 microseconds if the jump is performed.)

TMX - Test Mask and Exit.

The IWR is used as a mask to designate the accumulator bits to be tested. When the selected bits are all ones, the condition register is set to equal. When the mask bit is zero, the corresponding accumulator bit is ignored. If the condition register is set to equal, a conditional exit is performed to complete the return linkage established by the stack and branch instruction executed last. The stack pointer is decremented to the preceding level, which contains the address of the last stack and branch instruction executed. This address is then incremented by 2 to establish the address of the instruction following the stack and branch instruction. This is the instruction address where processing continues. (Timing: 3 microseconds if the jump is not performed, 4 microseconds if the jump is performed.)

BRU - Branch Unconditional.

An unconditional branch is performed to the address specified. The branch address can be any location within the current section of memory. (Timing: 4 microseconds.)

BRE - Branch on Equal.

A branch to the address specified is performed when the condition register, set by a previous instruction, is found to be set to equal. If this condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the branch is not performed, 4 microseconds if the branch is performed.)

BRH - Branch on High.

A branch to the address specified is performed when the condition register, which has been set by a previous instruction, is found to be set to high. If this condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the branch is not performed, 4 microseconds if the branch is performed.)

BRL - Branch on Low.

A branch is performed when the condition register, set by a previous instruction, is found to be set to low. If this condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the branch is not performed, 4 microseconds if the branch is performed.)

SBU - Stack and Branch Unconditional.

A stack and branch operation is performed regardless of the setting of the condition register. The current instruction address is stored and the stack pointer is incremented to the next level. (Timing: 3 microseconds.)

SBE - Stack and Branch on Equal.

A stack and branch operation is performed when the condition register, set by a previous instruction, is found to be set to equal. If the condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the stack and branch is not performed, 4 microseconds if the stack and branch is performed.)

SBH - Stack and Branch on High.

A stack and branch operation is performed when the condition register, set by a previous instruction, is found to be set to high. If the condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the stack and branch is not performed, 4 microseconds if the stack and branch is performed.)

SBL - Stack and Branch on Low.

A stack and branch operation is performed when the condition register, set by a previous instruction, is found to be set to low. If the condition is not met, the next sequential instruction is executed. (Timing: 3 microseconds if the stack and branch is not performed, 4 microseconds if the stack and branch is performed.)

EXB - Exit and Branch.

The exit and branch instruction branches to the address specified and decrements the stack pointer to the preceding level. The address specified in the operand is the instruction address within the current section where the processing continues. (Timing: 4 microseconds.)

EXU - Exit Unconditional.

This form of exit performs the return linkage established by the stack and branch instruction executed last. The stack pointer is decremented to the preceding stack level, which contains the address of the last stack and branch instruction executed. This address is then incremented by 2 to establish the address of the instruction following the stack and branch instruction. This is the instruction address where processing continues. (Timing: 4 microseconds.)

SAM - Select Auxiliary Memory.

This instruction provides a means of selecting memory areas beyond the 16K that can be addressed by the 14-bit memory address discussed earlier. It is used only in terminals equipped with memories larger than 16K. The low-order octal digit of the IWL selects an 8K memory block, as follows: A "1" chooses the second 8K block, a 2 the third 8K block, a 3 the fourth 8K block, etc. (Timing: 3 microseconds.)

SMS - Set Memory Section.

The set memory section instruction provides a means of transferring from the current section of memory to another section. A branch instruction (branch, stack and branch, or exit and branch) preceded by an SMS instruction will transfer to the address defined by the branch address and the section specified in the Set Memory Section instruction operand. (Timing: 4 microseconds.)

SMC - Set Memory Control.

This instruction controls the state of the U and V bits. A description of how the U and V bits affect memory addressing appeared earlier in this section. It is explained in detail in the Assembler Manual. (Timing: 4 microseconds.)

SSC - Set Memory Section and Control.

This instruction performs both the set memory section operation of SMS and the set memory control operation of SMC. (Timing: 4 microseconds.)

SAC - Set Arithmetic Condition.

This instruction forces the processor condition register to a condition that depends on the state of bits 4 and 5 of the accumulator at the time of the SAC instruction, as follows:

ACCUMULATOR BITS 5	4	CONDITION FORCED
1	0	Equal
0	1	High
0	0	Low
1	1	Equal

(Timing: 4 microseconds)

LSW - Load Sense Switches.

This instruction loads the state of 8 toggle switches into the accumulator. This instruction is available only for those units with the sense switch option. (Timing: 4 microseconds.)

EMP - Enable Memory Parity.

This instruction enables the parity checking circuits in the memory and clears any previous parity error. This instruction is used only with the memory that has parity checking capability. (Timing: 4 microseconds.)

LPS - Load Processor Status.

This instruction loads a processor status word into the accumulator. (Timing: 4 microseconds.)

DPI - Disable Processor Interrupt.

This instruction inhibits the automatic stack and branch that occurs upon receipt of an interrupt. It does not disable the interrupt signal applied to the processor. If the Disable Processor Interrupt instruction is executed and the interrupt is subsequently activated, the automatic stack and branch operation will occur only after the execution of an EPI instruction. (Timing: 4 microseconds.)

EPI - Enable Processor Interrupt.

This instruction enables the processor to sense an interrupt input and perform the automatic stack and branch to page 03-000 upon receipt of an interrupt. Certain conditions inhibit the interrupt; they were discussed earlier in this manual.

CPI - Clear Processor Interrupt.

An interrupt overflow condition occurs when a second interrupt occurs before the previous interrupt causes the automatic stack and branch operation. A Clear Interrupt instruction clears the interrupt overflow indicator. (Timing: 4 microseconds.)

TRM - Load Terminal Identification.

Loads the accumulator with the identification assigned to this terminal by the program interlocking plug, which is installed on the processor board. (Timing: 4 microseconds.)

LDA - Load Accumulator.

Loads the accumulator with the value specified by the immediate, the direct, or the indexed address. (Timing: 4 microseconds when immediate address form is used, otherwise, 6 microseconds.)

LDX - Load Index Register.

Loads the specified index register with the IWR. (Timing: 4 microseconds.)

LIA - Load Instruction Address.

Transfers the 8 least significant bits of the current instruction address to the specified index register. If the IWR is zero, the section and page portions of the current instruction address are transferred to the accumulator. If the IWR is not zero, the IWR is transferred to the accumulator. (Timing: 4 microseconds.)

STA - Store Accumulator.

Stores the contents of the accumulator in the address specified by the direct or indexed address. (Timing: 6 microseconds.)

ADA - Add to Accumulator.

Adds to the accumulator the value specified by the immediate, the direct, or the index address. (Timing: 4 microseconds when immediate address form is used, otherwise, 6 microseconds.)

ADX - Add to Index Register.

Adds the IWR to the index register specified. (Timing: 4 microseconds.)

SUA - Subtract from Accumulator.

Subtracts from the accumulator the value specified by the immediate, the direct, or the indexed address. (Timing: 4 microseconds when the immediate address form is used, otherwise, 6 microseconds.)

SUX - Subtract from Index Register.

Subtracts the IWR from the index register specified. (Timing: 4 microseconds.)

ANA - Logical AND to Accumulator.

Logically ANDs with the accumulator the value specified by the immediate, direct, or indexed address. (Timing: 4 microseconds when the immediate address form is used, otherwise, 6 microseconds.)

SAN - Shift and Logical AND to Accumulator.

This form of the logical AND instruction performs a right circular shift of the bits in the accumulator, by the number of positions specified in the shift count, before the logical AND of the IWR and the accumulator is performed. (Timing: 4 microseconds.)

ERA - Exclusive OR to Accumulator.

Performs an Exclusive OR of the accumulator and the value specified by the immediate, direct, or indexed address. (Timing: 4 microseconds when the immediate address form is used, otherwise, 6 microseconds.)

SER - Shift and Exclusive OR to Accumulator.

This form of the Exclusive OR instruction performs a right circular shift of the bits in the accumulator, by the number of positions specified in the shift count, before the Exclusive OR of the IWR and the accumulator is performed. (Timing: 4 microseconds.)

IRA - Inclusive OR to Accumulator.

Logically ORs the accumulator and the value specified by the immediate, direct, or indexed address. (Timing: 4 microseconds when the immediate address form is used, otherwise, 6 microseconds.)

SIR - Shift and Inclusive OR to Accumulator.

This form of the Inclusive OR instruction performs a right circular shift of the bits in the accumulator, by the number of positions specified in the shift count, before the inclusive OR of the IWR and the accumulator is performed. (Timing: 4 microseconds.)

CPA - Compare Accumulator.

Compares the contents of the accumulator to the value specified by the immediate, direct, or indexed address. The condition register value is changed to reflect the high, low, or equal result of the comparison. (Timing: 4 microseconds when the immediate address form is used, otherwise, 6 microseconds.)

CPX - Compare Index Register.

Compares the contents of the index register specified to the IWR. The condition register value is changed to reflect the high, low, or equal result of the comparison. (Timing: 4 microseconds.)

2-4. C4 VIRTUAL PROCESSOR INSTRUCTION SET.

With certain exceptions, which are described below, the C4 virtual processor executes the basic C4 instruction set described earlier. Of course, the C4VP also has several unique instructions for memory, processor, and I/O device control. These are also described below.

The first subject to be discussed is the instructions in the basic C4 processor set that either do not operate on the C4VP or that operate differently. These instructions are as follows:

a. Instruction 154, LSW - Load Sense Switches or EMP - Enable Memory Parity, can operate only as the latter in the C4VP.

b. Instruction 156, which is either DPI - Disable Processor Interrupt, EPI - Enable Processor Interrupt, or CPI - Clear Processor Interrupt in the basic C4 set, is used as a DDC - Device Demand Control Instruction in the C4VP.

c. Instruction 157, TRM - Load Terminal Identification in the basic C4 set, operates as RVP - Read VP State in the C4VP.

Next to be discussed are the instructions added to the C4VP repertoire to control the memory sections used or the processor program exchange. These instructions, along with the DDC - Device Demand Control and RVP- Read VP State, are listed in table 2-4 and are described below.

CSL - Card Select (Select Auxiliary Memory).

This instruction selects one of the eight possible memory blocks (0-7), specified by the IWR (OON), as the additional memory for the currently active processor program. The number of the block selected is a vital condition that is saved during processor program exchange instructions. (Timing: 4 microseconds.)

Table 2-4. Additional C4 Virtual Processor Instructions

Instruction	Format
	IWL: 7 6 5 4 3 2 1 0 — IWR: 7 6 5 4 3 2 1 0
CSL - Card Select (Select Auxiliary Memory) (141)	0 1 1 0 0 0 0 1 | 0 0 0 0 0 | MEMORY BLOCK
EPN - Enter Processor New (142)	0 1 1 0 0 0 1 0 | 1 0 0 0 0 | VP
EPO - Enter Processor Old (142)	0 1 1 0 0 0 1 0 | 0 0 0 0 0 | VP
EPC - Enter Processor Conditional (142) . . .	0 1 1 0 0 0 1 0 | 0 1 0 0 0 | VP
REL - Processor Release (143)	0 1 1 0 0 0 1 1 | 0 0 0 0 0 0 0 0
SCF/RCF - Set/Reset Condition Flag (144). . .	0 1 1 0 0 1 0 0 | 0 0 0 0 0 0 0 | *
* = 0 - Reset CF = 1 - Set CF	
DCS - Disk Card Select (External Auxiliary Memory Select) (145).	0 1 1 0 0 1 0 1 | 0 0 0 0 0 | MEMORY BLOCK
DDC - Device Demand Control (156)	0 1 1 0 1 1 1 0 | 0 0 0 0 0 | N
N = 0 Inhibit Disk VP Activation 1 Enable Disk VP Activation 2 Inhibit CBP VP Activation 3 Enable CBP VP Activation 4 Inhibit Activation of all VP's 5 Enable General Activation of all VP's 6 Inhibit Channel Processor Program 7 Enable Channel Processor Program	
RVP - Read VP State (157)	0 1 1 0 1 1 1 1 | 0 0 0 0 0 0 0 0

EPN/EPO/EPC - Enter Processor New/Old/Conditional.

These processor program exchange instructions are differentiated by the high-order octal digit of the IWR. The forms are:

a. Enter New (20n) - This instruction causes the vital conditions of the currently active processor program to be saved and enters the active program number into the active processor stack. The processor program number specified by the "n" in the IWR (20n) is loaded into the currently active processor register. In addition, the stack pointer is returned to SP-0 and the operating registers set to enable the new processor program operation. The first instruction executed is specified by the base memory locations P00-040 and P00-041 for the new program. The contents of these locations are updated prior to use. (Timing: 4 microseconds.)

b. Enter Old (00n) - This instruction causes the vital conditions of the currently active processor program to be saved, and the number of the program is entered into the active processor stack. The processor program specified by the IWR (00n) is loaded into the currently active processor register. The operating registers are loaded with the vital conditions that were saved for the new program, and the program execution begins immediately following the point at which it last released control. (Timing: 4 microseconds.)

c. Enter Conditional (10n) - The processor program number specified by the operand (10n) is activated as it is in an Enter New or Enter Old instruction. The activation type depends on the state of a flag at the time the specified processor program previously released control. If the flag was high when release was last executed, this instruction is performed as if it were an Enter New instruction. Otherwise, the instruction is executed as if it were an Enter Old instruction and the vital conditions are retrieved. (Timing: 4 microseconds.)

REL - Processor Release.

This instruction causes the vital conditions of the currently-active processor program to be saved and the processor program (VP) number at the top of the active processor stack to be loaded into the currently-active processor register. The operating registers are loaded with the saved vital conditions from the save area associated with the newly-activated processor program. Each processor program issues this instruction when it has fulfilled its immediate requirements. (Timing: 4 microseconds.)

SCF/RCF - Set/Reset Condition Flag.

This instruction sets or clears the condition flag. When a processor program releases control with the flag set high and is later specified in an Enter Conditional instruction, activation is accomplished as in an Enter New. Any form of an Enter New activation clears the condition flag. When a processor program releases control with the flag low and is later specified in an Enter Conditional instruction, activation of the processor program is that of an Enter Old instruction. (Timing: 4 microseconds.)

DCS - Disc Card Select.

This instruction enables one of the eight memory blocks specified by the "n" in the IWR (00n) to be selected as the additional memory for disc transfers. This enables the disc unit to perform read/write functions in any area of the maximum memory capacity. (Timing: 4 microseconds.)

DDC - Device Demand Control.

This instruction controls the ability of a device interrupt signal to activate its associated processor program (VP1 through VP3). A general clear condition activates the interrupt lockout condition, which disables all device interrupts. The host processor program (VP0), which is activated by a general clear, determines when device interrupts are allowed by executing this instruction to control the interrupt lockout conditions. (Timing: 4 microseconds.)

RVP - Read VP State.

This instruction loads the accumulator with the state data of the currently active processor program. The state data consists of the current processor program identity, the selected auxiliary memory block, memory parity status, and power failure status. The data format is as follows:

7	6	5	4	3	2	1	0
PS1	PS0	MS2	MS1	MS0	MP	PF	X

PS1 and PS0 are the number of the active VP.

MS2, MS1, and MS0 are the number of the selected auxiliary memory section.

MP is the memory parity error status bit.

PF is the power failure detected status bit.

(Timing: 4 microseconds.)

2-5. C5 PROCESSOR INSTRUCTION SET.

Although the C5 processor architecture and interrupt scheme are totally different from that of both the C4 and C4VP, it executes the basic C4 instruction set with only minor exceptions. The differences between the C4 and C5 instruction sets are as follows:

a. An instruction called EXI - Exit and Enable Interrupt (Operation Code 141) is available in the C5 processor set. Its format is shown in figure 2-12.

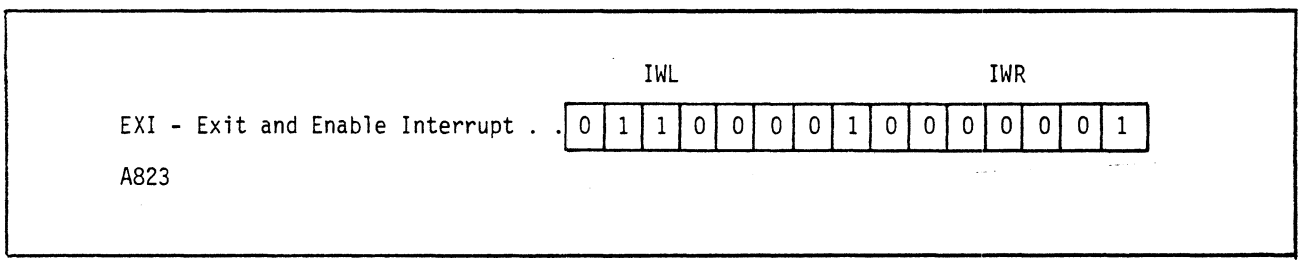


Figure 2-12. Format of EXI Instruction in C5 Processor Instruction Set

b. Instruction 154, LSW - Load Sense Switches or EMP - Enable Memory Parity, can operate only as EMP in the C5 processor.

c. Instruction 156, which is either DPI - Disable Processor Interrupt, EPI - Enable Processor Interrupt, or CPI - Clear Processor Interrupt in the basic C4 set is used only as DPI or EPI in the C5 processor. The CPI version is not available.

d. Instruction 157, TRM - Load Terminal Identification, is not available in the C5 processor instruction set.

e. Instruction 16N, Exit and Branch, has additional capability in the C5 processor. If the low-order bit of the IWR is a one, this instruction becomes Exit, Branch, and Enable Interrupt.

f. Instruction 22N, LIA - Load Instruction Address, is not available in the C5 processor instruction set.

g. Instruction 14N, SAM - Select Auxiliary Memory, is not available in the C5 processor set.

As shown above, the only instruction available in the C5 instruction set that is not used in the basic C4 set is EXI. Its description follows:

EXI - Exit and Enable Interrupt

This instruction performs the exit unconditional operation and, in addition, enables the interrupt circuits. Interrupt interrogation is inhibited during this instruction to prevent an interrupt from occurring until the next instruction is performed. (Timing: 4 microseconds.)

2-6. STANDARD C4 INPUT/OUTPUT INSTRUCTIONS.

As the name indicates, these instructions control the flow of information between the processor and the input/output devices. They are decoded and executed by the I/O controller or forwarded to the I/O device for execution.

All I/O instructions have a format in which the IWL is 17N; the "N" is a number that selects a specific I/O device. As soon as the processor determines that the operation code is in the "17" family, it allows the I/O controller to decode the "N" to operate one I/O device and to decode the IWR to determine the operation to be performed or forward the IWR to the selected device.

Operation of the I/O devices by the C4 processor in the Model 1501, 1501-40/43, 1502, and 1503 Intelligent Terminals is discussed in the following paragraphs. The specialized I/O devices used by the C4VP and C5 processors are described later in this section.

As mentioned above, the "N" in the 17N operation code selects I/O device. The number and devices currently used by the C4 processor and related I/O controller are as follows:

a. "0" selects the current cartridge tape drive.

b. "1" selects cartridge tape 1 of a pair. The IWR chooses the pair.

c. "2" selects cartridge tape 2 of a pair. Again, the IWR chooses the pair.

d. "3" selects the keyboard.

e. "4" selects the small CRT display.

f. "5" selects the serial I/O channel for use. Note that most units of peripheral equipment operate on the serial I/O channel. The unit to be operated is chosen by an address number sent over the I/O channel.

g. "6" selects what is called the communications interface. Available at this interface are the communications adapters and the LCRT display. The IWR chooses the specific device as well as designates the function to be performed.

h. "7" chooses the parallel I/O channel. Presently, only the disc units operate over this channel.

Cartridge Tape Instructions.

The cartridge tape drives are organized into pairs, and the C4 processor and I/O controller can operate up to 8 drives (four pairs). Section 1 of this manual describes the tape configuration available with each terminal.

When a source statement containing a cartridge tape drive instruction is entered for use by the Assembler System, it takes the form:

IOC,C#N;OYY

This is converted to machine language to mean: input/output instruction, device "N", IWR is OYY. The programmer would have placed a 0, 1, or 2 in the "N" position of the source statement and would have placed a 2-digit command in the "YY" positions. The codes available for use with the cartridge tapes are as follows:

170-OYY . . . Current tape channel
171-OYY . . . Tape cartridge 1
172-OYY . . . Tape cartridge 2

17N-000*. . . Start tape forward slow, with erase
17N-001 . . . Start tape forward slow
17N-002 . . . Start tape forward fast
17N-003 . . . Start tape reverse slow
17N-004 . . . Start tape reverse fast
17N-005 . . . Stop tape
17N-007 . . . Transfer byte, unconditional
17N-207 . . . Transfer byte, conditional (skip on busy)
17N-010 . . . Set write mode
17N-011 . . . Set read mode
17N-012 . . . Rewind
17N-016 . . . Select drive 1 if N = 1, drive 2 if N = 2, & load status
17N-026 . . . Select drive 3 if N = 1, drive 4 if N = 2, & load status
17N-036 . . . Select drive 5 if N = 1, drive 6 if N = 2, & load status
17N-046 . . . Select drive 7 if N = 1, drive 8 if N = 2, & load status

*Note: N = 0, 1, or 2.

Use of these instructions to program the cartridge tapes is discussed in Section 3, Programming Techniques.

Acquisition of the General Status Byte.

As shown in the list of cartridge tape instructions above, a "6" in the low-order octal position of the IWR acquires a general status byte. All but two bits of the general status byte concerns tape status.

The assembler source statement to acquire the general status byte would be:

```
IOC,C#n;0Y6
```

This would be interpreted as follows:

```
17N-0Y6 . . . . Load general status byte into accumulator.

Where:  N - Drive 0, 1, or 2 with the selected drive pair
And:    Y = Drive pair 0, 1, 2, 3, or 4.  Note that the "0" always
            chooses the currently-selected unit.
```

The status byte called for is then acquired and loaded into the accumulator. At that point, it can be examined bit-by-bit to determine status of the cartridge tapes. Shown below is the meaning of each bit in the general status byte:

```
Bit 0 . . . . . Keyboard error
    1 . . . . . Tape error
    2 . . . . . I/O status
    3 . . . . . Tape runaway
    4 . . . . . Tape cartridge out
    5 . . . . . Clip out
    6 . . . . . End of tape
    7 . . . . . Not used
```

Keyboard Instructions.

There are four instructions associated with the use of the keyboard. They can accept a keyboard character, sound the audible alarm, or acquire keyboard status. The form of the source statement provided to the Assembler System is as follows:

```
IOC,C#3;NNN
```

This is interpreted and converted to machine language to mean: select the keyboard and perform the function specified by the number in the NNN positions. The four instructions available are:

173-016 . . Read keyboard status. (See the general status byte above.)
173-007 . . Transfer byte, unconditional
173-207 . . Transfer byte, conditional (skip on busy)
173-013 . . Keyboard beep

The first instruction acquires the general status byte. As shown earlier, only bit 0 of the status byte is related to the keyboard. This bit, labeled keyboard error, indicates that a second keystroke has been entered before the first was accepted by the processor.

Instructions 173-007 and 173-207 are issued to read a keyboard character into the accumulator. The difference in these instructions is that the 007 version will wait until a keyboard character is entered, thus stalling the processor at the current instruction, while the 207 version will skip one instruction and continue if the keyboard input is not ready.

The last instruction is keyboard beep, which sounds the audible alarm for a very short time. Normally, this is used to acknowledge each keystroke so that the operator "hears" the keystroke. However, the instruction can be used to sound the audible alarm for any time interval chosen by the programmer.

Small CRT Display Instructions.

The small CRT display is controlled by an operation code of 174 in the IWL and a very complex function code in the IWR. This is entered as the following source statement:

IOC,C#4;FFF

The IWR is the function code portion of the instruction, and it determines what action will take place on the CRT. As shown in figure 2-13, the function code is further broken down into six parts. The "S" and "L" bits combine to select the page to be displayed. The "I" bit is the interleave bit. In the 8-line mode, it causes the first four display lines to be double-spaced and the second four display lines to be shown in the vacated lines. In the 4-line mode, it controls which locations of the page are displayed (I=1, locations 000-177; I=0, locations 200-377).

The "U" bit will cause any character in which bit 6 of the character code is a one to be underscored. If the "U" bit is a zero, bit 6 of each character code can select the alternate character set. The "D" bit will disable the CRT, and the "M" bit is the mode select bit to select either an 8-line or a 4-line display.

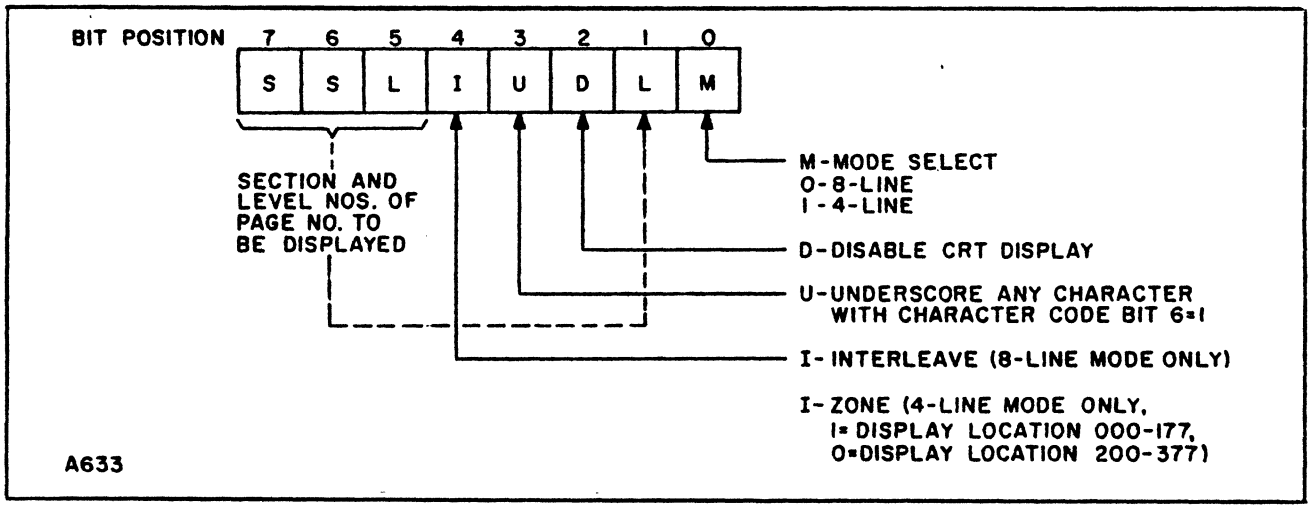


Figure 2-13. Small CRT Display Function Codes

Serial Input/Output Channel Instructions.

All units of peripheral equipment except the disc units operate on the serial I/O channel. This channel is operated by the instructions in the 175 group, which are coded in a source statement as:

IOC,C#5;ONN

The NN specifies the instruction to be executed, as follows:

175-000 . . . . Set transmit
175-001 . . . . Transfer byte, unconditional
175-201 . . . . Transfer byte, conditional (skip on busy)
175-002 . . . . Transmit data byte
175-003 . . . . Transmit control byte
175-004 . . . . Stop transmit
175-006 . . . . Inhibit line
175-007 . . . . Set device address
175-010 . . . . Set master mode
175-011 . . . . Set slave mode

The technique involved in programming the I/O units on the serial I/O channel are discussed in detail in Section 3 of this manual; however, there are several facts that will aid the reader in understanding operation of the serial I/O channel. They are:

a. Each unit of peripheral equipment has a 2-digit (one byte) octal address to distinguish it from other units on the channel.

b. An intelligent terminal establishes contact with a peripheral unit by transmitting the address of the desired unit on the channel to all units; only the selected unit responds.

c. After contact is established, the instruction Transmit Control Byte (175-003) is executed to send a control byte from the accumulator over the serial I/O channel to the selected unit. This byte is decoded by the peripheral unit to determine what action is to be taken. Contents of the control bytes are discussed in the Programming Techniques section.

d. Normally, the relationship of the terminal to the peripheral unit is master to slave; however, a terminal can be programmed to act as a slave of the master terminal.

Dual I/O Channel Instructions.

Most of the terminals can be equipped to operate two serial I/O channels, but only one can be active at any one time. Therefore, two instructions are provided to switch between channels. It should be noted that channel 1 is initially selected when the equipment is turned on.

The instructions and their source statements are shown below. All other dual SIO channel instructions are the same as the SIO instruction above.

```
IOC,C#3;ONN

173-006 . . . . Switch to channel 1
173-014 . . . . Switch to channel 2
```

Communications Interface Instructions.

Up to this point, the I/O instructions discussed were those that are decoded and executed by the I/O controller. The instructions using the communications interface are handled differently, however. In this case, the I/O controller determines only that the communications interface has been selected and forwards the IWR to the interface lines. Units that can be operated at this interface include the large CRT and any one of three communications adapters. These units then examine the IWR to determine which unit is selected and what function is to be performed. Source statements are described below.

Model 1534 Instructions

For the Model 1534 Asynchronous Communications Adapter, the statement is:

IOC,C#6;01N

In this statement, the 01 selects the adapter and the "N" is a number that chooses the function to be performed by the adapter, as indicated in the instructions below.

176-010 . . . . Transmit byte
176-011 . . . . Write control word held in accumulator
176-012 . . . . Read data byte
176-013 . . . . Read error byte
176-014 . . . . Read status byte into accumulator
176-015 . . . . Disconnect data set
176-016 . . . . Break received, reset
176-017 . . . . Connect data set

Model 1535 Instructions

For the Model 1535 Binary Synchronous Communications Adapter, the statement is:

IOC,C#6;02N

Again, the "N" selects the function the adapter is to perform, but the 02 selects the BSC adapter to perform that function. There are seven instructions that the adapter can perform; they are listed below.

176-020 . . . . Transmit byte
176-021 . . . . Write control word held in accumulator
176-022 . . . . Read data byte
176-023 . . . . Read error byte
176-024 . . . . Read status byte into accumulator
176-025 . . . . Load sync register
176-026 . . . . Load DLE register
176-027 . . . . Not used

Model 1539 Instructions

The third communications adapter available is called the Model 1539 Interruptible Communications Controller, and it can execute programs written for the 1535 as well as programs specifically prepared for the 1539. Hence, the source statements entered to program the 1539 may be either of the following:

IOC,C#6;00N, or
IOC,C#6;02N

Since only one communications adapter can be used in the terminal, there is no possibility of the 02N selecting the 1535. Therefore, the 1539 can respond to both 00N and 02N instructions, which are shown below.

176-000 . . . . Load sync register
176-001 . . . . Write control word 1
176-002 . . . . Send control byte
176-003 . . . . Read interface status
176-004 . . . . Load control register
176-005 . . . . Send fill register
176-006 . . . . Not used
176-007 . . . . Disable interrupt
176-020 . . . . Write data
176-021 . . . . Write control word 2
176-022 . . . . Read data
176-023 . . . . Read secondary status
176-024 . . . . Read primary status
176-025 . . . . Load fill register
176-026 . . . . Load alternate fill register
176-027 . . . . Enable interrupt

LCRT Instructions

The last unit available at the communications interface is the large CRT display. It may be installed in a terminal along with a communications adapter. Thus, if the large CRT is used, two units may be available at the communications interface, whereas only one communications adapter unit can be present if the LCRT is not used.

The source statement used to choose the LCRT and issue it commands is:

IOC,C#6;1NN

The "1" selects the LCRT and the "NNN" gives one of the instructions shown below.

176-100 . . . . Clear screen from cursor location
176-101 . . . . Write and increment cursor
176-102 . . . . Enable low intensity mode
176-103 . . . . Assume single intensity mode
176-104 . . . . Enable 24-line mode
176-105 . . . . Enable 12-line mode
176-106 . . . . Enable 32-line mode
176-107 . . . . Read and increment cursor
176-110 . . . . Backspace cursor
176-111 . . . . Read without cursor increment
176-112 . . . . Blank LCRT screen
176-113 . . . . System status request I (see response below)
176-114 . . . . System status request II (see response below)
176-115 . . . . Move cursor up one line
176-116 . . . . Move cursor down one line
*176-12X . . . . Exit C4 mode
*176-13X . . . . Enter C4 mode
*176-14X through 17X Position cursor, with accumulator contents as follows:
140 thru 147 = Lines 1 thru 8
150 thru 157 = Lines 9 thru 16
160 thru 167 = Lines 17 thru 24
170 thru 177 = Lines 25 thru 32

*X Indicates that the low-order octal digit has no significance.

Status Response I

Bits 0 thru 4 . . Vertical cursor position (line number)
Bit 5 . . . . . C4 mode
Bit 6 . . . . . 12-line mode
Bit 7 . . . . . 32-line; 128-character mode
Bits 5-7 all zero. . . . 24-line mode

Status Response II

Bits 0 thru 6 . . Horizontal cursor position (location in line)
Bit 7 . . . . . Screen blank

Parallel I/O Channel Instructions.

The parallel I/O channel is connected directly to memory input/output lines, and memory access by a unit on this channel is integrated with the basic processor timing cycle. At present, only the disc controller utilizes the parallel I/O channel for data flow between the memory and the disc. All I/O functions are coordinated by the disc controller, the I/O controller being limited in this case to decoding the channel number and forwarding a selection signal to the disc controller.

A source statement required to select the parallel I/O channel is:

IOC,C#7;NNN

The 3-place number represented by the NNN positions is an instruction forwarded to the disc controller for further decoding. Instructions are listed below:

177-200 . . . . Seek
177-201 . . . . Read
177-202 . . . . Write with read-back check
177-203 . . . . Write prep
177-204 . . . . Clear
177-205 . . . . Load parameters
177-206 . . . . Restore
177-207 . . . . Read status (see response below)
177-210 . . . . Not used
177-211 . . . . Write protect
177-302 . . . . Write without read-back check

Status Byte

Bit 0 . . . . . Seek error
1 . . . . . Write check error
2 . . . . . Read-back error
3 . . . . . Interrupt
4 . . . . . Write protect enabled
5 . . . . . Ready to seek, read, or write
6 . . . . . Address error
7 . . . . . Second command error

2-7. SPECIAL C4 VIRTUAL PROCESSOR INPUT/OUTPUT INSTRUCTIONS.

The C4VP can perform all the standard I/O instructions described earlier. In addition, it has a special feature called the direct execute sequence (DES). This feature is similar to an interrupt sequence in which a processor is forced to perform a routine held in a specific memory section when it is interrupted. The memory location selected is related to the address of the channel that caused the interrupt.

There are four units involved in the DES scheme, as follows:

a. Local asynchronous communications channels.

b. Long-distance asynchronous communications channels.

c. Utility channels, which include such functions as programmable timers and error monitors.

d. A DES controller, which coordinates the actions of the channels mentioned above with processor activities.

Before the special I/O instructions are discussed, it is necessary to briefly review the organization of the C4VP. The following facts determine how the special I/O instructions operate:

a. The C4VP is organized as if it were four separate functional processors. Each is dedicated to a specific task and has the following priority:

   (1) VP1 has the highest priority and is assigned to the user's primary data processing operation.

   (2) VP2 is assigned to operate the disc units and has the second highest priority.

   (3) VP3 is the channel background processor and the DES channel processor. When used as the background processor, it has the third highest priority. However, when used as a DES channel processor, it has the highest priority of all processors, including VP1.

   (4) VP0 is called the host processor. It is active only when no other processor is in use, and it performs the "housekeeping" routines.

b. A priority network senses requests for processor use, and it switches processor programs as required to maintain the following priority scheme:

Highest: DES channels in VP3
↓ VP1
VP2
VP3
Lowest: VP0

c. There may be as many as 64 of the DES channels. Each has specific locations in memory in which its servicing routine begins.

d. A DES channel processor program is activated only when its associated channel requires servicing. As soon as the immediate channel requirements are met, the channel processor program issues a release command. Other processor programs are then allowed to function until this channel again requires service.

e. The DES channel processor programs have the highest priority. In addition, a channel processor program can activate a lower-level processor program, called the channel background processor program (VP3). This program performs some functions that are required after a channel is serviced.

The DES controller operates three types of communications channels. The channels are mounted on the following printed circuit boards:

a. Quadruple Asynchronous Communications, which provides four channels of communications with a MODEM and subsequently long-distance lines.

b. 16-Channel Local Communications, which provides 16 channels of local communications but cannot operate with a MODEM.

c. A utility board, which is not a communications board but operates on the channels in the same way. The utility board monitors terminal conditions, such as keyboard character ready, and informs the processor when these conditions exist.

DES Controller Instructions.

Channel processor programs contain instructions to control input/output data transfers, status transfer, and command transfer between the processor and associated communication channels. The instruction format is 176-04n. A description of the instructions along with the octal codes is given below:

a. 176-040, Write Data Byte, which transfers the contents of the accumulator to the selected channel.

b. 176-041, Read Data Byte, which transfers data from the selected channel into the accumulator.

c. 176-042, Read Status, which transfers a status word from the selected channel into the accumulator.

d. 176-043, Force Channel, which causes the channel selected by accumulator bits 0-5 to request activation.

e. 176-044, Load Channel Control, which loads control circuits in the selected channel with the contents of the accumulator.

f. 176-045, Read Channel Identity, which loads the accumulator with the number of the selected channel.

g. 176-046, Set CBP Request Flag, which sets the channel background processor request flag in the selected channel.

Quadruple Asynchronous Communications Channel Instructions.

Each of these boards holds four identical communications channels capable of operating with a MODEM to communicate over long distances. A channel is selected by a 6-bit channel identification number. This number is held in accumulator positions 0-5 during a Force Channel instruction (176-043) or in read by the processor during a Read Channel Identity instruction (176-045).

Once a channel is selected, the processor sends it commands by using the 176-044, Load Channel Control instruction. In this case, the accumulator positions 0-3 hold the following commands:

00 - Set Data Terminal Ready
01 - Clear Data Terminal Ready
02 - Set Receive Mode
03 - Clear Receive Mode
04 - Set Transmit Mode
05 - Clear Transmit Mode
06 - Set Reverse Channel
07 - Clear Reverse Channel
10 - Set Rate High
11 - Clear Rate High
12 - Start Timer
13 - Stop Timer
14 - Terminate Transmit

The Read Status instruction (176-042) can read a status word from the selected channel to determine the reason the channel requested attention. This instruction returns the following status byte to the accumulator:

Bit 7 - Transmit interrupt
6 - Timer interrupt
5 - Ring indicator interrupt
4 - Not used
3 - Reverse channel
2 - Error detected
1 - Carrier detected
0 - Data Set Ready

Local Asynchronous Communication Channel Instructions.

These channels are similar to those discussed above except that they cannot operate with a MODEM and therefore are limited to local communications. In response to the 176-044, Load Channel Control instruction, however, they decode only bit 0 of the accumulator, as follows:

0 - Disable Transmit
1 - Enable Transmit

It should also be noted that there is no status word available from the local channels. The status of the condition register must be examined as described in Section 3 of this manual.

Utility Channel Instructions.

Seven of the eight available channels are used, and their assignments are as follows:

a. Channels 0 and 1, system timers.

b. Channel 2, report memory parity errors.

c. Channel 3, report power faults.

d. Channel 4, report keyboard inputs ready and activate servicing routine.

e. Channel 5, report SIO channel transfer ready and activate servicing routine.

f. Channel 6, report cartridge tape transfers ready and activate servicing routine.

The utility board monitors important conditions in the terminal and notifies the processor when these conditions require attention. When this board is used, its channels have the highest priority of all DES channels and thus the highest possible priority within the C4VP.

Channels 0 and 1

Channels 0 and 1, which are called the system timer channels, each provide selectable time intervals determined by accumulator bits 0-2. Since the timers are under program control, they are first selected by a Force Channel instruction and then are loaded by a Load Channel Control instruction. The timers each activate their channel processor program when the selected interval of time has expired.

These two channels each provide timing signals of 100 milliseconds, 1 second, and 10 seconds. Accumulator bits 0 and 1 select the interval. For example, if all the counters are operated in sequence, the 10-second time interval is produced. This is selected by a "1" in ACC-0 and 1. A "2" selects a 1-second interval, and a "3" selects the 100-millisecond interval. A Load Channel Control instruction (176-044) is executed, allowing bits ACC-0-2 to either start or stop the timers and select some time interval or stop the timer.

Channels 2 and 3

Channels 2 and 3 are used for error interrupts, with channel 2 handling memory parity errors and channel 3 power failure. Both channels activate their channel processor program upon detection of the associated error input.

To start channel operation, the processor program executes a Force Channel instruction. This selects either channel 2 or 3 and activates the channel processor program. In turn, this program issues a Load Channel Control instruction. Accumulator bits 1 and 7 are required to control the channel. Bit 1 either enables or disables the channel. Bits 0 and 1 of the accumulator are also decoded into four commands and made available at the board output, but this output is not used. The channels are identical, with the INT-2 (interrupt 2) signal being supplied for memory parity errors and the INT-3 signal for power supply failures.

A status byte can also be sent to channel 2 to be transmitted to the processor. Currently, this status byte is not being used, but if they were used, the status bits 4 through 7 could represent the following:

a. Status bit 7 = error in memory board 3.

b. Status bit 6 = error in memory board 2.

c. Status bit 5 = error in memory board 1.

d. Status bit 4 = error in memory board 0.

Channels 4, 5, and 6

Channels 4, 5, and 6 each contain a timer and an interrupt circuit. Inputs to these channels are as follows:

a. INT-4, indicates that the keyboard has an input ready for the processor.

b. INT-5, indicates that the SIO channel is ready to transfer a byte.

c. INT-6, indicates that the selected cartridge tape unit is ready to transfer a byte.

When each of the above conditions is present, the associated channel activates its channel processor program. This program then uses the normal I/O instructions to accomplish the data transfer and releases control of the channel. Since the three channels are identical, only channel 4 is described.

In order to start channel operation, the processor program executes a Force Channel instruction to select the channel and then a Load Channel Control instruction to establish the operating conditions. The load command signal is applied to both the timer section and the keyboard interrupt section. Accumulator bits 0-3 then control channel operation, as follows:

a. If bit ACC-3 is a one, it enables the keyboard signal to produce interrupt requests; if it is a zero, requests are disabled.

b. In the case of the timer, bit ACC-2 starts the timer if a one and stops it if a zero. ACC-0 and 1 select the timer interval for the three channels and are coded as follows:

ACC-0 and 1 = 1, the interval is 20 milliseconds.

= 2, the interval is 10 msec.

= 3, the interval is 1 msec.

2-8. C5 INPUT/OUTPUT INSTRUCTIONS.

In its present applications in the Model 1536 Intelligent Communications Adapter and in the Model 1580/1580-S Intelligent Terminal, the C5P operates with an input/output controller that is totally different from the C4 and C4VP I/O controller. Therefore, the set of C5P I/O instructions differs significantly from the C4 I/O instructions.

The overall organization of the instruction format is shown in figure 2-14. It operates in a manner similar to that of the C4. When the processor finds that the IWL is "17N", it allows the I/O controller to decode the "N" to select a device to be operated and then either decode the IWR to select a function or forward the IWR to an I/O device for decoding.

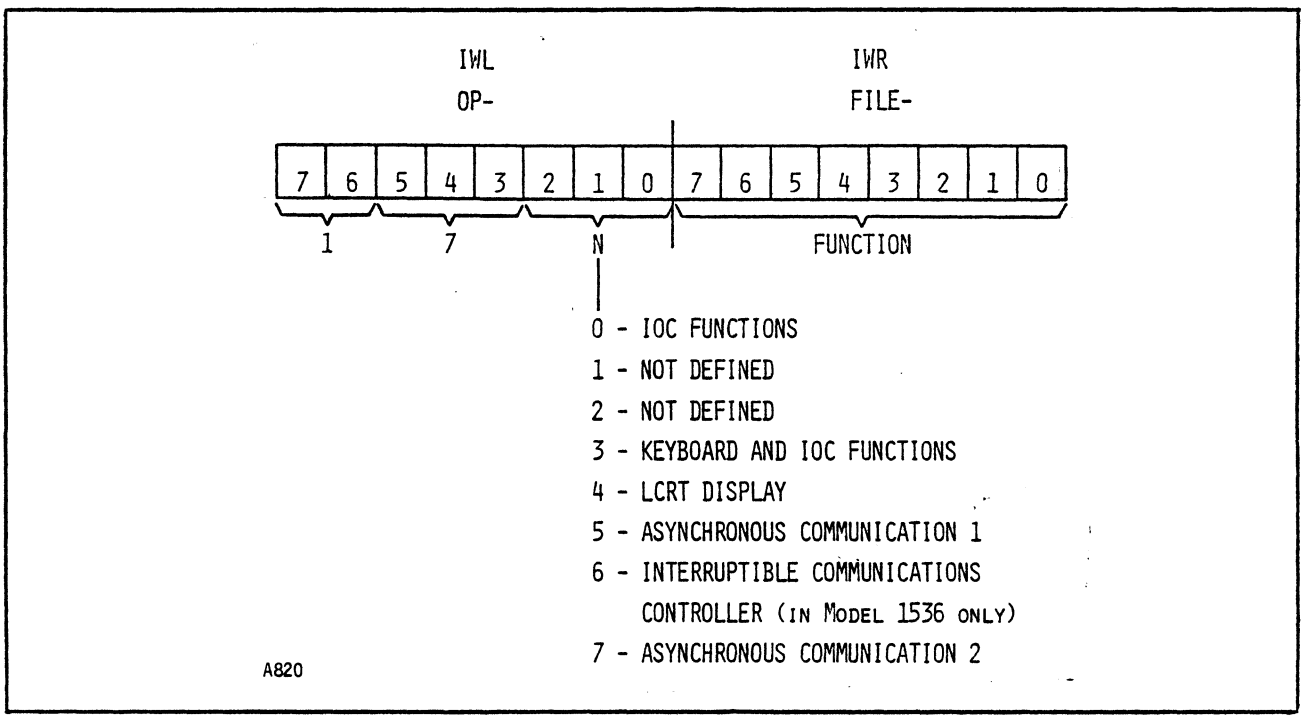


Figure 2-14. C5 I/O Instruction Format

As shown in figure 2-14, the I/O controller decodes the three low-order bits of the operation code into the octal values of 0 through 7, some of which are not currently used. The first group of instructions to be discussed are those in the 170-NNN group.

I/O Controller Function Instructions.

There are eight instructions in this category, as follows:

Code	Instructions
170-000	Clear Interrupt Timer
1	Load Timer Short
2	Load Timer Long
3	Clear Interrupt Mask
4	Load Interrupt Mask
5	Read Status
6	Read Terminal Identification
7	Read ROM

Timer Instructions

Within the I/O controller, a system timer is loaded with the desired time interval, which ranges from 104 microseconds to 4.23 seconds, and an interrupt signal is generated when the interval expires. The rate at which the timer is clocked is determined by the load instructions, 170-001 (long, 16.6 milliseconds) or 170-002 (short, 104 microseconds). The time interval is determined by the contents of the accumulator, which are loaded into the timer. The interrupt generated by the timer can be reset by the clear timer instruction, 170-000, but is normally cleared by the interrupt 0 acknowledgement.

Interrupt Mask Instructions

The I/O devices and I/O controller circuits that generate interrupts 0 through 7 must be enabled before the signals can be activated. Each interrupt supplies a memory address of a related interrupt servicing routine. Initially, all interrupts are inhibited due to the general clear condition. Unless a board slot enable signal is also required for a particular interrupt, the interrupts are then selectively enabled under program control by the load mask instruction (170-004). The mask bits are loaded from the accumulator contents. The clear mask instruction (170-003) clears the respective mask bit and inhibits the associated interrupt.

Read Status Instruction

This instruction (170-005) gates a 4-bit status word to the processor. The word indicates the state of the power supply, memory parity, and keyboard signal lines monitored by the I/O controller.

Read Terminal Identification Instruction

An address plug within the controller provides an identifying address for a specific unit in multi-unit installations. The read terminal ID instruction causes the address to be read and applied to the processor.

Read ROM Instruction

Program loading is initiated by the PROGRAM LOAD switch. This causes the processor to activate the bootstrap ROM for loading the program. The source of the program is determined by the contents of the bootstrap ROM.

Keyboard and Interrupt Control Instructions.

Since the 171 and 172 operation codes are not used, the next instructions to be discussed are those in the 173-NNN series. There are nine instructions in this group, as follows:

Code	Instruction
173-001	Enable Slot 1
173-101	Disable Slot 1
173-002	Enable Slot 2
173-102	Disable Slot 2
173-003	Keyboard Beep
173-004	Keyboard Tone Burst
173-007	Keyboard Read
173-2NN	Select Mainline Register Set
173-3NN	Select Interrupt Register Set

Interrupt Control Instructions

The "slot" instructions enable or disable the ability of the communications or peripheral interface I/O devices to supply their interrupts to the I/O controller. Presently, these instructions operate only the asynchronous communication channels.

Keyboard Instructions

There are three instructions in this group; two sound the audible tone, and the third reads the keyboard stroke. They are discussed below.

The keyboard contains an encoder and, when a key is depressed, the encoder produces a 6-bit code corresponding to the physical location of the key. Other information sent from the keyboard is:

a. NUM-SH (number shift), which indicates that the keyboard has been shifted to the "upper case" mode and that the keyboard output represents the upper character on the key.

b. ALPHA-SH (alphabetic shift), which indicates that the keyboard is in the "lower case" mode in which the keyboard output represents the lower character on the key.

c. CONT'L-SH (control shift), which indicates that the control (CTRL) key has been depressed.

The processor is notified that the keyboard character is ready by either an interrupt 1 or by a status indication when it reads a status word. In response, the processor issues the 173-007 instruction to read the character available.

Tone instructions control a 600 Hz tone at the terminal; the speaker is in the front of the terminal, not the keyboard. The beep is a 13.3 millisecond interval, and the burst is an alternating on and off signal that lasts a total of 1.5 seconds.

Index Register Set Selection Instructions

An index register selection system is incorporated into the C5 processor and I/O controller. The index register locations are assigned in sets of seven and are located in the first half of page 00 of memory. The two register selection instructions allow selection of the mainline (173-2NN) or the interrupt (173-3NN) index register set. Except for set numbers 04 through 07, which are the stack pointer locations and are not to be selected for index register locations, the NN of the IWR is the set number selected and corresponds to the two most significant bits of the register set octal address location in page 00, as follows:

Set Number	Register Locations
00	001-007
01	011-017
02	021-027
03	031-037
through	through
17	171-177

Large CRT Instructions.

The large CRT display instructions are similar to some of those used by the C4 and C4VP units; however, the IWL is 174, not 176. When the I/O controller decodes a 174 in the IWL, it signals the CRT circuits and they decode the IWR. The contents of the IWR are decoded as follows:

a. ONN-Load Cursor Position. This instruction loads the cursor location into the cursor vertical address counter and cursor horizontal address counter. The vertical address is specified by the IWR of the instruction, which determines the line number (00-27 octal). The horizontal address is determined by the accumulator contents, which select the horizontal character position (000 to 117 octal).

b. 10N-Enable CRT Mode (10N). This instruction determines the number of lines to be displayed, as follows: N=0 - blank, 1 - 4 lines, 2 - 8 lines, 3 - 12 lines, 4 - 16 lines, 5 - 20 lines, and 6 - 24 lines.

c. 140-Status Request I. This instruction enables a status word containing a vertical cursor position and CRT mode information to be sent to the processor. The contents of the status word are:

Bits 0 - 4 . . . . . Vertical cursor position (line number).

Bits 5 - 7 . . . . . CRT mode - number of display lines.

d. 240-Status Request II. This instruction enables a status word containing horizontal cursor position and underline mode information to be sent to the processor. The contents of the status word are:

Bits 0 - 6 . . . . . Horizontal cursor position (location on a given line)

Bit 7 . . . . . Underline mode

e. 200-Enable Dual Intensity Mode. This instruction enables a specific character to be displayed at a lower intensity than normal. If the high-order bit of the character code is a one, the character is displayed at low intensity.

f. 201-Enable Underline. This instruction enables an underlined character to be displayed. If the high-order bit of the character code is a one, the character is underlined.

g. 30N-Move Cursor. This instruction controls the cursor location contained in the cursor vertical and horizontal address counters and thus moves the cursor. This results in the cursor being moved to a different line or position along a line as specified by N, as follows: N = 0 increment (forward), 1 - backspace, 2 - up, 3 - down, and 4 - home.

Communication Channel Instructions.

At the present time, the I/O controller associated with the C5 processor can operate two asynchronous communications channels. One is selected by the 175 code and the second by a code of 177. When either of these codes is detected by the I/O controller, the controller allows the respective I/O channel to decode the IWR. In turn, the IWR produces the following instructions:

Instruction Code	Meaning
175 or 177-NNN	
000	Set Data Terminal Ready (DTR)
100	Reset Data Terminal Ready
001	Set Request to Send (RTS)
101	Reset Request to Send
002	Enable Receive
102	Disable Receive
003	Read Status Byte
103	Read MODEM Status Byte
004	Read Data Byte
005	Transmit Byte
105	Transmit Break
205	Force Transmit Interrupt
006	Transmit Secondary Channel Mark
106	Transmit Secondary Channel Space
007	Master Reset

Interruptible Communications Controller Instructions.

As mentioned at the beginning of this discussion, the C5 processor and its related I/O controller are used in the Model 1536 Intelligent Communications Adapter. In this case, they also operate a communications board called the Model 1539 Interruptible Communications Controller. Instructions used are in the 176-ONN series, and are the same as those described earlier in this section for the C4 I/O controller.

3-1. GENERAL.

The purpose of this section is to describe the fundamentals of programming the Series 1500 Intelligent Terminals and the associated peripheral units. It deals with the methods that are established by such factors as hardware capabilities, the characteristics of the assembler, or the availability of standard routines. For example, the subject of programming a display is covered in this section. The techniques used in this case are limited by the following: (1) the Series 1500 display character code, (2) the memory area available for displays, (3) the machine instructions available, and (4) the requirements of the Assembler System.

Readers familiar with programming will have recognized that the instruction repertoire and formats described in Section 2 were given in machine language, which is not used directly by the programmer. Two programming languages are available for use with the Series 1500 System. They are: (1) the CDE/BTL System, which consists of approximately 70 macro-instructions, and (2) the Assembler System Language, which includes instructions and is much closer to machine language than the CDE/BTL. (CDE means Complex Data Entry, and BTL is the abbreviation for Business Transaction Language.)

The CDE/BTL programming system is covered in two publications, CDE in 91002 and BTL in 91001. When a program written in CDE/BTL is assembled, it is converted into the machine instructions necessary to carry out the stated function. The programmer has no direct contact with the machine instructions and need not be concerned about machine characteristics other than those stated in the descriptions of the CDE/BTL instructions; therefore, the CDE/BTL system is not discussed further in this manual.

The Assembler System, however, does require that the programmer have a good working knowledge of the machine instructions and equipment characteristics. This information is provided in the remaining paragraphs of this section. Note that the operating instructions for the Assembler System are provided in publication 91021 and, of course, the programmer will require that manual when preparing programs in the assembler language.

3-2. RELATIONSHIP OF MACHINE INSTRUCTIONS TO THE ASSEMBLER SYSTEM.

The purpose of the Assembler System is to accept programmer's statements and convert them into a program in machine language. Programmer statements are called the "source", and the final program is called the "object".

The best way to illustrate the generation of an object instruction from a source statement is to use the ADA - Add to Accumulator instruction as an example. The machine language format and instruction description of this instruction follows in figure 3-1.

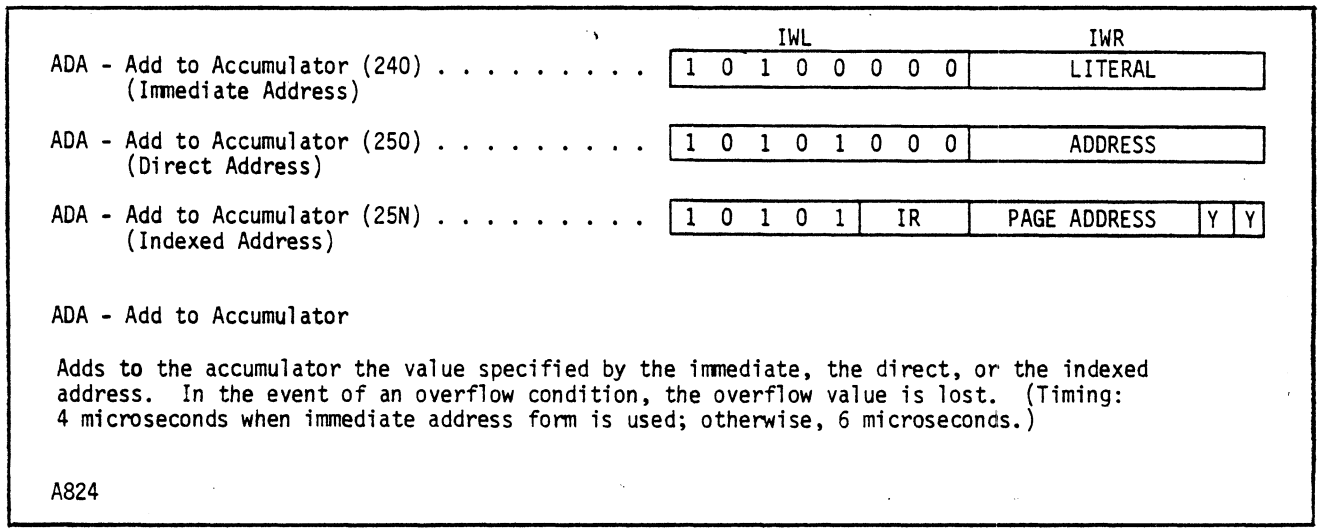


Figure 3-1. Format and Description of the ADA Instruction in Machine Language

As shown in the figure above, this instruction may have one of three forms, as follows:

a. The immediate address form in which the IWR, called the literal in this case, is added to the contents of the accumulator. (The literal may be any quantity.)

b. The direct address form in which the IWR is the address of a location in memory. Located in that address is a value to be read from memory and added to the contents of the accumulator.

c. The indexed address form. In this case, the IWR specifies a page number in which a data byte to be added to the accumulator is located. The three low-order bits of the IWL then specify an index register that holds a byte address. When the contents of the index register and the page number from the IWR are combined, they select a memory location from which data is read and added to the contents of the accumulator. The two low-order bits of the IWR determine whether the contents of the index register are to be incremented, decremented, or left unchanged after the instruction is executed.

Obviously, the source statements must differ for each form of the object instruction. This is shown below.

Form	Source Statement	Object (Machine Instruction)
Immediate	ADA,R#0;Literal	240-LLL
Direct	ADA,R#0;LAB±NNN	250-YXX*
Indexed	ADA,R#X;PPP	25I-YXZ00
	ADA,I#X;PPP	25I-YXZ10
	ADA,D#X;PPP	25I-YXZ11

*X in the object indicates 3 bits, Y - 2 bits, and Z - 1 bit

In the immediate address form above, the Assembler System converts "ADA" to an operation code of 240. The fact that R#0 (no index register) is specified eliminates the possibility of selecting an index register. In the IWR, the "LLL" is the quantity to be added to the contents of the accumulator. The programmer can enter the literal in any one of several forms, although it must be specified which form it is, and the Assembler System converts the literal to the form required by the processor.

The direct address form is shown next. Again, R#0 indicates that no index register is to be used in obtaining the address of the data byte.

The remaining portion of the direct address source statement is the address. LAB is a label specifying the symbolic address, and the "±NNN" indicates that the address that is to be used is that label plus or minus a 3-digit decimal number between 000 and 255. In response to this source statement, the Assember System converts "ADA" to an object code of 250, calculates the direct address, and places the direct address in the IWR of the object instruction.

The last form of the ADA instruction to be discussed is the indexed address form. In this case, the index register selection can take three forms:

a. R#X selects the index register specified by "X" and leaves the contents unchanged when the instruction is executed. When this is converted to machine language, the Assembler System places the register number in the IWL (the "I" position) and 00 in the two low-order bits of the IWR.

b. I#X also selects the index register specified by "X", but increments the contents by one by placing 10 in the low-order bits of the IWR.

c. D#X selects index register "X" and decrements the contents by one by placing 11 in the low-order bits of the IWR.

The byte address is obtained by reading the contents of the specified index register. This is combined with the page number selected by the six high-order bits of the IWR. The programmer gives the page number in three character positions, indicated by the "PPP" shown in the source statement. Either an absolute page number or symbolic page number can be specified. The absolute number takes the form "PNN", while the symbolic form is simply three characters, such as ABC.

When the Assembler System processes a source statement requiring the indexed address form of the ADA instruction, the object code generated is 251, 252, 253, 254, 255, 256, or 257, with the low-order octal digit specifying the index register involved. It then accepts or calculates the page number involved and places it in the IWR of the object instruction. The R, I, or D becomes a 2-bit code in the IWR.

After the Assembler System has processed the source statements, it can provide a printout of the results. Examples of source statements and object instructions as they would appear in a printout are shown below in figure 3-2.

```
PPP-LLL:  OPC-OPR.          E  SEQ. NO.  LAB:  CMD    OPERANDS          COMMENTS

P34-230:  211-004.             02-110.   CUR:  LDA,  R#1; DSP.
P34-232:  360-100.             02-120.         IRA,  R#0; OCT: 100.
P34-234:  231-004.             02-130.         STA,  R#1; DSP.
P34-236:  140-000.             02-140.         EXU;  000.
P34-240:  340-003.             02-150.   NUM:  CPA,  R#0; (0).
P34-242:  114-253.             02-160.         BRL;  RNG.
P34-244:  340-014.             02-170.         CPA,  R#0; (9).
P34-246:  114-252.             02-180.         BRH;  RNG.
P34-250:  104-264.             02-190.         BRU;  EX2.
P34-252:  125-004.             02-200.   RNG:  SBU;  ERR.
P34-254:  173-007.             02-210.         IOC,  C#3; 007.
P34-256:  240-100.             02-220.         ADA,  R#0; OCT: 100.
P34-260:  124-356.             02-230.         SBU;  TRN.
P34-262:  104-240.             02-240.         BRU;  NUM.
P34-264:  140-000.             02-250.   EX2:  EXU;  000.
A882
```

Figure 3-2. Examples of Instructions in an Assembler System Source Listing Printout

The left column in the sample printout gives the memory location in which the object instruction is stored. Note that the instructions occupy both an even and an odd address, but only the even address is listed.

The next column, entitled OPC-OPR (operation code-operand), lists the machine language for the instructions (or octal values of some constants). Then, the sequence number of this instruction is shown. The sequence number identifies each line, so that information can be added or deleted easily.

A column entitled "LAB" appears next. This lists any symbolic labels that a programmer may have assigned to a routine. Following the label column are the source statement columns "CMD", "OPERANDS", and "COMMENTS", which show the source statement originally entered by the programmer.

3-3. ORGANIZATION OF STORAGE.

The way in which the storage media are organized has a very significant effect on programming techniques; therefore, the three most frequently used media (memory, cartridge tapes, and discs) are described in the following paragraphs. Capacity and organization of the maxi-tapes are described later in this section.

Memory Pages.

A "page" in memory consists of 256 locations, each capable of holding one byte. This was discussed in the description of the processor architecture. The first 16,384 memory locations are organized into eight sections, numbered 0 through 7 as shown earlier in Section 2 of this manual. Then, within each section, the memory is divided into eight levels, also numbered 0 through 7. This results in each section holding eight pages.

The page number is composed of the section number and level number, and ranges from 00 through 77 octal. Within each page, the memory locations are numbered 000 through 377 octal.

When a terminal has a memory larger than 16K, the selection of the second and succeeding 8K blocks requires the use of the SAM-Select Auxiliary Memory instruction. The first 8K block (pages 00-37) is automatically selected when the high-order address bit (AD13) is a zero, which is the normal way in which address selection takes place when only a 16K memory is used. Address positions AD8 through AD12 then choose the page number.

In a terminal with only a 16K memory, the switch of AD13 from a zero to a one would select the next 8K block and allow pages 40 through 77 to be selected by AD8 through AD12. In a larger memory, however, the switch of AD 13 from a zero to a one has the effect of switching to an 8K block selected by the SAM instructions. A SAM instruction must be executed to select the next 8K block, whether the second, third, fourth, fifth, sixth, seventh, or eighth. For example, the SAM instruction with an IWL

of 145 selects the sixth 8K block to be addressed. Then the address positions AD8 through AD12 choose a page within the selected block, and the pages are effectively numbered 40 through 77 just as if they were the second 8K block in a 16K memory.

To summarize the selection scheme:

Selection Instruction	Block Selection
None, AD13 is a zero	1st (pages 00-37)
SAM is 141	2nd (pages 40-77)
SAM is 142	3rd (pages 40-77)
SAM is 143	4th (pages 40-77)
SAM is 144	5th (pages 40-77)
SAM is 145	6th (pages 40-77)
SAM is 146	7th (pages 40-77)
SAM is 147	8th (pages 40-77)

Some memory locations are reserved for special uses, as follows:

a. Locations 001 through 007 in the first page of every section (pages 00, 10, 20, 30, etc.) are used as index registers.

b. Locations 040 through 77 (octal) in page 00 are the instruction address stack.

c. Page 02, locations 000 through 003 are linkage locations for the loading program and the system reset switch.

d. P03, locations 000 through 003 are interrupt linkage locations.

Other memory pages are also pre-assigned for specific functions. These are discussed when the associated function is described, and they are shown in the standard memory map in the Appendix of this manual.

A memory page of 256 bytes is one of the basic data units in the Series 1500 terminals and is closely related to the other devices. For example:

a. The display on the small screen is 8 lines by 32 characters, for a total of 256 characters. Each character is represented by one byte in a page of memory. Only certain pages can be displayed, as described later in this section.

b. A standard record on the cartridge tape is 128 bytes, one-half a memory page.

c. A commonly used record on the maxi-tapes is 256 bytes, one memory page.

d. A sector on the disc is 256 bytes; again, one memory page.

Format of a Standard Cartridge Tape Record.

Most of the programs used by the Series 1500 Intelligent Terminals use a standard format for a record on the cartridge tapes. This record, which is shown in figure 3-3, consists of the following:

a. A three-byte preamble and a three-byte postamble. These bytes are used only by the tape handling circuits and are of no concern to the programmer other than the fact that the program must supply them when writing a record. This is handled automatically by the standard background routine discussed later in this manual.

b. An eight-byte label at the beginning of the record. This label contains a sequence number and control information, which is discussed in detail in the Assembler System Reference Manual.

c. 128 data bytes.

d. A one-byte cyclic redundancy code (CRC), which is a method of checking the record content.

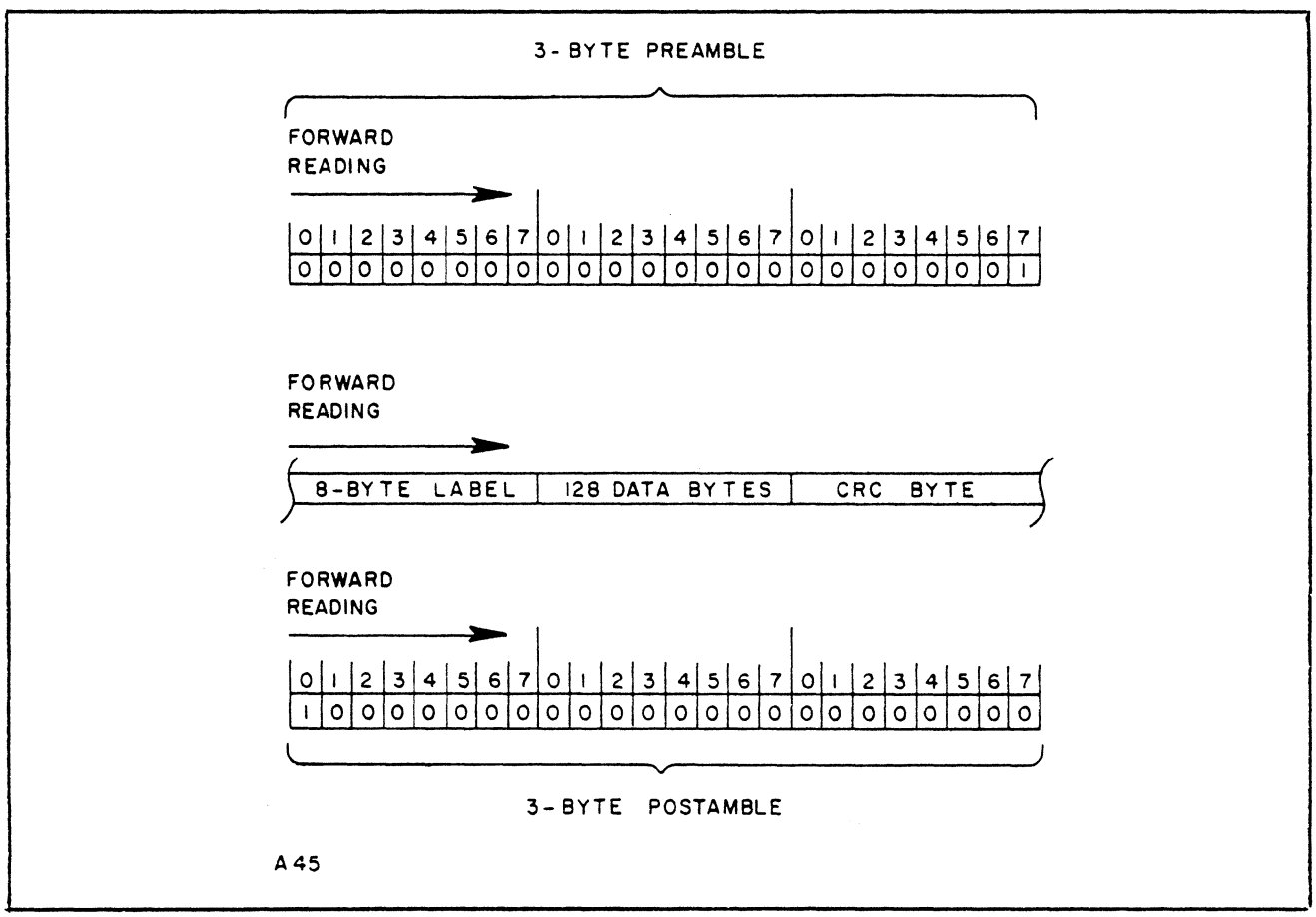


Figure 3-3. Format of a Cartridge Tape Record

Format of a Disc Sector.

Each disc records data in a series of concentric tracks. In the Model 1543, there are 204 tracks on each side of the disc, for a total of 408 tracks per physical disc platter. Since each track is divided into 24 sectors, there are nearly 10,000 sectors available on each disc. (The Model 1540 has only 203 tracks on each side of the disc.)

The larger disc-equipped terminals can operate as many as four Model 1543 disc drives, which each hold two discs. Thus, the maximum disc-storage capacity is 4 disc drives times 2 discs each, or 8 discs. This is approximately 80,000 sectors of storage capacity.

The first step involved in choosing a disc sector is to select the disc drive, which is numbered 0 through 3. Then, the disc itself is selected by a 0 for the disc pack or a 1 for the fixed disc. Next, the track is selected, and finally, the individual sector is located.

Most of the organization of disc storage is controlled by a program called the Data Base Manager, so the programmer need not be concerned about the details of data handling. (Programming of the disc units is also discussed later in this section.) However, the format of a disc sector may be of interest to some readers; it is shown in figure 3-4.

Note that each sector can be identified by a logical address, regardless of its physical position. The 2-byte address occupies the first position of the sector. The accompanying information allows circuit synchronization and error checking. The data is also labeled by a 12-byte header and is accompanied by synchronization and error checking information.

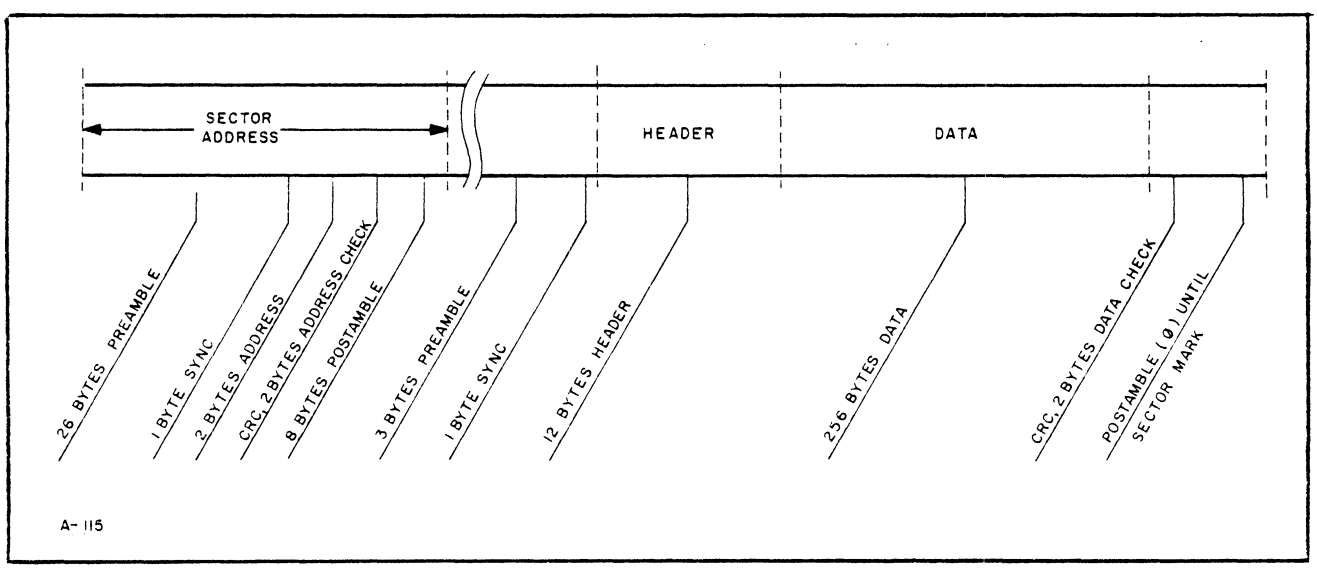


Figure 3-4. Disc Sector Format

3-4. SERIES 1500 CODE.

The description of processor architecture given in Section 2 of this manual clearly illustrated that the processor is designed to handle data in unsigned, eight-bit bytes. Through the use of complex routines, however, large numbers required for data processing operations such as payroll processing can be performed. The purpose of this description is to introduce the fundamental operations involved.

Each data byte is arranged to be entered and read in a tri-octal grouping, as shown in figure 3-5. The bit positions are numbered for the power of two that they represent, and the respective decimal and octal equivalents are also shown.

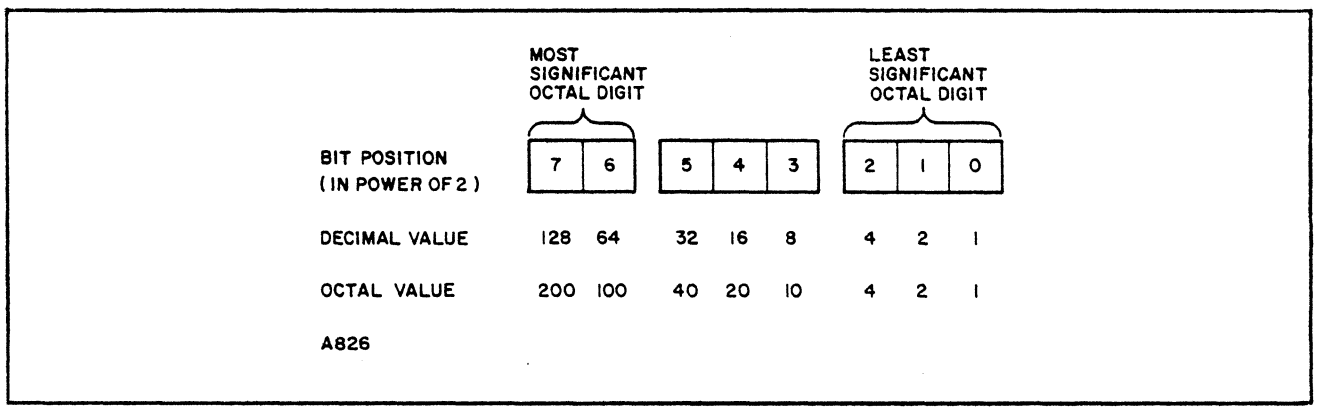


Figure 3-5. Tri-Octal Grouping of a Data Byte

The code used to represent numbers and characters in the processor is the Series 1500 code. Shown below are the Series 1500 codes for numbers and signs:

Character	Series 1500 Code	Binary Value
-	001	00 000 001
+	002	00 000 010
0	003	00 000 011
1	004	00 000 100
2	005	00 000 101
3	006	00 000 110
4	007	00 000 111
5	010	00 001 000
6	011	00 001 001
7	012	00 001 010
8	013	00 001 011
9	014	00 001 100

The Series 1500 code is also called an "excess 3" code because it is equal to the decimal number it represents plus three. When represented in the octal notation, it appears as shown above. For decimal digits 5 through 9, the relationship of the digit to the Series 1500 code requires illustration. This is given in the following two examples:

Decimal Digits	Add 3	Convert to Octal
5	8	10
9	12	14

For simple data entry operations, the Series 1500 code produces the respective decimal digit on the display screen. So the only programming operation required is to move the code into the memory location related to the desired display position. Of course, the keyboard input must first be translated to Series 1500 code.

Comparisons of absolute values are also simple. The value in the accumulator is compared against another value and the results set the condition register as follows:

Equal: Accumulator value equals unknown value.

High: Accumulator value is greater than unknown value.

Low: Accumulator value is less than unknown value.

Adding or subtracting numbers in the Series 1500 code does require correction of the results, however. Of course, the addition or subtraction is performed in binary, but the result is read in Series 1500 code.

Examination of the following operation will illustrate the problem: Assume that a decimal 2 and a decimal 3 are added when they are represented in the Series 1500 code. This produces the following results:

Decimal Digits	Series 1500 Code
2	005
+ 3	006
5	013

Note that the result is a 013 in the Series 1500 code, which according to the code listed above is equal to a decimal 8, while the answer should be 010 to represent a decimal 5. Obviously, 003 must be subtracted from the result in order to arrive at the Series 1500 code (010) representing the correct decimal digit (5). This forms the basis for the correction method, which is as follows: After adding N number of digits, subtract 003, N-1 times.

In the example above, the number of digits added was two, the 2 and the 3. Thus, N-1 is 1, or one time that the excess 3 must be subtracted to acquire the proper result.

In subtraction, the opposite effect takes place so a correction factor must be added to the result to arrive at the correct answer. In this case, after subtracting N number of digits, add 003, N-1 times.

The Assembler System automatically makes the correction, so the programmer is not burdened by the conversion. However, the reader should be aware that the correction is necessary and is being made.

3-5. BACKGROUND ROUTINES AND TABLES.

Available on a tape cartridge called the Standard I/O Library 1 are all the routines listed in table 3-1. Each is called a program segment and is identified by a three-character code. It should be noted that some of these segments are further subdivided, as discussed later in this section.

Most of these segments are programs intended to handle the transfer of information between the I/O devices and the processor. An example of such a segment is the MXP, Matrix Print Function. This segment handles the printing of data on the Model 1525 Printer, and relieves the programmer of the task of preparing a program to perform this function.

Other segments in the standard I/O library provide tables. An example of this is the STT, Standard Keyboard Translation Table, which is used to translate keyboard codes to Series 1500 codes for the programmer.

The segments selected and loaded into memory for each object program are called the background. Certain segments must always be used in order for the terminal to operate; hence, they are called the standard background. There are six segments in the standard background, as follows:

a. BTS - Bootstrap and Loader, which is activated by the PROGRAM LOAD switch and is required to load a program from the tape cartridge in drive 2.

b. SDP - Standard Dot Pattern, which is required to generate the characters on the display screen.

c. STT - Standard Keyboard Translation Table, which translates the code produced by each keystroke into the Series 1500 code.

d. MRW - Mini-Tape Read/Write, which handles the transfer of data to and from the cartridge tapes.

e. PSW and PSR, Page Select Write and Read, respectively, which allow the programmer to designate the memory page numbers involved in the transfer of data to and from the mini-tapes.

The memory locations in which these standard backgrounds are placed are shown in the standard memory map in the Appendix. Note that they occupy about half of memory section 0, although the bootstrap and loader is used only once and most of the positions that the BTS segment occupies while loading the object program can be used for storage of data by the object program.

Also listed in table 3-1 are the optional program segments, which can be added to an object program to perform specific functions.

Table 3-1. Program Segments in Standard I/O Library 1

Segment	Function	Memory Locations (in Decimal)
BTS - Bootstrap and Loader	Load program tape	P00-024 thru P00-255 (Cleared after loading is completed)
SDP - Standard Dot Pattern	Dot patterns for standard display character set	P04-000 thru P05-063
STT - Standard Keyboard Translation Table	Translate keyboard code to Series 1500 code.	P05-064 thru P05-121 P05-128 thru P05-185
MRW Mini-Tape Read/Write Function	Read and write on cartridge tapes	P06-000 thru P07-183
PSW Page Select - Write Function	Select memory area to be written onto tape	P07-184 thru P07-233
PSR Page Select - Read Function	Select memory area in which data read from tape is to be stored	P07-210 thru P07-241
IOC - Mini-Tape I/O Control	Mini-tape and keyboard I/O control	P03-006 thru P03-239

Table 3-1. Program Segments in Standard I/O Library 1 (Continued)

Segment	Function	Memory Locations (in Decimal)
M1X - Standard 9-Track Maxi-Tape (800 BPI)	Read or write up to 256 bytes on 9-track (800 BPI) maxi-tape	P15-000 thru P15-235
M7X - Standard 7-Track Maxi-Tape (556 BPI)	Read or write up to 256 bytes on 7-track (556 BPI) maxi-tape	P15-000 thru P15-235
M6X - Standard 9-Track Maxi-Tape (1600 BPI)	Read or write up to 256 bytes on 9-track (1600 BPI) maxi-tape	P15-000 thru P15-237
M1E - Extended Maxi-Tape Function, Medium Density	Read or write up to 1000 bytes on 7- or 9-track, medium density, maxi-tape	P14-216 thru P15-255
M6E - Extended Maxi-Tape Function, High Density	Read or write up to 1000 bytes on 9-track, high density maxi-tape	P14-216 thru P14-255
MXP - Matrix Print Function	*Printing and form control for 1525/ 1526 printers	P14-000 thru P14-103
LNP - Line Print Function	Printing and form control for 1551-1556 printers	P14-000 thru P14-081
C25 - 1525/1526 Print Option	Special form control option for 1525/1526 printers	P14-102 thru P14-197
C51 - 1551 Print Option	Special form control option for 1551 printers	P14-080 thru P14-197
C55 - 1555/1556 Print Option	Special form control option for 1555/1556 printers	P14-080 thru P14-197

*Requires Standard Dot Pattern SDP

Table 3-1. Program Segments in Standard I/O Library 1 (Continued)

Segment	Function	Memory Locations (in Decimal)
TYP - Typewriter Print Function	Printing and form control for 735 typewriter	P14-086 thru P14-197
P1S - Typewriter Single Character Print	Single character printing and form control for 735 typewriter	P14-000 thru P14-093
C35 - Typewriter Print Option	Special form control option for 735 typewriter	P14-000 thru P14-087
CT1 - Printer Translation Table (1500-to-USASCII)	Translates Series 1500 code to USASCII for printing	P05-192 thru P05-255
TT1 - Printer Translation Table (1500-to-typewriter)	Translates Series 1500 code to 735 typewriter code	P05-192 thru P05-255

Selection and use of the backgrounds is accomplished as follows: After a programmer has prepared a program and had it assembled by the Assembler System, the assembler must perform the object generation phase. In order to perform this function, the Assembler System requires the programmer to load a tape cartridge containing the background required by the program. It then generates an object program in which the selected background is placed in the respective positions shown in the memory map and the object program is integrated with the background. The result is a complete, self-loading program.

In order to prepare the background tape called for by the Assembler System, the programmer loads the standard I/O library tape and follows the instructions displayed. One item that must be selected from the tape in a standard background segment is B1A, B1B, B1C, or B1D, depending upon the memory and program size. Included in each of these segments are the standard background functions (BTS, SDP, MRW, STT, PSW, and PSR) and the page number of available pages. Shown below is the background segment and related memory requirements.

Segment	Memory Size	Memory Locations
B1A	4K	P01-000 thru P15-255
B1B	8K	P01-000 thru P31-255
B1C	16K	P01-000 thru P63-255
B1D	2K	P01-000 thru P07-255

Once the programmer chooses all the optional segments, they are transferred to his background tape. When completed, this is the tape supplied to the Assembler System.

Bootstrap and Loader Routine (BTS).

The bootstrap and loader routine, whose segment identification is BTS, is used to load all programs from the tape cartridge in drive 2. During the last stages of program assembly, the BTS segment is placed in the first two records of the object program. These two records serve to load the remainder of the program.

Assume that a tape cartridge containing an application program is placed in tape drive 2 to be loaded and operated. The operator pushes the PROGRAM LOAD switch to begin operation. This causes a ROM in the processor to perform an 8-instruction program to read the first record on the tape, which is the bootstrap routine record. After loading this record into locations P00-030 through P00-227 (octal), the ROM program branches to the first instruction of the bootstrap routine, now in P00-030.

The bootstrap routine is then executed. It reads the second record from the tape, which is the loader routine, and causes the loader routine to operate. In turn, the loader routine reads in the application program, which also includes the background required. When the loading operation is completed, the loader branches to location P02-000. Placed in this location by the Assembler System is an instruction linking the loader to the application program.

The bootstrap and loader routine performs three error checks during the load operation, as follows:

a. Sequence number checking - This operation checks the first byte in the label field of each record read from tape for proper sequence. If a sequence error occurs, the load operation is terminated and a pulsed error tone is initiated.

b. Tape Error - If a tape error occurs, the loader will retry eight times to read the record. If unsuccessful, the CRT will go blank, but no error tone will be generated.

c. Each record loaded into memory is read back and a CRC check is performed. This checks the memory and the data being stored. If this error occurs, a constant error tone is generated.

Standard Dot Pattern (SDP).

This segment is not a programming routine but rather a series of codes necessary to generate characters on the small display screen. These codes are always placed in the same locations, page 04 and part of page 05, because the display circuits automatically address these locations when generating the display. Refer to the standard memory map in the Appendix to determine the codes used and the locations in which they are stored.

Standard Keyboard Translation Table (STT).

This segment is also a series of codes rather than a program routine, and the location of the table is shown in the standard memory map. The translation table is used in the following manner: When the keyboard character is read by the processor, it is placed in the accumulator. The objective then is to acquire the Series 1500 code for that character. The programmer does this by adding an octal 100 or 200 to the keyboard character code; 100 is used for letters and 200 for numbers. This can be pre-established or controlled by the keyboard "shift" keys. For example, the code for a "J" is 045; this becomes 145 when added to the octal 100. The "145" is then used as an address to read the contents of location 145 in the translation table. From that address, the processor reads 026. (This is shown in the standard memory map.) The code 026 is the Series 1500 code for a "J", thus the translation has been accomplished. Assuming that the keyboard character is to be displayed, the programmer simply moves the 026 to the appropriate memory location and the "J" appears on the screen.

Standard Mini-Tape Read and Write Function (MRW).

This program segment is identified by the characters MRW, and is part of all four standard backgrounds. In turn, this segment is further subdivided into four segments:

a. RDF, Read Function.

b. WRF, Write Function.

c. BSF, Backspace Function.

d. D84, Timed Backspace Function.

The purpose of the RDF and WRF segments is to relieve the programmer of the task of preparing read and write routines. The programmer is required only to establish the data buffer area, link to the RDF or WRF segment, examine the status word, and if an error does occur, decide what action should be taken. The RDF and WRF routines will read or write the tape, make validity checks, make a sequence number check, and execute a sequence of up to eight retries if a read error is detected. The WRF routine also generates a label with a sequence number.

The record format is the standard format described earlier. The label is always read from or written into a fixed area of memory, P00-030 through 037 (octal). The data is read from or written into a pre-selected area, P01-000 through 127 (decimal), but this area can be changed by the programmer through the use of the page select routines (PRS and PSW).

Linkage to the read or write routine is made by placing a drive selection number in the accumulator and then executing a Stack and Branch instruction to RDF or WRF. Status is passed back from the read or write function to the condition register and index register 4. The conditions checked for on each read and write function are sequence errors, tape errors, and CRC errors.

The write function makes an additional check for the write enable pin being in place. A tape runaway error is generated if a tape write command is executed and the write enable pin is not in place on the bottom of the cartridge.

A segment called BSF, Backspace Function, is also available. It backspaces the tape by one record. After the drive is selected, the BSF routine is executed and the sequence number is decremented by one. The record backspaced is located approximately 1/2-inch before the read head and is thus in the position to be read.

Prior to executing the BSF routine, a drive selection number is placed in the accumulator. Then, the backspace function is entered by Stack and Branch instruction to BSF.

A timed backspace function is used only for the read error recovery procedure. It positions the tape for a read retry. It acts upon the current drive selected, and linkage is achieved by a Stack and Branch instruction to segment D84.

Page Select Read and Write Functions.

The standard mini-tape read/write function routines are set to read in or write from P01-000. If the programmer wants to use another area of memory, a page select routine is available. A Stack and Branch instruction is used to link to the page select routine. Linkage via the SBU instruction is to PSR for a read operation and PSW for a write operation. However, before the SBU is executed, a page select word must be placed in the accumulator. The format for the page select word is:

LL SSS PPP

Where: LL equals the memory location and is 0 through 3, representing 000, 100, 200, 300 (octal).

SSS is the section number (0 through 7).

PPP is the level number (0 through 7).

There are also variable starting and size entry routines available, designated SSR and SSW. The purpose of these routines is to provide the user with a method of selecting any starting location and data size desired. For this routine, the page select word is stored in three different registers. The location is stored in the accumulator, the level and section (page) numbers are stored in index register 1, and the size is stored in index register 2. Although the size is selectable, it should be noted that each physical tape record remains 128 data bytes.

These variable size routines are part of the standard page select read/write functions. Linkage to these routines is by the Stack and Branch instruction to the variable starting size read routine, SSR, and variable starting size write routine, SSW.

Mini-Tape I/O Control (IOC).

This segment is made up of seven routines, which are as follows:

a. Read (RDS) and write (WTS).

b. Write End of Text (EOT).

c. Compressed Write (CPW).

d. High-speed search (HSF).

e. Load program (LPD).

f. Keyboard entry (KEY or ALP).

While the IOC segment must be read from the Standard I/O Library in its entirety, each routine can be called individually once the IOC segment is loaded into memory. Some of these functions are similar to those in the standard background segment; however, the functions in the I/O control segment are more inclusive. For example, the read/write functions have incorporated in them an error notification and recovery routine, thereby relieving the programmer of the task of error recovery.

Read and Write Routines

These functions enable the user to select the desired drive and perform the read or write operation. They provide a standard method of error detection, notification, and recovery.

The standard tape format is used for these functions; that is, an 8-byte label and 128 bytes of data. Label manipulation is controlled by these routines; the programmer is required only to supply or accept the data from the buffer, which is locations P01-000 through P01-127 (decimal).

Prior to linking to the read or write routine, a drive selection word must be loaded into the accumulator. A Stack and Branch instruction to the RDS (read) or WRS (write) then activates the routine. At the end of the read operation, the programmer should check for an end-of-file status by executing a Branch on Low instruction to test the condition register. At the end of a write operation, the same instruction will detect the physical end-of-tape condition.

Write End-of-Text Routine

This routine records six inches of energy on the tape to form a special tape mark used to stop a high-speed search operation by setting the runaway error. The application of this function enables the quick location of the end of file.

The routine is normally executed after the end-of-file record is written. A Stack and Branch to EOT activates the routine; no drive selection is required because the currently selected drive is always used.

Compressed Write Routine

The purpose of the routine is to shorten the inter-record gap from the normal gap size of 1.1 inch to 0.43 inch during writing. This increases tape capacity from a maximum of 500 records, with the normal gap, to a maximum of 900 records with the compressed gap.

There are two ways of using the compressed write feature. First, within the application program, a source statement is included to alter the standard mini-tape write function to enter the compressed write feature upon completion of the write operation. Second, after execution of the normal tape write procedure, a Stack and Branch instruction to CPW is executed.

High-Speed Search Routine

The high-speed search function provides a method of advancing the tape in a forward or a reverse direction at high speed (40 inches per second). The distance to be searched is based on a record count controlled by the program. The programmer must know the record number on the tape where the desired data is located. The data is not read during the search but instead the number of records that pass the read head are counted.

To use the high-speed search function, the programmer has the option of either altering the high-speed search function during object generation, or executing a Stack and Branch instruction to the search feature.

Load Program Tape Option

The load program tape option simulates the hardware bootstrap load procedure. It provides the programmer with the option of selecting and loading one of several programs on a single program tape. This function is performed by reading the bootstrap record, resetting the stack pointer, moving the bootstrap routine to the assigned area, and finally, execution of the bootstrap routine.

The load program tape option can be used under operator control or strictly under program control. A Stack and Branch instruction to LPD is used to enter the routine.

Keyboard Entry

The keyboard entry function provides a standard method of handling keyboard entries and translations. It relieves the programmer of the task of executing I/O instructions, decoding the key code, and translating the key code via the keyboard translate table.

When using the keyboard entry function, any key depressed will automatically be translated to an alphanumeric code. There is also an option that allows for numeric override to cause all keys depressed to be translated into the numeric value. In either case, the translated character code will be held in the accumulator and in index register 1 when the keyboard entry function is complete. If CTRL key is held while the desired key is depressed, 100 is added to the translated character. For example, a START key is translated to a 201, but CTRL/START becomes 301.

Maxi-Tape Functions.

There are five program segments provided to control maxi-tapes, as follows:

a. M7X, which reads or writes up to 256 bytes on the 7-track, 556 BPI maxi-tape. This is the Model 1511 Tape Drive.

b. M1X, which reads or writes up to 256 bytes on the 9-track, 800 BPI maxi-tape. This may be either the Model 1512 or 1513 Tape Drive, depending upon reel size required.

c. M6X, which reads or writes up to 256 bytes on the 9-track, 1600 BPI maxi-tape. This is the Model 1514 Tape Drive.

d. M1E, in which the "E" means extended. This routine reads or writes up to 1024 bytes on 7- or 9-track, medium density (556 or 800 BPI) tapes. This includes the 1511, 1512, and 1513.

e. M6E, which reads or writes up to 1024 bytes on 9-track, high-density tape. This is the Model 1514.

These five segments are all available on the Standard I/O Library but are optional selections. Since they occupy the same memory locations, only one can be placed in memory at any one time unless special arrangements are made to relocate them.

M1X, M6X, and M7X are all basically identical in that they all control read, write, write-file-mark, backspace, and rewind operations. They are different in that they all control different density.

All three segments enable the programmer to write or read variable size records up to 256 bytes (one page). After each read or write operation, status is returned and stored in register 4 and the condition register. Both the read and write functions utilize the 8-retry technique before returning an error condition. In the event of a tape write error, the programmer is responsible for positioning the tape past the bad area.

The extended segments M1E and M6E are similar to the standard maxi-tape functions. However, they extend the capabilities of the variable or fixed records from the standard 256 bytes per record to 1024 bytes per record. Both extended functions have the capability of controlling reads, writes, write-file-mark, backspace, and rewind operations. They vary in that they control different densities.

Both extended functions return status after a read or write operation. The status word is available in register 4, and the condition register will contain the high, low, or equal condition. Both functions utilize the 8-retry technique before returning an error condition. In the event of a write error, the programmer is responsible for positioning the tape past the bad area.

Matrix Printer Function (MXP).

Segment MXP, Matrix Printer, is provided to control the Model 1525/1526 Printer. This printer employs a 5 by 7 dot matrix principle of printing and is capable of printing the full 64-character set. The Model 1525 is capable of printing 132 characters per line, while the 1526 is capable of printing only 80 characters per line.

The MXP function does not require a translation table; it uses the CRT dot pattern area for the associated character code. However, the MXP function does use the print buffer area, which is initiated as page 01 but can be changed.

The 1525 and 1526 have two print mode options. The first is the line mode, which prints either 80 characters per line (1526) or 132 characters per line (1525). In both cases, the carriage returns at the end of the print function. This mode is capable of handling up to 255 characters, with automatic carriage return and line feed for each print function.

The second mode, the increment mode, prints one character at a time. There is no carriage return at the end of the print function; instead, the carriage returns only in response to a carriage return command or when the line capacity is exceeded.

There are several print control options available with the MXP segment. They include left margin, carriage return, line feed, increment mode, and clear. Any of these options can be executed independently of the print function by loading a preassigned value in index register 4 and performing a stack and branch to the command function routine D99.

Upon completion of a print function, a status condition is returned to the condition register. An equal condition indicates a normal completion, and a high indicates an error. The accumulator holds the status word.

Line Printer Function (LNP).

Segment LNP, Line Printer, controls Models 1551 through 1556 printers. Because of the significant difference between printers, however, the operation of this routine with the Model 1551 Printer is discussed separately.

The Model 1551S and 1551D printers employ the 5 by 7 dot matrix principle of printing and are capable of printing the full 64-character set, with a line capacity of 132 characters. The 1551S has a print speed of 60 lines per minute; the 1551D has a print speed of 125 lines per minute. The printer form is controlled by a form control loop.

The print buffer is initiated as page 01. If other than page 01 is desired as the print buffer, the user must alter P14-007 to contain the desired print buffer page. The data must be in the Series 1500 code, as it is automatically translated before it is actually printed. The print translate table CT1 (1500-to-USASCII) is required for this translation. The CT1 function is part of the Standard I/O Library and must be called for the printing operation. CT1 will normally reside in page 05.

There are several printer control options available in this function; they include: bell alarm, feed-one-line-of-paper, vertical tab, form feed, elongate, de-select, select, clear, and delete. These options may be executed independently of the print function by loading an assigned value in register 4 and executing a Stack and Branch instruction to the command function routine D99.

The status word from the print operation is returned to the accumulator; the condition register indicates equal for normal completion and high for an error condition.

The Model 1553, 1554, 1555, and 1556 Printers operate at line speeds of 125, 200, 300, and 400 lines per minute, respectively. All use the USASCII code and print the full 64-character set. Printer form is controlled by a form control loop.

There are several print control operations available in the LNP segment for the 1553-1556 printers, including form feed, line feed, vertical tab, and slew-to-channel. The aforementioned options may be executed independently of the print function by loading a value in register 4 and executing a Stack and Branch instruction to the command function routine.

The print buffer is initiated as page 01. If other than page 01 is desired as a print buffer, the programmer must alter P14-007 to contain the desired print buffer page. The data must be in the Series 1500 code, as it must be translated before it is actually printed.

The printer translation table CT1 (1500-to-USASCII) is required for this translation. This table is part of the Standard I/O Library and must be called for the printing operation. CT1 will normally reside in page 05.

The status word from the print function will be returned to the accumulator; the condition register indicates equal for normal completion and high for an error condition.

Printer Translation Table.

The 1500-to-USASCII Printer Translate Table, segment CT1 in the Standard I/O Library, is used by the LNP segment print function to translate the Series 1500 code into printer-compatible USASCII code. It is used with a 1551S, 1551D, 1553, 1554, 1555, or 1556 Printer. The table is loaded into a fixed area of memory, P05-300 through P05-377 (octal) and is controlled by the respective print function.

Typewriter Print Function (TYP).

This segment controls the IBM 735 Selectric Typewriter. The 735 is capable of printing at a speed of 15 characters per second, in both upper case and lower case.

A print buffer and translation table are required for the print function. The buffer is initiated as page 01; however, it can be changed by altering the contents of P14-007. The data in the print buffer must be in the Series 1500 code, as it will be translated by the Print Translate Table TT1. This table is part of the Standard I/O Library and must be called in along with the print function.

The print control options available with the TYP segment are backspace, red ribbon shift, carriage return, black ribbon shift, tab, and index. These options can be called independently by loading a preassigned value in register 4 and executing a Stack and Branch instruction to the command function.

After completing the print operation, the routine will return status to the condition register and the accumulator. If the condition register is set to high, an error resulted from the print operation; the accumulator holds the status word.

Typewriter Single Character Print Function (P1S).

This segment controls the IBM 735 Selectric typewriter so as to print one character and then return control to the user's program. It uses the Series 1500 code; therefore, translation table TT1 is also required.

The P1S routine differs from the TYP routine mainly in that a buffer area is not required; instead, index register 4 is used to store the code of the character to be printed. P1S will translate the 1500 code to typewriter code and send the character to the typewriter to be printed.

The print control options available with this function are carriage return, index, tab, and backspace. These options can be executed independently of the print function by loading the assigned value into register 4 and performing the Stack and Branch instruction to "PCR".

After completing the print operation, status is returned to the condition register and the accumulator. If the condition register is set to equal, the print operation was completed normally. A low indicates a busy condition (buffer full), and a high indicates that an error condition exists. The accumulator holds the status word.

Printer Translation Table TT1 (1500-to-Typewriter).

This table translates the Series 1500 code into typewriter code. It is used only with an IBM Model 735 Selectric typewriter. Normally, it resides in memory locations P05-300 through P05-377 (octal). However, if a conflict arises, special arrangements can be made to relocate the tables.

3-6. SMALL CRT DISPLAY.

This is the basic display used on most of the Series 1500 Intelligent Terminals. Even terminals equipped with a large screen can operate in a mode in which they create a small screen display. In this case, the display appears in the top center section of the large screen.

Readers interested in creating a small CRT display immediately should proceed to the paragraph entitled "Programming Aids", which appears last in this material. Others should read the background material in the following paragraphs before using the programming aids.

Display Format.

The small CRT display is designed to display 256 characters in 8 rows of 32 characters each. Each character is formed by lighted dots within a 5-space-wide by 7-space-high matrix. There are 9 blank spaces between character rows and 3 blank spaces between character columns. This is shown in figure 3-6.

It should be noted here that the small CRT is a standard 5-inch television tube with the scanning pattern rotated 90 degrees. In this application, the fast scan is vertical and the slow scan is horizontal. In other words, the beam scans a column and them moves to the next column until all columns have been scanned; then, the process is repeated.

The characters are formed by unblanking the beam when a dot is to be formed and blanking it at all other times. The information that is used to control the unblanking originates in dot patterns stored in memory.

First, it should be noted that the dot patterns are always located in the same positions in memory, all of page 04 and the first 77 octal addresses in page 05. This is shown in the standard memory map on drawing 1804, which is located in the Appendix. Note that the 64 different characters that can be displayed are numbered in octal 00 through 77 left to right across the top of the drawing. The character code stored for a specific location on the screen selects a group of five memory addresses, which are shown vertically below each character code. For example, the character code for an "A" is 015. When the 015 is used to address memory, it selects the locations that hold the five dot patterns that make up an "A". Of course, the page was selected automatically by the equipment involved in generating the display.

During the first column of the first character position, the dot pattern stored in the first memory location is read out and shifted serially to the CRT. This causes unblanking of dots in the proper positions to form the first column of a character. Next, the scan proceeds to the first column of the first character in the second row. This process continues until the end of the first vertical scan line is reached. At this point, the beam is moved to the next column to be scanned and the process of reading the contents of memory locations and shifting them serially to the CRT is repeated. Since there are five vertical lines in every character, five memory locations are required to store each dot pattern. Therefore, 320 memory locations are required to store the dot patterns for the 64-character set.

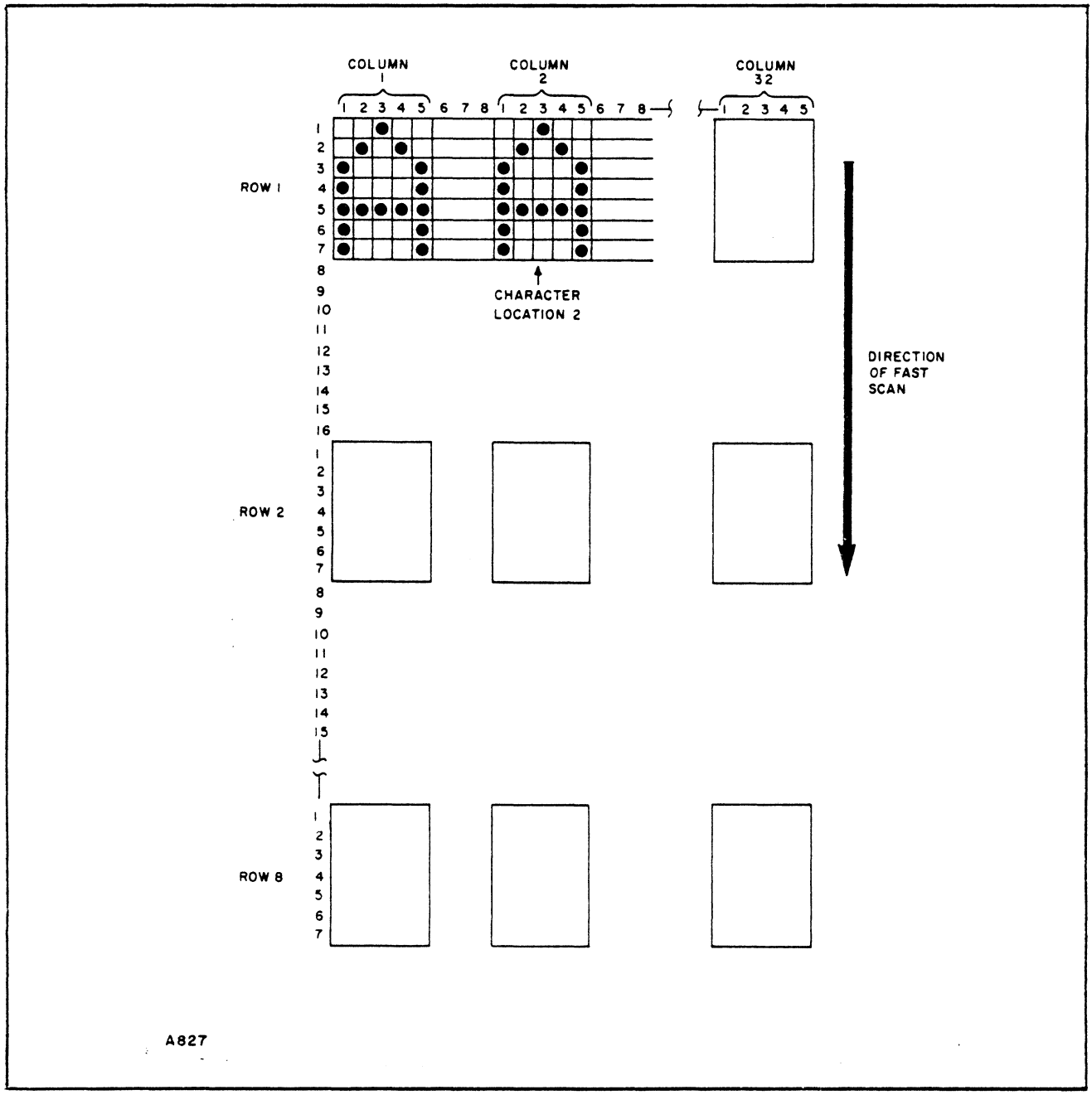


Figure 3-6. Format of Small CRT Display Area

The next point that must be considered in the readout of dot patterns is that the odd vertical lines are scanned during the odd field, and the even vertical lines are scanned during the even field. This requires that the dot patterns for columns 1, 3, and 5 of a character be read during the odd field, and columns 2 and 4 during the even field.

The following is an example of the sequence of events resulting in the display of a character at a specified location on the CRT: Assume the next character location to be displayed on the CRT is location 2 shown in figure 3-6, and also assume that the processor has stored the code 015 for an "A" in location 2 of the page to be displayed. First, the CRT display control circuits address the second location of the selected page. The code for the character A is read from memory and loaded into a buffer. This code is then used to address the dot pattern in memory, and the dot pattern for column 1 of the A is acquired and stored in another buffer. The dot pattern, which is 174 if read in octal, is then shifted serially to the CRT as an unblanking signal that turns the beam on and off. The result is shown in figure 3-7.

Instructions.

The next step in this description is to review the instructions involved in programming the small CRT. There is only one instruction format used to direct operation of the small display. Of course, the page to be displayed must be prepared but this is not done by the small CRT instruction.

The format of the small CRT instruction coded in assembler language is:

```
IOC,C#4;FFF
```

When converted to machine language by the assembler, this becomes 174 in the IWL and FFF in the IWR. "174" selects the small CRT, and "FFF" determines the page displayed and the display mode. This is shown in figure 3-8 below.

Note that bits 1, 5, 6, and 7 select the page in memory containing the codes for the characters to be displayed; the "S" bits selecting the section and the "L" bits selecting the level within the section. Since there are only two "S" bits and only two "L" bits, the pages (in octal) that can be selected are:

a. 00, 01, 02, and 03.

b. 10, 11, 12, and 13.

c. 20, 21, 22, and 23.

d. 30, 31, 32, and 33.

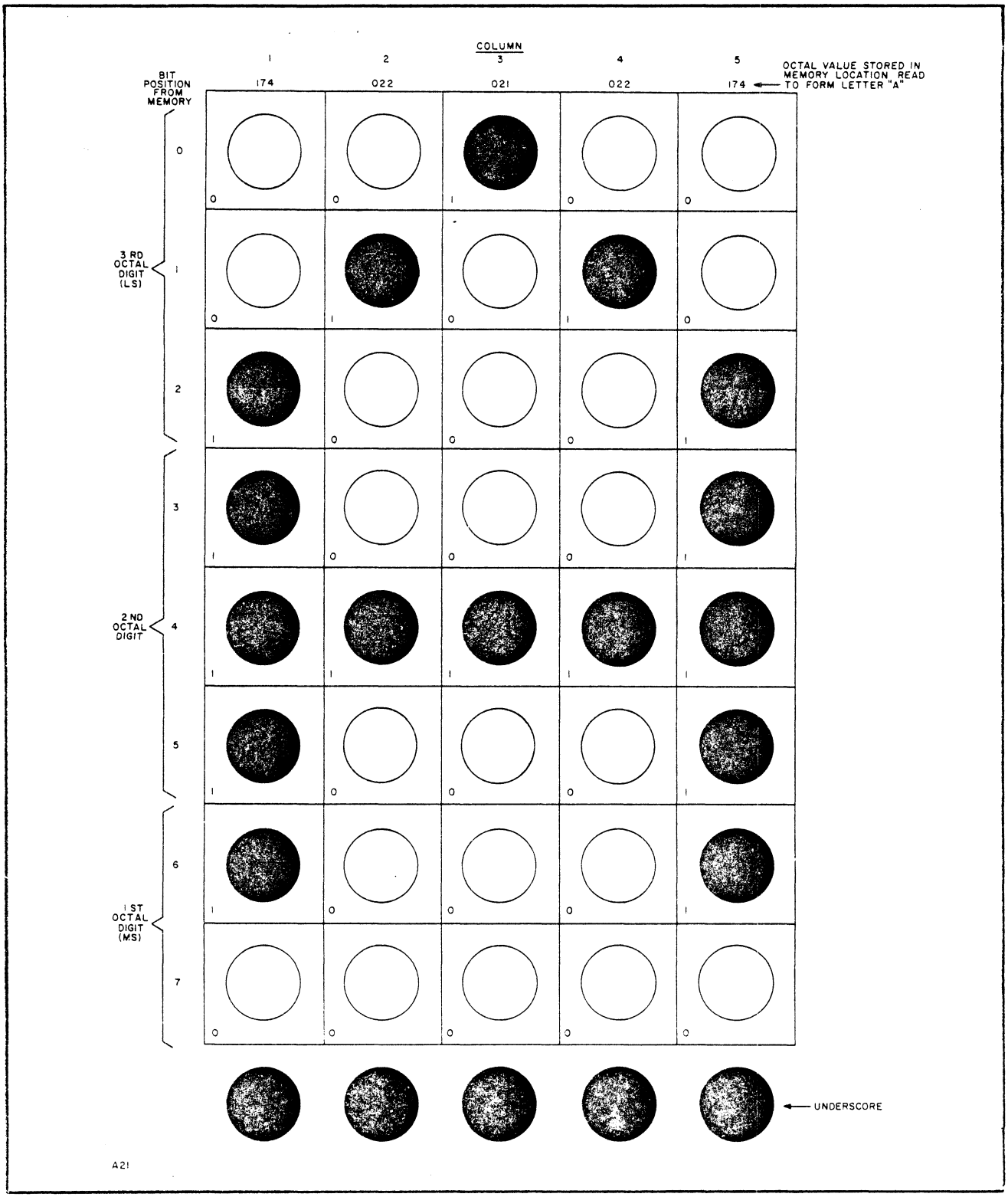


Figure 3-7. Formation of the Character A

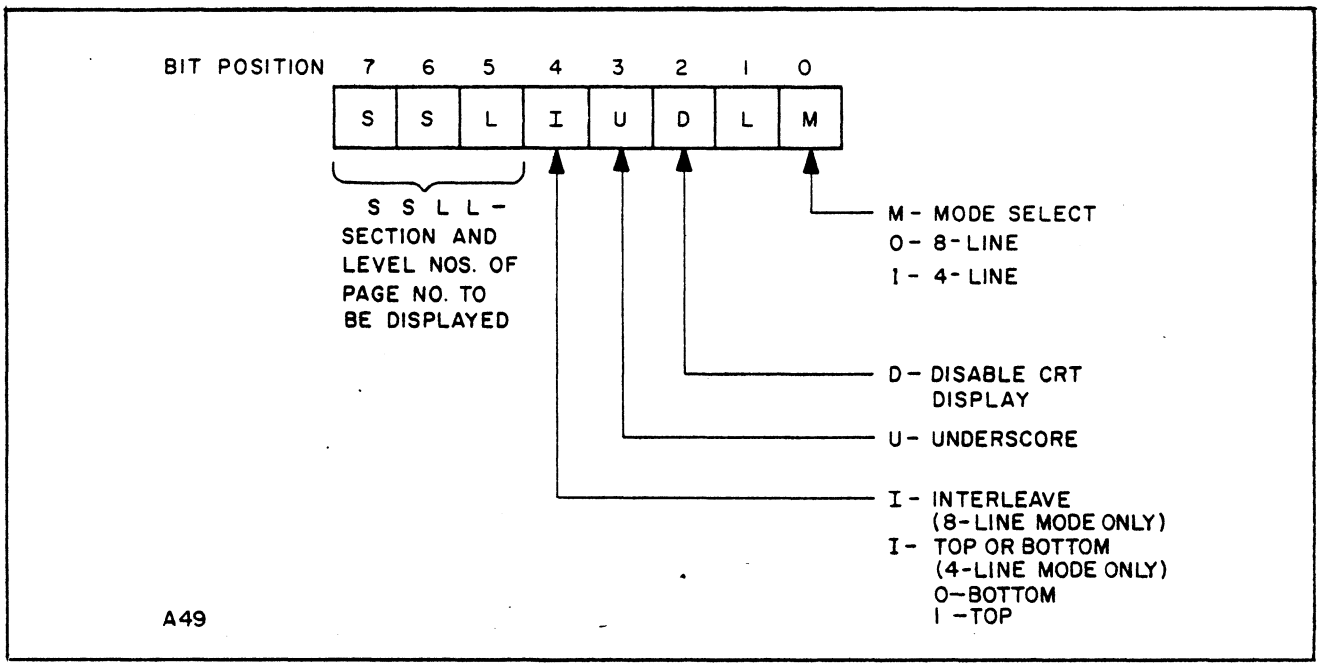


Figure 3-8. Format of IWR for Small CRT Instructions

The relationship of the page selection bits in the IWR (section positions 7 and 6 and level positions 5 and 1) is shown with respect to the overall memory address format below in figure 3-9. The result is that only 16 pages of memory are available for display. Each page selected consists of 256 memory locations, one location for each of the 256 character positions on the screen. Stored in each of these 256 locations is the 6-bit code that chooses the character to be displayed.

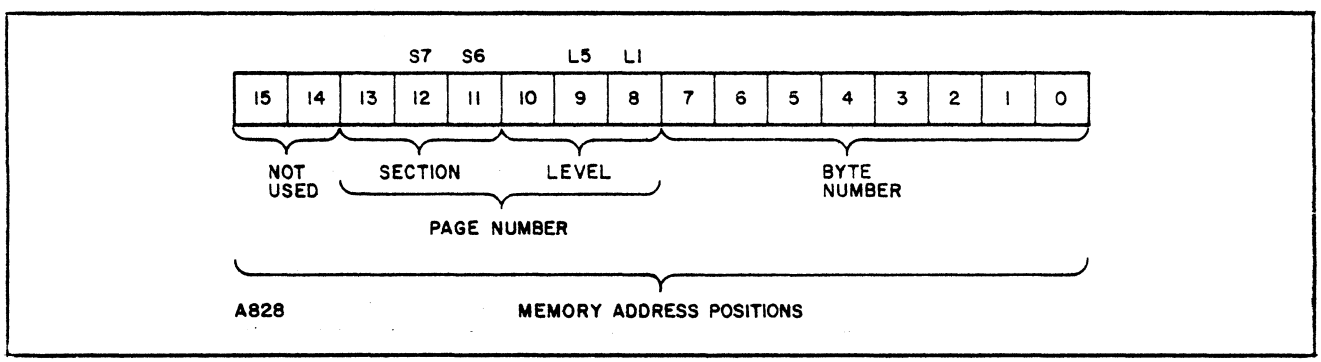


Figure 3-9. Memory Address Format

Bit 0 of the IWR is the mode selection bit. When it is a zero, the full 8-line display is created. When it is a one, it limits the display to four lines, thus saving memory space when only a limited display is required. Bit 2 is the disable CRT command. It prevents any display from appearing. Bit 3 is the underscore bit; its effect is to enable the underscore capability.

Bit 4 is the interleave bit in the 8-line display mode and the half-page, or "zone" bit, in the 4-line display mode. The function of this bit is explained in greater detail below.

There are four basic modes of operation for the small CRT display:

a. Normal mode -- 8 rows in sequence.

b. Interleave mode -- 8 rows interleaved.

c. Four top rows only.

d. Four bottom rows only.

In the normal display mode, a page of 256 characters is displayed in a continuous fashion, from memory locations 000 through 377 (octal) of the selected page, as shown in figure 3-10. In this case, the IWR of the instructions is as follows:

a. Bit 0 (mode) is a zero, indicating the full 8 lines are to be displayed.

b. Bit 4 (interleave) is a zero, indicating no interleaving.

If bit 4, the interleave bit, is a one, the 8-line interleaved mode is selected. In the interleave mode, the page of 256 characters is displayed in the sequence shown in figure 3-11.

If bit 0, the mode selection bit, is a one, only the top or bottom four rows are displayed. They are read from the memory locations shown for the normal mode; however, bit 4 is the zone bit in this case, and it chooses either the top four rows (if bit 4 is a one) or the bottom four rows (if bit 4 is a zero).

Underscore.

Bit 3 of the IWR selects the underscore capability. Assuming that the underscore is enabled, bit 6 of each character code (which is the seventh bit) is examined to determine whether or not that character position should be underscored. If bit 6 of the character code is a one, the underscore is produced. Its position is shown in figure 3-7.

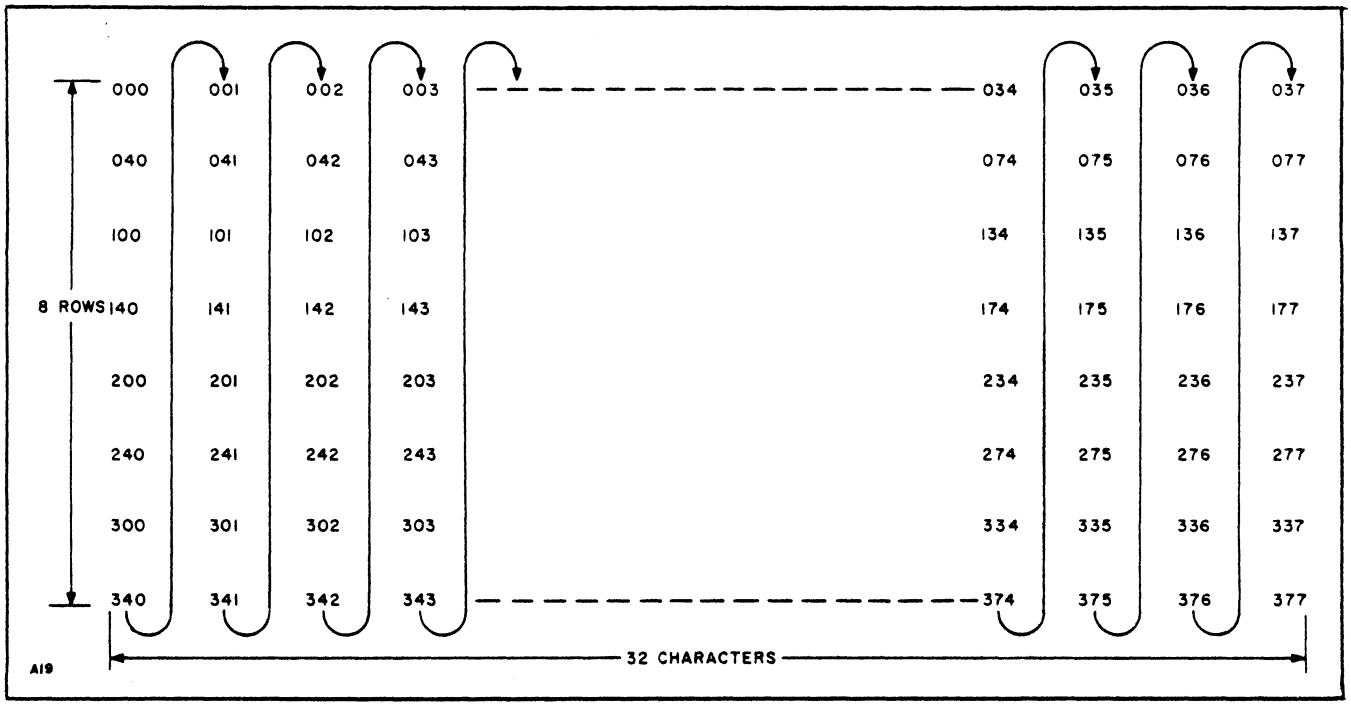


Figure 3-10. Display Sequence in Normal Mode

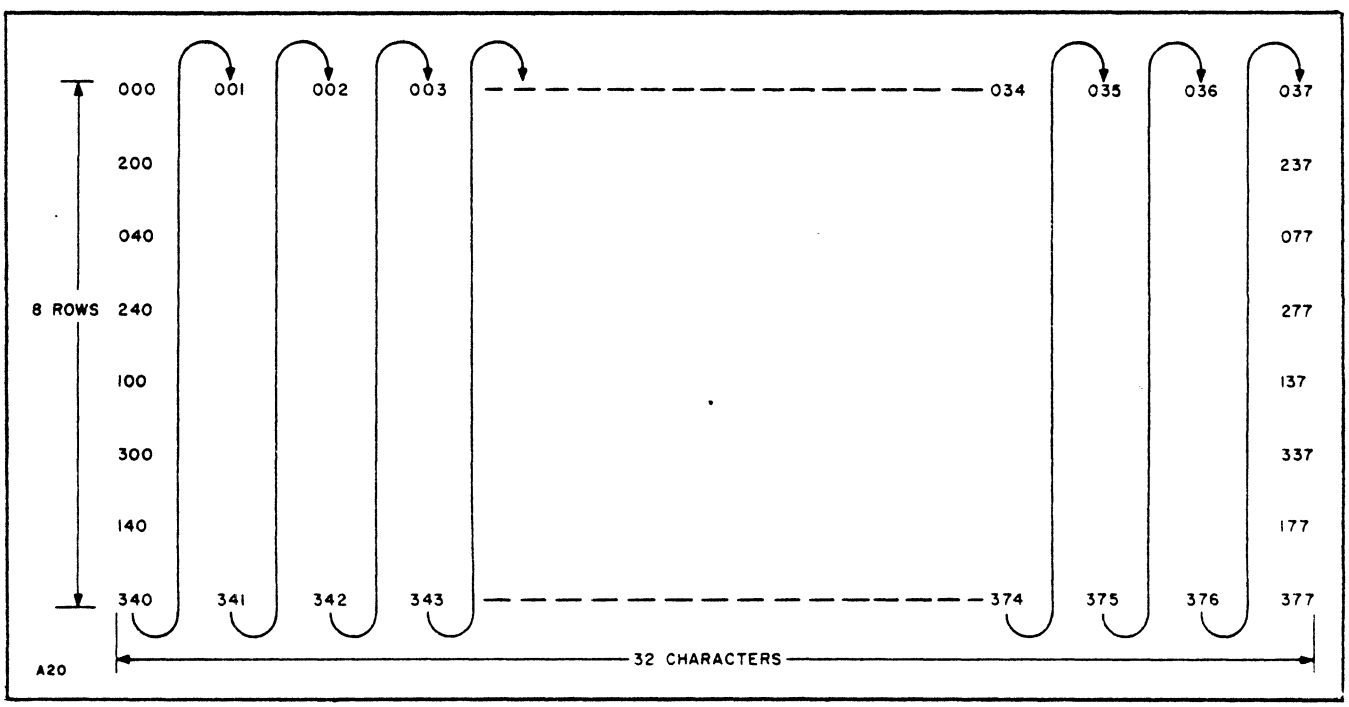


Figure 3-11. Display Sequence in Interleaved Mode

The underscore can be used as a cursor or it can underline several positions in which the operator is to enter data. The method by which the underscore is added to a character position is as follows: If a specific position is to have the underscore, the character code for that position is changed by adding an octal 100. For example, the code for a space is an octal 000; for an underscored space it is 100.

Alternate Character Set.

The dot patterns for an alternate character set can be placed in pages 06 and 07 in positions comparable to those for the standard set in pages 04 and 05. These could be patterns for lower-case characters, for example. The underscore bit of the IWR controls their use, as follows:

a. If the underscore is enabled by a one in bit 3 of the IWR, the alternate character set cannot be used.

b. If the underscore capacity is disabled by a zero in bit 3 of the IWR, bit 6 of each character code determines whether the standard or alternate set is used. If bit 6 for a specific character is a zero, the standard dot pattern is used; if it is a one, adding an octal 100 to the character code, the alternate dot pattern is used for that character.

It should be noted, however, that pages 06 and 07 are normally occupied by the standard background routines for the mini-tape. If in use, these routines must be relocated before an alternate character set is loaded.

Programming Aids.

The following paragraphs describe the basic operations involved in programming a display on the small CRT. Among the subjects covered are clearing a page, selecting a page, and turning the display off while making display changes.

Selection of the CRT for operations is very simple. The programmer's source statement to the assembler system is:

```
IOC,C#4;FFF
```

However, the definition of the "FFF" is complex. It depends upon page number, mode, and options selected.

Refer to figure 3-8, which shows format of the FFF in the IWR. The factor that complicates page number selection is the division of the page number into two parts: bits 5-7 and bit 1. There are two methods by which the programmer can define the FFF code. The first is to simply write out the bit positions grouped in the tri-octal format used in the Series 1500, enter the binary representation for page number and features, and then read the number in octal and place it in the FFF positions of a source statement.

The second method is to refer to table 3-2 and choose the number to be placed in the FFF positions, based on the page to be displayed and the features selected. The following is an example of using table 3-2: Assume that the upper half of page 21 (addresses 000 through 177) is to be displayed to produce a four-line display occupying the top four lines on the screen, and the underscore capability is to be enabled. Page 21 is the tenth line down in the page number column. Follow line 10 across. The first three FFF numbers encountered are 202, 223, and 203, but they all apply to the non-underscore mode. The next three FFF numbers are 212, 233, and 213; these all apply to the underscore mode, so the choice to be made is page size. Code 233 chooses the four-line mode, top half of page, so 233 is the code to be chosen to be placed in the FFF positions of the source statement. In this case, the statement would be: IOC,C#4;233.

One factor not mentioned up to now is the "disable CRT" bit in the FFF code. Note that this bit is a zero to allow the display and a one to disable it. The codes shown in table 3-2 all include a zero to enable the display. If the display was to be blanked out, a 004 would be placed in the FFF positions or added to the current selections. For example, assume that the source statement IOC,C#4;233 was given to display the upper half of page 21 as illustrated above. To blank out this display while changes were being made, the 004 would be added (IOC,C#4;237) and then removed when the display was to reappear.

The next subject to be discussed is the preparation of a page for display. Before the page is displayed, however, it should be cleared to all spaces to ensure that no extraneous data is displayed. The space code is 000. The following procedure can be used:

a. Select an index register and clear it to zeros. This register will be used as a location counter. Any of the following examples can be used:

   (1) DSP:EQU: P01 000
       LDX:R#1; ADL:DSP

   (2) LDX, R#1; OCT: 000

   (3) LDX, R#1; DEC: 000

Table 3-2. Value of the Small CRT IWR

Page Number	Base Number, No Features Selected		Enable Underscore		Interleave (8-Line Mode Only)	Interleave (8-Line Mode Only) and Enable Underscore.
	8-Lines	4-Lines-Top, Bottom	Add 10 (+10) to Base 8-Lines	4-Lines-top, Bottom	Add 20 (+20) to Base	Add 30(+30) to Base
00	000	021	001	010	031	011	020	030
01	002	023	003	012	033	013	022	032
02	040	061	041	050	071	051	060	070
03	042	063	043	052	073	053	062	072
10	100	121	101	110	131	111	120	130
11	102	123	103	112	133	113	122	132
12	140	161	141	150	171	151	160	170
13	142	163	143	152	173	153	162	172
20	200	221	201	210	231	211	220	230
21	202	223	203	212	233	213	222	232
22	240	261	241	250	271	251	260	270
23	242	263	243	252	273	253	262	272
30	300	321	301	310	331	311	320	330
31	302	323	303	312	333	313	322	332
32	340	361	341	350	371	351	360	370
33	342	363	343	352	373	353	362	372

b. Load a space code into the accumulator. This code will be transferred to all memory locations in the desired page (i.e., page 01). Either of the following forms will work:

(1) LDA,R#0;( )

(2) LDA,R#0;OCT:000

c. Store the accumulator, using index register 1 and incrementing the register, as follows:

STA,I#1;DSP

Note that DSP is equated with P01,000; therefore, page 01 is used along with the location in index register 1 to form the address.

d. Compare the contents of index register 1 to the constant 000. When they are equal, the selected page of memory will have been cleared to spaces. Any of the following instructions will achieve the comparison:

(1) CPX,R#1;OCT:000

(2) CPX,R#1;DEC:000

(3) CPX,R#1;DSP+000

e. A Branch on High (BRH) instruction will then allow the routine to loop until the location count is incremented to the end of the page. When the memory page is cleared, an exit to the display routine is taken. An example of the clearing routine follows:

```
CLR:  LDX,R#1;OCT:000.
      LDA,R#0;OCT:000.
      STA,R#1;DSP.
      CPX,R#1;OCT:000.
      BRH;CLR.
```

After the page is cleared, the programmer has the option of turning on the screen before the page is loaded or after the page is loaded. Both methods work; however, it is important to remember that as new data is transferred into a page that is being displayed, that new data is displayed.

The data to be displayed may be stored in memory in a page other than the displayable page. In this case, a routine that will load the data into the displayable area must be executed. This routine requires the use of two index registers -- one to control the source location, and the other to control the destination location. The Load Accumulator and Store Accumulator instructions achieve the transfer.

The source of the data may be the keyboard, in which case the data will be available in the accumulator. It must be remembered that if the character is in the accumulator as the result of a keyboard IOC instruction, it must be translated to Series 1500 code before it can be displayed. This translation is discussed in the description of keyboard programming.

3-7. LARGE CRT DISPLAY.

The large CRT (LCRT) option is a CRT and keyboard assembly that replaces a similar assembly housing a smaller CRT. This option, which uses a 12-inch CRT rather than the standard 5-inch CRT, has two major advantages:

a. A much larger display area. Up to 24 lines of 80 characters each can be displayed, while the standard unit can display only 8 lines of 32 characters each.

b. An independent memory that stores the data to be displayed, while the standard unit stores the data to be displayed in the terminal's main memory.

The LCRT option is treated as an I/O device, controlled by the IOC C#6 (176) instructions. As noted in Section 2 of this manual, the IOC C#6 instructions are also used to control the communications option. Only when the IWR is in the 1NN series does the instruction apply to the LCRT.

Display Format.

The display, like a television receiver, forms a picture on the front surface of the picture tube; however, the LCRT uses the beam to form only characters. The large CRT works in basically the same way as the standard CRT, but the LCRT scans horizontally, while the standard CRT scans vertically.

The LCRT display area is divided into 24 rows of 80 character positions each. Each character position has a related memory location in the independent memory in which the ASCII code for the character to be displayed is stored. Contrasting this with the standard CRT: the standard CRT has 8 rows of 32 characters each and stores the character codes in the terminal's main memory.

The memory in which the LCRT character codes are stored has the capacity of 4096 bytes, and the memory addresses are arranged so that the memory is related to the display as shown in figure 3-12. Note that columns 81 through 128 and rows 25 through 32 are not used for the display. However, these locations can be addressed by the processor and used for extra storage.

As shown in figure 3-12, the LCRT can create a display in the same format (8 rows of 32 characters each) as the standard CRT. However, since the LCRT is designed to use ASCII character codes, a conversion program must be run to replace the standard CRT dot pattern with ASCII code.

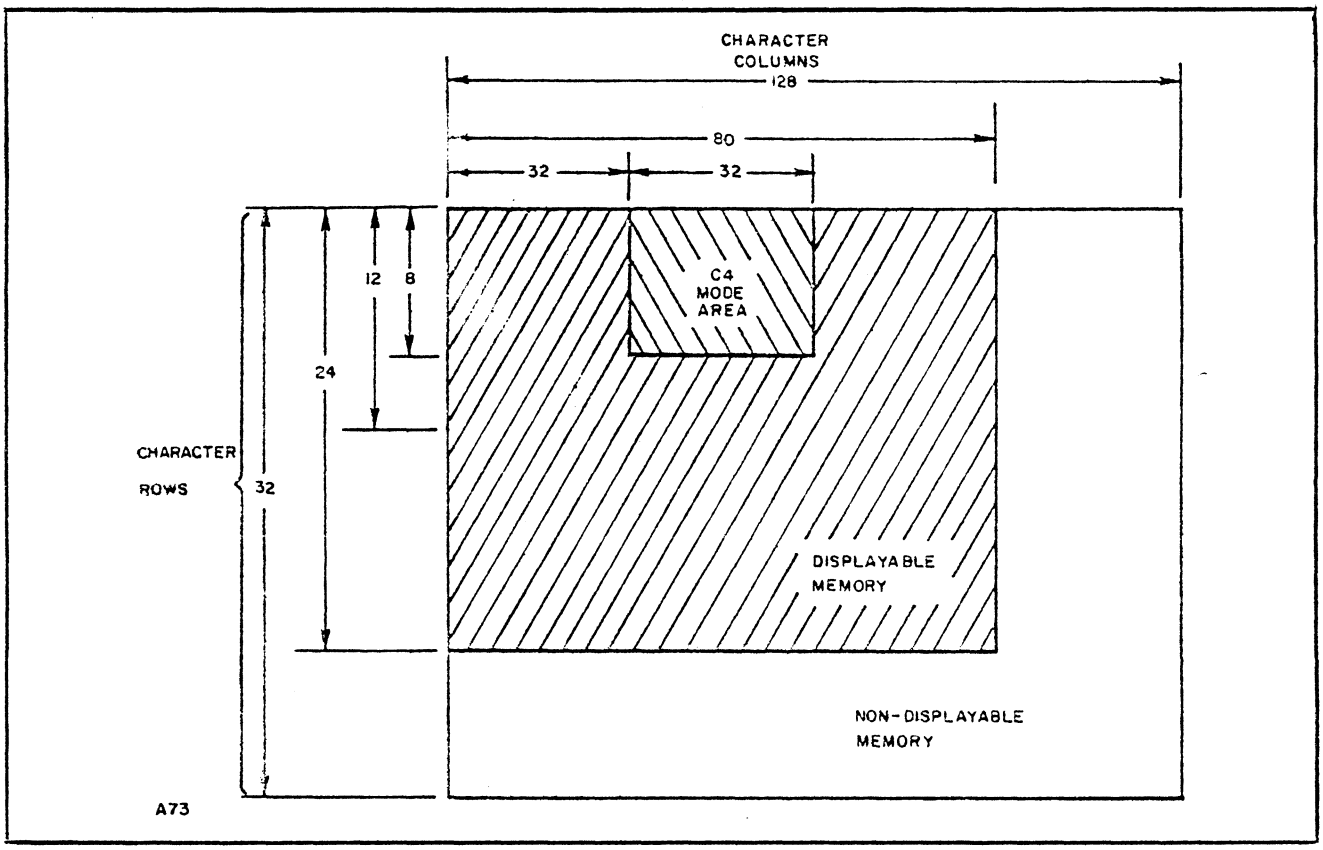


Figure 3-12. Memory Allocation vs. Display Area

Character Matrix.

Each character on the LCRT is made up of 12 scan lines, as shown in figure 3-13. The character codes read from the memory address a ROM which, in turn, supplies the dot patterns required to construct each line of the character. The ROM is read in synchronism with the beam as the beam progresses from one character position to the next.

A typical character is shown in figure 3-13. Each position consists of a 5 by 7 dot matrix for normal characters, a 5 by 2 area for lower-case letter descenders, and a 5 by 2 area for the cursor. The first and last columns are blank, as is the top row. Therefore, the overall character position is a 7 by 12 matrix.

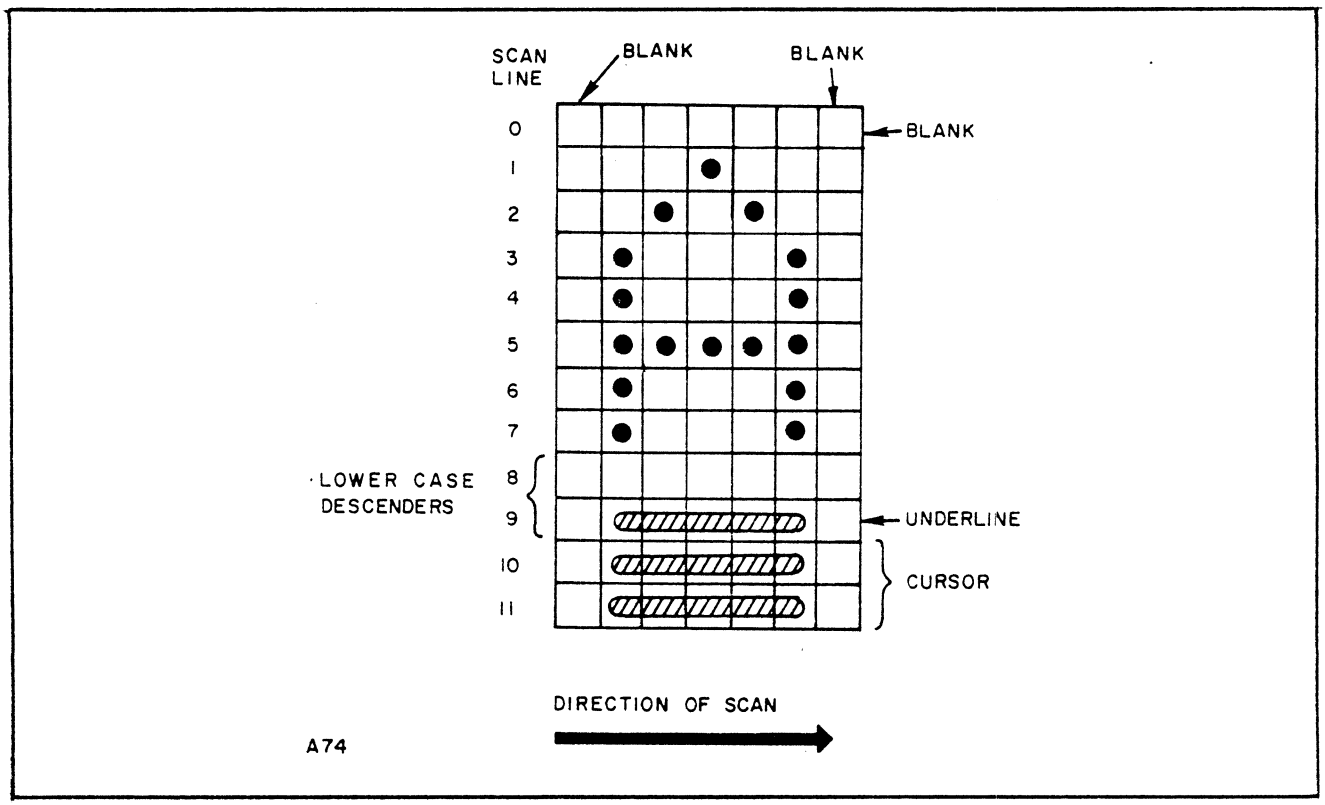


Figure 3-13. LCRT Character Matrix

The Cursor.

The term "cursor" as used in the description of the LCRT option has two meanings, as follows:

a. First, the cursor is a pair of lines beneath a specific character on the display. This is called the cursor symbol in this discussion.

b. Second, the cursor is also a specific memory address, which may or may not be in the displayable area. This is called the cursor location in this discussion.

Included in the LCRT circuits are cursor location counters that keep track of the memory location that the processor is dealing with, the cursor location. These counters are originally loaded by I/O instructions. They are also incremented or decremented by I/O instructions but are not the counters used to address memory while the display is being generated. However, they are related to the cursor symbol displayed, as follows: If the display of the cursor symbol is enabled, the cursor location is compared to each memory address as the character from that address is being displayed. When they match, the cursor symbol is displayed on the CRT. Thus, the cursor symbol is a visual indication of the current setting of the cursor location counters.

LCRT I/O Instructions.

The processor controls the LCRT by issuing I/O instructions in the 176-1FF family. The format of the "1FF" portion of the instruction is shown in figure 3-14, and the purpose of each instruction is as follows:

a. 100 - Clear Screen From Cursor Location. This instruction erases the currently displayed screen from the cursor location to the end of the screen. In the 12-line mode, only the memory associated with the 12 lines displayed is affected. In the 24-line mode, all of the visible display's memory is affected, and in the 32-line mode, all of the memory is affected. The effect of this instruction is to clear memory locations and step the cursor location to line 1, position 1, which is the "home" position.

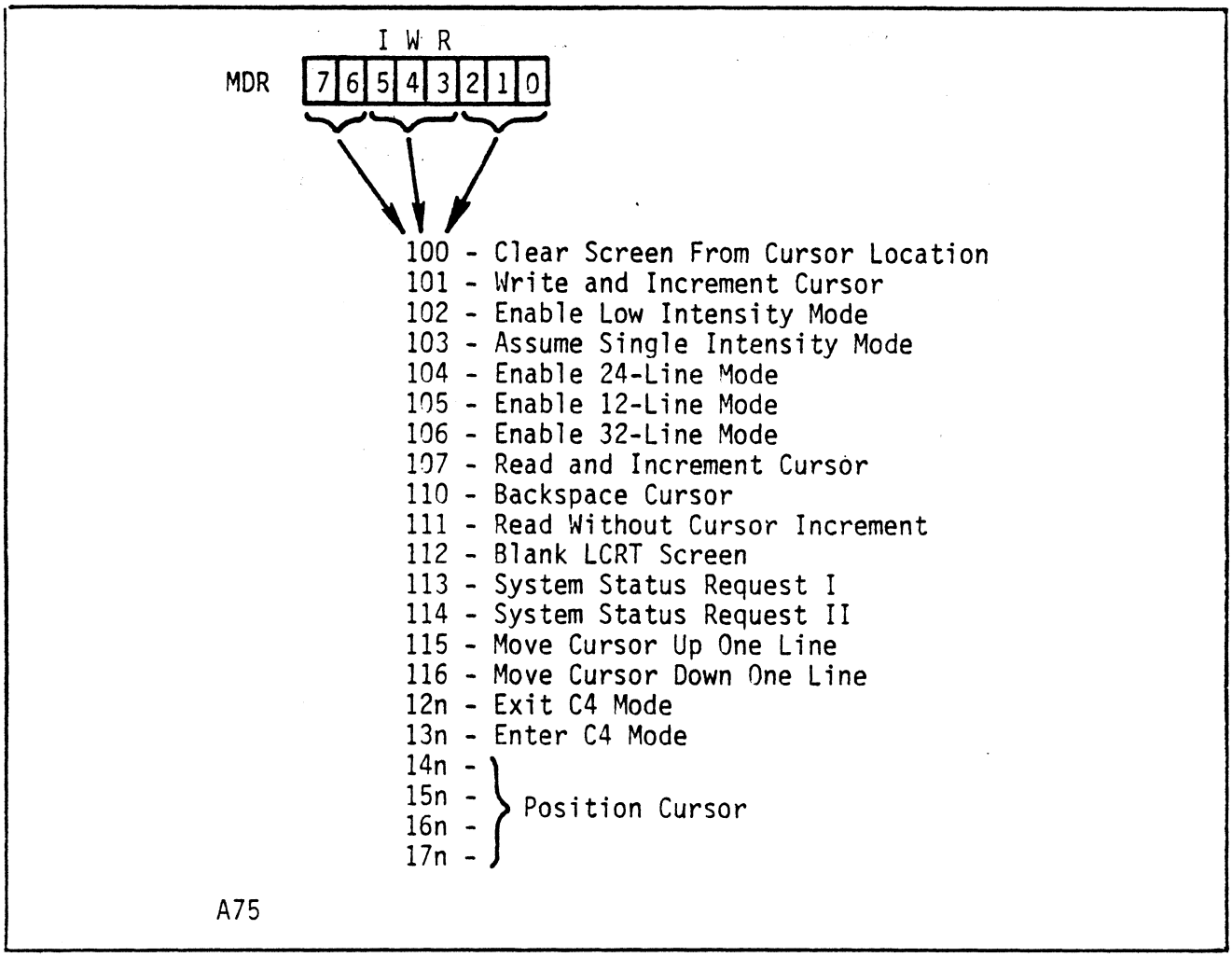


Figure 3-14 . Format of LCRT Instructions

b. 101 - Write and Increment Cursor. This instruction causes the contents of the accumulator to be written into the LCRT memory at the cursor location, which is then incremented by one. Note that this instruction is operable in 32-line mode even though the cursor is not visible, and may in fact point to non-displayable memory. The accumulator contents must be in USASCII code for the displayable memory locations.

c. 102 - Enable Low Intensity Mode. This instruction provides a means of reducing the intensity of a displayed character. If the high-order bit of the character code is a one, the character will be displayed at reduced intensity. Once set, this mode is cleared by instruction 103, or by operation of the system reset switch.

d. 103 - Assume Single Intensity Mode. This instruction cancels instruction 102. In the single intensity mode, the high-order bit of the character code causes the character to be underlined.

e. 104 - Enable 24-Line Mode. This instruction causes the display of 24 lines, 80 characters per line.

f. 105 - Enable 12-Line Mode. This instruction causes the display of 12 lines, 80 characters per line.

g. 106 - Enable 32-Line Mode. This instruction is provided to allow the programmer use of the entire 4096-byte memory. Stepping of the cursor location is extended to a 128-character "line", and a 32-line "screen". Hence, the entire 4096-character memory may be treated as a pseudo display.

h. 107 - Read and Increment Cursor. This instruction reads the contents of the LCRT memory specified by the cursor location into the accumulator; the cursor location is then incremented by one.

i. 110 - Backspace Cursor. This instruction causes the cursor location to be decremented by one, effectively moving the cursor backward one location on the current line.

j. 111 - Read Without Cursor Increment. This instruction reads the contents of the LCRT memory specified by the current cursor location into the accumulator.

k. 112 - Blank LCRT Screen. This instruction disables the display; however, it does not impair the read/write instructions to the LCRT memory. Subsequent issuance of this instruction alternately turns the display on and off.

l. 113 - Systems Status Request I. This instruction reads a status word into the accumulator. The contents of the status word are as follows:

Bits 0 - 4 . . . . . Vertical cursor position (line number)

Bit 5. . . . . "1" indicates that the LCRT is in the C4 mode.

Bit 6. . . . . "1" indicates that the LCRT is in the 12-line mode.

Bit 7. . . . . "1" indicates that the LCRT is in the 32-line, 128-character mode.

Bits 5, 6, and 7 all being "0" signifies the 24-line mode.

m. 114 - Systems Status Request II. This instruction reads a status word into the accumulator, and its contents are as follows:

Bits 0 - 6 . . . . . Horizontal cursor position (location on a given line).

Bit 7. . . . . "1" indicates that the screen is presently in a blank state.

n. 115 - Move Cursor Up One Line. This instruction causes the cursor location to be decremented by one line, effectively moving the cursor up one line. If the cursor is already on line one, this instruction has no effect.

o. 116 - Move Cursor Down One Line. This instruction causes the cursor location to be incremented by one line, effectively moving the cursor down one line. If the cursor is already on the last line, this instruction will cause a wrap-around; that is, the cursor will move to line one.

p. 12N - Exit C4 Mode. This instruction causes the LCRT to leave the C4 mode of operation and enter 12-line mode.

q. 13N - Enter C4 Display Mode. This instruction causes the LCRT to operate in the C4 mode of operation.

r. 14N through 17N. Position Cursor. This set of instructions allows the cursor location to be set to any of the 4096 memory locations.

The IWR "points" to the line, while the accumulator contents define the location (0 through 127) on the line. The IWR addresses lines as follows:

140 through 147 = lines 1 through 8

150 through 157 = lines 9 through 16

160 through 167 = lines 17 through 24

170 through 177 = lines 25 through 32

All of the I/O instructions mentioned above, except "Clear Screen", are executed within the 3-microsecond minimum execution time used in the processor. The Clear Screen instruction, however, requires an additional 1.29 microseconds for every character position to be cleared, and the LCRT circuits activate the stall signal to the processor while this instruction is being executed. This causes the processor to remain at the current instruction which, of course, is the Clear Screen instruction, until all positions are cleared.

One other I/O instruction also causes the stall signal to be activated. This is the Exit C4 Mode instruction. If the LCRT circuits receive this instruction while in the process of converting video from the small CRT into the large screen format, they activate the stall signal until the conversion is completed. Thus, the processor remains at the Exit C4 Mode instruction for the time necessary to make the conversion.

Modes of Operation.

There are four distinct modes of operation for the LCRT circuits. These modes are selected by the I/O instructions described above and are as follows:

a. 24-line, in which 24 lines of 80 characters each are displayed. This display occupies the full screen and is about 9 inches wide and 6-1/2 inches high.

b. 12-line, in which 12 lines of 80 characters each are displayed. This display also occupies the full screen but is "double-spaced".

c. C4 mode, in which 8 lines of 32 characters each are displayed in the upper center portion of the screen.

d. 32-line, which is not a true display mode but is the mode in which the processor gains access to the non-displayable portion of the memory. In this mode, 24 lines of 80 characters each are displayed.

When power is first applied, the LCRT begins operation in the C4 mode. A system reset signal or a program load signal in the processor also produce the clear signal required to place the LCRT in the C4 mode. Once in this mode, the Exit C4 Mode instruction must be given in order to move to another mode of operation, and this instruction places the LCRT circuits in the 12-line mode.

C4 Mode.

The C4 mode derived its name from the C4 processor used in the 1501, 1502, and 1503 Intelligent Terminals. When the LCRT is operating in this mode, it must use the data that would normally produce a small screen display.

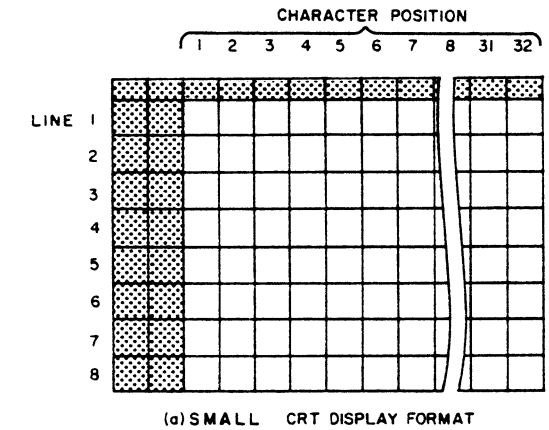
Figure 3-15(b) illustrates the normal small CRT scan. The sweep is vertical, painting top to bottom and left to right. Interleaving is employed so that the odd field consists of scan lines 1, 3, 5, and 7 of all characters and the even field produces lines 2, 4, 6, and 8 of all characters.

The display space is pictured in figure 3-15(a). This consists of eight lines, each containing 32 characters. Note the shaded areas. At the left are two positions of non-displayable data, which would consist of eight odd and eight even scan lines. At the top, or beginning, of each sweep are eight dot positions of non-displayable data. The shaded portions shown must be discarded in the data conversion.

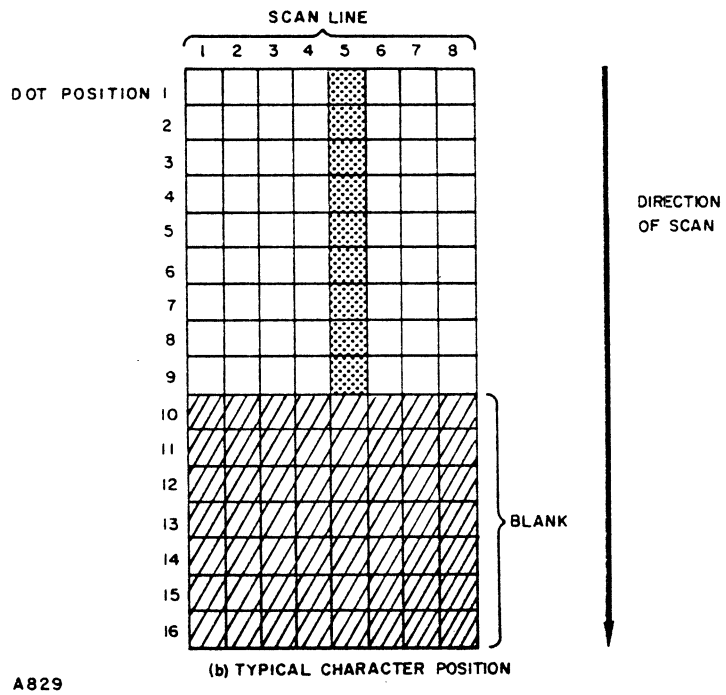
Each character space, which is shown in figure 3-15(b), consists of 16 dot positions vertically by 8 scan lines (4 odd and 4 even scan lines) horizontally. Of the 16 vertical dot positions, 9 are for character and cursor display and the remaining 7 positions are space.

In this conversion for the LCRT, the data is to be stored in LCRT memory so that normal display functions can be employed. This requires that an ASCII code be stored in the LCRT memory; therefore, only one of the eight vertical lines need to be utilized to specify the character in ASCII code. For this purpose, the third scan (line 5) of the odd scan is selected. This is the shaded area in figure 3-15(b). The original video that would have been displayed on the small CRT is replaced by the weighted ASCII code by a conversion program run by the processor. To summarize: The third odd scan line of each character position in the small CRT format carries an ASCII code that will select the character for display on the LCRT.

At this point, a brief explanation of the selection of scan line 5 to hold the weighted ASCII code is required. It was previously mentioned that data for the standard CRT is stored in the processor's memory. The dot patterns occupy page P04 and part of P05 for the standard character set and page P06 and part of P07 for the alternate character set.



(a) SMALL CRT DISPLAY FORMAT



(b) TYPICAL CHARACTER POSITION

A829

Figure 3-15. Small CRT Formats

Scan lines 1, 2, 3, and 4 of each character are stored in the first page of each pair (P04 and P06), while scan line dot patterns for line 5 are stored in the second page of each pair. Since scan line 5 is used to carry the weighted ASCII code, pages P04 and P06 are free to be used for other storage by the processor.

Options.

There are three options that may be installed in the LCRT. The programmer must take these options into consideration when using the large display. These options include:

a. The character set, which is controlled by a Read Only Memory (ROM) that is pre-programmed with the desired character set. At this writing, ROM's are available in standard ASCII code set or "Kana" (Japanese) code set. However, any code set can be provided upon special request.

b. The 12-line to 24-line mode is controlled by the addition of some circuits plus a jumper plug.

c. The underline capability is standard; however, there is an option to eliminate this feature.

Programming Aids.

The large CRT provides a wide range of options not available with the small CRT, and the following paragraphs cover some of the methods used to program these options. Before starting the description of programming methods, however, a review of the large CRT is in order. The large CRT has its own memory, part of which is used for display; the remaining part is used for general storage. The dot patterns displayed on the screen are held in a ROM, which uses the USASCII code to select the patterns. All data sent to the large CRT to be displayed must be in USASCII and have the USASCII character in the accumulator prior to executing a Write and Increment Cursor instruction.

It follows, then, that all data read from the displayable locations of the LCRT memory will be in USASCII and will have to be translated back into Series 1500 code, which is the standard internal code for the Series 1500 terminals. Data from the non-displayable locations can be handled in Series 1500 code without translation.

The cursor points to the next memory location to read from or write into. The cursor can be incremented, decremented, or relocated by I/O commands. It is also possible to determine the cursor location through the use of the status instructions. When the cursor is incremented past the limits of a line, it automatically goes to the first character position of the next line.

In the 12- and 24-line modes, the cursor will go back to line one or wrap around after the end of the display is reached. In 32-line mode, the cursor symbol is turned off, but the cursor location still points to the current memory location, advances to the next line, and wraps around to line 1 after line 32.

Code Translation

Conversion from Series 1500 code to USASCII code was discussed above. The generation of the translation table is a programmer's option. The table can be generated and stored on the object tape or disc, or the table can be generated each time the program is loaded. One method saves time; the other saves space.

Regardless of how the table is generated, it should be similar to the one in figures 3-16 and 3-17. It should be organized in a format that allows the code of the characters to be translated to be used to form the address of the new code. For example, to translate an "A" from Series 1500 code to USASCII, the code for an "A" is 015 octal; therefore, the USASCII code for an "A" should reside in location 015 of some designated page. As shown in figure 3-16, location P10-015, the USASCII code for the "A" is 101 octal.

The instructions involved in a conversion routine might take the following form:

Instruction	Comments
LDA,R#0 (X)	Series 1500 character code to be translated.
STA,R#0;R#1	Store Series 1500 character code in index register 1.
LDA,R#1;XLT	Use the Series 1500 character code to form the address of the USASCII code, where XLT is the page where the translation table is stored.

The USASCII code for an "A" will be in the accumulator after the above sequence; at this point, it can be transferred to the large CRT memory.

A similar sequence of instructions is used to translate from USASCII to Series 1500 code. Any time data is transferred from the LCRT memory to the terminal's main memory, it should be translated to Series 1500 code.

```
LAB: CMD  OPERANDS        COMMENTS

XLT: LOC, P10: 000.      TRANSLATE TBLS
 1500  CHARACTER CODES TO ASCII FOR LC
BL - + 0 1 2
     OCT, (040-055-053-060-061-062).
3 4 5 6 7 8
     OCT, (063-064-065-066-067-070).
9 A B C D E
     OCT, (071-101-102-103-104-105).
F G H I J K
     OCT, (106-107-110-111-112-113).
L M N O P Q
     OCT, (114-115-116-117-120-121).
R S T U V W
     OCT, (122-123-124-125-126-127).
X Y Z , £ @
     OCT, (130-131-132-054-043-100).
BL % $ * . <
     OCT, (177-045-044-052-056-074).
> / ( ) ? CENT
     OCT, (076-057-050-051-077-133).
= " ! ' : ;
     OCT, (075-042-041-047-072-073).
U-SCORE - & (
     OCT, (137-040-046-174).

     LOC, P10: 100.
BL - + 0 1 2
     OCT, (040-055-053-060-061-062).
3 4 5 6 7 8
     OCT, (063-064-065-066-067-070).
9 A B C D E - LOWER CASE
     OCT, (071-141-142-143-144-145).
F G H I J K - LOWER CASE
     OCT, (146-147-150-151-152-153).
L M N O P Q - LOWER CASE
     OCT, (154-155-156-157-160-161).
R S T U V W - LOWER CASE
     OCT, (162-163-164-165-166-167).
X Y Z - LOWER CASE - , £ @
     OCT, (170-171-172-054-043-100).
BL % $ * . <
     OCT, (177-045-044-052-056-074).
> / ( ) ? _
     OCT, (076-057-050-051-077-133).
= " ! ' : ;
     OCT, (075-042-041-047-072-073).
U-SCORE BL & LOG-NOT
     OCT, (137-040-046-174).

A832
```

```
LAB: CMD  OPERANDS        COMMENTS

     LOC, P10: 200.
ASCII TO  1500  CHARACTER CODES
BL
     OCT, (000).
LOWER CASE: A B C D E F
     OCT, (315-316-317-320-321-322).
G H I J K L
     OCT, (323-324-325-326-327-330).
M N O P Q R
     OCT, (331-332-333-334-335-336).
S T U V W X
     OCT, (337-340-341-342-343-344).
Y Z
     OCT, (345-346).

     LOC, P10: 240.
BL ! " £ $ %
     OCT, (000-070-067-050-054-053).
& ' ( ) * +
     OCT, (076-071-062-063-055-002).
, - . / 0 1
     OCT, (047-001-056-061-003-004).
2 3 4 5 6 7
     OCT, (005-006-007-010-011-012).
8 9 : ; < =
     OCT, (013-014-072-073-057-066).
> ? @ A B C
     OCT, (060-064-051-015-016-017).
D E F G H I
     OCT, (020-021-022-023-024-025).
J K L M N O
     OCT, (026-027-030-031-032-033).
P Q R S T U
     OCT, (034-035-036-037-040-041).
V W X Y Z CENT
     OCT, (042-043-044-045-046-065).
BL DSP-QUOTE
     OCT, (000-135-136-074).

MODIFIED LAST TWO FOR UPPER 'W' ())
CLR: SMS, CLL.           JUMP PAD TO
     BRU, CLL.           CLEAR SCREEN

CHECK U/L FLAG
CHK: LDA, R&0: U/L,      GET U/L FLAG
     TLX, 000: OCT:000. OFF - COND EQ
     EXU, 000.           ON - COND HI
```

Figure 3-16. Series 1500 Code to USASCII Translation

```
LAB.  CMD   OPERANDS        COMMENTS

      LOC,  P05; 300.
************************************<
TRANSLATE FOR LARGE CRT
************************************

TRD.  LDX,  R#1; OCT:000.
      LDX,  R#2; TR1.
      SBU,  TRL.              MOVE 1ST 3
      LDA,  R#0; OCT:020.     NOW DO MIDDLE
TRI:  STA,  I#1; SY1.
      ADA,  R#0; OCT:001.
      TLJ,  +06; OCT:032.     NUMBER DONE
      TLJ,  TRL; OCT:173.     LETTERS DONE
      BRU,  TRI.              KEEP ON TRUCKI
      LDA,  R#0; OCT:041.     START LETTERS
      BRU,  TRI.
TRL:  LDA,  I#2; TR1.         MOVE FROM TABL
      TLX,  000; OCT:377.
      STA,  I#1; SY1.
      BRU,  TRL.
TR1:  OCT,  (000-015-013-377).
      OCT,  (104-003-140-047-005-004).
      OCT,  (012-016-034-036-017-010).
      OCT,  (011-037-175-035-002-001).
      OCT,  (007-032-033-177-176-006).
      OCT,  (137-377).

A833
```

Figure 3-17. Weighted USASCII Code Generator

Initial Steps

The first instruction of the LCRT program must be IOC,C#6;12n. This instruction causes the LCRT to exit from the C4 mode and enter the 12-line mode. The instruction is necessary since the machine always comes on in the C4 mode when power is applied. This instruction also moves the cursor to home position. At this point, if the 24- or 32-line mode is required, the IOC,C#6;104 or IOC,C#6;106 instruction must be executed.

With the mode selected and the cursor in the home position, the LCRT memory can be loaded. When a character code is written into this memory, it is placed in the position to which the cursor is pointing. Writing can start in any location. (The cursor does not have to be at home position to start writing; this is being used as an example only.)

After the first character to be displayed is translated, it is left in the accumulator. The execution of the IOC,C#6;101 (Write and Increment Cursor) instruction transfers the contents of the accumulator to the LCRT memory and increments the cursor. The sequence of translating and writing is continued until the screen has been filled. The cursor will automatically go from the last position in a line to the first position of the next line, with wrap-around from the last position of the last line to the first position of the first line.

Moving the Cursor Symbol

Positioning of the cursor is achieved by the IOC,C#6;14n through 17n instructions. The function portion, 14n through 17n, will select the line. The position within the line is selected by placing the desired location in the accumulator before executing the position cursor instruction. The contents of the accumulator will fall in the range of 000 through 177 octal, representing the decimal values of 000 through 127.

Clearing the Screen

When clearing the screen by execution of the IOC,C#6;100 instruction, the programmer must realize that this instruction performs differently in each of the modes available. However, in all modes, the screen is cleared from the cursor position to the end of the screen. In 12-line mode, only the 12 lines being displayed are affected. The remaining memory is not altered. In the 24-line mode, the entire displayable memory is cleared. In the 32-line mode, all 4096 locations are cleared.

In all cases, the entire display does not have to be cleared. The cursor can be placed anywhere within the display, and the clear instruction will affect only that area from the cursor to the end of the display.

Acquiring Cursor Location

The situation may arise where the program requires knowledge of the location of the cursor. In this case, the status instructions which obtain the position of the cursor can be executed. The Systems Status Request I (IOC,C#6;113) instruction will leave the status word in the accumulator. Bits 0 through 4 of the accumulator will indicate the line (0 through 37 octal, 0 through 32 decimal) where the cursor is located. The Systems Status Request II (IOC,C#6;114) instruction will leave the status word in the accumulator. Bits 0 through 6 of the accumulator will indicate the position within the line (000 through 177 octal, 000 through 127 decimal) where the cursor is located.

Use of the Extra Memory

The non-displayable memory can be used as an extra storage area. Access to this area is achieved through the use of the Enable 32-Line Mode (IOC,C#6;106) instruction and the Write/Read and Increment Cursor (IOC,C#6;101 and IOC,C#6;107, respectively) instructions. The cursor symbol is not displayable in this mode, but it is effectively there and can be controlled as if it were displayed. The location acted upon by the read or write instruction is the location the cursor points to, even if outside the normal display area. There is no restriction on the data stored outside the display area, since it is not for display and does not have to be in USASCII code.

Using the LCRT in the C4 Mode

The C4 mode allows the LCRT to produce displays originated for the small CRT. This means that existing programs will function when the large CRT is added. The small CRT dot pattern must be altered, but this is the only change required to allow the C4 mode to function.

As mentioned earlier, the large CRT uses USASCII as the code to generate the display characters. Therefore, the CRT dot pattern located in page 04 and part of page 05 (alternatively, page 06 and part of page 07) will have to be changed for the C4 mode to function. The hardware dictates that locations 00 through 77 in page 05 (alternate 07) be used to contain the weighted USASCII code for the respective character.

The existing programs generated for the small CRT can be converted to be used with the large screen in C4 mode by using the Large Display Object Converter (Catalog No. 70153-00). This conversion consists of replacing the CRT dot pattern in page 05 (alternate 07) with the USASCII codes.

After the program has been converted, no other alterations are required. When power is turned on in the terminal with a large screen, the terminal will come on in the C4 mode; therefore, when the program is loaded, it will be ready to run.

The conversion does not change the IOC instructions. That is, all the IOC,C#4;FFF instructions in the original program remain the same; they are not converted to IOC,C#6;FFF type instructions.

If a large screen application requires the use of the C4 mode to enhance the operation of a program, the programmer may have to use both large screen and C4 modes of operation. In this programming situation, the program will have to combine instructions for both C4 mode and large screen mode. To aid the programmer, Translate Table (Catalog No. 60407-00), which is used to replace the dot pattern in P05,000-077, is available. This table is the same as the one used in the conversion routine; however, it is used when a program is generated instead of to convert an existing program. If the programmer elects to generate the translation table when the program is loaded, a table generator routine similar to the one in figure 3-17 can be used.

Large CRT Programming Restrictions

The following programming restrictions should be considered when programming the large CRT:

a. In the IOC, C#6;110 (Backspace Cursor) instruction, use of this instruction when the cursor is in the "home" position of any line is not recommended; it actually causes a line increment. If on the last specified line, it causes a wrap-around.

b. A Read instruction (IOC,C#6;107 or 111) or a Status Request instruction (IOC,C#6;113 or 114) cannot immediately follow a Write and Cursor Increment instruction (IOC,C#6;101). At least one instruction must separate these (i.e., IOC,C#6;101 write and increment, NOP:000 timing, IOC,C#6;107 read, or IOC,C#6;114 status request).

c. Following a Clear Screen instruction (IOC,C#6;100), issuance of any IOC,C#6 instruction will result in "stall" until the screen is cleared.

Large CRT Sample Programs

Shown in figure 3-17A is a sample program for the large CRT. The intention of this example is to show how the instructions are used; it is not necessarily a completely operational routine.

The translation routine shown in figure 3-17B is used to translate from USASCII to Series 1500 code. Before this translation routine is executed, the Series 1500 code table must be stored into memory. This same approach can be used to achieve any code translation. The only requirement is that the appropriate code table be stored in memory.

```
************************************
MINUS KEY - DELETE CHARACTER
MOVE THE LINE ONE POSITION TO LEFT
************************************
DEL: IOC, C#6; 114.      STATUS CK
     STA, R#0; R#6.      SAVE CRSR LOCT
     IOC, C#6; 107.      READ/INCREMENT
     IOC, C#6; 114.      CURSOR LOCATON
     TLJ, DL2; OCT:000. END OF LINE???
     IOC, C#6; 111.      READ/NO INCREM
     IOC, C#6; 110.      BKSP CURSOR
     IOC, C#6; 101.      WRITE/INCREMEN
     NOP, 000.           TIMING
     BRU, DEL+04.
DL2: IOC, C#6; 113.      LINE & STATUS
     SUA, R#0; OCT:001. DECRMNT
     ADA, R#0; OCT:140. SET FOR IOC
     LDX, R#1; II#+01.  ALTER INSTRUCT
     STA, R#1; II#.
     LDA, R#0; OCT:117. RESET CURSOR
II#: IOC, C#6; 166.      NEW LINE POS
     LDA, R#0; OCT:040. SET A ASCII BL
     IOC, C#6; 101.      WRITE/INCREMNT
     NOP, 000.           TIMING
BK@: IOC, C#6; 113.      STATUS OF LINE
     ADA, R#0; OCT:137. PUT CURSOR BAC
     LDX, R#1; RAA+01.  ALTER INSTRUCT
     STA, R#1; RAA.      STORE IT
RA@: LDA, R#0; R#6.      SAVED LOCATION
RAA: IOC, C#6; 166.      ALTERED LINE
     SMS, ST1.           EXIT TO
     BRU, ST1.           MAINLINE

A998
```

Figure 3-17A. Large CRT Sample Program 1

```
************************************
TRANSLATE ASCII CODE TO ICL CHAR COD
************************************
COG: CPA, R#0; OCT:141. CHECK FOR 'A'
     BRL, XX1.           OUTSIDE RANGE
     CPA, R#0; OCT:172. CHECK FOR 'Z'
     BRH, XX1.           OUTSIDE RANGE
     ADA, R#0; OCT:040. ADD L.C. DISPL
     BRU, XX2.            + CONTINUE
XX1: ADA, R#0; OCT:200. ADD DISPLACEMN
XX2: STA, R#0; R#2.      TEMP STORE
     LDA, R#2; XLT.      GET TRANSLATE
     EXU, 000.

A999
```

Figure 3-17B. Large CRT Sample Program 2

3-8. CARTRIDGE TAPE DRIVES.

Up to eight cartridge tape drives can be operated by the Model 1501 Intelligent Terminal; two are standard and six are optional. The optional drives are added in pairs in the Model 1533 Dual Drive, which is a separate unit. The larger terminals (1502, 1503, and 1504) are normally equipped with only two tape drives, although two additional drives are optional.

Recording Method.

The tape itself is very narrow (0.15 inches wide), and it records only one channel of serial data. A small cartridge holds 100 feet of this tape.

The data is recorded in the Manchester Code, which is also called phase encoding. In this method, a signal transition always occurs in the center of a bit period regardless of whether the bit is a one or a zero. Thus, the timing signals required to process the data when it is read can be developed directly from the signal itself. This allows considerable variation in tape speed with no effect on the data.

Record Format.

The format of the records on the cartridge tapes was discussed earlier in this section. Reviewing that description: each record consists of a preamble, a label, the data, a CRC check byte, and a postamble. The preamble and postamble are used only by the reading circuits and are of no concern to the programmer other than that the program must provide them when writing a record.

The format of each record was shown earlier in this section in figure 3-3. Note that the tape can be read in either direction. When the tape is read forward, the preamble produces 23 zeros followed by a one; when the tape is read backward, the postamble produces 23 zeros followed by a one. This is the synchronization pattern required by the reading circuits in order to locate the beginning of a record and start reading.

The size of a standard data record is determined by two factors:

a. Convenience of programming. The 128 data bytes occupy one-half a memory page or disc sector.

b. Physical position of the heads. In order to do a read-after-write operation to check the recorded data without moving the tape backward, the record size cannot exceed the distance between the write and read heads. The write and read heads are approximately 3/4-inch apart; therefore, the maximum record size would be approximately 148 bytes. The 148-byte record does not offer convenient data manipulation into and out of memory; therefore, the 128-byte record was established as standard.

Cartridge Tape Instructions.

The program segments discussed in the standard backgrounds description relieve the programmer of the task of programming most mini-tape functions. However, it may be necessary to alter the backgrounds for special applications, or a programmer may wish to program the tape functions without using the standard backgrounds. Therefore, the tape instructions are described in detail in the following paragraphs.

Input/output instructions are sent from the processor to the I/O controller to control the tape units. In turn, the I/O controller decodes the instructions and manipulates the tape. Figure 3-18 illustrates the format of the I/O instructions. Note that the two low-order bits of the IWL select the tape unit and that the IWR determines the action to be taken.

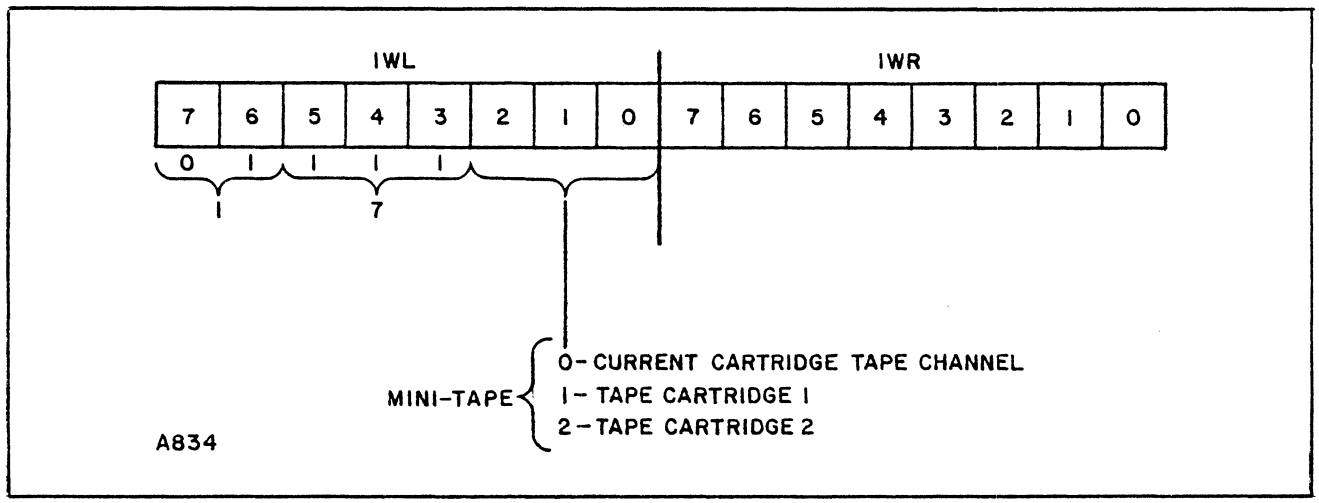


Figure 3-18. Format of Cartridge Tape Instructions

A source statement provided to the Assembler System would take the form:

IOC,C#0;0YY (170-0YY)

IOC,C#1;0YY (171-0YY)

IOC,C#2;0YY (172-0YY)

Where: the "IOC" denotes input/output and thus the 17 code. The number given after the "C" denotes drive selection. The "0YY" denotes the function to be performed.

Each of the functions is described below; the number given before the instruction title is the octal value of the IWR of the instruction.

a. 000-Start Tape Forward at Normal Speed with Erase. This instruction is normally given at the beginning of a writing sequence. It starts the tape moving forward at 10 inches per second and erases it until a write instruction is given. Approximately 30 milliseconds are required for the tape to reach a stable operating speed of 10 ips.

b. 001-Start Tape Forward at Normal Speed Without Erase. This instruction performs the same function as the preceding instruction except that the tape is not erased. It is generally given at the beginning of a read sequence.

c. 002-Start Tape Forward at High Speed. This instruction starts the tape moving forward at 40 inches per second.

d. 003-Start Tape in Reverse at Normal Speed. This instruction starts the tape moving in reverse at 10 ips. Erasing is prevented whenever the tape is moving in reverse. If the leader clip reaches the home position, indicating that the tape is rewound, the tape is stopped.

e. 004-Start Tape in Reverse at High Speed. This instruction performs the same function as the preceding instruction except that the tape is moved at 40 ips.

f. 005-Stop Tape. This instruction stops the tape.

g. 007 or 207-Transfer Byte. This instruction controls the transfer of characters to and from the tape. The transfer is controlled by a busy (not ready) condition within the tape circuits, and the instruction can be executed in two modes: (1) stall if busy (007), and (2) skip if busy (207). In the stall-if-busy mode, the program stops at the transfer byte instruction until a tape sprocket signal is generated, indicating that a byte has been written or read. In the skip-if-busy mode, the program automatically skips the next instruction and proceeds.

h. 010-Set Write Mode. This instruction starts the write operation and begins the timing sequence that controls the writing frequency, loads and shifts the tape buffer, and generates the sprocket signal.

i. 011-Set Read Mode. This instruction activates the tape read circuits. It is executed only once in a normal tape read sequence, and setting of the read condition clears the write condition.

j. 012-Rewind. This instruction sets a rewind flip-flop for the specified tape. This flip-flop is reset only by the clip-in signal, which indicates that the tape is fully rewound. This permits each tape to rewind independently of operations going on in other tape units.

Tape Unit Selection.

The cartridge tape drives are available in pairs. On the 1501, the first pair is located on the terminal; the remaining three pairs, which are housed in Model 1533's, are physically separated from the terminal. In the case of the 1502, 1503, and 1504, both pairs are located on the terminal. Tape selection is achieved by first selecting one of the four possible pairs and then selecting one of the two drives in a pair.

When power is first applied or whenever the general clear signal is generated, the first pair of tape drives is selected. Either the PROGRAM LOAD or SYSTEM RESET switch generates the general clear signal.

During the execution of a program, the pair selected can only be changed by executing a Read Status I/O instruction. The Read Status instruction takes the following format in an assembler source statement:

```
IOC,C#N;OY6
```

The following combinations of channel numbers and function codes are used to select any drive from 1 to 8 and load its status byte into the accumulator:

```
C#1; 016 - Select drive 1 and load status byte
C#2; 016 - Select drive 2 and load status byte
C#1; 026 - Select drive 3 and load status byte
C#2; 026 - Select drive 4 and load status byte
C#1; 036 - Select drive 5 and load status byte
C#2; 036 - Select drive 6 and load status byte
C#1; 046 - Select drive 7 and load status byte
C#2; 046 - Select drive 8 and load status byte
```

If the status of the current drive (the drive already selected) is required, an IOC,C#0;016 instruction may be used. This will hold true regardless of which pair is selected.

A Write Operation.

Before an attempt is made to write on the tape, careful consideration must be given to the physical tape, the tape cartridge, first records, records other than the first, inter-record gap, and the end of the tape. Points to be considered are discussed in the following paragraphs.

The tape is physically 100 feet long, with a leader and eyelet at the beginning, and an end-of-tape marker, which is a reflective strip, at the end of allowable recording area. In order to move the tape past the eyelet and past any portion of tape that may generate errors because of handling, the tape should be moved forward 21 inches before writing the first record. Therefore, a delay of 2.1 seconds should be introduced before the first record is written. The procedure for checking a "first record" condition is to examine bit 5 of the status word. If bit 5, "clip out" is a zero, the tape has been rewound and a first record delay is required.

When writing other than the first record, a 35-millisecond delay should be introduced. The reason for this delay is to allow the tape to reach the proper speed, 10 inches per second, before starting to write on tape.

Regardless of whether the record written is a first record or other than first record, a stop delay of 35 milliseconds must be executed. This delay generates part of the inter-record gap.

Normally, the write instruction is followed by a read instruction, in order to do a read check on the record that was just written. Refer to figure 3-19, which shows the position of the record with respect to the write and read heads immediately after writing. The record is positioned just before the read head. If a read instruction is executed at this point, the record can be check-read. Therefore, the read check sequence can be entered after the write operation without going into the write stop delay. Execution of the read instruction will terminate the write instruction.

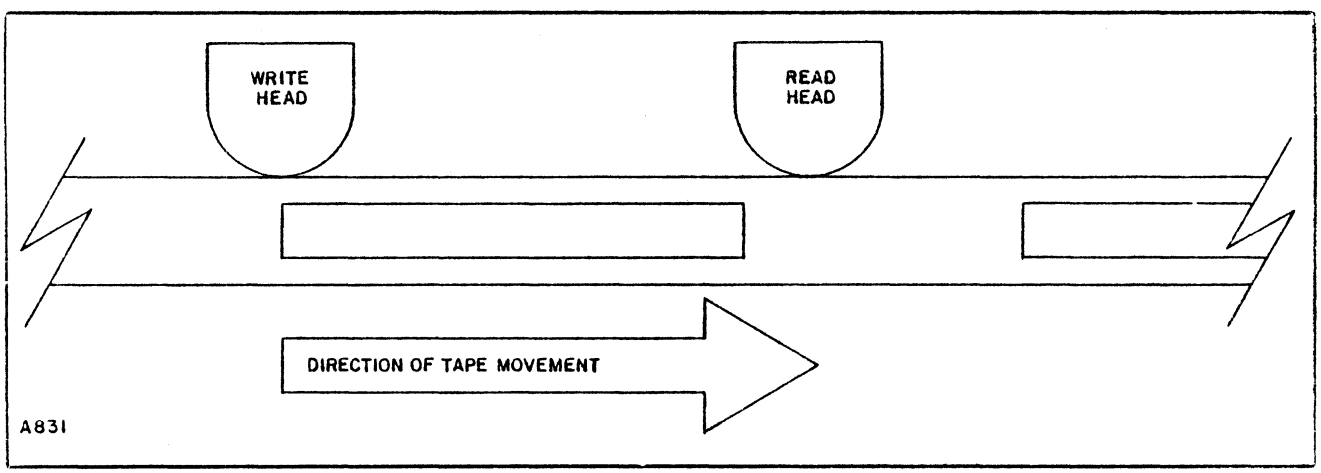


Figure 3-19. Length of Record with Respect to Head Positions

Refer to figure 3-3 and note the record format: that is, preamble, label, data, CRC, and postamble. All of these bytes except the first byte of the preamble must be furnished by the program. The first byte of the preamble (all zeros) is automatically provided. After the first byte is recorded, the program provides the second and all subsequent bytes of the record.

The label is not fixed at eight bytes; however, this format is used in the standard background routines and is used here as an example. The standard label is broken down into the following parts:

a. Byte 1 is the record sequence number. It begins with 001, goes to 377 octal, and then is reset to 001.

b. Byte 2 is an end-of-program load (code 375) or end-of-file (code 377). Byte 2 is usually 000 otherwise.

c. Bytes 3 and 4 are blank for the programmer's use. They are used only in the first record for linkage to part of the BTS program segment in the backgrounds.

d. Bytes 5, 6, and 7 are the program identification.

e. Byte 8 is used with program tapes to define where the program will load. For example, 000 is page 0, lower half of page; 001 selects page 0, upper half of page, etc.

The data bytes are made up of any data the programmer wishes to store on the tape. Obviously, this information could be either data or a program. As explained earlier, the length of the data portion of the record is, for most convenient use, 128 bytes in length.

The CRC (cyclic redundancy check) character is generated by the program and then written on the tape after the data has been written. The CRC is generated in the standard backgrounds by the use of an Add to Accumulator (ADA) and Shift and Exclusive OR (SER) instruction.

There is a location in memory assigned to store the CRC as it is being generated. In the standard background, the CRC is stored in location 008. After the data byte has been transferred to the I/O controller to be written on tape, the byte still remains in the accumulator. At this point, the contents of locations 008 are added to the accumulator. The contents of location 008 will be zeros on the first byte; after the first byte, the contents of location 008 will be the partial CRC. After the addition of the contents of location 008 to the accumulator, a SER instruction is executed with a constant of 242 and a shift count of 3. The result is then stored in location 008. This procedure continues through the entire data field. To generate the CRC that will be written on tape, the contents of location 008 are subtracted from a constant 125. The result is then written on tape as the CRC.

During the read function, as the data is read from the tape, a CRC byte is generated with the same subroutine used for the write operation. The difference is that on a read, the end result is not subtracted from the constant 125; instead, the CRC from tape is added to the CRC that was developed. The sum should be zero.

After the CRC is written, the postamble has to be written. This pattern is the opposite of the preamble pattern, allowing for backward reading. To ensure that the write circuit is on long enough to write the last character of the postamble, a dummy byte is transferred to the tape circuits (but is not actually recorded) before the write operation is ended.

Read Check and Rewrite Operations.

After completing the write operation, the next step is to perform a read check. During the read check, the data read from tape can be checked byte-by-byte against the data in the write buffer in memory. If an error is detected during this check, the record must be rewritten.

The first step in the rewrite sequence is to reposition the tape to a point before the incorrect record. The error would be detected at the end of the read check; therefore, tape forward motion must be stopped. The IOC,C#5;005 (stop) instruction is executed, followed by a 35-millisecond delay to allow the tape to physically stop. The IOC,C#0;003 (reverse slow) instruction will start tape moving in reverse. A delay of 179 milliseconds will allow time enough for the tape to move to a position that will place the write head before the incorrect record. At the end of the 179-millisecond delay, a stop tape instruction, followed by a 35-millisecond delay, will allow the tape movement to stop and the tape will be in a position to rewrite the record.

When rewriting the record, the tape will be erased as soon as the IOC,C#0;010 (write mode) instruction is executed. The actual writing of data should be delayed for 39 milliseconds, which is the record advance delay, to allow the tape to advance 0.3 inches, thereby physically moving the record location forward on tape. This will also increase the inter-record gap by 0.3 inches. After the record advance delay, the write operation is the same as a normal write. Each time the rewrite is attempted on the same record, the record will be moved by 0.3 inches; therefore, at some point, the incorrect area of the tape should be passed over.

The normal inter-record gap is 1.1 inches, which allows 640 records to be recorded on a cartridge. There is a programming method by which the inter-record gap can be reduced to 0.43 inches, thereby expanding the capacity of the tape cartridge to 900 records. The method is to move the tape backward after the write and read check sequences. This procedure will reduce the size of the inter-record gap. After the read check, the tape is stopped and the stop delay executed; then, the tape is moved in reverse for a duration of 63 milliseconds. After the 63-millisecond interval has elapsed, a tape stop instruction and a tape stop delay of 35 milliseconds are executed. At this point, the tape has been repositioned and is in position for writing the next record, with an inter-record gap of 0.43 inches.

A Read Operation.

Reading is ordinarily done with the tape moving forward. In this case, the processor uses the IOC,C#0;001 (start-forward-at-normal-speed-without-erase) instruction to start the tape moving forward at 10 ips. If the IOC,C#0;003 (start-tape-reverse-at-normal-speed) instruction is given, the motion control circuits move the tape in reverse at 10 ips. Erasing is automatically prevented when the tape is moving backward.

After the start forward normal instruction is given, a delay of 35 milliseconds is introduced to allow tape to reach operating speed. Then an IOC,C#0;011 (start read) instruction is given. The data read from tape is examined. The third byte of the preamble is detected, each byte read is loaded into a buffer register, and the processor is notified that a byte is available. The first byte handled in this manner will be the first byte of the label field; the preamble bytes are not sent to the processor.

Before the read operation is started, a memory buffer area and CRC location should be established. The new CRC is developed as each byte is read from tape and, at the end of the record, the new CRC is compared to the CRC from the tape. The IOC,C#0;007 (transfer byte) instruction is executed. This instruction waits until the tape unit responds with a data byte, then transfers the byte into the accumulator. From there, the data can be moved into the memory buffer area. This sequence is repeated until the complete record of data and the CRC have been read.

Upon completing the read sequence, an IOC,C#0;016 (read status) instruction loads the status of the currently selected tape drive into the accumulator and also clears the read operation. After the status word is stored, an IOC,C#0;005 (stop tape) instruction is executed to stop tape motion. At this point, a 35-millisecond delay is introduced to allow sufficient time for tape to stop. This delay is necessary if another tape operation is to be entered. It is a good practice to always program the delay after the tape operation.

In the event that an error has occurred, a reread of the record is in order. After the tape has stopped, the record in error will be physically positioned just past the read head. The tape must be repositioned before the record is read again. To move the tape backward, an IOC,C#0;003 (reverse slow) instruction is executed. This instruction causes tape to move in reverse at normal speed. A delay of 239 milliseconds will allow tape to move to a position that places the record before the read head. At the end of the delay, the IOC,C#0;005 (stop tape) instruction is executed to stop the tape. Again, a 35-millisecond delay is used to allow the tape to stop before the next read operation.

Status Byte.

As mentioned earlier, the status byte is loaded into the accumulator by the execution of the load status instruction. After the status byte has been loaded into the accumulator, a series of tests can be made to determine the condition of the tape drive. The format of the status byte is shown in figure 3-20.

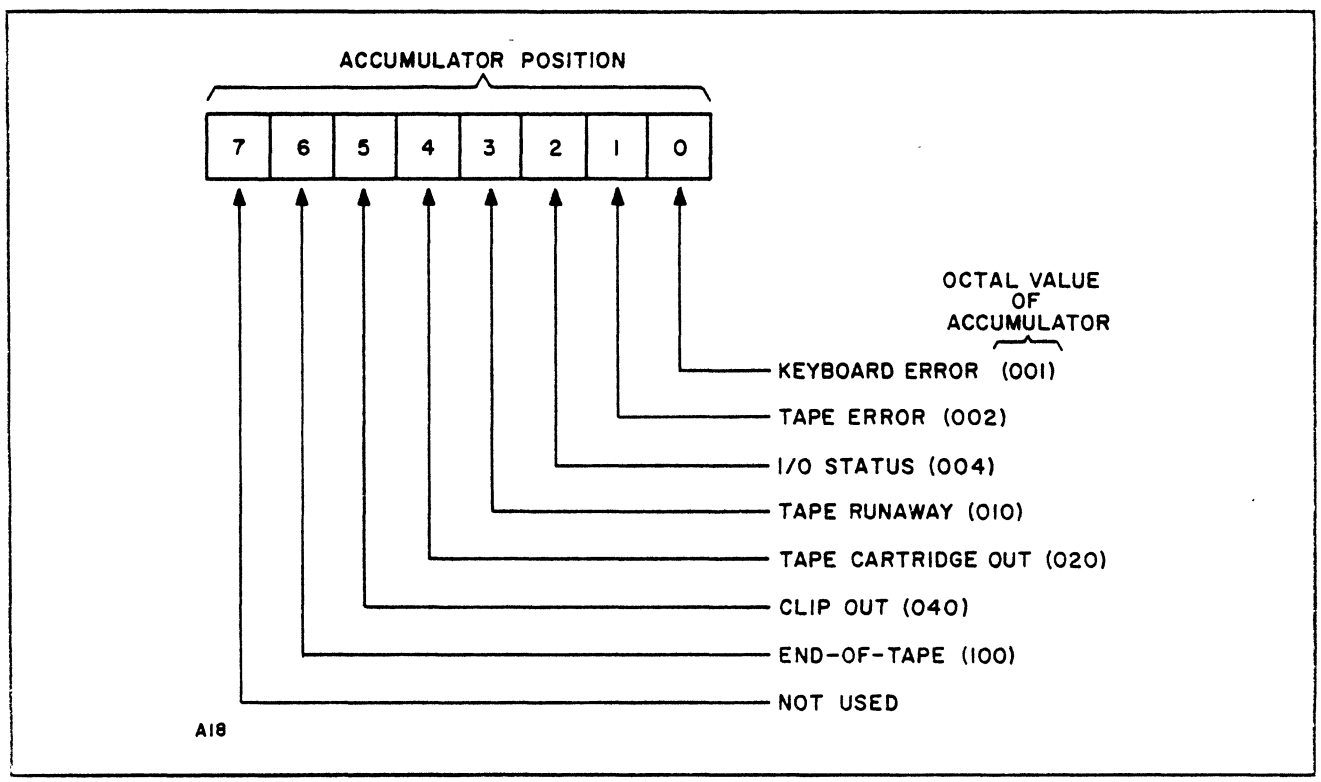


Figure 3-20. Format of Tape Status Byte

Five of the status bits apply to the cartridge tape; bit 7 is not used, and bits 0 and 2 apply to other units. The meaning of each bit is discussed below. In each case, the "1" state indicates the condition listed.

a. Bit 6 indicates that the end-of-tape reflective strip has been detected.

b. Bit 5 indicates that the tape is moved out, or away from the rewound position. If this bit is a zero, the tape is fully rewound.

c. Bit 4 indicates that the tape cartridge is not in place. If this bit is a zero, the cartridge is in place; however, this bit does not indicate whether or not the locking lever is closed.

d. Bit 3, tape runaway, indicates three possible conditions:

   (1) During a normal read operation, no energy is detected for 3 seconds.

   (2) During a search operation, energy is detected for a duration of longer than 50 milliseconds.

   (3) On a write operation, the write circuit is not functioning properly and no data is being written on the tape. The write enable pin being out can cause this error.

e. Bit 1, tape error, indicates three possible conditions:

   (1) On a read-after-write check sequence, an energy dropout is detected.

   (2) A phase error has occurred on a normal read or read-after-write check.

   (3) A tape runaway error, which is indicated by bit 3, has occurred.

High-Speed Search.

Since the records appear in serial form on magnetic tapes, there are times when it is desirable to move the tape at high speed while searching for a specific record. When searching, the tape can be moved at 40 inches per second, which is four times faster than normal.

There are two methods of tape search: (1) the record count method, and (2) end-of-text or file mark method.

The record count method counts the number of records that pass the read head while moving tape at high speed, in either forward or reverse. Therefore, the program must be capable of determining the record number of the record required.

The one weakness in this method of searching is that a bad spot may generate a false record indication. This condition, if not compensated for, could result in locating the wrong record. Normally, the approach to this problem is to establish a search record count that is less than the desired record, and then to read at low speed until the desired record is located.

To start the search routine, the first record should be read with the first record read procedure, which is covered in the read operation description given earlier. At this point, the tape is positioned at the end of the first record. The file's identification can be stored and the number of blocks the tape is to be moved established for the search operation.

The next step is to execute a high-speed-forward (IOC,C#0;002) instruction. This instruction causes tape to move at 40 inches per second. The tape should be allowed to move for 100 milliseconds to ensure that it is up to speed and into the next record before the set read instruction (IOC,C#0;011) is executed.

The next instruction executed is a transfer byte instruction (IOC,C#0;007). This causes the program to stall, waiting for a sprocket signal from the tape control circuits to signal the presence of a byte. In the search mode, however, a sprocket is generated only at the end of the block. Then, only a single sprocket is generated. This releases the stall condition, which indicates that another record has been read, and allows the program to continue.

At this point, the status byte should be loaded into the accumulator (IOC,C#0;016) and tested for a tape runaway (bit 3 is a one) condition. A runaway condition would indicate an end-of-file or end-of-text mark. If there is no runaway condition, the record search count is decremented and tested for zero. If the search is to continue, a delay of 20 milliseconds is introduced to allow the inter-block gap to pass by the read head. Then, the read subroutine is re-entered to count the next record.

This sequence is continued until the record count is decremented to zero. At that point, the tape is stopped with an IOC,C#0;005 instruction, and a delay of 100 milliseconds is initiated to allow the tape to stop. The next step is to read at normal speed so the records can be read and the labels examined to determine how many records, forward or reverse, the tape must be moved to locate the desired record.

The end-of-text method of searching requires that the tape be written with an end-of-text record at the end of each group of contiguous records. The end-of-text record should be about 6 inches long, but it requires no particular data format. The searching is achieved by moving tape at high speed, forward or reverse, until the end-of-text record is reached. End-of-text records are shown in figure 3-21.

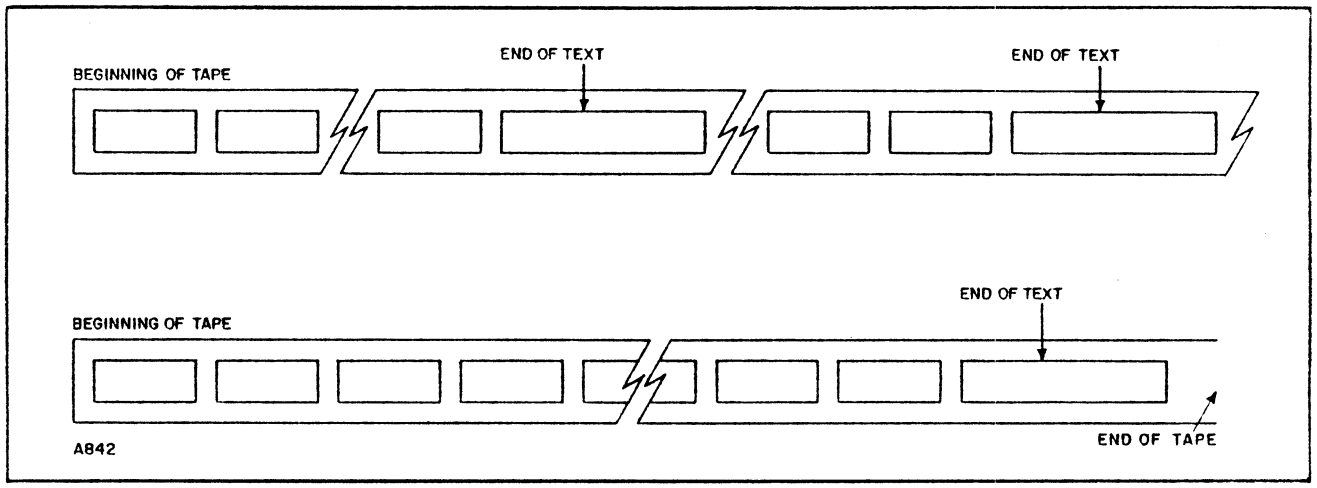


Figure 3-21. Use of the End-of-Text Records

The tape control circuits are designed so that if 50 milliseconds elapse after the read operation is started and energy is still being detected (i.e., data is being read), a tape runaway error is generated. The error will stop the read operation. Therefore, in an end-of-text search, a read loop will detect each record, as in the record count routine; however, when a long record is being read, a tape runaway error will be generated, which clears the read and run commands.

At this point, the tape is positioned with the read head somewhere past the center of the long record. By testing bit 3 of the status word, the program can determine that the tape is in this position. Next, the tape must be positioned to read the next good record. This is achieved with normal forward, start read delay, and a read tape instruction. Again, a design feature will prevent the reading of data until the energy being detected from the long record drops off and new energy is detected, indicating a new record. This record can be checked, if desired, or the high-speed search can be executed again to look for the next end-of-text record.

Before the search sequence is discussed, a brief explanation of the procedure to write the long end-of-text record is required. This procedure must be followed when creating a tape which consists of more than one file. The data on the tape would appear as shown in figure 3-21. To write this long block of data, a start tape forward, normal (IOC,C#0;000) instruction is executed first; then a 35-millisecond delay allows the tape to get up to speed. Next, a Set Write Mode instruction (IOC,C#0;010) is executed, followed by a Transfer Byte instruction (IOC,C#0;007). The data in the accumulator is not important.

After the first character is written on the tape, the sprocket signal will release the stall resulting from the Transfer Byte instruction. The writing will continue, even though the Transfer Byte instruction is not executed again. Therefore, a delay of 600 milliseconds is introduced after the Transfer Byte instruction in order to continue writing the long block. The 600-millisecond delay will allow the tape write to continue until a block six inches long has been written. At the end of the delay, the Set Read Mode instruction (IOC,C#0;001) is used to clear the write mode, and the Load Status instruction (IOC,C#0;016), in turn, is used to clear the read mode and return status. The inter-block delay is also introduced before the tape is stopped.

Before a search sequence is entered, the tape must be moved away from the rewound position by a normal first block read operation. A forward at high speed instruction (IOC,C#0;002), followed by a 100-millisecond start delay to ensure the tape is up to speed, then begins the search sequence.

The set read mode instruction (IOC,C#0;011) then enables the read circuits. Again, the end of each record generates a sprocket, which may or may not be used by the program in this application. However, the transfer byte (IOC,C#0;007) instruction will stall the program until the sprocket is received. The next instruction should be a load status instruction (IOC,C#0;016), which will load the status byte into the accumulator and clear the read mode. A delay of 20 milliseconds (inter-block gap) should be introduced before this sequence is repeated. This sequence is continued until the long block is read; then, a runaway error is generated, which stops the tape motion. The program detects this condition by reading the status byte, bit 3.

When the tape stops, the read head is positioned just past the center of the block. A start tape forward, normal instruction (IOC,C#0;000), followed by a 35-millisecond start delay and a read instruction (IOC,C#0;011) will condition the circuits to read the next record. Reading will not start until the first data record past the long block is read. At the end of the read operation, the tape should be stopped by execution of a Stop Tape instruction (IOC,C#0;005). The search can then be continued by going back to the Start Tape Forward at High Speed (IOC,C#0;002) instruction.

Programming Examples.

The following programming examples, which are shown in figures 3-21A, B, and C, are intended to show the reader some of the methods used to program the cartridge tape. Included are examples that use the background routines. Note that these examples are not complete routines, but only segments of routines.

```
LAB: CMD  OPERANDS       COMMENTS

TAPE WRITE ROUTINE
$11: LDA, R#0; OCT:002. SET DRIVE 2
     SMS, S#0.
     SBU, WRF.          WRITE RECORD
     BRH, *+04.
     EXU, 000.
     LDA, R#0; R#4.     STATUS CK
     TMJ, +16; OCT:040. CLIP OUT
     IOC, C#3; 013.     BEEP
     IOC, C#3; 207.     SKIP ON BUSY
     BRU, *+004.
     BRU, *-06.
     TLJ, $11; OCT:032. ERROR KEY
     TLJ, $11; OCT:232. CTRL/ERROR KEY
     BRU, *-012.        TRY AGAIN
WRITE ERROR HALT
     LDX, R#1; ER4.     SET MESSAGE
ERX: SBU, CLS.          CLEAR DSP
     LDA, I#1; ER1.
     TLJ, +06; OCT:377. END?
     STA, I#2; SBF.
     BRU, *-06.
     LDX, R#2; SBF+064.
     LDX, R#1; RST.
     LDA, I#1; RST.
     TLJ, +06; OCT:377.
     STA, I#2; SBF.
     BRU, *-06.
     IOC, C#4; 033.     ENABLE DISPLAY
     IOC, C#1; 012.     REWIND DECK 1
     IOC, C#2; 012.     REWIND DECK 2
     BRU, *+000.        ERROR ABORT

B1
```

Figure 3-21A. Tape Write Sample Program

```
LAB: CMD  OPERANDS       COMMENTS

EIGHT RETRY RECOVERY ROUTINE
D45: TMJ, -06; OCT:010. CHECK RUNAWAY
     SBU, D84.          TAPE BACKSPACE
     IOC, C#0; 001.     TAPE FORWARD
     SBU, D91.          35MS DELAY
     IOC, C#0; 011.     SET READ MODE
     IOC, C#0; 007.     TRANSFER BYTE
     IOC, C#0; 016.     RESET READ MOD
     LDX, R#3; OCT:000. CLR/RETRY/COUN
     LDA, R#0; OCT:000. 000 TO ACC.
     SBU, D10+06.
     ADX, R#4; OCT:001. INSERT TE.BIT
     BRH, D44.          BUMP SEQ/NO.&R
     BRU, RDF+06.       CONTINUE

B2
```

Figure 3-21B. Retry Recovery Sample Program

```
HIGH SPEED MECHANICS
HSF: IOC, C#2; 002.      TAPE H.S. FORW
     LDA, R#0; DIF.
     SUA, R#0; OCT:005.
     BRU, *+08.
HSR: IOC, C#2; 004.      H.S. REVERSE
     LDA, R#0; DIF.
     ADA, R#0; OCT:005.
     STA, R#0; R#5.
     LDX, R#1; DEC:035.
     SMS, D93.
     SBU, D93.
HSS: IOC, C#2; 011.      SET READ MODE
     LDX, R#2; 007.      STALL FOR BLOC
     LDX, R#1; DEC:007.
     SMS, D93.
     SBU, D93.
     SUX, R#5; OCT:001.
     CPX, R#5; OCT:000. CHECK COUNTER
     BRH, HSS.
     IOC, C#2; 005.      STOP TAPE
     LDX, R#1; DEC:035.
     SMS, D93.
     BRU, D93.           DELAY/EXIT

B3
```

Figure 3-21C. High-Speed Move Sample Program

3-9. SERIAL INPUT/OUTPUT CHANNEL.

All peripheral equipment except the disc operates on the serial I/O channel, which is a single coaxial cable terminated at both ends. Up to 64 units can communicate with one another over this link, with the intelligent terminal acting as master and the peripheral units, such as printers and tape units, acting as slaves.

Devices are connected to the SIO cable by BNC "T" connectors. They are interconnected by attaching the coaxial cable from one device to the next in a continuous chain. The cable is terminated at each end of the chain by using the terminating plug, which is attached to each unit near the SIO connector.

The processor initiates the operation by sending an I/O instruction to the I/O controller. In turn, the controller begins transmission on the line, addresses the selected unit (each unit has an address selection switch or plug), and sends commands as required to execute the instruction from the processor. The selected peripheral unit responds by returning its address, to verify that the link is established, and the status indications necessary to accomplish the exchange between it and the intelligent terminal.

Equipment connected to the SIO cable may be turned off when not in use without affecting other devices utilizing the cable. Units such as the Model 1501 Intelligent Terminal may also be operated with a non-I/O program without affecting or being affected by activity on the SIO cable.

As mentioned above, the peripheral units always operate in the slave mode; that is, they never initiate a data exchange and operate only in response to commands from a master. It was also stated that a terminal operates in the master mode. While this is usually the case, a terminal can also be programmed to operate in the slave mode and respond to a master terminal.

Only a unit operating in the master mode may initiate transmission. An address sequence, which is shown in figure 3-22, is used to establish communication between two units. The address sequence consists of an initial transmission of start and idle codes by the terminal operating in the master mode. This synchronizes all receivers connected to the line. The idle codes are followed by the transmission of the address of the slave device with which communication is to be established.

The slave device that recognizes its address is set to an active state. Once in an active state, the cessation of transmission by the master causes the slave to transmit the start and idle code, echo its address, and send its status. The transmission of a status byte by a slave device automatically causes the slave transmission to stop.

As the address echo is received by the master, it is compared to the address that was transmitted to ensure that communication with the proper device has been established. The status byte is transferred to the master terminal's accumulator for use by the program. The cessation of transmission by the slave causes the master to resume transmission, starting with start and idle codes in preparation for the transmission of commands and/or data.

The two stations automatically follow each other with idle code transmission until a Stop Transmit instruction is used at the master. After this, the master does not maintain synchronization after the next response from the slave. A pre-determined interval then elapses, and the communication link is de-activated.

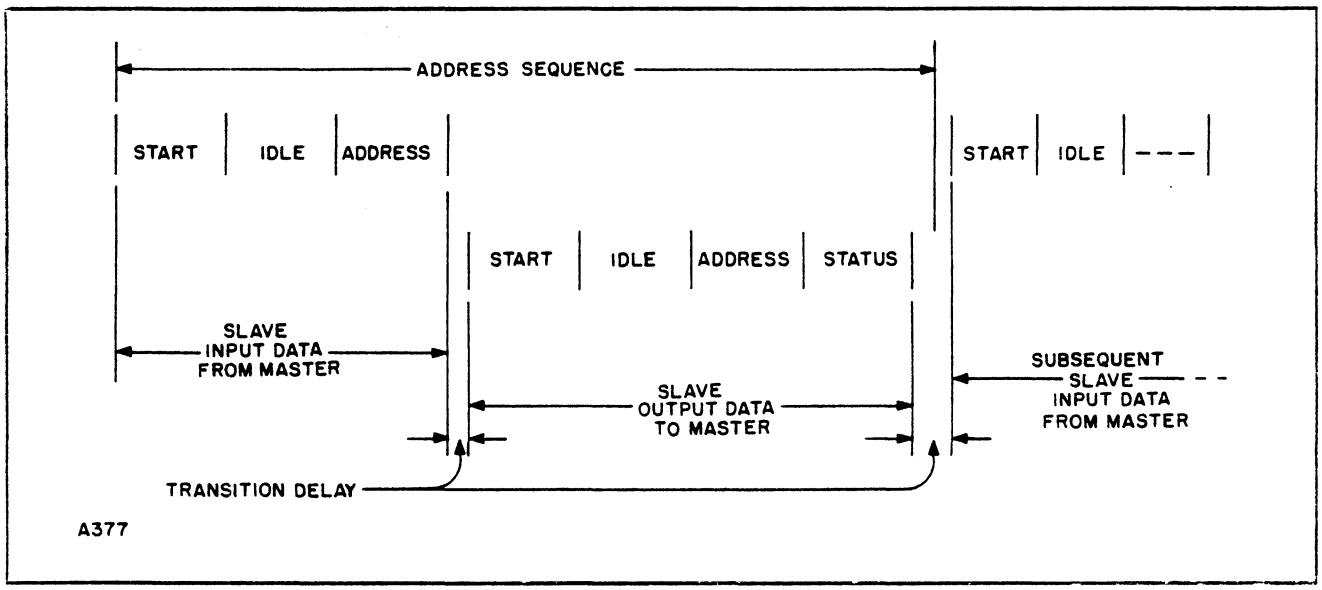


Figure 3-22. Address Sequence

Figure 3-23 shows an exchange between an intelligent terminal and a Model 1511 Magnetic Tape Unit which, of course, is operating in the slave mode. The number of transmissions in a group depends on the operation. The significant points of this communication technique are:

a. One of the devices is always transmitting except for a transition time of approximately one microsecond.

b. A device transmits idle bytes when not transmitting an information byte.

c. A device stops transmitting at the end of its sequence.

d. A device starts transmitting as a result of detecting that the other device has stopped transmitting.

Organization of Data on the SIO Line.

Six types of bytes can be transmitted over the SIO channel. They are: data bytes, command bytes, address bytes, status bytes, start bytes, and an idle byte used for synchronization. Each byte transmitted is supplemented by a hardware-generated ninth bit to distinguish between data and the other types of information. The ninth bit is zero for start and data bytes, and is a one for all other types of bytes. Figure 3-24 shows the format of all six types of bytes.

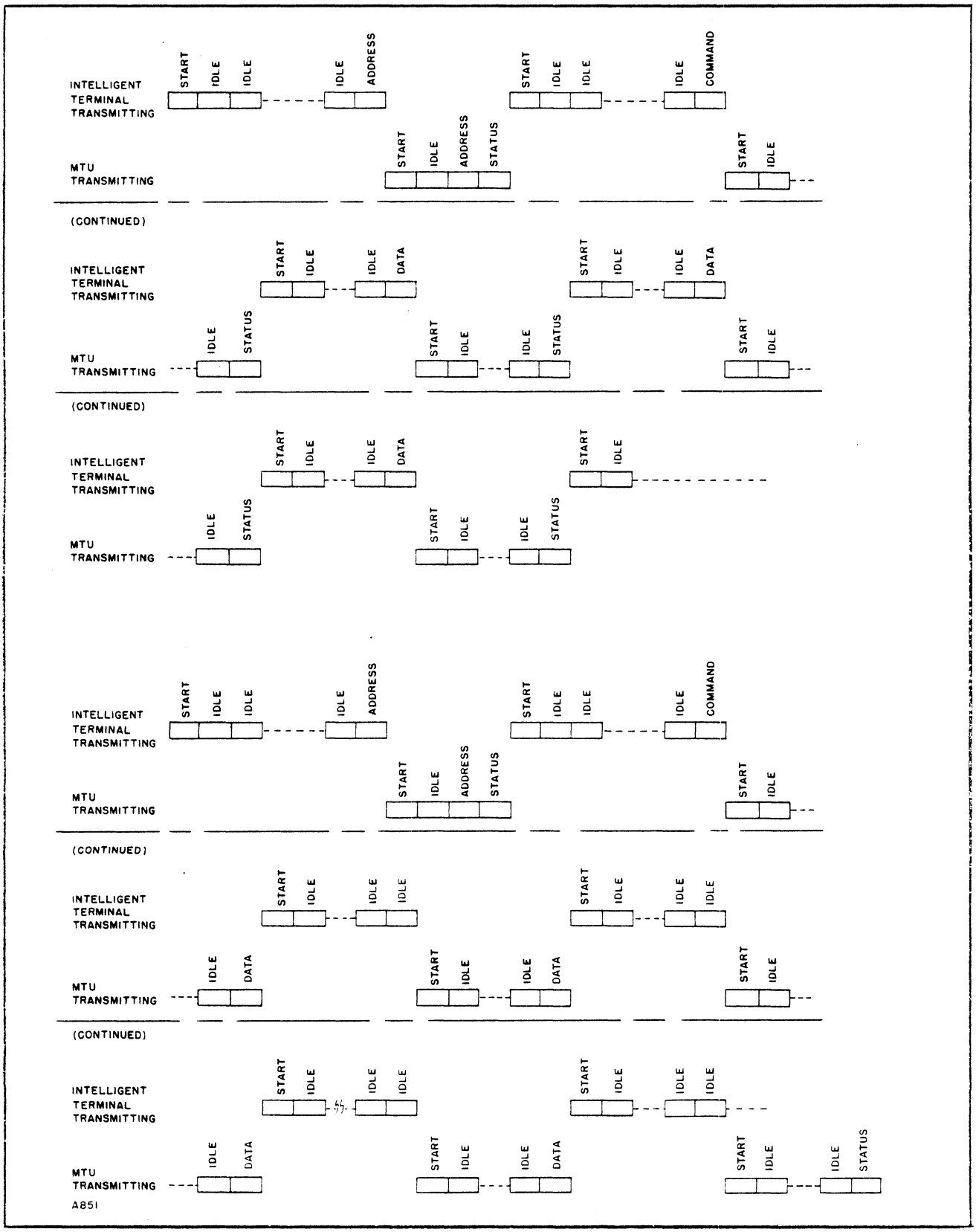


Figure 3-23. Typical Exchange Between an Intelligent Terminal (Master) and Magnetic Tape Unit (Slave)

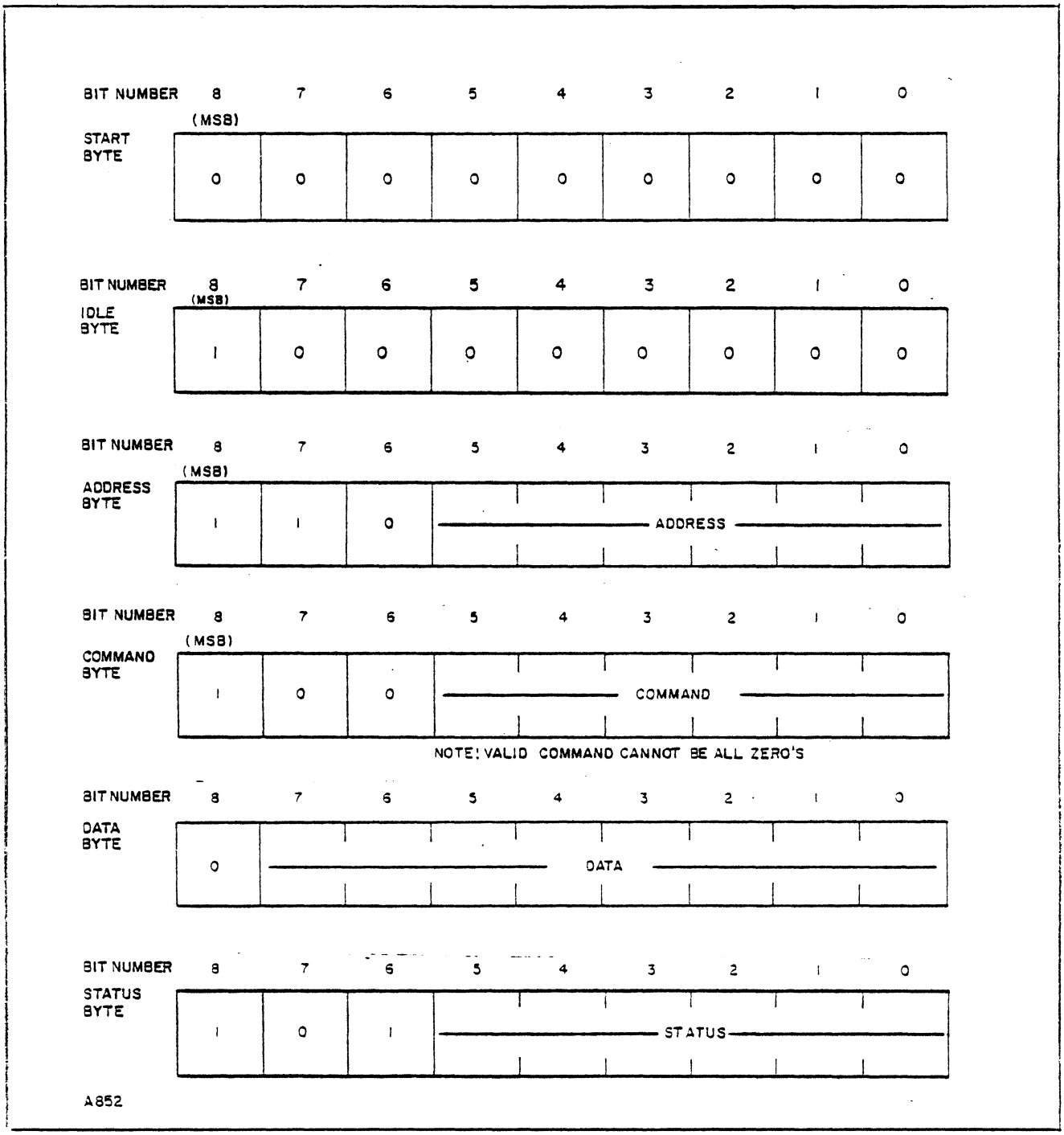


Figure 3-24. Serial I/O Channel Byte Formats

The content and use of each type of byte is as follows:

a. Start Byte. The start byte is used together with an idle byte for synchronization and timing. It is always the first byte of a transmission sequence, and is not used in any other way. The start byte contains all zeros.

b. Idle Byte. The idle byte is used for synchronization and timing. It is used together with a start byte at the beginning of a transmission sequence and it is used after that to maintain the synchronization between the master and slave units when there is no other data being transmitted.

c. Status Byte. A status byte has a one in positions 8 and 6, and a zero in position 7. The status byte indicates any of several different conditions, one for each bit. It is always transmitted by the slave as the last byte of the transmission sequence.

d. Address Byte. An address byte has ones in bit positions 7 and 8, and a zero in bit position 6. The remaining six bits specify an address. Thus, up to 64 addressable devices may be placed on a single line.

e. Command Byte. A command byte is characterized by a one in bit 8 and zeros in bits 7 and 6. The remaining six bits specify a particular command.

f. Data Byte. The data byte is characterized by a zero in bit position 8. The remaining bits carry the data.

SIO Channel Instructions.

The intelligent terminals control the SIO channel with the instructions in the "175" family that are listed in table 3-3 below. A source statement provided to the Assembler System would take the following form:

```
IOC,C#5;NNN
```

Where: the "IOC" specifies input/output and thus the 17 code. "C#5" is decoded to mean channel number 5, converting the statement to 175, and "NNN" specifies the function to be performed.

Table 3-3. SIO Channel Instructions

Statement	Machine Instruction Code	Instruction
IOC,C#5;000	175-000	Start Transmit
IOC,C#5;001	175-001	Transfer Byte, Unconditional
IOC,C#5;201	175-201	Transfer Byte, Skip on Busy
IOC,C#5;002	175-002	Transmit Data Byte
IOC,C#5;003	175-003	Transmit Control Byte
IOC,C#5;004	175-004	Stop Transmit*
IOC,C#5;006	175-006	Inhibit Line*
IOC,C#5;007	175-007	Set Device Address
IOC,C#5;010	175-010	Set Master Mode
IOC,C#5;011	175-011	Set Slave Mode

*Executed by Master Only

Each of the Serial I/O channel instructions is described in the following paragraphs. Action taken by each instruction is as follows:

a. The Start Transmit instruction causes an intelligent terminal operating in the master mode to initiate transmission. This instruction also releases the line if an Inhibit Line instruction has been previously executed.

b. The Transfer Byte instruction is used to move a byte from the SIO channel into the accumulator. It performs several functions, depending upon which part of an SIO communication is in progress when the instruction is executed. Normally, the Transfer Byte instruction causes the processor to stall if the SIO control circuits have not sent a sprocket signal indicating that a byte is available.

   If the intelligent terminal is operating in the master mode and transmission is in progress during which at least one Transmit Data Byte or Transmit Control Byte instruction has been executed, the Transfer Byte instruction causes the transmission to stop. In this case, the active slave should respond.

If the terminal is in the slave mode, the Transfer Byte instruction has no effect on the transmission, but causes the processor to stall until a sprocket signal is received. If the SIO channel is not active but the terminal is in the master mode, the Transfer Byte instruction causes the generation of a dummy sprocket signal.

In any case, once a sprocket has been sent, the Transfer Byte instruction causes the contents of the SIO output data register to be transferred to the accumulator, and the processor leaves the stall condition, allowing further instructions to be executed.

The skip-on-busy feature is used with the Transfer Byte instruction to avoid the stall condition any time the programmer knows that the reply from the slave will not occur for a considerable period of time. Since the Transfer Byte instruction is used to initiate the handover of transmission from the master to the slave, there is always an interval of about 20 microseconds between the time the Transfer Byte instruction is first executed and when a sprocket signal may occur. The occurrence of additional sprocket signals is dependent upon the particular program or devices being utilized.

c. A Transmit Data Byte instruction causes the contents of the accumulator to be transferred to the serial I/O channel circuits and transmitted on the I/O channel.

d. A Transmit Control Byte performs the same function as the Transmit Data Byte instruction, but the SIO circuits distinguish between the two instruction types by coding the ninth bit differently.

e. A Stop Transmit instruction is used by the master to terminate a communication exchange after the next transmission of status by the slave. Stop Transmit may be executed at any time during the final transmission of the master.

f. When an Inhibit Line instruction is executed by a master, the SIO line is driven to a high level and held at that state until a Start Transmit instruction is executed. After execution of the Inhibit Line instruction, every device connected to the cable is cleared by timing circuits that expire due to lack of further transmission line activity. A period of at least 15 microseconds must be allowed to elapse between the Inhibit Line instruction and the Start Transmit instruction.

g. The Set Device Address instruction transfers the accumulator contents to the address register and input data register of the SIO channel circuits, and it should be executed before the Set Slave instruction is used. Both slave and master must have the address of the slave stored in its address register.

h. The Set Master instruction conditions an intelligent terminal to function as the controlling station in an SIO exchange. Only the station that is in master mode can execute the Inhibit Line, Start Transmit, and Stop Transmit instructions.

i. The Set Slave instruction inhibits SIO circuits from responding if Inhibit Line, Start Transmit, or Stop Transmit instructions are executed, and conditions the station to operate as a slave device. Execution of this instruction also transfers the accumulator contents to the input data register of the SIO circuits in the I/O controller.

SIO Channel Status Byte.

As shown earlier in this description, bits 0 through 5 of the status byte sent to the terminal are available for status indications supplied by the active peripheral unit. The use and meaning of each of these bits depends upon the peripheral unit. For example, bit 2 from the Model 1511-1514 tape drives is the "tape at load point" indication, while the same bit from the printers means "printer error". The function of the status byte is to inform the program of the conditions that exist in the I/O device; that is, the availability of the device, errors that may have occurred on last device operation, etc.

The status byte is presented to the terminal at different intervals, depending upon the type of function and device. For example: on a tape write function, status is presented after each byte is received by the tape drive; however, on the tape read operation, the status is presented at the end of the read operation.

When any information is received over the serial I/O channel by the I/O controller section of the terminal, the high-order bits are checked to determine if the information is a status byte, address byte, data byte, or idle byte. In the case of a status byte, the following sequence of events takes place:

a. Bits 6, 7, and 8 are removed from the status word. They inform the terminal as to which type of byte has been received. Assuming that the byte received is a status byte, the I/O controller prepares a status byte for transfer to the processor. Bit positions 6 and 7 of the status byte sent to the processor indicate the address error and I/O error condition, respectively. These conditions are inserted into the status byte by the I/O controller to indicate unsuccessful address comparison and that the data on the line is invalid.

b. The status byte is transferred to the accumulator when the Transfer Byte (IOC,C#5;001) instruction is executed. This information is temporarily stored in memory. At this point, the program must make a test to determine if the information in the accumulator is an SIO channel status byte. This is done by reading the general status byte and examining bit 2, I/O status; this is an octal 004 in the accumulator, as shown in figure 3-20.

c. The I/O controller, which examines bits 6, 7, and 8 of the status byte from the peripheral unit, placed a one in bit position 2 of the general status byte. However, the general status byte (as opposed to device status) is only available on request; that is, the execution of a Load Status (IOC,C#0;016) instruction. This instruction loads the general status byte into the accumulator. Note that the device status, which was in the accumulator, should be stored in memory prior to execution of the Load Status instruction.

d. The general status byte now in the accumulator is tested for the presence of a one in bit position 2. A one indicates that a device status byte had previously been in the accumulator and must be returned to the accumulator and tested to determine the condition of the peripheral unit.

Sequence of Operation in the Master Mode.

When an intelligent terminal is to be operated in the master mode for use of the SIO channel, it must be programmed with the following considerations in mind:

a. The Set Master Mode instruction (IOC,C#5;010) is executed first. This instruction puts the terminal in the master mode, which allows it to have control over the SIO channel. All other units on the channel will be slaves.

b. Next, the Inhibit Line instruction (IOC,C#5;006) is executed. This clears all units and stops any transmission in progress. Note that a period of at least 15 microseconds must elapse between an Inhibit Line instruction and a Start Transmit instruction.

c. At this point, it is necessary to set into the address register the address of the slave unit with which contact is to be established. This is achieved by two instructions. The first is an LDA instruction. The address has the octal format 2NN, where the "2" forms the bits 6 and 7 of the SIO address byte code (refer to figure 3-24) and the NN represents the address (00 to 77) of the selected peripheral unit.

d. After the address is loaded into the accumulator, a Set Device Address instruction (IOC,C#5;007) is executed. This instruction transfers the address from the accumulator to the address register in the I/O controller.

e. The next instruction to be executed is the Transmit Control Byte instruction (IOC,C#5;003). This instruction causes the device address to be transmitted over the SIO line. The slave should be waiting for its address or have a not ready status waiting to be sent to the master at this point.

f. Next, the master waits for the status byte to be returned from the slave to determine the course of action. Normally, a Transfer Byte instruction (IOC,C#5;001) is executed to cause the waiting. This stalls the master terminal until the slave returns its status and address. If the selected slave returns a status byte indicating that it is not active, the I/O controller in the master terminal generates a dummy sprocket to release the stall condition in the master.

g. If the status byte returned by the slave indicates that the slave is ready, the master executes a Transmit Control Byte instruction (IOC,C#5;003) after having previously established the control byte in the accumulator. This byte directs the slave, and the byte contents depend upon the unit involved.

h. The next sequence of instructions establish the parameters to send data or receive data. Data may be transferred in either a burst mode or in a single character mode of communication. In the burst mode, the sending unit transmits continuously with the data interspersed between idle codes. Note that the programmer does not supply the idle codes. In using the burst mode of transmission, the characteristics of the devices involved must be considered so that the ability of the receiving unit to assimilate the data is not exceeded. In a single character mode of data transfer, each character is acknowledged by a transmission from the receiving unit. This mode of data transfer is normally used when the time of arrival of each character is critical. In all cases, the transmission of a status byte by a slave causes its transmission to terminate.

Sequence of Operation in the Slave Mode.

When power is turned on to the Series 1500 Intelligent Terminals, the SIO channel is set to the master mode by the general clear signal. In addition, the general clear signal selects the primary SIO channel if the terminal is equipped with dual channels. This assures that a newly activated terminal will not interfere with SIO activity from the master since, if good practices are followed, the communications to the master will be over the slave's secondary channel.

In order to employ a terminal as a slave, it is first necessary to establish certain conditions that will allow the SIO section to automatically respond to a transmission from the master station. Preparation for a link-up with the master is accomplished by the following sequence of instructions:

a. The accumulator is loaded with the terminal address by the Load Terminal ID (TRM) instruction. This instruction takes the address of the terminal from the address plug and loads it into the accumulator.

b. The Set Device Address (IOC,C#5;007) instruction is used to move the terminal ID to the address and input data registers in the I/O controller.

c. The accumulator is loaded with a status word; an LDA instruction is used. This status word is not a hardware status in the sense of a status word returned from a tape unit. However, it does inform the master as to the condition of the slave terminal; that is, what sort of task the slave is requesting from the master.

d. The next instruction should be a Set Slave Mode (IOC,C#5;011). This instruction will ensure that the unit is in slave mode and will transfer the status word from the accumulator to the input data register in the I/O controller.

e. The next instruction is a Transfer Byte (IOC,C#5;001) instruction, which will stall the processor until it is addressed by the master.

When the master sends the address of this slave over the SIO cable, the slave I/O controller will respond by sending its address and status to the master. Assuming no error occurs on the line, the master and slave would then be linked up, and communication can proceed.

At some point, the slave will finish the task and no longer require the services of the master terminal. When communication is complete, the slave should be programmed to terminate the transmission by executing a Transmit Control Byte instruction to send a completion status to the master.

There is more than one programming technique that can be employed when operating in the master/slave mode. If, for example, the programmer does not wish to stall the slave until the master goes into a scan sequence, a Skip on Busy instruction could be used in step "e" above.

The master program can be made to recognize a not ready status in the slave. In this case, the slave can set the address and the status words in its address register and input data register; then, upon polling, the master will receive the not ready status. Since the addressing sequence is strictly an I/O function, it will not effect the operation of the program in the slave.

Dual Serial Input/Output Channels.

One of the options available for the Model 1501 Intelligent Terminal is the dual serial I/O channel feature. This option allows the terminal to operate two standard input/output (SIO) cables rather than the standard single cable. However, there is only a single I/O controller section; therefore, control is switched between the two SIO cables.

Two input/output instructions are used to switch between the two channels. These channels are referred to as channel 1, the primary channel, and channel 2, the secondary channel. The IOC,C#3;006 (Select Primary SIO Channel) instruction is used to select channel 1, and the IOC,C#3;014 (Select Secondary SIO Channel instruction) is used to select channel 2. When a terminal is first turned on, it is automatically set to the master mode on the primary channel.

When switching channels, the following practices should be followed:

a. Execute the Set Slave Mode (IOC,C#5;001) instruction before the Select Primary Channel or Select Secondary Channel instruction (IOC,C#3;006 or IOC,C#3;014).

b. Then, execute the select primary or secondary instructions before the Set Master Mode (IOC,C#5;010) instruction.

c. Before switching from master mode on one channel to slave mode on the other channel, the Inhibit Line instruction (IOC,C#5;006) or Stop Transmit instruction (IOC,C#5;004) should be executed to terminate the line activity on the master channel.

3-10. COMMUNICATIONS ADAPTERS.

Some applications of the Series 1500 Intelligent Terminals require the exchange of data with other data processing systems that are located far from the terminal. When data is to be transferred over distances that make it impractical to use the serial I/O line, it becomes necessary to use the facilities of the communications common carriers. The most frequently used communication facility is the voice-grade telephone line. These lines can be used to transmit data provided there is equipment at both ends of the system to make the lines compatible with data processing equipment.

The data to be communicated is usually available from computers at high rates but the communication lines can transfer the data at relatively low speeds. In addition, the form of the data must be changed in order to remain within the limitations of the communications lines. Two units are required to overcome these problems. The first is a communications adapter that is added to an intelligent terminal, and the second is a MODEM (modulator-demodulator) that converts the information to a form that can be sent over communications lines.

Typical Equipment Arrangement.

A typical system is shown in figure 3-25. Here, a Model 1503 is equipped with a communications adapter, which is installed internally and then cabled to a MODEM. The telephone line is arranged so that the operator of the 1503 must dial the number of the remote processing unit, which may or may not be attended. Once the telephone connection is made, the data is transferred and the line is disconnected automatically. Note that the printer is provided in order to produce a record of the communications.

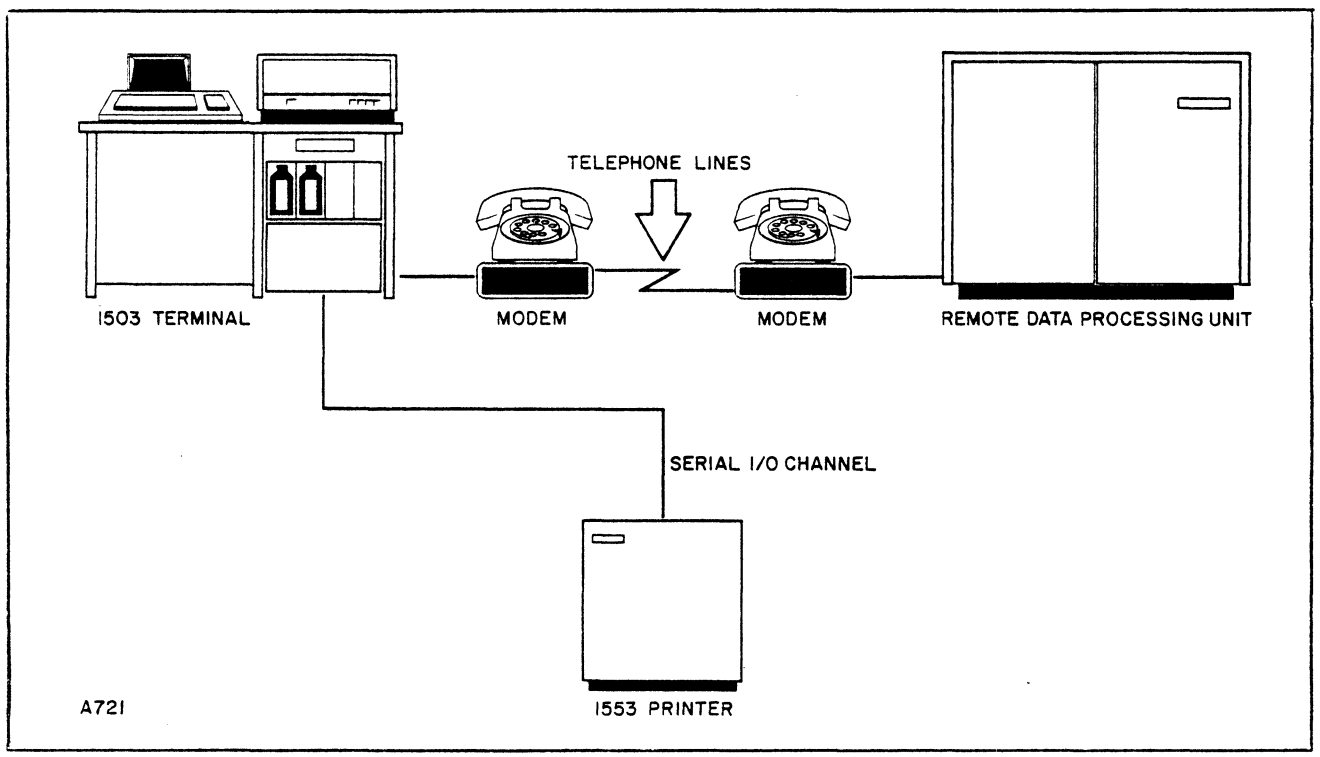


Figure 3-25. Typical Equipment Required

Line Arrangements.

There are three basic arrangements of communications lines between terminals and other processing systems, as follows:

a. Full-duplex, which provides two channels between units. In this case, a unit uses one channel to send and the other to receive.

b. Half-duplex, which provides only one channel between units. This channel is bi-directional but, of course, can carry data in only one direction at any one time. Each time that a unit finishes sending and the other takes its turn sending, the line must be "turned around". This introduces some delay, which is not encountered in the single-direction lines used in the full-duplex configuration.

c. The third case involves using most of the bandwidth of a half-duplex channel for sending in one direction while the remainder is used as a low-speed "reverse" channel. In this case, the reverse channel can carry only a few bits per second and is used to acknowledge messages and perform supervisory duties.

Also to be considered when discussing line arrangements is the possibility of a multi-point connection. (A single point connection is shown in figure 3-25.) In the multi-point arrangement, several terminals are connected to the line simultaneously and a central computer chooses one at a time to transfer data.

Forms of Data Transferred.

There are two basic techniques of data transmission: synchronous and asynchronous. In addition, there are two variations of the synchronous technique used by the Series 1500 communications adapters. The first of these is BSC, which means binary synchronous communications, and the second is SDLC, which means synchronized data link communications. The techniques used by each of the adapters is as follows:

a. Models 1534, 1534-1, and 1534-2: Asynchronous

b. Model 1535: BSC

c. Model 1539: Asynchronous, BSC, and SDLC.

Asynchronous communication is common with electromechanical terminal equipment such as teletypewriters. In this case, each transmission consists of a single character, such as an "A". Added to the character is starting and stopping information and, in some cases, error checking information. The receiving unit recognizes the beginning and ending of each character, and there is no transmission between characters.

In synchronous serial transmission, the data bits are transmitted without start and stop bits. This method saves on the number of bits to be transmitted for each character over the asynchronous method and thus allows data to be transferred at a rate up to 30 percent faster. With synchronous transmission, the sending and receiving units are synchronized by characters at the beginning of a transmission and synchronization is maintained until the unit ends its transmission.

Line Protocol.

Line protocol means the rules that govern the data transfer between units. It determines what control characters must be included in the data transmission, the order in which they appear, the organization of the data blocks, the codes used, etc. Therefore, the programmer must become familiar with the line protocol to be used before beginning to program a communications system.

Transmission Codes.

Data transmitted over a data link is in binary form, with a "mark" representing a one and a space representing a zero. This binary information is organized into groups of bits to form characters. Of course, the two units communicating with one another must be using the same code. In general, the codes used require between 5 and 8 bits per character. The four most commonly used codes are:

a. USASCII, which means United States of America Standard Code for Information Interchange, and is a 7-bit code.

b. Transcode, which is a 6-bit BCD character set.

c. EBCDIC, which means Extended Binary Coded Decimal Interchange Code, and is an 8-bit code.

d. Baudot, which is a 5-bit code commonly used with teletype equipment.

Models of Communications Adapters.

It was mentioned in Section 1 of this manual that there are five models of communications adapters available: the Models 1534, 1534-1, 1534-2, 1535, and 1539. An illustration summarizing the models of adapters appears in figure 3-26. Three of these, the 1534, the 1535, and the 1539 are each a single printed circuit board. The other two are free-standing units that each include the 1534 asynchronous communications board, a SIO (serial input/output) board, and an SIO adapter board. The 1534-2 also includes a level converter which connects directly to teletype equipment.

Relationship to the Intelligent Terminals.

The communications adapters are controlled by the processor program in the intelligent terminal. The processor initiates the communication by sending an I/O instruction to the I/O controller that selects the communication channel for operation. The I/O controller then activates the communications adapter.

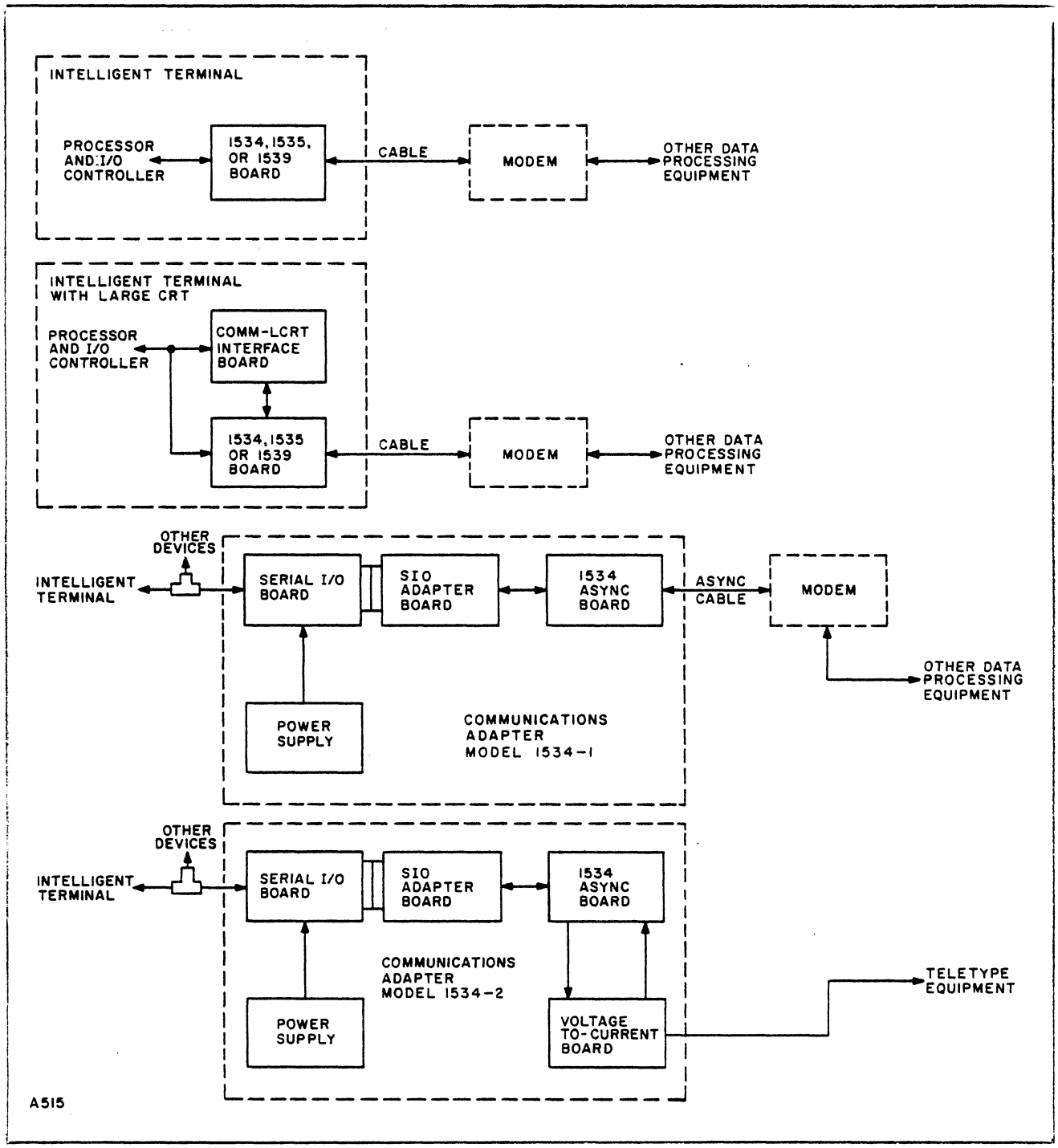


Figure 3-26. Models of Communications Adapters

In those configurations where only one Intelligent Terminal is going to use the communications adapter to operate one communications line, the adapter is mounted in a slot of the main chassis in the Intelligent Terminal. The adapter interfaces directly with the processor and the I/O controller. A special cable is used to connect the board to the MODEM. These adapters are Models 1534, 1535, and 1539.

The internally-installed communications adapters are controlled by instructions in the 176-YNN (IOC,C#6;YNN) format. A "176" selects the communication interface, and the "Y" position chooses the type of adapter. Instructions used are described later in this section.

The terminal supplies timing, commands, and characters; it receives characters and status. Data transfer is on a byte-by-byte basis. Commands are decoded by the communications adapter to recognize what operation is to be performed and to select the operating baud and data format. When these conditions are established, the communications adapter status is examined by the terminal to recognize when the next data byte can be transmitted or received, after which the byte is transferred.

Where more than one Intelligent Terminal is to use the communications adapter or one Intelligent Terminal is to operate more than one adapter, the adapter operates on the serial input/output (SIO) channel. These adapters are Models 1534-1 and 1534-2, which are free-standing units. The Model 1534 board is the basic element of these adapters. The board interfaces with the I/O controller via a serial I/O board and an I/O adapter board. These boards convert the serial information on the SIO channel to the parallel data and the control and timing signals required as inputs to the communications board to operate the board just as if it was plugged into an Intelligent Terminal.

Using the SIO channel, the Intelligent Terminal is a master unit controlling the operation of a slave peripheral unit. Each slave has a unique address, which enables the Intelligent Terminal to establish an exclusive communication path between the terminal and the slave. This path is used for both transmission and reception.

Relationship to the Communications Lines

The 1534, 1534-1, 1535, and 1539 are interconnected with a MODEM of the data communications equipment so that data may be transferred via carrier-type equipment. This requires that control and status signals be exchanged between the adapter and the MODEM to coordinate readiness, baud range, status, and transmit and receive signaling. These signals are established by industry standards.

Two of the standards that apply for this purpose are EIA RS-232-C and CCITT V24. EIA is the Electronic Industries Association and CCITT is the International Consulting Committee for Telephone and Telegraph.

Model 1534 Asynchronous Communications Adapter.

Both this adapter and the Model 1539 can conduct asynchronous communications. One common application for the asynchronous mode is communication with electromechanical terminal equipment, such as teletypewriters. A teletypewriter, when not printing, is in an idle mode waiting for a message. This message consists of a series of characters; each character will be preceded by a start bit and followed by one or two stop bits. Figure 3-27 represents a typical character format in which a start bit, eight data bits, and two stop bits are used.

The number of data bits in each character depends on the code being used, which depends on the type of equipment. Some systems use only six or seven data bits, or only one stop bit. The number of bits per character and the number of stop bits is controlled by the program.

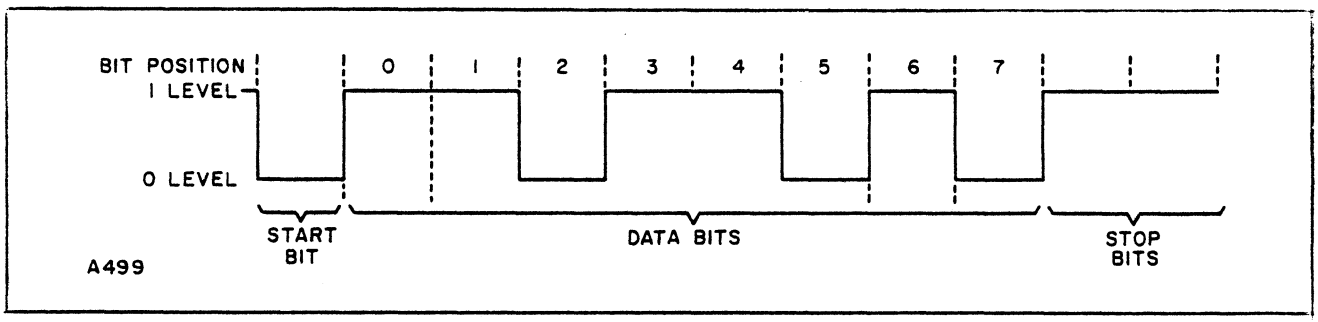


Figure 3-27. Asynchronous Serial Communications Format

Model 1534 Instructions

It was explained in Section 2 of this manual that the Model 1534, 1535, and 1539 are operated by the "176" instructions at the communications interface. The IWR chooses the type of adapter and selects the function to be performed. In some cases, a control word must have been previously placed in the accumulator.

The following paragraphs explain the function performed by each instruction decoded by the 1534. The three-digit octal number shown before the instruction title is the IWR. Note that the first two positions are always 01; these specify the 1534.

a. 010 - Transmit Byte. This instruction transfers the contents of the accumulator to the communications adapter to be transmitted.

b. 011 - Write Control Word. This instruction transfers the control word from the accumulator to a command decoder in the communications adapter.

c. 012 - Read Data Byte. This instruction transfers the contents of a buffer register in the communications adapter into the accumulator. The contents are the character received from the remote unit.

d. 013 - Read Error Byte. This instruction is the same as Read Status, below.

e. 014 - Read Status. This instruction loads the status byte from the adapter into the accumulator. The format of the status byte is described later in this section.

f. 015 - Disconnect Data Set. This instruction disconnects the data set (the MODEM and telephone) from the 1534. It does so by deactivating the DTR (data terminal ready) line to indicate that the intelligent terminal is no longer ready.

g. 016 - Break Received, Reset. This instruction clears the break received indicator in the 1534. It must be executed after the break received status is detected in the status word.

h. 017 - Connect Data Set. This instruction establishes a connection between the 1534 and the MODEM by activating the data terminal ready (DTR) signal to indicate that the intelligent terminal is ready.

Control Word

Instruction 176-011, Write Control Word, transfers the contents of the accumulator to the 1534, where the control word is decoded and acted upon. The format of the control word is shown in figure 3-28.

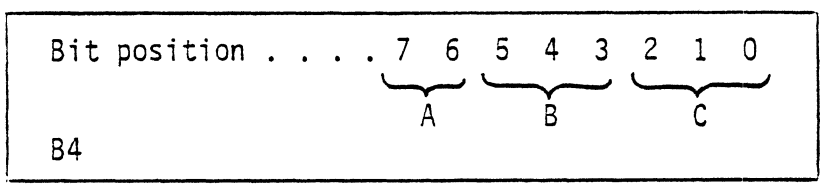


Figure 3-28. Format of Model 1534 Control Word

Note that the eight-bit control word is divided into three groups of bits, groups A, B, and C. The "A" group is either the parity selection group or part of a baud selection group. Parity selection is as follows:

A Value	Meaning
0	Odd parity
1	Even parity
2	Inhibit parity
3	Inhibit parity

The B group may be a format selection word, a MODEM command, or part of the baud selection group, depending upon the value of "C". This is shown below.

a. B Value, if format selection word, (C value = 7)

0 . . . . . 5-bit character, 1 stop bit
1 . . . . . 5-bit character, 2 stop bits
2 . . . . . 6-bit character, 1 stop bit
3 . . . . . 6-bit character, 2 stop bits
4 . . . . . 7-bit character, 1 stop bit
5 . . . . . 7-bit character, 2 stop bits
6 . . . . . 8-bit character, 1 stop bit
7 . . . . . 8-bit character, 2 stop bits

b. B Value, if MODEM command, (C value = 6)

0 . . . . . Set half-duplex mode
1 . . . . . Set 2nd channel low
2 . . . . . Set 2nd channel high
3 . . . . . Set rate low
4 . . . . . Set rate high
5 . . . . . Set RTS low
6 . . . . . Set RTS high
7 . . . . . Not used

c. A and B Values are Baud Select Byte 1 if C value = 3.

d. A and B Values are Baud Select Byte 2 if C value = 4.

The "C" group holds the command to be executed. Since there are three bits in this group, there are eight possible commands. These commands are listed below. In each case, the three-digit number preceding the command title is the entire control word and the digit on the right is the value of "C".

a. 000 - Master Reset. This command clears the 1534.

b. 001 - Set Transmit Break. This command causes a 600-millisecond break in the transmission.

c. 002 - Enable Receive Input. This command places the 1534 in the receive mode. It also takes the 1534 out of transmit mode unless it is operating in full-duplex mode.

d. AB3 or 4 - Load Baud Select Byte 1 or 2. These commands set the baud rate in the 1534. Note that the baud selection bytes are in the AB positions. When the command is a "3", baud selection byte 1 is loaded; when the command is a "4", baud selection byte 2 is loaded.

e. 005 - Receive Data Reset. This command clears the error group indicators, which consist of parity error, framing error, and overflow error.

f. XB6 - Decode "B" Position as MODEM Command. This command causes the 1534 to decode position B into the MODEM commands listed earlier. (The A position has no significance and thus is indicated with an X below.) The MODEM commands and the respective value of the control word are as follows:

(1) X06 - Set Half-Duplex Mode. This command sets the half-duplex mode in the 1534. The MODEM must be capable of half-duplex mode of operation.

(2) X16 - Set Second Channel Xmit. This command is used when the terminal is in the receive mode to signal the transmitting station that a message was in error (parity error, framing error, etc.).

(3) X26 - Reset Second Channel Xmit. This command clears the transmit condition set by the X16 command.

(4) X36 - Set Signal Rate High. This command generates a signal to the MODEM to set the signal rate high. The MODEM must be capable of two transmission rates. When this command is used, the baud rate in the 1534 must also be changed accordingly.

(5) X46 - Reset Signal Rate Low. This command generates a signal to the MODEM to set the signal rate low. When this command is used, the baud rate in the 1534 must also be changed accordingly.

(6) X56 - Set Request to Send. This command puts the primary channel of the MODEM into transmit and puts the secondary channel (when used) into receive.

(7) X66 - Reset Request to Send. This command puts the primary channel of the MODEM into the receive mode.

(8) X76 - Not used.

g. AB7 - Load Format Selection Word. This command causes the 1534 to load the A and B values, which control parity, number of bits in a character, and the number of stop bits, into internal circuits.

The status byte is acquired by the 176-014, Read Status, instruction. The meaning of each bit is shown below:

Bit	Meaning	Octal Value of Accumulator
0	Data set on.	001
1	Ring indicator on.	002
2	Communications error	004
3	Break received	010
4	Receive data available	020
5	Transmit register full	040
6	Second channel transmit data	100
7	Second channel receive data.	200

Model 1534 Programming Techniques.

The first point to be considered is that the format and operating conditions must be established in the Model 1534. After an instruction is sent to the 1534 to select a mode (number of data bits and stop bits), the adapter continues to operate in that mode until another instruction is received to change the mode.

An example of the instruction to select the character format is: IOC,C#6;011, Write Control Word, with the control word in the accumulator being 177. In this case, the Write Control Word instruction transfers the contents of the accumulator to the 1534. The "177" then selects even parity, eight bits per character, and two stop bits. (Refer to the control word description as required.)

Baud Selection

The speed at which data can be exchanged over a communications network is dependent upon the rate of transmission. This rate is referred to as the "baud". Baud is the maximum number of unit-length code elements that may be generated per second by the communications device. An easier way to remember baud is to think of it as bits per second.

The MODEM is internally wired to transmit at a selected baud. This can be changed by physically altering the strapping; however, after initial installation, the baud rate in the MODEM will usually remain fixed.

In order for the 1534 to communicate to the MODEM, the 1534 must be set to the same baud as the MODEM. The baud rate in the 1534 is selected by the program. The instruction used is the Write Control Word (IOC,C#6;011). The instruction must be executed twice, to send two baud-select bytes to the 1534. Table 3-4 lists the value of the control words that must be loaded into the accumulator.

Table 3-4. Baud Selection Control Words

Bit Rate	Byte 2	Byte 1	Bit Rate	Byte 2	Byte 1
15625	054	173	822.4	044	063
10417	064	173	801.3	024	033
7812.5	074	073	781.3	054	013
6250	074	133	781.3	164	073
5208.5	074	153	762.2	024	103
4464.4	034	163	744.0	054	043
3906.2	014	173	726.7	064	123
3906.2	154	173	710.2	034	053
3472.2	044	173	710.2	174	033
3125	064	073	694.4	014	123
2841.0	074	033	679.3	004	153
2604.2	074	113	664.9	004	063
2604.2	164	173	651.0	004	033
2404	034	143	651.0	174	113
2232.2	054	163	637.8	044	013
2083.3	024	173	625.0	024	003
1953.1	054	073	612.7	054	003
1953.1	174	073	601.0	064	103
1838.2	064	133	601.0	134	143
1736.2	074	053	589.6	074	043
1644.7	034	123	578.7	074	123
1562.5	014	153	568.2	034	153
1562.5	174	133	558.0	014	163
1488.1	004	163	558.0	154	163
1420.5	004	073	548.2	004	173
1358.7	044	033	538.8	044	073
1302.1	064	013	529.7	064	033
1302	174	153	520.8	074	013
1250.0	034	003	520.8	124	173
1201.9	054	103	512.3	034	103
1157.4	064	143	504.0	054	143
1116.1	074	063	496.0	064	163
1116.1	134	163	488.3	034	073
1077.6	034	133	488.3	154	073
1041.7	054	153	480.8	054	133
1008.1	024	163	473.5	064	153
976.6	014	073	466.4	034	063
976.6	114	173	459.6	014	133
947	044	133	459.6	164	133
919.1	064	053	452.9	044	153
982.9	034	023	446.4	024	063
868.1	014	113	440.1	014	033
868.1	144	173	434.0	044	113
844.6	004	143	434.0	174	053

Table 3-4. Baud Selection Control Words (Continued)

Bit Rate	Byte 2	Byte 1	Bit Rate	Byte 2	Byte 1
428.1	024	043	289.4	164	143
422.3	054	023	286.7	034	033
416.7	024	113	284.1	054	113
411.2	014	043	281.5	024	143
411.2	134	123	279.0	054	063
405.8	044	123	279.0	174	063
400.6	024	053	276.5	024	133
395.6	014	023	274.1	054	053
390.6	004	113	271.7	024	123
390.6	114	153	269.4	014	053
385.8	004	043	269.4	134	133
381.1	044	023	267.1	004	123
376.5	024	013	264.8	004	053
372.0	014	003	262.6	004	023
372.0	104	163	260.4	004	013
367.6	044	103	260.4	154	153
363.4	064	043	258.3	004	003
359.2	074	023	256.1	044	003
355.1	034	113	254.1	064	003
355.1	104	073	252.0	074	003
351.1	014	143	252.0	124	163
347.2	044	163	250.0	074	103
343.4	024	073	248.0	074	143
339.7	054	033	246.1	074	163
339.7	144	033	244.1	114	073
336.0	064	113	236.7	144	133
332.4	034	043	229.8	164	053
328.9	054	123	223.2	134	023
325.5	024	153	217.0	114	113
325.5	164	013	211.1	104	143
322.2	014	063	205.6	144	063
318.9	004	133	200.3	124	033
315.7	044	053	195.3	154	013
312.5	024	023	190.5	124	103
312.5	134	003	186.0	154	043
309.4	014	013	181.6	164	123
306.4	004	103	177.6	134	053
303.4	044	043	173.6	114	123
300.5	064	023	169.8	104	153
300.5	154	103	166.2	104	063
297.6	034	013	162.8	104	033
294.8	014	103	159.4	144	013
292.0	044	143	156.3	124	003
289.4	064	063	153.2	154	003

Table 3-4. Baud Selection Control Words (Continued)

Bit Rate	Byte 2	Byte 1	Bit Rate	Byte 2	Byte 1
150.2	164	103	86.8	144	163
147.4	174	043	85.9	124	073
144.7	174	123	84.9	154	033
142.0	134	153	84.0	164	113
139.5	114	163	83.1	134	043
137.1	104	173	82.2	154	123
134.7	144	073	81.4	124	153
132.4	164	033	80.5	114	063
130.2	174	013	79.7	104	133
128.1	134	103	78.9	144	053
126.0	154	143	78.1	124	023
124.0	164	163	77.4	114	013
122.1	134	073	76.6	104	103
120.2	154	133	75.8	144	043
118.4	164	153	75.1	164	023
116.6	134	063	74.4	134	013
114.9	114	133	73.7	114	103
113.2	144	153	73.0	144	143
111.6	124	063	72.3	164	063
110.0	114	033	71.6	134	033
108.5	144	113	71.0	154	113
107.0	124	043	70.4	124	143
105.6	154	023	69.8	154	063
104.2	124	113	69.1	124	133
102.8	114	043	68.5	154	053
101.5	144	123	67.9	124	123
100.2	124	053	67.3	114	053
98.9	114	023	66.8	104	123
97.7	104	113	66.2	104	053
96.5	104	043	65.7	104	023
95.3	144	023	65.1	104	013
94.1	124	013	64.6	104	003
93.0	114	003	64.0	144	003
91.9	144	103	63.5	164	003
90.8	164	043	63.0	174	003
89.8	174	023	62.5	174	103
88.8	134	113	62.0	174	143
87.8	114	143	61.5	174	163

Connecting the Date Set

Before data can be exchanged between the 1534 and the MODEM, a control connection must be established between the two units. The signal that achieves this is the data terminal ready (DTR). When active, this signal tells the MODEM that the terminal is ready to send or receive data. It is turned on by the Connect Data Set instruction (IOC,C#6;017) and turned off by the Disconnect Data Set instruction (IOC,C#6;015).

Transmitting from the Terminal

Before data can be sent over the telephone lines, the MODEM must be set to the transmit mode. This is achieved by executing the Write Control Word instruction with the Set Request to Send command (IOC,C#6;011, accumulator equal to 056). The transmit mode is cleared by the Reset Request to Send command (IOC,C#6;011, accumulator equal to 066).

Data to be transmitted is first loaded into the accumulator; then the Transmit Byte instruction (IOC,C#6;010) transfers the contents of the accumulator to the transmit register in the 1534. The 1534 adds the start bit, stop bits, and parity bit and transfers the character to the MODEM in serial form. The MODEM then transmits the character over the communication lines.

When transmitting a block of data, the program must be able to determine when the 1534 requires a character to be transmitted. If data bytes are sent to the 1534 too fast, the 1534 cannot accept them; if the bytes are sent at a rate that is too slow, then there will be a gap between characters on the line. The method used to determine if the 1534 is ready for another byte is to check the status byte. If bit 5 (040 octal) is set, the transmit register is full and the 1534 cannot accept another byte.

The instruction to check status is IOC,C#6;014, Read Status. When this instruction is executed, the status word is loaded into the accumulator, where it can be examined.

Receiving at the Terminal

When initiating a receive mode, the setup is basically the same as for setting up transmit mode. The exceptions are the Enable Receive command (IOC,C#6;011 with accumulator equal to 002), and the Read Data Byte instruction (IOC,C#6;012). The Enable Receive command conditions the 1534 circuits for the read mode, thereby enabling the 1534 to receive data from the MODEM. The Read Data Byte instruction transfers the contents of the 1534 input register to the accumulator.

The 1534 removes the start bit, parity bit, and stop bits from the data character before loading it into the input register. Therefore, the programmer does not have to concern himself with the extra bits that make up the character.

In the receive mode, as in the transmit mode, the Read Status instruction is used to determine if the 1534 has a byte available to be loaded into the accumulator. If the input register has a byte available, bit 4 (020 octal) will be set to a one in the status word.

Low/High Rate Selection

There will be times when the transmission of data is hampered by noise on the communications lines. In extreme cases, transmission may become impossible at the high baud rates. To compensate for the line conditions, some MODEMS are equipped with a low rate feature. For example, the MODEM may have a normal baud rate of 2000 baud, which is referred to as high rate, and a low baud rate of 1000 baud, which is referred to as low rate.

If conditions are such that communications at the high rate are resulting in a high error rate, and thus a large number of retransmissions, then it may be advantageous to go to the slower, or low rate. The Set Signal Rate High command is IOC,C#6;011, with the accumulator equal to 036. The Set Signal Rate Low command is IOC,C#6;011, with the accumulator equal to 046.

The Secondary Channel

Some systems operate with a feature known as secondary channel. The secondary channel is commonly used to signal the transmitting station that an error was detected by the receiving station. The operation of the secondary channel is under program control at both stations. The advantage of this method is that it allows the transmitter to send continuous messages with the assurance that if an error occurs, the receiving station will signal the transmitting station via the secondary channel.

Two commands and a status bit are used when programming the secondary channel option. The commands are Set Secondary Channel Transmit (IOC,C#6;011 with accumulator equal to 016), and Reset Secondary Channel Transmit (IOC,C#6;011 with accumulator equal to 026). The commands are used by the receiving station to activate the secondary channel.

The status bit is bit 7 (200 octal) Secondary Channel Receive Data. The status bit informs the transmitter that an error has occurred; it is generated when the secondary channel is activated by the receiving station.

Use of the Break Condition

When a teletype is the receiving unit, a break condition is required to fill the teletype operational requirements. The break is the lack of data for at least two character times, and it causes the teletype to disconnect. The break condition can be initiated in the terminal by execution of the Enable Send Break command (IOC,C#6;011, accumulator equal to 001). If the break condition is initiated by the teletype, an indicator is set in the 1534, and status bit 3 (010 octal) is set. By checking the status byte, the program can determine that a break condition was initiated by the receiving station.

After the break condition has been handled by the program, the indicator in the 1534 that is storing the break indication must be cleared. The Reset Receive Break instruction (IOC,C#6;016) is used for this purpose.

Selection of Half-Duplex or Full-Duplex Mode

The 1534 has the capability of operating in full- or half-duplex mode. In the full-duplex mode, the terminal can send and receive at the same time. To operate in the full-duplex mode, the MODEM must be adaptable to full-duplex and two pairs of communications lines are required.

When power is turned on in the 1534, the unit comes on in the full-duplex mode. If half-duplex mode is required, the Set Half-Duplex Mode command (IOC,C#6;011, accumulator equal to 006) must be executed.

Example of the Receiving Sequence

Before data can be received by the 1534, the following setup steps are required:

a. Master reset to clear the 1534.

b. Select half- or full-duplex mode.

c. Enable receive mode.

d. Select baud rate word 1.

e. Select baud rate word 2.

f. Select number of data bits, stop bits, and parity.

g. Connect the data set.

Table 3-5 shows examples of how many functions of the 1534 are programmed. The setup routine (SUP) is the first routine shown. Program control is transferred to program label "SUP" from the mainstream program by a Stack and Branch (SBU) instruction. This setup sequence is similar to the steps listed above.

The first command to be sent to the 1534 is the master reset command. The code 000 is loaded into the accumulator; then the IOC,C#6;011 instruction at program label "SSC" sends the command to the 1534. Program control returns to the set-up sequence; then the half-duplex command is loaded into the accumulator and is sent to the 1534 in the same manner as the master clear. This sequence continues through the IOC,C#6;017 instruction at label "DSC", at which point the 1534 has been connected and is ready to receive data.

The method used by the program to determine when a data byte has been received by the 1534 is to check the status word. Bit 4 (octal 020) in the status word is a one if the receive register is full. The sequence required to acquire the status word is at program label "LCT". The IOC,C#6;014 instruction gets the 1534 status word and loads it into the accumulator. The status word is saved and program control returned to the main program. (Note that the instructions to test bit 4 are not shown.)

If a data byte is available in the 1534, program control is transferred to label "RDB". The IOC,C#6;012 instruction transfers the data byte from the 1534 to the accumulator, and the exit returns program control back to the main program.

This sequence of byte-by-byte receiving of data continues until the data buffer is full or there is no more data being transmitted. When the transmitting station stops sending data, the carrier is eliminated (in the half-duplex mode). When this happens, bit zero (data set on) in the status word is set to a one and the program detects the change. However, other conventions have been established. For example, control characters, such as ETX (end of transmission) at the end of the message are reset. Whatever convention is used to detect the end of message, the data will be processed or another communication message will be handled after the message has been received.

Example of a Transmitting Sequence

The set-up sequence for transmitting is the same as for receiving, but with the addition of the set-request-to-send command. The setup for transmitting requires the following steps:

a. Master reset to clear the 1534.

b. Select half- or full-duplex.

c. Enable receive.

d. Select baud rate word 1.

e. Select baud rate word 2.

f. Select number of data bits, stop bits, and parity.

g. Connect data set (DTR).

h. Set request to send.

Note that in step c, the enable receive command is given for transmit; however, this will not affect the operation.

After the connect data set (send DTR) command at program label "DSC" in table 3-5 has been executed, the main program returns program control to the load accumulator instruction at label "ST1". After the IOC,C#6;011 instruction is executed, the 056 code puts the 1534 and MODEM in transmit mode.

The data byte to be transmitted is loaded into the accumulator and program control is transferred to the instruction at program label "SDB". The IOC,C#6;010 instruction transfers the data byte from the accumulator to the 1534 to be transmitted.

Program control is returned to the main program, then back to program label "LCT" to check the status word. Bit 5 (octal 040) of the status word indicates the state of the transmit register. When bit 5 is set to a one the transmit register is full, therefore, the program must wait before sending another byte to the 1534. If bit 5 is equal to zero, the transmit register is empty and the program can send the 1534 another byte to be transmitted.

The sequence of sending a byte of data to the 1534 and checking the status word is repeated for each byte of the message. Line protocal defines the start and end of the message. However, when the message has been sent, a second message may be sent or unit switched to the receive mode. In this example, the receive mode is set by the sequence starting at program label "CT1".

Table 3-5. Model 1534 Routines

SETUP		
SUP:	LDA,R#0; Oct: 000.	Master Reset
	SBU,SCC.	
	LDA,R#0; Oct: 006.	Half-duplex
	SBU,SCC	
	LDA,R#0; Oct: 002.	Enable Receive
	SBU,SCC	
	LDA,R#0;BU1.	Baud Rate 1
	SBU,SCC.	
	LDA,R#0;BU2.	Baud Rate 2
	SBU,SCC.	
	LDA,R#0; Oct: 177.	8 bits, 2 stop, and even parity
	SBU,SCC.	
	BRU,DSC.	

Table 3-5. Model 1534 Routines (Continued)

	SETUP (Continued)	
DSC:	IOC,C#6;017.	Connect data set (DTR)
	EXU,000.	Exit return
	SEND COMMAND TO ASYNC (1534)	
SCC:	IOC,C#6;011.	Send command
	EXU,000.	Exit return
	RECEIVE DATA	
RDB:	IOC,C#6;012.	Receive byte
	EXU,000.	Exit return
	SEND DATA	
SDB:	IOC,C#6;010.	Send byte
	EXU,000.	Exit return
	RECEIVE STATUS	
LCT:	IOC,C#6;014.	Get status
	STA,R#0;SST.	Save status
	EXU,000.	Exit return
	SET TRANSMIT MODE	
ST1:	LDA,R#0; Oct: 056.	Set request to send
	IOC;C#6;011.	Write control word
	EXU;000.	Exit return
	CLEAR TRANSMIT MODE	
CT1:	LDA,R#0; Oct: 066.	Reset request to send
	SBU,SS1.	
	EXU,000.	Exit return

Model 1534-1 and 1534-2 Asynchronous Communications Adapters.

As mentioned earlier in this section, the 1534 is also available as a free-standing unit. In this case, the model number is either 1534-1 or 1534-2. The difference between the two units is that the 1534-2 operates the standard teletype current loop instead of a MODEM. Therefore, when a 1534-2 is used, the output is connected directly to the teletype equipment. This type of operation eliminates the need for the MODEM; however, it restricts the distance the teletype equipment can be located away from the terminal to the length of the cable available.

The 1534-1 and -2 is connected to the serial I/O channel instead of being internally installed in an intelligent terminal, as is the 1534. This requires a different programming approach. For example, all instructions sent to the 1534-1 or -2 are sent by use of the Transmit Control Byte instruction (IOC,C#5;003). Other than this difference, the 1534-1 and -2 respond to the same instructions as the 1534.

Keeping the aforementioned concepts in mind, the reader may wish to review the description of the serial I/O channel before proceeding.

Model 1534-1 and 1534-2 Programming Techniques.

There are two different programming approaches that can be used for the Model 1534-1 and -2. The first uses a dedicated communications concept in which only the communications program is operational. This approach simplifies programming; however, it slows the terminal to the relatively low speed of communications. The second method, the overlap concept, allows more than one unit (communications and printer) to be operational in a given time interval. While this method is more difficult to program, it utilizes the time between transmitting to or receiving from the communications device to do the printing and thus increases the effective data transfer rate of the terminal.

The method to be discussed is the overlap method. Table 3-6 lists the instructions required to allow reception or transmission, and the following paragraphs define the reasons for and results of each instruction.

Example of a Receiving Sequence

The first requirement is to set up the 1534-1 to receive data, and this requires the following instructions to be sent to the unit:

a. Master reset.

b. Half- or full-duplex mode.

c. Enable receive mode.

d. Select baud rate word 1.

e. Select baud rate word 2.

f. Transmit/receive control word.

g. Connect the data set.

Refer to table 3-6. The first sequence of instructions (SUP), the setup sequence, is in the same order as the sequence listed above. The first instruction loads the master reset command code into the accumulator. From the accumulator, the command is transferred to the input/output section of the terminal, then over the serial I/O channel, and finally to the 1534-1. All instructions and data follow the same path from the terminal to the 1534-1.

Before the instruction can be sent to the 1534-1, however, the 1534-1 must be logically connected to the intelligent terminal. This connection is established through the addressing sequence. Refer to table 3-6 and follow the program through SS1, SS2 to ADR. Note that the SS1 and SS2 labels allow for multiple uses of the routines.

The serial I/O address sequence (ADR) first clears the serial I/O line in preparation for the address. The octal 200 is added to the address to signal the presence of an address byte on the SIO line. Note that the IOC,C#5;007 and IOC,C#5;000 instructions together take the address from the accumulator, send it to the I/O control section of the terminal, and establish the conditions to enable the address to be sent over the SIO line. The IOC,C#5;003 instruction, at instruction label "XYA", causes the address to be transmitted over the SIO cable to the 1534-1.

When the 1534-1 receives the address, it responds by returning its address and status. The instruction sequence at program label "XIO" examines bits 6 and 7 of the status word to determine if an address or I/O error occurred during the address sequence. At the end of the sequence, before the exit, the accumulator is restored with the master reset command (000). This program has been written as a multi-use type of program. Therefore, entry may be from several different points. For this reason, the accumulator is stored and reloaded at several points, with seemingly no reason. However, if all possible applications of the program are examined, the manipulation of the accumulator will become apparent.

The exit causes the program to go to the LDA instruction at line SS2 + 2. The branch then takes program control to label "XXB", where the IOC,C#5;002 instruction is executed. This is a C#5 instruction that is normally used to transmit data. However, the command in the accumulator is the master reset command for the 1534-1. Therefore, the command is handled as data when it is sent through the I/O section and the serial I/O interface in the 1534-1.

Up to this point in the program, all SIO commands have been directed to the SIO interace or adapter in the 1534-1. The master reset command is directed to the communications board in the 1534-1 unit; again, this is achieved by sending the command over the SIO line as data and then executing a Transmit Control Byte instruction to the SIO adapter in the 1534-1.

After the IOC,C#5;002 instruction is executed, the exit transfers program control to line SS1 + 2; then the branch transfers program control to label "SCC". At label "SCC", the 011, which is the load command code for the communications board, is loaded into the accumulator.

At label "XXA", the IOC,C#5;004 instruction stops transmission on the SIO line after the master reset code is received by the 1534-1. Then, the IOC,C#5;003 instruction transmits the command code, 011, to the 1534-1. This command code then indicates to the 1534-1 that the code was actually a command. The communications board is then cleared.

Program control will then to go label "XIO", where the status is again checked. This status will be the result of the 011 command code sent to the 1534-1. The error routine is not covered in this example, since the purpose of the example is to explain how to program the terminal. The error sequence must analyze the error and then go into a retransmission sequence. The sequence depends upon the application of the adapter.

After the status has been checked, the exit transfers program control back to the setup sequence "SUP". The next command to be sent to the 1534-1 is the half- or full-duplex command. This command is necessary only if the MODEM is capable of full-duplex mode of operation.

This same sequence is repeated for each command in the setup sequence with the exception of the connect data set sequence. This sequence starts at label "DSC" and sends only a 017 command to the 1534-1. The 011 command code will not follow the 017, as was the case for the other commands. As a result of receiving the 017 command, the 1534-1 logically connects to the MODEM and is ready to receive data.

The program will exit from the setup sequence after status is checked for the 017 command. This exit is the instruction at program line XIP + 2.

Control returns to the mainstream of the program, where the input buffer is established in preparation to receive data. When a data byte has been received by the 1534-1, bit 4 in the status word is set. The program will have to interrogate the 1534-1 for status to check for this condition. Upon determining that a data byte is available, program control is transferred to the receive data sequence to get the byte of data from the 1534-1 and load it into memory.

The first sequence to consider in acquiring status is the receive communications status sequence. Refer to table 3-6, program label "LCT"; this is the entry point into the sequence. Again, the stack and branch directs program control to the address sequence to address the 1534-1. The address sequence checks the SIO line and address status; this is not to be confused with the 1534-1 status.

After the completion of the address sequence, program control is returned to the load accumulator instruction at program line "LCT" + 2. The 014 command causes the 1534-1 to send its status to the terminal. The 1534-1 status word will be handled on the SIO line as a data word. There will also be a line status word sent to the terminal. The line status word is checked by the sequence at program label XIO and is stored in "SAV". The 1534-1 status word is stored at label "SST" by the instruction at program "LLX", and program control is again returned to the mainstream of the program.

The program has to repeatedly check the 1534-1 status to determine if a data byte is available. When a byte is available, program control is transferred to the receive data sequence.

The receive data sequence, program label "RDB" first addresses the 1534-1. If no line errors are detected, the read receive data command (012) is sent to the 1534-1. Upon receiving this command, the 1534-1 sends the data byte over the SIO line to the terminal. The data byte is stored temporarily and the status byte is received and checked; if no line error is detected, the data byte is loaded into the accumulator and program control transferred to the mainstream program.

The sequence of checking status and transferring a data byte must be repeated for each byte received. This sequence continues until the data buffer is full or the transmitting station stops sending data.

When the transmitting station stops transmitting, the carrier is eliminated. The carrier is a high-frequency signal which is used to transmit data between the sending MODEM and the receiving MODEM. When the carrier is lost, the receiving terminal will put a zero in bit zero of the status word. When the zero is detected in the status word, the program knows that the carrier is off and, therefore, the transmitting station has stopped transmitting.

Example of a Transmitting Sequence

The setup for transmit mode is basically the same as for receive mode. The difference, of course, is that the request to send is used. The same sequence that is used for receiving setup can be used for transmit if, at the end, a Set Request to Send instruction is executed. If the program is operating in full-duplex, then both receive and transmit modes are set.

The mainstream of the program executes a stack and branch to program label "SUP" to setup the 1534-1. This sequence is the same as in the receive mode. In this example, the same setup sequence that was used for receiving will be used for transmitting. After the setup, however, the 1534-1 must be set to transmit. This is accomplished at program label "ST1" in table 3-6 by loading the accumulator with a (056) command code, then sending the command to the 1534-1.

With the 1534-1 and MODEM in the transmit mode, data can be transmitted. (The half-duplex mode is also assumed to be used.) The main program will load the data byte into the accumulator and execute a stack and branch to program label "SDC". The sequence required to transmit data is as follows:

a. Address the 1534-1.

b. Check status.

c. Transmit data byte.

d. Transmit load transmit register command.

e. Check status.

Refer to table 3-6. Program control is transferred to label "SDC", then to "XXC", and then to label "ADR", which is the address sequence. After the address is sent to the 1534-1, at label "XYA", the status is checked at label "XIO". If no errors occurred, the exit takes program control to the instruction at label "XXB". The IOC,C#5;002 instruction transmits the data byte from the accumulator to the 1534-1 via the serial I/O line.

The exit transfers program control to label "SDC" +2, where the load transmit register command (010) is loaded into the accumulator in preparation to be sent to the 1534-1. This command informs the 1534-1 that the previous byte was data and is to be transmitted.

The instructions at program label "XXA" and "XYA" clear the SIO line and send the (010) command to the 1534-1. Status is checked by the sequence starting at program label "XIO". If the status indicates no errors, then program control goes back to the mainstream of the program. In order for the program to determine when the 1534-1 is ready to accept another byte of data to be transmitted, the program must check for the 1534-1's status word. Bit 5 of the status word is a one when the transmit register is full. The sequence starting at program label "LCT" checks status. This sequence continues for each data byte that is transmitted. The number of bytes to be transmitted is, of course, under program control.

After the entire message has been transmitted, it may be necessary for the 1534-1 to go into a receive mode to accept an answer or message. The only requirement to go into receive mode is to reset the request to send.

Refer to table 3-6. If program control is transferred to label "CT1", the command code 066 is loaded into the accumulator. This is the reset request to send command. When this command is sent to the 1534-1, it goes into the receive mode.

Table 3-6. Model 1534-1 and 1534-2 Routines

```
                                SETUP

SUP:  LDA,R#0; OCT: 000.        Master reset
      SBU,SS1.
      LDA,R#0; OCT: 006.        Half-duplex
      SBU,SS1.
      LDA,R#0; OCT: 002.        Enable Receive
      SBU,SS1.
      LDA,R#0; BU1.             Baud rate 1
      SBU,SS1.
      LDA,R#0; BU2.             Baud rate 2
      SBU,SS1.
      LDA,R#0; OCT: 177.        8 bit, 2 stop, and even parity
      SBU,SS1.
      BRU,DSC.

DSC:  SBU;ADR.
      LDA,R#0; OCT: 017.        Connect data set (DTR)
      BRU,XXA.

                        TRANSMIT COMMAND TO ASYNC

SCC:  LDA,R#0; OCT: 011.        Command code
      BRU,XXA.

                        SIO ADDRESS SEQUENCE

ADR:  STA,R#0;SAV.              Save accumulator
      IOC,C#5;006.              Inhibit line
      LDA,R#0;XAD.              Address of 1534-1
      ADA,R#0; OCT: 200.        Define address byte
      IOC,C#5;007.              Set address
      IOC,C#5;000.              Set transmit
      ADA,R#0; OCT: 000.        Timing delay
      BRU,XYA.                  Go to transmit address
```

Table 3-6. Model 1534-1 and 1534-2 Routines (Continued)

```
                    TRANSMIT DATA TO 1534-1

SDC:  SBU,XXC.
      LDA,R#0; OCT: 010.          Load transmit register command

                    SEND CONTROL MODE BYTE 1

XXA:  IOC,C#5;004.                Stop transmit

XYA:  IOC,C#5;003.                Transmit control byte
      BRU,XIO.                    Go get status

SS1:  SBU,SS2.
      BRU,SCC.

SS2:  SBU,ADR.
      LDA,R#0;SAV.
      BRU,XXB.

                    SEND CONTROL MODE BYTE 2

XXD:  IOC,C#5;004.                Stop transmit
      IOC,C#5;003.                Transmit control byte

                    RECEIVE BYTE SEQUENCE SIO

      IOC,C#5;001.                Receive byte
      STA,R#0;SAV.                Save accumulator

XIO:  IOC,C#5;001.                Receive byte (status)
      ANA,R#0; OCT: 300.          Clear lower 6 bits (bit 6 & 7 status)
      CPA,R#0; OCT: 000.          Check for address or I/O error
      BRE,XIP.                    No error

                                  The error sequence is entered if bit
                                  6 address error or bit 7 I/O error is
                                  set.  A retry sequence would then be
                                  executed.

XIP:  LDA,R#0;SAV.                Restore accumulator
      EXU,000.                    Exit
```

Table 3-6. Model 1534-1 and 1534-2 Routines (Continued)

```
                    SEND DATA MODE BYTE SIO

XXC:  STA,R#0;SAV.          Save accumulator
      SBU,ADR.              Address unit

XXB:  IOC,C#5;002.          Transmit data
      EXU,000.              Exit

                    RECEIVE DATA FROM ASYNC

RDB:  SBU,ADR.
      LDA,R#0; OCT: 012.    Read data byte (receive)
      BRU,XXD.

                    RECEIVE COMMUNICATIONS STATUS

LCT:  SBU,ADR.              Address sequence
      LDA,R#0; OCT: 014.    Get 1534-1 status
      SBU,XXD.

LLX:  STA,R#0;SST           Save status
      EXU,000               Return

                    SET TRANSMIT MODE

ST1:  LDA,R#0; OCT: 056.    Set request to send
      SBU,SS1.
      EXU,000.              Exit, go to mainstream program

                    CLEAR TRANSMIT MODE

CT1:  LDA,R#0; OCT: 066.    Reset request to send
      SBU,SS1.
      EXU,000.              Exit, go to mainstream program
```

Model 1535 Binary Synchronous Communications (BSC) Adapter.

Both the Model 1535 and the Model 1539 can conduct synchronous communications. The advantage of synchronous over asynchronous is that synchronous transmission is up to 30 percent faster, the reason being that synchronous does not require start and stop bits for each character. Timing is achieved through synchronization characters at the beginning and during each data block or message.

These "sync" characters, which are shown in figure 3-29, are program-controlled; that is, the programmer may use any bit configuration desired; however, the standard USASCII code for sync is 026 octal. The code structure for the 1535 is either 6 or 8 bits. Figure 3-29 represents a typical data block in the synchronous format.

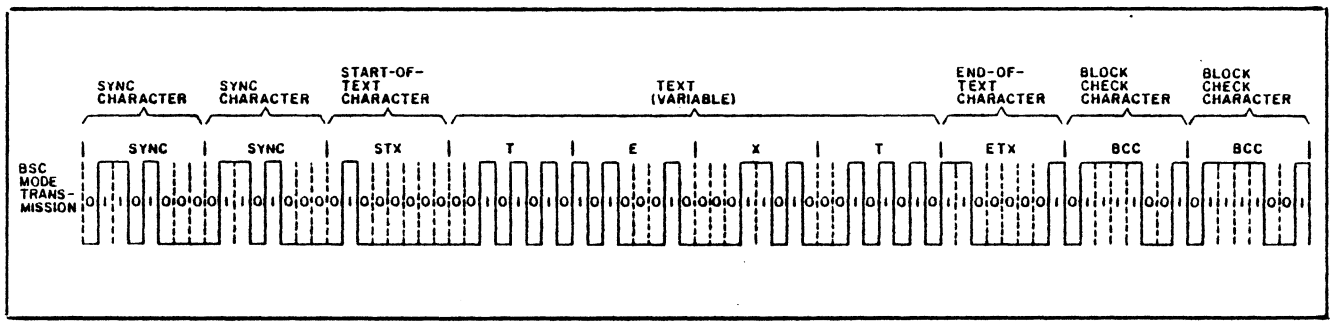


Figure 3-29. Format of a Synchronous Transmission

At the start of each transmission (that is, each block of data), two sync characters are required to establish and maintain synchronization. The sync character is also used as a fill character in the absence of data during transmission.

The start-of-text (STX) character is used by the program to determine the start of the data field, and the end-of-text (ETX) marks the end of the data field. Codes for these two characters are selected by the programmer; however, there is a USASCII standard of 002 octal for STX and 003 octal for ETX. The important fact to remember with regard to these two characters is that they appear only as a data byte to the equipment; the program must check for STX and ETX.

The block check character (BCC), shown at the end of the message in figure 3-29, is generated by the program, following the CRC_{16} polynomials specified by IBM Specification GA27-3004-1. The 16 bits of the BCC are the last two bytes of the data block.

The BCC is completely under program control. When the program is in the transmit mode, the BCC is generated and transmitted. When the program is in the receive mode, the BCC is developed as data bytes are received and the new BCC is compared to the BCC received with the data block.

After the BCC, two pad characters of all one's should be sent over the line before line turnaround. This ensures that the BCC has been sent before the terminal and MODEM go into receive mode.

Certain modes require leading pad characters. These are either sync characters or alternating zeros and ones.

Model 1535 Instructions

The Model 1535 is also operated by instructions in the "176" group. As with the other communications adapters, "176" selects the communications interface and the IWR chooses the type of adapter and the function to be performed. In some cases, a control word must have been previously established in the accumulator.

The following paragraphs explain the function performed by each instruction decoded by the 1535. The three-digit octal number shown before the instruction title is the IWR. Note that the first two positions are always 02.

a. 020 - Transmit Byte. This instruction transfers the contents of the accumulator to the communications adapter to be transmitted.

b. 021 - Write Control Word. This instruction transfers a control word from the accumulator to a command decoder in the communications adapter. The format of the control word is shown later in this description.

c. 022 - Read Data Byte. This instruction transfers a received data byte from the communications adapter into the accumulator.

d. 023 - Read Error Byte. This instruction is not required with the Model 1535. If a Model 1539 adapter, which can execute programs written for a 1535, is used, this instruction is "Read Secondary Status".

e. 024 - Read Status Byte. This instruction transfers a status byte from the communications adapter into the accumulator. The format of the status byte is shown later in this description.

f. 025 - Load Sync Register. This instruction transfers a sync character code from the accumulator into the sync register in the adapter.

g. 026 - Load DLE Register. This instruction transfers a DLE character code from the accumulator into the DLE register in the adapter.

Control Word

Instruction 176-021, Write Control Word, transfers the contents of the accumulator to the 1535, where the control word is decoded and acted upon. The octal value of the control word and the function of each command are as follows:

a. 000 - Set Transmit. This command clears the receive mode and starts transmission. It activates the DTR (data terminal ready) and RTS (request to send) signals to the MODEM. Sync characters are transmitted until the data is available.

b. 001 - Select 6-Bit Code. This command specifies that a 6-bit code is to be used in the adapter for both transmission and reception.

c. 002 - Set Receive. This command places the adapter in the receive mode. It clears the transmit mode, and it activates the DTR line to the MODEM. The adapter begins to look for synchronization characters.

d. 003 - Search for Synchronization. The command restarts the search for sync characters in the data being received. It clears the "synchronized" status previously achieved.

e. 004 - Set Secondary Channel Carrier On. This command activates the secondary channel in order to signal the unit sending data to the communications adapter.

f. 005 - Set Secondary Channel Carrier Off. This command deactivates the secondary channel.

g. 006 - Set Timer. This command starts a 1/4-second timer. The indication that the timer has expired is provided in the status word.

h. 007 - Set Low (600 or 900 baud). This command selects the low-range baud rate.

i. 010 - Reset Low (1200 or 1800 baud). This command selects the high-range baud rate.

j. 011 - Set Transparent Mode. Places the adapter in the transparent mode. Use of this mode is discussed later in this description.

k. 012 - Reset Transparent Mode. Places the adapter in the non-transparent mode.

l. 013 - Initialize. This command is the equivalent of a general clear. It causes the adapter to stop all functions in progress and disconnect from the MODEM.

m. 014 - Set Test. This command causes the communications adapter to place itself in the test mode.

n. 015 - Set External (MODEM) Clock. This command causes the adapter to use the MODEM clock as a source of timing.

o. 016 - Set End. This command inhibits decoding of the sync characters. It permits the transfer of all characters from the receive register to the accumulator. This is necessary in order that the CRC may be checked.

p. 017 - Set 600/1200 and Reset 900/1800. This command selects either the 600 or 1200 baud rate, depending upon whether the low or high range is selected, and clears the selection of the 900 or 1800 baud rate.

Status Byte

The status byte is acquired and placed in the accumulator by the 176-024, Read Status Byte, instruction. The meaning of each bit is shown below.

Bit	Meaning	Octal Value of Accumulator
0	Secondary channel data	001
1	Ring indicator on	002
2	Data set ready	004
3	Receiver Q register overflow	010
4	Receiver not in sync	020
5	Receive data available	040
6	Timer expired	100
7	Transmit register empty	200

After the status byte is placed in the accumulator, each bit can be examined. Since the meaning of each bit is rather complex, further explanation is necessary. This appears in the following paragraphs. In each case, a one in the respective bit position is the active state.

a. Bit 0 - Secondary Channel Data. This bit indicates that the secondary channel has been activated at either the sending or receiving unit.

b. Bit 1 - Ring Indicator On. This bit indicates that the telephone at this adapter is ringing and thus an incoming call is present.

c. Bit 2 - Data Set. This bit indicates the status of the telephone/MODEM combination. In the transmit mode, it indicates that the DSR (data set ready) and CTS (clear to send) signals are active. In the receive mode, it indicates that the DSR and CD (carrier detected) signals are both present.

d. Bit 3 - Receive Q Register Overflow. This bit indicates that a receive overflow has occurred. A second character has been received before the first was transferred to the accumulator; hence, the first character was lost.

e. Bit 4 - Receiver Not In Sync. After synchronization is achieved, this bit is cleared to a zero. If synchronization is lost, it is set to a one.

f. Bit 5 - Receive Data Available. This bit indicates that a received data byte is available for transfer to the accumulator. It is cleared when the transfer takes place.

g. Bit 6 - Timer Expired. This bit indicates that the 1/4-second timer started by the Set Timer command has expired.

h. Bit 7 - Transmit Register Empty. A one in this bit position indicates that the next byte to be transmitted can be transferred from the accumulator to the communications adapter.

Model 1535 Programming Techniques.

The following paragraphs outline the steps involved in programming the 1535. Also included are sample routines that can be used to send and receive data.

Organization of the Message Block

The message to be transmitted should be assembled in memory before starting transmission. It must include all the control characters that are required for the complete message. For example: the two sync characters, start-of-text (STX) character, data, end-of-text (ETX) character, and two block check characters. The only characters that are inserted by the adapter are the sync characters as a fill character or, in the transparent mode, the DLE and sync characters as fill characters. Of course, the programmer must be familiar with the line protocol before preparing the message block.

Transmitting Sequence

Before data can be transmitted to a remote site, the terminal must be initialized and logically connected to the MODEM. Initialization requires the following steps:

a. Master clear

b. Load sync register

c. Load DLE register (transparent mode only)

d. Set 6-bit mode, if applicable

e. Select baud rate

f. Set transmit mode

g. Load Q register with data to send

The master clear to the 1535 is accomplished by the Initialize command, which is IOC,C#6;021 with the accumulator equal to 013. This instruction causes the adapter to assume the 900 baud rate, the 8-bit code, disengage itself from the MODEM, and cease all functions.

There is no restriction in the adapter as to the code that is used as the sync or DLE characters. The adapter contains a sync register and a DLE register, which can be loaded with any code. (The sync character is used in both non-transparent and transparent modes, while the DLE code is used only in transparent mode.) The standard sync and DLE codes are:

Sync	DLE
USASCII - 026	USASCII - 020
EBCDIC - 062	EBCDIC - 020
6-Bit Transcode - 72	6-Bit Transcode - 77

Loading of the sync and DLE registers requires several instructions. In both cases, the Set Test Mode instruction (IOC,C#6;021, accumulator equal to 014) and the Initialize (clear) instructions must be executed. The following routine loads the sync and DLE registers:

```
LDA,R#0; OCT: 014  }
IOC,C#6;021        }   Set Test Mode
LDA,R#0; OCT: 062      Load accumulator with EBCDIC sync character
IOC,C#6;020            Transfer accumulator contents to Q register
IOC,C#6;025            Transfer Q register to sync register
SMS,S#0                Set memory section
SBU,D91                *35 ms. delay
LDA,R#0;  OCT: 020     Load accumulator with EBCDIC DLE character
IOC,C#6;020            Transfer accumulator contents to Q register
IOC,C#6;026            Transfer Q register to DLE register
LDA,R#0; OCT: 013  }
IOC,C#6;021        }   Initialize.  Reset test mode.
```

*The 35 ms. delay is required to ensure that the sync character is loaded before sending the DLE character to the terminal.

The 1535 is capable of operating in either a 6- or 8-bit code format. Initialization places the terminal in 8-bit code format. The instruction used to select 6-bit mode is IOC,C#6;021, with the accumulator equal to 001.

There are four different baud rates that can be selected internally in the 1535 adapter. These rates are 600, 900, 1200, and 1800 baud. There are three instructions involved in selecting the baud rate. They are IOC,C#6;021, with the accumulator equal to 007, 010, or 017. The selection of the four baud rates is as follows:

a. 900 baud - Selected by initialization.

b. 600 baud - After initialization, the IOC,C#6;021 instruction, with accumulator equal to 017, must be executed.

c. 1800 baud - After initialization, the IOC,C#6;021 instruction, with accumulator equal to 010, must be executed.

d. 1200 baud - After initialization, the IOC,C#6;021 instruction, with accumulator equal to 010, and IOC,C#6;021, with accumulator equal to 017, must be executed.

The 1535 terminal can also operate with an externally-selected clock. In this mode, the baud rate will be determined by the MODEM, as the MODEM is the source of the external clock. To select the external clock, the execution of the IOC,C#6;021, accumulator equal to 015, instruction is required. This instruction must be executed after the initialization instruction.

The final step before sending data is to put the terminal in transmit mode and logically connect it to the MODEM. This is achieved by executing the IOC,C#6;021 instruction, with the accumulator equal to 000. The Set Transmit Instruction will do the following:

a. Activate the RTS (request to send), which causes the MODEM to go into the transmit mode.

b. Activate the DTR (data terminal ready), which indicates to the MODEM that the 1535 is ready to send or receive data.

c. Clear the receive mode.

d. Initiate the continuous transmission of sync characters until data is available to send.

At this point, the terminal and MODEM are ready to start transmitting the data. The data to be transmitted is loaded into the accumulator; then the IOC,C#6;020 instruction is executed. This instruction transfers the contents of the accumulator to the Q register in the adapter. The data is then transmitted over the line.

The program is responsible for determining when the adapter needs another byte to transmit. It first reads the status word. The IOC,C#6;024 instruction causes the adapter to present its status by loading the status word into the accumulator. The program can then check the status word; if bit 7 is a one, then the transmit Q register is empty and is ready to receive another byte from the accumulator. If bit 7 is a zero, the transmit Q register is full and cannot accept another byte. The sequence of checking status and then sending a byte to the 1535 is continued for the complete message.

If data is not available at the Q register for transmission, the 1535 automatically inserts a sync character if it is operating in the non-transparent mode and DLE combination if it is operating in the transparent mode. In both cases, the code is referred to as the idle code. As mentioned earlier, the sync and DLE characters are held in the sync and DLE registers. These are the registers that were loaded by the aforementioned sequences.

Receiving Sequence

Before data can be received from a remote site, the terminal must be initialized and logically connected to the MODEM. The sequence is very similar to the transmit initialization.

Initialization requires the following steps:

a. Master clear

b. Load sync register

c. Load DLE register (transparent mode only)

d. Set 6-bit mode, if applicable

e. Select baud rate

f. Set receive mode

g. Load accumulator with receive data from Q register

h. Set end after ETX has been detected.

Note that steps a through e are the same as for the transmitting sequence and it is not necessary to cover them again. The Set Receive instruction IOC,C#6;021, accumulator equal to 002, is executed to put the terminal in receive mode and logically connect it to the MODEM. Executing the Set Receive instruction will achieve the following:

a. Activate DTR (data terminal ready), which indicates to the MODEM that the 1535 is ready to send or receive data.

b. Cause the adapter to look for synchronization, which means that the adapter is searching for the sync character in preparation to receive data.

c. Clear transmit. The clearing of the transmit mode is necessary during a turnaround operation; that is, going from transmit to receive mode. If the sequence had started from an initialize condition, this would not be required.

After executing the Set Receive instruction, the terminal will start looking for sync characters and, at some point, start receiving data. The sync characters are not sent to the accumulator; therefore, the first character to be received should be a start-of-text character. The STX character will cause bit 5 in the status word (receive data available) to be set to a one.

The program must execute a Read Status instruction (IOC,C#6;024) to check for status. Upon detecting the presence of a data character, the program must execute a Read Data Byte instruction, IOC,C#6;022. This instruction causes the contents of the receive Q register to be transferred to the accumulator. The program must repeatedly check status and, when a byte is available in Q register, transfer it to the accumulator.

The Set End instruction, IOC,C#6;021, accumulator equal to 016, will inhibit decoding of the sync character, thereby permitting the transfer of all characters from the receive Q register to the accumulator. This action is necessary for two reasons: (1) If the CRC character is equal to the sync character, it would not be detected and, (2) in order to detect sync characters that are interspersed in the message to maintain synchronization.

If the data received is not transferred out of the receive Q register as fast as data is being received, a data overrun will occur. When this condition exists, bit 3 of the status word will be set to a one. Bit 3 is reset when the Search for Sync instruction is executed.

Line Turnaround

When the program changes from a receive mode to a transmit mode, or vice-versa, it is not normally necessary to go through the entire setup sequence. The only instruction required to change from a receive mode to a transmit mode is the transmit instruction. To go from a transmit mode to receive mode requires the execution of the Set Receive and the Search for Sync instructions. The Search for Sync instruction, IOC,C#6;021, with the accumulator equal to 003, causes the receiving circuits to consider the adapter out of synchronization and look for synchronization characters.

Transparent Mode

The transparent mode permits the use of a wide range of coded data. This is achieved by treating the normally-restricted control character codes as a bit pattern, thus permitting unrestricted data coding to enable transmission of binary data, packed-decimal, machine-language programs, etc.

When a control character is required in transparent mode, the character must be preceded by a DLE character to be recognized as a control character. This is shown in figure 3-30.

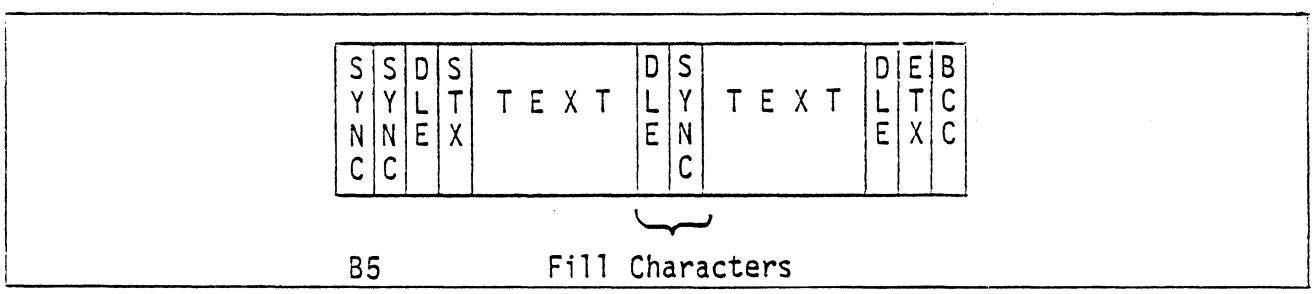


Figure 3-30. Use of the DLE Character

There are three instructions used in conjunction with transparent mode. They are Set Transparent Mode IOC,C#6;021 with the accumulator equal to 011, Reset Transparent Mode IOC,C#6;021 with the accumulator equal to 012, and Load DLE Register IOC,C#6;026. Before writing a program for transparent mode, reference should be made to IBM System Reference Library publication entitled "General Information - Binary Synchronous Communications", Order No. GA27-3004-2.

Timer.

The adapter is equipped with a 1/4-second timer, which is program-controlled. The timer output is related to bit 6 of the status word. When bit 6 is equal to a one, the timer is inactive or has expired. When bit 6 is a zero, the timer is active.

The Set Timer instruction IOC,C#6;021, accumulator equal to 006, is used to activate the timer. Once activated, the timer will run until it expires; there is no reset instruction.

Secondary Channel

Some communications systems are equipped with the secondary channel capabilities described at the beginning of this section. The secondary channel is commonly used to signal the transmitting station that an error was detected by the receiving station. The operation of the secondary channel is under program control at both the transmitting and receiving stations.

The advantage of the secondary channel application is that it allows the transmitting terminal to send continuous messages with the assurance that, if an error occurs, the receiving station will signal the transmitting station via the secondary channel.

Two instructions and a status bit are used when programming the secondary channel. The instructions are Set Secondary Channel, IOC,C#6;021, accumulator equal to 004, and Reset Secondary Channel, IOC,C#6;021, accumulator equal to 005. The status bit is bit 0 of the status word. The status bit, when equal to a one, indicates to the transmitter that an error has occurred; this bit is generated when the secondary channel is activated by the receiving station. The Set Secondary Channel instruction is used by the receiving station to activate the secondary channel.

Test Mode

The 1535 is equipped with self-testing capabilities. The test mode is entered by executing the Set Test instruction IOC,C#6;021, with the accumulator equal to 014.

This causes the following sequence to take place:

a. The MODEM interface becomes inactive, as follows:

   (1) Request to Send is inactive.

   (2) Data Terminal Ready is inactive.

   (3) Transmit Data for both primary and secondary channels are forced on.

   (4) All signals from the MODEM are ignored.

   (5) The transmit CRC is gated out. The transmitted character pattern will then have a known CRC that is sent through the normal transmit channel and compared with the receive CRC calculation.

b. The following signals are simulated:

   (1) Ring Indicator is active.

   (2) Clear to Send is active if Request to Send is set.

   (3) Data Set Ready is active if Data Terminal Ready is set.

   (4) Carrier Detected is active if the adapter is in the transmit mode.

c. The transmitted data is connected to the received data input.

d. The communications adapter considers itself to be in both the transmit and receive modes.

After entering the test mode, the transmit and receive instructions can be executed. The data that is transmitted is turned around and received. In this manner, the adapter can be tested. The complete self-test is in the Communications Test, Publication Number 30010.

Example of Transmit Routine

The remaining paragraphs of this section explain the examples of the 1535 programming routines given in table 3-7. First to be discussed is transmission.

Before the 1535 can transmit data, it must be placed in the transmit mode. The following steps are required:

a. Master clear

b. Load sync register

c. Load DLE register

d. Select baud rate

e. Set transmit mode

f. Load Q register with data to send.

Program control is transferred, by the mainstream program, to program label "SUP" shown in table 3-7. The initialize command (013) is loaded into the accumulator and sent to the 1535. This command will clear the controls in preparation for the transmit command sequence. The Set Test Mode command (LDA,R#0;OCT: 014) conditions the adapter for the loading of the sync and DLE registers.

Because it houses the sync and DLE registers, the 1535 is not restricted to a fixed sync or DLE character. That is, any bit combination can be used. In this case, the sync character is supplied by the mainstream program. The LDA instruction loads the sync character into the accumulator. Next, the IOC,C#6;020 instruction transfers the contents of the accumulator to the "Q" register in the 1535, and the IOC,C#6;025 instruction transfers the sync character from the "Q" register to the sync register.

A delay of 35 ms is initiated to allow the sync register loading to be completed before the Load DLE Register instruction is executed. The delay used in this example is the D91 delay located in the standard background routine for the cartridge tapes.

The DLE character, like the sync character, is under program control. The DLE is loaded into the DLE register by the same procedure as the sync character. The only difference is the Load DLE instruction - the IOC,C#6;026.

After the sync and DLE registers are loaded, the adapter must be switched from the test to the normal mode. The initialize command is sent to the adapter to do this.

The initialize command selects the 900 baud rate. If the 900 baud rate is desired, no further baud selection is required. If, however, the 600 baud rate is to be selected, program control must be transferred to program label B06, where the 017 command selects the 600 baud rate. The 1800 baud rate is selected by the 010 command, which is executed if program control is transferred to program label B18.

Both the 010 and 017 commands must be sent to the adapter to select 1200 baud. If program control is transferred to program label B12, this is done. When executing the 010 and 017 commands, the order in which the commands are sent to the adapter is not significant.

The final step in the initialization sequence is to set transmit. Program control is transferred to program label "STR", where the transmit command is loaded into the accumulator. The command is sent to the adapter at program label "SR1". This causes the 1535 to be logically connected to the MODEM and transmit sync characters until a data character is available.

When the mainstream program has data available to be transmitted, program control is transferred to "TXM". Before the data is sent to the adapter, the 1535 status word must be checked to ensure that the Q register is empty and the data set is ready.

The Read Status instruction at program label "TOT" acquires the status byte, and the "TMJ" instruction checks for data set ready (bit 2) and Q-register empty (bit 7). The program must wait until these conditions are satisfied before continuing. It will then transfer to program label "TQU" where the data byte will be sent to the Q-register to be transmitted.

The transmit character sequence is executed each time a character is to be transmitted. This holds true for all the characters in the data message, including control characters. When the message is completed (that is, the program stops sending data but the transmit has not been cleared), sync characters are transmitted.

Example of Line Turnaround Routine

Normally, the initial conditions remain the same when the mode is switched from transmit to receive. However, there may be exceptions, in which case a new initialization sequence must be executed. In this example, shown in table 3-7, the assumption is made that the initial conditions will be the same. Therefore, line turnaround will only require changing from transmit to receive mode.

Program control is transferred to program label "SRO". The Set Receive command there clears the transmit mode and sets the receive mode, and the Search for Sync command causes the adapter to look for sync characters. The reception of two consecutive sync characters indicates the start of the message. After synchronization has been achieved, the 1535 starts loading the data into the receive Q-register.

When changing from a receive mode to transmit mode, the program will only send the Set Transmit command to the terminal. This command will clear the receive mode and set the transmit mode. Data can then be transmitted, as it is the responsibility of the receiving terminal to achieve synchronization.

Example of a Receive Routine

The initialization sequence for the receive mode is the same as that for the transmit mode, up to the point where the set receive command is sent to the terminal. At program label "SRO", the receive command is sent to the terminal, followed by the Search for Sync command. The adapter then synchronizes with the incoming message, and starts loading the characters into the receive Q-register. As each character is loaded, status bit 5 is set to one.

The program checks status, at program label "RB3", and detects the availability of a byte. At this point, the status word must also be checked for an overflow condition and a timer expiration. Assuming these conditions do not exist, the first character is loaded into the accumulator by the command at program label RCV. In most cases, this character is the start-of-text (STX) character. The sequence of checking status and transferring a character from the adapter to the accumulator continues for each character of the message.

After the ETX character has been received, program control is transferred to program label "END" to execute the Set End command. This permits all characters to be transferred from the adapter to the accumulator. This is necessary to (1) allow the sync character to be loaded into memory so that the program can maintain sync and, (2) load the CRC character into memory if it is identical to a sync character code.

A timer check is made during the "RB3" routine. This timer can be used to check for time duration between messages or the duration of messages. Excessive time periods in either case may be an indication that a problem exists and the program should exit to prevent hang-up.

The receive overflow condition is checked at program label "TMU". If the overflow condition is detected, a separate routine will have to determine what action is necessary to recover from the condition. The overflow error is generated when a new character is loaded into the receive Q-register before the previous character was transferred to the accumulator. Therefore, at least one character will be lost, and when the CRC is calculated, it will not compare with the received CRC.

Table 3-7. Model 1535 Sample Routines

SETUP		
SUP:	LDA,R#0; OCT: 013	Initialize
	IOC,C#6;021	Clear to 1535
	LDA,R#0; OCT: 014	Set test mode
	IOC,C#6;021	Command to 1535
	LDA,R#0;Sync	Load accumulator with sync character
	IOC,C#6;020	Accumulator→Q
	IOC,C#6;025	Load sync register
	SUB,D91	Delay
	LDA,R#0;DLE	Load accumulator with DLE character
	IOC,C#6;020	Accumulator→Q
	IOC,C#6;026	Load DLE register
	LDA,R#6; OCT: 013	Initialize (clear test mode)
	IOC,C#6;021	
	EXU,000	Exit to main program to define baud rate

Table 3-7. Model 1535 Sample Routines (Continued)

```
                        BAUD SELECT

B06:  LDA,R#0; OCT: 017          Select 600 baud
      IOC,C#6;021
      EXU,000                    Exit

B18:  LDA,R#0; OCT: 010          Select 1800 baud
      IOC,C#6;021
      EXU,000                    Exit

B12:  SBU,B06                    Select 600 baud
      SBU,B18                    Select 1800 baud
      EXU,000                    Exit

                        SET TRANSMIT

STR:  LDA,R#0; OCT: 000          Transmit command
      BRU,SR1                    Set transmit

                     TRANSMIT CHARACTER

TXM:  STA,R#0;TS2                Temporary storage

DEL:  Delay Sequence             To wait for Data Set Ready and Q empty

TOT:  IOC,C#6;024                Get status
      TMJ,TO2; OCT: 204          Data Set ready and Q empty
      BRE,TQU
      BRU,DEL

TQU:  LDA,R#0;TS2
      IOC,C#6;020                Transmit byte
      EXU,000
```

Table 3-7. Model 1535 Sample Routines (Continued)

SET RECEIVE

```
SR0:  LDA,R#0; OCT: 002        Set receive command
      IOC,C#6;021              Send command
      LDA,R#0; OCT: 003        Set look for sync

SR1:  IOC,C#6;021              Send command
      EXU,000                  Exit

RB3:  IOC,C#6;024              Get status
      TMJ,TO2; OCT: 040        Receive - Q
      BRE,TMU
      TMJ,TO4; OCT: 100        Go to time out sequence
      BRU,RB3                  If bit 6 of status word is set, then
                               the 1/4-second timer is expired; then
                               the time clock must be incremented.

TMU:  TMJ,TO2; OCT: 010        Receive overflow
      BRE                      Go to handle overflow condition
      SBU,RCV

RCV:  IOC,C#6;022              Transfer Q register to accumulator
      EXU,000                  Exit

END:  LDA,R#0;016              Set end command
      IOC,C#6;021              Send command
      EXU,000                  Exit
```

Model 1539 Interruptible Communications Controller.

The Model 1539 Interruptible Communications Controller is a single printed circuit board that allows an intelligent terminal to communicate with other data processing equipment over long-distance communications lines. In most of the terminals, the 1539 can be substituted for a Model 1535 BSC Adapter, and the 1539 will operate under control of the programs written for the 1535. However, the 1539 is not compatible with programs written for a Model 1534 Asynchronous Communications Adapter.

The most significant difference between the 1539 and the other two adapter boards is that the 1539 can operate in any one of many modes, three of which are: (1) asynchronous, (2) BSC, and (3) SDLC. The modes are selected by the program.

Another significant feature of the 1539 is its interrupt capability. This is discussed further under the heading "Interrupt Handling" later in this description.

Both the asynchronous and BSC modes were described earlier in this section; therefore, it is not necessary to cover that material here. It should be noted, however, that all instructions required for the asynchronous mode of operation have been designed into the controller but as of this writing, no programs have been prepared for the asynchronous mode in the 1539 controller. For this reason, the asynchronous mode is not covered in detail in this section.

SDLC Mode

SDLC means synchronized data link communication. The basic form is similar to BSC in that it is synchronous, but data is organized into frames as shown in figure 3-31 rather than the BSC format shown earlier in this section.

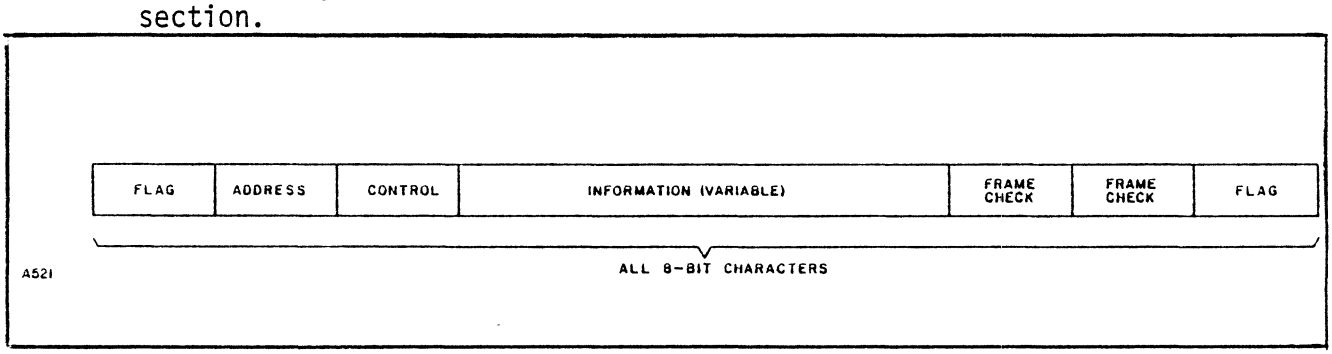


Figure 3-31. SDLC Transmission Format

A "flag" character begins and ends a frame. Like the control characters used in the BSC mode, the flag, address, control, and frame check information is interpreted by the terminal receiving the information. Except for the flag, these bit patterns are handled by the 1539 just as if they were data. The SDLC mode is basically a "burst" mode in which all the data in a frame is transmitted without interruption and then flags are transmitted as "fill" characters between frames.

There is one other control character used in the SDLC mode; that character is "abort", which means a transmission is being terminated. An abort control character is an octal 177 and a flag character is an octal 176.

The SDLC mode of operation uses a unique method to transmit and recognize control characters. As mentioned above, the flag character is 176 (01, 111, 110) and the abort character is 177 (01, 111, 111), which means that each of these characters requires at least six consecutive ones.

Whenever the 1539 is transmitting in the SDLC mode, the zero insertion circuits are activated. These circuits count the number of ones transmitted in succession, and after five consecutive ones they insert a zero. Of course the unit receiving the data must remove the added zero; this is the function of the zero deletion circuits.

In order to transmit a flag or an abort character, the processor sends a "send control byte" instruction to the 1539 and supplies the control character to the Q-register and subsequently to the transmitter. (Between frames, the transmitter automatically inserts flag characters in the data stream.) Sensing that a control character is to be sent, the control circuits prevent zero insertion from taking place. This produces a character in which at least six ones in a row appear.

When the 1539 is receiving in the SDLC mode, the zero deletion circuits count the number of consecutive ones received. If this count exceeds five, the control circuits set a status bit. When the processor reads the status word after being interrupted, it recognizes that a control character was received.

Model 1539 Instructions

As mentioned earlier in this description, the "176" group of instructions is used to control all communications adapters and, in addition, the Model 1539 can execute the instructions also used with the Model 1535. Of the "176" instructions, the 1539 responds to two groups: (1) 176-00n, and (2) 176-02n. The function performed by each of the instructions in these groups is described below. Note that the three octal digits preceding the instruction title are the IWR of the instruction, and that an asterisk indicates that these instructions are compatible with the Model 1535 instructions.

a. 000-Load Sync Register. This instruction loads the contents of the accumulator, which is a sync or flag character, into a match character holding register in the 1539. The octal value of the sync character depends on the code being used, as follows: (1) Transcode - 072, (2) EBCDIC - 062, and USASCII - 026. The flag character is an octal 176.

   Characters received from the MODEM are compared with the stored character. When the match is achieved, the receiver section recognizes that a sync or flag character has been received.

b. 001-Write Control Word 1. This instruction causes a control word held in the accumulator to be transferred to the 1539 and to be decoded by the command decoder.

c. 002-Send Control Byte. This instruction loads the transmitter section of the 1539 with the contents of the accumulator, which is a flag or abort control character, and causes the character to be transmitted. This applies to only the SDLC mode.

d. 003-Read Interface Status. This instruction reads the interface status byte, which is concerned with the MODEM, and transfers it to the accumulator.

e. 004-Load Control Register. This instruction loads the contents of the accumulator into control registers 1 and 2 in the 1539. The contents of these control registers define the clock rate, word length, and parity control for both the receiver and transmitter sections. Refer to figure 3-32 for the word format.

f. 005-Send Fill Register. This instruction allows the transmission of the contents of the primary fill register, which is a sync character or SDLC flag character, to the MODEM.

g. 006-Not Used.

h. 007-Disable Interrupt. This instruction disables the ability of the 1539 to send an interrupt signal to the processor.

i. 020 - Write Data.* This instruction loads the contents of the accumulator into the Q-register in the 1539 and subsequently transfers it to the transmitter section.

j. 021-Write Control Word.* This instruction causes a control word held in the accumulator to be transferred to the 1539 and decoded by the command decoder.

k. 022-Read Data.* This instruction loads the contents of the receiver section into the accumulator.

l. 023-Read Secondary Status.* This instruction transfers the controller status byte (secondary status) to the accumulator.

m. 024-Read Primary Status.* This instruction transfers the interrupt status byte (primary status) to the accumulator.

CONTROL WORD FORMAT

7	6	5	4	3	2	1	0

BAUD (SEE TABLE BELOW)

WORD LENGTH AS FOLLOWS:
0 0 - 8 BITS
0 1 - 7 BITS
1 0 - 6 BITS
1 1 - 5 BITS

PARITY AS FOLLOWS:
0 0 - NONE
1 0 - EVEN
1 1 - ODD

BAUD SELECTION

POSITION 7 6 5 4	BAUD SELECTED	TIME PER BIT	
0 0 0 0	1200	833US	
0 0 0 1	1800	555US	MICROSECONDS
0 0 1 0	4800	208US	
0 0 1 1	9600	104US	
0 1 0 0	110	9.09MS	
0 1 0 1	150	6.66MS	MILLISECONDS
0 1 1 0	300	3.33MS	
0 1 1 1	2400	417US	
1 0 0 0	75	13.33MS	MILLISECONDS
1 0 0 1	50	20.0MS	
1 0 1 0	19200	52US	
1 0 1 1	19200	52US	MICROSECONDS
1 1 0 0	2400	417US	
1 1 0 1	600	1.66MS	
1 1 1 0	200	5.00MS	MILLISECONDS
1 1 1 1	135 (134.5)	7.43MS	

A522

Figure 3-32. Control Word Format

n. 025-Load Fill Register.* This instruction loads the contents of the accumulator into the primary fill register in the transmitter section and into the match holding register in the receiver section. This character is a sync character in the BSC mode and a flag character in the SDLC mode.

o. 026-Load Alternate Fill Register.* This instruction loads the contents of the accumulator into the alternate fill register. In the BSC mode, this is a DLE (data link escape) character whose octal value is: Transcode - 037, EBCDIC - 020, and USASCII - 020. However, any value can be loaded.

p. 027-Enable Interrupt. This instruction enables the 1539 to interrupt the processor.

Control Word 1

Unlike the 1535, the Model 1539 responds to two control words. The first is loaded by instruction 176-001, Write Control Word 1, and is unique to the 1539. The second is loaded by instruction 176-021, Write Control Word 2, and this control word is compatible with the 1535 instruction.

Control word 1 is discussed first in the following paragraphs. The instructions involving control words are written in the form: 176-001-000, in which the last three digits are the octal value of the control word in the accumulator. These digits are shown in the following descriptions:

a. 000 - Reset Transmit. This command takes the 1539 out of the transmit mode.

b. 001 - Terminate Transmission. This command ends the current transmission after the current byte has been transmitted.

c. 002 - Reset Receive. This command takes the 1539 out of the receive mode.

d. 003 - Set Full Duplex Operation. This command allows full duplex operation by preventing the Set Receive command from clearing the transmit mode and the Set Transmit command from clearing the receive mode.

e. 004 - Set SDLC Mode. This command places the 1539 in the SDLC mode, which is also called the bit-synchronous mode.

f. 005 - Disable Idle Fill. This command terminates the transmission sequence when there is no data available for transmission and thus prevents the sending of fill characters.

g. 006 - Clear Timer. This command clears the 250-millisecond timer and the timer expired status.

h. 007 - Enable Idle Fill. This command allows fill characters to be transmitted by preventing the termination of transmission when there is no longer any data to transmit.

i. 010 - Set High MODEM. This command will set the MODEM speed high if feature is selected on 1539. A strap feature on the 1539 will select a constant MODEM high speed.

j. 011 - Set Low MODEM. This command will select low MODEM speed provided the 1539 has not been strapped to select high speed.

k. 012 - Reset 2 Stop Bits. This command will reset the 2 stop bit select in asynchronous mode.

l. 013 - Controlled Clear. This command clears the 1539.

m. 014 - Reset DTR. This command removes the DTR (data terminal ready) signal.

n. 015 - Send Break. This command produces a 600-millisecond break in the transmission.

o. 016 - Two Stop Bits. This command selects two stop bits to be used in the asynchronous mode.

p. 017 - Set Asynchronous. This command places the 1539 in the asynchronous mode.

Control Word 2

The 176-021 instruction, Write Control Word 2, provides the control words described below. Since this instruction is also used with the 1535, the Model 1539 must respond in the same way as the 1535 in order to execute communications programs prepared for the 1535. There are some minor differences, however. As indicated in the following descriptions, the 1539 does not respond to control words 007-set low range, 010-reset low range, 014-set test, 016-set end, and 017-select 600/1200 and reset 900/1800.

The 176-021 instruction provides the following control words:

a. 000 - Set Transmit. This command places the 1539 in the transmit mode. It also takes the 1539 out of the receive mode unless the 1539 is operating in the full-duplex mode.

b. 001 - Set 7-Bit Mode. This command controls the least significant bit of the word length select code, and it selects the 7-bit mode if an 8-bit word had been previously selected. This is the set 6-bit mode command in the 1535.

c. 002 - Set Receive. This command places the 1539 in the receive mode. It also takes the 1539 out of the transmit mode unless the 1539 is operating in the full-duplex mode.

d. 003 - Resync. This command clears the results of the previous searches for sync or flag characters and initiates a new search. It is used only when the 1539 is operating in the receive mode.

e. 004 - Mark Secondary Channel. This command forces the secondary channel data to signal to a "mark" (a one).

f. 005 - Space Secondary Channel. This command forces the secondary channel data signal to a "space" (a zero).

g. 006 - Start Timer. This command starts the 250-millisecond timer.

h. 011 - Enable Alternate Fill Sequence (Set Transparent Mode). This command causes the DLE character to be transmitted before each fill character in the BSC mode.

i. 012 - Disable Alternate Fill Sequence (Reset Transparent Mode). This command removes the DLE character from before each fill character in the BSC mode.

j. 013 - Initialize. This command produces a clear signal to many of the circuits in the 1539. It does not, however, reset the receiver and transmitter as the Controlled Clear command (176-001-013) does.

k. 015 - Select MODEM Clock. This command selects the MODEM clock as the source of the transmit and receive timing signals.

Status Words

Because of the complexity of the 1539, three status words are required to fully report conditions of interest to the programmer. They are the interrupt status word, the controller status word, and the interface status word.

As its name implies, the interrupt status word, which is read by instruction 176-024, Read Primary Status, reports on conditions that will generate an interrupt signal. The meaning of each bit in this status word is as follows:

a. Bit 0 - Secondary channel transition detected. This bit is activated on detection of any change of state on the secondary channel.

b. Bit 1 - Ring indicator. This bit is active whenever the ring indicator signal is present from the MODEM and the DTR (Data Terminal Ready) signal is not active.

c. Bit 2 - Data set. This bit is active if the 1539 is in the transmit mode and a CTS (Clear To Send) signal is present from the MODEM or if the 1539 is in the receive mode and the carrier detect signal is present from the MODEM. This is the only bit of the eight in this status word that does not generate an interrupt.

d. Bit 3 - Receiver error. This bit is active whenever a parity error, framing error, or receiver overrun error is present.

e. Bit 4 - Transmission shut-down/abort. This signal is active when a shutdown of the transmitter occurs.

f. Bit 5 - Receiver holding register full. This signal is present when a data byte is available for transfer to the accumulator.

g. Bit 6 - Timer expired. This signal is present whenever the 250-millisecond timer has expired.

h. Bit 7 - Transmit holding register empty. This signal is present when the transmitter can accept another byte.

The next type of status word is the controller status word, which reflects the conditions of the controller. This status word is read by the 176-023, Read Secondary Status, instruction, and it consists of:

a. Bit 0 - Secondary channel state. This bit is active when the secondary channel is in the "marked" state.

b. Bit 1 - Control byte detected. This bit is active when a control byte is received in the SDLC mode.

c. Bit 2 - Interrupt pending. This bit indicates that an interrupt condition exists.

d. Bit 3 - Receive overrun. This bit indicates that another character has been received before the processor accepted the last one.

e. Bit 4 - Receiver in sync. This bit indicates that synchronization has been achieved in the BSC or SDLC mode.

f. Bit 5 - Framing error. Indicates the absence of the proper stop bits in a character received in the asynchronous mode.

g. Bit 6 - Parity error. This bit is active if a parity check is enabled during a receive operation and a parity error is detected.

h. Bit 7 - Data set. This status bit is the same as bit two of the interrupt status word and is explained below.

The interface status word deals primarily with conditions at the 1539-to-MODEM interface. This word is read by the 176-003, Read Interface Status, instruction, and contains the following bits:

a. Bit 0 - 4-wire switch status. This bit monitors the RTS (Request To Send) signal to the MODEM. When the switch is in the two-wire position, the bit is equivalent to the transmit mode flip-flop. In the four-wire position, it is equivalent to the DTR (Data Terminal Ready) signal.

b. Bit 1 - Transfer enabled. This bit indicates transmission is in progress.

c. Bit 2 - Interrupt enabled. This bit indicates that the controller's interrupt circuit is enabled.

d. Bit 3 - Carrier detect. This bit indicates the presence of the carrier detected signal from the MODEM.

e. Bit 4 - Sending fill. This bit indicates a fill sequence is in progress.

f. Bit 5 - Data set ready. This bit indicates the presence of a DSR (Data Set Ready) signal from the MODEM.

g. Bit 6 - Sending break. This bit indicates that a transmission break is in progress.

h. Bit 7 - Clear to send. This bit indicates the presence of a CTS (Clear To Send) signal from the MODEM.

Model 1539 Programming Techniques.

The purpose of the following paragraphs is to outline the steps involved in programming the Model 1539 in the BSC mode. As of this writing, no programs have been prepared to use the 1539 in either the asynchronous or SDLC modes; therefore, the techniques have yet to be developed. The instruction sequences covered in this section are mainly for explanation purposes; they are not necessarily the best approach to programming the 1539.

Of course, the line protocol established for a specific communication system has a very significant effort on programming the system and, therefore, the programmer must become familiar with this information. The general information for operation in the BSC mode is given in an IBM document referenced in the Model 1535 description; for the SDLC mode, the reader should refer to IBM Synchronous Data Link Control General Information, number GA27-3093-1.

Organization of the Message Block

The first step in programming a transmission is to assign a buffer area for the message. The message to be transmitted should be assembled in memory before starting transmission. It must include all the control characters that are required for the complete message. For example: the two sync characters, start-of-text (STX) character, data, end-of-text (ETX) character, and two block check characters for the BSC mode. The only characters that are inserted by the hardware are the sync character as a fill character or, in transparent mode, the DLE (Data Link Escape) and sync characters as fill characters.

Transmitting Sequence

The Model 1539 must be initialized before data can be transmitted to a remote site. This requires the following steps:

a. Master clear

b. Load sync register

c. Load DLE register (transparent mode only)

d. Select word length and baud rate

e. Set full-duplex operation

f. Set transmit mode

First, the Controlled Clear, IOC,C#6;001, instruction with the accumulator equal to 013 will clear the 1539. Next, formats must be established. There is no restriction in the 1539 to limit the code used for the sync or DLE characters; the sync register and DLE register can be loaded with any code. There are, however, standard codes that are used for sync and DLE characters. The sync character is used in both the non-transparent and transparent modes of operation, while the DLE character is used only in the transparent mode. The standard sync and DLE codes are:

Sync	DLE
USASCII - 026 octal	USASCII - 020 octal
EBCDIC - 062 octal	EBCDIC - 020 octal
6-bit Transcode - 72 octal	6-bit Transcode - 77 octal

The sync register can be loaded by executing either the IOC,C#6;000, Load Sync Register, or the IOC,C#6;025, Load Fill Register, instruction. However, the IOC,C#6;025 instruction is the 1535-compatible instruction. The IOC,C#6;000 instruction is a 1539 instruction used in both synchronous and SDLC modes of operation. Both instructions achieve the same results.

Loading the data link escape (DLE) register is achieved by executing the Load Alternate Fill Register instruction (IOC,C#6;026). This instruction is a 1535-compatible instruction and is used only when the 1539 is operating in the synchronous mode. When loading the sync or DLE character, the respective character is first loaded into the accumulator and then the related instruction, IOC,C#6;000 or IOC,C#6;026 is executed.

A single eight-bit control word is used to select baud rate, word length, and parity. Refer to figure 3-32 for the control word format.

The 1539 is capable of operating at several baud rates. These rates are selected by bits 4 through 7 of the control word. The word length is selected by bits 2 and 3 of the control word and is 5, 6, 7, or 8 bits in length. Parity is selected by bits 0 and 1 and can be no parity, odd parity, or even parity.

When making up the control word, the entire network must be considered; that is, what the host processor or remote terminal are expecting on the line, and what they are sending over the line. The program must make up the control word and load it into the accumulator. Then, the Load Control Register instruction, IOC,C#6;004, transfers the contents of the accumulator to the 1539.

The control word is effective for both the receive and transmit modes of operation. It is not necessary to execute the instruction twice if operating in full-duplex mode.

After initial power application or after a controlled clear, the 1539 will be in the half-duplex mode. Therefore, on initialization, the only mode that must be selected is the full-duplex mode.

If the operation is to be full-duplex, the Set Full Duplex Operation instruction (IOC,C#6;001, accumulator equal to 003) must be executed. This prevents the receive mode from being cleared when the transmit mode is set and prevents the transmit mode from being cleared when the receive mode is set, thus enabling full-duplex mode of operation.

Execution of the Set Transmit instruction (IOC,C#6;021, accumulator equal to 000) puts the 1539 in the transmit mode and logically connects it to the MODEM. The 1539 will raise the Request to Send (RTS) interface signal to the MODEM. When the Clear to Send (CTS) is subsequently sensed, the transmit register empty status and associated interrupt signal are generated to notify the processor.

At this point, the initialization for the transmit sequence is complete, and program control would be transferred back to the mainstream program. The main program could be operating on any type of task; however, when the interrupt is generated by the 1539, a program sequence must be entered to supply a character, either control or data, to be transmitted.

The Write Data instruction (IOC,C#6;020) transfers the contents of the accumulator to the transmit "Q" register in the 1539 controller to be transmitted. Each time this register is empty, an interrupt is generated, and a character must be sent to the 1539 to be transmitted. This sequence continues until the entire message has been sent.

If data is not available at the transmit "Q" register for transmission, the 1539 automatically inserts a sync character (when operating in non-transparent mode) or DLE and sync characters (when operating in transparent mode). Both the sync and DLE characters are referred to as idle codes.

Receiving Sequence

Initialization for the receiving sequence is very similar to the transmitting sequence initialization. The differences are setting receive mode and end. Initialization requires the following steps:

a. Master clear.

b. Load sync register.

c. Load DLE register (transparent mode only).

d. Select word length and baud range.

e. Set full-duplex operation.

f. Set receive mode.

g. Set end after ETX has been detected.

The first five steps of the receiving sequence are the same as the transmitting sequence; therefore, the following explanation begins at step f.

Executing the Set Receive instruction (IOC,C#6;021, accumulator equal to 002) puts the 1539 in receive mode and logically connects it to the MODEM. This action is achieved through the following signals:

a. Activate Data Terminal Ready (DTR), which tells the MODEM that the 1539 is ready to send or receive data.

b. Look for synchronization, which causes the 1539 to search for the sync character in preparation to receive data.

c. Clear transmit - In half-duplex mode, clearing of transmit mode is necessary during a turnaround operation; that is, going from transmit to receive mode. In the full-duplex mode, the transmit mode is not cleared. In this mode, both receive and transmit states will remain active at the same time.

After execution of the Set Receive instruction, program control is transferred back to the mainstream program. However, the 1539 will start looking for sync characters and at some point start receiving data. The sync characters are not sent to the accumulator therefore, the first character to be received should be a start-of-text character. The STX character will cause bit 5 in the interrupt status word to be set to a one and an interrupt to be generated.

The interrupt handler determines that a character has been received and executes a Read Data instruction (IOC,C#6;022), which transfers the contents of the receive "Q" register to the accumulator. Then program control is again returned to the mainstream program to further process the data character. (The complete description of interrupt handling is covered under a separate paragraph.)

In certain programming conditions, the requirement may arise to load the sync character into the receive buffer area. This action may be necessary for two reasons: (1) in the event that a CRC character is equal to the sync character, and (2) in order to detect sync characters that are interspersed in the message to maintain synchronization. In order to establish the conditions necessary to transfer the sync character, the Set End instruction (IOC,C#6;021, accumulator equal to 016) must be executed.

If the data is not transferred out of the receive "Q" register as fast as data is received, a data overrun will occur. When this condition exists, bit 3 of the interrupt status word will be set to a one, and an interrupt is generated. The action taken under this condition depends upon the program. However, the remaining data in this block cannot be read with any validity; therefore, a retransmission may be in order. In any case, bit 3 will be reset to zero when a Search for Sync instruction (IOC,C#6;021, accumulator equal to 003) is executed.

Line Turnaround

When operating in half-duplex mode, the line must be turned around when changing from transmit to receive mode, or vice-versa. It is not necessary to go through the entire setup sequence to achieve line turnaround; the only instruction required to change from receive mode to transmit mode is the set transmit instruction. To go from a transmit mode to receive mode requires the execution of the Set Receive and the Search for Sync instructions.

Transparent Mode

The transparent mode permits the use of a wide range of coded data. This is achieved by treating the normally-restricted character codes as a bit pattern, thus permitting the unrestricted data coding that enables transmission of binary data, packed-decimal, machine-language programs, etc. When a control character is required in transparent mode, the character must be preceded by a DLE character to be recognized as a control character. This is shown below in figure 3-33.

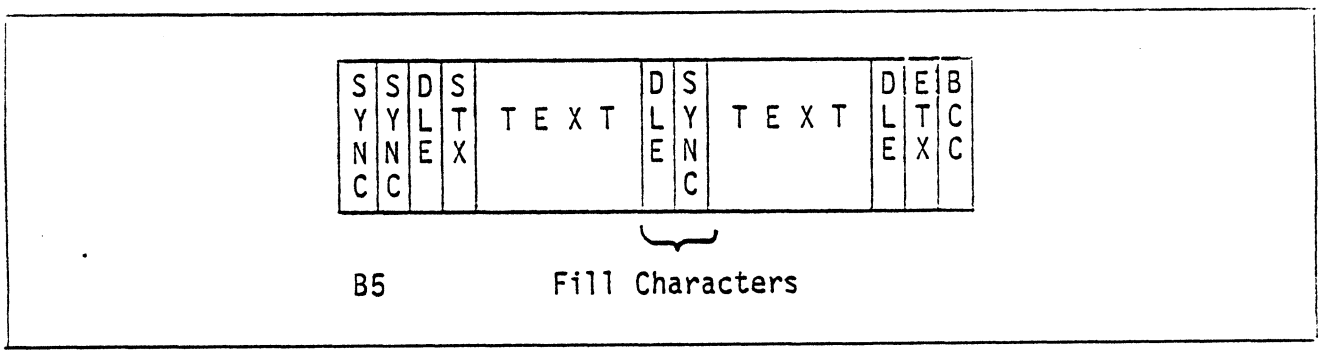


Figure 3-33. Use of the DLE Character in the Transparent Mode

There are three instructions used in conjunction with the transparent mode. They are Set Transparent Mode IOC,C#6;021 with the accumulator equal to 011, Reset Transparent Mode IOC,C#6;021 with the accumulator equal to 012, and Load Alternate Fill Register, IOC,C#6;026. Before writing a program for transparent mode, reference should be made to IBM System Reference Library publication entitled "General Information - Binary Synchronous Communications", order no. GA27-3004-2.

Use of the Timer

The 1539 is equipped with a one-quarter (1/4) second timer, which is program-controlled. The timer output is gated to bit 6 of the interrupt status word. When bit 6 is a one, the timer is inactive or has expired. When bit 6 is a zero, the timer is active.

The Set Timer instruction (IOC,C#6;021, accumulator equal to 006) is used to activate the timer. Once activated, the timer runs until it expires or is cleared by the Clear Timer instruction, IOC,C#6;001, accumulator equal to 006. The Clear Timer instruction also clears the status bit.

Use of the Secondary Channel

Some systems are equipped with the secondary channel capabilities. The advantage of the secondary channel is that it allows the transmitting terminal to send continuous messages with the assurance that if an error occurs, the receiving station will signal the transmitting station via the secondary channel. The operation of the secondary channel is under program control at both the transmitting and receiving stations.

Two instructions and a status bit are used when programming the secondary channel. The instructions are Mark Secondary Channel (IOC,C#6;021, accumulator equal to 004) and Space Secondary Channel (IOC,C#6;021, accumulator equal to 005). The status bit is bit 0 of the interrupt status word. The status bit, when a one, indicates to the transmitter that an error has occurred; this bit is generated when the secondary channel is activated at the receiving station by the Mark Secondary Channel instruction.

Example of a Transmit Routine

Initialization for transmission in the BSC mode requires the following steps:

a. Master clear.

b. Load sync register.

c. Load DLE register.

d. Select baud rate, word length, and parity.

e. Set transparent mode.

f. Set transmit.

Refer to table 3-8, program label "SUP". Before program control is transferred to program label "SUP", the transmit data buffer should be established. Depending on the system configuration, it may be necessary to inhibit interrupts. In either case, the clear command will clear the 1539. This will clear all registers and controls in the controller.

As mentioned earlier, the sync and DLE characters are not restricted to a fixed code combination; therefore, any code can be loaded into the respective register. These codes must be unique to the entire data link. That is, all units communicating with each other must be set up to recognize the same sync and DLE characters. The Load Sync Register instruction (IOC,C#6;025) loads the sync register with the contents of the accumulator, and the Load DLE instruction (IOC,C#6;026) loads the DLE register with the contents of the accumulator.

Baud rate, word length, and parity are selected by placing the control word in the accumulator and executing the Load Control Register instruction (IOC,C#6;004). The control word of 003 will select a baud rate of 1200, word length of 8 bits, and odd parity. Refer to figure 3-32 for a complete description of the control word.

This example assumes the transparent mode of operation. However, transparent mode does not have to be set at this point in the sequence. The programmer may desire a separate sequence for convenience.

At this point, program control is transferred back to the mainstream program, where a decision is made as to transmit or receive mode. In this example, program control would be returned to program label "TRM", the set transmit mode instruction sequence.

After the transmit command (IOC,C#6;021, accumulator equal to 003) is sent to the controller, the controller will logically connect to the MODEM. When this connection is complete, the controller is ready to start sending data. Because the transmit "Q" register is empty, an interrupt will be generated.

To understand the sequence of events, we must go back in time to the point at which the Set Transmit instruction was sent to the controller. It will take a period of time before the controller can connect to the MODEM. In the meantime, program control is transferred to the mainstream program. The main program must enable interrupts if it has not already done so. With interrupts enabled, the mainstream program can continue with no concern about the communications controller. When the 1539 controller has connected to the MODEM, it will generate the interrupt and set bit 7 in the interrupt status word to a one.

The interrupt forces program control to page 03, location 000. At this location is a disable interrupt instruction, which will prevent any further interrupts until the current interrupt has been processed. The next instruction will achieve a branch to the interrupt handler routine. The interrupt handler will then get the status word via a Read Primary Status instruction (IOC,C#6;024). After the status word has been loaded into the accumulator, the interrupt routine will examine each bit. Refer to table 3-9, Interrupt Handler Routine, as required. In this example, bit 7 is equal to a one. When bit 7 is checked, program control is transferred to program label I05.

At program label I05, the sequence of instructions will send a total of six sync characters. Of course, six interrupt sequences similar to the one just covered will be executed.

With the sync characters transmitted, the next character will normally be the Start-of-Text (STX) character. The STX character is strictly a program control character; the 1539 does not recognize STX as anything except data. In the case of transparent mode, the STX character must be preceded by a DLE character. Again, each character that is sent is sent because the transmit "Q" register was empty, which generated an interrupt.

This sequence will continue for the entire message. After the message has been sent, the program will either transmit another message, go into receive mode to receive a response, or terminate transmission.

Example of a Receiving Routine

If the receive sequence is entered following a transmit sequence and the transmit sequence was not terminated, the only instruction required to change modes is the Set Receive instruction (IOC,C#6;021, with the accumulator equal to 002). Initialization for receive mode requires the following steps:

a. Master clear

b. Load sync register

c. Load DLE register

d. Select baud rate, word length, and parity

e. Set transparent mode

f. Set receive

The initialization sequence for receiving is the same as for transmitting, through and including the set transparent mode instructions. Refer to table 3-8, program label SUP. Again, program control returns to the mainstream program after transparent mode is set. Up to this point, the sequence is common to both transmit and receive; therefore, the main program must make the decision as to transmit or receive.

Program control will be transferred to program label SRR to set up receive mode. Sending the receive command to the controller will achieve the following:

a. Activate Data Terminal Ready - DTR. When active, tells the MODEM that the 1539 controller is ready to send or receive data.

b. Clear transmit - The clearing of the transmit mode is necessary during a turnaround operation; that is, going from transmit to receive mode when operating in half-duplex. If the sequence had started from an initialized condition, transmit would already be cleared.

The Resync command causes the controller to look for the sync character in preparation to receive data. These sync characters are not sent to the accumulator; the first character to be sent to the accumulator is the Start-of-Text character (STX).

Program control will first be transferred back to the mainstream program, where interrupts are enabled. Upon receiving the STX character, the 1539 sets bit 5 in the interrupt word and generates an interrupt.

Each character is handled in the same way. When a character is received into the receive Q register, bit 5 of the status word is set and an interrupt generated.

The BCC will be the last character of the block or buffer to be received. It is possible that the code combination for the BCC is the same as for a sync character, in which case the 1539 would not normally send it to the accumulator. The Set End instruction, IOC,C#6;021, accumulator equal to 016, will inhibit decoding of the sync character, thereby permitting the transfer of all characters from the receive Q register to the accumulator.

In some applications, the programmer may insert sync characters within the data block to maintain synchronization. In this case, the Set End instruction is used at the start of receiving the data block. Refer to program label "END".

Example of Interrupt Handling

In order to understand the concepts involved in programming the 1539, the programmer must have a basic understanding of program interrupt procedure. An interrupt is generated by the 1539 to signal the Series 1500 Intelligent Terminal that a condition exists that requires attention. This interrupt causes program control to be transferred to page 03, location 00. This location must contain the start of a linkage routine that links to the interrupt handler.

There are four conditions that cause the interrupt: (1) transmit register empty, (2) receive register full, (3) control timer expired, and (4) a significant status change. Along with generating an interrupt, each of these conditions sets a related bit in the interrupt status word.

The interrupt handler must execute a Read Primary Status instruction (176-024) to obtain the status word. It must then examine the status word to determine which condition causes the interrupt and thus, the course of action that must be taken.

The complete breakdown of the interrupt status word and the response is as follows:

a. Bit 7 - Transmit Holding Register Empty. This signal is present when the transmitter can accept another byte. This interrupt stays active until corrective action is taken, normally, a Write Data instruction.

b. Bit 6 - Timer Expired. This signal is presented whenever the 250-millisecond timer is expired. It stays active until a Clear Timer or Start Timer instruction is executed.

c. Bit 5 - Receive Holding Register Full. This signal is present when a data byte is available for transfer to the accumulator. It stays active until a Read Data instruction is executed.

The remaining bits are the status change bits. Bit 2 does not cause an interrupt, therefore, it is not included in the following descriptions.

a. Bit 4 - Transmission Shutdown/Abort. This signal is active whenever a shutdown of the transmitter occurs; this may be predetermined or the result of a hardware abort.

b. Bit 3 - Receive Error. This bit is active whenever a parity error, framing error, or receiver overrun error occurs.

c. Bit 1 - Ring Indicator. This bit is active whenever the ring indicator signal is present from the MODEM and the Data Terminal Ready (DTR) signal is not active. It indicates that an incoming call has been received.

d. Bit 0 - Secondary Channel Transition Detected. This bit is activated when any change of state is detected on the secondary channel. It is normally used when the receiving station is to signal the transmitting station.

When the interrupt is generated by the 1539, program control is transferred to page 03, location 00. Located there is a link to an interrupt handler routine, an example of which is shown in table 3-9. Refer to table 3-9 and examine the instructions through program label "TKS".

The first instruction disables any further interrupts until the present interrupt has been handled. Then the interrupt handler routine is entered.

When the interrupt is generated, there is no way to know where the main program is in its sequence. Therefore, the contents of the accumulator and the processor status must be saved to enable the mainstream program to continue from the point at which it was interrupted after the interrupt has been handled.

The first requirement of the interrupt handler is to store the accumulator and the status of the processor. The "queue switch", which is set if a character is queued for transmission, is cleared. If a character had been in the accumulator to be transmitted, that character would be stored in location "SVA" until after the interrupt is handled.

At this point, the housekeeping operation is complete and the handler can proceed to get the status word from the communications controller. As mentioned above, the interrupt is generated as a result of any of the seven conditions in the controller. By examining the status word, the handler will determine the cause of the interrupt.

Each of the bits in the status word is examined with a test mask and jump instruction; if the test is satisfied, a routine is entered to handle the condition. In some cases, a code is loaded into the accumulator to inform the mainstream program of the status of the controller. This is, of course, done before program control is transferred back to the mainstream program. Refer to table 3-9, program label "TKS", through and including program label "XMI".

Before program control can be transferred back to the mainstream program, certain housekeeping functions must be completed. These functions include:

a. Check the "queue switch" to determine if a character was queued for transmission.

b. Correct the last stack pointer address of the maintenance program.

c. Restore the condition register and the accumulator.

d. Enable interrupts.

Refer to table 3-9, program label "RET", the first four instructions. If a character had been queued for transmission when the interrupt occurred, the queue switch would have been set. The character would have been stored in the "SVA" area at the start of the interrupt handler. In this situation, the only requirement of the interrupt return sequence is to restore the accumulator and return program control by the exit instruction.

When the interrupt was generated by the 1539 and program control was transferred to page 03, location 00, an automatic stack and branch operation was executed. Therefore, the exit instruction will return program control to the mainstream program.

If the queue switch is not set, then further steps must be taken before program control is transferred. The stack pointer address will have to be decremented; this involves a check for wrap-around and a page overflow check. The accumulator and the arithmetic condition registers have to be restored; their contents were saved during the start of the interrupt handler routine.

Before the stack pointer address can be decremented, it must be located. There are 16 stack pointer addresses, and any one of the 16 could be the pointer to be decremented. To locate the proper one, the program executes a "LPS", Load Processor Status instruction, which loads the stack pointer into the accumulator. The stack pointer is a number from 00 to 17 octal. As the reader will recall from the processor instruction description, this number "points" to a pair of memory locations in page 00, between 40 and 77 octal.

Refer to the three instructions, starting with the LPS instruction, in the return sequence. The LPS instruction loads the current stack pointer into the accumulator. The SAN instruction will first shift the pointer, which effectively multiplies it by two, then masks the value with a 36 octal. The ADA instruction adds a 36 octal to this quantity. This combined action forms a memory address that is two less than the current stack pointer. For example, if the current stack pointer was 3, the memory addresses would be 46/47. After modification, the new addresses would be 44/45, which is stack pointer 2.

The next check that must be made is for stack wrap-around. This is achieved by comparing the results of address modification to 36 octal. If the modified address is less than 36 octal, wrap-around did occur, in which case an octal 76/77 will be used to address memory.

In either case, the memory address is loaded into index register 7. It is then loaded into the accumulator and decremented by 2.

A check for underflow is made next to determine if the page count must be decremented. If this action is required, the page count is loaded into the accumulator and decremented by one.

With the stack pointer address modified, only the condition register and the accumulator are left to be restored. The contents of both of these registers were stored in the first part of the interrupt handler routine; therefore, it is just a matter of reloading the registers.

The last item before the return exit is to enable interrupts to allow another interrupt to be processed if and when it does occur.

Table 3-8. Example of Model 1539 Transmit and Receive Routines

```
                              SETUP

SUP:  LDA,R#0;OCT:013          General Clear Command
      IOC,C#6;021              Issue Command
      LDA,R#0;SYN              Load Sync Character
      IOC,C#6;025              Issue Command
      LDA,R#0;DLE              Load DLE Character
      IOC,C#6;026              Issue Command
      LDA,R#0;OCT:003          Setup 1200 Baud, 8 bits, and odd parity
      IOC,C#6;004              Issue Command
      LDA,R#0;OCT:011          Load Transparent Command
      IOC,C#6;021              Issue Command
      EXU,000                  Return

                        SET TRANSMIT MODE

TRM:  LDA,R#0;OCT:000          Set Transmit Command
      IOC,C#6;021              Issue Command
      EXU,000                  Return

                     SET RECEIVE AND RE-SYNC

SRR:  LDA,R#0;OCT:002          Receive Command
      IOC,C#6;021              Issue Command
      LDA,R#0;OCT:003          Re-Sync Command
      IOC,C#6;021              Issue Command
      EXU,000                  Return

                              SET END

END:  LDA,R#0;OCT:016          Set End Command
      IOC,C#6;021              Issue Command
      EXU,000                  Return

                        TRANSMIT SYNC CHARACTER

I05:  ADX,R#5;OCT:001          Inc. Sync Que Switch
      CPX,R#5;OCT:006          Have All Syncs Been Sent
      BRH,T15                  Yes
      IOC,C#6;005              Send Contents of Primary Fill Register
      EXU,000
```

Table 3-8. Example of Model 1539 Transmit and Receive Routines (Continued)

```
                    TRANSMIT DLE CHARACTER

DLE:  ADX,R#5;OCT:200          Set Que Switch
      LDA,R#0;DLE              DLE Code
      BRU,I20                  Go Transmit DLE

                    TERMINATE

TRM:  LDA,R#0;OCT:001          Terminate Command
      IOC,C#6;001              Send Command

                    RECEIVE DATA

RCV:  IOC,C#6;022              Receive Character
      EXU,000                  Return

                    TRANSMIT DATA

T15   LDA,R#0;CMD              Control Byte from Mainstream Program
      CPA,R#0;OCT:040          Should DLE Be Sent
      BRE,DLE                  Yes. Go Send DLE
      CPA,R#0;OCT:020          Data to be Sent
      BRE,T16                  Yes. Go Send Data
      CPA,R#0;OCT:010          To Terminate
      BRE,TRM

T16   LDA,R#0;DAT              Data to be Transmitted

T20   IOC,C#6;020              Write Data
      EXU,000                  Return
```

Table 3-9. Model 1539 Interrupt Handler Routine

```
P03-000   DPI,000                    Disable Interrupt
P03-002   SMS,S#3                    Set Mem. Section
P03-004   BRU,CIH                    Comm. Interrupt Handler

                        INTERRUPT HANDLER

          CIH:  SMC,C#2              Enable Registers in all sections
                STA,R#0;SVA          Save Accumulator
                LPS,000;             Get Processor Status
                STA,R#0;SVC          Save Processor Status
                LDA,R#0;OCT:000      Set Off
                STA,R#0;QQS          Queue Switch

          TKS:  IOC,C#6;024          Get Comm. Status
                TMJ,+04;OCT:010      Receive Error
                BRU,*+8              No Continue
                SBU,SRR              Re-Sync Line
                LDA,R#0;(B)          Status Code
                BRU,GOR+2            Go Return
                TMJ,+02;OCT:040      Receive "Q" Ready
                BRE,RXI              Yes-Go
                TMJ,+04;OCT:001      Second Channel
                BRU,*+8              No-Continue
                SBU,SRR              Re-Sync Line
                LDA,R#0;(5)          Status Code
                BRU,GOR+2            Go Return
                TMJ,+02;OCT:020      Xmit Abort
                BRE,RET              Yes-Go Return
                TMJ,+04;OCT:002      Go Ring Indicator
                BRU,CKT              Not Ring Indicator
                LDA,R#0;CMD          CCB Command
                TLJ,+04;OCT:001      Look for RI Cmd
                BRU,RET              Return Int.

          GOR:  LDA,R#0;(1)          OP Comp.
                STA,R#0;STS          TO Status
                BRU,RET              Return Int.
```

Table 3-9. Model 1539 Interrupt Handler Routine (Continued)

INTERRUPT HANDLER (Continued)		
CKT:	TMJ,+04;OCT:100	Timer Expired
	BRU,XMI	Not Timer
	SBU,CTM	Clear Timer
	LDA,R#0;STS	Status of CCB
	TLJ,+04(0)	Go if Busy
	BRU,RET	Return
	LDA,R#0;CMD	CCB Command
	TLJ,+04;OCT:003	Go Xmit CTL Command
	BRU,CM4	Not Xmit CTL
	ADX,R#3;OCT:001	Increment Timer Count
	LDA,R#0;R#3	Get count
	CPA,R#0;SCC	Match Sec CTL
	BRE,*+8	Yes-Timeout
XMI:	TMJ,+04;OCT:200	Go Xmit Int.
	BRU,RCI	Not Xmit Int.
	LDA,R#0;CMD	Control Byte from Mainstream Program
	TLJ,+04;(0)	Go Comm. Active
	BRU,TRM	Return
	CPA,R#0;OCT:004	Transmit Data
	BRE,I05	Go Transmit
	BRU,ERR	Error Routine because of No Match
SET RECEIVE AND RE-SYNC		
SRR:	LDA,R#0;OCT:002	Set Receive Command
	IOC,C#6;021	Issue Command
	LDA,R#0;OCT:003	Re-Sync Command
	IOC,C#6;021	Issue Command
	LDA,R#0;OCT:012	Reset Transparent Mode
SR2:	IOC,C#6;021	Issue Command
	EXU,000	Return

Table 3-9. Model 1539 Interrupt Handler Routine (Continued)

RETURN FROM INTERRUPT

```
RET:  LDA,R#0;QQS          Mini-Queue Switch
      TLJ,+06;OCT:000      Go Not Set
      LDA,R#0;SVA          Restore Accumulator
      EXU,000              Return Caller
      LPS,000              Get Processor Status
      SAN,S#7;OCT:036      {
      ADA,R#0;OCT:036      { Current Stack Base Minus 2
      CPA,R#0;OCT:036      Check for Stack Wrap
      BRH,*+4              No-Continue
      LDA,R#0;OCT:076      Last Stack Location
      STA,R#0;R#7          Load X7
      LDA,R#7;P00          Get Stack Pointer Address
      SVA,R#0;OCT:002      Subtract 2 from Location Address
      CPA,R#0;OCT:376      Check for Page Underflow
      BRL,*+8              No Bypass Page Decrement
      STA,I#7;P00          Store Location Address
      LDA,R#7;P00          Get Page Address
      SVA,R#0;OCT:001      Decrement Page Address
      STA,R#7;P00          Store New Page Address
      LDA,R#0;SVC          Load Condition Code of Mainstream Program
      SAC,000              Restore Condition Register
      LDA,R#0;SVA          Restore Accumulator with Mainstream
                             program data
      EPI,000              Enable Interrupts
      EXU,000              Return to Mainstream Program
```

3-11. PRINTERS.

There are two classes of printers available in the Series 1500 line, matrix printers and line printers. Models 1525/1526 and 1551 S/D are matrix printers while the Models 1553 through 1556 are line printers. All models operate on the serial input-output channel and have related program segments in the Standard I/O Library.

Model 1525/1526 Printer.

The Model 1525/1526 uses a 5 by 7 dot matrix system of printing, and attains speeds up to 33 characters per second. As a result of the matrix approach to printing, the printers are not restricted to a particular code or character set. The printing of each dot in the matrix is entirely under program control. When printing under control of the standard background routine, the standard dot pattern matrix (SDP) is used to generate the characters for this printer.

The Model 1525 has a platen that allows up to 132 characters per line, whereas the Model 1526 has a platen that only allows up to 80 characters per line. Both printers are able to accommodate friction-feed, single-copy journal roll or up to five copies of pin-feed, fan-fold paper.

Model 1525 Instructions.

It was mentioned above that the printers all operate on the serial I/O channel and thus receive both data and commands over this line. The information to be sent is first placed in the accumulator and then sent to the I/O controller for subsequent transfer to the printer. Whether the information is identified as data or as a command depends upon the SIO channel instruction that caused its transfer. The Transmit Control Byte instruction is used to send commands. In this case, the contents of the accumulator are a command for the printer. There are five such commands that can be issued to the Model 1525/1526 printers. They are as follows:

a. 002-Left Margin (no line feed). This command returns the print carriage to the left margin. No line feed is generated.

b. 004-Carriage Return (line feed). This command returns the print carriage to the left margin and simultaneously generates a line feed command.

c. 012-Line Feed Only. This command advances the paper through the printer one line. The print carriage will be positioned one line below its previous position after execution of this command.

d. 014-Incremental Mode. This command prepares the printer to operate in a character-by-character mode. Maximum printing speed in the incremental mode is 16-2/3 characters per second.

e. 050-Clear. This command clears the underflow status indicator and the incremental mode status. In addition, it halts any printing operation in progress.

Model 1525 Status Word.

Handling of the status words returned by peripheral units was discussed in detail earlier in this section in the description of the serial input/output channel. Since bits 6 and 7 of the status word are used to indicate errors in the serial I/O channel circuits (bit 6 is addressing error and bit 7 is I/O error), only bits 0 through 5 are available to indicate printer status. The meaning of each of these bits is described below. Note that a one in the respective bit position indicates the condition described.

a. Bit 3 - Printer Stopped. This signifies that the printer is stopped either because it is out of paper or because the cover is not closed.

b. Bit 2 - Underflow. This indicates that a print error has occurred because a byte of print data was not received in time to be included in the print cycle. This status bit must be reset by the execution of a clear instruction.

c. Bit 1 - Busy. This indicates that the printer is performing an operation. The conditions that will set this bit are:

   (1) The printer is performing a print operation.

   (2) The printer is performing a carriage return.

   (3) The printer is performing a line feed.

d. Bits 0, 4, and 5 are not used.

Model 1525 Programming Methods.

When printing, the programmer has the option of writing a routine to perform the print function or using the standard background function, "MXP". In this explanation of programming methods, some of the basic coding from the background is used to explain the programming procedure.

The concept of matrix printing will be covered before the programming involved in printing a character is discussed. A printed character is formed by a combination of dots; these dots, in turn, are printed as a result of the matrix bit pattern. Figure 3-34 shows the blow-up of the printed character "A". Five patterns must be used to form this character. Therefore, five patterns must be sent from the terminal via the SIO line to the printer. Relate the print character "A" to the CRT dot pattern character code in memory location 015 in pages 04 and 05. Notice that seven bits are used in the character columns and five bits in the character rows.

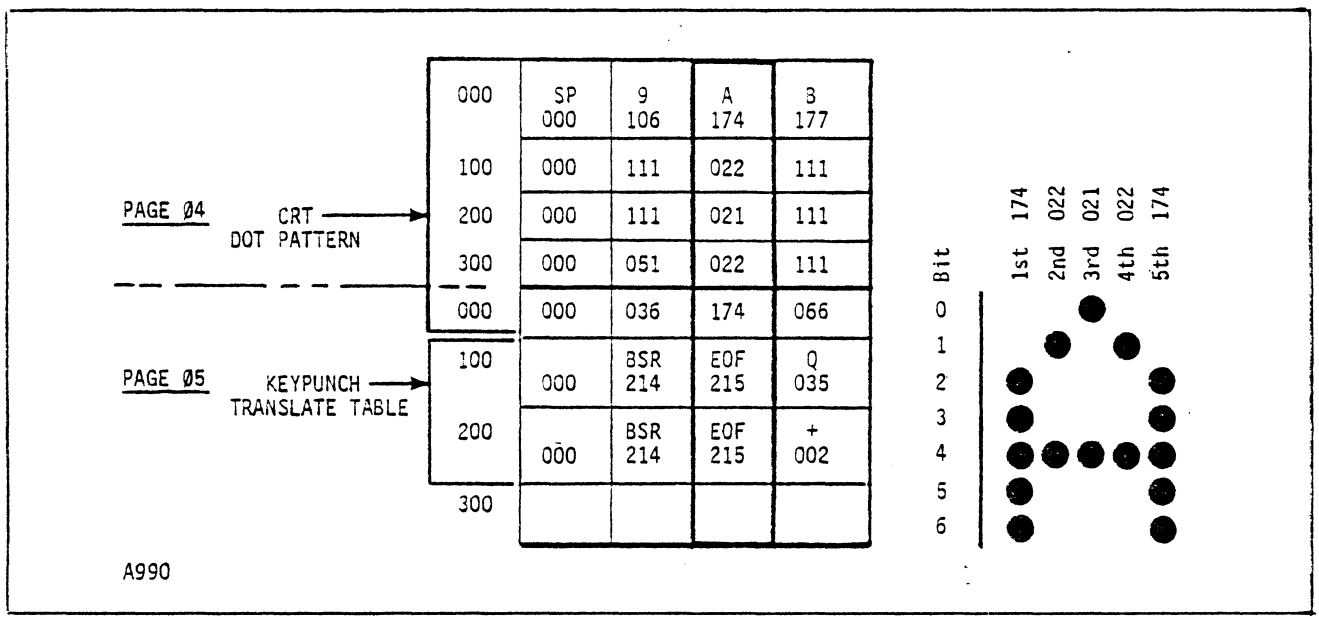


Figure 3-34. Example of a Printed Character

Prior to execution of a print function, the print buffer must be loaded with the data to be printed. In this example, the print buffer is established in page 01, and the following steps are required to print a character:

a. Address printer.

b. Check status.

c. Send dot pattern (five times).

d. Check status.

Table 3-10 lists a print routine. Program control is transferred to label PF1 to address the printer. In turn, the stack and branch instruction transfers control to label "D98". First, the Inhibit Line instruction clears all units connected to the SIO line. An address of 013 octal is used in this case as the address of the printer. The 200 that is added to the address is used to form the complete address word of 213. Bits 6 and 7 are used to notify the SIO channel of an address word. Refer to figure 3-25 for the format if a review is necessary.

Table 3-10. Examples of Model 1525 Routines

Print Function		
PF1:	SBU, D98.	Address Printer
	SBU, D96+2.	Check for Error
D95:	SBU, D96.	Check Printer Status
	LDA, I#3;P01.	Load Data (R#3 starting address)
	ANA, R#0;OCT:077.	Mask Out Unprintable Character
	STA, R#0;R#4.	Save Data Byte
	LDA, R#4;P04.	Load Dot Pattern Character
	IOC, C#5;002.	Send Matrix and Print
	SBU, D96.	Check Status
	ADX, R#4;OCT:100.	Add Pointer Bit for Next Dot Pattern
	CPX, R#4;OCT:077.	Check for End of Character Matrix (end page 04)
	BRH, *-10.	Not End of Page 04 Loop
	LDA, R#4;P05.	Last Byte of Matrix is in Page 05
	IOC, C#5;002.	Print Last Matrix of Character
	CPX, R#2;OCT:000	Check for Increment Mode
	BRE, *+14.	Yes - 1 Character, No Carriage Return

Table 3-10. Examples of Model 1525 Routines (Continued

Print Function			
	SUX, R#2;OCT: 001.	Decrement Size	
	CPX, R#2;OCT:000.	Check for End	
	BRH, D95.	No - Then Repeat to Print Next Character	
	SBU, D96.	Check Ready	
	LDA, R#0;OCT:004.	End of Line Carriage Return	
	SBU, D97.	Send Command	
	LDX, R#1;OCT:002.		
	BRU, END	End	
I/O Channel Address Function.			
D98:	IOC, C#5;010.	Set Master Mode	
	IOC, C#5;006.	Inhibit Line	
	LDA, R#0;OCT:013.	Load Address	See Note
	ADA, R#0;OCT:200.	Insert High Order Tag	
	IOC, C#5;007.	Set Device Address	
	IOC, C#5;000.	Start Transmit	
	BRU, D97.	Delay 3 microseconds	
Form Control Function.			
D99:	SBU, D98.	Address Unit	
	LDA, R#0;R#4.	Load Command	
	TLJ, +06;OCT:050.	Clear Command	
	SBU, D96.	Test Ready	
	LDA, R#0;R#4.	Load Command	
	BRU, D97.	Send Command	

Note: The time delay required after the Inhibit Line instruction is satisfied by the execution time of these instructions.

Table 3-10. Examples of Model 1525 Routines (Continued)

Printer Status Check Function.		
D96:	SBU, *+08.	Check Status
	TLX, 000;OCT:000.	Status is OK
	TLJ, -04;OCT:002.	Not Ready
	EXB, EXT.	Go to Error Routine
	LDA, R#0;OCT:010.	Status Command
D97:	IOC, C#5;003.	Transmit Control Byte
	IOC, C#5;001.	Receive Transfer Byte Unconditional
	STA, R#0;R#1.	Save Status
EXT:	EXU, 000.	Return Exit
End of Print Function.		
END:	SMS, S#0.	
	SBU, D93.	Delay 5.6 milliseconds
	SBU, D97-02.	Load Status
	ANA, R#0;OCT:002.	Drop Ready Bit
	SBU, D96+02.	Check for Error
BMP:	EXU, 000.	

The IOC,C#5;007 instruction transfers the address to the SIO circuits to be sent out on the SIO line and to the printer. The Start Transmit instruction IOC,C#5;000 releases the line, which was held in the clear condition by the inhibit instruction. Then, at program label "D97", the Transmit Control Byte instruction IOC, C#5;003 sends the address to the printer. In response, the printer returns its status word, which is accepted by the Receive Transfer Byte instruction IOC, C#5;001.

Program control is returned to PF1+2 by the exit instruction. Then, at program label D96+2, the status word is checked for line errors. The exit returns control to label D95, where the stack and branch moves control to D96, and a second stack and branch moves control to the load accumulator instruction, which loads the read status command. The next sequence of instructions will acquire the printer status word. This command is necessary to obtain the status after each column of a character is sent to the printer. Because of the relatively slow speed of the printer, the program could send data at too fast a rate. To determine when the printer is ready for a character column, a check status instruction is executed and a check for the printer being ready is made. The program remains in this loop until the printer is ready to receive another character column.

At program label D95+2, the first character of the print buffer is loaded into the accumulator. For this example, the character is assumed to be an "A"; the code for an "A" is 015 octal. The next instruction checks the character code with a 77 octal. For the character to be printable, the code must be in the range of 000 through and including 077. The patterns for the character "A" are in four locations of page 04 and one location in page 05. The next sequence of instructions loads the first column into the accumulator and transfers it to the printer. After each column is sent to the printer, the status is checked. Assuming the status is good, program control returns to label D95+14. The address of 015 is used for the first column, so 100 is added to this address to get 115, which is the address of the second column. The address is checked against 077 to determine when the last address in page 04 is reached. When the 100 is added to the 315, an overflow occurs, and the address will again be 015. At this point, the next column is read from page 05, location 015.

When the last column is sent to the printer, a check is made to determine if the mode is "increment". If the increment mode switch was set, the print function is complete. If not, the buffer size is decremented and a check is made to determine if the complete buffer has been printed. This sequence is repeated for each character in the buffer.

The next subject to be discussed is the sending of commands to the printer. The subroutine to send these commands to the printer starts at program label D99. The command code must be in register R#4 when this routine is entered. The sequence for sending a command is very similar to sending data; that is, the unit must be addressed, the status is checked, and the command code is sent to the printer if the status check was good.

Model 1551 S/D Printer.

The Model 1551 Printer also employs the dot matrix printing technique. The standard character is made of a 5 by 7 dot pattern, and an elongate function will print a 10 by 7 dot pattern. This is normally used to print headings. There are two models of the 1551; the 1551S uses a single print head and is capable of printing 60 lines per minute, while the 1551D uses a dual head and prints 125 lines per minute.

The form movement is controlled by a two-channel vertical format tape, which operates the feed mechanism. In turn, the vertical format tape is operated by the program in the intelligent terminal. Various sizes and thicknesses of pin-fed forms can be accommodated by both models, and both are capable of printing up to 132 characters per line.

Model 1551 Instructions.

The instructions sent to the Model 1551 are transferred over the serial I/O channel in the same manner as those for the 1525/1526 printers. A command is placed in the accumulator and sent by a Transmit Control Byte instruction. There are 10 such commands available for the 1551, as follows:

a. 007 - Bell Alarm. Sounds a 3-second audible alarm at the printer.

b. 012 - Feed One Line of Paper. Advances the paper vertically one line.

c. 013 - Vertical Tab. Moves the form to the next channel 5 punched in the printer carriage control tape.

d. 014 - Form Feed. Moves the form to the next channel 7 punched in the printer carriage control tape. The standard function of this feature is to advance the form vertically to the next top of form (line 1).

e. 015 - End of Line or Carriage Return. Causes the printer to fill the balance of the print buffer in the printer with spaces and initiate a print cycle.

f. 016 - Elongate. This feature prints each of the 5 horizontal matrix character values twice, producing a 10 x 7 character instead of the normal 5 x 7 character. When using this feature, the complete line will be elongated. The maximum number of characters that may be elongated is 66.

   When using the printer Model 1551D, the characters to be elongated will have a starting address of 000 for head 1, which will elongate the first 33 characters, and 066 for head 2, which will elongate 33 or more characters starting at that address.

g. 021 - Select. This feature turns the printer on-line, which makes it available to receive data.

h. 023 - De-Select. This feature turns the printer off-line, which prevents the printer from receiving data.

i. 050 - Clear. This feature clears the interface status. It is used prior to the Select command, and it acts as a general clear to the printer.

j. 177 - Delete. Clears the printer buffer and initializes the printer electronics. This may be used to reset the printer electronics after an error has occurred.

Model 1551 Status Word.

Only three of the bits in the status word available from this printer actually apply to printer operations. Bits 6 and 7 indicate serial I/O channel conditions (bit 6 addressing error, and bit 7 I/O error), and three bits (2, 4, and 5) are not used. The meaning of the three status bits available is discussed below. Note that each bit is a one when it indicates the condition described.

a. Bit 3 - Printer Error. Indicates that the printer is out of paper or that the cover over the print mechanism is open.

b. Bit 1 - Printer Busy. Indicates that the printer is engaged in a print operation and is not yet ready to accept a new character.

c. Bit 0 - Select Switch. This bit indicates the position of the select switch on the printer. A one indicates that the printer is on-line; a zero, off-line.

Model 1551 Programming Methods.

When performing the print functions, the programmer has the option of using the standard background function, "LNP", or writing the program. The program that will be used in this example is basically the same as the one used by the standard background routine. All the line printers, 1551 through 1556, can be controlled by the sample program. That is not to say that the sample program is complete in all aspects; it is only a guideline to proper programming.

All of the line printers use the USASCII code as their print code. Therefore, a translation table must be available to translate the Series 1500 code to USASCII. There is a standard background translation table, CT1, which is loaded into page 05, locations 300 through 377. This table is used in the following example; however, the programmer may wish to generate a table instead of using the "CT1" table.

Prior to execution of a print function, the print buffer must be loaded with the data to be printed. In this example, the print buffer area is initiated as page 01; however, any unused area can be designated as the print buffer area. With the buffer setup, the following steps are required to print a line:

a. Address printer.

b. Check status.

c. Send characters to printer (for one line).

d. Check Status.

Program control is transferred to program label "PF1" to start the print function; then, to program label D98 to address the printer. (Refer to table 3-11.) The Inhibit Line instruction (IOC, C#5;006) in segment D98 clears all units connected to the SIO line. In this example, the address of 013 octal is used as the printer address. The 200 is added to form the complete address of 213. Bits 6 and 7 are used to notify the SIO channel that this is an address word. (The address word format is shown in figure 3-25.)

Table 3-11. Examples of Model 1551 Routines

Print Function		
PF1:	SBU; D98.	Address Printer
	SBU; D96+4.	Check for Error
D95:	SBU; D96.	Check Status
	LDA, I#3;P01.	Load Data
	IRA, R#0;OCT:300.	Adjust Pointer (table starts at P05, loc. 300)
	STA, R#0;R#5.	Save Data Byte
	LDA, R#5;P05.	Translate Byte
	IOC, C#5;002.	Print Character to Printer
	SUX, R#2;OCT:001.	Decrement Size (R#2 is buffer size)

Table 3-11. Examples of Model 1551 Routines (Continued)

	Print Function	
	CPX, R#2;OCT:000.	Check for End of Buffer
	BRH; D95.	No, then Repeat Sequence
	SBU; D96.	Check Status
	LDA, R#0;OCT:015.	End of Line
	IOC, C#5;002.	Transmit Data Command
	BRU, BMP.	Bump Current Line
D96:	LDA, R#0;OCT:010.	Load Status Command
	SBU; D97.	Go Send Command
	TLX, 000;OCT:000.	Yes? Ready
	TLJ, -06;OCT:002.	Not Ready Loop
	CPA, R#0;OCT:000.	Set High
	EXB; EXT.	Error Return
	I/O Channel Addressing Function	
D98:	IOC, C#5;006.	Inhibit Line
	LDA, R#0;OCT:013.	Load Address
	ADA, R#0;OCT:200.	Insert High Order Tag
	IOC, C#5;007.	Set Device Address
	IOC, C#5;000.	Start Transmit
	ADA, R#0;OCT:000.	Delay 3 microseconds
D97:	IOC, C#5;003.	Transmit Command
	IOC, C#5;001.	Receive Status
	STA, R#0;R#1.	Save Status
EXT:	EXU; 000.	Return Exit

Table 3-11. Examples of Model 1551 Routines (Continued)

	Form Control Function	
D99:	SBU; D98.	Address Unit
	LDA, R#0;R#4.	Load Command
	TLJ, +12;OCT:020.	1555/1556
	TLJ, +10;OCT:050.	Clear Command
	SBU; D96.	
	LDA, R#0;R#4.	
	IOC, C#5;002.	Send Command
	LDA, R#0;OCT:010.	Get Status Command
	BRU; D97.	Go Send Command
BMP:	EXU; 000.	

The Set Device Address instruction (IOC,C#5;007) transfers the address to the SIO circuits in the I/O controller in preparation to be sent out on the SIO line and to the printer. The Start Transmit instruction (IOC,C#5;000) releases the clear on the line that was established by the inhibit instruction; then, the delay ensures that the printer has time to get ready to accept the command.

The Transmit Command instruction (IOC,C#5;003) at program label "D97" will transmit the address out on the SIO line. The printer is the only unit on the SIO line that has been hardware-wired to recognize the address; therefore, the printer will receive the address and return its status.

Status is accepted by the Transfer Byte instruction (IOC,C#5;001) and saved to be checked. Program control is returned to PF1+2, then transferred to D96+4. The status is checked for a ready condition; if not ready, the program goes into a loop checking status until a ready condition is detected. When the printer is ready, the return exit takes program control to program label D95. Here, program control is again transferred out to check status. This check is not necessary on the first pass; however, this is the start of the program loop that will send each character to the printer; once the loop is entered, the status check is necessary.

Before a character can be printed, it must be translated into USASCII code. In this example, the CT1 print translation table is used. A portion of the table is shown in figure 3-35.

The first character to be translated will be loaded into the accumulator by the LDA instruction at program label D95 + 2. Remember that register 3 holds the starting address, and the buffer is in page 01. The constant 300 octal is Inclusive OR'ed to the character code; this example will assume character "A". Therefore, the accumulator will contain 315, which is 015 plus the 300. The 315 will be used as the address of the USASCII character, which is loaded into the accumulator by the LDA instruction at program label D95 + 8.

Refer to figure 3-35. The character at location 315, page 05, is an "A". This character is then sent to the printer by the Transmit Data Byte instruction (IOC,C#5;002). The character will not be printed until the complete line of characters has been sent to the printer. If fewer than 132 characters are to be printed, an end of line code, 015 octal, is sent to the printer after the last character, and the printer will insert spaces for the remaining columns.

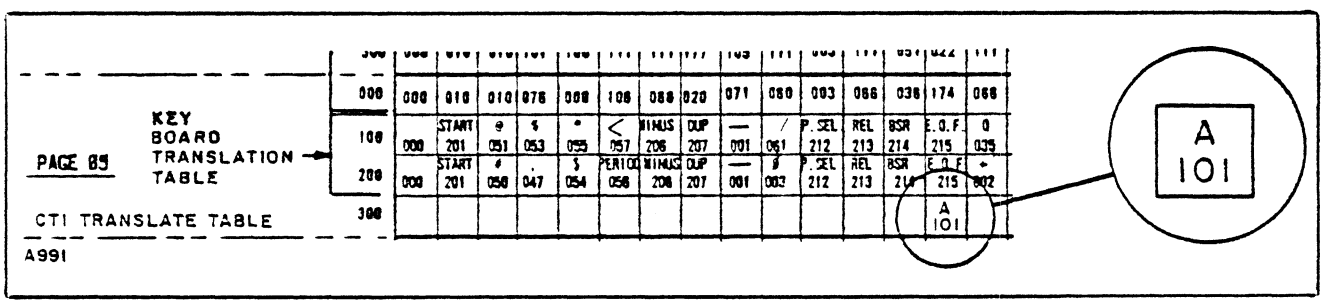


Figure 3-35. Portion of the CT1 Translation Table

After the character is sent to the printer, the buffer size count, which is held in register 2, is decremented. The count will also be checked for an end of buffer condition. If the end condition is not detected, the sequence starting at program label D95 is repeated until R#2 is equal to zero. At this point, the end of line code, octal 015, is sent to the printer. Upon receiving this code, the printer fills the remaining columns, if any, with spaces, and then prints the line.

Program control is returned to the mainstream program, where the print line count is increased by one. If additional lines are to be printed, the next sequential buffer can be set up, or if a control function is required, the function can be performed. In any case, program control is transferred to program label "PF1" to print data and to program label D99 for control functions.

If an error condition is detected in the printer, a clear command must be issued before another print function can be entered. The clear command is issued, in this programming example, by placing the clear command (050) in register 4 and executing a stack and branch to program label D99, the form control function. The clear command will clear the printer interface.

In this example, the form control command is loaded into register 4. Program control is transferred to program label D99, where control is then transferred to D98 and the addressing sequence is executed. After the unit has been addressed and the status checked, the command is loaded into the accumulator. Two checks are made on the command code: First is the Enable Bottom of Form (020), which is applicable for 1555 and 1556 only. Second is for the clear command. In both cases, the command is sent to the printer with no further checking.

If the function code is other than the BOF or clear, the status of the printer is checked at program label D96. Upon return, if the status is okay, the form command is sent to the printer. Note that status is checked on the form command before program control is returned to the mainstream program.

Form Control Tape.

The procedure for producing a Model 1551 form control tape is covered in detail in the Operator Instructions for the 1551 S/D. It is desirable, however, that the programmer have a general idea of how the tape is organized. Such a general description appears in the following paragraphs.

Form movement in the 1551 is controlled by a paper tape mounted in the vertical format unit, which is located on the left side of the printer. The tape is an 8-channel, paper-tape one-inch wide. There are ten sprocket holes to the inch; these are located between channels 3 and 4 on the tape. The tape reader and paper feed mechanisms are mechanically linked so that each line feed advances both the paper by one line and the paper tape by one sprocket hole. The standard unit generates six lines per inch. If the optional form feed unit is used, the operator can select 6, 8 or 10 lines per inch. Figure 3-36 shows a section of the tape.

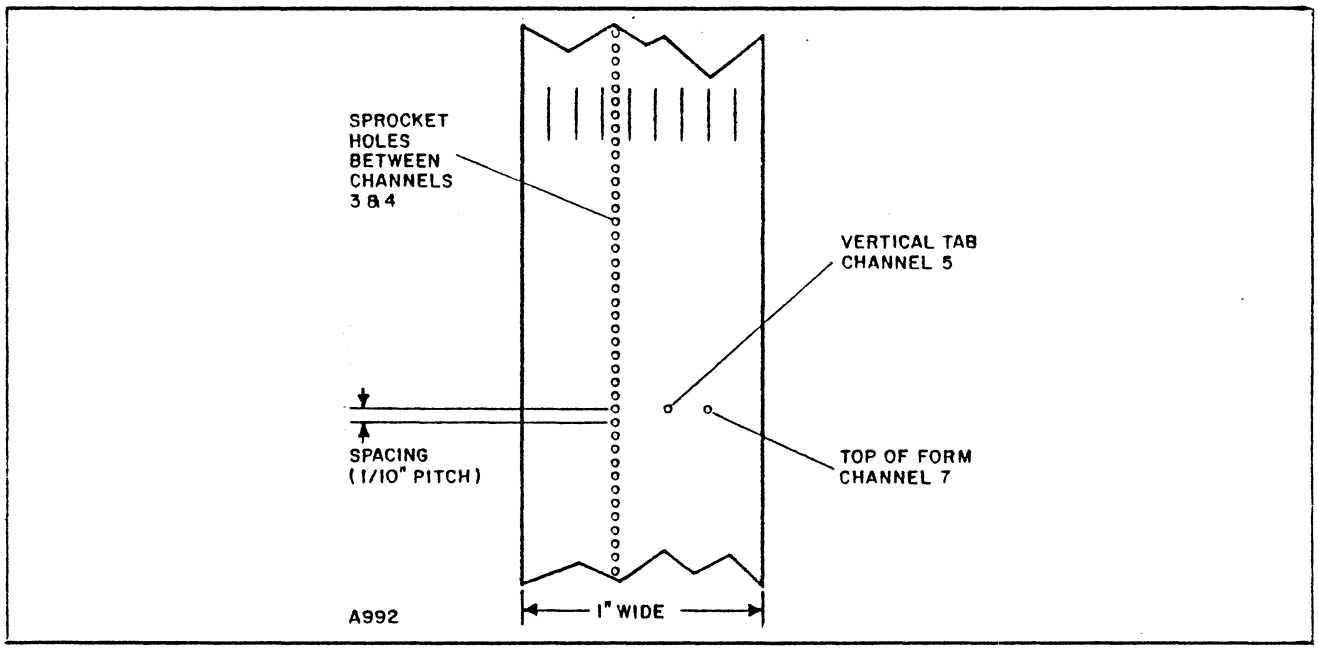


Figure 3-36. Section of the Model 1551 Format Control Tape

The command Vertical Tab is defined by a hole in channel 5 only and Top-of-Form is defined by a hole in channel 7 only. Vertical Tab holes are spaced six sprocket holes apart in channel 5 (corresponding to a 1-inch tab), and Top-of-Form holes are spaced 66 sprocket holes apart in channel 7 (corresponding to an 11-inch form).

Reception of a Vertical Tab code (octal 013) advances the paper (and tape) to the next hole in channel 5. For example, if the holes in channel 5 are spaced six sprocket holes apart, each Vertical Tab will advance the paper six lines (one inch).

Similarly, reception of a Form Feed code (octal 014) or pressing the TOP OF FORM switch advances the paper (and tape) to the next hole in channel 7.

Model 1553-1556 Printers.

The 1553-1556 line printers offer print speeds of from 125 to 400 lines per minute, with 132 characters per line. These units are equipped with a vertical form control loop, which is under program control, to control movement of the printed form. The 1553-1556 printers are available with a wide variety of character sets and they can print several copies on fan-fold paper.

Model 1553-1556 Instructions.

Commandsavailable to control the 1553-1556 printers are described below. They are sent to the printer over the serial I/O channel in the same manner as commands to the 1525/1526 and 1551 printers described earlier. Eight such commands apply to the 1553-1556 printers, as follows:

a. 012 - Feed One Line of Paper. Advances the paper vertically one line.

b. 013 - Vertical Tab. Advances the form by the number of lines equal to the contents of the tab counter. No action occurs if the tab counter is zero.

c. 014 - Form Feed. Moves the form to the next channel 1 punched in the printer vertical control tape. The standard function of this feature is to advance the form vertically to the next top of form (line 1).

d. 015 - End of Line or Carriage Return. Causes the printer to fill the balance of the print buffer in the printer with spaces and initiate a print cycle.

e. 017 - Shift In. Allows loadingof the tab counter with any two-digit decimal number. The most significant digit of the tab counter must be shifted in first, and two digits must always be shifted in.

f. 020 - Bottom of Form. Moves the paper to the bottom of the form, which is controlled by channel 3 in the format control tape. After a Bottom of Form command, the user must execute a Clear command prior to performing any other print function.

g. 023 - Slew to Channel. Moves the paper until a punch in the selected channel of the control tape is detected. After executing the Slew to Channel command, the printer is conditioned to interpret the next byte received as the channel number, as follows:

Channel 1 - Top of Form

Channel 2 - Not Available

Channel 3 - Bottom of Form

Channels 4-12 - User Option

h. 050 - Clear. This command clears the printer interface, and acts as a general clear to the printer. It should be executed to clear a printer error condition and the bottom of form condition.

Model 1553-1556 Status Word.

The status word provided by the 1553-56 printers is very similar to that from the 1551 printers. Bits 6 and 7 apply to errors in the I/O channel, and bits 2 and 5 are not used. The meaning of the four remaining status available is discussed below. Note that each bit is a one when it indicates the condition described.

a. Bit 4 - Bottom of Form. Indicates that a Bottom of Form command has moved the form until a punch in channel 3 of the form control tape was reached.

b. Bit 3 - Printer Error. Indicates that the printer is out of paper or that the cover over the print mechanism is open.

c. Bit 1 - Printer Busy. Indicates that the printer is engaged in a print operation and is not yet ready to accept a new character.

d. Bit 0 - Select Switch. This bit indicates that the printer is on-line.

Model 1553-1556 Programming Methods

The sample program used in the 1551 print example can be used for the 1553 through 1556 printers. However, there are two major differences in the programming: (1) The control functions are different, and (2) the vertical format unit is different. There is also a small difference in the status word. These differences, while significant, will not change the basic programming concept.

If an error condition is detected in the printer, a clear command must be issued before another print function can be entered. The clear command is issued, in the programming example, by placing the clear command (050) in register 4 and executing a stack and branch to program label D99, the form control function. The clear command will clear the printer interface and also the bottom of form condition.

The bottom of form is not an error condition. However, when detected, bottom of form must be cleared by issuing a clear command to the printer prior to the next print function.

The form control unit, called the vertical format unit, offers an 8-channel form control loop. Figure 3-37 is representative of an 8-channel form control loop. However, the 1553 through 1556 printers have an option available to expand the form control loop up to 12 channels.

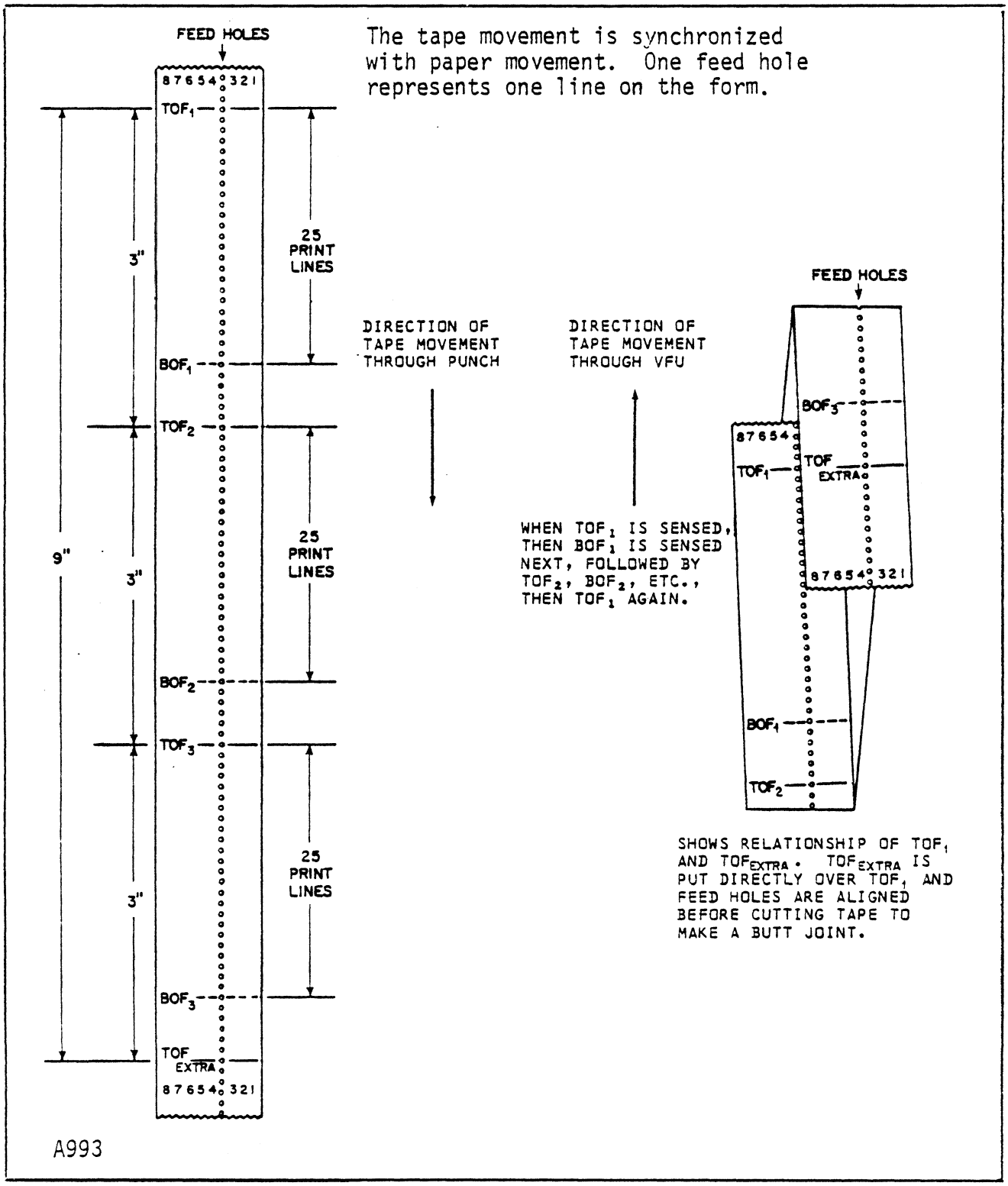


Figure 3-37. Model 1553-1556 Form Control Tape

The code sent to the printer is USASCII; this holds true with the form feed channel number code. However, the channel control circuits use only the low-order 4 bits of the byte, enabling a count of from 0 to 16 decimal. If the channel number is 8, then the USASCII characters (8, H, X, or ,) can be used, since the low-order four bits are 1000 (binary).

The vertical format unit can perform four basic functions. They are:

a. Line feed.

b. Slew to top of form (form feed).

c. Slew to channel number.

d. Vertical tab.

The single space or line feed function is achieved by sending the printer a function code of 012 octal. As its name implies, the function causes the form to advance one line. When printing, the programmer does not have to send a line feed command for each line of printing. The form will automatically advance one line after printing. The line feed function is used if additional lines are to be spaced.

The form feed or slew to top of form will advance the form to the top of the next form. The top of the form is identified by a hole punched in channel 1 of the control loop. Refer to figure 3-37. The form feed is achieved by sending the printer the form feed command, 014 octal; upon receiving the command, the printer will advance the form to the next channel 1.

The slew to channel command, 012 octal, provides a method to move the form any number of lines, including into successive forms, until the respective channel is sensed in the control loop. For example, if a slew to channel 7 is given after printing the last line, line 25 of form 1, the form would advance to line 10 of the next form. Refer to figure 3-37. As a second example, locate the channel 5 holes in figure 3-37. Assume the printer form is positioned at line 20 of the uppermost form represented by the control loop. If a slew to channel 5 command is given, the form will advance to the third form, line 4, and stop.

When the slew to channel command is sent to the printer, the printer will interpret the next byte received as the channel number. The form will then be advanced until the channel is detected.

The vertical tab, function code 013 octal, provides a method of advancing the form a determined number of lines; it is controlled by a count sent to the vertical format unit. To achieve the tab, the tab counter must

first be loaded with a count. The procedure for loading the tab counter is to first send the printer the shift in command (017 octal). This command will condition the printer so that the next two bytes will be loaded into the tab counter. The most significant byte will be loaded first. The counter is a binary counter, and only the least significant four bits of each byte will be used to load the counter. The maximum count that can be loaded is a 99 decimal. The two bytes would take the form of XXXX1001 XXXX1001 if a maximum count is used. However, in normal operation, the count would be less than equal to the total number of lines in a form. Again, the steps required to achieve vertical tab are:

a. Send shift in command (017).

b. Send most significant digit of count.

c. Send least significant digit of count.

d. Send vertical tab command (013).

After the command has been received by the printer, the paper (form) will be advanced and the counter will be decremented by one for each line the paper advances. If the top and bottom of form are indicated by a hole punched in channels 1 and 3, respectively, the total line count between the bottom of one form and the top of the next form will only decrement the counter by one. If the count is a zero, there will be no form movement.

The bottom of form command (020) operates somewhat differently from other commands. When this command is executed, it sets a condition to cause the printer to look for the next bottom of form, that is, a hole punched in channel 3 of the vertical form control loop. When channel 3 is detected, bit 4 in status word is set to a one. The program must be checking for this condition if a slew to the next top of form is required. The printer <u>will not</u> slew to the top of the next form automatically.

3-12. MAXI-TAPE UNITS.

There are four maxi-tape units available; they are the Models 1511, 1512, 1513, and 1514. The purpose of these units is to provide tape drives for the Series 1500 line that can handle tape reels compatible with larger computer systems.

Typical magnetic tape is a flexible plastic tape coated on one side with a thin layer of ferromagnetic material. The most commonly used computer grade magnetic tape has a polyester backing and is 0.5 inches wide and 0.0015 inches thick.

There are three basic recording formats used with the maxi-tape units, as follows:

a. A 7-track, 556 bits per inch form, which is used by the Model 1511 in 7-inch reels.

b. A 9-track, 800 bits per inch form. This is used by both the Model 1512 and 1513, with the 1512 accommodating 7-inch reels and the 1513 using up to 10-1/2-inch reels.

c. A 9-track, 1600 bits per inch form, which is used only in the Model 1514. The 1514 can also accommodate up to 10-1/2-inch reels.

Although the 9-track, 800 bits per inch format is the most commonly used in the Series 1500, all three formats are described below.

Tape Formats.

In the Model 1511, seven-bit characters are written in seven separate tracks along the length of the tape as shown in figure 3-38. Data to be recorded is first grouped into bytes of six bits each. The byte is then recorded, together with a parity bit, across the width of the tape, with one bit in each track. Odd parity is used.

A density of 556 characters per inch is used in the Model 1511. This bit density is based on the data recording rate and the tape speed. The data recording rate is 6.954 KHz, and the tape speed is 12.5 inches per second.

The characters are grouped into records. Each record may contain twelve or more data characters. In addition to the data characters in each record, a longitudinal redundancy check character (LRCC) is recorded as the last character of each record.

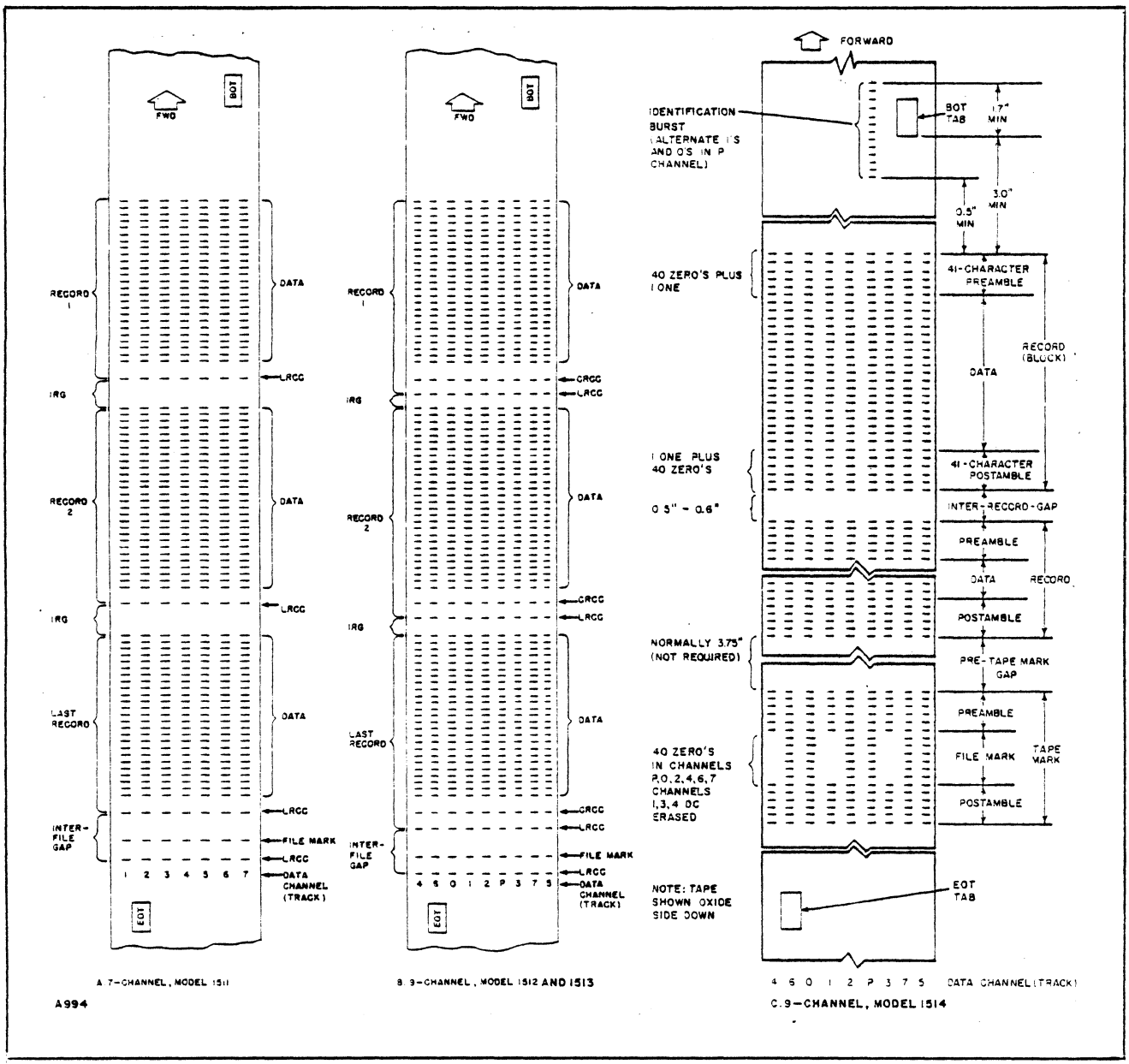


Figure 3-38. Tape Formats

In the Models 1512 and 1513, the tape format is the same as the format for the Model 1511, with the following exceptions:

a. The characters contain nine bits; eight data bits and one odd-parity bit.

b. The data recording rate is 10,000 KHz, and the character density is 800 characters per inch.

c. Following the last data character and before the LRCC, is a cyclic-redundancy check character (CRCC). Both of these characters provide a means of checking data operations.

d. Consecutive data channels are not allocated to consecutive tracks. This organization increases tape system reliability.

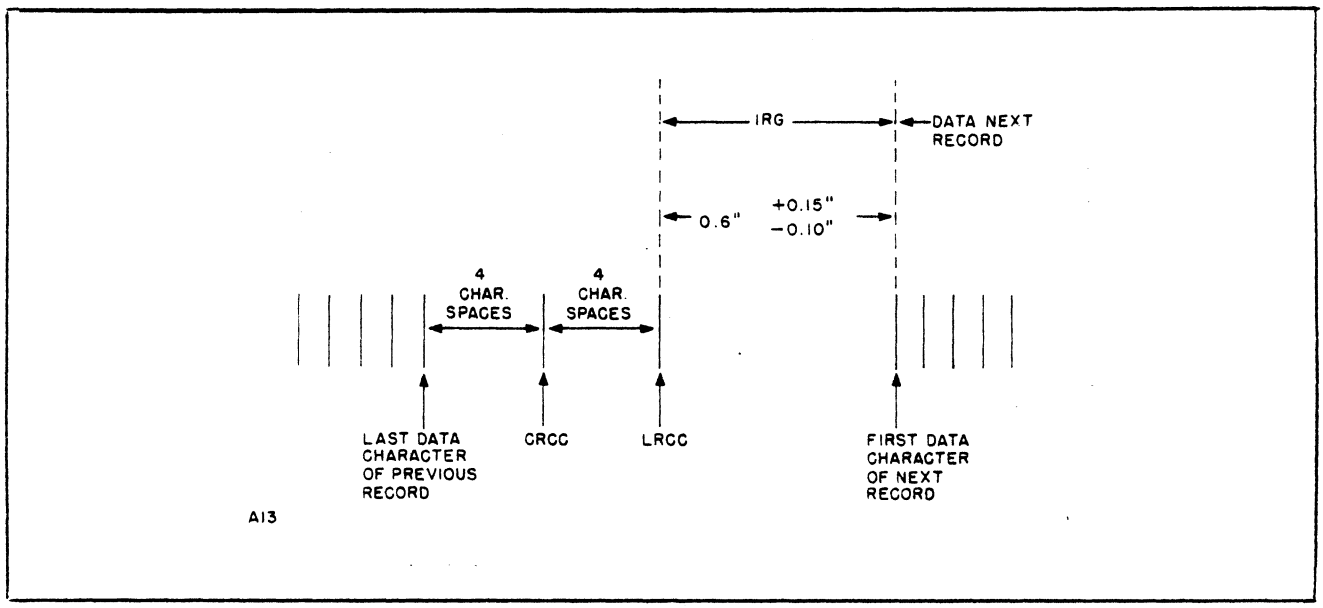
In all models, a 0.6-inch length of erased tape is left between each record. This length of erased tape is called the inter-record gap (IRG) and serves as a stopping place for the tape head, since only an entire record may be read or written at one time.

Records are grouped into major units called files. A file may contain any number of records and is limited only by the length of the tape. A special record called a tape mark or sometimes an end-of-file (EOF) mark is used to separate files. A tape mark consists of a file mark and an identical LRCC eight character spaces from it. The tape mark is usually preceded by approximately 3.75 inches of erased tape, but this extra gap is not a requirement. (See figure 3-39.)

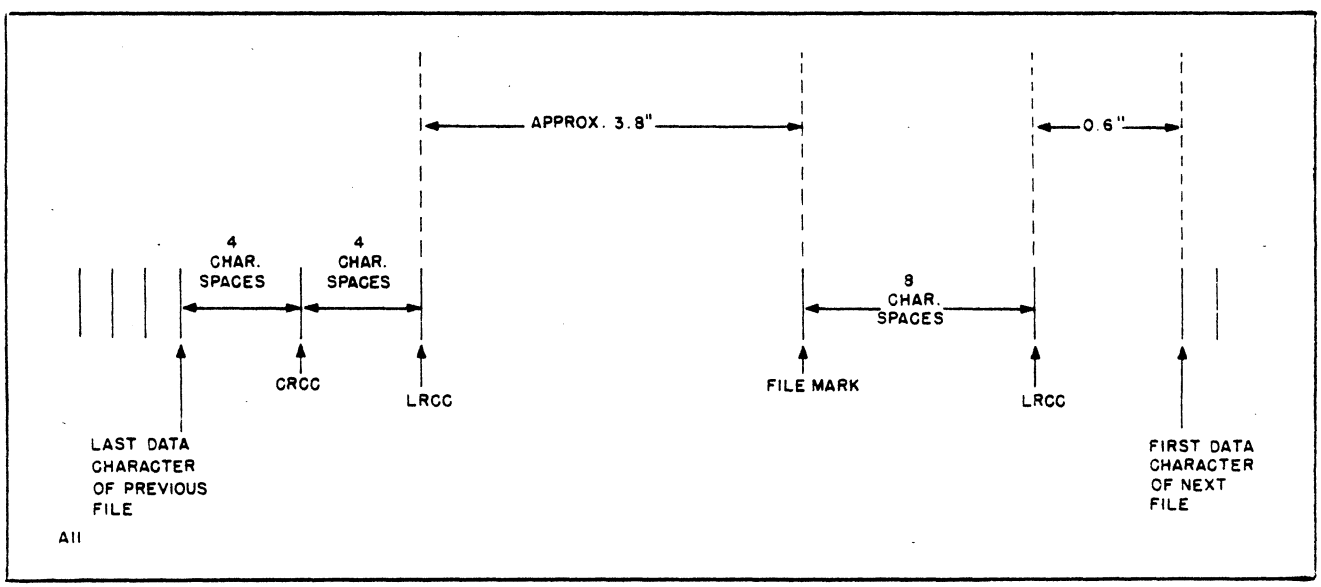
A reflective strip is placed on each end of the tape. The strip at the beginning of the tape is called the beginning-of-tape (BOT) marker, whereas the strip at the end of the tape is called the end-of-tape (EOT) marker.

In the Model 1514, nine-bit characters are written in nine separate tracks along the length of the tape as shown in figure 3-38. Data to be recorded is first grouped into bytes of eight bits each. The data byte is then recorded, together with a parity bit, across the width of the tape with one bit in each track. Odd parity is used.

The nine-bit characters are recorded along the tape at a density of 1600 characters per inch. This character density is based on the data recording rate and the tape speed. The data recording rate is 20 KHz, and the tape speed is 12.5 inches per second. The characters are grouped into records.



(a) Spacing at End of Record



(b) Spacing at End of File

Figure 3-39. Record Spacing

Each record may contain twelve or more data characters plus a preamble and a postamble which are 41 characters each. The preamble is 40 consecutive zeros followed by a one bit in all nine tracks, and the postamble is a one bit followed by 40 consecutive zeros in all nine tracks. A 0.6-inch length of erased tape is left between each record. This length of erased tape is called the interrecord gap and serves as a stopping place for the tape head, since only an entire record may be read or written at one time.

As is the case for the other maxi-tapes, records are grouped into major units called files. A file may contain any number of records and is limited only by the length of the tape. A special record called a tape mark, or sometimes an end-of-file (EOF) mark, is used to separate files. A tape mark consists of a file mark plus a preamble and a postamble. The tape mark is usually preceded by approximately 3.75 inches of erased tape, but this extra gap is not a requirement.

A reflective strip is placed on each end of the tape. The strip at the beginning of the tape is called the beginning-of-tape (BOT) marker, whereas the strip at the end of the tape is called the end-of-tape (EOT) marker. The beginning of a phase-encoded tape includes a group of bits called an identification burst. This consists of alternating ones and zeros which extend beyond the BOT as shown in figure 3-38.

Record Length.

Record length is not physically limited on the maxi-tape units. The number of characters per record is controlled by the programmer. That is, the programmer may select the record size that best suits the application.

When selecting the record size the programmer should take into account the physical qualities of magnetic tape. Physical defects in tape as a result of aging and dirt from handling can make error recovery difficult when extremely long records are used. Under certain conditions, bad spots on tape can generate a false beginning of record when the tape is being backspaced during error recovery. While there is no fixed number for record size, 1024-character records is generally accepted as the upper limit.

There are two record sizes used in the standard background. They are, 256 and 1024-characters per record. Both record sizes are fully supported by the background error recovery routine.

Maxi-Tape Instructions.

Instructions used to control the maxi-tape drives are forwarded to the drives over the serial I/O channel. Refer to the description of the SIO channel as necessary to review how this is done.

There are nine instructions to which the maxi-tape drives respond. They are described below. In each case, the two-place octal number preceding the instruction title is the code for the instruction.

a. 01 - Write. The write command causes the MTU to:

(1) Start tape motion forward at normal write speed.

(2) Energize write and erase heads.

(3) Start the delay start counter.

(4) Send a status byte back to the Intelligent Terminal, indicating MTU status when the command was received.

(5) Stop transmitting after the status byte is transmitted.

While the delay start counter is counting and the tape motors are getting up to write speed, the Intelligent Terminal detects the break in transmission. It then transmits the first byte to be written on tape, after which it stops transmitting. When the delay start count is completed, the MTU writes the first byte on tape. It also transmits a status byte, and then stops transmitting. The write clock determines the rate at which data is written on tape. The Intelligent Terminal and the MTU continue to alternate transmissions of data bytes and status bytes until the Intelligent Terminal sends a stop write command.

b. 02 - Backspace (One block). The backspace command causes the MTU to initiate tape motion in the reverse direction. When the tape has been backspaced one record and the read-write head is in the IRG, the MTU sends a status byte to the Intelligent Terminal and then stops transmitting. The MTU then waits for the next command.

c. 03 - Read (One block). The read command causes the MTU to start tape motion forward at the normal reading speed. The data is transmitted to the Intelligent Terminal as a continuous stream. The time between data bytes is filled with idle bytes.

d. 04 - Stop Write. The stop write command is used to terminate a write operation or an erase operation. When used to end a write operation, the MTU writes the CRCC and LRCC check characters, stops the tape motion, de-energizes the write and erase heads, returns a status byte to the Intelligent Terminal, and then stops transmitting. When the stop write command is used to terminate an erase operation, the MTU stops the tape motion, de-energizes the write and erase heads, returns a status byte to the Intelligent Terminal, and then stops transmitting.

e. 05 - Erase. The erase command causes the MTU to start tape motion forward at the normal write speed with both the write and erase heads energized. The clock pulses to the write head, however, are inhibited. Erasing continues until a stop write command is received.

f. 06 - Rewind and 07 - Rewind With Interlock. The rewind and rewind with interlock commands both cause the MTU to start tape motion in the reverse direction at the normal rate. As soon as the MTU is rewinding, the status byte is returned to the Intelligent Terminal, indicating that the MTU does not require any additional supervision to complete the rewind operation. Rewinding continues until the load point of the tape is reached. At the load point, tape motion is stopped, bit 0 (not ready) of the status byte is cleared, and, if the rewind with interlock command has been issued, the MTU goes off-line. Once off-line, operator intervention is required before any other operation (e.g., read, write, erase) may be performed.

g. 10 - Status. The status command causes the MTU to return a single status byte to the Intelligent Terminal. The status command may be issued at any time that a read or write operation is not being performed. It is commonly used to detect when the MTU has completed rewinding, detect the end-of-tape (EOT) mark during an erase operation, and check for errors after a write command.

h. 11 - Write File Mark. The write file mark command causes the MTU to write a special character called the end-of-file character. When the MTU receives the command, it initiates forward tape motion at the normal write speed, proceeds to write the file mark, return a status byte to the Intelligent Terminal, and then stop transmitting. In order to properly complete the operation, the Intelligent Terminal then issues a stop write command to the MTU.

Maxi-Tape Status Word.

The same status word is available from all four models. As is the case with all peripheral units operating on the serial I/O channel, bit 6 (addressing error) and bit 7 (I/O error) pertain to the conditions in the SIO circuits. Thus, only bits 0 through 5 are available to indicate tape status.

The meaning of bits 0-5 of the status word is described below. Note that the respective bit is a one to indicate the condition described.

a. Bit 5 - Tape Runaway. A tape runaway condition is defined as a read, write, or rewind command that results in tape motion for two seconds without data being detected by the read circuits. This condition is most often caused by a break in the tape, although it may be caused by improper mounting of the tape on the MTU transport and a failure that prevents writing or reading data.

b. Bit 4 - File Mark Detected. This bit indicates that the read circuits have detected a file mark. Bit 4 will be set under one of three conditions: (1) in the final status byte returned to the Intelligent Terminal during a write-file-mark sequence, (2) at the end of a read operation if a file mark has been read, and (3) at the end of a backspace or rewind operation if a file mark was passed over.

c. Bit 3 - Tape Error. This bit indicates that a tape error has been detected. A tape error may be one of the following conditions:

   (1) An attempt to perform a write operation on a tape whose supply reel has had the write enable ring removed (a file protect condition).

   (2) A data byte has not been transferred to the MTU by the Intelligent Terminal in time to be written onto the tape.

   (3) A parity error has been detected.

   (4) An error in the CRCC or the LRCC has been detected.

d. Bit 2 - Tape at Load Point. This bit indicates that the tape is at the load point (beginning-of-tape marker).

e. Bit 1 - End-of-Tape Warning. This bit indicates that the tape is positioned at, or is past, the end-of-tape (EOT) marker. This bit is cleared by either rewinding or backspacing the tape over the file mark.

f. Bit 0 - Not Ready. This bit indicates that the MTU is not ready to accept commands. It is set under any one of the following conditions:

   (1) The tape has not been loaded on the tape transport.

   (2) The tape is rewinding.

   (3) The MTU is off-line.

Programming Methods.

The programming example used in this discussion is based on the 1512 or 1513 tape transport operating at a recording density of 800 bits per inch with a 9-channel recording format. This combination was selected because it is the most widely used in the Series 1500.

Write Operation

Before the write function is entered, the write buffer must be loaded. In this example, page 01 is used as the buffer area. The sequence required to write a record on tape is:

a. Address the tape unit.

b. Check status to ensure that the unit is ready.

c. Send a write command to unit.

d. Send data to be written.

e. Send a stop write command.

f. Check status.

Refer to table 3-12, program label D31, the start of the write function. Program control is transferred from the mainstream program to D31, then to D98 to address the unit. The Set Master Mode instruction places the terminal in the master mode, which gives the terminal control over the I/O channel. The Inhibit Line instruction clears all units on the line and stops any transmission that may be in progress. A delay of 15 microseconds is achieved by the time it takes to load the buffer size and set the address pointer for the buffer. The A80 routine will be entered again later in the sequence.

Program control is returned to D98+6, where the Start Transmit instruction is executed. This instruction releases the clear on the SIO line that was initiated by the Inhibit Line instruction, and sets up the proper conditions for transmission.

Before the write sequence was entered, the device address was loaded into register #4. This address will be modified to identify the address word to the SIO circuits. After the address has been formed, the Set Device Address instruction (IOC,C#5,007) transfers it into the SIO section of the I/O controller in preparation for transmission out on the SIO line. Program control is then transferred to D97, where the address is sent over the SIO line to the tape unit. Upon recognition of its address, the tape unit returns its status.

Table 3-12. Sample Maxi-Tape Routines

Maxi-Tape Read Function		
D30:	SBU,D98.	Go to Address Device
	BRH;D32+18.	Return if SIO or Address Error, or Not Ready
	SBU;D34.	Go to Check Status (Return when Tape Unit is Ready)
	SBU;A80.	Copy PSD Instruction
	LDA,R#0;OCT:003.	Load Read Command
	IOC,C#5;003.	Transmit Command
	IOC,C#5;001.	Receive Byte (DATA)
	STA,I#3;P01.	Store Data Byte
	STA,R#0;R#4.	Store Status
	IOC,C#0;016.	Load Status Byte
	TMJ,+14;OCT:004.	Test for Status (I/O Status)
	SUX,R#2;OCT:001.	Decrement Size Once
	CPX,R#2;OCT:000.	Test for End
	BRH,D30+12.	Check Finished No
	SBU;D32.	Final Check
	SBU;D33.	Tape Back-Up
	BRU;D30+06.	Error-Retry
	SBU;D32+20.	Final Status Check
	BRU;D30+30.	Return-Exit

Table 3-12. Sample Maxi-Tape Routines (Continued)

Maxi-Tape Write Function		
D31:	SBU;D98.	Go to Address Device
	BRH;D32+18.	Address OK. Yes
	SBU;D34.	No. Check Status
	SBU;A80.	Load PDS Instruction
	LDA,R#0;OCT:001.	Load Write Command
	SBU;D97.	Go Transmit Command
	LDA,I#3;P01.	Load Data Byte
	IOC,C#5;002.	Transmit Data Byte
	IOC,C#5;001.	Receive Status
	ANA,R#0;OCT:375.	Strip off EOT Bit
	TLJ,+14;OCT:000.	Error? Yes
	LDX,R#1;OCT:044.	Load for 100 ms (microseconds) delay (for tape Motion to Stop)
	SBU,D93.	Make Delay #1
	LDA,R#0;OCT:010.	Load Status Command
	SBU;D97.	Go Transmit Control Command
	SBU;D32+06.	Test for Error
	BRU;D31+46.	Error? Yes
	SUX,R#2;OCT:001.	Decrement Size Once
	CPX,R#2;OCT:000.	Test for End
	BRH;D31+12.	Finished? No
	LDA,R#0;OCT:004.	Load Stop Write Command
	IOC,C#5;003.	Transmit

Table 3-12. Sample Maxi-Tape Routines (Continued)

Maxi-Tape Write Function (Continued)		
	SBU;D32.	Final Status Check
	SBU;D33.	Back-up Tape
	BRU;D31+06.	Try Again
Final I/O Status Check Function		
D32:	LDX,R#1;OCT:011.	Load for 23 ms delay
	SBU;D93.	Make Delay #3
	IOC,C#5;001.	Receive Status
	STA,R#0;R#4.	Save Status in R#4
	IOC,C#0;016.	Load General Status Byte
	TMJ,+10;OCT:004.	Error? Yes
	CPX,R#5;OCT:010.	Check for 8 Retries
	ADX,R#5;OCT:001.	Step Retry Counter
	BRH;D32+36.	Last Try? Yes
	EXU;000.	Return Exit
	LDA,R#0;R#4.	Retrieve Status
	TLJ,+14;OCT:000.	Any Error?
	TMJ,-12;OCT:010.	Tape Error? Yes
	TMJ,+08;OCT:002.	EOT Command? Yes
	TMJ,+06;OCT:020.	Detected File Mark? Yes
	CPA,R#0;OCT:000.	Force Compare Greater Than
	BRU;D32+36.	Return
	CPX,R#0;OCT:377.	Force Compare Less Than
	EXB,D37+06.	Return

Table 3-12. Sample Maxi-Tape Routines (Continued)

	Backspace Maxi-Tape Function	
D33:	LDX,R#1;OCT:016.	Load for 32 ms delay
	SBU;D93.	Make Delay #1
	LDA,R#0;OCT:002.	Load Back-up Command
	SBU;D97.	Go Transmit Command
	LDX,R#1;OCT:016.	Load for 32 ms delay
	SBU;D93.	Make Delay #1
	EXU;000.	Return Exit
	Pre-Status Check Function	
D34:	LDX,R#5;OCT:000.	Clear Retry Counter
	LDA,R#0;OCT:010.	Load Send Status Command
	SBU;D97.	Go Transmit Command
	TMJ,-06;OCT:001.	Ready? Yes
	EXU;000.	Return Exit
	Write File Mark Function	
D35:	SBU;D98.	Go Address Device
	BRH;D37+12.	Error? Yes
	LDA,R#0;OCT:011.	Load Write File Mark Command
	SBU;D97.	Go Transmit Command
	LDA,R#0;OCT:023.	Load File Mark Character
	IOC,C#5;002.	Transmit
	IOC,C#5;001.	Receive Byte (Status)
	LDA,R#0;OCT:004.	Load Stop Write Command

Table 3-12. Sample Maxi-Tape Routines (Continued)

	Write File Mark Function (Continued)	
	IOC,C#5;003.	Transmit Command
	IOC,C#5;001.	Receive Status
	TMJ,+02;OCT:020.	Test for File Mark
	BRU;D37+06.	EOJ and Return
	Rewind Maxi-Tape Function	
D36:	SBU;D98.	Go Address Device
	BRH;D37+12.	Error? Yes
	LDA,R#0;R#5.	Load Command (Command is loaded into R#5 before entering routine)
	BRU;D37+08.	Return
	Maxi-Tape Backspace & EOJ Function	
D37:	SBU;D98.	Go Address Device
	BRH,D37+12.	Error? Yes
	SBU;D33.	Backspace Tape
	LDA,R#0;OCT:010.	Load Status Command
	IOC,C#5;004.	Stop Transmit Instruction
	SBU;D97.	Transmit Command
	EXU;000.	Return Exit
	I/O Channel Setup	
D98:	IOC,C#5;010.	Set Master Mode
	IOC,C#5;006.	Inhibit Line
	SBU;A80.	Delay Total of 15 ms

Table 3-12. Sample Maxi-Tape Routines (Continued)

	I/O Channel Setup (Continued)	
	IOC,C#5;000.	Start Transmit
	LDA,R#0;R#4.	Load Address (Address is loaded into R#4 before entering routine)
	ADA,R#0;OCT:200.	Insert High-Order Address Tag
	IOC,C#5;007.	Set Device Address
	SBU;D97.	Go Transmit
	STA,R#0;R#4.	Save Status
	ANA,R#0;OCT:301.	Mask for SIO, Address, and Not Ready
	CPA,R#0;OCT:000.	Compare for Error Set Condition Register
	EXU;000.	Return
	Transmit Control Command	
D97:	IOC,C#5;003.	Transmit Control Byte
	IOC,C#5;001.	Transfer Byte (Status)
	EXU;000.	Return
	Align Register Contents	
A80:	LDX,R#2;OCT:200.	Load Size (128 bytes)
	LDX,R#3;OCT:000.	Set Address Pointer
	EXU;000.	Return
	Time Delay Linkage	
D93:	SMS;S#0.	
	BRU;P06;210	Delay is D92 Mini-Tape at P06-322

(The time delay is in mini-tape routine)

Upon receiving the status byte, program control is returned to D98+16. The status word is saved in register #4 to be used by the mainstream program. At this point in the address sequence, the only error conditions that are important are: SIO error, address error, and not ready condition. These conditions are represented by bits 7, 6, and 0, respectively, in the status word. The ANA,R#0;OCT:301 instruction tests these bits, other bits from consideration and the compare instruction will set the condition register, which will eventually be used by the mainstream program. After the compare instruction, program control will be returned to program label D31+2.

If an error is present at this point in the program, program control will be transferred back to the mainstream program. The types of errors that might be expected at this point would be: (1) tape unit not on, (2) no tape on unit, and (3) tape not properly mounted. These conditions will generate a not ready status.

Assuming there are no errors, program control will be transferred to D34 to again check status. This is a redundant check which is not necessary at this point. After this check, program control will return to the stack and branch instruction at D31+6, where control will transfer to A80.

The sequence at program label A80 will load the address pointer and the block size into registers R#3 and R#2, respectively. Control will then be returned to D31+8.

The write function code is loaded into the accumulator in preparation to be sent to the tape unit. Program control is then transferred to program label D97, where the code is sent to the unit. The status is received but not checked at this time. Control will then transfer back to D31+12.

With the unit moving tape, the program can start sending data to be recorded. The data byte will be loaded into the accumulator at program label D31+12, then transmitted to the tape unit by the IOC,C#5;002 instruction. Assuming no errors, program control will transfer to D31+34, where the record size is decremented by one. A test is then made to determine if the record count is equal to zero; if not equal to zero, control is returned to D31+12. This sequence is repeated for each byte of the record.

After the last byte of the record has been sent to the tape unit, the size count will be decremented to zero. Refer to table 3-12, label D31+38. This branch condition is not satisfied; therefore, the stop write command is sent to the tape unit. Program control is transferred to D32, where final status is checked.

A delay of 34 milliseconds is initiated to wait for the final status word to be sent back from the tape unit. The unit will return three words. The first two make up a two-word CRCC, which is the same CRCC written on tape. This CRCC is not used by this program, but could be used as a check if a CRCC were developed by the mainstream program. The last byte will be the final status word. The delay will use up time until the last status word is returned. The status word is received and stored in register #4; the Load Status Byte instruction IOC,C#0;016 will load the general status byte. If bit 2 of the general status byte is a one, the last byte received by the Transfer Byte instruction (IOC,C#5;001) was the device status word. The status word that was stored in R#4 is retrieved and checked for errors. If no errors are detected, control is passed to label D37+06.

In this example, transmission on the line will be shut down. This procedure is not always necessary. However, when this procedure is followed, the Inhibit Line instruction is not necessary when the line is turned on. The load status command, along with the Stop Transmit instruction, will shut down the SIO line.

Note that at program D97+2, the Transfer Byte instruction is executed but has no effect in this instance. When the write function is completed, program control is transferred back to the mainstream program, the condition register is set, and the final status word is stored in register #4.

Refer to table 3-12, program label D31+18. If the end-of-tape status were present at the beginning of the write operation, it would be masked out by the "ANA" instruction, the reason being that if an end-of-tape condition is present, there is still enough space to record the present data record. Therefore, the EOT is ignored at the start of the write operation. The final status word will have the EOT bit set, which will alert the mainstream program.

At line D31+20, a check is made for errors. If an error is present, the program will attempt to recover and write the record. The 100-millisecond delay is used to allow time for the tape unit to stop before a backspace function is executed. The status word again is called in by the sequence at program label D97. Program control then returns to D31+30 and is transferred to D32+06. The general status word is checked to ensure that the received word was status and then the status word is examined to determine the type of error. If a tape error is detected, the program will go into an eight-retry loop. If at the end of eight retries the error is still present, control is returned to the mainstream program.

The check for EOT at line D32+26 should not be entered during a write sequence. An error other than EOT had to be present for the program to enter line D31+22. On a tape write operation, the only errors that should be detected are a tape error or tape runaway.

Write File Mark Operation

The write file mark function will write a single file mark character on tape. For the 9-channel at 800 BPI, this character is an 023 octal. After the file mark has been written, the file mark status bit, bit 4, will be set to a one. This status will be present in the final status byte.

The write file mark function is entered at program label D35 in table 3-12. The address sequence at D98 is the same as a write function. Again, a check is made for errors as a result of the address sequence.

Program line D35+4 loads the write file mark command code (011) into the accumulator. The command is sent to the tape unit by the routine at D97. Upon receiving the command, the unit will start moving tape and will write the next byte it receives as a file mark; in this example, it is the 023 code. The program will then send a stop write command and wait for the final status. The final status byte should contain the file mark condition, that is, bit 4 set to a one. A test is made for the file mark, and the condition register is set; line D35+20 makes the test.

The same shutdown procedure is followed as in the write function. After shutdown, program control will be returned to the mainstream program.

Read Operation

A read buffer must be established prior to executing the read function. In this example, page 01 is selected. The record size is 128 bytes, which is 200 octal. The sequence of events required to read a record is:

a. Address the tape unit.

b. Check status to ensure that the unit is ready.

c. Receive data.

d. Check status.

Refer to table 3-12, program label D30. The address sequence is the same as in a write function; that is, the address is sent to the unit and the unit status is checked. If an error is detected, control is returned to the mainstream program, and in this case, the status word will be in R#4.

With no errors, program control will transfer to A80 to load the page and starting location of the buffer. At this point, with all initial conditions established, the read command can be sent to the tape unit; program line D30+10 does this. The program will then advance to the

Receive Byte instruction (IOC,C#5;001) and wait until the first data byte is received. When the byte is received, it is stored in the read buffer. This same byte is stored in R#4, which is the status register. The reason for this is the program cannot determine if the byte is data or status until the Load Status Byte instruction (IOC,C#0;016) is executed. If the byte is a status byte, bit 2 of the general status word will be set to a one. If an error occurs within the record, a status byte will be sent from the tape unit to the terminal. If no errors occur, the status byte will be the last byte received by the terminal.

Assuming the received byte is data, the program will fall through the test instruction on line D30+20. The next two instructions will decrement the record size and check for an end-of-record. If the end is not found, the program will loop back to D30+12 to receive the next byte. This loop will be repeated for each data byte.

If the read buffer is initialized with a count that is equal to the record size, then the program will fall through the test at line D30+26 upon receiving the last byte of the record. Program control will transfer to D32 to make the final status check. The delay is again used to ignore the CRC character, which is not used in this example. Since the unit is a 9-track unit, there will be two bytes for the CRC character, both of which are ignored.

The final status word will follow the CRC character. The status word is stored in R#4 and checked for errors. If an error is detected, the program will either try to read the record again or return control to the mainstream program.

3-13. DISC UNITS.

There are two types of disc units available: Model 1540 and Model 1543. A single, non-removable disc is housed in the 1540, while the 1543 houses both a non-removable disc and a removable disc pack of 5440 type. Refer to Section 1 of this manual for photographs of both models of the disc unit and the disc pack.

A disc controller, which consists of two printed circuit boards, is added to each intelligent terminal to allow the terminal to operate the disc units. Up to four disc units (either Model 1540, 1543, or any combination of models) can be operated by the controller. Thus, the maximum capacity would be four Model 1543 units.

Disc Format.

An overall understanding of disc operation is better achieved if first the physical format of the disc is grasped. The disc is commonly called a "platter". Data is recorded on both sides of the disc. There are two heads, which allow a choice of which side of the disc is to be used. The disc is divided into tracks and each track is divided into 24 equal segments called sectors. The 1540 disc has 406 tracks (203 on each side) and the 1543 disc has 408 tracks (204 on each side). Figure 3-40 illustrates the format of a sector of information.

Each sector contains its own logical address as well as the 268 header and data bytes. The sector address comprises an address preamble (215 zeros followed by a one), a logical sector address of 2 bytes representing this unique disc location, a sector address check of 2 bytes (16-bit CRC), and an address postamble made up of 8 bytes of zeros. Data that follows contains a data preamble (31 zero bits followed by a one bit), header data containing 12 bytes, 256 data bytes, and a data postamble composed of zeros until the sector ends.

Within each track, any logical address can be assigned to any physical sector. The programmer also has the option of allowing intervening sectors to exist between consecutively-addressed sectors. Since the programmer is familiar with the data being processed, this option of interleaving sectors allows one sector to be used, data to be processed, and a second consecutively-addressed sector to be used at about the time it physically appears under the read-write head. In effect, this option allows the programmer to control to a great degree the amount of access time needed to read or write on the disc.

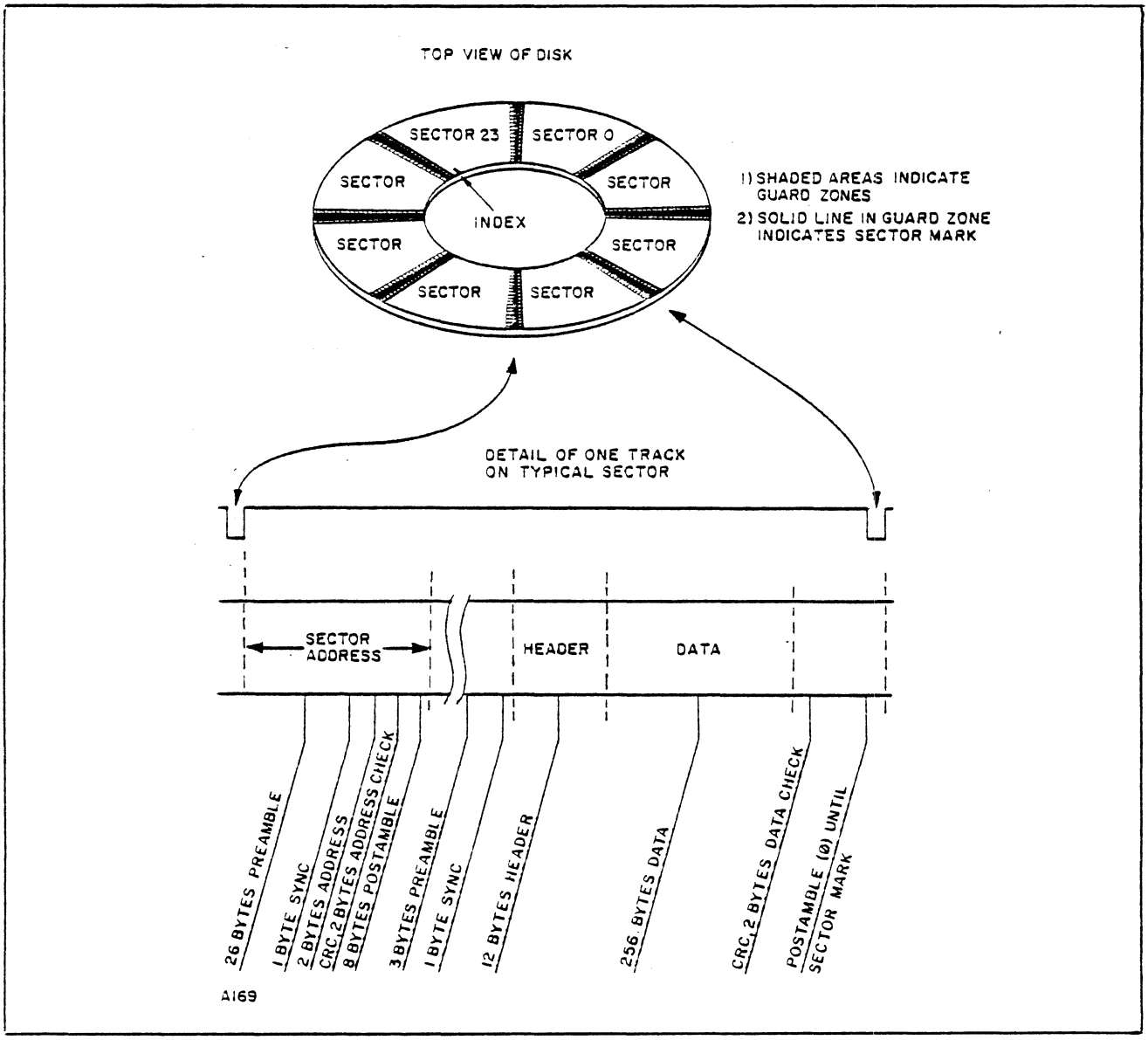


Figure 3-40. Disc Sector Format

Access to Memory.

Because of the high rate at which data can be transferred between a disc and memory, the disc controller is designed to be independent of the processor. It informs the processor when it requires memory access, and the processor coordinates this with the basic processor timing cycle. Once the disc controller has gained access to memory in this manner, the data transfer takes place over a direct access path; that is, the

data does not go through most processor circuits but is directly connected to the memory input/output multiplexers. Refer to the description of the processor architecture if a review of the processor timing cycle and organization is necessary.

Disc Instructions.

Discs are controlled by the processor with the 177 family of I/O instructions. The disc controller receives instructions from the processor in a specific order as determined by the programmer. In order to control interaction between the disc controller and the processor, timing is supplied by the processor to the controller and responses from the controller are provided to the processor.

The instructions available are shown in figure 3-41. Each of the instructions performs a task that coincides closely with the literal translation of its name, as follows:

a. 203 - Write Preparation. This instruction is part of a program that initializes the disc. It writes the logical address in the physical sectors specified by the programmer. This instruction would not normally be used again for that particular disc unless the format was changed or the addressing in some way became incoherent.

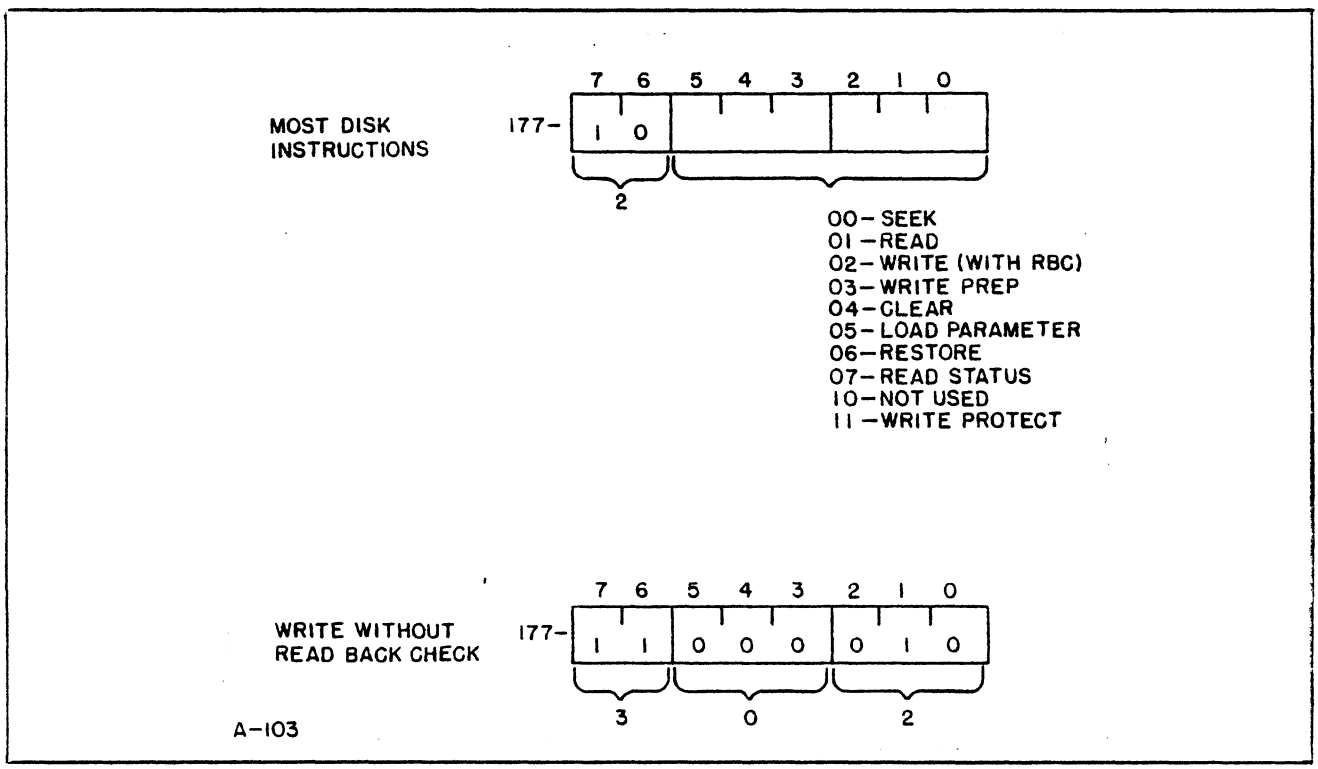


Figure 3-41. Format of Disc Controller Instructions

b. 205 - Load Parameter. This instruction passes operational parameters to the disc controller for each read or write operation. It contains all information that determines where in the memory data handling between the processor and the disc unit commences, and which disc and which track and sector on the selected disc are to be used. The format of the parameter bytes is shown in figure 3-42, and a typical parameter list appears in figure 3-43.

c. 200 - Seek. The Seek instruction causes the selected disc to search until the track address defined by the Load Parameter instruction is attained. Upon completion of the operation, the disc controller interrupts the processor.

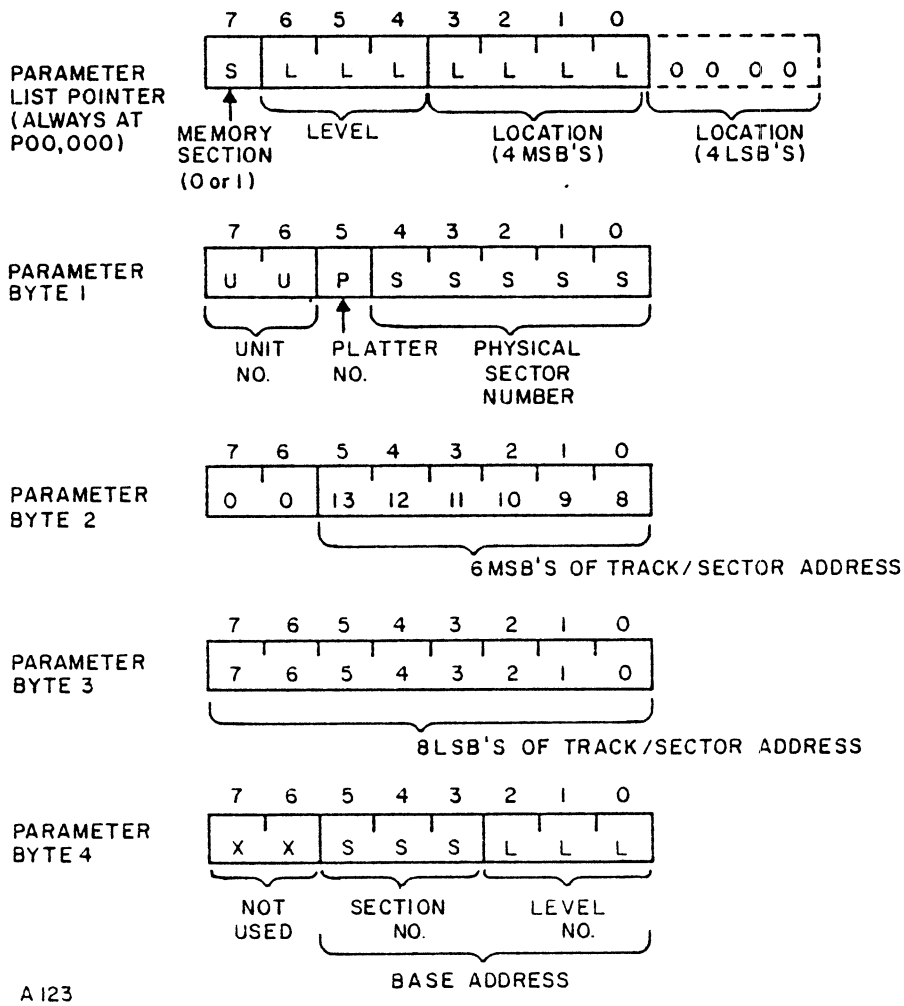


Figure 3-42. Format of Parameter Bytes

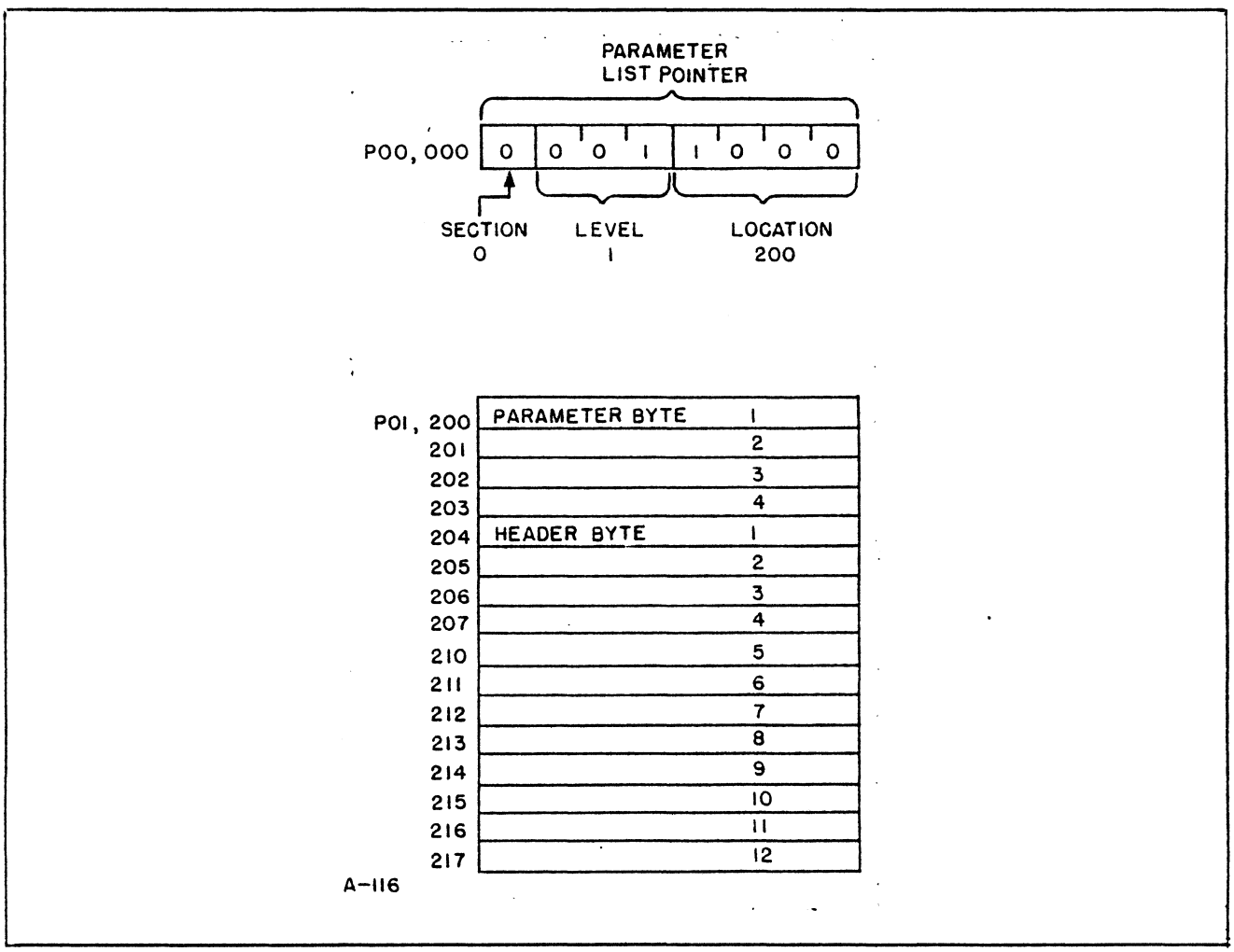


Figure 3-43. Typical Parameter Table

d. 201 - Read. The Read instruction first performs a seek operation to locate the specified physical track. A comparison is then made between the track/sector address word written on the disc and the track/sector address word defined by the load parameter operation. After equality is established, the 12-byte header and 256-byte data are read and transferred to the portion of memory specified by the parameter list information. Upon completion of the read operation, the disc controller interrupts the processor.

e. 202 - Write. The Write instruction first performs a seek operation in much the same manner as a Read instruction. When the sector is located, the 268 header and data bytes are written on the disc from the selected portion of memory. After this is completed, a readback check is performed. All that was written is compared byte-by-byte with the source data in the memory. When the readback check is completed, the processor is interrupted. A write operation without the readback check can also be used.

f. 206 - Restore. The Restore instruction is required whenever a seek incomplete status occurs. This instruction allows the read-write head to be positioned over track 0 on both sides of the disc.

g. 211 - Write Protect. The Write Protect instruction is used to prevent the writing on the selected disc. This capability may be restored by the depressing of a manual override PROTECT switch on the front panel of the disc unit. Note that this instruction applies only to the Model 1543.

h. 207 - Read Status. The completion of several types of operations and error conditions cause the disc controller to generate an interrupt signal to the processor. The processor must then issue a Read Status instruction to determine the cause of the interrupt. A status word is then read into memory location P00-000. However, the contents of location P00-000 cannot be guaranteed for more than one instruction time. Therefore, they must be read immediately after execution of the Read Status or Load Parameter instruction.

i. 303 - Write Without Readback Check. This operation is identical to the normal Write instruction with the exception that the interrupt is given at completion of the write operation and no readback check is performed.

j. 204 - Clear. The Clear instruction gives the programmer the means of providing a general reset to the disc interface circuits of the disc controller. This instruction would be used by the programmer initially, and following an error condition.

Disc Status Word.

The disc status word is acquired by the Read Status instruction. All eight bits have significance, and a status word of all zeros indicates a not ready condition. The meaning of each bit is described below. Note that the bit is a one to indicate the condition described.

a. Bit 7 - Second Command Error. A second command to disc unit was issued before completion of operation in progress.

b. Bit 6 - Address Error. The logical track/sector address not found.

c. Bit 5 - Ready to Seek, Read, or Write. This bit indicates that the disc is ready for operation. The power is on, the disc pack is loaded, the equipment drawer is closed, the LOAD/RUN switch is in RUN, the start-up cycle is complete.

d. Bit 4 - Write Protect Enabled. This bit indicates that the selected disc is write-protected.

e. Bit 3 - Interrupt. This bit indicates that a disc interrupt occurred.

f. Bit 2 - Readback Error. This bit indicates the readback check after a write operation failed due to one of several possible conditions.

g. Bit 1 - Write Check Error. This bit indicates that writing is not taking place properly due to one of several possible conditions.

h. Bit 0 - Seek Error. This bit indicates that the seek operation was not completed properly due to one of the following: (1) a seek to the improper track address occurred, (2) the command to move the heads to a specified track did not occur, or (3) a command to move the heads to a track address greater than 203 occurred.

Preparation of the Parameter Table.

As previously stated, the parameter bytes shown in figures 3-42 and 3-43 define the data source, memory locations, and logical disc address. The four parameter bytes are stored in memory and are sent to the disc controller when the Load Parameter instruction, IOC,C#7;205, is executed.

Refer to figure 3-43. The four parameter bytes are the first four bytes of the parameter table. The remaining bytes of the parameter table are the twelve header bytes. The parameter table must reside in section 0 or 1 of memory.

In order to locate a unique parameter table among the many tables that may be stored in memory, a parameter list pointer byte is provided. This byte is always stored in page 00, location 000 prior to executing the Load Parameter instruction. The pointer defines the page and location of the least significant address of the parameter table.

To summarize: the parameter list pointer and parameter table must be loaded prior to execution of the Load Parameter instruction. The Load Parameter instruction reads the parameter list pointer byte from page 00, location 000, and uses it to define the address of the first of the four parameter bytes. The four parameter bytes are then sent, in ascending order, to the disc controller. In turn, the disc controller stores these four parameter bytes for use in the read, write, or search operation to follow.

Listed below is a sample program that loads the parameter list pointer and first four bytes of the parameter table. The instructions required to load the twelve header bytes are not shown.

This parameter list will reside in memory, starting at page 01-020. The disc unit selected is unit 1, platter 1, the track/sector address 13-313, octal, and the memory transfer page is P11, octal.

```
LDX,R#1;OCT:020.        Set Index Register for Storage
LDA,R#Ø;OCT:140.        Select Unit 1, Platter 1
STA,I#1;PØ1.            Store Parameter Byte 1
LDA,R#Ø;OCT:013.        Load Most Significant Address
STA,I#1;PØ1.            Store Parameter Byte 2
LDA,R#Ø;OCT:313.        Load Least Significant Address
STA,I#1;PØ1.            Store Parameter Byte 3
LDA,R#Ø;OCT:011.        Load Page Transfer Base
STA,I#1;PØ1.            Store Parameter Byte 4
LDX,R#2;ØØØ.            Set Up for Storage in 000
LDA,R#Ø;Ø21.            Load Pointer with Parameter List
STA,R#2;PØØ.            Store In Location 000
IOC,C#7;2Ø5.            Execute Load Parameter Instruction
```

There are several points that must be taken into consideration when preparing and using the parameter list. They are:

a. The contents of location P00-000 can only be guaranteed for one instruction time; therefore, they should be loaded by the instruction preceding the Load Parameter instruction.

b. A minimum of a 2-instruction delay must be used between a Load Parameter instruction and a Read Status instruction.

c. A minimum of a 5-instruction delay must be used between a Load Parameter instruction and any other disc operation instruction; that is, Write, Read, Write Prep, or Seek.

Handling Disc Controller Interrupts.

As stated earlier, the disc controller has direct access to memory. This approach allows it to perform a read, write, or seek operation while the processor is executing a program. Upon completing an operation, the controller must notify the processor. This is achieved through the interrupt system.

When the controller completes an operation, an interrupt is generated. This causes program control to be transferred to page 03, location 000. The transfer is achieved by the interrupt forcing an automatic stack and branch to P03-000. Instructions starting at P03-000 will perform two tasks. First, all further interrupts must be locked out until the present interrupt has been processed. Secondly, a linkage must be provided to the interrupt handler routine.

The interrupt handler must make a test to determine if the interrupt is from communications (the Model 1539) or from the disc. A status word must be checked to make this decision. For the disc, this is done by executing the Read Status instruction (IOC,C#7;207). This instruction loads the disc status word into page 00, location 000. The interrupt handler can then examine the status word and determine the cause of the interrupt.

Intelligent terminals can handle two concurrent interrupts; that is, if two interrupts are generated at the same time, one will be handled and the terminal will remember that another interrupt is pending. If a third interrupt is generated, an interrupt overflow condition is set. This is recorded in bit 6 in the processor status word. The processor status word is available in the accumulator after execution of the Load Processor Status (LPS) instruction. An overflow is an error condition and would indicate that the processor is not keeping up with the I/O devices.

There are certain conditions that lock out interrupts. These conditions are:

a. An instruction causing exit condition (EXU or EXB).

b. (Any 17X-XXX instruction). Note that a stall condition on the keyboard prevents an interrupt from occurring.

c. Any accumulator shift instruction.

d. Any branch condition between memory sections.

e. The selected cartridge tape is moving (except rewind).

f. The interrupt lock-out flip-flop.

g. When operating with both disc and communications, the communications interrupt is locked out during the disc data transfer time. The duration of the communications interrupt lock-out depends upon the program being executed while disc data is being transferred. The average lock-out time is 1340 microseconds, with the shortest being 1072 and the longest being 1876.

The interrupt control instructions are discussed in Section 2 of this manual. A brief description is repeated here because of their importance in handling disc interrupts. Instructions involved are as follows:

a. 156-001 - Enable Processor Interrupt (EPI). After execution of this instruction, interrupts are allowed. Each causes an automatic stack and branch to P03-000.

b. 156-000 - Disable Processor Interrupt (DPI). After execution of this instruction, the automatic stack and branch to P03-000 is inhibited.

c. 156-002 - Clear Processor Interrupt (CPI). This instruction resets the interrupt overflow indicator.

d. 156-005 - Enable Interrupt - Reprocess. This instruction does not appear in the instruction list in Section 2, and there is no mnemonic code for the 156-005. Furthermore, this instruction is only executable on disc-equipped systems. The reason being that the decoding of the instruction is achieved partly in the disc controller and partly in the processor. Execution of this instruction followed by an EXU instruction enables interrupts but prevents the incrementing of the IAW. Thus, the return Exit from the interrupt subroutine will re-execute the interrupted instruction.

Programming Methods.

The programming examples in the following paragraphs cover the read, write, and interrupt routines used for the disc. These routines are explained in detail with respect to executing the instructions required to achieve the read and write functions. It is not the intent of these examples to cover the setup of the parameter list; it is assumed that the parameter list and pointer are set up before control is given to the read or write function. In these examples, register 6 is used as intermediate storage for the pointer. This is necessary because location P00-000 must be loaded just prior to execution of the Load Parameter instruction.

Read Function

Before entering the read function, the pointer is stored in register 6, and the parameter list starts at page 00, location 300. Program control is then transferred to program label E01 in table 3-13.

The sequence of events required to complete the read function is as follows:

a. Clear software interrupt switch.

b. Load parameters.

c. Check status.

d. Issue read instructions.

e. Go into a wait loop to wait until the read function is complete. (Interrupt is returned upon completion.)

f. Check for errors.

g. Return control to mainstream program.

The setup procedure starts at program label SUP, and the first step is to clear the software interrupt switch. This switch could have been set from a previous disc operation. When a disc operation is completed, an interrupt is generated by the disc and sent to the terminal; then when the interrupt signal is received, the program sets the switch. (The interrupt sequence is covered later in this section.)

A clear instruction (IOC,C#7;204) is sent to the disc to clear any conditions that may be present as a result of a previous disc function. It should be executed during initialization, and following an error condition.

Table 3-13. Disc Programming Routines

	Disc Read Routine	
E01:	SBU;SUP.	Disc I/O Setup
	BRH;RET.	On Error Retry High
	IOC,C#7;201.	Issue Read
	SBU;WIT.	Wait, Loop Until I/O Function is Complete
	BRL;*-8.	Error Return (Retry if Desired)
RET:	EXU;000.	Return: Set High If Error Set Equal If No Error
	Disc Write Routine (Read Back Check)	
E02:	SBU;WSU.	Disc Write Setup
	BRH;RET.	On Error Retry High
	IOC,C#7;202.	Write & Read Back
	SBU;WIT.	Wait, Loop Until I/O Function is Complete
	BRL;*-8.	Error Routine (Retry if Desired)
	EXU;000.	No Error: Return Equal
	Disc Write Routine (No Read Back Check)	
E32:	SBU;WSU.	Disc Write Setup
	BRH;RET.	On Error: Retry High
	IOC,C#7;302.	Write: No Read Back
	SBU;WIT.	Wait, Loop Until I/O Function is Complete
	BRL;*-8.	Error Routine (Retry if Desired)
	EXU;000.	No Error: Return Equal

Table 3-13. Disc Programming Routines (Continued)

Disc Write-Prep Routine		
E03:	SBU;WSU.	Disc Write Setup
	BRH;RET.	On Error: Retry High
	IOC,C#7;203.	Write - Prep
	EXU;000.	No Error: Return Equal
Disc Write Setup Routine		
WSU:	SMC;C#2.	
	SBU;SUP.	Common Setup
	BRH;RET.	On Error: Exit High
	LDA,R#0;STT.	Load Status
	ANA,R#0;OCT:020.	Write Protected?
	TLX,000;OCT:000.	No - Return Equal
	LDA,R#0;OCT:377.	Return High
	CPA,R#0;OCT:000.	
	EXU;000.	Exit
Disc I/O Setup Routine		
SUP:	SMC;C#2.	Set for Page 0
	LDA,R#0;OCT:000.	
	STA,R#0;ITP.	Clear to Interrupt Switch
	IOC,C#7;204.	Clear Disc
	LDX,R#1;OCT:000.	PML Pointer Location
	LDA,R#0;R#6.	Get PML Pointer
	STA,R#1;P00.	PML Pointer to P00-000

Table 3-13. Disc Programming Routines (Continued)

Disc I/O Setup Routine (Continued)		
	IOC,C#7;205.	Load Parameters
	ADA,R#0;OCT:000.	No Operation
	ADA,R#0;OCT:000.	No Operation
	IOC,C#7;207.	Read Status
	LDA,R#1;P00.	
	STA,R#0;STT.	Save Status
	TMX,000;OCT:040.	Ready to Seek, Read, or Write: Return Equal
	LDA,R#0;OCT:377.	Return High
	CPA,R#0;OCT:000.	
	EXU;000.	Exit
Wait for Disc I/O Complete Or Timeout		
WIT:	SMC;C#2.	Set for Page 0
	LDX,R#3;DEC:112.	320 ms Timeout
	EPI;000.	Enable Interrupt
	LDX,R#1;OCT:000.	Set 2.828 ms
WI1:	LDA,R#0;ITP.	Disc I/O Done?
	TMJ,+02;OCT:010.	Disc Interrupt
	BRE;WI3.	Yes. Check It
	SUX,R#1;OCT:001.	No - Wait for 2.82 ms
	CPX,R#1;OCT:000.	
	BRH;WI1.	
	SUX,R#3;OCT:001.	

Table 3-13. Disc Programming Routines (Continued)

Wait for Disc I/O Complete Or Timeout (Continued)		
	CPX,R#3;OCT:000.	Timeout:
	BRH;WI1.	No - Try Again
	DPI;000.	Timeout - Disable I
	IOC,C#7;204.	Clear Disc
WI2:	LDX,R#3;OCT:000.	Set for EXU High
	CPX,R#3;OCT:377.	Set Condition Register
	BRE;RET.	Return
	LDA,R#0;ERC.	Get Error Count
	ADA,R#0;OCT:001.	Increment Error Count
	STA,R#0;ERC.	Save New Value
	TLJ,-04;DEC:008.	If 8, Make It 9 to Force High
	LDA,R#0;ERC.	Error Count
	CPA,R#0;DEC:008.	Return Low if ERC Less Than 8, High If ERC Greater Than 8
	EXU;000.	
WI3:	LDA,R#0;STT.	Get Status
	ANA,R#0;OCT:307.	No - Protect/Ready
	TLJ,+10;OCT:000.	Any Errors: No
	TMJ,+04;OCT:001.	Seek Incomplete
	TMJ,+02;OCT:100.	Address Compare Error
	BRE;WI2-4.	
	BRU;WI2.	Other Error
	LDX,R#3;OCT:377.	Default EXU Equal
	BRU;WI2+002.	Exit Equal

Table 3-13. Disc Programming Routines (Continued)

Interrupt Handler		
P03-000	SMS;ITR.	Set Memory Section
P03-002	BRU;ITR.	Branch to Interrupt Handler
ITR:	SMC;C#2.	Set Memory Control
	STA,R#0;SAV.	Save Accumulator
	LPS;000.	Get Condition
	STA,R#0;SCR.	Save Condition for Return
	LDA,R#0;R#1.	
	STA,R#0;SR1.	Save Register 1
	LDX,R#1;OCT:000.	
	IOC,C#7;207.	Read Status
	LDA,R#1;P00.	
	STA,R#0;STT.	Save Status
	STA,R#0;ITP.	Set Interrupt Switch
	TMJ,+06;OCT:010.	Equal If Interrupt is From Disc
	LDA,R#0;SAV.	Restore Accumulator
	SBU,	If Other Interrupt Present (Example: Communications), Branch to Handler
	LDA,R#0;SR1.	Retrieve Register 1
	STA,R#0;R#1.	
	LDA,R#0;SCR.	Restore Condition

Table 3-13. Disc Programming Routines (Continued)

Interrupt Handler (Continued)	
SAC;000.	Register
LDA,R#0;SAV.	Restore Accumulator
OCT:(156-005).	Re-Enable Interrupt
EXU;000.	Return

Register 1 is used to hold the location portion of the address, page 00, location 000. This location will contain the parameter list pointer prior to execution of the Load Parameter instruction. The status word will also be loaded into page 00, location 000 after execution of the Read Status instruction.

The Load Parameter instruction (IOC,C#7;205) calls out the parameter list pointer from page 00, location 000 and uses the page and location address to define the starting location of the parameter list. Then, the four parameter bytes are sent to the disc controller. These four bytes are stored in control registers in the controller to be used during the read operation.

Two no-operation instructions are used to satisfy the hardware requirements. Their only use is to use up time to enable the hardware to completely execute the Load Parameter instruction. After the Load Parameter instruction has loaded the control registers in the controller, the Read Status instruction (IOC,C#7;207) checks the validity of the parameter bytes. If for any reason the parameter bytes are not valid, the controller does not return a ready status. A ready status indicates that the controller is ready to do a seek, read, or write operation. In this example, if the program detects anything other than a 040 status, the condition register is set to "high" and program control is returned to the main program.

Program control returns to E01+2. The branch on high is the return link in the event of an error in the parameter bytes. With no errors, the read instruction (IOC,C#7;201) is issued to the controller. Upon receiving this instruction, the controller searches for the address and then starts reading data from the disc and storing it directly into memory. This transfer from the disc to memory is achieved with no further intervention by the program. This is where the title "direct memory access channel" is derived.

There are two courses of action that can take place during the time it takes to transfer the 256 data bytes from disc to memory. They are:

a. The program can go into a wait loop and just cycle until the interrupt is generated by the controller to signal the completion of the read operation. The wait loop must check the interrupt switch to detect the completion.

b. The program can go into an unrelated task and operate on it during the transfer time. Again, the interrupt switch must be checked to determine when the read operation is complete.

In this example, the program goes into a wait loop. After issuing the read instruction, program control is transferred to program label WIT.

A 320-millisecond delay is setup to provide a time limit on the read function. If the delay expires before the interrupt is generated, the disc controller is cleared and program control transferred back to the mainstream program with an error condition indicated. The delay is setup by use of two registers, R3 and R1. Register R3 is loaded with a count of 112 decimals. Register R1 is loaded with zeros and is decremented 256 times until it again is zero. Each time R1 goes through zero, R3 will be decremented by 1; thus, R1 will be decremented 256 X 112, or 28,672 times. This figure times the total loop time is approximately equal to 320 milliseconds.

When an interrupt is generated, the interrupt switch (program label ITP) is set by the interrupt sequence. (The interrupt sequence will be covered later.) The switch is set by placing the status word in location ITP.

Refer to the status byte definition. When the controller generates an interrupt, bit 3 of the status word is set to a one. This condition is checked by the TMJ instruction at program label WI1+2. If no interrupt is present, the counter is decremented and control goes back to label WI1.

Upon receiving an interrupt, program control will be transferred to WI3 by the branch on equal instruction on line WI1+4. The status word is then checked for error indications. The "protect" and "ready" bits are masked out; they are not important at this point. If no errors are present, control is transferred to line WI3+14. The 377 is loaded into register R3 to be used to force the condition register to equal. At program label WI2+2, the busy switch is cleared. The condition register is set to equal, and program control is returned to the mainstream program.

The sequence starting at program label WI3 checks the status byte for errors. If an error is detected, program control is transferred to WI2-4 or WI2. If a seek or address error is detected, a clear instruction must be sent to the controller. This is done at program label WI2-4. A "high" condition is set to enable entry into the error count increment sequence. In this example, eight retries are attempted before the read is considered unsuccessful. If the count is less than eight, a read instruction is again executed after the error count is incremented. The EXU at program line WI2+18 returns control to label E01+8. With the condition register set to "low", the read instruction is executed again.

Write Function

With the write function, as with the read function, the parameter pointer must be stored in register R6, and the parameter list must be stored in memory. The setup of the parameter list is not covered in this description, and those portions of the write function that are the same as the read function are not covered again.

The sequence of events required to complete the write function is as follows:

a. Clear software interrupt switch.

b. Load parameters.

c. Check status.

d. Issue write instruction.

e. Go into wait loop to wait until the write function is complete. (Interrupt is returned upon completion.)

f. Check for errors.

g. Return control to mainstream program.

After the parameter list has been setup and the parameter list pointer loaded, program control is transferred to label E02. From E02, control transfers to the write setup sequence WSU. The write setup is basically the same as the read setup, with the exception that on a write sequence the status byte must be checked to determine if the disc is write-protected. The write protect indicator is bit 4 of the status byte. This test is made at program line WSU+8.

After the write protect test is made, control is returned to line E02+2 if the disc is not protected. If no errors were detected, the write with readback instruction is issued. The program then goes into a wait loop until the write function is complete. An interrupt is then generated by the controller. The interrupt sequence sets the interrupt switch ITP, which is checked by the wait sequence. Upon detecting the interrupt switch set, the wait routine checks for errors; if no errors are detected, control is transferred to the mainstream program via the write routine.

The wait sequence and the eight retry procedure are the same for the write routine as they are for the read routine. These sequences are not explained again.

Write With No Readback

The write without readback routine is the same as a write with readback from a programming aspect. The readback check is a hardware function for the write with readback instruction. The write with no readback is not covered in detail; however, the coding does appear in table 3-13, starting at program label E32.

Interrupt Handler Routine

When the disc controller generates an interrupt signal, further interrupts are disabled and control is transferred to page 03, location 000. This location is the start of the interrupt linkage sequence, which transfers control to the interrupt handler routine. In some cases, the programmer may wish to have the interrupt handler follow in sequence in page 03. This method is acceptable; however, most programmers put the handler in some other page. In this example, the interrupt handler resides at program label ITR.

The interrupt linkage sequence will branch to the interrupt handler routine at program label ITR. The interrupt handler routine will achieve the following:

a. Store the critical conditions.

b. Read the status byte and save it.

c. Set the software interrupt switch.

d. Check the status byte to see if the interrupt is from the disc.

e. If the interrupt is not from the disc, branch to a separate interrupt handler.

f. Restore critical conditions.

g. Enable interrupts.

h. Return.

The interrupt handler routine starts at program label ITR. The first step is to save the critical conditions; that is, the accumulator, condition register and processor status, and register R1. Register R1 will be used in the routine; therefore, its contents are saved. In a disc operation, register R1 probably holds the delay count at this time.

After the conditions have been stored, the next step is to get the status byte and save it. The Read Status instruction (IOC,C#7;207) will load the status byte into page 00, location 000. It is then stored in locations STT and ITP. Label ITP is the status switch, which is checked by the wait routine to determine when the interrupt occurred. The status byte is then checked to determine if the interrupt was generated by the disc controller. In that case, bit 3 of the status byte will be set to a one; otherwise bit 3 will be zero. When bit 3 is zero, control is transferred to another handler to handle the interrupt. The other interrupt sources are the program interrupt switch and the Model 1539 communications adapter unit.

Making the assumption that the interrupt is from the disc controller, control is transferred to line ITR+28. The next few instructions restore register R1, the condition register, and the accumulator. The enable interrupt instruction (156-005) again enables interrupts, and causes the exit instruction to return control to the instruction that was interrupted. The interrupted instruction is then executed.

In this example, the interrupted instruction is probably one of the instructions within the wait loop. In which case the wait sequence will detect the interrupt, by examining the interrupt switch, then checking the status of the disc controller.

APPENDIX A

MEMORY MAP

DRWG. NO. 001804-004

SCRATCH AREA ► SEQUENCE STORAGE AREA ◄ LABEL FIELD ◄ START BOOTSTRAP LOAD

END BOOTSTRAP LOAD →

PAGE 00 — XR1 XR2 XR3 XR4 XR5 XR6 XR7 CRC 1 2 3 4 5 6 7 SP0 SP1 SP2 SP3 SP4 SP5 SP6 SP7 SP8 SP9 SP10 SP11 SP12 SP13 SP14 SP15

PAGE 01

PAGE 02 — * MAIN PROGRAM STARTS

PAGE 03 — * INTERRUPT

SPARE

PAGE 04 — CRT BOT PATTERN

| | | |

APPENDIX B

STANDARD I/O LIBRARY 1

TABLE OF CONTENTS

STANDARD I/O LIBRARY 1

INSTRUCTIONS

1. INTRODUCTION TO I/O LIBRARY 1 90044

2. STANDARD BACKGROUNDS (B1A, B1B, B1C, B1D) 90030

2.1 Bootstrap/Loader 90024

2.2 Standard Dot Pattern 90021

2.3 Standard Keyboard Translate Table 90020

2.4 Standard Mini Read/Write Function 90023

3. BACKGROUND OPTIONS

3.1 Mini-Tape I/O Control 90022

3.2 Standard Maxi Tape Functions 90043

3.3 Extended Maxi Tape Functions 90028

3.4 Line Print Function (1551) 90040

3.5 Line Print Function (1555/1556) 90046

3.6 Matrix Print Function 90041

3.7 Typewriter Print Function 90047

3.8 Typewriter Single Character Print Function 90045

3.9 Standard Printer Control Options 90029

3.10 Printer Translate Table (1501 to USASCII) 90048

3.11 Printer Translate Table (1501 to Typewriter) ... 90049

PROCEDURE TITLE

STANDARD I/O LIBRARY 1 (80005)

Catalog No. 90044

PROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

1. INTRODUCTION

The Standard I/O Library 1 contains the four standard backgrounds and the optional functions, which will enable the user to build a specialized background to fit his needs. A list of backgrounds and optional functions contained on this Library are described in Table 1.

2. OPERATIONS

The I/O Library Monitor program was used to build this Library. For detailed operating instructions, see Document 90055. An example of how to retrieve a background and a print function is as follows:

2.1 Load the Standard I/O Library on Deck #2. The following will be displayed:

```
*SINGER 1500 SERIES*    XX/XX/XX
(LIBRARY NAME)   CAT.NO.XXXXX-XX
1. FILE          3. REVIEW LIBRARY
2. CALL          4. REORGANIZE
```

2.2 Depress "2" for the CALL Option. The following will be displayed:

```
CALL FROM
SEG: ID.  FROM-LOC.  TO-LOC.
SEG: ???  (PNN-LLL, PNN-LLL)
```

2.3 Key in the segment ID of the background required. (EX. B1A).

2.3.1 If a background is called from the Library, no other segments may be called at the same time.

2.3.2 Depress the "SKIP" Key to complete the entry.

2.4 Depress the "EOJ" Key. The following will be displayed:

```
FILE INTO

1. LOAD PROPER TAPE ON DECK-1.

2. LOAD "LIBRARY"   ON DECK-2.

    (DEPRESS START KEY)
```

2.4.1 Mount a scratch tape on Drive #1.

2.4.2 Depress the "START" Key. The Library tape on Drive #2 will locate the "B1A" background and copy it onto Drive #1.

2.4.3 After the B1A background has been copied from the Library, both tapes will rewind. The following will be displayed to indicate the completion of the "CALL" Option.

```
CALL FROM



      FINAL ENTRY COMPLETED
```

2.5 Leaving the tapes on Drive #1 (B1A) and Drive #2 (Library), repeat Steps 2.1 and 2.2.

2.6 Key in the Segment ID of the Print Function required (EX. LNP).

2.6.1 If an error is made during the key-in, depress the "BKSP FIELD" Key to re-enter that field.

2.6.2 Depress the "SKIP" Key to complete the entry.

2.7 Depress the "HOM" Key to continue entry of the next segment.

2.7.1 Key in the Segment ID of the Translate Table required with the LNP Function (EX. CT1).

2.7.2 Depress the "SKIP" Key to complete the entry.

2.8 Depress the "EOJ" Key, then the "START" Key.

2.8.1 The two functions will be located on the Library and superimposed on the B1A Background.

2.8.2 Both tapes will rewind. The specialized background is located on Drive #1.

3. COPY LIBRARY

The REORGANIZATION Option of the I/O Library may be used to copy the Library from Drive #2 to Drive #1. (See Document 90055).

TABLE 1

SEGMENT	CONTENTS	MEMORY LOC.	FUNCTIONS	CATALOG NO.
SDP	Standard Dot Pattern	P04-000 thru P05-063	Character Matrix for Display	60029-01
STT	Standard Kbd. Xlate Table	P05-064 thru P05-185	Translate Keyboard Code to 1501 Code	60030-01
MRW	Standard Mini Read/Write	P06-000 thru P07-241	Read/Write on Cartridge Tape	60031-02
IOC	Mini Tape I/O Control	P03-006 thru P03-239	Std. Mini Tape & Kbd. I/O Control	60144-00
M1X	Std. 9-Track Maxi (800 BPI)	P15-000 thru P15-235	Read/Write up to 256 Bytes from 9 Track (800 BPI)	60154-00
M7X	Std. 7-Track Maxi (556 BPI)	P15-000 thru P15-235	Read/Write up to 256 Bytes from 7 Track (556 BPI)	60155-00
M6X	Std. 9 Track Maxi (1600 BPI)	P15-000 thru P15-237	Read/Write up to 256 Bytes from 9 Track (1600 BPI)	60156-00
M1E	Extended Maxi Tape Func. Medium Density	P14-216 thru P15-255	Read/Write up to 1000 Bytes for 7 or 9 Track Med. Density	60150-00
M6E	Extended Maxi Tape Func. High Density	P14-216 thru P15-255	Read/Write up to 1000 Bytes for 9 Track, High Density	60151-00
LNP	Line Print Function	P14-000 thru P14-081	Print & Form Control for 1551, 1555, 1556 Printers	60032-02
MXP	Matrix Print Function	P14-000 thru P14-103	Print & Form Control for 1525/1526 Printers	60027-01
TYP	Typewriter Print Function	P14-000 thru P14-087	Print & Form Control for Typewriter	60033-01
P1S	Typewriter Single Character Print	P14-000 thru P14-093	Single Character Print & Control for 735 Typewriter	60034-01

SEGMENT	CONTENTS	MEMORY LOC.	FUNCTIONS	CATALOG NO.
C51	1551 Print Option	P14-080 thru P14-197	Special Form Control Option for 1551 Printer	60152-00
C55	1555/1556 Print Option	P14-080 thru P14-197	Special Form Control Option for 1555/1556 Printers	60172-00
C25	1525/1526 Print Option	P14-102 thru P14-197	Special Form Control Option for 1525/1526 Printers	60067-00
C35	Typewriter Print Option	P14-086 thru P14-197	Special Form Control Option for 735 Typewriter	60173-00
CT1	Printer Translate Table (1501 to USASCII)	P05-192 thru P05-255	1501 to USASCII Printer Translate Table	60158-00
TT1	Printer Translate Table (1501 to Typewriter)	P05-192 thru P05-255	1501 to Typewriter Translate Table	60159-00
B1A	Std. 4K Background	P01-000 thru P15-255	Load 4K Memory	70031-00
B1B	Std. 8K Background	P01-000 thru P31-255	Load 8K Memory	70032-00
B1C	Std. 16K Background	P01-000 thru P63-255	Load 16K Memory	70033-00
B1D	Std. 2K Background	P01-000 thru P07-255	Load 2K Memory	70072-00

STANDARD BACKGROUNDS

Page 1 Cont. On 2

PROCEDURE TYPE: Instructional

Catalog No. 90030

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

1. INTRODUCTION

There are four standard backgrounds, which are needed to generate a functional program. Each background contains the same resident functions, which aid and control the use of the Cartridge Tape Drives, CRT Displays, and Keyboard Translation.

1.1 These backgrounds are used by the Assembler (Object Generation) to superimpose the user's program onto it, so as to make a self-loading functional program.

2. SPECIFICATION

These backgrounds are located on the Standard I/O Library as a complete file, and will reside in memory as shown in Figure 1 after loading. The remainder of memory is cleared to zeros except the areas which hold the functional routines selected by the user.

SEG	FUNCTION	NO. OF BLOCKS	CATALOG NO.
B1A	4K Background	32 + EOF	70031
B1B	8K Background	64 + EOF	70032
B1C	16K Background	128 + EOF	70033
B1D	2K Background	16 + EOF	70072

2.1 The Segment ID is used to call these backgrounds from the Standard I/O Library.

2.2 The first two blocks of each background is the bootstrap and loader. The third block will start loading at P01-000.

2.3 Each block is 136 bytes. They are in the same format, except the Bootstrap, as the standard records (8 Byte Label + 128 Bytes of data).

2.4 The eight position of the label should not be altered. This flags the Assembler during Object Generation to where the data will be superimposed on the background.

2.5 The standard backgrounds are generated by using the Skeleton backgrounds (generated by Engineering) and putting on the resident functions listed on Table 1.

2.6 After the loading process, the linkage to the user's program is made to P02-000.

3. OPERATION

The user's program is responsible for the following:

3.1 Rewinding the program tape.

3.2 Control and loading of overlays.

3.3 The linkage from P02-000 to the start of his program.

4. BACKGROUND FUNCTIONS

4.1 The resident functions contained in the standard backgrounds are listed in Table 1.

4.2 Refer to the Bootstrap procedure for information on program loading and error definition.

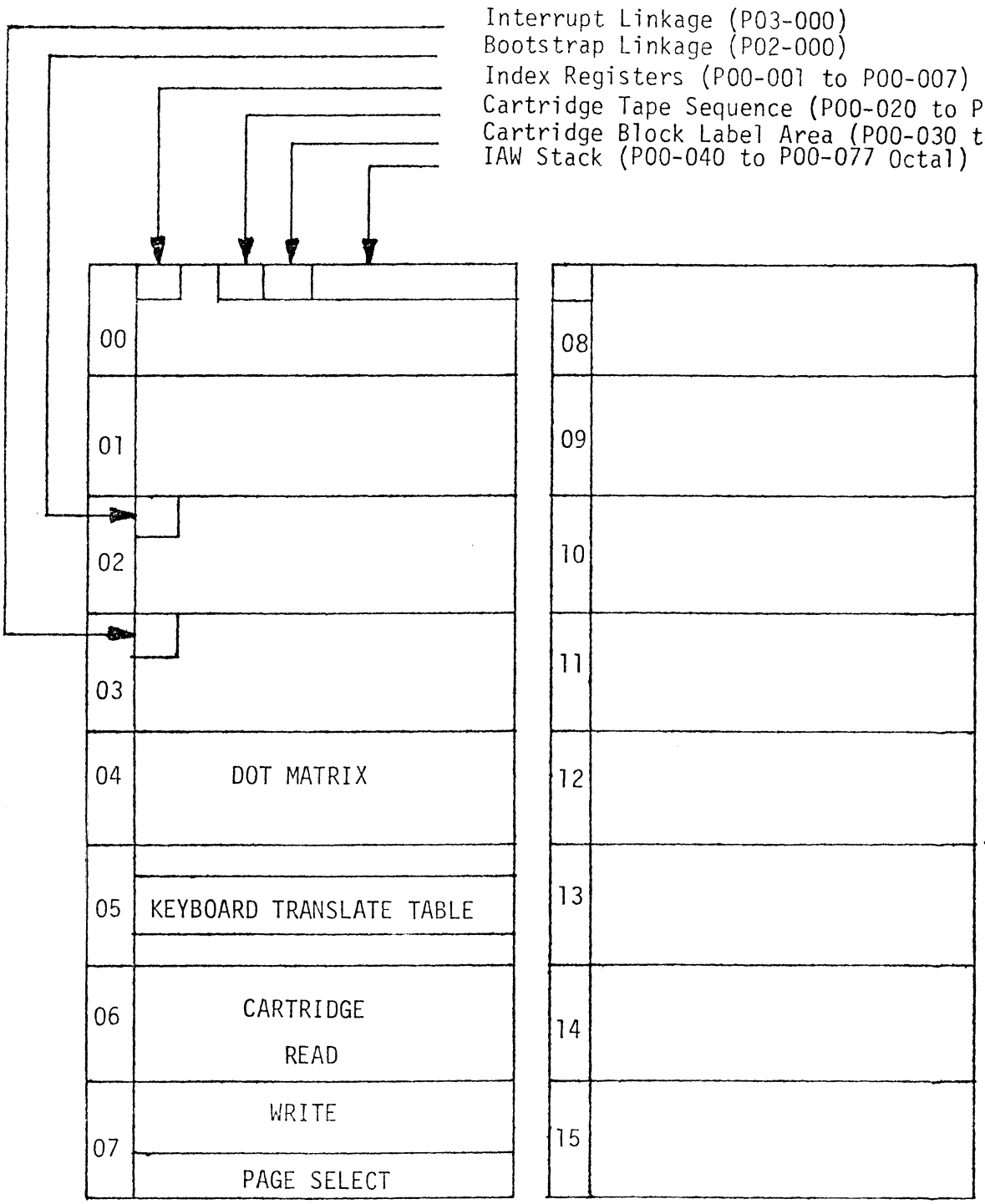
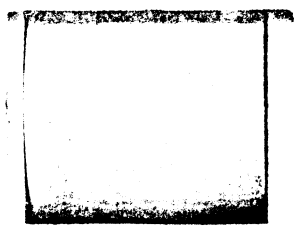


FIGURE 1. STANDARD BACKGROUND RESIDENT FUNCTIONS

TABLE 1

FUNCTIONS RESIDENT IN BACKGROUNDS "B1A - B1B - B1C - B1D"

SEGMENT	CONTENTS	MEMORY LOC'S LOADED (DECIMAL)	FUNCTION
BTS	Bootstrap & Loader	P00-024 thru P00-255 (Cleared after Load)	Load Program Tape.
SDP	CRT Dot Pattern	P04-000 thru P05-063	Character Matrix for Display
STT	Standard Keyboard Translate Table	P05-064 thru P05-121 P05-128 thru P05-185	Translate Keyboard Code to 1501 Code
MRW	Mini-Tape Read Write Function	P06-000 thru P07-183	Read & Write on Cartridge Tape
PSW	Page Select Write Function	P07-184 thru P07-233	Allow Write to Tape from various areas of memory.
PSR	Page Select Read Function	P07-210 thru P07-241	Allow Read from Tape into various areas of memory.



BOOTSTRAP/LOADER

Catalog No. 90024

EDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

1.0 INTRODUCTION

The Bootstrap and Loader records are the first two records on a Background tape and are designed to load the Background and the user's program into memory in modular blocks of data. The Program Load operation is initiated by the user when the external Program Load switch is activated. This causes the hardware to load the Bootstrap record from Deck 2 and start execution of the Bootstrap/Load Procedure.

2.0 SPECIFICATION

2.1 Segment ID - 'BTS'.

2.2 Source Catalog Number - 60028

2.3 Object Catalog Number - None

2.4 Special Requirements

2.4.1 Memory Requirements - P00-020 thru P00-377.

2.4.2 Registers Used - 1, 2, 3, 4, 5, 6.

2.4.3 Stack Pointer must be at Stack Level Ø.

3.0 METHOD

3.1 Record Characteristics -

3.1.1 Each physical record is comprised of an 8-byte label followed by 128 bytes of data.

3.2 Loading - (All locations are in Octal notation)

3.2.1 During loading, the Block label is stripped off and stored in P00-030 thru P00-037. The data portion of the records are sequentially loaded into memory starting at memory location P01-000.

3.3 Error Checks -

3.3.1 The Sequence number (first byte in the label field) is checked as each block is read from tape. If a sequence error occurs, the loading operation is terminated and a pulsed error tone is initiated.

3.3.2 If a tape error occurs, the block will be retried 8 times before stopping. The CRT will go blank. NO ERROR TONE IS GENERATED.

3.3.3 Each program block loaded into memory is read back and a CRC Check is performed, one byte at a time. This checks the memory and the data being stored. If this error occurs, a constant error tone is generated.

3.4 End of Load -

3.4.1 When a record has been loaded that contains an End of Load indicator (375 in the second byte of the block label) the Load operation will be terminated.

3.4.2 Locations P00-100 thru P00-367 are cleared to blanks. Locations P00-000 thru P00-077 are undetermined.

3.4.3 After completion of the above, the program causes an unconditional branch to P02-000 for linkage to the user program.

3.4.4 Rewinding the program tape must be done by the user program.

PROCEDURE TITLE: STANDARD DOT PATTERN

Page 1 Cont. On 2

Catalog No. 90021

PROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

STANDARD DOT PATTERN

Instructional

1.0 INTRODUCTION

The hardware uses the character code as a pointer within the standard Dot Pattern to determine the desired 5 x 8 dot matrix to display for a given character. Within the standard Dot Pattern it is possible to display 64 unique patterns. These patterns may be altered by the user to display any 5 x 8 matrix for any character. The Dot Pattern is part of the Standard Backgrounds.

2.0 SPECIFICATION

2.1 Segment ID - 'SDP'

2.2 Source Catalog Number - 60029

2.3 Object Catalog Number - None

2.4 Special Requirements

2.4.1 Memory Requirements - P04-000 thru P05-077 (Octal).

3.0 METHOD

3.1 The standard Graphic Display is for upper case Alpha characters, Numeric, and Special characters.

3.2 Example

3.2.1 Referring to the Memory Map layout found in the Programmer's Reference Manual we find that the character 'A' is defined as having the matrix values of 174, 022, 021, 022, and 174.

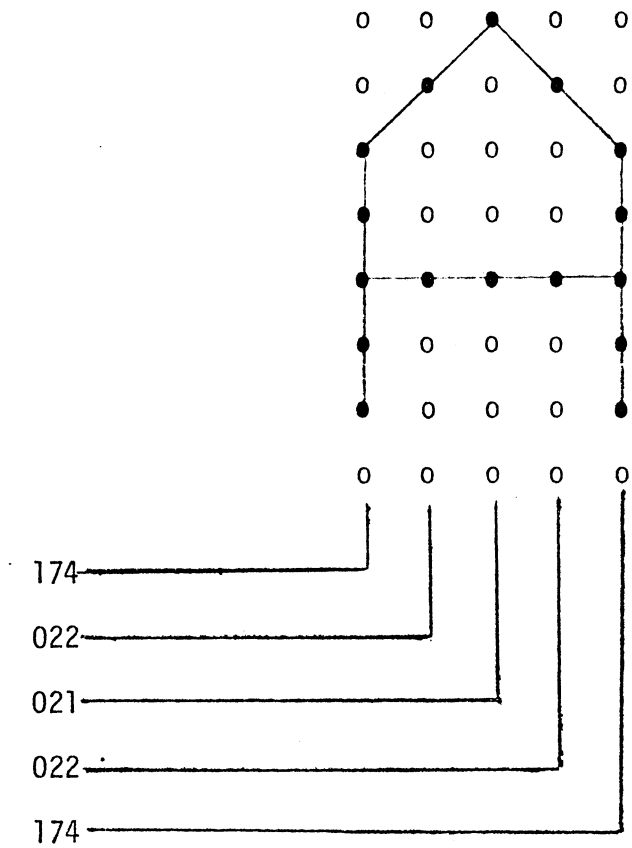
In Binary representation of each value, we have the following:

(174) 01 111 100
(022) 00 010 010
(021) 00 010 001
(022) 00 010 010
(174) 01 111 100

By shifting the 5 x 8 matrix 90^{0} in a counter-clockwise direction, we now have the 5 x 8 matrix displayed by the hardware.

STANDARD DOT PATTERN

Instructional



REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

STANDARD KEYBOARD TRANSLATE TABLE

Instructional

Catalog No. 90020

Date 03/01/74

1.0 INTRODUCTION

The Keyboard Translate Table is used to translate the hardware key code (which is generated by the hardware when a key is depressed) into the corresponding character code. The system is designed to provide code hardware transparency. Any keyboard character may be translated to any desired code. This Table is part of the Standard Backgrounds.

2.0 SPECIFICATION

2.1 Segment ID - 'STT'

2.2 Source Catalog Number - 60030

2.3 Object Catalog Number - None

2.4 Special Requirements

2.4.1 Memory Requirements - P05-100 thru P05-172 (Octal)
P05-200 thru P05-272 (Octal)

3.0 METHOD

3.1 Keyboard Entry -

When a character key is depressed on the Keyboard after a Keyboard Read, the hardware loads the key code into the Accumulator. The NUM (Numeric), CTRL (Control) and ALPHA (Alphabetic) are three special keys that act in Bits 6 and 7 of the key code for any key depressed in conjunction with one of them. NUM turns on Bit 6, CTRL turns on Bit 7, and ALPHA turns on both 6 and 7. If none of the special keys are depressed, Bits 6 and 7 remain off.

3.2 Translation -

3.2.1 The Control Bits (Bits 6 and 7) are taken care of as follows:

3.2.1.1 6th Bit ON: Do not change

3.2.1.2	7th Bit ON:	Turn 7th Bit off (reset after translation if desired).
3.2.1.3	6th and 7th Bits ON:	Turn 6th and 7th Bits off (reset after translation if desired).

3.2.2 Store the result in an index register.

3.2.3 Add to the index register the displacement within the page of the beginning of the Translate Table. The standard Translate Table in P05, for example, starts at decimal location 064. Therefore, add decimal 064 to the value of the index register containing the key code.

3.2.4 Load the Accumulator using indexed addressing and the page where the Translate Table resides. The Accumulator now contains the character code for the key that was depressed.

3.3 Special Printer Control characters are included in this Table.

PROCEDURE TITLE: STANDARD MINI READ/WRITE FUNCTIONS

Catalog No. 90023

PROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

TABLE OF CONTENTS

1. INTRODUCTION 4

2. SPECIFICATION 4

2.1 Segment ID 4
2.2 Source Catalog Number 4
2.3 Object Catalog Number 4

3. MINI-TAPE READ FUNCTION 5

3.1 Block Sequence 5

4. RETURN STATUS 7

4.1 Definition 7
4.2 Read Error Recovery Procedure 8

5. TAPE READ ENTRY POINTS 9

5.1 Tape Read 9
5.2 Time Delay Routines 9
5.3 Read Backspace 9

6. PARAMETERS 9

7. MINI-TAPE WRITE FUNCTION (WRF) 10

8. SEQUENCE NUMBER GENERATION 12

8.1 The Sequence Number of the First Block on Tape 12
8.2 Sequence Number Overflow 12

9. RETURN STATUS 13

9.1 Definition 13
9.2 Tape Error 13
9.3 Retry Error 14
9.4 Write Enable Pin Check 14

10. CARTRIDGE WRITE OPERATING INSTRUCTIONS 14

10.1 Parameters 14

11. BACKSPACE FUNCTION (BSF) 15

12. BACKSPACE FUNCTION (TIMED)..... 15

13. PAGE SELECT ROUTINE..... 16

13.1 Page Select Word..... 16
13.2 PSR Example..... 17
13.3 Variable Starting Location and Size Entry..... 17
13.4 Page Overflow..... 17

1. INTRODUCTION

The Mini Tape Read/Write handlers are resident functions of the standard Backgrounds (B1A, B1B, B1C, B1D). They are used to transfer data going to and coming from the standard Cartridge Tape Drives. The validity of the data during transfer and the block sequence is checked in several ways to insure that no data is lost.

1.1 To simplify the deck selection, a deck select word is loaded into the Accumulator before entering the Read or Write function. The deck select word is the number of the deck to be used (Ex. 001 for deck 1). These functions support up to eight cartridge drives.

2. SPECIFICATION

2.1 Segment ID - 'MRW'

2.2 Source Catalog Number - 60031

2.3 Object Catalog Number - None

3. MINI-TAPE READ FUNCTION

The Mini-Tape Read function reads a 136 byte block from a selected tape cartridge. The first 8 bytes of the block is the block label, and is always read into a fixed area, while the remaining 128 bytes may be read into any page of memory, starting at any location selected (using the Page Select Routine, PSR). With the Deck Select Word loaded into the accumulator, Stack and Branch to the RDF Function. Figure 1 is a short sample of the coding for the Read Tape Function. Figure 7 is a functional flow diagram showing the operation of the RDF routine.

```
TRD: LDA, R#Ø; DEC:001.        Select Deck 1
     SMS; S#0.                 Set Section
     SBU; RDF.                 Stack to Read Function
     BRH; TER.                 Return from RDF-Test for Error
     BRL; EOF.                 Check for End of File.
     EXU; 000.                 No error - Exit

TER:  (User's Handling Procedure)
```

Figure 1 - Read Tape Function-Sample Coding

3.1 Block Sequence -

The Sequence Check routine compares the sequence number of the block being read to the contents of the selected Sequence Storage location. This location is set to 001 by the Deck Select routine when a "Clip-In" status is detected. The first tape block sequence number thus is 001, and the value is incremented by "1" each time a block is successfully read. If the sequence number of the block being read from tape does not compare, the condition register is set HIGH, and the error is passed to the user.

3.1.1 Associated with each Tape Deck is a Memory Location which retains the binary sequence number of the last block which was correctly read from tape. These memory locations are listed in Figure 2.

FIGURE 2 -BLOCK SEQUENCE STORAGE

	SEQUENCE STORAGE LOCATION	DECK
Primary Decks	$P00\text{-}022_8$	Deck 1
	$P00\text{-}023_8$	Deck 2
Additional Decks	$P00\text{-}024_8$	Deck 3
	$P00\text{-}025_8$	Deck 4
	$P00\text{-}026_8$	Deck 5
	$P00\text{-}027_8$	Deck 6
	$P00\text{-}020_8$	Deck 7
	$P00\text{-}021_8$	Deck 8

3.1.2 If the sequence bypass flag is set when the deck was selected, the block is not sequence checked. However this flag is used to re-initialize the Sequence Storage location. When flag is detected, and the block is correctly read (no tape error or CRC check) the Sequence Number in the current block is transferred to the Selected Sequence Location. This feature will be generally used to initialize the Sequence Number in the high speed block locate routines. It also may be used in Sequence Error Recover Procedures.

3.1.3 The sequence bypass flag is set by having the higher order bit set on the Deck Select Word. (Ex. 201 is for Deck 1).

4. RETURN STATUS

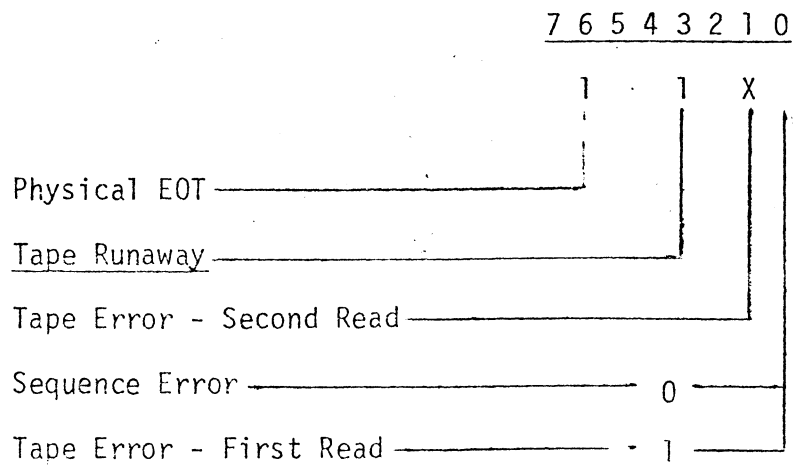
The Read Function status is passed to the user via the condition register. A Return Status is set to the condition of the function when terminated or completed. The status should be check at the completion of each function to insure that the block was read correctly.

4.1 Each block read is checked for Sequence Errors, Tape Errors, and CRC Errors. In the event of a tape error, the Read function will allow 8 retries before skipping the bad record and reading the next record. If that record is out of sequence a HIGH condition is set or if that record can not be read after 8 retries, a HIGH condition is set. The Status Word in R#4 will indicate this (Memory Location P00-004).

4.1.1 Read Return Status

CONDITION REGISTER	INDICATION
"="	Function completed - No errors
"<"	End of File
">"	Tape Error! Status Word contained in Register #4

4.1.2 Status Word in R#4



4.2 Read Error Recovery Procedure

HIGH after a Read operation will allow the user to examine the operation for three possible failures which may have existed. The procedure is as follows:

4.2.1 Check the tape status word stored in R#4.

4.2.2 If the lower order bit is on which means an 8-retry error occurred during the first read, a timed back up then a Read back up is executed to reposition tape, then the record is read again. The lower order bit of the tape status word is normally the keyboard error bit and is not used during the Read Operation. This bit is stripped off.

4.2.3 If the tape error bit is on along with the lower order bit, the recovery is handled the same way as 4.2.2. This indicates two records were bad (8 retry each).

4.2.4 The third condition which may exist is a sequence error without a tape error (No lower order bit or tape error bit on). To handle this, a timed back up then a Read with the sequence delete bit set. The record which was out of sequence is reread and the new sequence number is set.

4.2.5 To skip the bad record and continue with the next good record, the Recovery Procedure in 4.2.4 is used.

4.2.6 If a tape runaway has occurred, the result is an irrecoverable error. The position of the last "good" record can not be determined.

5. TAPE READ ENTRY POINTS

A summary of the subroutine linkages is shown in Table 1. Entry Points within this area are as follows:

5.1 Tape Read: RDF.

5.2 Time Delay Routines:

5.2.1 D91: 35 millisecond delay.

5.2.2 D92: Clear R#1 to 000 before entry produces a .7 second delay.

5.2.3 D93: Delay 2.828 milliseconds for every count in R#1.

5.3 Read Backspace:

5.3.1 BSF: Backspace the tape previously selected, one record (energy back up).

6. PARAMETERS

The Data I/O Buffer Parameters initially set as follows:

6.1 Base Address of data store instruction P01.

6.2 Address Pointer (Octal: 000).

6.3 Size of Record (Octal: 200).

See Paragraph 13. for Page Select Macro.

7. MINI-TAPE WRITE FUNCTION (WRF)

The Mini-Tape Write Function writes a 136-byte block to tape on a selected Deck. The first 8 bytes of the block is the block label, and is always written to tape from a fixed area of memory. This block label carries the block sequence number. The remaining 128 bytes may be written from any selected area, by specifying the output buffer using the Page Select Routine (PSW).

7.1 With the Deck Select Word loaded into the Accumulator, a Stack and Branch to the WRF routine is executed. The Write function writes one block to the selected tape deck, and delivers the block status to the Condition Register. A read-after-write is performed on each block to verify the data written on tape. A CRC byte is written at the end of the block and checked. The Write function allows 8 retries on an invalid compare before notifying the user's program. A Return Status is set in the Condition Register at the completion of the function.

7.1.1 Block Format

The format for the block written to tape is as shown in Figure 3. A "split write" technique is used, in that the 8 byte block label is always written memory Locations P00-030 to P00-037, while the 128 bytes may be written from any output buffer.

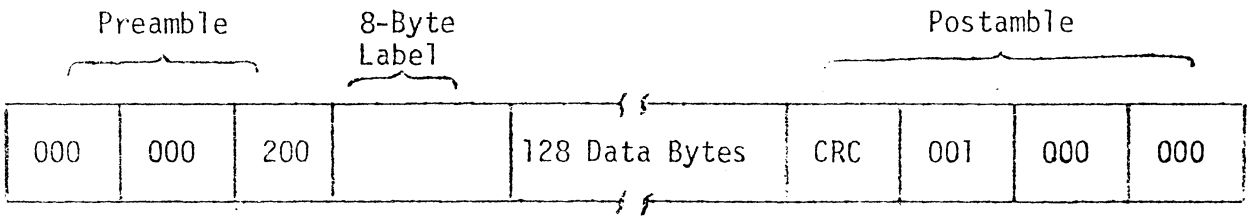


Figure 3 - Write Block Format

5. TAPE READ ENTRY POINTS

A summary of the subroutine linkages is shown in Table 1. Entry Points within this area are as follows:

5.1 Tape Read: RDF.

5.2 Time Delay Routines:

5.2.1 D91: 35 millisecond delay.

5.2.2 D92: Clear R#1 to 000 before entry produces a .7 second delay.

5.2.3 D93: Delay 2.828 milliseconds for every count in R#1.

5.3 Read Backspace:

5.3.1 BSF: Backspace the tape previously selected, one record (energy back up).

6. PARAMETERS

The Data I/O Buffer Parameters initially set as follows:

6.1 Base Address of data store instruction P01.

6.2 Address Pointer (Octal: 000).

6.3 Size of Record (Octal: 200).

See Paragraph 13. for Page Select Macro.

7. MINI-TAPE WRITE FUNCTION (WRF)

The Mini-Tape Write Function writes a 136-byte block to tape on a selected Deck. The first 8 bytes of the block is the block label, and is always written to tape from a fixed area of memory. This block label carries the block sequence number. The remaining 128 bytes may be written from any selected area, by specifying the output buffer using the Page Select Routine (PSW).

7.1 With the Deck Select Word loaded into the Accumulator, a Stack and Branch to the WRF routine is executed. The Write function writes one block to the selected tape deck, and delivers the block status to the Condition Register. A read-after-write is performed on each block to verify the data written on tape. A CRC byte is written at the end of the block and checked. The Write function allows 8 retries on an invalid compare before notifying the user's program. A Return Status is set in the Condition Register at the completion of the function.

7.1.1 Block Format

The format for the block written to tape is as shown in Figure 3. A "split write" technique is used, in that the 8 byte block label is always written memory Locations P00-030 to P00-037, while the 128 bytes may be written from any output buffer.

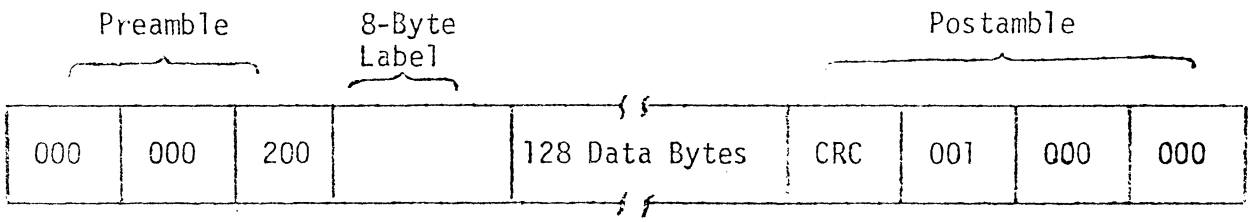


Figure 3 - Write Block Format

7.1.2 The output data area is initialized to P01-000 thru P01-177 (Octal) when the routine is loaded. However, the user may change the location of the output data area, if desired, by execution of the Page Select Routine prior to the execution of the Write Function.

7.1.3 At the start of the Write function, there is a delay of 2.1 seconds to allow the tape to advance approximately 21 inches before the first block to be written. Figure 4 is a short sample of the coding to link to the Write Function.

```
TWR: LDA, R#Ø; DEC: 002.      Select Channel (Deck 2)
     SMS; S#Ø.                 Set Section
     SBU; WRF.                 Stack to Write Function
     BRH; TER.                 Return from WRF-Test for error
     BRL; EOT.                 End of Tape
     EXU; 000.                 No error - Exit
```

TER: (User's Error Handling Procedure)

Figure 4 - Stack and Branch to Write Function

8. SEQUENCE NUMBER GENERATION

Integrated in the Tape Write Function is a Sequence Generator Sub-function. This Sequence Generator Sub-function tests the Status Word generated in the selection sub-function for the Clip In. If the Clip is in the deck selected, the Sequence Generator initializes the associated Sequence Storage Location to 000. Otherwise, the Sequence Storage Location is incremented and transferred to the Sequence Number Location of the Block Label Field (See Figure 3).

8.1 The Sequence Number of the first block on tape is $(001)_8$.

8.1.1 Associated with each tape deck is a Memory Location which retains the binary sequence number of the last block which was correctly written to tape. These areas are the same as in the Read Function (see Figure 2.)

8.2 Sequence Number Overflow

8.2.1 When the record Sequence Number byte reaches 377 (Octal) the next record to be written will cause an overflow to 000 (Octal).

9. RETURN STATUS

The Write function status is passed to the user via the Condition Register. A Return Status is set to the condition of the function when terminated or completed. The status should be checked at the completion of each function to insure that the block was written correctly.

9.1 Each block written is checked for Sequence Error, Tape Error, and CRC Error. In the event of a tape error, the Write function will allow 8 retries before notifying the user program. The Write function Return Status conditions are listed below.

9.1.1 Write Return Status

CONDITION REGISTER	INDICATION
"="	Function Completed - No Errors.
"<"	Physical End of Tape detected.
">"	Tape Error! Status Word contained in Register #4.

9.2 Tape Error -

If the Condition Register indicates a tape error (condition HIGH) the error status byte will be contained in Register #4 (Memory Location P00-004). Figure 5 shows the format of the Error Status Word located in Register #4.

9.2.1 Status Word in R#4

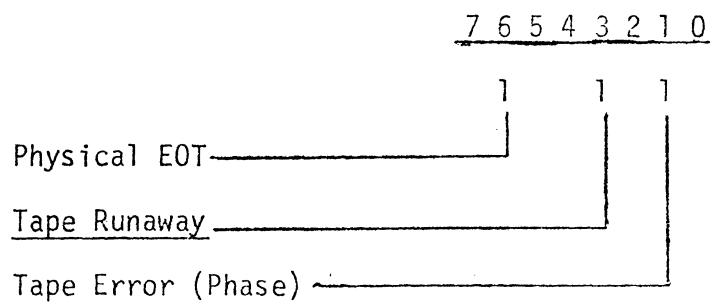


Figure 5 - Status Word

9.3 Retry Error -

An 8 retry error during a Write function will leave the "bad" record in front of the Write Head. If the Write function is executed at this time, the record will be advanced 2.4 inches (.3 of an inch for every retry). This procedure should not be repeated more than 10 times. Each retry uses an incremented erase skip of 0.3 inch, with a timed back up of 178 milliseconds.

9.4 Write Enable Pin Check -

A "Tape Runaway" error will be generated if a Tape Write command is executed and the write-enable pin is not in place in the cartridge. The tape will not advance.

9.4.1 To complete the Write operation, the user must remove the tape and insert the write enable pin before repeating the Write function.

9.4.2 To read the tape at this time, the user must first reset the Write Mode in order to prevent the tape from being erased during the Read operation. This is done as follows:

```
IOC, C#Ø; 003.    Tape Reverse (Erase Off)
IOC, C#Ø; 005.    Stop Tape
```

10. CARTRIDGE WRITE OPERATING INSTRUCTIONS

A summary of the subroutine linkages is given in Table 1. Entry points within this area are as follows:

Tape Write Function - WRF
Time Delay Routines - D91, D92

10.1 Parameters -

The data I/O Buffer parameters are initially set as follows:

10.1.1 Base Address of Data Load Instruction - Page 01.

10.1.2 Base Address of Data Compare Instruction - Page 01.

10.1.3 Address Pointer - Octal. 000.

10.1.4 Record Size - Octal: 200.

11. BACKSPACE FUNCTION (BSF)

The Record Backspace Function allows the tape to be backspaced one block when reading or writing. The user interfaces to the Backspace Function through the Deck Selection Routine. After the deck is selected, the Backspace routine is executed and the Sequence Storage Location is decremented by "1". The block which was backspaced will be located approximately ½ inch behind the Read Head. Figure 6 is a short sample of the coding to link to the Backspace Function.

```
LDA, R#Ø; OCT:001.        Select Deck
SMS; S#Ø.                 Set Section
SBU; BSF.                 Backspace Function

Continue
```

Figure 6 - Backspace Function-Sample Coding

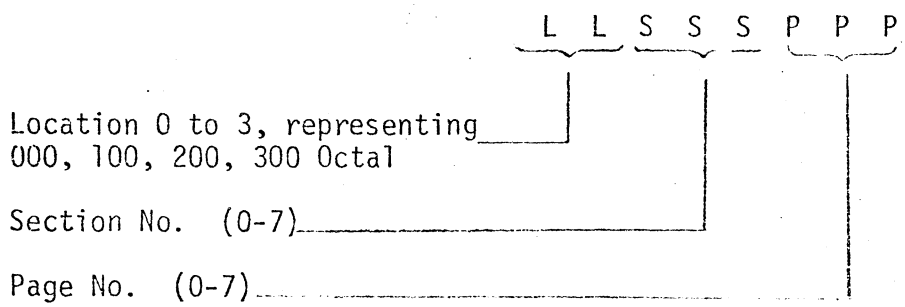
12. BACKSPACE FUNCTION (TIMED)

The Timed Backspace Function is used only for the Read Error Recovery procedure. It positions the tape for a Read Retry. This function acts upon the current Deck selected. (Entry Linkage is D84).

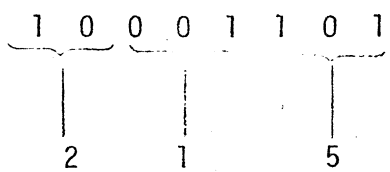
13. PAGE SELECT ROUTINE

The Tape Read and Tape Write routines are set to read or write into P01-000. If the user desires to read or to write into another area of memory, a Stack and Branch instruction to the desired function (PSR or PSW) is required. The accumulator must first be set to the desired Page and Location for the data. (Page Select Word).

13.1 Page Select Word -



Thus:



13.2 Figure 13.2.1 shows an example of the Page Select Read (PSR) routine. See Table 1 for Equate Reference.

13.2.1 Read 128 Characters into P08 starting at Location 128.

```
LDA, R#Ø; OCT:210.      LL-SSS-PPP
SMS; S#Ø.               Set Section Ø
SBU; PSR.               Enter Function
SBU; TRD.               Read Tape
```

13.3 Variable Starting Location and Size Entry (SSW, SSR)

The purpose of this option is to provide the user with a method of selecting any starting Location and Data Size desired.

13.3.1 Location - Load the Accumulator with the desired starting Location.

13.3.2 Page - Load Register Number 1 with the desired Page Select Word.

13.3.3 Size - Load Register Number 2 with the desired Data Size. The standard Record will be transferred and verified but only the data within the Size specified will be stored (SSR) or written (SSW).

13.4 Page Overflow -

If the starting Location for the Read falls between decimal 128 and 255, there is a Page Wrap Around and the residual characters are stored starting at Location 000 of the selected Page.

SUMMARY OF SUBROUTINE LINKAGES TO STANDARD BACKGROUNDS

PAGE SELECT READ	PSR: EQU: P07, 224. ACC = LLSSSPPP	(OCT: 340)
PAGE SELECT WRITE	PSW: EQU: P07, 184. ACC = LLSSSPPP	(OCT: 270)
READ FUNCTION	RDF: EQU: P06, 000. ACC = Unit #1 thru 8.	(OCT: 000)
WRITE FUNCTION	WRF: EQU: P07, 036. ACC = Unit # 1 thru 8.	(OCT: 044)
B.S. FUNCTION	BSF: EQU: P06, 030. ACC = Unit #1 thru 8.	(OCT: 036)
DELAY 35 MILLISECONDS	D91: EQU: P06, 206.	(OCT: 316)
DELAY (R#1-1)	D92: EQU: P06, 208. With R#1=000, .7 seconds delay.	(OCT: 320)
DELAY (CPX)	D93: EQU: P06, 210. R#1=XXX, 2.828 milliseconds per count of R#1.	(OCT: 322)
VARIABLE READ	SSR: EQU: P07, 230. R2 = Size, ACC = Loc., R1 = Page	(OCT: 346)
VARIABLE WRITE	SSW: EQU: P07, 190. R2 = Size, ACC = Loc., R1 = Page	(OCT: 276)
TIMED BACKSPACE (READ)	D84: EQU: P06, 196. Current Deck Selected (129 MSD)	(OCT: 304)

TABLE 1

SUMMARY OF OPERATING INSTRUCTIONS - MINI-TAPE

RETURN CONDITIONS:

" " Physical EOT (Write) or EOF recognition (Read 376 or 377 in second byte of label)

" " Error where R#4 carries the status bits:

XØØ:	Segment Check Error	1XX:	Physical EOT
XØ2:	Tape Error	XXX:	Any combination of the above.
X12:	Runaway		
X2X:	Cassette Out		

REGISTER USAGE: (All in Section Ø)

PSR, PSW:	R#1 & 2.
SSR, SSW:	R#1 & 2.
RDF, WRF:	R#1 thru 5, Scratch Loc. 8 (DEC)
BSF:	R#1, R#2, & R#5
D91, D92, D93:	R#1

MACRO LOCATIONS:

RDF, BSF, D91, D92, & D93:	P06, 000 thru P07, 035. (Decimal) Not Optional
WRF:	P07, 036 thru 183. (Oct: 044-267) User may Overlay
PSW: (See Below)	P07, 184 thru 209. (Oct: 270-321) User may Overlay
PSR: (See Below)	P07, 224 thru 241. (Oct: 340-361) User may Overlay
SSW: (See Below)	P07, 190 thru 209. (Oct: 276-321) User may Overlay
SSR: (See Below)	P07, 230 thru 241. (Oct: 346-361) User may Overlay
Common Page Select Area	P07, 210 thru 223. (Oct: 322-337) User may Overlay

SEQUENCE NUMBERS: (First Block = #001)

Unit 1 & 2:	P00, 018 thru 019 (Oct: 022-023) Not Optional
Unit 3 & 4:	P00, 020 thru 021 (Oct: 024-025) Optional Scratch Locations
Unit 5 & 6:	P00, 022 thru 023. (Oct: 026-027) Optional Scratch Locations
Unit 7 & 8:	P00, 016 thru 017. (Oct: 020-021) Optional Scratch Locations
READ/WRITE BUFFER:	P01, 000 - Unselected
LABEL LOCATIONS:	P00, 024 thru 031. (Oct: 030-037)

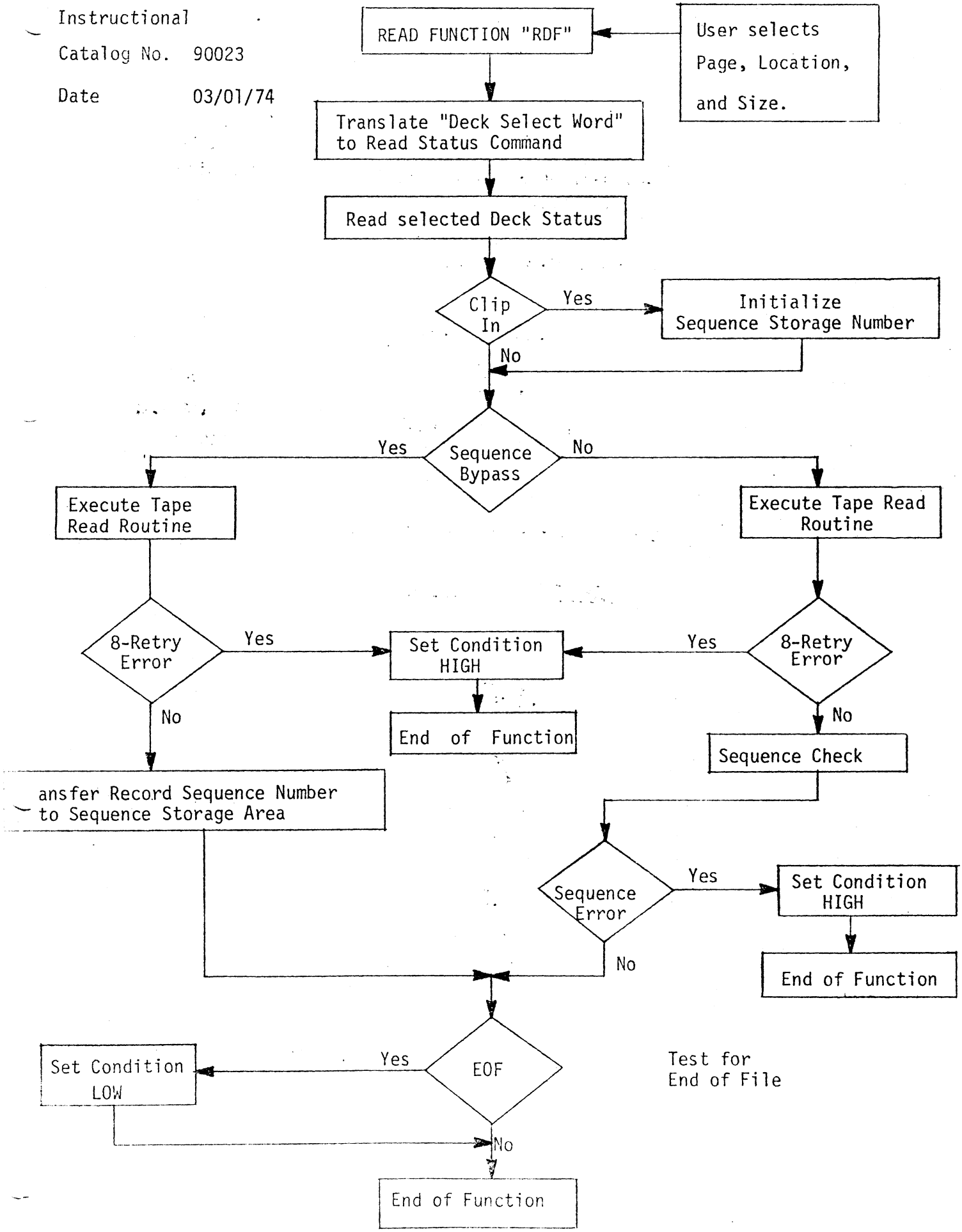


Figure 7 - RDF ROUTINE - FUNCTIONAL FLOW DIAGRAM

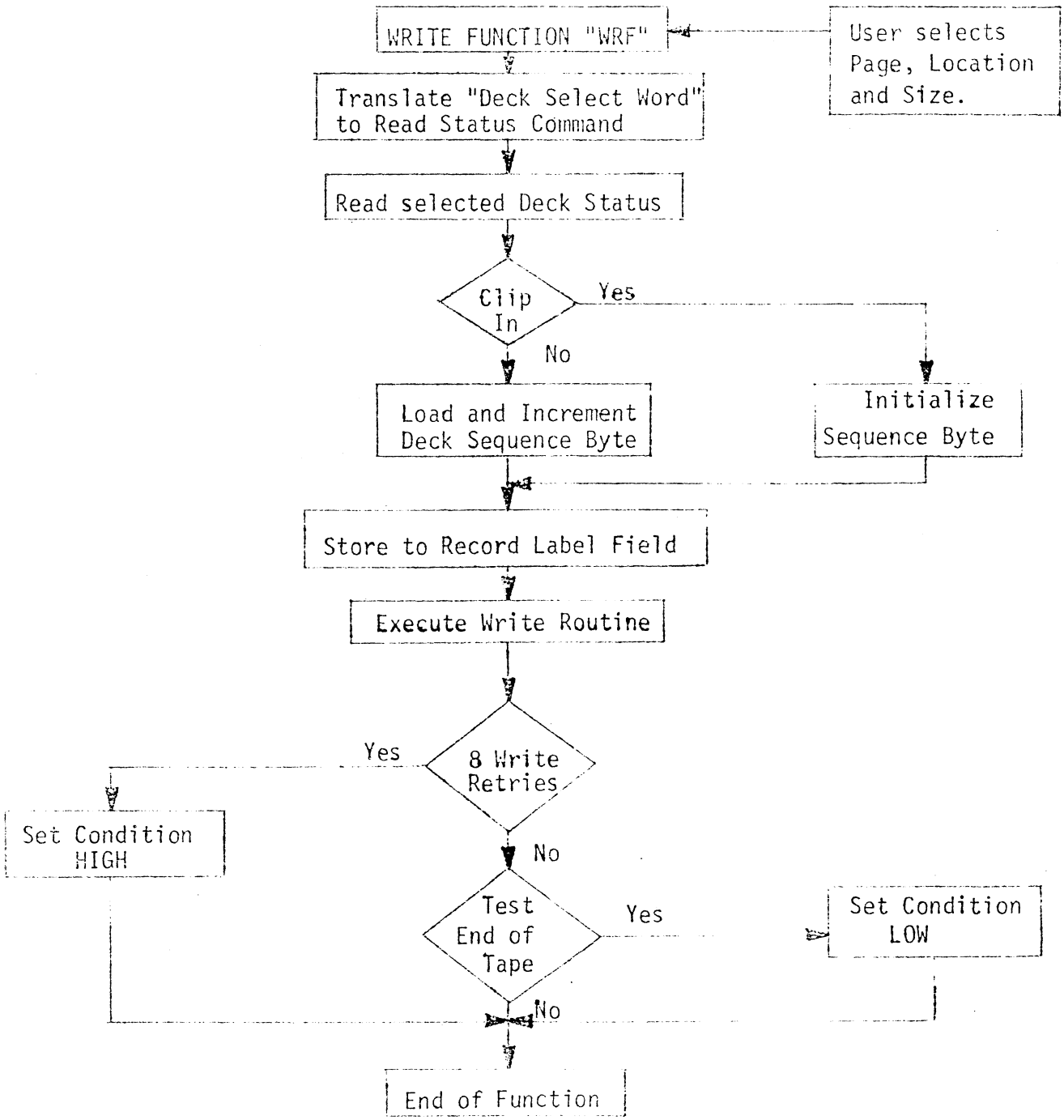


Figure 8 - WRF ROUTINE - FUNCTIONAL FLOW DIAGRAM

PROCEDURE TITLE: MINI-TAPE I/O CONTROL

Catalog No. 90022

PROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

TABLE OF CONTENTS

1.0 INTRODUCTION 3

2.0 SPECIFICATION 3

3.0 READ/WRITE MINI TAPE HANDLERS 4

3.1 Description 4
3.2 Deck Select Word 4
3.3 Read Operation 5
3.4 Write Operation 6
3.5 Error Detection 6
3.6 Error Notification 6
3.7 Error Recovery Procedures 6
3.8 Special Requirements 7
3.9 Entry Point Linkage 7

4.0 WRITE END OF TEXT (MINI TAPE) 8

4.1 Description 8
4.2 Operating Procedures 8
4.3 Special Requirements 8
4.4 Entry Point Linkage 8

5.0 COMPRESSED WRITE OPTION 9

5.1 Description 9
5.2 Operating Procedures 9
5.3 Special Requirements 10
5.4 Entry Point Linkage 10

6.0 KEYBOARD ENTRY 11

6.1 Description 11
6.2 Operating Procedures 11
6.3 Special Requirements 11
6.4 Entry Point Linkage 12

7.0 HIGH SPEED SEARCH 13

7.1 Description 13
7.2 Operating Procedures 13
7.3 Example 15

8.0 LOAD PROGRAM TAPE OPTION 16

8.1 Description 16
8.2 Function Usage 16
8.3 Special Requirements 16
8.4 Entry Point Linkage 16

1.0 INTRODUCTION

The Mini Tape I/O Control segment contains several Input/Output Control functions that may be accessed by the user to perform such I/O operations as: Mini Read/Write (with error notification and recovery). Keyboard Read and Translate, End of Text Write function, Compressed Write Option, High Speed Search function, and a special Program Load operation.

Functions not utilized may be overlayed with a user program.

2.0 SPECIFICATION

2.1 Segment ID - 'IOC'

2.2 Source Catalog Number - 60144

2.3 Object Catalog Number - None

3.0 READ/WRITE MINI TAPE HANDLERS

3.1 Description -

These functions are designed to standardize the usage of the Mini Tape Read/Write macros by enabling the user to select the desired Deck, perform the Read or Write operation, and provide for a standard method of error detection, notification and recovery procedure.

3.2 Deck Select Word -

Before linking to the Read Handler or the Write Handler, the Accumulator must be loaded with the number of the Deck to be selected. Bit 7 of the Deck Select Word is a flag position which allows the bypass of the block sequence check during the Read function. If the sequence bypass option is used, a new sequence number base is generated by transferring the sequence number of the block read to the proper sequence storage location. The format of the Deck Select Word is as follows.

	Binary Format of Deck Select Word (7 6 5 4 3 2 1 0)	Generated Deck Select Command
Select Deck 1 Internal	0 0 0 1	171 016
2 Decks	0 0 1 0	172 016
3 Optional	0 0 1 1	171 026
4 Decks	0 1 0 0	172 026
5	0 1 0 1	171 036
6	0 1 1 0	172 036
7	0 1 1 1	171 046
8	1 0 0 0	172 046

Sequence Bypass Flag (Read Only)

Deck Select Field

3.3 Read Operation -

Load the Accumulator with the desired Deck Select Word explained above, then stack and branch to the Mini Read Handler. Error detection, notification and recovery procedure is handled by the Mini Tape Read Handler. When the program returns from this function, the record has been successfully read. If the End of Cartridge Record has been read, the hardware Condition Register is set low.

3.3.1 Example:

```
LDA, R#Ø; OCT:001.      Select Deck-1
SMS; S#Ø.               Set Control Section
SBU; RDS.               Read Handler
BRL; EOC.               End of File?
                        No - Continue.
```

3.4 Write Operation -

Load the Accumulator with the desired Deck Select Word, then stack and branch to the Mini Write Handler. Error detection, notification and recovery procedure is handled by the Mini Tape Write Handler. When the program returns from this function, the record has been successfully written. After the Write operation, the user may desire to test for the physical End of Tape indicator.

3.4.1 Example:

```
LDA, R#Ø; OCT:002.      Select Deck-2
SMS; S#Ø.               Set Section
SBU; WTS.               Write Handler
BRL; EOT.               End of Tape?
```

3.5 Error Detection -

The standard Mini Tape Read and Write functions provide for an automatic error retry of 8 times. If, at this time, the record has not been processed correctly, the operator will be notified.

3.6 Error Notification -

If an 8-retry error has occurred, a continuous error tone will be sounded.

3.7 Error Recovery Procedure -

3.7.1 Depress the 'ERROR' Key to retry the function 8 more times.

3.7.2 Depress the 'CTL-ERROR' Keys to bypass and ignore the error record.

3.7.3 Any other key will terminate the alarm and await either of the above responses.

NOTE: If a Mini Tape Read Runaway has occurred, a continuous error tone will be sounded. No recovery procedure has been provided.

3.8 Special Requirements -

3.8.1 The standard Mini Tape Read/Write routine must be used.

3.8.2 Memory Requirements -

3.8.2.1 The Mini Tape Read/Write Handlers reside in Memory Locations P03-006 thru P03-131 (Octal).

3.8.2.2 Selected Deck Number Save Area resides in Memory Location P00-015 (Octal).

3.9 Entry Point Linkages -

3.9.1	RDS: EQU: P03, 006.	Mini Read Handler
3.9.2	WTS: EQU: P03, 048.	Mini Write Handler
3.9.3	ERR: EQU: P03, 062.	Error Alarm

4.0 WRITE END OF TEXT (Mini Tape)

4.1 Description -

The Write End of Text function records 6 inches of energy on tape. This forms a special tape mark record, which is used to stop a high speed search operation by setting the runaway error. The function of this is to locate the end of a file quickly.

NOTE: No attempt should be made to read this record.

4.2 Operating Procedures -

4.2.1 After writing an End of Cartridge record, Stack and Branch to the Write End of Text function. The End of Text will be written on the current deck selected.

4.3 Special Requirements -

4.3.1 The Write End of Text function utilizes the time delay and stop tape features of the Mini Tape Read function.

4.3.2 The Write End of Text function resides in Memory Locations P03-132 thru P03-153 (Octal).

4.4 Entry Point Linkage -

4.4.1 EOT: EQU: P03, 090. Write End of Text

5.0 COMPRESSED WRITE OPTION

5.1 Description -

The Compressed Write function is used in conjunction with the Mini Tape Write routine to shorten the IRG (Inter-record Gap) from the normal gap size of 1.1 inch to .43 inch. This enables the approximate tape capacity to be increased from a maximum of 500 records (normal gap) to a maximum of 900 records (compressed gap).

5.2 Operating Procedures -

There are two approved methods of utilizing the Compressed Write feature.

5.2.1 Method 1:

Within the user program include the source statement to alter the standard Mini Tape Write function to enter the Compressed Write feature upon completion of the Write operation. Using this method, when the program returns from the Write operation, the tape will automatically be positioned for a Compressed Gap.

5.2.1.1 Example:

```
ORG: P07, 168.
BRE; CPW.            Compressed Write
```

5.2.2 Method 2:

After execution of the normal Tape Write procedure, Stack and Branch to the Compressed Write feature.

5.2.2.1 Example:

```
LDA, R#Ø; OCT:002.     Select Deck-2
SBU; WTS.              Write Record
SBU; CPW.              Compress Gap
LDA, R#Ø; R#4.         Load Status
TMJ, +04; OCT:100.     End of Tape?
```

5.3 Special Requirements -

5.3.1 The Compressed Write function utilizes the time delay and stop tape features of the Mini Tape Read function.

5.3.2 The Compressed Write function resides in Memory Locations P03-154 thru P03-167 (Octal).

5.3.3 When using Method 1, the standard Mini Tape Write function must be used.

5.4 Entry Point Linkage -

5.4.1 CPW: EQU: P03, 108. Compressed Write

6.0 KEYBOARD ENTRY

6.1 Description -

The Keyboard Entry function provides a standard method of keyboard entry and translation. There is also a numeric override function which automatically translates any key entered to the numeric value. Also, if the 'CTRL' Key was depressed in conjunction with a functional key, this will be indicated in the translated result.

6.2 Operating Procedures -

6.2.1 Alpha/Numeric Entry:

This is the normal mode of keyboard entry whereas any key depressed will be translated to the Alpha value unless the 'NUM' Key is also depressed. With the 'NUM' Key depressed, a numeric translation will occur.

6.2.2 Numeric Override -

In this mode of operation, any key depressed will automatically be translated to the Numeric value.

6.2.3.1 After execution of the Keyboard Entry function, the translated result will be in the Accumulator and in Register 1.

6.2.3.2 If the 'CTRL' Key was depressed, the translated result will contain the 6th Bit on.

6.3 Special Requirements -

6.3.1 Register Usage:

Registers 1, 6, and 7.

6.3.2 Memory Requirements:

The Keyboard Entry function resides in Memory Locations P03-170 thru P03-251 (Octal).

6.3.3 Translate Table:

The Translate Table must reside in the standard assigned area (P05-100 thru P05-217 (Octal)).

6.4 Entry Point Linkage -

6.4.1 KEY: EQU: P03, 124. Numeric Override Entry

6.4.2 ALP: EQU: KEY-04. Alpha/Numeric Entry

7.0 HIGH SPEED SEARCH

7.1 Description -

The high speed search function provides the user with a method of advancing the tape in a forward or a reverse direction at high speed (40 inches per second). The distance to be searched is based on a record count controlled by the user program.

7.2 Operating Procedures -

7.2.1 Set-Up

7.2.1.1 The basic High Speed Search function, after detecting a record at high speed, will enter the user program to determine if the search has been completed. To perform this function, the user must alter the High Speed Search procedure to provide linkage to the user update counter procedure.

Alter the High Speed Search as follows:

```
ORG: P03, 180.

SMS; S#X.          Set Section

SBU; UDC.          User Update Counter
```

7.2.2 User Update Counter Procedure

7.2.2.1 Test Status for a tape runaway. This would indicate that the file has been exceeded and that an end of text record has been detected.

```
LDA, R#Ø; R#4.     Load Tape Status

TLX, 000; OCT:010. Runaway?
```

7.2.2.2 If NOT a tape runaway, decrement the Search Counter and test for the end of the Search operation.

- If not at the end, set the Condition LOW and exit. This will return control back to the High Speed Search function whereupon it will wait for another block to be detected.
- If at the end, set the Condition Register EQUAL and exit. This will return control back to the High Speed Search function whereupon it will stop the tape and return to the user program.

7.2.3 Execution of High Speed Search

7.2.3.1 Select Deck to be searched.

7.2.3.2 Set up user Search Counter. The Search Count should be 5 blocks short of the number required. This takes into account the blocks lost during the start and stop time of the Search function, and a 2 block safety factor.

7.2.3.3 Stack and Branch to the High Speed Forward (HSF) function or the High Speed Reverse (HSR) function.

7.2.3.4 When the High Speed Search function has been completed, perform the following:

- Test the status (in R#4) for a tape runaway.
- Read a record with the Sequence Delete Bit On.

7.2.3.5 The record must be checked for the record desired.

7.2.3.6 If the record is not the desired record, the user must correct by going forward or reverse until the correct one is found.

7.3 Example -

7.3.1 Set Up within user program - alter High Speed Search function.

```
HSF: EQU: P03, 170.
HSR: EQU: P03, 174.
     ORG: P03, 180.
     SMS; S#1.          Set Section
     SBU; UDC.          User Update Counter
```

7.3.2 User Update Counter Procedure

```
     ORG: P10, 000.
UDC: LDA, R#Ø; R#4.          Test Status
     TMX, 000; OCT:010.      Runaway?
     SUX, R#7; DEC:001.      Decr. Search Counter
     LDA, R#Ø; DEC:000.      Test for End of Search
     CPA, R#Ø; R#7.          No - Set Low Cond.
     EXU; 000.               Yes - Set Equal
```

7.3.3 User Search Function

```
SRH: IOC, C#2; 016.          Select Deck-2
     LDX, R#7; DEC:050.      Set Search Counter
     SMS; S#Ø.
     SBU; HSF.               High Speed Search Forward
     LDA, R#Ø; R#4.          Search Completed - Test Status
     TMJ, +XX; OCT:010.      Runaway?
     LDA, R#Ø; OCT:202.      Deck Select-No Sequence Ck.
     SMS; S#Ø.
     SBU; RDS.               Read Deck-2
      -------
```

8.0 LOAD PROGRAM TAPE OPTION

8.1 Description -

The Load Program Tape Option simulates the hardware bootstrap load procedure. This function is performed by reading the bootstrap record, resetting the stack pointer, moving the bootstrap to the assigned area, and finally execution of the bootstrap.

8.2 Function Usage -

8.2.1 Load a program under operator control (Deck-2 only).

8.2.1.1 Position the tape on Deck 2 so that the next record to be read is the bootstrap.

8.2.1.2 Branch to the operator controlled Load Program Tape Option.

8.2.1.3 The operator may now depress the 'START' Key to load the program from Deck-2.

8.2.2 Load a program under program control.

8.2.2.1 Position the tape (on either Deck) so that the next record to be read is the bootstrap.

8.2.2.2 Load the Accumulator with the desired Deck Select Word (see Read/Write Mini Tape Handlers).

8.2.2.3 Branch to the program controlled Load Program Tape Option.

8.3 Special Requirements -

8.3.1 The Load Program Tape Option resides in Memory Locations P03-314 thru P03-357 (Octal).

8.4 Entry Point Linkages -

8.4.1 Operator Controlled - LDP: EQU: P03, 204.

8.4.2 Program Controlled - LDP: EQU: P03, 217.

PROCEDURE TITLE

STANDARD MAXI TAPE FUNCTIONS

PROCEDURE TYPE: Instructional

Catalog No. 90043

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial
01	5, 6	R. Clifford	9/6/74	R.W.C.

Originator R. W. Clifford, Project Leader Date 03/26/74

Approval W. V. Hamilton Date 3/26/74

1. INTRODUCTION

There are three software functions which are used to handle the reading writing, and control functions on the three standard Compatible Maxi Tape Units. These functions enable the user to write and read variable size records up to 256 characters (1 Page).

2. GENERAL

The following is the standard Compatible Maxi Tape Units and software which are supported for the standard functions.

2.1 Standard Maxi Tape Function for 7 Track Medium Density

2.1.1 Segment ID - 'M7X'

2.1.2 Software Catalog Number - 60155

2.1.3 Standard 7 Track 556 BPI CTU.

2.2 Standard Maxi Tape Function for 9 Track Medium Density

2.2.1 Segment ID - 'M1X'

2.2.2 Software Catalog Number - 60154

2.2.3 Standard 9 Track 800 BPI CTU.

2.3 Standard Maxi Tape Function for 9 Track High Density

2.3.1 Segment ID - 'M6X'

2.3.2 Software Catalog Number - 60156

2.3.3 Standard 9 Track 1600 BPI CTU.

3. BACKGROUND CONSTRUCTION

These functions are located on the Standard I/O Library and can be called from the Library to build a special background containing one of these functions.

3.1 "M7X" is the call segment for the 7 Track Medium Density function. It will be placed on the background starting at decimal location P15-000 to P15-235.

3.2 "M1X" is the call segment for the 9 Track Medium Density function. It will be placed on the background starting at decimal location P15-000 to P15-235.

3.3 "M6X" is the call segment for the 9 Track High Density function. It will be placed on the background starting at decimal location P15-000 to P15-237.

4. OPERATION

The location and size parameters are preset to P01-000 and 128 bytes. These may be changed by altering the following locations:

4.1 For 7 and 9 Track Medium Density

4.1.1 Size Parameter: P15-227.

Enter Size of Record, 256 bytes (1 Page) is maximum.

4.1.2 Starting Location: P15-229.

Enter the Starting Location within the Data Page.

4.1.3 Data Page: P15-015 and P15-051.

Enter the Page Allocation in increment form. This is where the data will be transferred.

4.2 For 9 Track High Density

4.2.1 Size Parameter: P15-233.

Enter Size of Record, 256 Bytes (1 Page) is maximum.

4.2.2 Starting Location: P15-235.

Enter the Starting Location within the Data Page.

4.2.3 Data Page: P15-015, P15-069, P15-075, P15-081, P15-087.

Enter the Page Allocation in increment form. This is where the data will be transferred.

4.3 Function Registers -

These registers must be set up each time the function is entered.

4.3.1 R#4: Unit's Address.

The standard address of a CTU is 01. This is entered in R#4 as 001.

NOTE: The Unit's Address is hardware wired in the CTU and if changed, the Address loaded in R#4 must be the same.

4.3.2 R#5: Command Register.

Load the command into R#5 before entering the command function.

4.3.2.1 R#5 = 006, Rewind.

4.3.2.2 R#5 = 007, Rewind with Interlock

4.3.3 The Time Delay function used in the Read and Write functions is D93 and is part of the Mini Tape Read/Write function.

4.4 Function Entry

A Stack and Branch instruction must be used to enter these functions. All Locations are in Decimal notation.

4.4.1 Read Function:

```
RTP: LDX, R#4; OCT:001.    Load Address
     SMS; S#1.             *
     SBU; D30.             Go to Read
     BRH; RER.             Check for Error
     BRL; FMK.             Check for FM
     EXU; 000.             Return
```

4.4.2 Write Function:

```
WTP: LDX, R#4; OCT: 001.   Load Address
     SMS; S#1.             *
     SBU; D31.             Go to Write
     BRH; WER.             Check for Error
     BRL; EOT.             Check for End of Tape
     EXU; 000.             Return
```

4.4.3 Write File Mark:

```
WFM: LDX, R#4; OCT: 001.   Load Address
     SMS; S#1.             *
     SBU; D35.             Go to Function
     BRH; WER.             Check for Errors
     EXU; 000.             Return
```

4.4.3.1 File Mark Characters:

- 7 Track 556 BPI - Octal: 017.
- 9 Track 800 BPI - Octal: 023.
- 9 Track 1600 BPI - Special Record

4.4.4 Tape Backspace:

```
TBS: LDX, R#4; OCT: 001.   Load Address
     SMS; S#1.             *
     SBU; D37.             Go to Function
     EXU; 000.             Return
```

4.4.5 Rewind Tape:

```
RWD: LDX, R#4; OCT: 001.    Load Address
     LDX, R#5; OCT:006.     Load Command
     SMS; S#1.              *
     SBU; D36.              Go to Function
     EXU; 000.              Return
```

4.4.6 Status Request:

```
SRT: LDX, R#4; OCT: 001.    Load
     LDX, R#5; OCT:010.     Status Command
     SMS; S#1.              *
     SBU; D36.              Go to Function
     EXU; 000.              Return
```

NOTE: The status will be in the Accumulator. It is used to check Unit Ready before reading or writing.

4.5 Entry Point Linkages -

```
D30: EQU: P15, 000.     Read Function
D31: EQU: P15, 038.     Write Function
D35: EQU: P15, 150.     Write File Mark
D36: EQU: P15, 174.     Command
D37: EQU: P15, 182.     Tape Backspace
```

5. RETURN STATUS

After each Read or Write operation, the status will be stored in R#4 and the condition register set as follows:

5.1 (=) - Condition Register set to equal indicates the function executed without any errors.

5.2 (<) - Condition Register set low indicates a File Mark has been read, or the physical End of Tape has been reached during a read or a write operation. The user must determine this by checking the Return Status.

5.3 (>) - Condition Register set high indicates one of the following errors occurred during the operation:

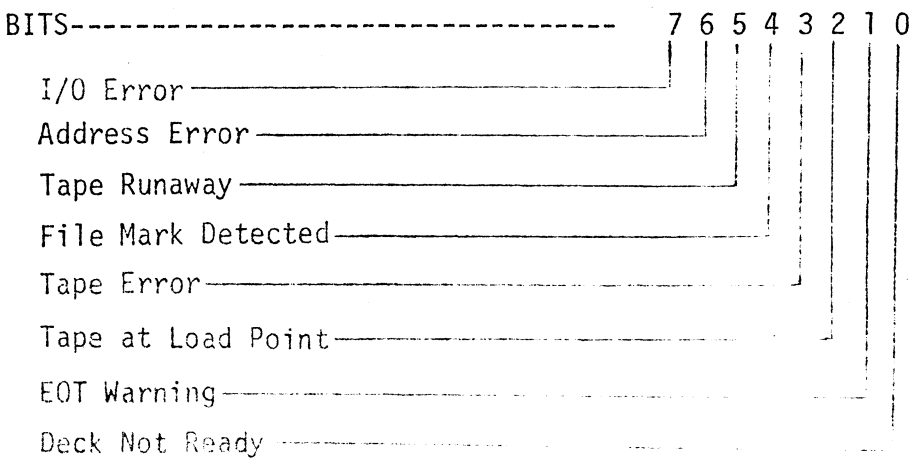
5.3.1 8 Retry Error: This function was not successful after 8 retries. Bit 3 present in the Status Word (R#4) indicates this.

5.3.2 Tape Runaway: No record detected on tape. Bit 5 present in the Status Word (R#4) indicates this.

5.3.3 Address Error: Unit did not respond to its address. Bit 6 present in the Status Word (R#4) indicates this.

5.3.4 I/O Error: Hardware error. Bit 7 present in Status Word (R#4) indicates this.

5.4 Status Word stored in R#4 -



6. ERROR RECOVERY

The following subroutines may be used to recover from a high condition after a function has been executed.

6.1 Write 8 Retry Error -

This routine will back the tape up, erase the bad block, and move the block forward 3 inches over the bad area of tape. This routine should not be repeated more than three times in succession, because it may cause a tape runaway when reading.

```
ERC: SBU; D37.    Back Up Tape
     SBU; D35.    Write FM (Erase 3 Inches)
     SBU; D37.    Back Up Tape
     BRU; WTP.    Re-Write Block
```

6.2 Read 8 Retry Error -

This routine will back the tape up over the bad block and repeat the Read.

```
     SBU; D37.    Back Up Tape
     BRU; RTP.    Re-read Block
```

6.3 If an I/O error or an Address error occurs, the problem should be corrected before repeating function.

6.3.1 Address error may be caused by improper Address or improper connection to SIO interface.

6.3.2 I/O Error is a hardware error and may not be recoverable.

7. VARIABLE BLOCK READ

When using the Read function to read variable size blocks, the Size Parameter must be set to at least 4 characters more than the largest block to be read. (Maximum size - 256 Bytes).

7.1 7 Track, 556 BPI and 9 Track, 1600 BPI Variable Read -

The last character to be stored in the buffer after a read will be the final tape status. This character may have to be deleted if not used.

7.2 9 Track, 800 BPI Variable Read -

The last three characters stored in the buffer after a read will be the CRC Byte, CRC Parity and the final tape Status Byte. These characters may have to be deleted if not used.

8. When using the Read function to read Fixed Size Blocks, the Size Parameters may be set to the actual data size. This method will read and store only the data portion of the record.

9. REGISTER USAGE

R#1 = Delay count for standard delay routine.

R#2 = Indexing.

R#3 = Buffer Pointer.

R#4 = Status and Address Register.

R#5 = Retry Counter and Command Register.

ROCEDURE TITLE

EXTENDED MAXI TAPE FUNCTIONS

Catalog No. 90028

OCEDURE TYPE: Instructional

Date 3/1/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

1. INTRODUCTION

There are two software functions which are used to handle the reading, writing, and control functions on the three standard Compatible Maxi Tape Units. These functions enable the user to write and read variable size records up to 1000 characters.

2. GENERAL

The following is the standard Compatible Maxi Tape Units and software which are supported.

2.1 Extended Maxi Tape Function for 7 and 9 Track Medium Density

2.1.1 Segment I.D. is M1E.

2.1.2 Software Catalog Number is 60150.

2.1.3 Standard 9 Track 800 BPI CTU.

2.1.4 Standard 7 Track 556 BPI CTU.

2.2 Extended Maxi Tape Function for 9 Track High Density

2.2.1 Segment I.D. is M6E.

2.2.2 Software Catalog Number is 60151.

2.2.3 Standard 9 Track 1600 BPI CTU.

3. BACKGROUND CONSTRUCTION

These functions are located on the Standard I/O Library and can be called from the Library to build a special background containing one of these functions.

3.1 "M1E" is the call segment for the 7 and 9 Track Medium Density function. It will be placed on the background starting at decimal location P14-216 to P15-255.

3.2 "M6E" is the call segment for the 9 Track High Density function. It will be placed on the background starting at decimal location P14-216 to P15-255.

4. OPERATION

Function parameters must be set up before entering the functions.

4.1 Size Parameters -

These parameters are used by the Read and Write functions and are not altered.

NOTE 1: The Maxi Tape function uses the direct addressing capability, therefore, the PØØ parameters must be within the current control section.

NOTE 2: This function supports records up to 1000 characters long. The Write Retry will not support records any longer.

4.1.1 POO-009: Starting Page.

This is the Page where the transfer will begin. Enter the Page Allocation in increment form.

4.1.2 POO-010: Starting Location.

This is the Location in the starting Page where the transfer will begin.

4.1.3 POO-011: Ending Page.

This is the Page where the transfer will stop. Enter the Page Allocation in increment form.

4.1.4 POO-012: Ending Location.

This is the Location in the Ending Page where the transfer will stop.

4.2 Function Registers -

These registers must be set up each time the function is entered.

4.2.1 R#4: Unit's Address.

The standard address of a CTU is 01. This is entered in R#4 as 001 or 201.

NOTE: The Unit's Address is hardware wired in the CTU and if changed, the address loaded in R#4 must be the same.

4.2.2 R#5: Command Register.

Load the command into R#5 before entering the command function.

4.2.2.1 R#5 = 006, Rewind.

4.2.2.2 R#5 = 007, Rewind with INterlock.

4.2.2.3 R#5 = 023, Write File Mark on a 9 Track, 800 BPI CTU.

4.2.2.4 R#5 = 017, Write File Mark on a 7 Track 556 BPI CTU.

4.2.2.5 R#5 is not required to write a File Mark on a 9 Track 1600 BPI CTU.

4.2.3 The Time Delay function used in the Read and Write functions is D92 and is part of the Mini Tape Read/Write function.

4.3 Function Entry

A Stack and Branch instruction must be used to enter these functions.

4.3.1 Read Function:

```
RTP: LDX, R#4; OCT:201.        Load Address
     SNS; S#1.                 *
     SBU; D30.                 Go to Read
     BRH; RER.                 Check for Error
     BRL; FMK.                 Check for FM
     EXT; 000                  Return
```

4.3.2 Write Function:

```
WTP: LDX, R#4; OCT:201.      Load Address
     SMS; S#1.               *
     SBU; D31.               Go to Write
     BRH; WER.               Check for Error
     BRL; EOT.               Check for End.of Tape
     EXU; 000.               Return
```

4.3.3 Write File Mark (9 Track, 1600 BPI)

```
WFM: LDX, R#4; OCT:201.      Load Address
     SMS; S#1.               *
     SBU; D35.               Go to Function
     BRH; WER.               Check for Errors
     EXU; 000.               Return
```

4.3.4 Write File Mark (7 Track, 556 BPI)

```
WF1: LDX, R#4; OCT:201.      Load Address
     LDX, R#5; OCT:017.      File Mark Character
     SMS; S#1.               *
     SBU; D35.               Enter Function
     BRH; WER.               Check for Error
     EXU; 000.               Return
```

4.3.5 Write File Mark (9 Track, 800 BPI)

```
WF2: LDX, R#4; OCT:201.      Load Address
     LDX, R#5; OCT:023.      File Mark Character
     SMS; S#1.               *
     SBU; D35.               Enter Function
     BRH; WER.               Check for Error
     EXU; 000.               Return
```

4.3.6 Tape Backspace:

```
TBS: LDX, R#4; OCT:201.      Load Address
     SMS; S#1.               *
     SBU; D37.               Go to Function
     EXU; 000.               Return
```

4.3.7 Rewind Tape:

```
RWD: LDX, R#4; OCT:201.      Load Address
     LDX, R#5; OCT:006.      Load Command
     SMS; S#1.               *
     SBU; D36.               Go to Function
     EXU; 000.               Return
```

NOTE: Rewind with Interlock Command is 007.
Rewind Normal Command is 006.

4.3.8 Status Request:

```
SRT: LDX, R#4; OCT:201.      Load
     LDX, R#5; OCT:010.      Status Command
     SMS; S#1.               *
     SBU; D36.               Go to Function
     EXU; 000.               Return
```

NOTE: The status will be in the Accumulator. It is used to check Unit Ready before reading or writing. (See paragraph 5.4)

4.4 Entry Point Linkages

```
D30: EQU: P15, 000.      Read Function
D31: EQU: P15, 060.      Write Function
D35: EQU: P15, 176.      Write File Mark
D36: EQU: P15, 196.      Rewind and Status
D37: EQU: P15, 202.      Tape Backspace
```

5. RETURN STATUS

After each Read or Write operation, the status will be stored in R#4 and the condition register set as follows:

5.1 (=) - Condition Register set to equal indicates the function executed without any errors.

5.2 (<) - Condition Register set low indicates on a Read operation a file mark has been read, or on a Write, the End of Tape has been reached.

5.3 (>) - Condition Register set high indicates one of the following errors occurred during the operation:

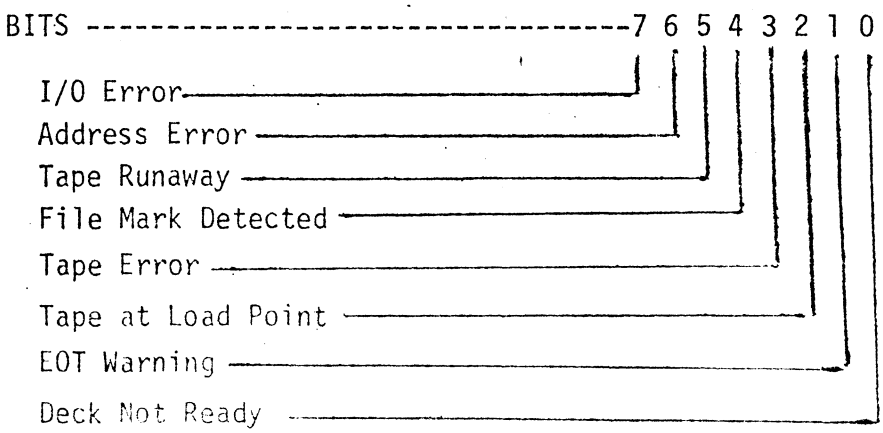
5.3.1 8 Retry Error: This function was not successful after 8 retries. Bit 3 present in the Status Word (R#4) indicates this.

5.3.2 Tape Runaway: No record detected on tape. Bit 5 present in the Status Word (R#4) indicates this.

5.3.3 Address Error: Unit did not respond to its address. Bit 6 present in the Status Word (R#4) indicates this.

5.3.4 I/O Error: Hardware Error. Bit 7 present in Status Word (R#4) indicates this.

5.4 Status Word Stored in R#4 -



6. ERROR RECOVERY

The following subroutines may be used to recover from a high condition after a function has been executed.

6.1 Write 8 Retry Error -

This routine will back the tape up, erase the bad block, and move the block forward 3 inches over the bad area of tape. This routine should not be repeated more than three times in succession, because it may cause a tape runaway when reading.

```
ERC: SBU; D37.      Back Up Tape
     SBU; D35.      Write FM (Erase 3 Inches)
     SBU; D37.      Back Up Tape
     BRU; WTP.      Re-write block
```

6.2 Read 8 Retry Error

This routine will back the tape up over the bad block and repeat the Read.

```
     SBU; D37.      Back up Tape
     BRU; RTP.      Re-read block
```

6.3 If an I/O Error or an Address Error occurs, the problem should be corrected before repeating function.

6.3.1 Address error may be caused by improper Address or improper connection to SIO interface.

6.3.2 I/O Error is a hardware error and may not be recoverable.

7. VARIABLE BLOCK READ

When using the Read function to read variable size blocks, the Size Parameter must be set to at least 4 characters more than the largest block to be read.

7.1 7 Track, 556 BPI and 9 Track, 1600 BPI Variable Read -

The last character to be stored in the buffer after a read will be the final tape status. This character may have to be deleted if not used.

7.2 9 Track, 800 BPI Variable Read -

The last three characters stored in the buffer after a read will be the CRC Byte, CRC Parity and the final tape Status Byte. These characters may have to be deleted if not used.

8. FIXED BLOCK READ

When using the Read function to read Fixed Size Blocks, the Size Parameters may be set to the actual data size. This method will read and store only the data portion of the record.

9. REGISTER USAGE

R#1 = Delay Count for Standard Delay Routine

R#2 = Indexing

R#3 = Buffer Pointer

R#4 = Status and Address Register

R#5 = Retry Counter and Command Register

LINE PRINT FUNCTION (1551)

PROCEDURE TYPE: Instructional

Catalog No. 90040

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial
01	#6	R. Clifford	6/5/74	RWC
02	#6	R. Clifford	6/24/74	RWC
03	#5	R. Clifford	9/4/74	RWC

Originator R. H. Clifford, Robert Clifford Date 03/26/74

Approval [illegible] Date 3/26/74

1.0 INTRODUCTION

This function contains the basic routines to allow the 1551S or 1551D Printers to provide a hard copy output. The Model 1551S Line Printer develops a 5 x 7 Dot matrix at the rate of 165 characters a second, with the minimum capability of printing lines of 132 characters maximum length at a speed of 60 lines per minute. The Model 1551D Line Printer develops a 5 x 7 Dot Matrix at the rate of 330 characters a second, with the minimum capability of printing lines of 132 characters maximum length at a speed of 125 lines per minute. This function provides for the printing of the standard 64 character set as well as special printer control commands.

2.0 SPECIFICATION

2.1 Segment ID - 'LNP'

2.2 Source Catalog Number - 60032

2.3 Object Catalog Number - None

2.4 Special Requirements -

2.4.1 Memory Requirements - P16-000 thru P16-121 (Octal)

2.4.2 Registers Used - 1, 2, 3, 4

3.0 OPERATING INSTRUCTIONS

3.1 Print Buffer -

The Print Buffer is initiated as Page 01. If other than Page 01 is desired as the Print Buffer, the user must alter P14-007 (decimal) to contain the desired Print Buffer Page in increment form.

3.2 Print Line -

3.2.1 Move the data to be printed to the assigned print buffer. This data must be in the standard 1501 code as it will be translated before it is actually printed.

3.2.2 Load Index Register 2 with the number of characters in the buffer to be printed.

3.2.3 Load Index Register 3 with the starting Location within the assigned print buffer.

3.2.4 Stack and Branch to the Print function to perform the print operation. The form will automatically be advanced one line after the print. If the print line exceeds 132 characters an automatic Line Feed will occur after each completed line.

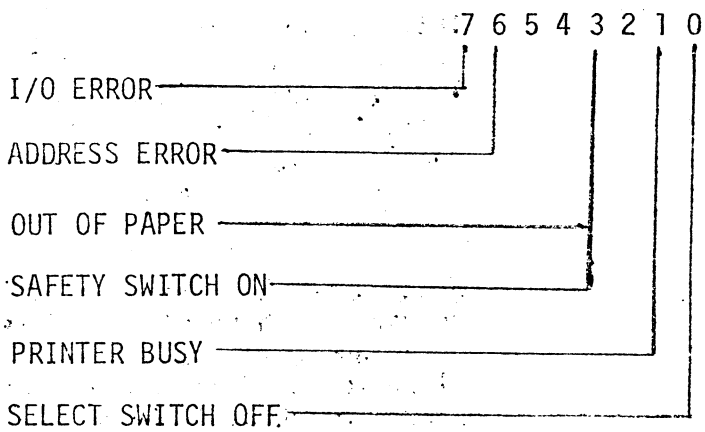
3.3 Print Status -

3.3.1 Upon returning from the Print function, the condition register will be set as follows:

3.3.1.1 Condition Register set EQUAL indicates the function has been properly completed.

3.3.1.2 Condition Register set HIGH indicates an error condition.

3.3.2 The print status word will be in the accumulator. The description of this status word is shown below.



3.4 Printer Translate Table

The Printer Translate Tables are used to translate the standard keyboard code into USASCII code.

3.4.1 The Printer Translate Table 'CT1' (1501 to USASCII) must also be called from the I/O Library for operation of the print function.

3.4.2 There is also included within the standard Keyboard Translate Table the Translate Table for the special printer control options. This Table must be present when utilizing these special options.

3.5 Printer Control Options

3.5.1 Bell Alarm - A three-second audible alarm will be sounded by the printer.

3.5.2 Feed 1 Line of Paper - Vertically advance the paper 1 line.

3.5.3 Vertical Tab - Slew the form to the next Channel 5 punched in the printer carriage control tape.

3.5.4 Form Feed - Slew the form to the next Channel 7 punched in the printer carriage control tape. The standard function of this feature is to vertically advance the form to the next top of form (line 1).

3.5.5 Elongate - This feature prints each of the 5 horizontal matrix character values twice producing a 10 x 7 character instead of the normal 5 x 7 character. When using this feature, the complete line will be elongated. The maximum number of characters that may be elongated is 66.

When using the printer Model 1551D the characters to be elongated will have a starting address of 000 for head 1 which will elongate the first 33 characters and 066 for head 2 which will elongate 33 more characters starting at that address.

3.5.6 De-Select - This feature turns the printer off-line which prevents the printer from receiving data.

3.5.7 Select - This feature turns the printer on-line which makes it available to receive data.

3.5.8 Clear - This feature clears the interface status. This command is used with the Select command to reset the De-Select condition.

3.5.9 Delete - Clears the printer hardware buffer and initializes the printer electronics. This may be used to reset the printer electronics after an error has occurred.

3.6 Printer Control Operation -

3.6.1 The Printer Control options may be executed independent of the Print function by loading the assigned values into Register Number 4 and performing a Stack and Branch to the Form Control Function. The assigned values are as follows.

OCTAL 007 - Bell Alarm

OCTAL 012 - Feed 1 Line of Paper

OCTAL 013 - Vertical Tab

OCTAL 014 - Form Feed

OCTAL 016 - Elongate Character

OCTAL 023 - De-Select (Printer Off-Line)

OCTAL 021 - Select

OCTAL 050 - Clear

OCTAL 177 - Delete

3.7 Examples

3.7.1 Print a Line -

```
LDX, R#2; DEC:132.          Load Size
LDX, R#3; DEC:000.          Load Start Location
SBU; PF1.                   Print Function
BRH; PER.                   Print error if high
```

3.7.2 Feed 1 Line of Paper -

```
LDX, R#4; OCT:012.          Load Command
SBU; D99.                   Form Control Function
```

3.7.3 Turn Select On -

```
LDX, R#4; OCT:050.          Clear Printer
SBU; D99.
LDA, R#0; OCT:021.          Send Select Command
IOC, C#5; 002.
EXU; 000.
```

REQUEST STATUS

4. ENTRY POINT LINKAGE

4.1 PF1: EQU: P14, 000. Print Function

4.2 D99: EQU: P14, 062. Form Control Function

PROCEDURE TITLE

LINE PRINT FUNCTION (1555/1556)

Catalog No. 90046

PROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

1. INTRODUCTION

This function contains the basic routines to allow the Model 1555 or Model 1556 to provide hard copy output. The Models 1555 and 1556 operate at a speed of 300 and 400 lines per minute respectively. The standard unit is capable of printing 1501 Codes by translating to the USASCII 64 character upper case code set. The printer functionally responds to special printer control commands as: Form Feed, Line Feed, Vertical Tab, and Slew to Channel.

2. SPECIFICATION

2.1 Segment ID - 'LNP'

2.2 Source Catalog Number - 60032

2.3 Object Catalog Number - None

2.4 Special Requirements -

2.4.1 Memory Requirements - P16-000 thru P16-121 (Octal)

2.4.2 Registers Used - 1, 2, 3, 4

3. OPERATING INSTRUCTIONS

3.1 Print Buffer -

The Print Buffer is initiated as Page 01. If other than Page 01 is desired as the Print Buffer, the user must alter P14-007 (decimal) to contain the desired Print Buffer Page in increment form.

3.2 Print Line -

3.2.1 Move the data to be printed to the assigned print buffer. This data must be in the standard 1501 Code as it will be translated before it is actually printed.

3.2.2 Load Index Register 2 with the number of characters in the buffer to be printed.

3.2.3 Load Index Register 3 with the starting Location within the assigned print buffer.

3.2.4 Stack and Branch to the Print function to perform the print operation. The form will automatically be advanced one line after the print. If the print line exceeds 132 characters an automatic Line Feed will occur after each completed line.

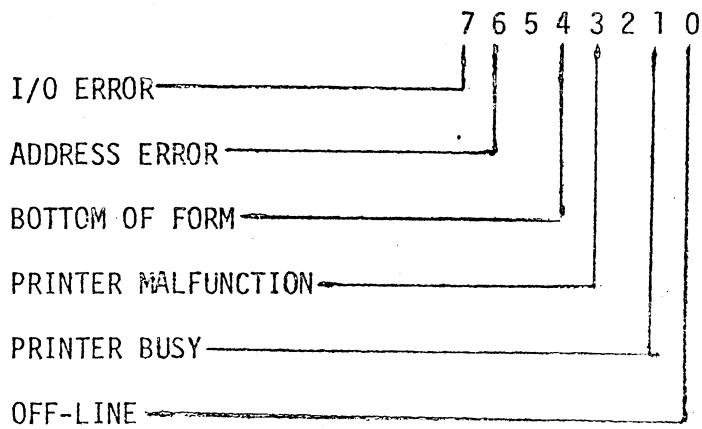
3.3 Print Status -

3.3.1 Upon returning from the Print function, the condition register will be set as follows:

3.3.1.1 Condition Register set EQUAL indicates the function has been properly completed.

3.3.1.2 Condition Register set HIGH indicates an error condition or Bottom of Form.

3.3.2 The print status word will be in the accumulator. The description of this status word is shown below.



3.4 Printer Translate Table

The Printer Translate Tables are used to translate the standard keyboard code into USASCII code.

3.4.1 The Printer Translate Table 'CT1' (1501 to USASCII) must also be called from the I/O Library for operation of the print function.

3.4.2 There is also included within the standard Keyboard Translate Table the Translate Table for the special printer control options. This Table must be present when utilizing these special options.

3.5 Printer Control Options

3.5.1 Line Feed - Vertically advance the paper one line.

3.5.2 Form Feed - Slew the form to the next Channel 1 punched in the printer carriage control tape.

3.5.3 Bottom of Form - After execution of this command, the Status Word will indicate a Bottom of Form condition when a VFU Channel 3 is detected. This option may be reset by executing a Clear Command.

3.5.4 Shift In Code -

3.5.4.1 The Shift In Code allows loading of the Vertical Tab Counter with any two-digit decimal number.

3.5.4.2 The counter is decremented for each line of paper advanced until the Counter reaches zero. This includes lines printed or those implicit in a slew. However, the lines between the bottom of form and the top of form are interpreted as a single line advance.

3.5.4.3 When the Vertical Tab Command is received and the Vertical Tab Count is not zero, the paper advance is started. The paper stays in continuous motion until the Counter reaches zero.

3.5.4.4 The most significant digit of the Vertical Tab counter must be shifted in first. Two digits must be shifted in.

3.5.5 Vertical Tab - When the Vertical Tab code is received by the printer, the form will be advanced the number of lines equal to the contents of the Tab Counter. No action occurs if the Tab Counter is zero.

3.5.6 Slew to Channel - By executing the Slew to Channel procedure the paper will slew until it detects the selected VFU Channel. After executing the Slew to Channel Command, the printer is conditioned to interpret the next byte received as the Channel Number.

CHANNEL 1 - Top of Form

CHANNEL 2 - Not Available

CHANNEL 3 - Bottom of Form

CHANNELS 4-12 - User Option

3.5.7 Clear - This Command is used to reset the Bottom of Form option and clears printer error.

3.6 Printer Control Operation

3.6.1 The following control bytes may be inserted into the data buffer and is included in the print size. The printer commands will be executed prior to printing the data.

OCTAL 235 - Line Feed

OCTAL 254 - Form Feed

3.6.2 The printer control options may be executed independent of the print function by loading the assigned values into Register 4 and performing a Stack and Branch to the Form Control function. The assigned values are as follows:

OCTAL 012 - Line Feed

OCTAL 014 - Form Feed

OCTAL 020 - Bottom of Form

OCTAL 017 - Shift In

OCTAL 013 - Vertical Tab

OCTAL 032 - Enable VFU

OCTAL 050 - Clear Interface & Bottom of Form Enable

3.6.3 If the Bottom of Form has been enabled and a HIGH condition results from a print operation, the user must test for an error or Bottom of Form Status. If Bottom of Form, the user must execute the Clear procedure prior to performing any other print functions. (The Bottom of Form must be re-enabled for subsequent forms).

3.7 Examples

3.7.1 Print a Line

```
LDX, R#2; DEC:132.      Load Size
LDX, R#3; DEC:000.      Load Start Location
SBU; PF1.               Print Function
BRH; ERR.               Print Error (or Bottom of Form if enabled)
```

3.7.2 Line Feed

```
LDX, R#4; OCT:012.      Load Command
SBU; D99.               Form Control Function
```

3.7.3 Shift In and Vertical Tab

```
LDX, R#4; OCT:017.      Shift In Command
SBU; D99.
LDX, R#4; DEC:001.      Set to Vertical
SBU; D99.               Tab 10 Lines
LDX, R#4; DEC:000.
SBU; D99.
LDX, R#4; OCT:013.      Vertical Tab
SBU; D99.
```

3.7.4 Slew to Channel

```
LDX, R#4; OCT:032.      Set Slew Channel
SBU; D99.
LDX, R#4; DEC:012.      Slew to Channel 12
SBU; D99.
```

4. ENTRY POINT LINKAGES

```
4.1  PF1: EQU: P14, 000.      Print Function

4.2  D99: EQU: P14, 062.      Form Control Function
```

PROCEDURE TITLE

MATRIX PRINT FUNCTION

Catalog No. 90041

PROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

1.0 INTRODUCTION

This function contains the basic routines to allow printing on the Model 1525 or the Model 1526 Matrix printer. Each character to be printed will be sent to the printer in a 5 x 7 bit matrix (5 bytes) which will be read from the CRT Dot Pattern Area for the associated character. A translate table is not required with this function.

2.0 SPECIFICATION

2.1 Segment ID - 'MXP'

2.2 Source Catalog Number - 60027

2.3 Object Catalog Number - None

2.4 Special Requirements

2.4.1 This function utilizes the time delays from the standard Mini Tape Read Function.

2.4.2 Memory Requirements - P16-000 thru P16-147 (Octal)

2.4.3 Registers Used - 1, 2, 3, 4.

3.0 OPERATING INSTRUCTIONS

3.1 Print Buffer

The print buffer is initiated as Page 01. If other than Page 01 is desired as the print buffer, the user must alter P14-007 (decimal) to contain the desired print buffer Page in increment form.

3.2 Print Modes

3.2.1 Line Print -

This mode will print either a maximum of 80 characters (Model 1526) or 132 characters (Model 1525) per line. The carriage will return after the print function has been completed. In addition, if the number of characters printed exceeds the print line capacity, an automatic carriage return will be initiated following each completed line of print.

A maximum of 255 characters can be printed by one print operation. The print speed is 33-1/3 characters per second.

3.2.2 Increment Mode -

This mode allows the printing of one character at a time. There will not be a carriage return at the end of the print function. A carriage return will result only when the print line capacity has been exceeded or when the command is given. Increment mode must be set before printing. The speed is 16-2/3 characters per second. System Reset or the Clear Command will reset the Increment Mode.

Increment Mode must be reset before entering Line Mode.

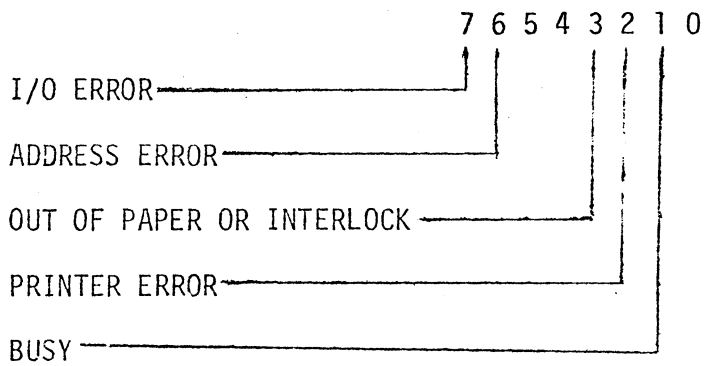
3.3 Print Status

3.3.1 Upon returning from the print function, the Condition Register will be set as follows:

3.3.1.1 Condition Register set EQUAL indicates the function has been properly completed.

3.3.1.2 Condition Register set HIGH indicates an error has occurred.

3.3.2 The print status word will be in the accumulator. The description of this status word is shown below.



NOTE: Print Error, Address Error or I/O Error may be cleared by turning the Printer power OFF then ON and reprinting that line.

3.4 Printer Control Options

3.4.1 Left Margin -

This function performs a carriage return but does not execute a Line Feed.

3.4.2 Carriage Return -

This function performs a carriage return and a Line Feed.

3.4.3 Line Feed -

Vertically advance the form one line without a carriage return.

3.4.4 Incremental Mode -

Set print mode for single character print.

3.4.5 Clear -

Set print mode to line print and reset Printer Error.

3.5 Printer Control Operation

The Printer Control options are executed independent of the Print function by loading the assigned values into Register Number 4 and performing a Stack and Branch to the Form Control Functions. The assigned values are as follows:

OCTAL 002 - Left Margin

OCTAL 004 - Carriage Return

OCTAL 012 - Line Feed

OCTAL 014 - Incremental Mode

OCTAL 050 - Clear

The Print Status is in the Accumulator (refer to Paragraph 3.3.2).

3.6 Examples

3.6.1 Print a Line

```
LDX, R#2; DEC:080.      Load Size
LDX, R#3; DEC:000.      Load Start Loc.
SMS; S#1.
SBU; PF1.               Go to Print
BRH; PER.               Error Routine
EXU; 000.               Return
```

3.6.2 Character Print Mode

3.6.2.1 Set Increment Mode

```
LDX, R#4; OCT:014.      Set Command
SMS; S#1.
SBU; D99.               Form Control Function
EXU; 000.               Return
```

3.6.2.2 Print One Character and Increment Location Pointer

```
LDX, R#2; DEC:000.      Must be 000
LDX, R#3; DEC:000.      Starting Location
SMS; S#1.
SBU; PF1.               Print one character
BRH; PER.               Error Routine
EXU; 000.               Return
```

3.6.3 Command Function

```
LDX, R#4; OCT:004.      Set Command
SMS; S#1.
SBU; D99.               Form Control Function
EXU; 000.               Return
```

3.7 Entry Point Linkage

3.7.1

```
PF1: EQU: P14, 000.     Print Function
D99: EQU: P14, 062.     Form Control Function
```

PROCEDURE TITLE

TYPEWRITER PRINT FUNCTION

Catalog No. 90047

ROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

1. INTRODUCTION

This function contains the basic routines to obtain hard copy output from the IBM 735 Selectric Typewriter at a speed of 15 characters per second. Both upper and lower case characters may be printed in addition to special printer control options.

2. SPECIFICATION

2.1 Segment ID - 'TYP'

2.2 Source Catalog Number - 60033

2.3 Object Catalog Number - None

2.4 Special Requirements

2.4.1 Memory Requirements - P16-000 thru P16-127 (Octal)

2.4.2 Registers Used - Registers 1, 2, 3, 4

3. OPERATING INSTRUCTIONS

3.1 Print Buffer -

The Print Buffer is initiated as Page 01. If other than Page 01 is desired as the Print Buffer, the user must alter P14-007 (decimal) to contain the desired Print Buffer Page in increment form.

3.2 Print Line -

3.2.1 Move the data to be printed to the assigned print buffer. This data must be in the standard 1501 Code as it will be translated before it is actually printed.

3.2.2 Load Index Register 2 with the number of characters in the buffer to be printed.

3.2.3 Load Index Register 3 with the starting Location within the assigned print buffer.

3.2.4 Stack and Branch to the Print function to perform the print operation. The form will automatically be advanced one line after the print.

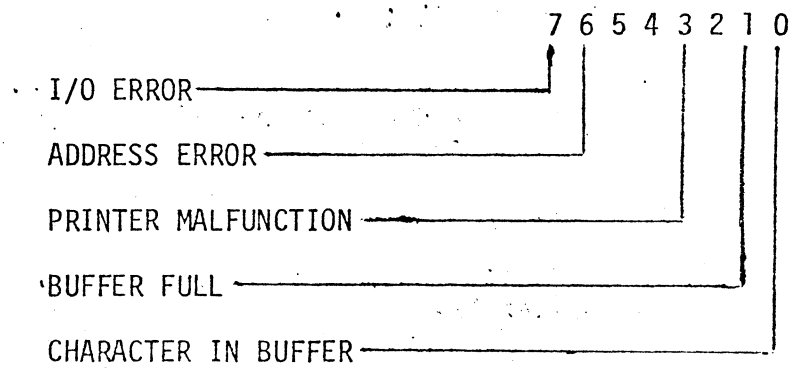
3.3 Print Status -

3.3.1 Upon returning from the Print function, the Condition Register will be set as follows:

3.3.1.1 Condition Register set EQUAL indicates the function has been properly completed.

3.3.1.2 Condition Register set HIGH indicates an error condition.

3.3.2 The print status word will be in the accumulator. The description of this status word is shown below.



3.4 Printer Translate Table -

The Printer Translate Tables are used to translate the standard keyboard code into the corresponding Typewriter code.

3.4.1 The Printer Translate Table 'TTI' (1501 to Typewriter) must also be called from the I/O Library for operation of the print function.

3.4.2 Print Characters

3.4.2.1 Standard 1501 Codes 'A' thru 'Z' with Bits 6 and 7 off will be printed as upper case characters.

3.4.2.2 Corresponding 1501 Codes 'A' thru 'Z' with Bit 6 on will print as lower case characters.

3.5 Printer Control Options -

3.5.1 Carriage Backspace - Backspace the character print position one location.

3.5.2 Red Ribbon Shift - (Optional)

3.5.3 Carriage Return - This function performs a line feed and resets the print position to the left margin.

3.5.4 Black Ribbon Shift - (Optional)

3.5.5 Tab - Skip the print mechanism to the next preset typewriter tab.

3.5.6 Index - Vertically advance the form one line. This function does not reset the print position to the left margin.

3.6 Printer Control Operation -

3.6.1 The printer control options may be executed independent of the print function by loading the assigned values into Register 4 and performing a Stack and Branch to the Form Control function. The assigned values are as follows:

OCTAL 002 - Carriage Backspace

OCTAL 003 - Red Ribbon Shift

OCTAL 004 - Carriage Return

OCTAL 006 - Black Ribbon Shift

OCTAL 011 - Horizontal

OCTAL 012 - Index (Line Feed)

OCTAL 050 - Clear Interface

3.7 Examples -

3.7.1 Print a Line

```
LDX, R#2; DEC:130.    Load Size
LDX, R#3; DEC:000.    Load Start Location
SBU; PF1.             Print Function
BRH; ERR.             Print Error
```

3.7.2 Line Feed

```
LDX, R#4; OCT:012.    Load Command
SBU; D99.             Form Control Function
```

4. ENTRY POINT LINKAGES

```
4.1  PF1: EQU: P14, 000.      Print Function

4.2  D99: EQU: P14, 062.      Form Control Function
```

PROCEDURE TITLE

TYPEWRITER SINGLE CHARACTER PRINT FUNCTION (P1S)

Catalog No. 90045

PROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

Originator C. W. Clifford, Robert Fisher Date 03/26/74

Approval W. W. Tomaselli Date 3/26/74

1. INTRODUCTION

This function allows the IBM 735 Selectric typewriter to print one character and return to the user's program. When inserted into the background and loaded, this function will reside in decimal addresses

P14-000 to P14-093.

2. SPECIFICATION

2.1 Segment ID - 'P1S'

2.2 Source Catalog Number - 60034

2.3 Object Catalog Number - None

2.4 Special Requirements

2.4.1 Register Usage - 1, 2, 3, 4

3. OPERATIONAL REQUIREMENTS (DATA)

3.1 Load Index Register #4 with the code for the character to be printed.

3.2 Store the location of the I/O address to P14-013.

3.3 This routine uses the bit serial I/O and requires unit number designation. The routine is initialized to address unit 5, and the user must modify the 6th instruction in the routine (currently LDA, R#Ø; OCT:205) to change the addressed unit. The 735 Selectric will lock its keyboard when addressed. If it is necessary for the user to access the keyboard, an undesignated unit should be addressed to unlock the keyboard.

3.4 Link to: PCR: EQU: P14, 000 (P16-000 Octal)

4. OPERATIONAL REQUIREMENTS (PRINTER CONTROL)

4.1 Load Index Register #4 with the printer line control in Octal, as follows:

204 - Carriage Return

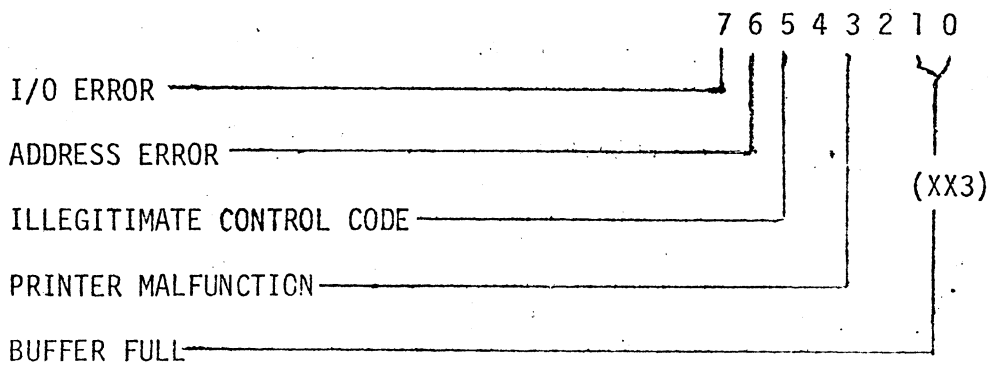
212 - Index

211 - Tab

202 - Backspace

5. RETURN STATUS

The condition register set to EQUAL indicates the function has been properly completed. The condition register set to LOW indicates a Busy condition (buffer full). If the condition register is set to HIGH after a print function (error condition), the Status Word will be in the accumulator. Figure 1 shows the configuration of the Status Word for the PIS function.



(with one of the above conditions)

FIGURE 1. Status Word - PIS Function.

6. PRINT ERROR RESET

The printer error reset implies that one or more characters or control functions have not been completed. The user must call this to the operator's attention. As an alternate, the user may print error flags or re-print the item in process. The error reset is as follows:

6.1 The I/O Error or the Address Error will reset with the user's next Stack and Branch to PCR.

6.2 The Illegitimate Control Code must be replaced with a flag of the user's choice.

6.3 The Printer Malfunction Error requires an error reset. This operation requires the saving of the byte in Index Register #4, replacing it with Octal 250 for error reset; then restoring the byte to proceed with the print program.

7. REQUIRED RELATED FUNCTIONS

The Typewriter Translate Table TT1 must also be called from the Standard I/O Library and inserted into the background, to complete the requirements for this operation.

PROCEDURE TITLE

STANDARD PRINTER CONTROL OPTIONS

Catalog No. 90029

PROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial
01	4	R. Lasher	6/18/74	RAL

Originator Robert Lasher Date 03/26/74

Approval Wm Howard Jr. Date 3/26/74

1.0 INTRODUCTION

The print options are designed to provide the user with the basic print options as: Skip to Top of Form, Line Feed, Skip to Channel, Bottom of Form Test and Elongate. The Print functions and the Print options are designed to be interchangeable with other print functions and print options without altering the user program.

2.0 SPECIFICATION

2.1 1551 Print Options

2.1.1 Segment ID - 'C51'

2.1.2 Source Catalog Number - 60152

2.1.3 Memory Requirements - P16-120 thru P16-243 (Octal)
P16-276 thru P16-325 (Octal)

2.2 1555/1556 Print Options

2.2.1 Segment ID - 'C55'

2.2.2 Source Catalog Number - 60172

2.2.3 Memory Requirements - P16-120 thru P16-265 (Octal)
P16-276 thru P16-325 (Octal)

2.3 1525/1526 Print Options

2.3.1 Segment ID - 'C25'

2.3.2 Source Catalog Number - 60067

2.3.3 Memory Requirements - P16-146 thru P16-275 (Octal)
P16-276 thru P16-325 (Octal)

2.4 Typewriter Print Options

2.4.1 Segment ID - 'C35'

2.4.2 Source Catalog Number - 60173

2.4.3 Memory Requirements - P16-126 thru P16-255 (Octal)
P16-276 thru P16-325 (Octal)

3.0 OPERATING INSTRUCTIONS

3.1 Printer Control Table

The user must include within his program the following information in the assigned area.

3.1.1 Form Size - This is the overall size of the form.

3.1.2 Print Line - This is the number of lines that may actually be printed on one form. This count will be used to determine the Bottom of Form.

3.1.3 Current Line - This is the current line number and is controlled by the printer options to determine the number of lines to skip when a Skip to Channel is requested, and to determine the Bottom of Form. This must be initialized by the user to 001.

3.1.4 Channel Assignments -

There are 12 channels available to the user. By specifying the corresponding line/channel numbers, the user may advance the form to an assigned line without any specialized user programming.

3.1.5 Example -

```
     ORG: P14, 199.
SIZ: DEC: (066).          Form Size 66 Lines
PSZ: DEC: (060).          Print lines-Allow for margin
CUR: DEC: (001).          Current line - Initialize to 1
CH1: DEC: (001).          Channel-1 - Line 1
CH2: DEC: (015).          Channel-2 - Line 15
CH3: DEC: (000).          Channel-3 - Not Used
CH4: DEC: (000).          Channel-4 - Not Used
CH5: DEC: (000).          Channel-5 - Not Used
CH6: DEC: (040).          Channel-6 - Line 40
CH7: DEC: (000).          Channel-7 - Not Used
CH8: DEC: (000).          Channel-8 - Not Used
CH9: DEC: (000).          Channel-9 - Not Used
C10: DEC: (000).          Channel-10- Not Used
C11: DEC: (000).          Channel-11- Not Used
C12: DEC: (059).          Channel-12- Line 59
```

3.2 Print Option (Stack and Branch to the desired Function)

3.2.1 Skip to Top of Form - This feature will vertically advance the form to the next top of form Line-1. The current line will be reset to Line 1.

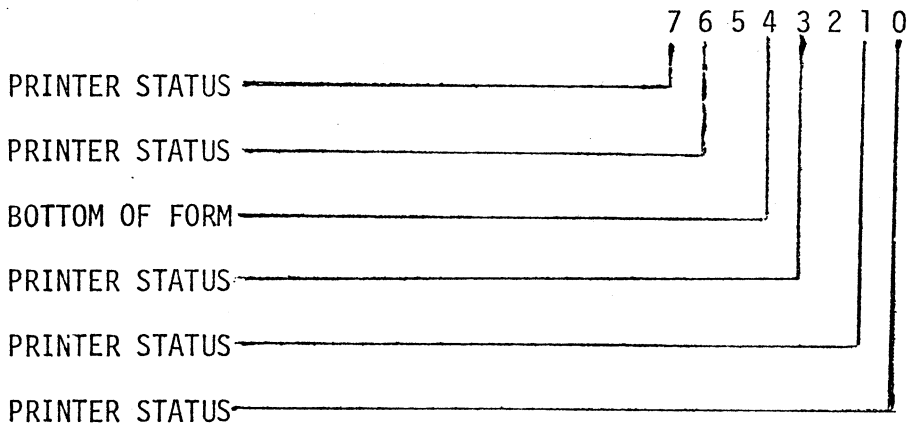
3.2.2 Line Feed - Vertically advance the form the number of lines specified in Register Number 2. The current line count will be incremented the number of lines advanced. If the print line size specified has been exceeded, the status will indicate the Bottom of Form has been reached. However, the current line count will not be reset.

3.2.3 Skip to Channel - Vertically advance the form to the corresponding line number specified in the printer control table by loading Register Number 2 with the desired Channel number (in Octal). If the channel number line assignment is less than the current line, a Top of Form will be initiated prior to the line feed operation. The current line will be set to correspond to the line assigned to the channel selected.

3.2.4 Elongate - This feature conditions the 1551 printer to print the next line using the Elongate capability. Other Models will ignore the command and return with the Printer Status in the accumulator.

3.3 Print Status -

Upon returning from the selected Print option, the accumulator will contain the Bottom of Form Status indicator in addition to the printer status. The Bottom of Form is set by the Printer options based on the Printer Control Table.



Refer to the Print Function for explanation of Printer Status.

3.4 Special Requirements -

3.4.1 Register Usage

Register 1 - Work Register

Register 2 - Skip Count or Channel

Register 3 - Work Register

Register 4 - Command Register

3.4.2 This segment must be used with the Standard Print Functions.

3.4.3 The Printer option segment must be called from the Library after the standard Print function. The option segment alters the standard Print function.

3.5 Entry Point Linkage -

3.5.1
```
TOF: EQU: P14, 190.   Skip to Top of Form
VTT: EQU: P14, 192.   Line Feed
SKP: EQU: P14, 194.   Skip to Channel
ELG: EQU: P14, 196.   Elongate
```

PROCEDURE TITLE

PRINTER TRANSLATE TABLES (1501 TO USASCII)

Catalog No. 90048

PROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

1.0 INTRODUCTION

The 1501 to USASCII Printer translate tables are used by the print functions to translate the 1501 keyboard code into printer compatible USASCII Code. The usage of this table is controlled by the print functions.

2.0 SPECIFICATION

2.1 Segment ID - 'CT1'

2.2 Source Catalog Number - 60158

2.3 Object Catalog Number - None

2.4 Special Requirements

2.4.1 Memory Requirements - P05-300 thru P05-377 (Octal)

3.0 PRINT CHARACTERS

The 64 standard 1501 characters will be printed with the exception of the following special characters:

3.1 1501 Code 052 prints as "\".

3.2 1501 Code 065 prints as "^".

3.3 1501 Code 074 prints as "[".

3.4 1501 Code 075 prints as "]".

3.5 1501 Code 077 prints as "_".

PROCEDURE TITLE

PRINTER TRANSLATE TABLE (1501 TO TYPEWRITER)

Catalog No. 90049

PROCEDURE TYPE: Instructional

Date 03/01/74

REVISION LEVEL STATUS

Revision Level	Pages Affected	Revised by	Date	Initial

1. INTRODUCTION

The 1501 to Typewriter Printer Translate Table is used by the print function to translate the 1501 Keyboard Code into Typewriter Code. The usage of this Table is controlled by the print functions.

2. SPECIFICATION

2.1 Segment ID - 'TT1'

2.2 Source Catalog Number - 60159

2.3 Object Catalog Number - None

2.4 Special Requirements

2.4.1 Memory Requirements - P05-300 thru P05-377 (Octal).

International Computers

ICL Plaza
Route 12 North
Utica, New York 13502
Phone 315-797-5750

ICL Series 1500
Programmer's Reference Manual
Technical Publication 7851