BOLT BERANEK AND NEWMAN INC

CONSULTING • DEVELOPMENT • RESEARCH

HOSPITAL COMPUTER PROJECT
MEMORANDUM NUMBER SIX-A

HARDWARE STATUS REPORT

25 JANUARY 1965

CAMBRIDGE NEW YORK CHICAGO LOS ANGELES

HOSPITAL COMPUTER PROJECT

MEMORANDUM SIX-A
HARDWARE STATUS REPORT

Principal Investigator
Bolt Beranek and Newman Inc.

Jordan J. Baruch

Principal Investigator
Massachusetts General Hospital

G. Octo Barnett, M.D.

25 January 1965

Submitted to:

Mr. Maurice E. Odoroff
Chief, Program Analysis Branch
National Institute of General
Medical Sciences
National Institutes of Health
Bethesda, Maryland

James B. Hartgering, M.D.
Director of Research + Education
American Hospital Association
840 North Lake Shore Drive
Chicago, Illinois

TABLE OF CONTENTS

I. INTRODUCTION 1

II. HARDWARE 3

A. Some of the Problems 5

B. The Basic Central Processor and ASCII Input 10

Central Processor 11
Memory System 11
Programming 12
Number System 13
Instruction Format 13
Indirect Addressing 14
Operating Speeds 16
Standard Instruction List 16

Arithmetic Instructions 17
Logical Instructions 19
General Instructions 21
Augmented Instructions 25

Perforated Tape Punch 37
Sequence Break Mode 39
Data Communication System 44

C. Additional Instructions 47

General Instructions 49
Augmented Instructions 52
Little Drum Instructions 70
Programming the Data Communication System 75

D. The I-O Processor, Data Channel Programming 80

1. Introduction 81
2. Data Channel 85
2.1 Introduction 85
2.2 Data Channel Hardware 86
2.2.1 Channel Input Register (CIR) 86

2.2.2	Data Address Counter (DAC)	86
2.2.3	Command Address Counter (CAC)	87
2.2.4	Channel Output Register (COR)	88
2.2.5	Channel Word Counter (CWC)	89
2.2.6	Continue Control Register (CCR)	89
2.2.7	Data Direction Flip-Flop (DDF)	90
2.2.8	Program Break Request Flip-Flop (PBF)	90
2.2.9	Channel Status Register (CSR)	90
2.2.10	Device Connect Register (DCR)	90
2.2.11	Need a Completion Pulse Flip-Flop (NAC)	91
2.2.12	Error Stop Register (ESR)	91
2.3	Data Channel Instructions	91
2.3.1	Initialize and Connect the Channel-NLZ	92
2.3.2	Load Data Address-LDA	94
2.3.3	Data Channel Jump-DCJ	95
2.3.4	Load Controller Command-LCC	95
2.3.5	Load Word Count Command-Introduction	96
2.3.6	Load Word Count and Control-LWC	98
2.3.7	Channel Stop-CST	100
2.3.8	Dummy Word Count	101
2.4	Device Controller Commands	102
2.4.1	Fastrand Drum Features	102
2.4.2	Fastrand Controller Commands	107
2.4.3	Uniservo III-C Features	109

2.4.4	Magnetic Tape Controller Command	113
3.	Data Channel Programming	115
3.1	Data Channel Command Sequence	115
3.2	Scatter-Gather	116
3.3	Skip Mode	117
3.4	Data Compare Mode	118
3.5	Illegal Combinations	120
4.	iot Instructions	121
4.1	Rem Rand Out	121
4.1.1	Connect Data Channel	123
4.1.2	Initiate Drum Activity	124
4.1.3	Initiate Mag. Tape Activity	126
4.1.4	Initiate Channel Activity	128
4.2	I-O Subsystem Status	131
4.3	Rem Rand In	131
4.3.1	Channel Status Register	137
4.3.2	Data Channel Command Address Register	142
4.3.3	Data Channel Word Count Register	143
4.3.4	Data Channel Data Address Register	144
4.3.5	Fastrand Controller Status	145
4.3.6	Fastrand Drum Location	149
4.3.7	Fastrand Track Location	151
4.3.8	Uniservo III-C Controller Status	153
4.3.9	Uniservo III-C Read Status	159
4.3.10	Uniservo III-C Write Status	163

Appendix A - Magnetic Tape Recovery Mode
Appendix B - Ready Reference Summary

COMMAND DESCRIPTION LOCATIONS

Arithmetic Instructions

ADD	Add	17
SUB	Subtract	17
MUL	Multiply	17
DIV	Divide	18
IDX	Index	18
ISP	Index and Skip if Positive	19
TAD	Two's complement add	51
AAI	Add to ac from io	54

Logical Instructions

AND	Logical and	19
XOR	Exclusive or	20
IOR	Inclusive or	20

General Instructions

LAC	Load accumulator	21
DAC	Deposit accumulator	21
DAP	Deposit address part	21
DIP	Deposit instruction part	21
LIO	Load in-out register	22
DIO	Deposit in-out register	22
DZM	Deposit zero in memory	22
XCT	Execute	22
JMP	Jump	23
JSP	Jump and save program counter	23
CAL	Call subroutine	23

JDA	Jump and deposit accumulator	24
SAD	Skip if different	24
SAS	Skip if same	25
DCH	Deposit character	49
DCH I	Index and deposit character	50
LCH	Load character in accumulator	50
LCH I	Index and load character	51

Augmented Instructions

LAW	Load the accumulator with	25
SFT	Shift Group	25
RAR	Rotate accumulator right	26
RAL	Rotate accumulator left	27
SAR	Shift accumulator right	27
SAL	Shift accumulator left	27
RIR	Rotate io right	27
RIL	Rotate io left	27
SIR	Shift io right	28
SIL	Shift io left	28
RCR	Rotate combination right	28
RCL	Rotate combination left	28
SCR	Shift combination right	28
SCL	Shift combination left	29
SKP	Skip Group	29
SZA	Zero accumulator	30
SPA	Positive accumulator	30
SMA	Minus accumulator	30
SZO	Zero overflow	30
SPI	Positive io	31

	SZS	Zero sense switch	31
	SZF	Zero flag	31
	SNI	Skip on non-zero io	61
	SNI I	Skip on zero io	61
OPR	Operate Group		32
	CLI	Clear io	32
	LAT	Load accumulator from test switches	32
	LAP	Load accumulator with program counter	32
	CMA	Complement accumulator	33
	HLT	Halt	33
	CLA	Clear accumulator	33
	CLF	Clear program flag	33
	STF	Set flag	34
	NOP	No operation	34
	CMI	Complement io	52
	LAI	Load ac from io	52
	LIA	Load io from ac	52
	SWP	Swap ac and io	52
	AAI	Add to ac from io	54
	CLL	Clear link bit	54
	CML	Complement link	55
	IDA	Index ac	55
	IDC	Index character	55
	IFI	Inclusive or to flags from io	56
	IAI	Inclusive or to ac from io	56
	IIF	Inclusive or to io from flags	56
	SCF	Special clear flags	57
	SCI	Special clear io	57
	SCM	Special complement ac	57

	SZL	Skip on zero link	57
	XAI	Exclusive or to ac from io	57
IOT	In-Out Transfer Group		34
	RPA	Read paper tape, alphanumeric	34
	RPB	Read paper tape, binary	35
	RRB	Read reader buffer	36
	PPA	Punch paper tape, alphanumeric	37
	PPB	Punch paper tape, binary	38
	ESM	Enter sequence-break mode	40
	LSM	Leave sequence-break mode	40
	CBS	Clear break system	41
	DSC	Deactivate sequence-break channel	41
	ASC	Activate sequence-break channel	41
	ISB	Initiate sequence-break	41
	CAC	Clear all channels	41
	EEM	Enter extend mode	42
	LEM	Leave extend mode	43
	CKS	Check status	62
	ERG	Enter ring mode	62
	LRG	Leave ring mode	63
	RCK	Read clock	63
	RNM	Rename memory	64
	RSM	Reset memory selection	64
	ERM	Enter restricted mode	64
	LRM	Leave restrict mode	66
	RTB	Read trap buffer	67
	RRI	Data channel in	67
	RRO	Data channel out	68
	DBA	Drum break address	70

DIA	Drum initial address	70
DWC	Drum word count	70
DCL	Drum core location	71
DRA	Drum read address	71
RCH	Receive character	77
RCC	Receive character and release scanner	77
TCB	Transmit character via buffer	77
TCC	Transmit character via counter	77
RRC	Read receiver counter	78
SSB	Set send buffer	78
RSC	Release scanner	78
CSC	Clear scanner	78

REFERENCES

LIST OF ILLUSTRATIONS

Figure 1. - System Block Diagram 83

Figure 2. - I-O Subsystem Block Diagram 84

Figure 3. - Fastrand Drum Schematic 1 104

Figure 4. - Fastrand Drum Schematic 2 105

Figure 5. - Magnetic Tape Format 111

Figure 6. - Data Channel Indicators 134

Figure 7. - Fastrand Controller Indicators 135

Figure 8. - Magnetic Tape Controller Indicators 136

MEMORANDUM 6-A
STATUS REPORT
HOSPITAL COMPUTER PROJECT

I. INTRODUCTION

This report is in two parts and deals with hardware, hardware-dependent software and some user-service software. It includes:

Memorandum 6-A
Section II - Hardware

A. A brief account of the problems that are met by the design of the hardware configuration.

B. A description of the basic processor and Teletype interface*.

C. A description of the many instructions that have been added to the basic processor to facilitate its present use.

D. A description of the I-O processor and its instructions.

Memorandum 6-B
Section III - Software

A. The Executive, directly dependent software

1. Teletype service routine

* Parts of these descriptions are extracted with permission from Digital Equipment Corporation's manuals.

2. The Swapper
3. The Real Time Clock
4. Fastrand Routines
5. Tape Routines

B. User Service Software

1. Floating Point
2. Syntax Verifier
3. Get and Put
4. Job Hunter
5. Event Detector
6. Type Out Text

C. Programmer's Utility Routines

1. Editor
2. Midas
3. DDT
4. Dump T/S + Restore
5. "T/S Up"

II. HARDWARE

In order to discuss the hardware choice and design, we must consider the special real-time problems of the Hospital Computer Project. These special requirements, and the state of the art when the equipment was ordered dictated, in large part, the present configuration.

The Central Processor

The present Hospital Computer System grew out of one of the country's first operating Time-Shared Computer Projects. This Project was supported under grant number PH43-62-850 from the National Institutes of Health to Bolt Beranek and Newman Inc. and has been elsewhere reported in the literature. The central processor on that Project was a Digital Equipment Corporation PDP-1.

The PDP-1 is an 18-bit 5 microsecond computer. It uses one's-complement arithmetic. Its I-O* is basically word or character oriented. The basic machine has no data channel and all I-O is done on a character or word basis through one of the live registers. It comes basically equipped with 4,096 words (which we shall approximate by 4K) of memory, with a single-channel interrupt system. In 1961 the peripherals generally made available for the PDP-1 consisted of a Soroban typewriter, paper tape punch, paper tape reader, a display scope, and a light pen. A 16-channel interrupt system was added to the basic configuration and additional Soroban typewriters were purchased to give more I-O capa-

* Input-Output operation

bility. Prior to January of 1962 some special purpose Time-Sharing was done with the PDP-1. These efforts were mainly special purpose programs to permit up to three users to sit at console typewriters and to perform a limited number of functions.

Three major additions were made to the basic PDP-1 to enable us to build a Time-Sharing system for it. The first was the addition of more memory, another 4K, and the associated logic to permit the PDP-1 to do extended memory addressing. The scheme here, basically, was to permit the Executive system to stay in one core and to run the user in another core.

The second addition was a rather fast swapping drum. The drum permitted the exchange of 4K of core for 4K of drum to be done in 35 milliseconds. That is, during one revolution of the drum 4K of core memory could be written onto one drum field and 4K could be read off another drum field. The drum had the capacity to store 22 x 4,096 words. Such a swapping drum has been more or less central to the four Time-Sharing systems that have been developed for the PDP-1. The Swapping technique, that is, writing core memory out and reading another load in, is the basic approach in most present Time-Sharing systems that have limited core capacity.

The third major addition consisted of "trapping" logic. Here the computer is made to run in one of two modes, either in Executive mode where all commands are possible or in user mode where those commands which may stop the

machine or affect the I-O area are trapped and then examined by the Executive. In addition, a moderate amount of memory protection was provided for the Executive core bank. The number of typewriters was increased to five and a five station Time-Sharing and debugging system called Simbug was built.(1)

As an experimental system, Simbug was quite successful. It was used extensively for about six months. In June of 1962, just prior to the completion of the Simbug system, Bolt Beranek and Newman Inc. was awarded a contract to investigate a medical communication system. The techniques to be used had as their foundation, those of Time-Sharing. A prototype system(2) was built around the Simbug computer and during this effort we came to understand a great deal about our computational needs. The new central processor, PDP-1d-45, reflects the solution to most of our earlier frustrations and pretty well embodies those changes that we feel are needed in order to handle the restricted set of problems that hospital use represents.

A. Some of the Problems

It was decided rather early in our studies to use some type of serial teleprinter as the initial I-O device, that is Teletypes would serve initially as the remote consoles. We did tie four 60 word-per-minute KSR 28 Teletypes to the PDP-1b. However, we shortly saw two problems:

1. The 3-row keyboard of the Model 28 was unsuited to use by persons habituated to the "standard" typewriter keyboard. We, therefore, went to the Model 33 with its more familiar keyboard.

2. Second, and more fundamental was the timing problem.

The 11 baud Teletype code of the Model 33 consists of 1 baud of start pulse, 7 information pulses, followed by a bit reserved for parity, terminated by 2 stop pulses. The computer's receiver consisted of a serial shift register into which these pulses went. The start pulse would clear the shift register and the information would trickle in and be forwarded to the computer when it was done. Valid information was known to be in the buffer for only 2-1/2 baud times, that is, from the middle of the parity baud until the beginning of the next start baud. This is a period of approximately 25 milliseconds. Hence it was necessary that the PDP-1 should at no time delay longer than 25 milliseconds in handling a Teletype character lest it miss a character from the Teletype line. The swapping drum, because of its high transfer rate, occupies the full

> capacity of the PDP-1b during its revolution time of 35 milliseconds. It was therefore possible to lose a character. On the PDP-1b this problem was met by putting a second character of buffering into the Teletype receivers to provide a full character time plus the 25 milliseconds or 125 milliseconds in toto.

For a similar reason, magnetic tape was quite a problem for the PDP-1b under Time-Sharing. If a user needed to read or write on magnetic tape, it was necessary to lock out all other users during the reading time; one could not run the swapping drum during the periods that magnetic tape was being used. Thirdly, because the swapping drum occupied the full computer's capabilities, every swap cost 35 milliseconds of computation time. Hence, if there were only 10 swaps per second going on, it would cut the efficiency of computation down to about 60%. This fact severely limited the number of users that could be handled.

To meet the problem of the little drum monopolizing the computer, an independent memory scheme divides the potential 64K of the new memory into four 16K groups. Each of these groups has its own independent memory buffer register and memory address register. Logically, memory is treated very much like a piece of I-O gear, that is, the main arithmetic and control unit makes a request to memory for a piece of data and restores modified or unmodified data to memory. The memory control was also given priority

logic to permit more than one processor to make access to it and to respond appropriately. The configuration on the PDP-1d-45 consists of one bank of 16K words and two banks each containing 4K words. The 16K bank is used for the Executive and the 4K banks are used for the users.

With the independent memory scheme then we may run one user in one of the 4K banks while swapping an additional user into the other 4K bank. Hence, swapping and computation occur simultaneously. Since the computer is always running and always has access to at least the 16K bank of memory, never being blocked out by the little drum, we may handle interrupts from the communication lines at the proper rates and need only single character line buffers.

The original central processor also had severe high-speed I-O problems because of its multiply and divide logic. During the process of multiplication and division it used the memory buffer as a third arithmetic element. Hence, during these processes it was impossible to read or write from memory. Divide, the worst case situation, can take as long as 42 microseconds. Hence, if a high-speed I-O device is trying to feed memory, it cannot do so through the normal memory buffer on the PDP-1 since that memory buffer may be occupied in doing a divide. It is possible to get around this problem with appropriate buffering, but when the transfer rates get high enough, it becomes desirable to take the data channel commands directly out of memory. Buffering, under this condition, becomes inordinately cumbersome.

In the new processor we therefore decided to add a data channel handled as an independent processor. This channel has the ability, after once being activated by an I-O command from the central processor, to make direct memory references for additional commands or for data transfers. It does so independently of the central processor, directly through the memory buffer system. It is, after the central processor and the little drum, the third processor on the memory buffer system.

One of the great deficiencies of the PDP-1b system was its lack of the bulk storage needed by the medical information system. To meet this need we acquired a large bulk-storage drum. This device provides approximately 60 million characters of storage and a 1.1 megacycle bit transfer rate. In addition, we acquired an initial group of two magnetic tape units for long-term bulk storage. The bulk drum, the tape units, the controllers for these devices, and the data channel were acquired from the Univac Division of Sperry Rand, who were most cooperative in implementing many of our design changes.

Because it would not be known in advance in which bank the user would be running, it was necessary to add what are essentially relocation registers or page registers, four 2-bit rename registers which work in the following manner:

If the computer makes a memory reference to bank 0 then the contents of rename register 0 are substituted for the bank address and hence the memory reference will go to the physical bank specified by rename register 0. This mode of operation prevents multiple addressing and gives us all the relocation facilities that we need at present. In addition, memory protection registers were added to protect each of the 16K banks when the machine was in user mode.

A trap buffer was added to the trapping logic so that when an instruction was trapped it would be easy to determine what the instruction was. Prior to the installation of the trap buffer we had to look in the memory location addressed by the program counter to see what the instruction was. In some instances this led to difficulties since the instruction may have been an "execute" command while we were really interested in the command at the end of the "execute" chain.

The experience and logic reported above led to the hardware system whose details are set forth below.

B. The Basic Central Processor and ASCII Input

The central processor is a high-speed, solid state digital computer. It is a single address, single instruction, stored program computer with powerful program features. Five-megacycle circuits, a magnetic core memory and fully parallel processing make possible a computation rate of 100,000 additions per second. Preventive maintenance is provided for by built-in marginal checking circuits.

Central Processor

The central processor contains the control, arithmetic and memory addressing elements and the memory buffer register. The word length is 18 binary digits. Instructions are performed in multiples of the memory cycle time of five microseconds. Add, subtract, deposit, and load, for example, are two-cycle instructions requiring 10 microseconds. Multiplication by subroutine requires 325 microseconds on the average. An optional automatic multiply and divide package is available. In our case multiplication requires an average of 20 microseconds. Program features include: single address instructions, multiple step indirect addressing and logical arithmetic commands. Console features include: flip-flop indicators grouped for convenient octal reading, six program flags for automatic setting and computer sensing and six sense switches for manual setting and computer sensing.

Memory System

The coincident-current, magnetic core memory holds 4096 words of 18 bits each. Memory capacity may be readily expanded, in increments of 4096 words, to a maximum of 65,536 words. The read-rewrite time of the memory is five microseconds, the basic computer rate. Driving currents are automatically adjusted to compensate for temperature variations between 50 and 110 degrees Fahrenheit.

Programming PDP-1

The central processor contains the Control Elements, the Memory Buffer Register, the Arithmetic Element, and the Memory Addressing Element. The Control Element governs the complete operation of the computer including memory timing, instruction performance and the initiation of input-output commands. The Arithmetic Element, which includes the Accumulator and the In-Out Register, performs the arithmetic operations. The Memory Addressing Element, which includes the Program Counter and the Memory Address Register, performs address bookkeeping and modifications.

The programming features include:

Multiple step indirect addressing

Boolean operations

Twelve variations of arithmetic and logical shifting, operating on 18 or 36 bits

Fifteen basic conditional skip instructions (expandable by combining to form the inclusive OR of the separate conditions)

Three different subroutine calling instructions

Micro-coded operate instructions

Index and Index-Conditional instructions

Execute instruction

Load-immediate instructions

Six independent flip-flops, called "program flags", are available for use as program switches or special in-out synchronizers.

Number System

The processor is a "fixed point" machine using binary arithmetic. Negative numbers are represented as the one's complement of the positive numbers. Bit 0 is the sign bit which is ZERO for positive numbers. Bits 1 to 17 are magnitude bits, with Bit 1 being the most significant and Bit 17 being the least significant. To avoid a frequent point of confusion in one's complement arithmetic, the representation of -0 is automatically changed to +0 in certain arithmetic operations.

The conversion of decimal numbers into the binary system for use by the machine is performed by subroutines. Similarly the output conversion of binary numbers into decimals is done by subroutine. Operations for floating point numbers are handled by interpretive programming

Instruction Format

The Bits 0 through 4 define the instruction code; thus there are 32 possible instruction codes. The instructions may be divided into two classes:

Memory reference instructions

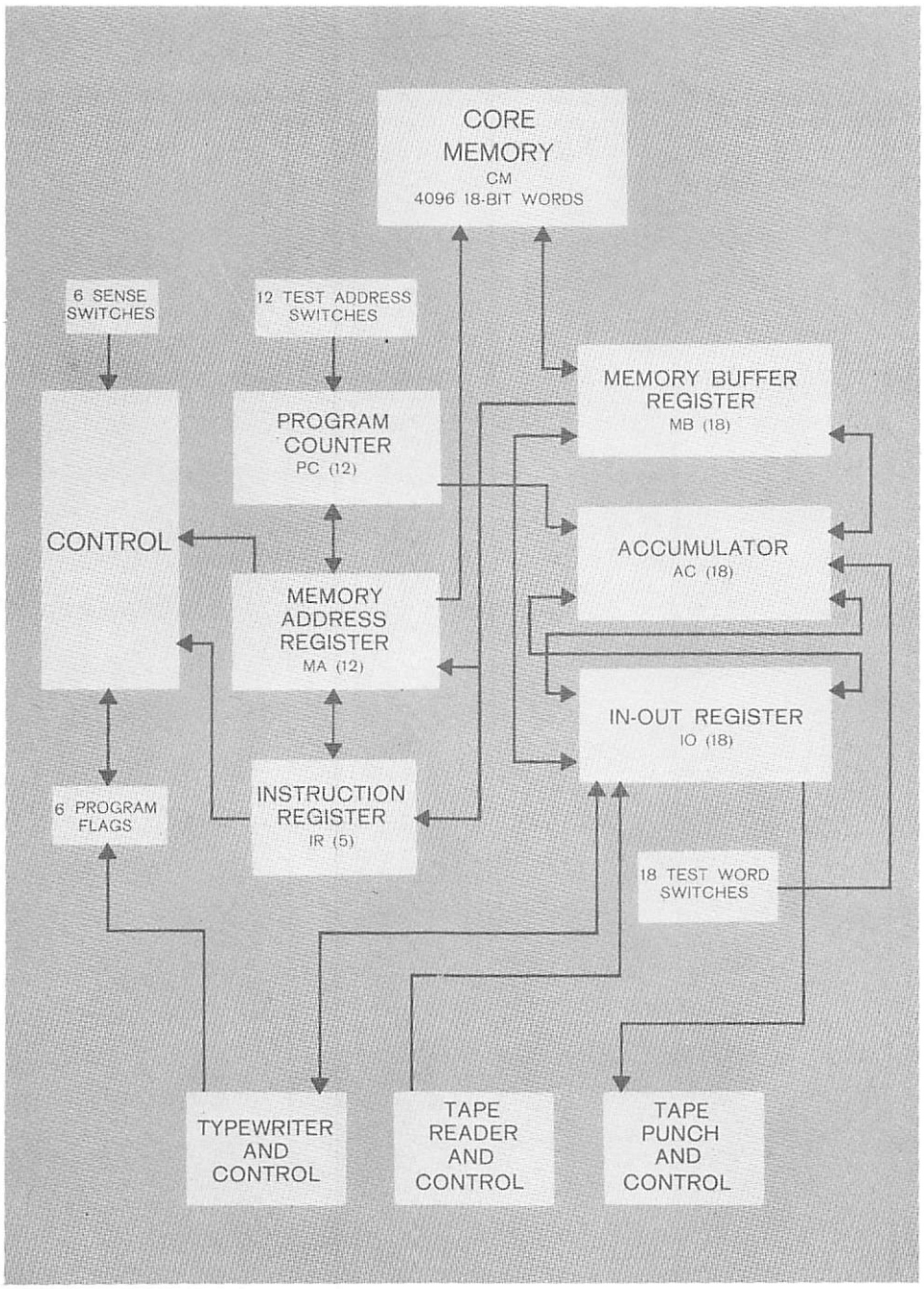
Augmented instructions

In the memory reference instructions, Bit 5 is the indirect address bit. The instruction memory address, Y, is in Bits 6 through 17. These digits are sufficient to address 4096 words of memory.

The augmented instructions use Bits 5 through 17 to specify variations of the basic instruction. For example, in the shift instruction, Bit 5 specifies direction of shift, Bit 6 specifies the character of the shift (arithmetic or logical), Bits 7 and 8 enable the registers (01 = AC, 10 = IO, and 11 = both) and Bits 9 through 17 specify the number of steps.

Indirect Addressing

A memory reference instruction which is to use an indirect address will have a ONE in Bit 5 of the instruction word. The original address, Y, of the instruction will not be used to locate the operand, jump location, etc., of the instruction, as is the normal case. Instead, it is used to locate a memory register whose contents in Bits 6 through 17 will be used as the address of the original instruction. Thus, Y is not the location of the operand but the location of the location of the operand. If the memory register containing the indirect address also has a ONE in Bit 5, the indirect addressing procedure is repeated and a third address is located. There is no limit to the number of times this process can be repeated.



Basic Logic Organization

Operating Speeds

Operating times are multiples of the memory cycle of 5 microseconds. Two-cycle instructions refer twice to memory and thus require 10 microseconds for completion. Examples of this are add, subtract, deposit, load, etc. The jump, augmented and combined augmented instructions need only one call on memory and are performed in 5 microseconds.

In-Out Transfer instructions that do not include the optional wait function require 5 microseconds. If the in-out device requires a wait time for completion, the operating time depends upon the device being used.

Each step of indirect addressing requires an additional 5 microseconds.

Standard Instruction List

This list includes the title of the instruction, the normal execution time of the instruction, (i.e., the time with no indirect address), the mnemonic code of the instruction, and the operation code number. In the following list, the contents of a register are indicated by C ( ). Thus C (Y) means the contents of memory at Address Y: C (AC) means the contents of the Accumulator; C (IO) means the contents of the In-Out Register.

MEMORY REFERENCE INSTRUCTIONS

Arithmetic Instructions

Add (10 μsec)
add Y Operation Code 40

The new C (AC) are the sum of C (Y) and the original C (AC). The C (Y) are unchanged. The addition is performed with 1's complement arithmetic. If the sum of two like-signed numbers yields a result of the opposite sign, the overflow flip-flop will be set (see Skip Group instructions). A result of minus zero is changed to plus zero.

Subtract (10 μsec)
sub Y Operation Code 42

The new C (AC) are the original C (AC) minus the C (Y). The C (Y) are unchanged. The subtraction is performed using 1's complement arithmetic. When two unlike-signed numbers are subtracted, the sign of the result must agree with the sign of the original Accumulator, or overflow flip-flop will be set (see Skip Group instructions). A result of minus zero is changed to plus zero, with the exception that (-0)-(+0)=(-0).

Multiply (14 to 25 μsec)
mul Y Operation Code 54

The product of C(AC) and C(Y) is formed in the AC and IO registers. The sign of the product is in the AC sign bit. IO Bit 17 also contains the sign of the product. The magnitude of the product is the 34-bit string from AC Bit 1

through IO Bit 16. The C(Y) are not affected by this instruction. If the entire product results in a minus zero it is changed to a plus zero.

Divide (30 to 40 μsec, except on overflow, 12 μsec)
div Y Operation Code 56

The dividend must be in the AC and IO registers in the form indicated in the instruction, Multiply. IO bit 17 is ignored. The divisor is the C(Y). At the completion of the instruction, the C(AC) are the quotient and the C(IO) are the remainder. The sign of the remainder (in IO bit zero) is the sign of the dividend. The instruction that follows a DIV will be skipped unless an overflow occurs. The C(Y) are not affected by this instruction. If the remainder or quotient result in minus zero, that value is changed to plus zero.

If the magnitude of the high order part of the dividend is equal to or greater than the magnitude of the divisor, an overflow is indicated. In this case, the following instruction is not skipped. The original C(AC) and C(IO) are restored. The overflow flip-flop is not affected.

Index (10 μsec)
idx Y Operation Code 44

The C (Y) are replaced by C (Y) + 1 which are left in the Accumulator. The previous C (AC) are lost. Overflow is not indicated. If the original C (Y) equals the integer, -1, the result after indexing is plus zero.

Index and Skip if Positive (10 μsec)
isp Y Operation Code 46

The C (Y) are replaced by C (Y) + 1 which are left in the Accumulator. The previous C (AC) are lost. If, after the addition, the Accumulator is positive, the Program Counter is advanced one extra position and the next instruction in the sequence is skipped. Overflow is not indicated. If the original C (Y) equals the integer, -1, the result after indexing is plus zero and the skip takes place.

LOGICAL INSTRUCTIONS

Logical AND (10 μsec)
and Y Operation Code 02

The bits of C (Y) operate on the corresponding bits of the Accumulator to form the logical AND. The result is left in the Accumulator. The C (Y) are unaffected by this instruction.

LOGICAL AND TABLE

AC Bit	Y Bit	Result
0	0	0
0	1	0
1	0	0
1	1	1

Exclusive OR (10 μsec)
xor Y Operation Code 06

The bits of C (Y) operate on the corresponding bits of the Accumulator to form the exclusive OR. The result is left in the Accumulator. The C (Y) are unaffected by this order.

EXCLUSIVE OR TABLE

AC Bit	Y Bit	Result
0	0	0
0	1	1
1	0	1
1	1	0

Inclusive OR (10 μsec)
ior Y Operation Code 04

The bits of C (Y) operate on the corresponding bits of the Accumulator to form the inclusive OR. The result is left in the Accumulator. The C (Y) are unaffected by this order.

INCLUSIVE OR TABLE

AC Bit	Y Bit	Result
0	0	0
0	1	1
1	0	1
1	1	1

GENERAL INSTRUCTIONS

Load Accumulator (10 μsec)
lac Y Operation Code 20

The C (Y) are placed in the Accumulator. The C (Y) are unchanged. The original C (AC) are lost.

Deposit Accumulator (10 μsec)
dac Y Operation Code 24

The C (AC) replace the C (Y) in the memory. The C (AC) are left unchanged by this instruction. The original C (Y) are lost.

Deposit Address Part (10 μsec)
dap Y Operation Code 26

Bits 6 through 17 of the Accumulator replace the corresponding digits of memory register Y. C (AC) are unchanged as are the contents of Bits 0 through 5 of Y. The original contents of Bits 6 through 17 of Y are lost.

Deposit Instruction Part (10 μsec)
dip Y Operation Code 30

Bits 0 through 5 of the Accumulator replace the corresponding digits of memory register Y. The Accumulator is unchanged as are Bits 6 through 17 of Y. The original contents of Bits 0 through 5 of Y are lost.

Load In-Out Register (10 μsec)
lio Y Operation Code 22

The C (Y) are placed in the In-Out Register. C (Y) are unchanged. The original C (IO) are lost.

Deposit In-Out Register (10 μsec)
dio Y Operation Code 32

The C (IO) replace the C (Y) in memory. The C (IO) are unaffected by this instruction. The original C (Y) are lost.

Deposit Zero in Memory (10 μsec)
dzm Y Operation Code 34

Clears (sets equal to plus zero) the contents of register Y.

Execute (5 μsec plus time of instruction executed)
xct Y Operation Code 10

The instruction located in register Y is executed. The Program Counter remains unchanged (unless a jump or skip were executed). If a skip instruction is executed (by xct y), the next instruction to be executed will be taken from the address of the xct y plus one or the address of the xct y plus two depending on the skip condition. Execute may be indirectly addressed, and the instruction being executed may use indirect addressing. An xct instruction may execute other xct commands.

Jump (5 μsec)
jmp Y Operation Code 60

The Program Counter is reset to Address Y. The next instruction that will be executed will be taken from Memory Register Y. The original contents of the Program Counter are lost.

Jump and Save Program Counter (5 μsec)
jsp Y Operation Code 62

The contents of the Program Counter are transferred to bits 6 through 17 of the AC, the state of the overflow flip-flop to bit zero of the AC, and zeroes fill bits 1 through 5 of the AC. When the transfer takes place, the Program Counter holds the address of the instruction following the jsp. The Program Counter is then reset to Address Y. The next instruction that will be executed will be taken from Memory Register Y. The original C (AC) are lost.

Call Subroutine (10 μsec)
cal Y Operation Code 16

The address part of the instruction, Y, is ignored. The contents of the Accumulator are deposited in Memory Register 100. The contents of the Program Counter (holding the address of the instruction following the cal) are transferred to bits 6 through 17 of the AC, the state of the overflow flip-flop to bit zero of the AC, and zeroes fill bits 1 through 5 of the AC. The next instruction that will

be executed is taken from Memory Register 101. The cal instruction requires that the indirect bit be ZERO. The instruction may be used as part of a master routine to call subroutines.

Jump and Deposit Accumulator (10 μsec)
jda Y Operation Code 17

The contents of the Accumulator are deposited in Memory Register Y. The contents of the Program Counter (holding the address of the instruction following the jda) are transferred to bits 6 through 17 of the AC, the state of the overflow flip-flop to bit zero of the AC, and zeroes fill bits 1 through 5 of the AC. The next instruction that will be executed is taken from Memory Register Y + 1. The jda instruction requires that the indirect bit be a ONE, but indirect addressing does not occur. The instruction is equivalent to the instructions dac Y, followed by jsp Y + 1.

Skip if Accumulator and Y differ (10 μsec)
sad Y Operation Code 50

The C (Y) are compared with the C (AC). If the two numbers are different, the Program Counter is indexed one extra position and the next instruction in the sequence is skipped. The C (AC) and the C (Y) are unaffected by this operation.

Skip if Accumulator and Y are the same (10 μsec)
sas Y Operation Code 52

The C (Y) are compared with the C (AC). If the two numbers are identical, the Program Counter is indexed one extra position and the next instruction in the sequence is skipped. The C (AC) and C (Y) are unaffected by this operation.

AUGMENTED INSTRUCTIONS

Load Accumulator with N (5 μsec)
law N Operation Code 70

The number in the memory address bits of the instruction word is placed in the Accumulator. If the indirect address bit is ONE, (-N) is put in the Accumulator.

Shift Group (5 μsec)
sft Operation Code 66

This group of instructions will rotate or shift the Accumulator and/or the In-Out Register. When the two registers operate combined, the In-Out register is considered to be an 18-bit magnitude extension of the right end of the Accumulator.

Rotate is a non-arithmetic cyclic shift. That is, the two ends of the register are logically tied together and information is rotated as though the register were

a ring. Shift is an arithmetic operation and is, in effect, multiplication of the number in the register by $2 \pm n$, where N is the number of shifts; plus is left and minus is right.

As bits are shifted out from one end of a register they are replaced at the other end by ones if the number is negative and zeroes if the number is positive. The sign bit is not shifted.

The number of the shift or rotate steps to be performed (N) is indicated by the number of ONE's in Bits 9 through 17 of the instruction word. Thus, Rotate Accumulator Right nine times is 671777. A shift or rotate of one place can be indicated nine different ways. The usual convention is to use the right end of the instruction word (rar 1 = 671001).

When operating the PDP-1 in the single-step or single-instruction mode, shift group instructions may appear to be operating incorrectly (i.e., judging from the indicator lights of the control console). This occurs because some shift group instructions overlap into the beginning of the next instruction.

Rotate Accumulator Right (5 μsec)
rar N Operation Code 671

Rotates the bits of the Accumulator right N position, where N is the number of ONE's in Bits 9-17 of the instruction word.

Rotate Accumulator Left (5 μsec)
ral N Operation Code 661

Rotates the bits of the Accumulator left N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Shift Accumulator Right (5 μsec)
sar N Operation Code 675

Shifts the contents of the Accumulator Right N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Shift Accumulator Left (5 μsec)
sal N Operation Code 665

Shifts the contents of the Accumulator left N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Rotate In-Out Register Right (5 μsec)
rir N Operation Code 672

Rotates the bits of the In-Out Register right N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Rotate In-Out Register Left (5 μsec)
ril N Operation Code 662

Rotates the bits of the In-Out Register left N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Shift In-Out Register Right (5 μsec)
sir N Operation Code 676

Shifts the contents of the In-Out Register right N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Shift In-Out Register Left (5 μsec)
sil N Operation Code 666

Shifts the contents of the In-Out Register left N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Rotate AC and IO Right (5 μsec)
rcr N Operation Code 673

Rotates the bits of the combined registers right in a single ring N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Rotate AC and IO Left (5 μsec)
rcl N Operation Code 663

Rotates the bits of the combined registers left in a single ring N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Shift AC and IO Right (5 μsec)
scr N Operation Code 677

Shifts the contents of the combined registers right N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Shift AC and IO Left (5 μsec)
scl N Operation Code 667

Shifts the contents of the combined registers left N positions, where N is the number of ONE's in Bits 9-17 of the instruction word.

Skip Group (5 μsec)
skp Operation Code 64

This group of instructions senses the state of various flip-flops and switches in the machine. The address portion of the instruction selects the particular function to be sensed. All members of this group have the same operation code. The instructions in the Skip Group may be combined to form the inclusive OR of the separate skips. Thus, if Address 3000 is selected, the skip would occur if the overflow flip-flop equals ZERO or if the In-Out Register is positive.

The combined instruction would still take 5 microseconds.

The intent of any skip instruction can be reversed by making Bit 5 (normally the Indirect Address Bit) equal to ONE. For example, the Skip on Zero Accumulator instruction, with Bit equal to one, becomes Do Not Skip on Zero Accumulator.

Skip on ZERO Accumulator (5 μsec)
sza Address 0100

If the Accumulator is equal to plus ZERO (all bits are ZERO), the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

Skip on Plus Accumulator (5 μsec)
spa Address 0200

If the sign bit of the Accumulator is ZERO, the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

Skip on Minus Accumulator (5 μsec)
sma Address 0400

If the sign bit of the Accumulator is ONE, the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

Skip on ZERO Overflow (5 μsec)
szo Address 1000

If the overflow flip-flop is a ZERO, the Program Counter is advanced one extra position and the next instruction in the sequence will be skipped. The overflow flip-flop is cleared by the instruction. This flip-flop is set only by an addition or subtraction that exceeds the capacity of the Accumulator. (See definition of add and subtract instructions). The overflow flip-flop is not cleared by arithmetic operations which do not cause an

overflow. Thus, a whole series of arithmetic operations can be checked for correctness by a single szo. The overflow flip-flop is cleared by the "Start" Switch.

Skip on Plus In-Out Register (5 μsec)
spi Address 2000

If the sign digit of the In-Out Register is ZERO, the Program Counter is indexed one extra position and the next instruction in sequence is skipped.

Skip on ZERO Switch (5 μsec)
szs Addresses 0010, 0020,...0070

If the selected Sense Switch is ZERO, the Program Counter is advanced one extra position and the next instruction in the sequence will be skipped. Address 10 senses the position of Sense Switch 1, Address 20 Switch 2, etc. Address 70 senses all the switches. If 70 is selected all 6 switches must be ZERO to cause the skip.

Skip on ZERO Program Flag (5 μsec)
szf Addresses 0001 to 0007

If the selected program flag is a ZERO, the Program Counter is advanced one extra position and the next instruction in the sequence will be skipped. Address 1 selects Program Flag 1, etc. Address 7 selects all program flags which must be ZERO to cause the skip.

Operate Group (5 μsec)
opr Operation Code 76

This instruction group performs miscellaneous operation on various Central Processor Registers. The address portion of the instruction specifies the action to be performed.

The instructions in the Operate Group can be combined to give the union of the functions. The instruction opr 3200 will clear the AC, put TW to AC, and complement AC.

Clear In-Out Register (5 μsec)
cli Address 4000

Clears (sets equal to plus zero) the In-Out Register.

Load Accumulator from Test Word (5 μsec)
lat Address 2000

Forms the inclusive OR of the C (AC) and the contents of the Test Word. This instruction is usually combined with Address 0200 (Clear Accumulator), so that C (AC) will equal the contents of the Test Word Switches.

Load Accumulator with Program Counter (5 μsec)
lap Address 0100

Forms the inclusive OR of the C (AC) and the contents of the Program Counter (which contains the address of the instruction following the lap) in AC bits 6 through 17. Also, the inclusive OR of AC bit zero and the state of the overflow flip-flop is formed in AC bit zero. This instruction is

usually combined with address 0200 (clear accumulator) so that the C (AC) will equal the contents of the overflow flip-flop (in AC bit zero) and the contents of the Program Counter (in AC bits 6 through 17). AC bits 1 through 5 are filled with zeroes.

Complement Accumulator (5 μsec)
cma Address 1000

Complements (changes all ones to zeroes and all zeroes to ones) the contents of the Accumulator.

Halt
hlt Address 0400

Stops the computer.

Clear Accumulator (5 μsec)
cla Address 0200

Clears (sets equal to plus zero) the contents of the Accumulator.

Clear Selected Program Flag (5 μsec)
clf Address 0001 to 0007

Clears the selected program flag. Address 01 clears Program Flag 1, 02 clears Program Flag 2, etc. Address 07 clears all program flags.

Set Selected Program Flag (5 μsec)
stf Addresses 0011 to 0017

Sets the selected program flag. Address 11 sets Program Flag 1; 12 sets Program Flag 2, etc. Address 17 sets all program flags.

No Operation (5 μsec)
nop Address 0000

The state of the computer is unaffected by this operation, and the Program Counter continues in sequence.

In-Out Transfer Group (5 μsec without in-out wait)
iot Operation Code 72

The variations within this group of instructions perform all of the in-out control and information transfer functions. If Bit 5 (normally the Indirect Address bit) is a ONE, the computer will enter a special waiting state until the completion pulse from the activated device has returned. When this device delivers its completion, the computer will resume operation of the instruction sequence. See additional instructions below.

Read Perforated Tape, Alphanumeric
rpa 720001

This instruction reads one line of tape (all eight Channels) and transfer the resulting 8-bit code to the Reader Buffer.

If bits 5 and 6 of the rpa instruction are both zero (720001), the contents of the Reader Buffer must be transferred to the IO Register by executing a rrb instruction. When the Reader Buffer has information ready to be transferred to the IO Register, Status Register Bit 1 is set to one. If bits 5 and 6 are different (730001 or 724001) the 8-bit code read from tape is automatically transferred to the IO Register via the Reader Buffer and appears as follows:

IO BITS	10	11	12	13	14	15	16	17
TAPE CHANNELS	8	7	6	5	4	3	2	1

The remaining bits of the IO Register are set to zero.

The code of the off-line preparation typewriter (Friden FIO-DEC Recorder-Reproducer) contains an odd parity bit. This bit may be checked by the read-in program. The FIO-DEC Code can then be converted to the Concise (6-bit) Code merely by dropping the eigth bit (parity).

Read Perforated Tape, Binary
rpb 720003

The instruction reads three lines of tape (six channels per line) and assembles the resulting 18-bit word in the Reader Buffer. For a line to be recognized by this instruction Channel 8 must be punched (lines with Channel 8 not punched will be skipped over). Channel 7 is ignored. The instruction sub 5137, for example, appears on tape and is assembled by rpb as follows:

Channel	8	7	6	5	4	3	2	1
Line 1	X		X				X	
Line 2	X		X		X			X
Line 3	X			X	X	X	X	X
Reader Buffer	100		010	101	001	011		111

(Vertical dashed line indicates sprocket holes and the symbols "X" indicate holes punched in tape).

If bits 5 and 6 of the rpb instruction are both zero (720002), the contents of the Reader Buffer must be transferred to the IO Register by executing a rrb instruction. When the Reader Buffer has information ready to be transferred to the IO Register, Status Register Bit 1 is set to one. If bits 5 and 6 are different (730002 or 724002) the 18-bit word read from tape is automatically transferred to the IO Register via the Reader Buffer.

Read Reader Buffer
rrb 720030

When the rpa or rpb instructions are given with bits 5 and 6 both zero (720001 or 720002) information read from tape fills the Reader Buffer, but is not automatically transferred to the IO Register. To accomplish the transfer, these instructions must be followed by a rrb instruction. In addition, the rrb instruction clears Status Register Bit 1.

Read-In Mode

This is a special mode activated by the "Read-In" switch on the console. It provides a means of entering programs which does not rely on programs already in memory. Pushing the "Read-In" switch starts the reader in the binary mode. The first group of three lines, and alternate succeeding groups of three lines, are interpreted as "Read-In" mode instructions. Even-numbered groups of three lines are data. The "Read-In" mode instructions must be either "deposit in-out" (dioY) or "jump" (jmp Y). If the instruction is dio Y, the next group of three binary lines will be stored in memory location Y and the reader continues moving. If the instruction is jmp Y, the "Read-In" mode is terminated, and the computer will commence operation at the address of the jump instruction.

PERFORATED TAPE PUNCH

The standard Perforated Tape Punch operates at a speed of 63 lines per second. It can operate the alphanumeric mode or the binary mode.

Punch Perforated Tape, Alphanumeric
ppa 720005

For each In-Out Transfer instruction one line of tape is punched. In-Out Register Bit 17 conditions Hole 1. Bit 16 conditions Hole 2, etc. Bit 10 conditions Hole 8.

Punch Perforated Tape, Binary
ppb 720006

For each In-Out Transfer instruction one line of tape is punched. In-Out Register Bit 5 conditions Hole 1. Bit 4 conditions Hole 2, etc. Bit 0 conditions Hole 6. Hole 7 is left blank. Hole 8 is always punched in this mode.

SEQUENCE BREAK MODE

The purpose of the Sequence Break Mode (or program interrupt) is to allow concurrent operation of several in-out devices and the main program sequence. It also provides a means of indicating to the computer that an in-out device is ready to accept or furnish data.

Interrupt requests can be received from a maximum of 16 lines. Each such request sets a unique status bit. If the channel is free, the main program sequence is interrupted after completion of the current memory cycle and the C (AC) are automatically stored in memory location zero, the C (PC) in location 1, and the C (IO) in location 2. The time required to accomplish this is 15 μsec. The C (PC) as stored in location 1 includes the state of the overflow flip-flop in bit zero. The Program Counter is then reset to the address 0003 and the program begins operating in the new sequence. The program beginning at location 0003 is usually designed to inspect the status bits, through the use of the Check Status instruction, to determine which in-out device caused the interrupt. A jump to the appropriate in-out subroutine can then be executed. Each such subroutine is terminated by the following instructions:

lac	0000 (to restore the AC)
lio	0002 (to restore the IO)
jmp (indirect)	0001 (to resume the main program)

The last of these three instructions restores the overflow and PC flip-flops and frees the channel thus allowing the next interrupt request received by the system to be processed. Interrupt requests that occurred while the channel was busy set status bits, and cause interrupts when the channel next becomes free.

In the standard PDP-1 the reader, punch, and typewriter are attached to the One-Channel Sequence Break System and five status bits are defined (see Check Status Instruction). The number of status bits is expanded as required by optional in-out equipment.

Enter Sequence Break Mode
esm 720055

This instruction turns on the Sequence Break System, allowing automatic interrupts to the main sequence to occur.

Leave Sequence Break Mode
lsm 720054

This instruction turns off the Sequence Break System, thus preventing interrupts to the main sequence. Should interrupt requests occur while the system is off, the status bits will, nevertheless, continue to be set.

Clear Sequence Break System
cbs 720056

This instruction clears certain control flip-flops in the Sequence Break System thus nullifying the effect of any interrupt requests just granted or about to be granted (i.e., just prior to the transfer of the C (AC) to location zero).

Deactivate Sequence Break Channel
dsc 72kn50

Turn off the channel specified by kn, where kn equals 00 for channel zero, 01 for channel one, etc., and 17 for channel fifteen.

Activate Sequence Break Channel
asc 72kn51

Turn on the channel specified by kn.

Initiate Sequence Break
isb 72kn52

Initiate a sequence break on the channel specified by kn regardless of whether the channel is on or off.

Clear All Channels
cac 720053

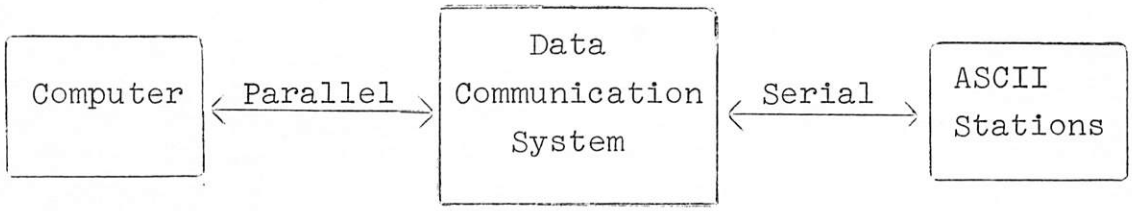
Turns off all sixteen channels.

Enter Extend Mode (5 μsec)
eem 724074

This instruction places the computer in the single-level, indirect address mode called "extend." In this mode, all memory reference instructions that are indirectly addressed refer to the location of a word which is taken as a 16-bit effective address. This address is contained in bits 2 through 17 of the specified word. The Program Counter (PC) and the Memory Address Register (MA) both become 16-bit registers. When a jsp, jda, cal, or lap (with address 300) instruction is executed, the AC receives the state of the overflow flip-flop in bit zero, the state of the indirect address mode (extend = 1, normal = 0) in bit 1, and the contents of the extended Program Counter in bits 2 through 17. Instructions not indirectly addressed are executed as in the standard PDP-1, but refer to the 4096 words in the memory module designated by the program counter extension, PC bits 2 through 5. Only bits 6 through 17 of the extended Program Counter act as a counter. Therefore, unless a transfer of control is indicated, an instruction in location 7777 is followed by the instruction in location 0000 of the same memory module, as specified by PC bits 2 through 5. In the extend mode, the cal instruction uses memory locations 0100 and 0101 in memory module zero.

Leave Extend Mode (5 μsec)
lem 720074

This instruction places the computer in the multiple-level, indirect address mode called "normal." In this mode, the PDP-1 operates as usual and all addressing refers to the 4096 words in the memory module designated by the program counter extension, PC bits 2 through 5. As in the extend mode, the instructions jsp, jda, cal, and lap (with address 300) supply the AC with the contents of the overflow, indirect address mode, and PC flip-flops. In the normal mode, the cal instruction uses memory locations 0100 and 0101 in the memory module designated by the program counter extension, PC bits 2 through 5.



The Data Communication System (DCS) is a real-time interface between Serial ASCII stations and a computer. Its basic function is to receive and transmit characters. It will work at bit rates from DC to 400 bits per second.

A. When receiving; characters of different data rates and unit codes arrive from the stations in serial form.

1. The DCS converts the signals to Digital voltage levels.

2. The characters are converted from serial to parallel form.

3. The characters are forwarded to the computer.

B. When transmitting: characters in parallel form are presented to the DCS by the computer.

1. The characters are converted to serial ASCII form of the correct data rate and unit code.

2. They are converted from Digital voltage levels to station signal levels.

3. They are sent to the stations.

C. The Flag Scanner:

1. Decodes and interprets computer instructions.

2. Forwards received characters to the computer upon request from the computer.

3. Sends characters to the transmitter modules when instructed by the computer.

4. Scans each line in search of activated flags.

5. Notifies the computer when an activated flag has been found.

6. Forwards the station number requiring service to the computer upon request from the computer.

The DCS contains a precision, crystal-controlled clock which generates highly accurate timing pulses. The transmitter and receiver modules use the pulses to sample the serial signals. An additional crystal clock can be added to accommodate multiple speeds.

A crystal clock is also used to generate timing pulses that control the search logic of the scanner.

The scanning mechanism is modular. Each expansion permits an additional eight (8) stations (1 group) to be scanned.

A rotating priority scanner notifies the computer when an active flag has been found. The computer program requests the station number and then handles the character. Programmed priority of the stations is permitted.

The flag scanner operates at very high speeds:

1. The maximum total time required to examine 64 inactive stations: 32 microseconds.

2. The maximum total time to search, notify the computer, and continue search for 64 simultaneously active stations: 544 microseconds (exclusive of computer interrupt and programming cycles).

3. The minimum time required to find the next active station upon being released by the computer: 6 microseconds.

4. The maximum time required to find the next active station (station being serviced minus one) upon being released by the computer: 92 microseconds.

Note that the volume of communications traffic generated is limited by the speed of the computer program to handle those stations and not by the speed of the DCS.

C. Additional Instructions

The following section describes in detail the operation of the instructions in the new PDP-1d-45 which have been incorporated to facilitate character handling, multi-precision arithmetic Time-Sharing and ASCII communications.

SUMMARY OF PROGRAM ACCESSIBLE NEW HARDWARE

	Real-Time Clock	A 16-bit millisecond clock read into I-O by rck.
*INC	Increment ff	Set when executing dch i or lch i; performs the automatic counting feature of these instructions.
*LD	Load/Deposit ff's	Receives the character pointer bits, i.e., MB 0 and 1 when performing lch, dch, lch i, dch i instructions.
LNK	Link ff	Stores the carry out of AC bit 0. Used when performing double precision arithmetic. Cleared by cll; complemented by cml.
	Memory Rename	Four pairs of flip-flops, r_0, r_1, r_2, r_3. Set by rnm; reset by rsm. Access to a physical bank of memory is determined by the value of the r register addressed.
PROTECT	Memory Protect ff's	Four flip-flops set and cleared by erm and I-O bits 0-3. Causes the instruction addressing a protected memory to be trapped when operating on restrict mode.
RM	Restrict Mode ff	Set by erm; reset by lrm. Causes the central processor to operate in restrict mode.
RNG	Ring Mode ff	Set by erg; reset by lrg. Also affected by certain Special Operate instructions. Causes the central processor to operate in ring mode.
	Trap Buffer	An 18-bit register read into I-O and cleared by rtb.

*Not directly accessible by program.

DESCRIPTION OF OPERATION

Memory Reference Instructions

Deposit Character in Memory
dch Y Oper. Code 14

This instruction is interpreted as being deferred and therefore requires three memory cycles for execution. Location Y contains a two bit character pointer and an address, X. (An address greater than 12 bits requires the computer to be in Extend Mode). The C(Y) are deposited in Memory Buffer, MB and MB bits 0 and 1 are transferred to the Load-Deposit register, LD. The remaining bits specify the address, X, of the location in which the character is deposited. The LD register specifies which third of the memory location receives the character. The character is always deposited from AC bits 0-5 as follows:

LD	Action
01	Deposit AC bits 0-5 in MB 0-5
10	Deposit AC bits 0-5 in MB 6-11
11	Deposit AC bits 0-5 in MB 12-17

At the completion of the instruction the AC is rotated left 6 bits.

If the character pointer is 00, the instruction does nothing.

Index and deposit Character in Memory
dch i Y Oper. Code 15

This instruction is identical to dch Y except the LD register is automatically indexed, and restored in memory location Y. Note, indexing is completed before the character is stored in memory location X. This instruction may be initially entered with 00 in bits 0 and 1 of Y.

After three characters have been deposited in X the character pointer in Y will equal three. When the fourth character is to be stored the pointer is placed in LD via the Memory Buffer, indexed to 01 and the data address indexed to X+1. Thus, each word is indexed modulo three.

Load Character in AC
lch Y Oper. Code 12

This instruction is interpreted as being deferred and, therefore, requires three memory cycles for execution. The C(Y) are deposited in MB, and bits 0 and 1 are transferred to LD. The remaining bits specify the address, X, of the data. LD specifies which character is loaded in AC.

LD	Action
01	Load AC from MB 0-5 and leave in AC 0-5
10	Load AC from MB 6-11 and shift into AC 0-5
11	Load AC from MB 12-17 and shift into AC 0-5

This instruction clears AC bits 6-17 and leaves a single character in AC bits 0-5.

Index and Load Character in AC
lch i Y Oper. Code 13

This instruction is identical to lch Y except the LD register is automatically indexed, and restored in memory location Y. Note, indexing is completed before the character is loaded in AC. This instruction may be entered with 00 in bits 0 and 1 of Y. After the instruction has been executed three times the character pointer in Y will equal three. When the fourth character is to be loaded, the pointer is placed in LD, indexed to 01, and the data address indexed to X+1. Thus the AC receives the fourth character from the left end of location X+1.

Two's Complement Add
tad Y Oper. Code 36

The state of the Link flip-flop is sensed and cleared. If a ONE, one is added to the AC, i.e., +1 to AC bit 17. The C(Y) are then added to the C(AC) and the result is left in the AC. The C(Y) are unchanged. An end-around-carry from bit 0 to bit 17 is always suppressed and the carry out of bit 0 is transferred to the Link flip-flop. Note, "minus zero" i.e., all ones, is not changed to plus zero.

OPERATE GROUP

Complement In-Out Register

cmi Oper. Code 770000
Complements the In-Out Register.

Load Accumulator from In-Out Register

lai Oper. Code 760040
Transfers the C(I-O) to the Accumulator.
The I-O remains unchanged.

Load In-Out Register from Accumulator

lia Oper. Code 760020
Transfers the C(AC) to the In-Out Register.
The AC remains unchanged.

Swap Accumulator and In-Out Register

swp Oper. Code 760060
Exchanges the C(AC) and the C(I-O).

Note: See Operate Group Event Table for detailed sequence of execution of these instructions.

Operate Group Event Table

This table indicates the order in which the instructions in the Operate Group are executed. For completeness, all the instructions in this group have been specified. Note that execution of the instructions lai, lia, and swp is completed in the next cycle, i.e., simultaneously with the next instruction.

Event Time Current Cycle	Instruction	Action
7	cli, cla	
8	clf, stf, lap, lat	
9	cmi, cma, hlt	0 ⟶ run (hlt only)
10	lai, swp	0 ⟶ MB
Event Time Next Cycle		
0	lai, swp	I-O ⟶ MB
1	lai	MB —Jam⟶ AC
1	swp, lia	0 ⟶ I-O
1	lia	AC —Jam⟶ MB
1	swp	AC ⟷Jam MB
2	swp, lia	MB ⟶ I-O

THE SPECIAL OPERATE GROUP

The Special Operate group of instructions is a new set of micro-programmed instructions. It is defined as:

spo = 740000
or
spo i = 750000

The Indirect Address bit equal one has the usual inverse significance when used with the Skip on Zero Link instruction, szl, that is, to skip on non-zero link. However, an unconditional skip will occur if the indirect feature is used with any other instruction in this group.

In conjunction with those instructions affecting the program flags and except where noted, the Ring Mode and Link flip-flops may be considered as two additional flags. Ring Mode is also affected by "erg" and "lrg" and reference should be made to them for further details.

Add to AC from I-O

aai Oper. Code 740003
Execution of this instruction adds the contents of I-O to AC and leaves the result in AC. The I-O is unchanged.
Note that (+0)+(+0)=(+0); all other cases produce -0.

Clear Link

cll Oper. Code 740010
Execution of this instruction clears the Link flip-flop.

Complement Link

cml Oper. Code 740004

Execution of this instruction complements the Link flip-flop.

Index Accumulator

ida Oper. Code 740400

Increases the C(AC) by one. Note 1+(-0)=1; 1+(+0)=1; and 1+(-1)=-0.

Index Character

idc Oper. Code 741CC0

The contents of the Accumulator are considered as a character pointer. Execution of this instruction indexes the pointer and leaves the result in AC.

Thus:

If AC 0,1 = 00 make AC 0,1 = 01

If AC 0,1 = 01 make AC 0,1 = 10

If AC 0,1 = 10 make AC 0,1 = 11

If AC 0,1 = 11 make AC 0,1 = 01 and add 1 to the address part of the AC.

Note: Execution of idc with the AC = -0 yields 400000 rather than 200000. That is, the carry from AC 2 to AC 1 is <u>not</u> suppressed.

Inclusive OR to Flags from I-0

ifi Oper. Code 742000

Execution of this instruction forms the inclusive OR of I-0 bits 10-17 in the program flags. The I-0 register is unchanged.

I-0 10 = 1 ⟶ Link f.f.
I-0 11 = 1 ⟶ Ring Mode f.f.
I-0 12 = 1 ⟶ Program Flag 1
⋮
I-0 17 = 1 ⟶ Program Flag 6

Inclusive OR to AC from I-0

iai Oper. Code 740002

Execution of this instruction forms the inclusive OR of the AC and I-0 in the AC. The I-0 is unchanged.

Inclusive OR to I-0 Flags

iif Oper. Code 744000

Executions of this instruction forms the inclusive OR of the program flags in I-0 bits 10-17. The Program Flags are unchanged.

Link = 1 ⟶ I-0 10
Ring = 1 ⟶ I-0 11
Prog. Flag 1 = 1 ⟶ I-0 12
⋮
Prog. Flag 6 = 1 ⟶ I-0 17

Special Clear Flags

scf Oper. Code 740040

Execution of this instruction clears the six program flags and the Ring Mode flip-flop. Note, the Link flip-flop is not cleared. (See "cll").

Special Clear I-O

sci Oper. Code 740100

Execution of this instruction clears the I-O register.

Special Complement Accumulator

scm Oper. Code 740200

Execution of this instruction complements the AC and adds one to it if Link equals one. The Link flip-flop is then cleared. A carry out of AC bit zero will reset the Link to one.

Skip on Zero Link

szl Oper. Code 740020

The next instruction in sequence is skipped if the Link flip-flop is a zero.

Exclusive OR to AC from I-O

xai Oper. Code 740001

Execution of this instruction forms the exclusive OR of the AC and I-O in the AC. The I-O is unchanged.

Refer to Special Operate Group Event Table for micro-program execution times.

SPECIAL OPERATE GROUP EVENT TABLE

Event Time Current Cycle	Instruction	Action
7	cll	0 → Link
	sci	0 → I-O
	scf	0 → PF 1-6 0 → Ring
	szl	Skip/not skip
8	iif	Link → I-O 10 Ring → I-O 11 PF 1-6 → I-O 12-17
	ifi	I-O 10 → Link I-O 11 → Ring I-O 12-17 → PF 1-6
	scm	Comp. AC; if Link=1, +1 → AC 17, 0 → Link
9a	cml	Comp. Link
	ida	+1 → AC 17
10	idc	Index Char.
	aai ∨ iai ∨ xai	0 → MB
Event Time Next Cycle		
0	aai ∨ iai ∨ xai	I-O → MB
1	aai ∨ xai	MB —XOR→ AC
	iai	MB —IOR→ AC
2	aai	MB —CARRY→ AC

Programming Notes

1. Executing cll ∨ iif transfers the old value of the Link f.f. to I-0.

2. Executing scf ∨ iif transfers the old values of Ring Mode f.f. and the program flags to I-0.

3. Executing cll ∨ scf ∨ iif transfers the old values of Link, Ring and program flags to I-0.

4. Executing cll ∨ szl i will produce a skip if Link were a one.

5. Executing scm ∨ ida in Ring Mode suppresses a carry from AC 15 to AC 14.

6. Executing scm with Link = 1 and AC = 0 is equivalent to a NOP. Link is unchanged.

7. Executing scm ∨ idc performs the ones complement of AC followed by the normal character indexing. For this case the Link bit is not cleared nor does it affect scm.

8. Executing scm ∨ ida performs normal scm but ida carry from bit 0 to bit 17 is suppressed and Link is set instead.

9. Executing scm ∨ ida ∨ idc performs the one's complement of AC followed by indexing AC followed by normal character indexing. Link bit does nothing and is unchanged.

10. Executing scm ∨ aai performs normal scm but aai carry from bit 0 to bit 17 is suppressed and Link is set instead.

11. Executing scm ∨ ida ∨ aai with

Link = 1
AC = 0
I-0 =-0

yields,

Link = 0
AC = 0
I-0 =-0

Since the Link bit acts as a single stage counter, in effect, it has overflowed.

SKIP GROUP

Skip on Non-Zero I-O

sni Oper. Code 644000

If the I-0 register is not equal to plus zero (all bits are zero) the Program Counter is advanced one extra position and the next instruction in the sequence is skipped.

Skip on Zero I-O

sni i Oper. Code 654000

If the I-0 register is equal to plus zero the Program counter is advanced one extra position and the next instruction in the sequence is skipped.

IN-OUR TRANSFER GROUP

Check Status

cks Oper. Code 720033

This instruction, in addition to checking the status of various in-out devices specified in the PDP-1 Manual indicates in or out of Sequence Break Mode and checks the status of the 630 Data Communication System and the Drum System Type 23.

I-O 5 = 1 - PDP-1d Console unlocked
I-O 6 = 1 - In Sequence Break Mode
I-O 16 = 1 - 630 Data Communication System busy
I-O 17 = 1 - Drum System Type 23 busy

Enter Ring Mode

erg Oper. Code 720011

Execution of this instruction sets the Ring Mode flip-flop and the computer operates in the Ring Mode. That is, the carry from AC bits 15 to 14 when executing the following instructions:

dch i
lch i
ida
idc

is suppressed. Thus the pointer will be indexed as shown in these examples:

Leave Ring Mode

lrg Oper. Code 720010

Execution of this instruction clears the Ring Mode flip-flop.

Read Clock

rck Oper. Code 720032

Clears the I-O register and transfers the contents of a 16-bit millisecond counter to I-O bits 2-17.

The clock is counted every millisecond and a sequence break on Channel 17 is generated every 32 milliseconds.

When 1 minute has elapsed a sequence break is generated on Channel 10 and the clock is cleared.

Rename Memory

rnm Oper. Code 72xy66

The term rename is somewhat misleading. There are four 2-bit memory bank pointer registers, the so-called rename registers, r_0, r_1, r_2, r_3.

All addressing is in a sense indirect. Bits 2 and 3 of the address designate which of the rename registers is to be used. The specified rename register in turn specifies which bank of memory is to be used.

The rnm instruction, 72xy66 sets r_x to the value y. y must be 0, 1, 2 or 3.

Reset Memory Bank Selection

rsm Oper. Code 720067

This resets the rename registers to their "normal" state, $r_0 = 00$, $r_1 = 01$, $r_2 = 10$, $r_3 = 11$.

Enter Restricted Mode

erm Oper. Code 720065

Execution of this instruction sets the Restrict Mode FF to one and loads the Protect Memory Register with I-O bits 0-3. If I-O bits 0-3 are zero the computer will be in restricted mode but no memory bank will be protected and the corresponding protect flip-flops will be cleared if previously set.

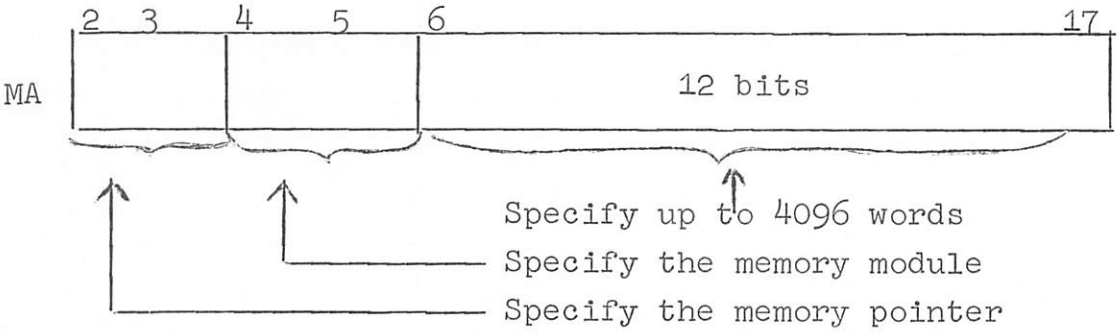
I-O 0=1 - Trap if address bits 2, 3 are 00
I-O 1=1 - Trap if address bits 2, 3 are 01
I-O 2=1 - Trap if address bits 2, 3 are 10
I-O 3=1 - Trap if address bits 2, 3 are 11

A bank of memory refers to a cabinet containing up to 16K words. A memory module refers to 4K words. Thus each bank may have up to 4 memory modules, and a system having a maximum of 65K words would have 4 memory banks and 16 memory modules.

The present system contains three banks of memory:

Bank Zero - 16384 words
Bank One - 4096 words
Bank Two - 4096 words

Memory Banks and Modules are addressed as follows:



When operating in the restricted mode, the following instructions:

An iot instruction
A halt instruction
An incorrect (illegal) operation code
Any instruction addressing through a memory pointer which is protected
A "jmp i" to a non-existent memory

will be trapped and if operating in sequence break mode, a sequence break will occur on Channel 16. Behavior of the restrict system when not in sequence break mode is unspecified.

Leave Restricted Mode

lrm Oper. Code 720064
Execution of this instruction just clears the Restrict Mode FF.

Read Trap Buffer

rtb Oper. Code 720035

Execution of this instruction transfers the contents of the Trap Buffer to the I-O register and then clears the buffer. At this time I-O has the following meaning:

I-O

0=1 - An iot instruction was trapped
1=1 - An illegal instruction was trapped
2=1 - A halt instruction was trapped
3=1 - Jmp i 00x1 or 00x5 (not presently meaningful)
4=1 - Protected memory access attempted
5-17- MB bits 5-17 at time instruction was trapped.

Rem-Rand In

rri Oper. Code 72AA37

Execution of this instruction clears the I-O and transfers the status of the external equipment specified by the AA bits. At this time the I-O has the following meaning.

AA	I-O
00	Data Channel Status
01	Fastrand Controller Status
02	Mag. Tape Controller Status
03	Not Used
04	Not Used

AA	I-O (Continued)
05	Data Channel - Command Address Register
06	Data Channel - Word Count Register
07	Data Channel - Data Address Register
10	Fastrand Unit Status (Location)
11	Fastrand Unit Status (Track)
12	Mag. Tape Unit Read Status
13	Mag. Tape Unit Write Status

Except for the 12 or 13 status test, the information requested is returned within the current instruction time. Detailed status information may be found elsewhere.

Rem-Rand Out

rro Oper. Code 72AA17

Execution of this instruction transfers the contents of the I-O to the external equipment specified by the AA bits. At this time the I-O has the following meaning.

AA	I-O
00	Connect Data Channel to specified controller
01	Initiate Fastrand drum motion in the specified mode.
02	Initiate Magnetic Tape activity in the selected unit.
03	Not Used
04	Not Used

AA I-O (Continued)

05 Initiate Data Channel activity. In normal operation the I-O contains a 16-bit starting address of the first Data Channel command. Execution of this instruction directs the channel to access this command.

Note: The preceding iot output instructions address the Data Channel or the device controllers. Additional selection, i.e., of a device or unit, is specified by the I-O. If the Data Channel or selected controller can accept the output instruction, the next instruction in sequence is skipped.

LITTLE DRUM SYSTEM TYPE 23 INSTRUCTIONS

Drum Break Address

dba Oper. Code 722061

I-0 6-17 - Specify the drum address

When the physicial drum address equals the address in the I-0 a sequence break is made on Channel 5 which must be on. Until the break occurs, the drum may not be asked to do other things.

Drum Initial Address

dia Oper. Code 720061

Execution of this instruction causes the following fields of the I-0 register to be transferred to drum control.

I-0 0 = 1 - Read (0, don't read)
I-0 1 - 5 - Specify drum field to be read (if any)
I-0 6 -17 - Specify the initial drum address for read, write or swap.

Drum Word Count

dwc Oper. Code 720062

Execution of this instruction causes the following fields of the I-0 register to be transferred to drum control.

I-0 0 = 1 - Write (0, don't write)
I-0 1 - 5 - Specify drum field to be written

I-O 6 - 17 - Specify the number of words to be transferred by read, write or swap. 0 means 4096 words.

Drum Core Location

dcl Oper. Code 720063

Execution of this instruction causes the following fields of the I-O register to be transferred to drum control and initiates the transfer of data to or from magnetic core memory. During this time "Drum Busy" is set and is cleared upon completion of the transfer.

I-O 2 - 3 - Specify the physical memory bank. Rename logic is ignored.
I-O 4 - 5 - Specify the memory module.
I-O 6 -17 - Specify the core starting address.

Drum Read Address

dra Oper. Code 722062

This instruction checks the drum status. The current drum address and the error flags are read into the I-O register which has the following meaning:

I-O 0=1 - An error has occurred.
I-O 1=1 - A parity error has occurred during reading.
I-O 2=1 - A transfer error occurred. That is, memory control did not acknowledge a request for a memory cycle.

I-O 3-5 - Not used.
I-O 6-17- The current drum address.

The I-O is not actually set up until 8μs after the end of the instruction. The only instruction concerning the I-O that may follow a dra is dio.

Drum Program Sequence

To initiate a drum transfer the drum instructions must be given in the following sequence

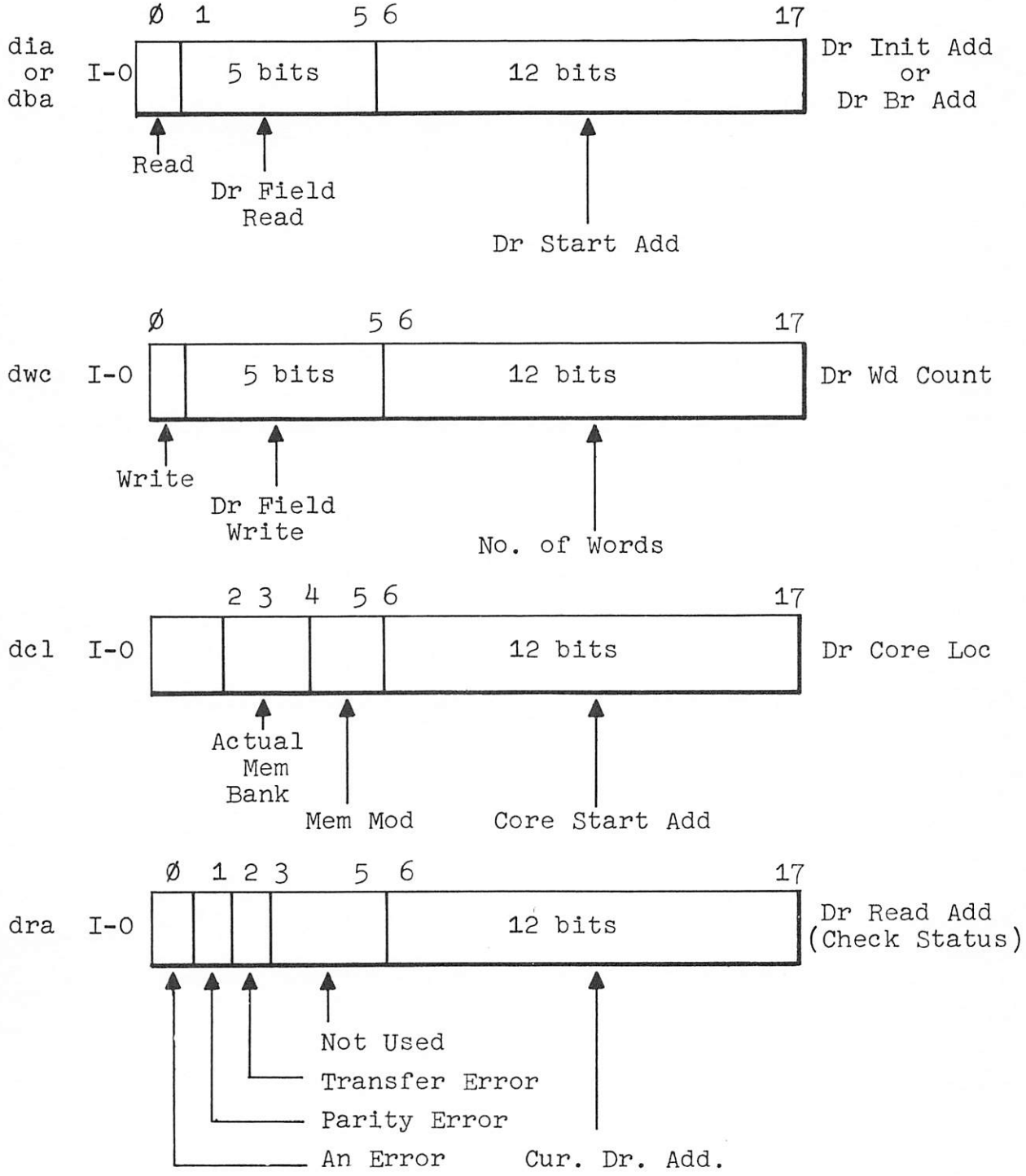
```
dia            dba
 .              .
 .              .
dwc    or      dia
 .              .
 .              .
dcl            dwc
                .
                .
               dcl
```

The transfer of information begins when the current drum address matches the address specified by the "dia" instruction. The minimum time before a transfer can begin is 250 microseconds.

When the specified number of words has been transferred a signal is sent to the Sequence

Break System. An attempt to do something other than dra before the break will lead to unspecified behavior.

The following page summarizes the function of the I-O Register when used with the drum instructions.


Ø 1 5 6 17
dia or dba I-0
5 bits
12 bits
Dr Init Add or Dr Br Add
Read
Dr Field Read
Dr Start Add
Ø 5 6 17
dwc I-0
5 bits
12 bits
Dr Wd Count
Write
Dr Field Write
No. of Words
2 3 4 5 6 17
dcl I-0
12 bits
Dr Core Loc
Actual Mem Bank
Mem Mod
Core Start Add
Ø 1 2 3 5 6 17
dra I-0
12 bits
Dr Read Add (Check Status)
Not Used
Transfer Error
Parity Error
An Error
Cur. Dr. Add.

PROGRAMMING THE DATA COMMUNICATION SYSTEM

A character is received in the following manner:

The receiver turns on a flag at the completion of receiving a Teletype character. The scanner, utilizing a counter, is constantly sampling the receiver flags. When a flag is found on, the counter stops, the scanner flag is turned on and a sequence break on Channel 6 is generated. At this time the number in the counter is the number of the receiver requesting service, and by implication, the station number. As a result of the sequence break the program reads the station number into the I-O register and may or may not release the scanner. If the scanner is not released the program subsequently reads the character from the receiver requesting service to the I-O register, and releases the scanner. Reading the character clears the receiver flag. Releasing the scanner clears the scanner flag.

A character is transmitted in the following manner:

Assume that it is desired to transmit characters to an idle station. Normally, it is the receipt of a character that causes a program break that calls up the Teletype Service Program.

However, to initiate transmission to an idle station some method of boot-strapping must be

employed. For this case, a send buffer is provided which the program loads with a specific station number. The program then issues a transmit instructions which loads the transmitter selected by the number in the send buffer with the character to be sent. The associated receiver flag is cleared at this time. This action proceeds independently of the scanner.

Upon receipt of the character transmitted, the receiver flag is set and when sensed by the scanner a sequence break is generated. The character is received as previously indicated and is "echo" checked.

The transmission of another character to the same station now employs the scanner counter rather than the send buffer. The scanner is released and the receiver flag cleared.

DCS Instructions

The 630 Data Communication System has been assigned one basic iot instruction: 720022, which is microprogrammed to form the following instructions. Adding 002000, i.e., bit 7=1, to the octal equivalent will cause the I-O register to be cleared prior to execution of the instruction.

Receive Character

rch Oper. Code 722022

The I-O is cleared and the character in the selected receiver is put in I-O bits 10-17. The selected receiver flag is cleared.

Receive Character and Release Scanner

rcc Oper. Code 723022

The I-O is cleared and the character in the selected receiver is put in I-O bits 10-17. The selected receiver flag is cleared and the scanner is released (bit 8=1).

Transmit Character via Buffer

tcb Oper. Code 724022

The character in I-O bits 10-17 are sent to the transmitter specified by the send buffer. The associated receiver flag is cleared. The scanner is unaffected. Execution of this instruction initiates the actions necessary to transmit the character serially to the Teletype station.

Transmit Character via Counter

tcc Oper. Code 725022

The character in I-O bits 10-17 are sent to the transmitter specified by the scanner counter. The associated receiver flag is cleared and the scanner released. Execution of this instruction

initiates the actions necessary to transmit the character serially to the Teletype station.

Read Receiver Counter

rrc Oper. Code 722122

The I-O is cleared and the contents of the receiver counter, i.e., scanner, are read to I-O 12-17. The Scanner is not released.

Set Send Buffer

ssb Oper. Code 724122

Transfers the Teletype station number in I-O bits 12-17 to the send buffer. This instruction is used to select an idle station for transmission.

Release Scanner

rsc Oper. Code 721122

Clears the scanner flag and releases the scanner. This instruction may be used in the initialization procedures which should also clear all receiver flags.

Clear Scanner

csc Oper. Code 725122

The scanner flag is cleared and the scanner is reset and released to being examining Group 0.

This instruction may be used to give priority service to 8 high-speed lines after providing service to a predetermined (by program) number of low-speed lines.

Note, scanning proceeds in two phases. The first phase examines each group sequentially, the second phase examines sequentially each line within a group when an active group has been found.

D. The I-O Processor

The following section describes the instructions for the I-O processor and explains its relationship to the Hospital Compuer System as a whole.

DATA CHANNEL PROGRAMMING MANUAL

1. INTRODUCTION

In the Hospital Computer System the Data Channel may be considered as the third of the four processors the system is capable of handling, the other two being the Drum System Type 23 (Little Drum) and the PDP-1d-45 itself, i.e., the Central Processor. (See Figure 1). Independent memories provide simultaneous operation of more than one processor at a time. Requests for the same memory bank or cabinet by two or more processors are honored on a predetermined priority basis as follows.

Little Drum has the highest priority and will lock out any other processor from accessing the same memory until the drum transfer is completed. Next in the priority assignment is the Data Channel. The third and fourth processors will alternate in priority between them on a word-at-a-time basis.

The Data Channel provides for the transfer of data between memory and one of up to eight device controllers. At the present time the I-O subsystem consists of two controllers: a Fastrand Drum Control Unit, FSCU, and a Magnetic Tape Control Unit, MTCU. In turn the FSCU can handle up to four Fastrand Drums, and the MTCU can handle up to sixteen Uniservo III-C units. (See Figure 2.) The present system consists of one Fastrand drum and two tape units. Non-data transfers can be initiated both by the data channel and the Central Pro-

cessor. In normal operation the time it takes to position a selected device prior to initiating a data transfer would not add to processing time, since the Data Channel can be transferring data to one controller while another controller is positioning a device under Central Processor command.

Another level of priority is provided by the 16-channel Sequence Break System. Unlike memory access priority, this priority structure is relevant to the PDP-1d itself and determines the order in which requests for Central Processor access are honored. Completion of a data transfer, successful or non-successful, is made to the Central Processor from the Data Channel via Sequence Break Channel 1. Completion of the positioning of a device is made to the CPU from the selected controller via Sequence Break Channel 4.

Although the Data Channel may operate independently of the CPU, the CPU is required to activate the Data Channel. Also, Channel or Controller status can only be checked by the CPU. From the standpoint of the computer, the Data Channel may be considered as another controller. Thus all computer to "controller" communication involves two basic iot instructions, Rem Rand Out, RRO, and Rem Rand In, RRI. Acceptance of an RRO instruction by a controller advances the Program Counter by one. Except for the interrogation of a Uniservo, acceptance of an RRI instruction is guaranteed within the current instruction time, i.e., within 5.2 microseconds.

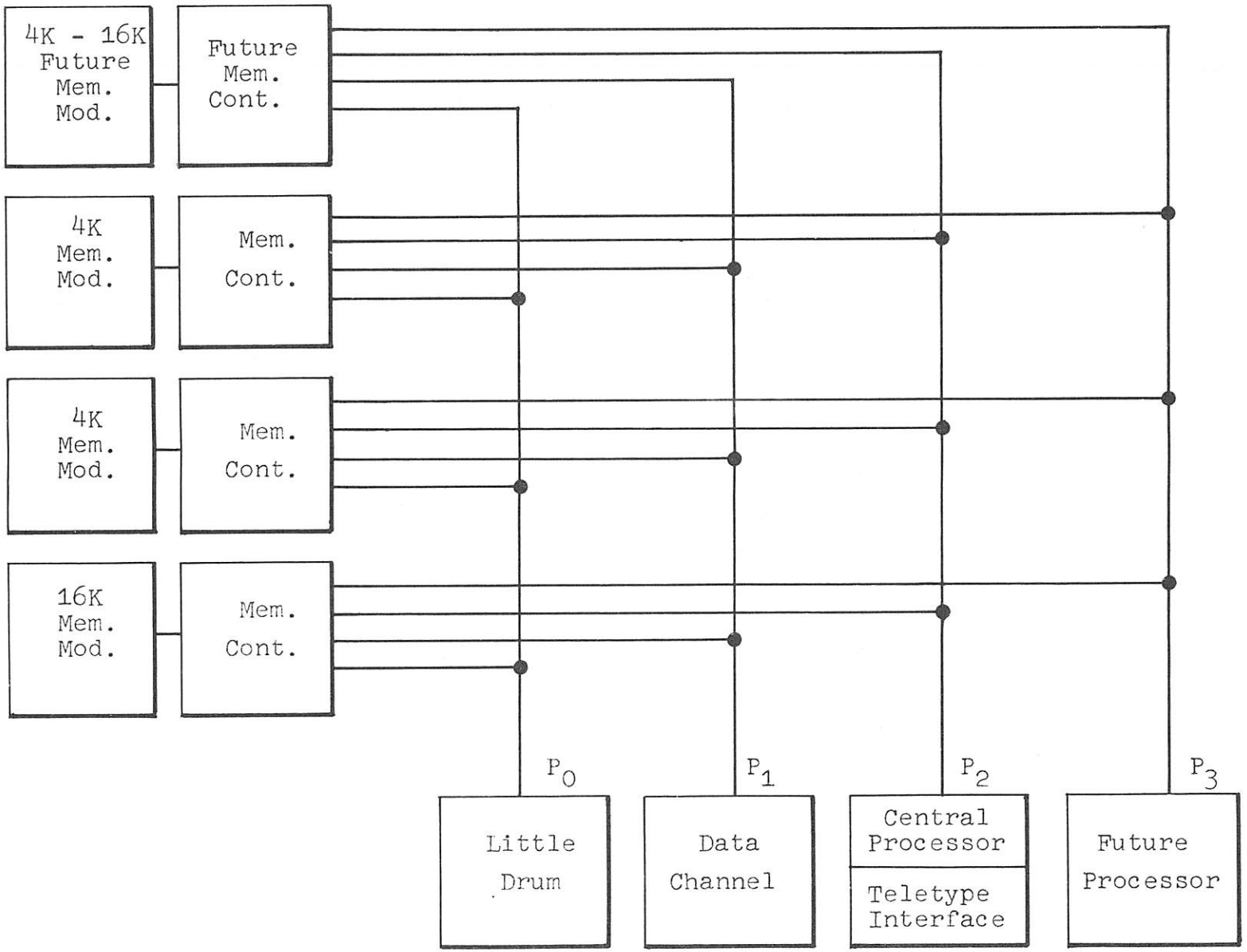


Figure 1. System Block Diagram

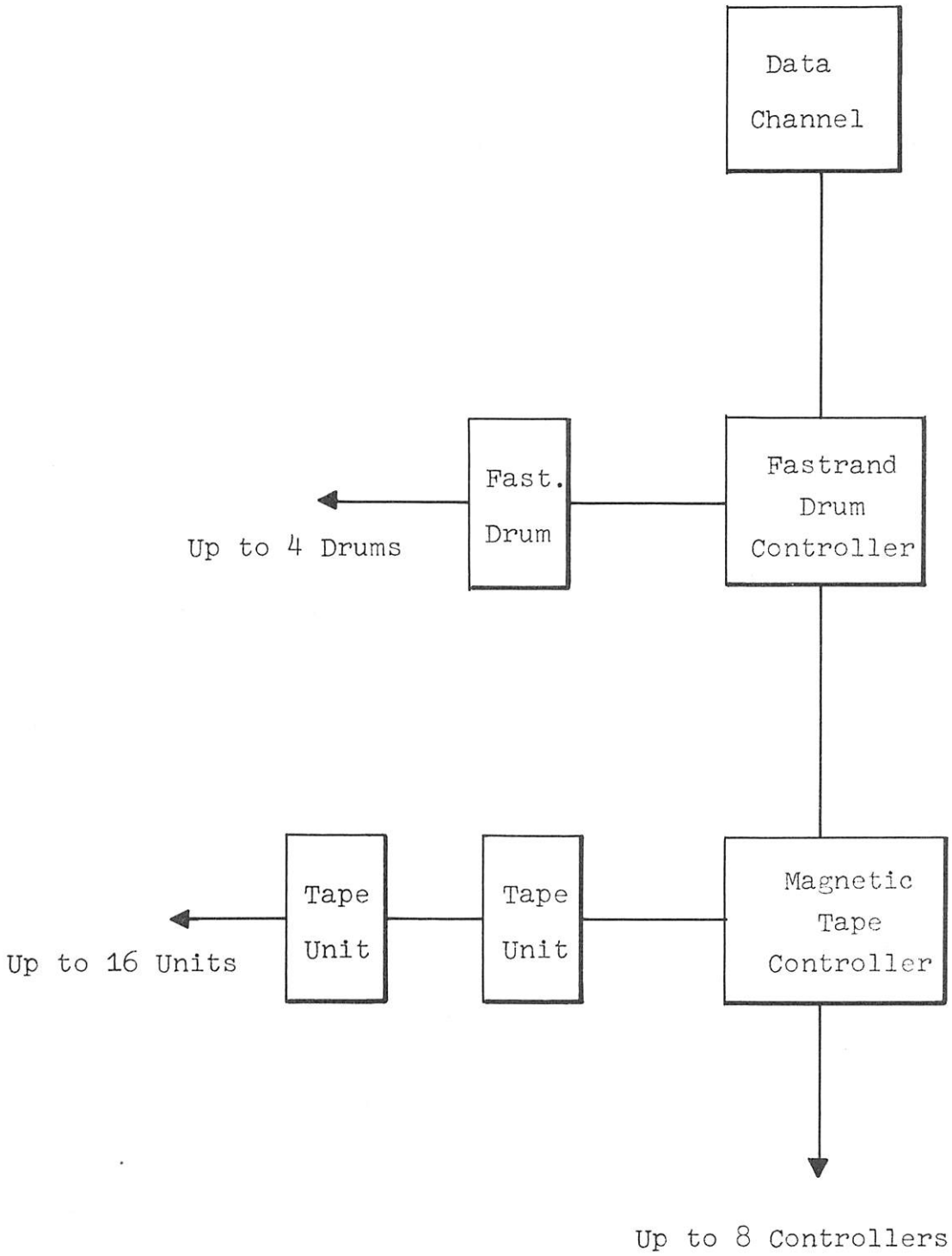


Figure 2. I-O Subsystem Block Diagram

2. DATA CHANNEL

2.1 Introduction

The Data Channel is activated upon receipt of a RRO instruction addressed to it. At this time, the contents of the I-O are sent to the Channel and contain the address of the first Data Channel instruction of the Data Channel Program. Activation of the Channel causes it to access this memory location for its first instruction. Subsequent memory access for commands proceed sequentially and automatically at basic memory cycle rates until an instruction specifying the number of words to be transferred is detected. At this time the Data Channel ceases accessing commands and transfers control to the device controller to which it became connected. (One of the commands in the Channel program connected the Channel to the controller.) The device controller thereupon initiates the data transfer by making successive memory requests, at a rate dictated by the device itself, beginning at the initial address specified for data. (Another command in the Channel program specified an initial data address.) In normal operation the data transfer continues until the specified number of words have been transmitted, at which time control reverts to the Data Channel, and the Channel program is resumed. Unless the Data Channel encounters a special Stop instruction, it will continue to perform in the above described manner.

2.2 Data Channel Hardware

Before proceeding with a detailed description of the instructions, i.e., the Channel and Central Processor instructions, a brief description of the registers and the flip-flops accessible by program is in order. Since the Data Channel transfers 18-bit words to and from memory, it is natural that its registers be basically 18 bits long.

2.2.1 Channel Input Register (CIR)

CIR is loaded with a full 18-bit word every time a memory read request is answered. The main purpose of this register is to obtain a word from memory without disturbing the Channel Output Register (COR). If the channel is in a DATA CYCLE, the word loaded into CIR is immediately transferred to COR where it remains for a device control unit to write the data.

If the channel is in a COMMAND CYCLE, however, the word loaded into CIR may be transferred to any one of four registers. Upon receipt of the word in CIR, the first three bits of the word are decoded and the appropriate action is taken.

2.2.2 Data Address Counter (DAC)

DAC is a 16-bit register loaded from CIR whenever the channel is in a Command Cycle and CIR receives a word whose most significant bits are 11. DAC is used to address future memory cycles for data (as opposed to com-

mands). Its value represents the location in core memory to which, or from which, the next data word is to be transferred.

Each time a new data word is transferred to the channel, DAC is incremented by 1. Although DAC is 16 bits long, it functions as a 12-bit counter, and normally steps from 7777 to 0000. The remaining four high-order bits, i.e., bits 2 to 5, in addition to specifying memory pointer and module may be employed to provide an address of "all ones" which when detected in the channel provide a dummy core location for transfers so that words can be conveniently skipped.

When reading and all ones are detected, the remaining words are read from the device without accessing memory. When writing and all ones are detected, memory access ceases and the balance of the words transmitted to the device control are all zeros. In both cases, DAC is no longer incremented.

2.2.3 Command Address Counter (CAC)

CAC can be loaded from two separate sources. The first is by iot rro. If I-O bits zero and one are zero, acceptance of this instruction by the channel gates into CAC the least significant 16 bits then in the I-O register. The second is from CIR. Whenever the channel is in a Command Cycle and CIR receives a word whose most significant bits are 100, the least significant 14 bits of CIR are transferred to CAC.

Thus, the Data Channel can be directed to access its starting command in any location in memory but cannot be directed to jump from one 16K bank to another. CAC is used to address future memory cycles for commands (as opposed to data). Its value represents the location in core memory from which the next channel command is to be taken.

Each time a new Command word is transferred to the channel via CIR, CAC is incremented by 1.

2.2.4 Channel Output Register (COR)

COR is loaded either from CIR or from the selected device control via the Data In Bus, DIB. It is from this register that words are transferred to memory or to a device control unit. Sometimes the word in COR is interpreted as data and at other times as a command.

If the channel is in a Command Cycle, the word in COR is interpreted as a device command word and a pair of pulses are sent to the selected device control unit to load the Command word into the device control unit. On the other hand, if the channel is in a Data Cycle, the word in COR is interpreted as data and remains in COR for the selected device control or memory to strobe when it needs the next data word.

When data is being transferred from a device control into memory, it will first pass through COR having been transferred there from the Data In Bus. It is this register that provides the necessary storage to synchronize data transfers between a device controller and core memory.

2.2.5 Channel Word Counter (CWC)

CWC is loaded from CIR whenever the channel is in a Command Cycle and CIR receives a word whose most significant bits are 00. CWC is used to keep track of how many data words are to be transferred. Its value represents the number of data words remaining to be transferred.

Each time a new data word is transferred to or from the Channel, CWC is decremented by 1 until it reaches 0. During dummy transfers CWC is decremented by the Device Controller each time a dummy word is transferred. After reaching zero, no more data words are transferred.

Loading CWC with all zeros will cause the channel to stop all future operations and initiate a program break request to the Central Processor.

2.2.6 Continue Control Register (CCR)

CCR is a 2-bit register which is loaded by bits 2 and 3 of CIR under the same conditions as CWC. The value of this register determines when the next command for the channel will be extracted from memory. CCR is cleared each time a request for a new command is made.

2.2.7 Data Direction Flip-Flop (DDF)

DDF is loaded under the same conditions as the CWC by bit 4 of CIR. If a "zero", the data flow between core and Channel (into core memory or out of core memory) will be the same as the flow of data from Channel to device control. If a "one", the data flow between core and Channel will be opposite that of the flow between Channel and device control.

This bit enables the Channel to compare words in memory with those read from the device controller.

2.2.8 Program Break Request Flip-Flop (PBF)

PBF is loaded under the same conditions as CWC by bit 5 of CIR. If the bit is a "one", a program break will be requested when the continue pulse as specified by CCR is generated. PBF is cleared each time a program break request is made.

2.2.9 Channel Status Register (CSR)

CSR is an 18-bit register which provides various status conditions existing in the selected device control unit and the Channel.

2.2.10 Device Connect Register (DCR)

DCR is a 3-bit register which provides the address of the Device Controller to be connected to the Data Channel. It is loaded by a channel instruction and by an iot-output instruction.

2.2.11 Need A Completion Pulse Flip-Flop (NAC)

NAC is set by an iot-output instruction and is significant when the system is operating out of sequence break mode. It provides a completion signal to the Central Processor in lieu of a sequence break.

2.2.12 Error Stop Register (ESR)

ESR is a 4-bit register and specifies the conditions that will produce an error stop.

Note: Since these registers and the channel instructions are intimately related, it is suggested that this section be re-read upon completion of the next section.

2.3 Data Channel Instructions

The Data Channel is started by the Central Processor which sends it a 16-bit command starting address. The Channel proceeds to access commands sequentially until it encounters a Word Count instruction. During this time the Channel is operating in the Command Cycle. When the Word Count instruction is detected, the Channel enters the Data Cycle and is essentially slaved to the controller and its associated peripheral device. Upon completion of the data transfer the Channel returns to the Command Cycle to obtain a new set of command words. Normally, this cycling process continues until the channel encounters a Stop command, at which time the channel becomes not busy but remains connected to the device controller.

Several channel instructions are available that specify the action to be performed by the Data Channel. In a Command Cycle, each time CIR is loaded, the high order bits are examined and the appropriate actions are taken.

A description of the Channel commands follows.

2.3.1 Initialize and Connect the Channel - NLZ

Execution of this instruction loads CIR with the command specified by CAC and performs a decode of the high-order first three bits, i.e., CIR 0, 1, and 2. For this instruction these bits will be $101=5_8$. Bit 3 equal one directs the channel to connect to the controller specified by CIR 15, 16 and 17 and loads the DCR with these bits. Bit 4 equal one permits the error stop flip-flops to be redefined as specified by CIR 9, 10, 11 and 12. Upon completion of this instruction the next sequential command is accessed.

The following is the format of NLZ instruction:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
CIR	1	0	1	a	b	X	X	X	X	E_1	E_2	E_3	E_4	X	X	C_1	C_2	C_3

a = 1: Connect Data Channel to controller specified by C bits.

C_1	C_2	C_3	Connect Channel To
0	0	0	Itself
0	0	1	FSCU
0	1	0	MTCU
0	1	1	Reserved for future controllers
thru			
1	1	1	

Note: The channel connected to itself is used when performing certain diagnostic operations discussed elsewhere.

b=1: Set "error" stop conditions specified by the E bits. Note, the term error is somewhat misleading since not all of the conditions producing a stop are truly errors.

E_1 = 1: Set the Channel's "Ignore Data Errors" FF to one. When operating in this mode, data errors are ignored and do not stop the data transfer.

E_1 = 0: Set the Ignore Data Errors FF to zero. When operating in this mode, data errors are not ignored and cause the Channel to come to an abnormal stop on the word in error.

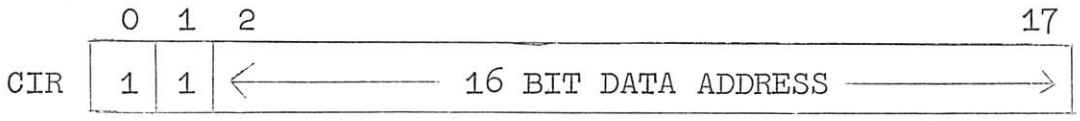
E_2 = 1: Set the Channel's "Stop on Comparison" FF to one. If the channel is operating in the Compare mode (see Section 3.4), the channel will stop when the word count goes to zero if all the words compared.

E_3 = 1: Set the Channel's "Stop on Non-Compare" FF to one. If the channel is operating in the Compare mode, the channel will stop on the word that did not compare.

E_4 = 1: Spare, not used at present time.

2.3.2 Load Data Address - LDA

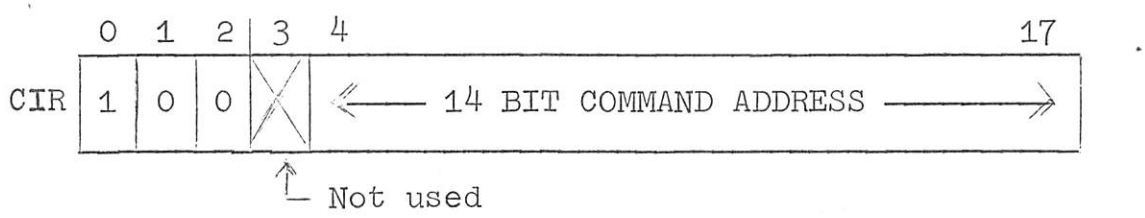
Execution of this instruction loads CIR with the command specified by CAC and performs a decode of the high-order first two bits, i.e., CIR 0 and 1. For this instruction these bits will be 11. The least significant 16 bits of CIR are then transferred to DAC. Upon completion of this instruction the next sequential command is accessed.



Note that although it is possible to specify the starting address of data anywhere in core memory, DAC is a 12-bit counter (bits 6 - 17) and returns to 0000 from 7777. Thus, data may be accessed only within a selected memory module of 4096 words.

2.3.3 Data Channel Jump - DCJ

Execution of this instruction loads CIR with the command specified by CAC and decodes CIR bits 0, 1 and 2. For this instruction they will be $100 = 4_8$. The least significant 14 bits of CIR are transferred to CAC. Upon completion of this instruction the next sequential command is accessed.

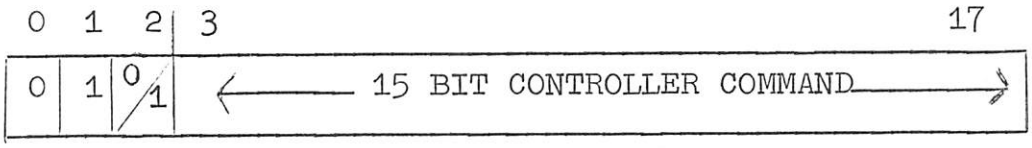


Note: DCJ does not affect CAC bits 2, 3 and therefore restricts the channel program to the previously selected memory pointer..

2.3.4 Load Controller Command - LCC

Execution of this instruction loads CIR with the command specified by CAC and decodes CIR bits 0 and 1. For this instruction they will be 01. The contents of CIR are transferred to COR and the command part is sent to the controller

to which the Data Channel is connected. The controller is given 1 millisecond to accept the instruction. If the acceptance is received within this time, the next sequential channel command is accessed. If the time-out is exceeded, the channel halts with the "Controller Abnormal" bit of the Channel Status register set. (See Section 4.2 for discussion of I-O subsystem status conditions.)



Note 1: COR bit 2 specifies to the controller whether the command is a Type 1 or Type 2 command, and is meaningful only to the Fastrand Drum Controller.

COR 2 = 0 - Type 1
COR 2 = 1 - Type 2

See Section 2.4 for details of the controller commands.

Note 2: Acceptance of a device controller command by the Data Channel clears the error or abnormal conditions of the channel but not the Error Stop conditions.

2.3.5 Load Word Count Command - Introduction

Available to the programmer is another instruction which provides the channel with a word count, enables the programmer to control its mode of operation and starts the data transfer, i.e., the "Load Word Count and Control" command.

As mentioned previously, whenever the Channel halts, an unconditional program break is returned to the CPU over Sequence Break Channel #1. However, the programmer may wish to program sequence breaks at known times within the body of the channel program. PBF provides this capability, and if a one a program break will be generated when the next command is accessed.

The programmer may wish to put the Data Channel in a search or compare mode. When DDF is a one and the Channel is reading data from a device, the direction of the data transfer between core memory and the Channel is reversed, i.e., from write to read, and successive words from memory are compared with the corresponding word from the device.

Note, the decision to stop or not to stop as a result of the comparison is determined by the Error Stop conditions established by the NLZ instruction.

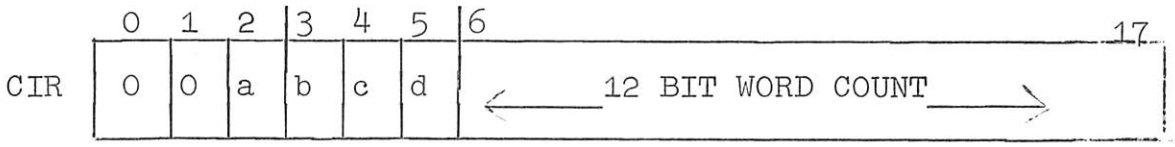
The programmer may also wish to define the point in the transfer of data that the Channel should access the next command. Among other things this feature enables Scatter-Gather and non-stop mag. tape operations. The state of the Continue Control register, CCR, determines this mode of channel operation.

2.3.6 Load Word Count and Control - LWC

Execution of this instruction loads CIR with the command specified by CAC and decodes CIR bits 0 and 1. For this instruction they will be 00. The contents of CIR are distributed to four registers:

CIR 2, 3 ⟶ CCR
CIR 4 ⟶ DDF
CIR 5 ⟶ PBF
CIR 6-17 ⟶ CWC

and the Data Channel enters the Data Cycle and initiates the data transfer. At the completion of the transfer, as defined by CCR, the Channel returns to the Command Cycle and resumes accessing commands.



a	b	Point in Process Next Command is Accessed
0	0	Man. Adv: A Push Button Must Be Depressed
0	1	WC→ 0 : Word Count Equals Zero
1	0	EOR : End of Record
1	1	Ready : At Rest

c = 1: Places Data Channel in comparison mode, i.e., sets DDF.

d = 1: Requires that a program break be given when the next channel command is accessed, i.e., sets PBF.

Note 1: EOR and Ready are logically the same for the Fastrand For Magnetic Tape, Ready means the tape has come to a complete stop; EOR means the End of Record has been reached but the tape is still moving.

Note 2: The significance of continuing when the Word Count goes to zero depends upon whether the data transfer is a read or write operation. (Read from device; write on device.)

When reading: If the number of words of the record is less than the number of words to be read, the next record will be read.

If the number of words of the record is greater than the Word Count, then a Scatter Read is intended and at WC —>0 time the two commands required are a new data address and a new Word Count.

When writing: If Continue is defined at WC —> 0 a Gather Write is intended and at WC ——> 0 time a new data address and a new Word Count are required.

Note 3: Writing ceases when the Word Count is counted down to zero. This cessation, when writing on magnetic tape, itself produces an EOR. Therefore, it is possible to perform a non-stop tape write operation by continuing on EOR.

For the Fastrand drum, EOR is independent of reading or writing and is equivalent to READY when transferring data. When seeking a track, EOR is not meaningful, therefore, "Continue on Ready" should be used.

2.3.7 Channel Stop - CST

Execution of this instruction loads CIR with the command specified by CAC and decodes CIR bits 0 and 1. For this instruction they will be 00. The Data Channel enters the Data Cycle with Run and Busy FF's reset, i.e., the Channel halts. (See Channel Status Register for significance of Run and Busy.) A minimum of 15 zeros is required to cause a channel halt, although for programming ease it may be defined as an LWC of all zeros.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
CIR	0	0	0	b	c	d	0	0	0	0	0	0	0	0	0	0	0	0

Note 1: If b, c, d are ones, the Continue Register will be set to 01, i.e., continue on WC —>, DDF will be set, and PBF will be set and then reset. However, no action occurs as a result of setting these conditions.

Note 2: CWC is cleared to all zeros.

2.3.8 Dummy Word Count

When the Data Channel is used to seek a track, or perform a non-data tape operation, a dummy word count is necessary to establish the Continue definition, set Compare mode, and request a program break. The latter two conditions are optional but the Continue condition must be set to Ready.

The dummy word count is given after LCC and allows the Channel to continue the channel program. The dummy word count may also be used, when performing a multi-sector operation or a non-stop magnetic tape data transfer, to provide a "Continue on EOR" condition to allow the Channel to accept a new device command. For this case, the dummy word count should immediately follow the LWC command.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
CIR	0	0	1	1	c	d	0	0	0	0	0	0	0	0	0	0	0	0

Note 1: Setting Fastrand Mode by Data Channel by itself, i.e., without seeking a track, hangs up the Data Channel even with the proper Continue condition established.

2.4 Device Controller Commands

The process of activating a peripheral device, i.e., a Fastrand drum or Uniservo III-C, requires four levels of control. First, the Data Channel must be activated by the PDP-1d and instructed where it is to find its first command. Second, the Data Channel must be given sufficient information about how it is to operate, which controller to connect to, how many words to transfer and the starting address of data in memory. Third, sufficient information must be given a selected controller to enable it to select a device, determine its mode of operation, activate it, and control it during the data transfer. Fourth, the Data Channel must activate the controller and thereby initiate the data transfer. With the exception of the controller commands, these facts have been previously set forth.

Before proceeding with the details of the Device Controller commands, a brief summary of the more important features of the Fastrand and Uniservo III-C will be given.

2.4.1 Fastrand Drum Features

The Fastrand Drum is actually two drums or cylinders each revolving asynchronously to each other at approximately 880 revolutions per minute. The total capacity of the Fastrand is roughly 450 million bits which are recorded serially in tracks around the cylinders at 1000 bits per inch. At a 1.1 megacycle rate, this gives a bit separation of 0.9 microseconds, which for an 18-bit word, means that words are available from the drum every 16.2 micro-

seconds. This is the basic data transfer rate of significance to the programmer. Random access to each cylinder is accomplished by a set of 32 movable heads, for a total of 64, mounted on a boom suspended between the upper and lower cylinders. Each head can be moved to one of 96 individual tracks. The process of moving the boom is called "seeking." When seeking, all 64 heads move together and come to rest at the same relative position within its set of 96 tracks. The time to position the boom is a function of the number of tracks to be covered. However, the average boom positioning time is approximately 92 milliseconds. This is in comparison to 20 microseconds, the time to switch heads. Thus, it is desirable to retain related information within the area covered by a boom position, i.e., a "tiny" or "logical" drum. See Figures 3 and 4.

Information is stored on tracks in sectors. There are 64 sectors per track each containing 51 words. Within each track each sector is addressable and, therefore, has a sector address which precedes the 51 words. Of these, the first word is a "tag" word which, for list structuring, points to the next free sector within a "logical" drum. Thus a tag word consists of head and sector information.

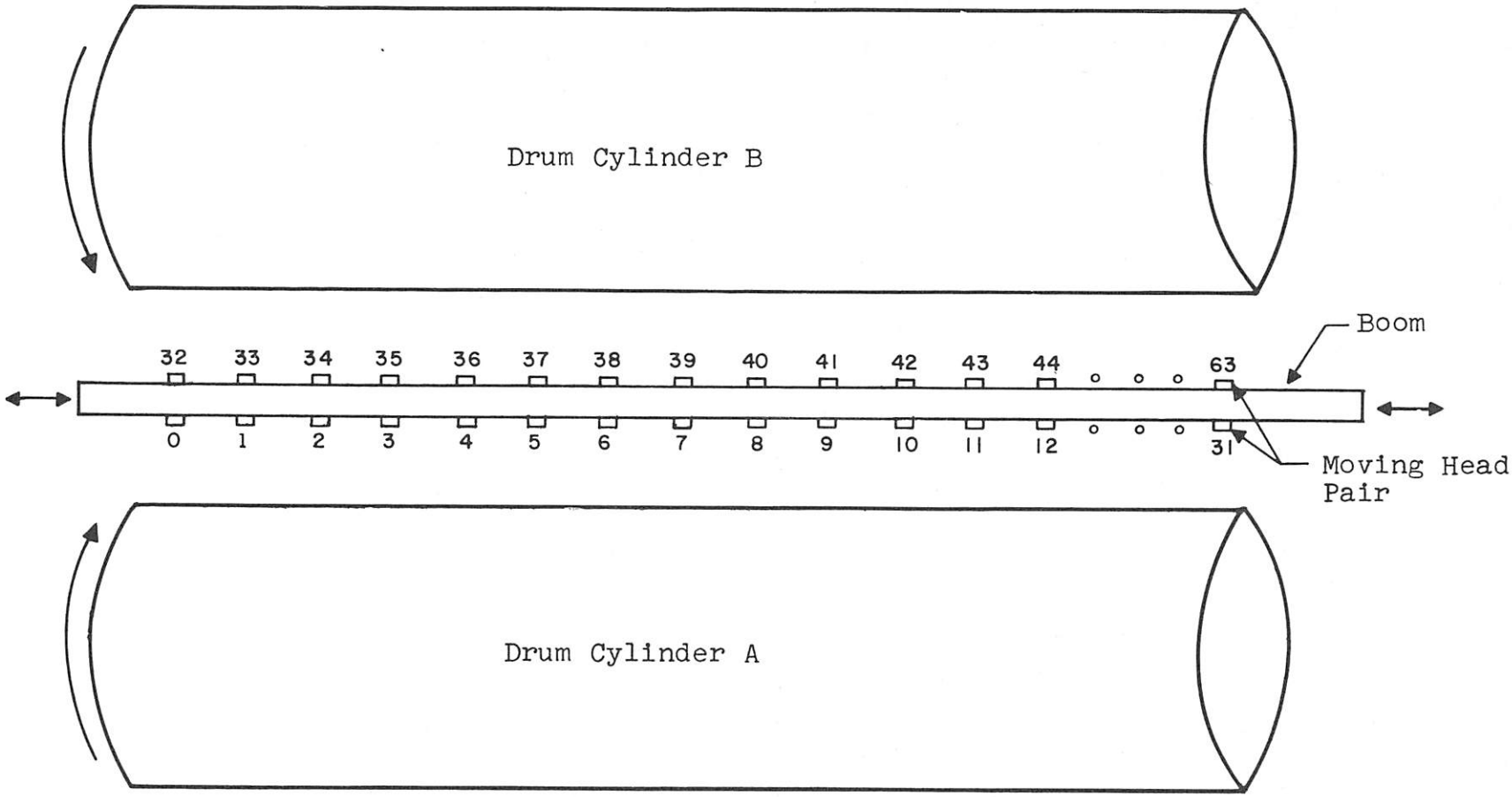


Figure 3. Fastrand Drum Schematic 1

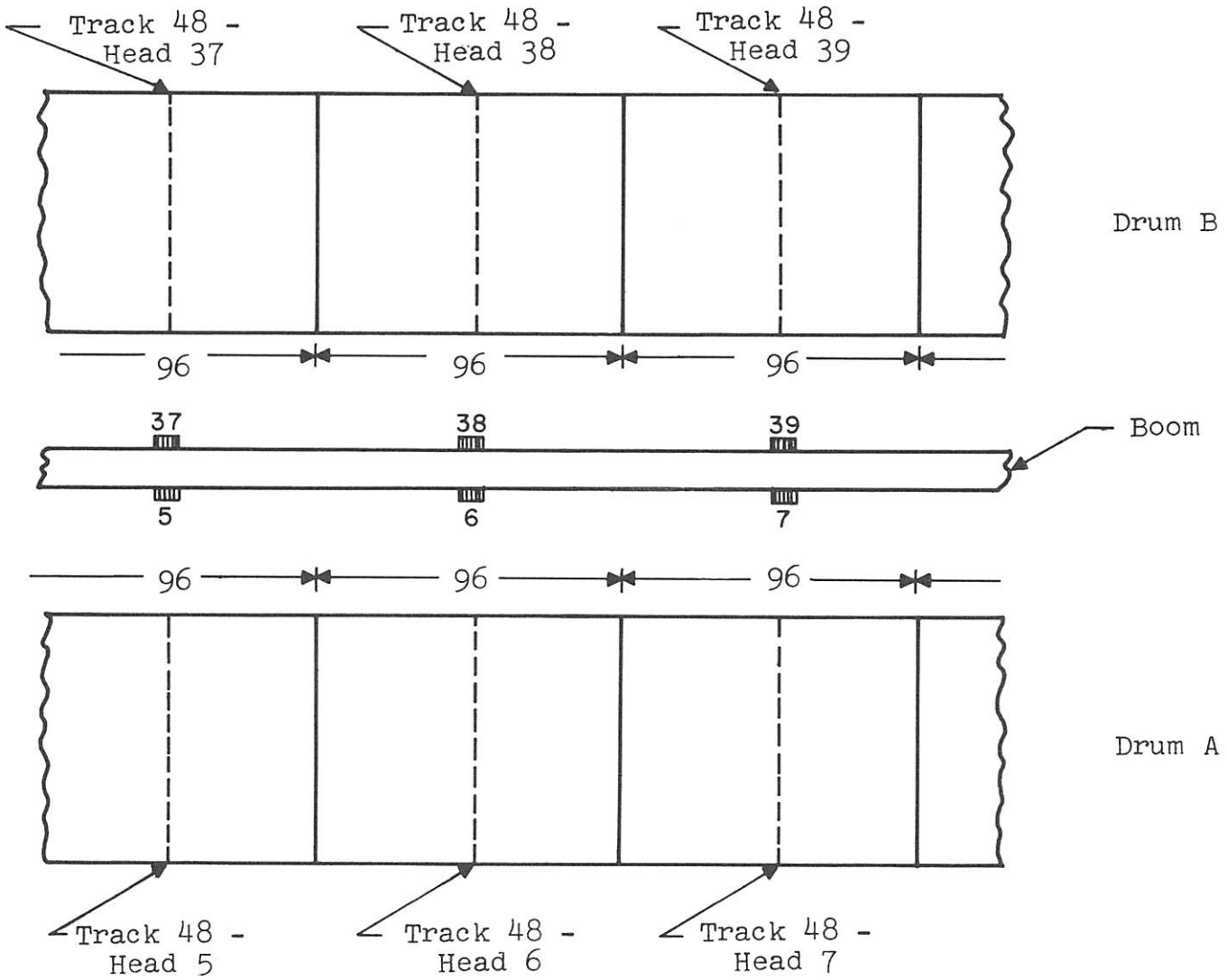


Figure 4. Fastrand Drum Schematic 2

The remaining 50 words are data and are physically separated from the tag but are contiguous to each other. Thus, the amount of data that can be stored within a single "tiny" drum can be computed as follows:

64 sectors per track x 50 words per sector = 3200 words of data per track.

3200 words per track x 32 heads per cylinder - 102,400 words per boom position per cylinder.

102,400 words x 2 = 204,800 words per tiny drum.

In terms of core loads, i.e., groups of 4096 words, this number represents the equivalent of 50 core loads.

Thus, the total drum capacity in terms of core loads of data is:

50 core loads per T.D. x 96 T.D. per cylinder x 2 cylinders = 9600 core loads.

In addition to the 64 movable heads, there are 8 fixed heads, 7 of which are available for storing information and one is used for test purposes.

Thus, to perform a drum operation the programmer must seek a track on a selected unit, specify the head and sector, and designate one of four combinations of reading or writing tag and data.

2.4.2 Fastrand Controller Commands

Normally, seeking a track is done by a computer iot followed by Data Channel operation of the drum. However, since it is possible for the data Channel to perform both seeking and data transfers, two types of Device Controller commands are provided. Type 1 enables the programmer to select a head and sector and designate reading or writing. A Type 2 command may be given to seek a track on a specified unit and set the mode of operation.

A detailed description of their formats follows:

Type 1 FSCU Command

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
COR	0	1	0	a	b	H_f	H_b	H_4	H_3	H_2	H_1	H_0	S_5	S_4	S_3	S_2	S_1	S_0

a	b	ACTION
0	0	Read Tag: Read Data : RT/RD
0	1	Read Tag: Write Data : RT/WD
1	0	Write Tag; Read Data : WT/RD
1	1	Write Tag; Write Data : WT/WD

Note 1: The S bits specify up to 64 sectors, i.e., $(00)_8$ to $(77)_8$.

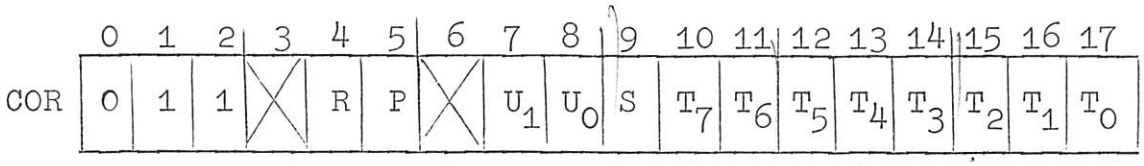
Note 2: The H bits, H_4 - H_0, specify up to 32 movable heads, i.e., $(00)_8$ to $(37)_8$.

Note 3: $H_f = 0$: Specifies that selected head is movable.
$H_f = 1$: Specifies that selected head is fixed.
$H_b = 0$: Specifies that lower bank of 32 heads be used (i.e., those associated with Drum Cylinder "A").
$H_b = 1$: Specifies that upper bank of 32 heads be used (i.e., those associated with Drum Cylinder "B").

Note 4: When a fixed head is designated, i.e., $H_f = 1$, bits H_b H_4 H_3 specify which head. At this time the remaining H bits, H_2 H_1 H_0, should be zero.

Note 5: Normal drum usage is to Write Tag/Read Data or Read Tag/Write Data on a single sector basis. When these functions are performed on a multisector basis the tag word of succeeding sectors is not transmitted to the Data Channel for RT/WD, nor is it written when performing a WT/RD. When it is desired to write or read all the tag words, i.e., for a drum dump or when a track contains program information rather than list structured data, WT/WD or RT/RD may be used.

Type 2 FSCU Command



Note 1: R = 1: Set Recovery Mode.
P = 1: Set Prep Mode.
R = 0 and P = 0: Normal Mode.

Note 2: The U bits select one of four Fastrand drum units. Selection of a non-existent unit produces an abnormal condition.

Note 3: The S bit is the Seek bit.
S = 0: Do not seek track.
S = 1: Seek track.

Note 4: The T bits are the track selection bits. At the present time seven bits, T_6 - T_0, are used to select up to 96 tracks. T_7 is reserved for Fastrand II which will have twice the capacity of Fastrand I.

2.4.3 Uniservo III-C Features

The Uniservo III-C is Remington Rand's magnetic tape handler designed to be compatible with IBM tape. A brief description of the tape format and the significant features of the Uniservo III-C are presented.

The magnetic tape is 1/2-inch wide, has a MYLAR base, and is oxide coated on one side.

Information is recorded on the coated side of the tape in variable-length records, separated by 3/4-inch record gaps. See Figure 5. The length of each record depends on the number of frames or characters it contains. Each frame contains seven positions, or channels, across the width of the tape in which 1-bits or 0-bits are recorded. Information bits are recorded in channels 1 through 6 and a parity bit is recorded in channel 7. When operating in even parity mode, the parity bit is a 1 if the number of ones recorded in channels 1 - 6 is odd; the parity bit is 0 if this number is even. When operating in odd parity mode, the reverse is true.

Following the last frame in a record and separated by a gap of three frames from it, a longitudinal check or parity frame is recorded. In this frame a 1-bit is recorded in any channel in which the sum of recorded 1-bits (for that record) is odd; thus even longitudinal parity on the tape is ensured. Recognition of the longitudinal parity frame marks the time from which the End Of Record, EOR, is computed.

The separation between frames and the width of the gap depend on the recording density. For low recording density, i.e., 200 bits per inch, the interframe gap is 45 microseconds. For high recording density, i.e., 556 bits per inch, the interframe gap is 16 microseconds.

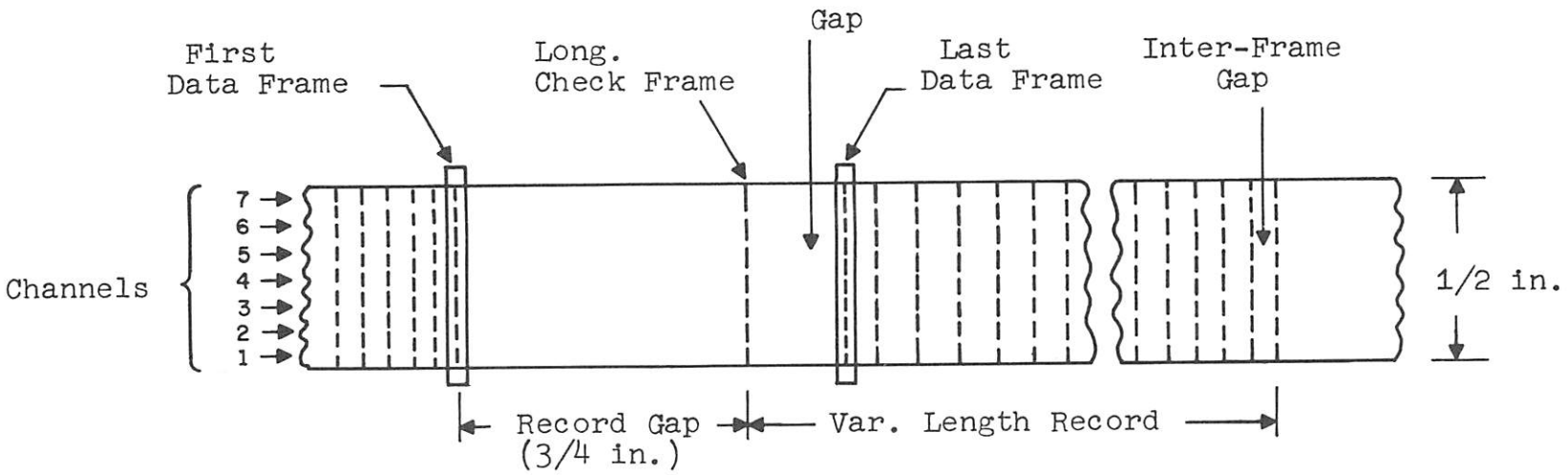


Figure 5. Magnetic Tape Format

Load-point and end-of-tape warning markers, which are aluminum strips on the tape, cause signals to be produced which indicate tape position during read, write, or rewind operations. When either marker is detected, a signal is produced. The load-point marker is 10 feet from the beginning of the tape. When this marker is detected the tape controller is alerted that the tape handler is in the first-block condition; that is, the tape handler is ready to perform a read or write operation, and the first frame area of the first tape block is under the read-write head. The end-of-tape warning marker is 14 feet from the end of tape. When this marker is detected, the tape controller is alerted that enough tape remains to write one more record of not more than 2000 characters.

Under control of the MTCU, the primary functions of the Uniservo III-C tape handler are to transport magnetic tape past a 7-channel read-write head, erase from and record onto tape digital information stored on the tape, and transfer it to the controller.

It moves tape, either forward or backward, at a speed of 112.5 inches per second, and rewinds it at 360 inches per second. The time to rewind a full reel of tape is 80 seconds. The time to start or stop tape motion is 3 milliseconds. The time to position the tape loops for forward or backward motion is 600 milliseconds. When operating in low density mode, the character rate of the tape handler is 22,500 characters per second; in high density mode, the rate is 62,500 characters per second.

2.4.4 Magnetic Tape Controller Command

The details of the Mag. Tape Controller Command follow. Note that upon receiving this command the MTCU decodes its various fields to instruct the tape handler to perform the specified operation.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
COR	0	1	0	C_4	C_3	C_2	C_1	C_0	B_5	B_4	B_3	B_2	B_1	B_0	U_3	U_2	U_1	U_0

C_4	C_3	C_2	C_1	C_0	ACTION
0	0	0	0	0	No Operation
0	0	0	1	0/1	Position Tape Loops for Forward/Backward
0	0	1	0	0/1	Rewind No Interlock / With Interlock
0	0	1	1	1	Erase Tape (5 inches)
0	1	0	0/1	0/1	Write Even Parity; High Dens. / Odd Parity; Low Dens.
0	1	1	0	0/1	Write End of File High Dens. / Low Dens.
1	0	0	0/1	0/1	Read Even Parity; High Dens. / Odd Parity; Low Dens.
1	1	0	0/1	0/1	Space For. "B" Blocks EP; HD / OP; LD
1	1	1	0/1	0/1	Space Back. "B" Blocks EP; HD / OP; LD

Note 1: NOP may be used to select a Uniservo and set the block count.

Note 2: Spacing forward checks parity; spacing backward does not.

Note 3: The B bits specify the number of blocks to be spaced. The maximum number is 63. Spacing zero blocks is equivalent to a NOP.

Note 4: The U bits specify which tape unit is to be selected. Selection of a non-existent unit produces an abnormal condition.

Note 5: There is no Type 2 MTCU command.

3. Data Channel Programming

3.1 Data Channel Command Sequence

A programmer may employ several methods of operating the I-O sybsystem. The following sections discuss some of these and point out their relevant features.

In setting up a Data Channel program a typical sequence of instructions could be:

NLZ - Connect channel to specified controller and set initial conditions.

LDA - Load the initial data address.

LCC - Load device controller with device command.

LWC - Load channel with word count and controls.

CST - Stop.

When instructed by the PDP-1d to start, i.e., where to access its first command, the channel automatically and sequentially accesses the first three commands and waits for an acceptance of the device command from the connected controller before accessing the fourth. Execution of this instruction causes the Data Channel to leave the Command Cycle and enter the Data Cycle in which it remains until the completion of the data transfer. At this time the Channel returns to the Command Cycle, accesses the fifth command, decodes it, stops

further command access, and unconditionally sends a sequence or program break over break channel #1 to the PDP-1d.

If the programmer desired to perform additional data transfers following the initial one, the CST could be replaced by another sequence of commands, or by a jump (DCJ) to another sequence of instructions.

3.2 Scatter-Gather

Scatter-Gather operations are enabled by the "Continue on Word Count→0" condition and require that the Data Channel be given a new word count and a new data address at the proper time.

Gather-Write writes information from different areas of storage in a single contiguous block or record. Scatter-Read reads information from a single area of storage and write it in segments in different storage areas.

The following Data Channel program does a Gather-Write on a single Fastrand drum sector, picking up the tag word from one area of core storage and the 50 data words from another. (It is assumed the track has already been selected.)

```
NLZ : 540001   /connect channel to FSCU
LDA : 601000   /set up data address of tag word
LCC : 260101   /write tag, write data, head 01, sector 01
LWC : 040001   /word count = 1, cont. on WC→0
LDA : 601100   /set up data address of data words
```

```
LWC : 100062    /word count = 50, cont. on EOR
CST : 000000    /halt
```

The next program does a Scatter-Read of the same sector.

```
NLZ : 540001    /connect channel to FSCU
LDA : 602000    /set up data address of tag word
LCC : 200101    /read tag, read data, head 01, sector 01
LWC : 040001    /word count = 1, cont. on WC→0
LDA : 602100    /set up data address of data words
LWC : 100062    /word count = 50, cont. on EOR
CST : 000000    /halt
```

3.3 Skip Mode

There may be times when it is desired to skip over a known number of words and then resume the data transfer. To accomplish this the programmer should load DAC with all ones and set the word count equal to the number of words to be skipped. If Continue is defined as word count going to zero, it is intended to read or write after skipping the desired number of words. Thus, a new data address and a new word count should follow the "Continue on WC→0" word count instruction.

If it is desired to read the first part of data and skip over the balance of the record or sector, DAC should first be set to the initial core location for data and the word count should equal the number of words to be read with "Continue on WC→0" specified. When the channel resumes accessing

commands, the next connamds should load DAC with all ones and set the word count equal to the balance of the data, with "Continue on EOR" defined. Thus, by the proper choice of the continue definition the programmer can select which part or parts of a record or sector he wishes to skip.

3.4 Data Compare Mode

When the Channel's Data Direction flip-flop is set to one the Channel operates in the Data Compare Mode, and words read from a device are compared to words read from magnetic core storage. The programmer has the option of stopping the Data Channel when a word read from a device does not compare with the corresponding word in memory, or of stopping the Channel when a predetermined number of words read match the corresponding number of words in memory. For the former case the "Stop on Non-Compare" flip-flop should be set to one; for the latter case the "Stop on Comparison" flip-flop should be set to one.

When the Data Channel stops because the data compared (or did not compare as the case may be), further transmission of data between the controller and the Channelceases but the Channel remains busy until it receives an End of Record, EOR, from the controller.

If the Program Break flip-flop is set to one, a Channel 1 sequence break will be made at the earliest when the Data Channel stops abnormally. By checking the Channel Status Register (see Section 4.3.1) the programmer can determine

that the abnormal stop was indeed caused by a comparison "error." The earliest a break can occur is when the continue pulse is generated and continue is defined as WC→0."

If the Program Break flip-flop is not set, a Channel 1 sequence break will be made when the Data Channel becomes not busy, i.e., at the end of the record. This is the latest in time a Channel 1 break can occur.

Thus, if it were desired to compare the first six words of a block of data and receive a program break at the earliest point in time, the NLZ instruction should set "Stop on Compare", LDA should specify the initial core location of the words to be compared, LWC should specify six words, set PBF and DDF, and define continue as "WC→0."

This feature of continuing on WC→0 may be employed to permit reading the remaining words of the sector or record without waiting one revolution for the sector or backspacing one block, respectively. This will occur if the next two commands (following the LWC) are LDA and LWC, where LDA establishes the initial core location for data and LWC sets up a word count equal to or greater than the remaining number of words to be read, resets DDF, set PBF and specifies "Continue on EOR." Thus, by the time the second sequence break is generated a decision will have been made either to accept the data or the channel program will have been modified to permit a search of the next sector or record.

3.5 Illegal Combinations

Certain combinations of Data Channel Instructions are illegal and, if given, will cause an abnormal condition to occur. (A detailed discussion of abnormal conditions may be found in the section on Status.)

1. If the Data Channel is used to perform non-data transfer operations (instead of using the Central Processor), and the Channel is instructed to resume or continue on EOR instead of READY, the Channel hangs up.

These operations are:

a. Erase
b. Rewind
c. NOP
d. Position Loops
e. Space Zero Blocks
f. Seek

2. If the following channel commands are given within a record interval, i.e., prior to EOR, the Channel comes to an abnormal halt.

a. NLZ
b. LCC
c. CST

3. If the Data Channel is used to establish Fastrand mode without seeking a track, i.e., S bit = 0, the Channel hangs up.

4. iot Instructions

In addition to the Data Channel instructions, the programmer has available to him two basic iot instructions, Rem Rand Out, RRO; and Rem Rand In, RRI. They may be employed to perform the following functions:

RRO

1. Connect the Data Channel
2. Start the Data Channel
3. Set the Fastrand drum mode
4. Seek a drum track
5. Set magnetic tape mode
6. Move magnetic tape
7. Set diagnostic mode

RRI

1. Check I-O Subsystem status

Note: Item 7 refers to a special mode of operation used primarily for test purposes. For completeness, the functions performed in this mode are enumerated but not set forth in detail.

4.1 Rem Rand Out

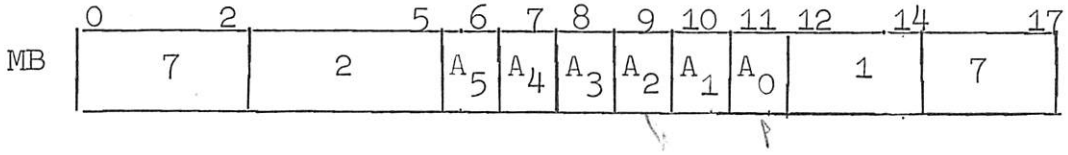
rro Oper. Code 72AA17

Execution of this instruction transfers the contents of the I-O register to the selected "controller." Acceptance of this instruction steps the computer's program counter causing the next computer instruction

to be skipped. The AA bits select the "controller" and the two most significant bits of I-O determine its response. The remaining 16 bits of I-O convey additional information meaningful to the selected controller.

Bit zero of the I-O register determines whether or not the instruction is to be accepted unconditionally. If a one, the instruction is unconditionally accepted. If a zero, acceptance is conditional upon the controller being not busy and available. In normal operation, bit zero is zero. The unconditional acceptance mode is used for maintenance and diagnostic purposes.

The Central Processor is informed of the end of the operation by receiving a completion pulse from the selected controller. When operating in sequence break mode, the completion pulse will be returned to Sequence Break Channel #4. When operating out of sequence break, the completion pulse will be sent to the Central Processor if the "controller's" NAC flip-flop is set.



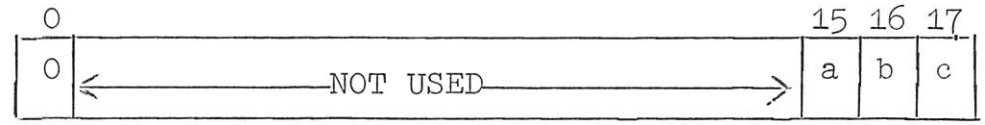
When operating in "Need-A-Completion" mode bits 5 and 6 of the Memory Buffer, MB, are significant. When opposite in value they signify a request for a completion pulse from the selected controller, and set its NAC flip-flop.

Note, normal operation is sequence break mode. The NAC mode is rarely used. Thus A_5 will always be zero.

The significance of the I-O for the various values of the "A" field when RRO is given, follows:

4.1.1 Connect Data Channel AA=00:

Unlike all other iot output instructions, Connect Data Channel can be accepted only if the Channel is available and not busy. Acceptance of the instruction steps the PC and loads the DCR with I-O bits 15, 16 and 17.



a	b	c	Connect Channel To
0	0	0	Itself for Diagnostic Purpose
0	0	1	Fastrand Control Unit (FSCU)
0	1	0	Uniservo Control Unit (MTCU)
0 1 to 1 1	1 1 to 1 1	1 1 to 1 1	Reserved for future controllers

4.1.2 Initiate Drum Activity AA=01:

When I-O bit 0 is zero, this instruction can be accepted only if the FSCU is available, not busy, and is not connected to the busy Data Channel. When I-O bit 0 is a one, this instruction is unconditionally accepted. Acceptance steps the PC. The meaning of the I-O is dependent upon bit 0.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
I-O	0	X	X	X	R	P	H_b	U_1	U_0	S	T_7	T_6	T_5	T_4	T_3	T_2	T_1	T_0

Note 1: R=1: Set Recovery Mode

P=1: Set Prep Mode

R=0 and P=0
or
*R=1 and P=1
: Normal Mode

* Returned as 00 when checking status.

Note 2: H_b specifies which drum cylinder, A or B, is selected when interrogating the drum for its current sector address.

Note 3: U_1U_0 select the unit.

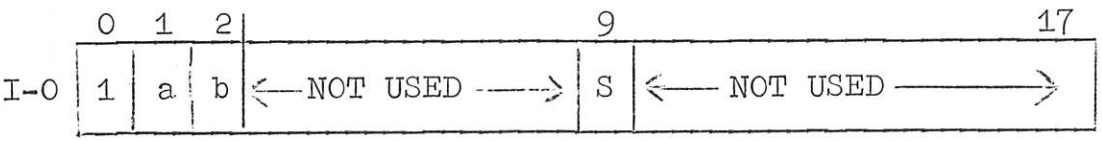
Note 4: T_7 - T_0 select the track.

Note 5: S=1, seek specified track and generate a completion pulse when new track has been found. S=0, do not seek track. The track register is unchanged and no completion pulse is generated.

Note 6: If the instruction is accepted, the R, P, H_b and unit bits are always loaded into the Drum Controller.

Note 7: If the FSCU is instructed to seek a non-existent track, the boom will be positioned to the last, i.e., highest numbered, track and a completion pulse returned. However, subsequent attempts to read or write will produce a "Wrong Address" error. (See FSCU status.)

Drum Diagnostics



Note 1: a=1 sets a phase error on the last bit of the current word.

b=1 resets the Phase Error F.F. in the FSCU

Note 2: When writing, A=1 causes a real phase error to occur on the Fastrand. This condition will be detected on subsequent readback of the data.

When reading, A=1 causes a simulated phase error to occur in the FSCU.

Note 3: If both "a" and "b" equal one, a phase error will be generated and the Phase Error F.F. will remain set.

4.1.3 Initiate Mag. Tape Activity AA=02:

When I-O bit 0 is zero, this instruction can be accepted only if the MTCU is available, not busy, and not connected to the busy Data Channel. When I-O bit 1 is one, this instruction is unconditionally accepted. Acceptance steps the PC. The meaning of the I-O is dependent upon bit 0.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
I-O	0	X	R	C_4	C_3	C_2	C_1	C_0	B_5	B_4	B_3	B_2	B_1	B_0	U_3	U_2	U_1	U_0

C_4	C_3	C_2	C_1	C_0	ACTION
0	0	0	0	0	No Operation
0	0	0	1	0/1	Position Tape Loops for For'd / Back'd
0	0	1	0	0/1	Rewind No Interlock / With Interlock
0	0	1	1	1	Erase Tape (5 inches)
0	1	1	0	0/1	Write End of File High Dens. / Low Dens.
1	1	0	0/1	0/1	Space For. "B" Blocks EP; HD / OP; LD
1	1	1	0/1	0/1	Space Back."B" Blocks EP; HD / OP; LD

Note 1: The attempt to read or write by iot is illegal and will cause an abnormal condition to arise.

Note 2: NOP may be used to select a Uniservo and set the block count.

Note 3: Spacing forward checks parity; spacing backward does not.

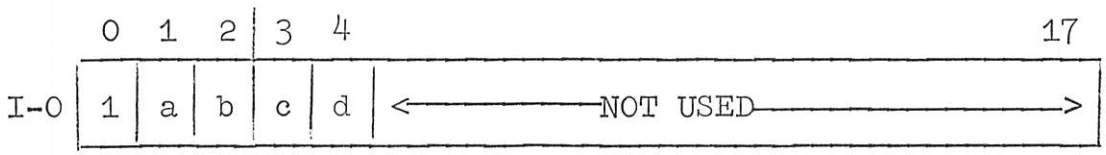
Note 4: The B bits specify the number of blocks to be spaced.

Note 5: The U bits specify the tape unit.

Note 6: R=1 specifies recovery mode.
R=0 specifies normal mode.

Note 7: Normal completion time varies with the type of operation performed; abnormal completion is returned in a minimum of 200 microseconds.

Tape Diagnostics



Note 1: a=1 causes all mag. tape operations, except a rewind, to come to an abnormal STOP.

b=1 produces an early completion signal over Sequence Break Channel #4, but does not alter the current tape operation.

c=1 when writing causes a single bit of one frame of data to be written incorrectly thus producing a lateral parity error on readback.

d=1 introduces a tape malfunction condition causing the current operation to come to an abnormal STOP.

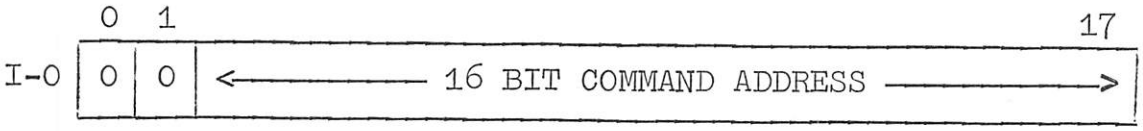
4.1.4 Initiate Channel Activity AA=05

When I-O bit 0 is zero, this instruction can be accepted only if the Data Channel is available and not busy. Under these conditions I-O bit 1 determines whether to START the Channel or to CONTINUE its accessing commands if it had previously been halted.

When I-O bit 0 is one, this instruction is unconditionally accepted. For this condition I-O bit 1 enables certain diagnostic functions to be performed.

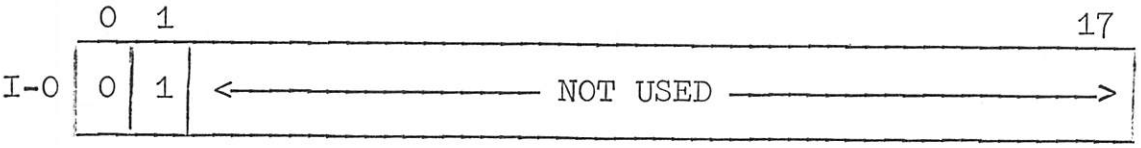
Acceptance of the instruction steps the PC.

START



Note 1: Acceptance of this instruction loads the command address into CAC and STARTS the Channel to access commands.

CONTINUE

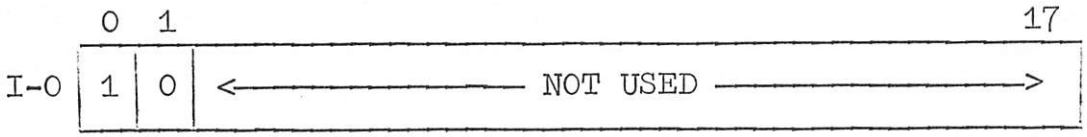


Note 2: Acceptance of this instruction causes the Channel to CONTINUE accessing commands from the address specified by the current value of CAC.

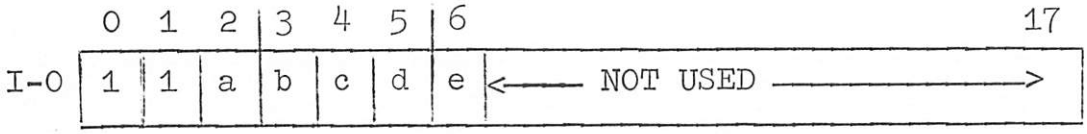
Note 3: This is a second type of CONTINUE signal. (The first is that generated by the Data Channel as specified by CCR). A third CONTINUE signal may be generated by the CONTINUE switch on the PDP-1d control panel.

Data Channel Diagnostics

STOP AT NEXT EOR



Note 4: The Channel is instructed to unconditionally stop at the next End Of Record.



Note 5: a=1 causes the Data Channel to unconditionally STOP now.

b=1 sets the Data Channel into the Command cycle.

c=1 causes the Channel to make a one word memory request from the location specified by DAC if in Data Cycle; from CAC if in Command cycle.

d=0 sets the direction of data flow to be CHANNEL → MEMORY.

d=1 sets the direction of data flow to be MEMORY → CHANNEL, provided the Channel is connected to itself, i.e., DCR=0.

e=1 sets the Channel into the Data cycle.

Note 6: a=0; b=0; c=0, and e=0 have no effect.

4.2 I-O Subsystem Status

When operating the I-O Subsystem, the programmer may wish to know the status of the operation being performed or just completed. Prior to initiating an I-O transfer he may wish to determine the availability of the equipment about to be used. During an I-O transfer he may wish to ascertain how far the operation has proceeded, i.e., how many words remain to be processed. Upon completion of the transfer he will want to know whether or not the transfer was normal or abnormal. If abnormal, he will need to know whether the abnormal condition was caused by a data error or a malfunction. If a data error occurred, he may wish to enter the appropriate error correction routine. If a malfunction occurred, he may wish to call it to the attention of an operator or maintenance engineer. Thus, a knowledge of I-O Subsystem status is essential to proper Data Channel programming.

Status is checked by executing the iot, Rem Rand In, RRI.

4.3 Rem Rand In

rri Oper. Code 72AA37

Execution of this instruction clears the I-O and transfers the status of the external equipment specified by the AA bits. At this time the I-O contains the following information.

AA	I-O
00	Data Channel - Channel Status Register
01	Fastrand Controller Status
02	Mag. Tape Controller Status
03	Not Used
04	Not Used
*05	Data Channel - Command Address Register
*06	Data Channel - Word Count Register
*07	Data Channel - Data Address Register
10	Fastrand Unit Status (Location)
11	Fastrand Unit Status (Track)
12	Mag. Tape Unit Read Status
13	Mag. Tape Unit Write Status

Note 1: The items marked with an asterick contain additional D.C. information.

Note 2: Except for the 12 or 13 status tests, the information requested is always returned within the current iot instruction time.

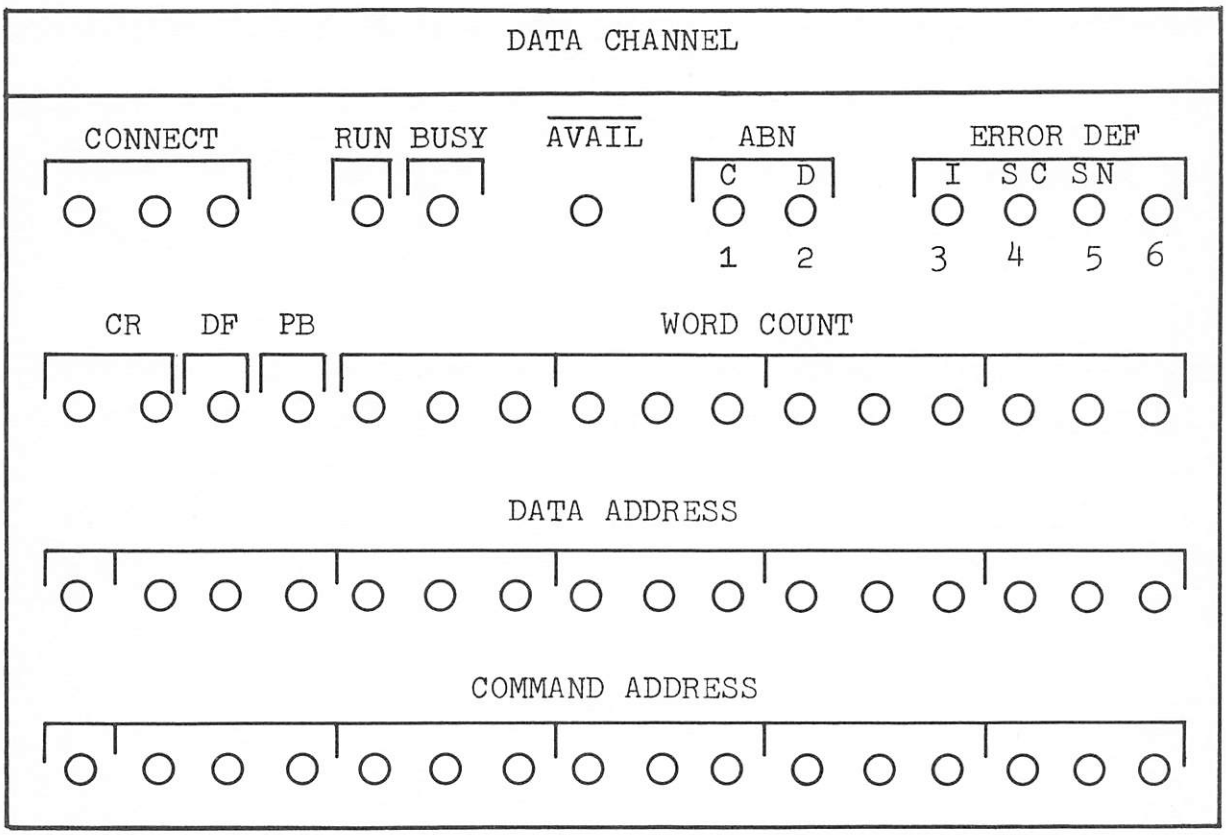
Note 3: A completion pulse will be returned to the PDP-1d only if requested, i.e., by sending the NAC level to the selected equipment. This action puts the computer in a waiting state and is not recommended.

Note 4: When checking Magnetic Tape status, i.e., tests 12 or 13, it is recommended that the iot-rri be followed by two NOP's. This is preferable to an in-out wait and guarantees the availability of tape status prior to executing the next significant instruction.

Note 5: Although the execution of an rri instruction may be completely asynchronous to the operation of the selected equipment, it is guaranteed that "correct" information is returned. This is done by delaying the sending of status to the PDP-1d until the register being interrogated contains stable information.

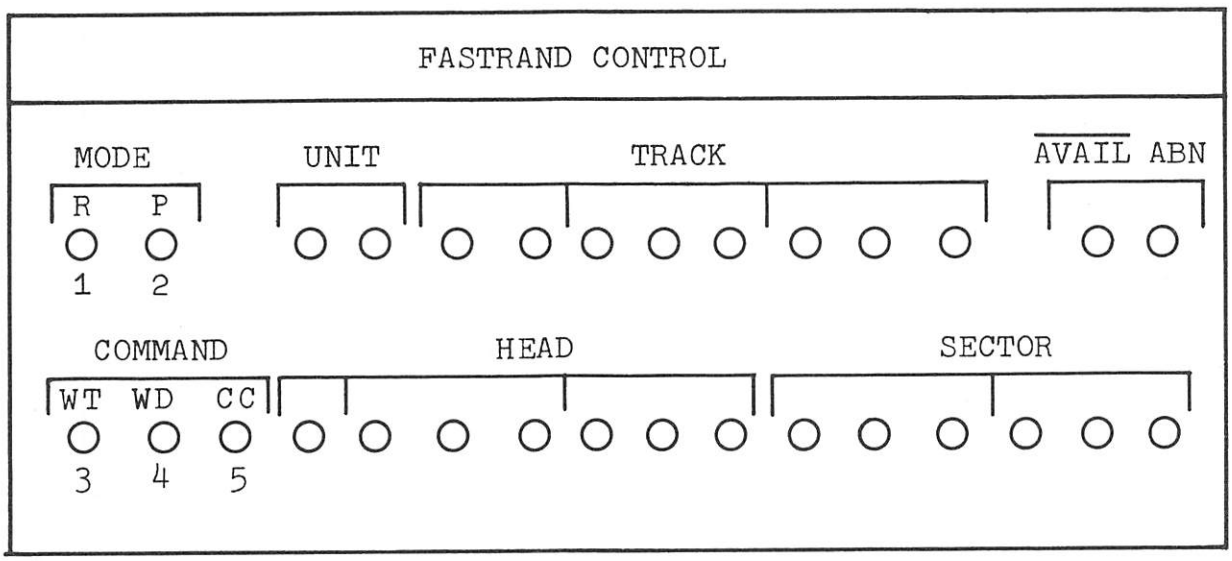
Note 6: The status conditions are cleared at various times. See individual tests for details.

Associated with the I-O Subsystem status are the various indicators mounted above the PDP-1d control panels, and those associated with the FSCU and MTCU maintenance panels, normally not visible. In presenting a detailed description of the status conditions, some attempt will be made to relate the condition to the visible display. The significance of the I-O for the various values of AA follows:



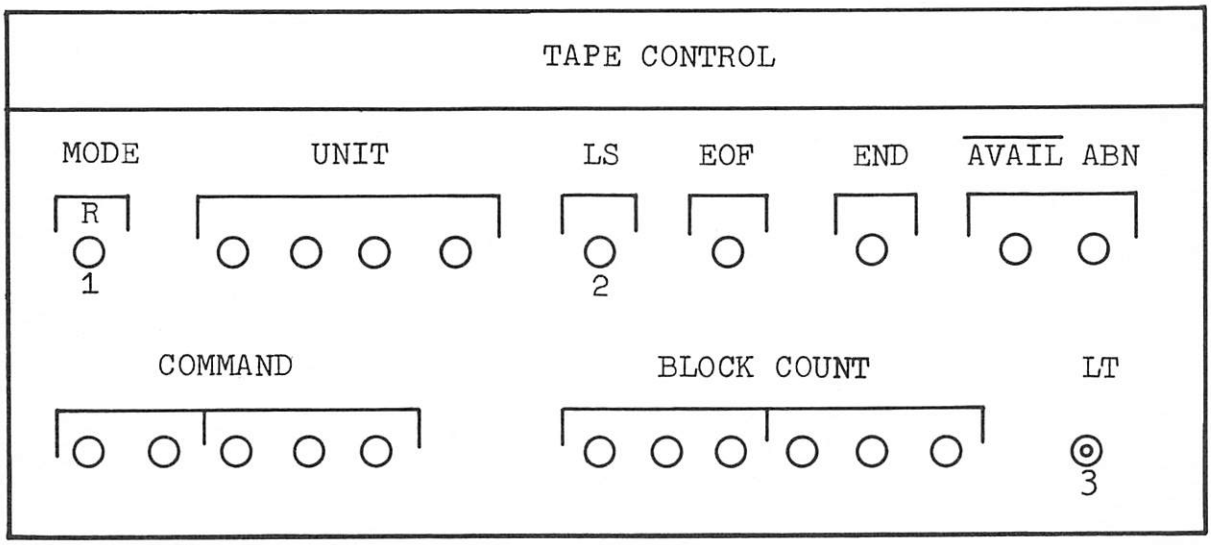
Note: 1 - when lit signifies Channel Abnormal
2 - when lit signifies Controller Abnormal
3 - when lit signifies Ignore Data Errors
4 - when lit signifies Stop on Comparison
5 - when lit signifies Stop on Non-Comparison
6 - Spare

Figure 6. Data Channel Indicators



Note: 1 - when lit signifies Recovery Mode
2 - when lit signifies Prep Mode
3 - when lit signifies Write Tag
4 - when lit signifies Write Data
5 - when lit signifies Connected to Data Channel

Figure 7. Fastrand Controller Indicators



Note: 1 - when lit signifies "Recovery Mode"
2 - when lit signifies "Low Slice Used"
3 - Lamp Test

Figure 8. Magnetic Tape Controller Indicators

4.3.1 CHANNEL STATUS REGISTER: AA=00 (See Figure 6)

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
0	ABNORMAL	The "OR" of bits 2, 3, bit 10 and compared, bit 11 and did not compare.	
1	CHANNEL UNAVAILABLE	The Channel is OFF LINE, i.e., D.C. Off-Line switch is set. In this mode it is possible to activate the Channel but the RRO iot's do not cause +1→PC. Thus CHAN. UNAVAIL. is not considered an abnormal condition. Therefore, it is not sufficient to test bit 0 alone to determine the cause of not skipping the next instruction.	Reset Off-Line switch
2	DEVICE CONTROLLER ABNORMAL	Except for data errors, the conditions that cause CONTROLLER ABNORMAL are related to its own internal operation. A data error will produce CONT. ABN. if it is not being ignored, as defined by bit 9. 1) FSCU Abnormal Conditions: a) Address Error and not in Prep Mode b) Multiple Phase Error c) Sector Overflow Error d) Missing Sector e) Track Error f) Selected Fastrand is OFF LINE or is NOT READY Note: Above conditions returned as status, only if FSCU is not in Maintenance Mode and is connected to Data Channel.	Start Clear Maint. Panel Channel Com. Accepted Computer Com. Accepted Manually

CHANNEL STATUS REGISTER: AA=OO

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
2	DEVICE CONTROLLER ABNORMAL	2) MTCU Abnormal Conditions: a) Non-Recoverable Error b) Illegal Command c) Tape Unit Busy d) End of File e) Past End of Tape f) Skew Error g) Comparison Error Note: MTCU Abnormal returned as status only when connected to Data Channel.	Start Clear Maint. Panel Channel Com. Accepted Processor Com. Accepted
3	CHANNEL ABNORMAL	The "OR" or bits 4, 5 and 14.	
4	DATA LATE	A memory request for data was made but memory did not respond in time, i.e., before the next word was needed by the Controller.	Start Clear Power On Maint. Panel Clear iot Start/Cont.
5	CHANNEL COMMAND REJECTED	A non-allowable Channel command was given in the record interval of a data transfer, i.e., 1) NLZ - Initialize and Connect 2) LCC - Device Command 3) CST - Halt	Start Clear Power On Maint. Panel iot Start/Cont. Device Command

CHANNEL STATUS REGISTER: AA=00

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
6	DATA ERROR	A device Controller detected any one of the following data errors: 1) FSCU a) Longitudinal Parity Error b) One or more Phase Errors Note: Above conditions returned as parity only if FSCU is connected to Data Channel. 2) MTCU a) Longitudinal Parity Error b) Lateral Parity Error	Start Clear Maint. Panel Channel Com. Accepted Computer Com. Accepted Start Clear Maint. Panel Channel Com. Accepted Proc. Com. Acc.
7	NON-COMPARE	The word(s) being compared did not compare.	Start Clear Power On Maint. Panel
8	- -	Spare - returned as a Zero.	
9	IGNORE DATA ERROR	The state of E_1, i.e., bit 1 of Error Stop Register.	Start Clear Power On Maint. Panel NLZ
10	STOP ON COMPARE	The state of E_2, i.e., bit 2 of Error Stop Register.	same

CHANNEL STATUS REGISTER: AA=00

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
11	STOP ON NON-COMPARE	The state of E_3, i.e., bit 3 of Error Stop Register.	Start Clear Power On Maint. Panel NLZ
12	SPARE	The state of E_4, i.e., bit 4 of Error Stop Register.	same
13	CHANNEL BUSY	The Data Channel is busy accessing commands, transferring data, or waiting for the next control signal from the device controller. Associated with BUSY are two important flip-flops, RUN and GO. Neither of these are returned as status. RUN is displayed on the DC indicator panel. When RUN is on the DC is able to make memory requests. When GO is on the connected controller is enabled to send or receive information. GO is the Start/Stop signal to the controller. For normal operation RUN and BUSY are turned on by the iot START channel. BUSY remains on as long as the DC is executing a channel program or accessing commands. RUN is turned off by a Device Command given at the proper time and is turned on again when the command is accepted by the controller.	CST Abnormal Cond.

CHANNEL STATUS REGISTER: AA=00

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
13	CHANNEL BUSY (cont'd)	Execution of a Word Count instruction turns on GO. Except for Controller Command Reject (see bit 14), any time an abnormal condition arises RUN and GO are turned off, thus stopping the Controller and further memory requests. BUSY remains on until the next EOR or READY signal is received. Execution of a channel Stop instruction, CST, given at the right time turns off RUN and BUSY together. Given at the wrong time, CST turns off RUN and GO first, and BUSY at the next EOR or READY.	
14	CONTROLLER COMMAND REJECT	The Device Controller was not able to accept the device command sent to it by the DC. The controller is given one millisecond to accept the command. If it is not accepted within this time BUSY and RUN are turned off with CHANNEL ABNORMAL indicated.	Start Clear Power On Maint. Panel Device Command Next Continue signal
15	DCR0	The state of bit 0 of the Connect Register.	Start Clear Power On Maint. Panel NLZ iot Connect
16	DCR1	The state of bit 1 of the Connect Register.	
17	DCR2	The state of bit 2 of the Connect Register.	

4.3.2 DATA CHANNEL COMMAND ADDRESS REGISTER: AA=05

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
0	WC<8	The number of words to be processed is less than 8.	
1	CHANNEL BUSY	The DC is busy accessing commands or transferring data.	CST or Abnormal Cond.
2	CAC 2	The state of CAC 2.	CAC is cleared by DCJ or iot START
3	CAC 3	" " " CAC 3.	
4	CAC 4	" " " CAC 4.	
5	CAC 5	" " " CAC 5.	
6	CAC 6	" " " CAC 6.	
7	CAC 7	" " " CAC 7.	
8	CAC 8	" " " CAC 8.	
9	CAC 9	" " " CAC 9.	
10	CAC 10	" " " CAC 10.	
11	CAC 11	" " " CAC 11.	
12	CAC 12	" " " CAC 12.	
13	CAC 13	" " " CAC 13.	
14	CAC 14	" " " CAC 14.	
15	CAC 15	" " " CAC 15.	
16	CAC 16	" " " CAC 16.	
17	CAC 17	" " " CAC 17.	

4.3.3 DATA CHANNEL WORD COUNT REGISTER: AA=06

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
0	WC<4	The number of words to be processed is less than 4.	
1	WC<8	The number of words to be processed is less than 8.	
2	CONTINUE CONTROL	The state of CCR bit 0.	Start Clear Maint. Panel Power On LWC
3	CONTINUE CONTROL	The state of CCR bit 1.	
4	DATA DIRECTION	The state of DDF.	"
5	PROGRAM BREAK REQUEST	The state of PBF.	"
6	CWC 6	The state of CWC 6.	CWC is cleared to all ones by LWC
7	CWC 7	" " " " 7.	
8	CWC 8	" " " " 8.	
9	CWC 9	" " " " 9.	
10	CWC 10	" " " " 10.	
11	CWC 11	" " " " 11.	
12	CWC 12	" " " " 12.	
13	CWC 13	" " " " 13.	
14	CWC 14	" " " " 14.	
15	CWC 15	" " " " 15.	
16	CWC 16	" " " " 16.	
17	CWC 17	" " " " 17.	

4.3.4 DATA CHANNEL DATA ADDRESS REGISTER: AA=07

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
0	- -	Spare - returned as a one.	
1	DAC ALL ONES	The Data Address Register is all ones.	
2	DAC 2	The state of DAC 2.	DAC is cleared by LDA
3	DAC 3	" " " " 3.	
4	DAC 4	" " " " 4.	
5	DAC 5	" " " " 5.	
6	DAC 6	" " " " 6.	
7	DAC 7	" " " " 7.	
8	DAC 8	" " " " 8.	
9	DAC 9	" " " " 9.	
10	DAC 10	" " " " 10.	
11	DAC 11	" " " " 11.	
12	DAC 12	" " " " 12.	
13	DAC 13	" " " " 13.	
14	DAC 14	" " " " 14.	
15	DAC 15	" " " " 15.	
16	DAC 16	" " " " 16.	
17	DAC 17	" " " " 17.	

4.3.5 FASTRAND CONTROLLER STATUS: AA=01 (See Figure 7)

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
0	ABNORMAL	The OR of bits 2 through 11	
1	CONTROLLER UNAVAILABLE	The Fastrand Controller is unavailable because the OFF LINE or MAINT mode switch on the controller maintenance panel is set. With either switch set the controller will not accept RRO iot instructions. If MAINT switch is set the controller will not accept Device Commands from the Channel.	Manual reset switches
2	ADDRESS SECTION ERROR	The "OR" of WRONG ADDRESS (see bit 8) and MISSING SECTOR. Missing Sector is not directly returned as status and means that the FSCU was unable to find the specified sector within one full drum revolution.	Start Clear Maint. Panel Chan. Com. Accepted Computer Com. Accepted
3	TAG SECTION ERROR	An error occurred in the TAG SECTION of the sector. If an error was detected in the ADDRESS section of a sector tag section errors are ignored. If the Address section was read properly, a Tag Section Error has the following meaning: 1) A Phase Error or Longitudinal Parity Error was encountered in the Tag section. 2) A Track error occurred, i.e., the Boom moved while the heads were reading or writing the Tag Section.	Start Clear Maint. Panel Chan. Com. Accepted Computer Com. Accepted

FASTRAND CONTROLLER STATUS: AA=01

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
4	DATA SECTION ERROR	An error occurred in the DATA SECTION of the sector. If an error was detected in the ADDRESS or TAG section of the sector, Data Errors are ignored. If both the Address and Tag sections were read without error, a Data Section Error has the following meaning: 1) A Phase Error or Longitudinal Parity Error was encountered in the Data Section. 2) The Boom moved while the heads were reading or writing the Data Section.	Start Clear Maint. Panel Chan. Com. Accepted Computer Com. Accepted
5	SINGLE PHASE ERROR	A single Phase Error occurred in a "good" sector while reading. Bits 2, 3, 4, and 8 may be checked to determine where in the sector the phase error occurred. A "good" sector is one whose sentinel (which precedes the Address) is 001100. A phase error inverts the meaning of information following the error, causing "ones" to be read as "zeros" and vice versa. Data in error may be recovered by entering Recovery Mode.	Start Clear Maint. Panel Chan. Com. Accepted Computer Com. Accepted
6	LONG. PARITY ERROR	A Longitudinal Parity error occurred while reading. Bits 2, 3, and 4 may be checked to determine where the error occurred, i.e., the Tag or Data sections. Note, when writing, longitudinal parity characters are generated for the tag and data sections. When reading, the tag and its LPC are checked for odd parity, and similarly, the data and its LPC are checked for odd parity.	Same

FASTRAND CONTROLLER STATUS: AA=01

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
7	MULTIPLE PHASE	Two or more Phase Errors were detected within a single section (Address, Tag, or Data) of a sector. This is a non-recoverable error.	Start Clear Maint. Panel Chan. Com. Accepted Computer Com. Accepted
8	WRONG ADDRESS	The Unit, Track, and Head information read from a "good" sector did not compare with the U, T, and H fields of the instruction currently being executed, or a phase error was detected anywhere in the address section of the sector.	same
9	SECTOR OVERFLOW	An attempt was made to perform a multi-sector read or write operation starting at or before sector 63 (i.e., the last sector) and ending beyond it. Note: When performing a multi-sector data transfer the Sector Search Register in the FSCU is automatically counted up from its initial value.	same
10	"ON" TRACK ERROR	The Boom moved during a read or write operation.	same
11	SELECTED FASTRAND NOT READY	Selected unit is NOT READY because of one or more of the following conditions: 1) Power Off 2) Heads Not Flying 3) Off Line 4) Not Plugged into the FSCU 5) A Physical Interlock Condition is present	Manual Intervention

FASTRAND CONTROLLER STATUS: AA=01

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
12	- -	Spare - returned as a zero.	
13	- -	Spare - returned as a zero.	
14	- -	Spare - returned as a zero.	
15	- -	Spare - returned as a zero.	
16	PREP MODE	The Controller's Prep Mode flip-flop is set.	Start Clear Maint. Panel Chan. Command Accepted (Type 2) Comp. Command Accepted
17	RECOVERY MODE	The Controller's Recovery Mode flip-flop is set.	Maint. Panel Chan. Command Accepted (Type 2) Comp. Command Accepted

4.3.6 FASTRAND DRUM LOCATION: AA=10

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
0	FASTRAND CONTROLLER BUSY	The FSCU is connected to the Data Channel which is Busy, or the Fastrand Drum Boom is moving.	
1	FSCU CONNECTED TO CHANNEL	The FSCU is connected to the Data Channel.	
2	FASTRAND SECTOR VALID	The Sector currently being read from the Fastrand (See Sector Register) is a valid sector, i.e., 1) The Boom is not seeking a track and the selected Unit is ON TRACK. 2) The selected unit is READY (see AA=01, bit 11). 3) This is a "good" sector and U, T, and H compare. 4) This is a "good" sector whose address has no phase error. 5) A clear caused by one of the following conditions is not occurring: a) Acceptance of a channel instruction b) Acceptance of a computer instruction.	
3	UNIT REGISTER	The state of bit 1 of the Fastrand Unit Select Register.	Maint. Panel Chan. Com. Accepted (Type 2) Computer Com. Accepted
4	UNIT REGISTER	The state of bit 0 of the Fastrand Unit Select Register.	

FASTRAND DRUM LOCATION: AA=10

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
5	HEAD REG.	The state of bit H_f.	Maint. Panel Chan. Com. Accepted (Type 1)
6	HEAD "	" " " " H_b.	
7	HEAD "	" " " " H_4.	
8	HEAD "	" " " " H_3.	
9	HEAD "	" " " " H_2.	
10	HEAD "	" " " " H_1.	
11	HEAD "	" " " " H_0.	
12	SECTOR "	" " " " SR_5.	Prior to loading it with new sector address.
13	SECTOR "	" " " " SR_4.	
14	SECTOR "	" " " " SR_3.	
15	SECTOR "	" " " " SR_2.	
16	SECTOR "	" " " " SR_1.	
17	SECTOR "	" " " " SR_0. Note: The Sector Register contains the address of the current sector being read from the drum, i.e., is the current drum location. The contents of the Sector Search Register are not returned as status.	

4.3.7 FASTRAND TRACK LOCATION: AA=11 (See Figure 7)

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
0	FSCU BUSY	The Fastrand Controller is connected to the Data Channel which is Busy, or the Fastrand Drum Boom is moving.	
1	FSCU UNAVAILABLE	See AA=01, bit 1.	
2	FSCU CONNECTED TO CHANNEL	The Fastrand Controller is connected to the Data Channel.	
3	FSCU UNIT REG.	The state of bit 1 of the Fastrand Unit Select Register.	Maint. Panel Chan. Com. Accepted (Type 2) Comp. Com. Accepted
4	FSCU UNIT REG.	The state of bit 0 of the Fastrand Unit Select Register.	
5	- -	Spare - returned as zero.	
6	- -	Spare - " " "	
7	- -	Spare - " " "	
8	- -	Spare - " " "	
9	BOOM IN MOTION	The boom of the selected Fastrand is seeking a track.	

FASTRAND TRACK LOCATION: AA=11

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
10	TRACK REG.	The state of bit 7 of the Track Address Register.	Maint. Panel Chan. Com. Accepted (Seek) Comp. Com. Accepted (Seek)
11	TRACK "	" " " " 6 " " " " "	
12	TRACK "	" " " " 5 " " " " "	
13	TRACK "	" " " " 4 " " " " "	
14	TRACK "	" " " " 3 " " " " "	
15	TRACK "	" " " " 2 " " " " "	
16	TRACK "	" " " " 1 " " " " "	
17	TRACK "	" " " " 0 " " " " "	

4.3.8 UNISERVO III-C CONTROLLER (MTCU) STATUS: AA=02 (See Figure 8)

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
0	MTCU ABNORMAL	The OR of bits 2, 3, 4, 5, 6, 7, 9; bits 1, 9, 11 of an A12 or A13 status test.	
1	MTCU UNAVAILABLE	The Mag. Tape Controller is OFF-Line, i.e., the Off-Line switch on the maint. panel is "ON." Note: If the MTCU is Off-Line it will not accept Channel Commands or RRO iot instructions.	Manual Reset of switch
2	NON-RECOVERABLE ERROR	One of the following conditions arose prior to or during a tape operation: 1) UNISERVO III-C was unable to operate due to: a) Power Off b) Rewound with Interlock c) Write Lock-out (No Write Ring) d) Short in Trellis, i.e., cable to tape unit shorted. 2) MTCU unable to operate due to malfunction of a Timing Counter. a) Runaway b) Start Delay Error c) Stop Delay Error 3) Abnormal Condition found on Tape Reel due to: a) Interblock Gap Noise b) Drop-out of Data Frame(s) c) Non-Erasable Section of Tape 4) An Echo Check error occurred, i.e., did not confirm that a data frame was written. 5) An iot Tape Diagnostic with bit 4 a one. 6) A delay checking circuit is malfunctioning.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted

MTCU STATUS: AA=02

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
3	ILLEGAL COMMAND	The MTCU was given an illegal command: 1) The Command field contained a non-recognizable bit combination, i.e., did not decode into a known tape function. 2) The computer issued a read or write instruction. 3) Attempt was made to write in Recovery mode. 4) Unacceptable command for non-stop operation. a) A non-stop Write followed by a Backspace, Position, Rewind, NOP, or Erase instruction. b) A non-stop Forward Space or Read followed by a Backspace, Position, Rewind, NOP, or Erase instruction. c) A non-stop Backspace followed by a Read, Write, Forward Space, Position, Rewind, NOP, or Erase instruction. Note the following combinations are acceptable. a) A non-stop Write followed by a Forward Space or Read instruction. b) A Forward Space or Read followed by a Write instruction.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted
4	UNISERVO BUSY ERROR	The selected tape unit is busy rewinding (no interlock) or reversing loops and cannot perform the current command. This condition is meaningful to the PDP-1d rather than the Data Channel. If the D.C. gives a command to a busy tape unit, the command is rejected and CHANNEL ABNORMAL and CONTROLLER COMMAND REJECT are set. However, if the PDP-1d issued a command to a tape unit	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted

MTCU STATUS: AA=02

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
4	UNISERVO BUSY ERROR (cont'd)	that was busy rewinding or reversing loops, the MTCU would accept the command, i.e., +1→PC, and approx. 250 microseconds later send back a channel #4 sequence break. The same thing would happen if the tape unit were not busy. Therefore, UNISERVO BUSY ERROR provides a means of distinguishing between abnormal and normal completion of processor commands. (See Section 4.3.9, Uniservo Busy.)	
5	LATERAL PARITY ERROR	A lateral parity error has occurred during a Read, Write, or Space Forward operation. Note: Between the Uniservos and the Mag. Tape Controller is a Tape Adapter Module, TAM. Strictly speaking TAM is part of the MTCU, but it is useful to differentiate between them. TAM has two seven-bit data registers: a High Slice (or Gain) register, and a Low Slice register. The High Slice register has a tendency to suppress noise but also reject weak signals. The Low Slice register has a tendency to accept weak signals but also may pick up noise. TAM takes the data read from a tape unit and loads both registers simultaneously. Only the High Slice register is used to check lateral parity. Normally, TAM sends the MTCU the output of this register. However, if it detects a parity error in a data frame, it sends the MTCU the contents of the Low Slice register and notifies the controller of this fact.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted

MTCU STATUS: AA=02

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
5	LATERAL PARITY ERROR (cont'd)	It is necessary to distinguish between a read and a write operation. 1) Read or Space Forward During a read or space forward operation if TAM detects a parity error, the contents of the Low Slice register are sent to the MTCU and the Low Slice Flip-Flop is set. (See bit 8.) The tape controller performs a parity check of data it has received. If it detects a parity error it sets the Lateral Parity Error flip-flop. 2) Write During a write operation the data just recorded is read back and sent to TAM but not to the Controller. TAM checks the parity of the data read back as described above. No parity check is performed by the controller. Instead, upon sensing a Low Slice signal, it sets the Lateral Parity Error flip-flop. Thus, when writing, the "Low Slice Used" bit when set means that on read-back the High register contained an error and says nothing about the validity of the character in the Low register.	

MTCU STATUS: AA=02

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
6	LONGITUDINAL PARITY ERROR	A Longitudinal Parity Error has occurred during a read, write, or space forward operation. When writing, a longitudinal parity character, LPC, is generated on the basis of even parity, and is the last character on the tape for that block. When reading, TAM counts (modulo 2) the number of ones in each tape channel including the corresponding bit of the LPC. If the count for any channel is odd, a Long. Parity Error has occurred. The LPC itself is not checked for lateral parity nor is it transmitted to the MTCU. Note, in Recovery Mode the contents of the Mod. 2 counter are sent to the MTCU to permit single error detection and correction. See Appendix "A."	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted
7	SKEW ERROR	When writing, the read-back indicated the spread, i.e., the time distribution, of the seven bits of the character exceeded its allowable limit.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted
8	LOW SLICE USED	A lateral parity error was found in the High-Slice register and data was accepted from the Low-Slice register.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted

MTCU STATUS: AA=02

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
9	END OF FILE	An End-of-File character (0011110) was detected when reading or spacing tape forward.	same
10	RECOVERY MODE	The MTCU is capable of operating in Recovery Mode.	Start Clear Maint. Panel Comp. Com. Accepted
11	CONNECTED TO CHANNEL	The MTCU is connected to the Data Channel.	
12	BLOCK COUNT	The state of B_5 of the Block Counter.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted
13	BLOCK "	" " " B_4 " " " "	
14	BLOCK "	" " " B_3 " " " "	
15	BLOCK "	" " " B_2 " " " "	
16	BLOCK "	" " " B_1 " " " "	
17	BLOCK "	" " " B_0 " " " "	
		Note: The Block Counter indicates the number of blocks to be processed.	

4.3.9 UNISERVO III-C READ STATUS: AA=12

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
0	TEST INVALID	The RRI iot instruction was given while the MTCU was Busy and the Uniservo was not interrogated. However, information other than tape status is valid, i.e., bits 3-7, 10, 11, 14-17. Note 1: If the tape unit whose status is to be checked is not the one specified by the Unit Register, then the RRI iot must be preceded by an RRO iot that merely sets the unit register, i.e., a NOP. Note 2: If the status test is valid the controller is unable to accept a Channel Command or another Computer Command until the end of the status check. Note 3: Unlike the other RRI iot instructions status is not returned within 5.2 microseconds, i.e., the current instruction cycle time. Therefore, it is suggested that this RRI iot be followed by two PDP-1d NOP instructions to guarantee that the I-O has the proper information in it.	The partial or complete execution of the status test clears the Invalid Condition.
1	UNISERVO FAULT	The UNISERVO III-C is not operable due to: a) Power Off b) Rewound with Interlock c) Short in Trellis, i.e., cable to tape unit shorted. d) Write Lock-out (No Write Ring) Note: These are Non-Recoverable Errors, i.e., require manual intervention.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted

UNISERVO III-C READ STATUS: AA=12

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
2	UNISERVO BUSY	The Uniservo is busy rewinding tape or reversing tape loops. Note 1: The MTCU may receive a device command while the tape unit to be selected is busy. However, it will not send back an acceptance to the Data Channel. Thus, in addition to "TAPE UNIT BUSY" the D.C. status will indicate "CONTROLLER COMMAND REJECT." Note 2: The MTCU does not remain busy while the tape unit is rewinding or positioning loops. Measured from the time of acceptance of a device command the MTCU will be busy for approx. 250 microseconds if the current command is a Position or Rewind order, and approx. 10 milliseconds for a Rewind with Interlock. A completion signal is returned to the Data Channel or the computer when the MTCU becomes not busy. Note 3: Positioning of loops requires 600 milliseconds.	See adjacent remarks
3	COMM. REG.	The state of C_4 of the MTCU Command Register.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted
4	COMM. "	" " " C_3 " " " " "	
5	COMM. "	" " " C_2 " " " " "	
6	COMM. "	" " " C_1 " " " " "	
7	COMM. "	" " " C_0 " " " " "	

UNISERVO III-C READ STATUS: AA=12

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
8	LOAD POINT	The selected tape unit is rewound without interlock and is ready to be operated.	
9	PAST END POINT	The End of Tape Warning was sensed and tape was spaced, read, written, or erased beyond the warning marker. This condition results in "Controller Abnormal" (AA=00 bit 2), and "MTCU Abnormal" (AA=02 bit 0). Note: If the tape is rewound or backspaced over the marker, this condition goes away and will not re-appear until the next forward operation causes the marker to be sensed.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted
10	MTCU BUSY	The Magnetic Tape Controller is busy executing a Channel command or a Computer instruction.	Start Clear Maint. Panel Normal or Abnormal Completion of Command
11	COMP. ERROR	The Low Slice and High Slice registers did not compare during the read-back of a write operation. Note 1: This is the third check performed during a write operation, the other two being Lateral Parity Check (AA=02 bit 5) and Skew check (AA=02 bit 7). Note 2: Detection of a Comp. Error results in "MTCU Abnormal" (AA=02 bit 0). Note 3: When writing, if "Low Slice Used" is set "Comp. Error" should also be set.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted

UNISERVO III-C READ STATUS: AA=12

I-O BIT	CONDITION	SIGNIFICANCE	CLEAR
12	- -	Spare - returned as a one.	
13	- -	Spare - returned as a one.	
14	UNIT REG.	The state of U_3 of the MTCU Tape Unit Reg.	Start Clear Maint. Panel Chan. Com. Accepted Comp. Com. Accepted
15	UNIT "	" " " U_2 " " " " " "	
16	UNIT "	" " " U_1 " " " " " "	
17	UNIT "	" " " U_0 " " " " " "	

4.3.10 UNISERVO III-C WRITE STATUS: AA=13

This RRI iot, i.e., 721337, is used solely to determine whether or not the selected tape unit is write locked out, i.e., has no write ring mounted on the tape reel. When the tape reel does have a write ring the status conditions returned are identical to those returned for a Read Status check. When the tape reel does not have a write ring the status conditions returned are different from those of a Read check. It is this difference that enables the programmer to determine the presence or absence of a write ring.

The following tables amplify and explain the preceding remarks:

Table 1 - No Write Ring Check

Case		Tests Performed (A13 followed by A12)	Status Conditions Returned	Remarks
1	Not Busy	A13	Tape Fault	No Write Ring
		A12	NO Tape Fault	
2	Not Busy and Not Rewound	A13	Tape Fault	No Write Ring
		A12	NO Tape Fault	
3	Not Busy and Rewound (No Interlock)	A13	Tape Fault Load Point	No Write Ring
		A12	NO Tape Fault Load Point	
4	Not Busy and Rewound (Interlock)	A12	Tape Fault	Rewound with Interlock; Write Ring indeterminate
		A13	Tape Fault	
5	Busy	A12	NO Tape Fault Busy	Write Ring indeterminate
		A13	NO Tape Fault Busy	

Table 2 - Write Ring Check

1	Not Busy	A13	NO Tape Fault	Write Ring
		A12	NO Tape Fault	
2	Not Busy and Not Rewound	A13	NO Tape Fault	Write Ring
		A12	NO Tape Fault	
3	Not Busy and Rewound (No Interlock)	A13	NO Tape Fault Load Point	Write Ring
		A12	NO Tape Fault Load Point	
4	Not Busy and Rewound (Interlock)	A13	Tape Fault	Rewound with Interlock; Write Ring indeterminate
		A12	Tape Fault	
5	Busy	A13	NO Tape Fault Busy	Write Ring indeterminate
		A12	NO Tape Fault Busy	

Summary - Write Ring Test

1. The selected Uniservo III-C has a write ring if there is no tape fault condition for either test and the tape unit is not busy.

2. The selected Uniservo III-C does not have a write ring if the status conditions for both tests are different.

3. If both tests indicate a tape fault, the tape is rewound with interlock; for this case it is not possible to check for a write ring.

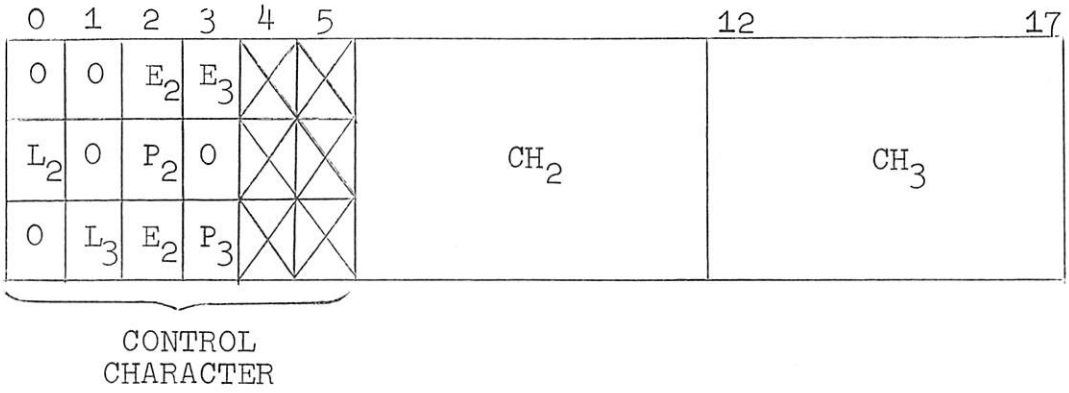
4. If the selected Uniservo III-C is busy at the time of the tests, it is not possible to check for a write ring.

APPENDIX A

Magnetic Tape Recovery Mode

Recovery mode is entered when it is desired to correct a single error occurring in a data frame. In essence a read-back of the record in error provides a pair of markers or pointers whose intersection marks the erroneous bit to be corrected. For every two characters read a third is constructed which indicates which of the two characters is in error. (If both were in error, this would be indicated by the control character.) At the end of the read, the longitudinal error character is returned either by itself or preceded by the last character. Which of these conditions prevails will be indicated by the control character. Furthermore, the control character will indicate whether or not the last character is in error. The longitudinal error character provides the second pointer and indicates in which one of the seven channels the error has occurred. The seventh error channel bit pointer is contained in the control character itself. If more than a single error has occurred, more than one channel error bit pointer will be set. If the number of words in the record is even, the longitudinal error character will appear by itself. If the number is odd, the error character will be preceded by the last character.

Thus at the end of the recovery read operation the data in core memory will have the following formats:



CH_2 = CHARACTER TWO

CH_3 = CHARACTER THREE

E_2 = CH_2 IS IN ERROR

E_3 = CH_3 IS IN ERROR

L_2 = CH_2 IS THE LONG. ERROR CHARACTER

L_3 = CH_3 IS THE LONG. ERROR CHARACTER

P_2 = 7th ERROR CHANNEL BIT POINTER

Since recovery mode is useful only if a single parity error has occurred, the following background information is presented to assist the programmer in determining whether or not more than one error has occurred.

If data errors are not being ignored, the occurrence of a parity error while reading will cause the data transfer to end abnormally.

In particular, if a lateral parity error occurs, the word containing the character in error and all succeeding words are not transferred to the Data Channel. However, the tape unit will continue to move tape until the entire record has been "read." At the end of the operation the Word Count will indicate the remaining numbers of words not read and the Data Address Counter will be pointing to the last good word read. A non-stop read will halt at the end of the first record.

If a longitudinal parity error has been detected but no lateral parity error has occurred, the entire record will have been read. However, for this situation to occur an even number of errors must have occurred within a character. For this case recovery mode will be of no use since subsequent read-back in this mode will not point out the character in error.

If an even number of errors have occurred in a single channel (including the lateral parity channel) and an odd number of errors have occurred within a character, the operation will end with a lateral parity indication but no longitudinal parity error indication. Again recovery mode is useless since it will not reveal the character(s) in error.

Thus, a prerequisite for entering recovery mode is the indication of a lateral parity error and a longitudinal parity error. If re-reading the record in recovery mode indicates that more than one error has occurred, the characters in error cannot be reconstructed unless the context of the data in the record is known.

Having read the record in recovery mode, it is desirable to look at the longitudinal error character first before proceeding to examine each data character. If such an examination indicates more than one channel bit pointer is set, more than one error has occurred and recovery mode is of little use. Note, at the end of the recovery read the Data Address Counter will be pointing to the last "word" read.

Programming Hints:

1. Since 2/3 of a word at a time are recovered for each word of the record, to read the entire record the word count must be at least 1-1/2 times the record length. To accomplish this it will be convenient to give the maximum word count of 4096 words, i.e., all ones.

2. The Word Count word should specify continue on EOR (or READY).

3. The NLZ command should set the error definition to ignore data errors.

4. Recovery mode can only be entered by an RRO iot. Thus it is convenient to set the recovery mode bit and backspace one block with the same instruction.

5. For records greater than 3730 words, i.e., 2/3 of 4096, it will be necessary to perform a scatter read type of operation which first specified a continue on WC→0, followed by a new word count and continue on EOR.

6. It is illegal to try to write in recovery mode.

APPENDIX B

Ready Reference Summary

Channel Commands

NLZ:
- 56010C - Connect to C and stop on non-compare
- 56020C - Connect to C and stop on comparison
- 56040C - Connect to C and ignore data errors
- 56050C - Connect to C, ignore data errors, stop on non-compare
- 56060C - Connect to C, ignore data errors, stop on comparison
- 540X0C - Change connection, but not stop definition
- 520E0X - Change stop definition, but not connection

LDA:
- 60AAAA - Begin data transfer at AAAA of Core 0
- 61AAAA - " " " " " " " 1
- 62AAAA - " " " " " " " 2
- 63AAAA - " " " " " " " 3
- 64AAAA - " " " " " " " 4
- 65AAAA - " " " " " " " 5
- 66AAAA - " " " " " " " 6
- 67AAAA - " " " " " " " 7
- 70AAAA - " " " " " " " 10
- 71AAAA - " " " " " " " 11
- 72AAAA - " " " " " " " 12
- 73AAAA - " " " " " " " 13
- 74AAAA - " " " " " " " 14
- 75AAAA - " " " " " " " 15
- 76AAAA - " " " " " " " 16
- 77AAAA - " " " " " " " 17
- 777777 - Skip the number of data words specified by the word count

DCJ: 4AAAAA - Jump to the command specified by the 14 bit address

LCC: 2DDDDD - Load controller with Type 1 Command
3DDDDD - Load controller with Type 2 Command

LWC: 00WWWW - Man. advance, normal transfer
01WWWW - Man. advance, normal trans., program break
02WWWW - Man. advance, compare mode
03WWWW - Man. advance, compare mode, program break
04WWWW - WC→0, normal transfer
05WWWW - WC→0, normal transfer, program break
06WWWW - WC→0, compare mode
07WWWW - WC→0, compare mode, program break
10WWWW - EOR, normal transfer
11WWWW - EOR, normal transfer, program break
12WWWW - EOR, compare mode
13WWWW - EOR, compare mode, program break
14WWWW - Ready, normal transfer
15WWWW - Ready, normal transfer, program break
16WWWW - Ready, compare mode
17WWWW - Ready, compare mode, program break

CST: 000000
010000
020000
030000
040000
050000
060000
070000
} Channel stop

Dummy Word Count:

100000 - Cont. on EOR, normal transfer
110000 - Cont. on EOR, normal transfer, program break
120000 - Cont. on EOR, compare mode
130000 - Cont. on EOR, compare mode, program break
140000 - Cont. on Ready, normal transfer
150000 - Cont. on Ready, normal transfer, prog. break
160000 - Cont. on Ready, compare mode
170000 - Cont. on Ready, compare mode, prog. break

Device Controller Commands (from Data Channel)

FSCU - Type 1:

20HHSS - RT/RD, moving heads
21H0SS - RT/RD, fixed heads
22HHSS - RT/WD, moving heads
23H0SS - RT/WD, fixed heads
24HHSS - WT/RD, moving heads
25H0SS - WT/RD, fixed heads
26HHSS - WT/WD, moving heads
27H0SS - WT/WD, fixed heads

Note: Which drum cylinder is specified by H_b, i.e., bit 6.

H_b = 0 specifies lower drum (drum "A") cylinder

H_b = 1 specifies upper drum (drum "B") cylinder

Type 2:

30U[OTT]TT - Clear R and P, don't seek track
31U[OTT]TT - Set Prep mode, don't seek track
32U[OTT]TT - Set Recov. mode, don't seek track
30U[1TT]TT - Clear R and P, seek track
31U[1TT]TT - Set Prep mode, seek track
32U[1TT]TT - Set Recov. mode, seek track

Note: Establishing mode without seeking a track hangs up the Data Channel.

MTCU - Type 1:

200BBUU - Set Unit no. and Block count (NOP)
204XXUU - Position forward
206XXUU - Position backward
210XXUU - Rewind, no Interlock
212XXUU - Rewind with Interlock
216XXUU - Erase tape
220XXUU - Write Even par. and High density
222XXUU - Write Even, High
224XXUU - Write Odd, High
226XXUU - Write Odd, Low
230XXUU - Write EOF High
232XXUU - Write EOF Low
240XXUU - Read Even, High
242XXUU - Read Even, Low
244XXUU - Read Odd, High
246XXUU - Read Odd, Low
260BBUU - Space forward, Even, High
262BBUU - Space Forward, Even, Low

264BBUU - Space Forward, Odd, High
266BBUU - Space Forward, Odd, Low
270BBUU - Space Backward, -- , High
272BBUU - Space Backward, -- , Low
274BBUU - Space Backward, -- , High
276BBUU - Space Backward, -- , Low

Note: "B" field is six bits long and occupies bits 8 - 13;
"U" field is four bits long and occupies bits 14 - 17.

MTCU - Type 2:

None

iot Commands

rro: 72AA17

00 - Connect Data Channel
01 - Initiate Drum Activity
02 - Initiate Mag. Tape Activity
05 - Initiate Channel Activity

rri: 72AA37

00 - Data Channel Status
01 - FSCU Status
02 - MTCU Status
05 - DC - CAC
06 - DC - CWC
07 - DC - DAC

10 - Fastrand Unit Status (Location)
11 - Fastrand Unit Status (Track)
12 - Mag. Tape Unit Read Status
13 - Mag. Tape Unit Write Status

Format of I-O for rro

AA = 00: 0XXXX0 - Connect Channel to itself
0XXXX1 - Connect Channel to FSCU
0XXXX2 - Connect Channel to MTCU

AA = 01: 00[0UU][0TT]TT - Clear R + P, sel. drum A, don't seek
01[0UU][0TT]TT - Set P, select drum A, don't seek
02[0UU][0TT]TT - Set R, select drum A, don't seek
03[0UU][0TT]TT - Set R + P, sel. drum A, don't seek
00[1UU][0TT]TT - Clear R + P, sel. drum B, don't seek
01[1UU][0TT]TT - Set P, select drum B, don't seek
02[1UU][0TT]TT - Set R, select drum B, don't seek
03[1UU][0TT]TT - Set R + P, sel. drum B, don't seek
00[0UU][1TT]TT - Clear R + P, sel. drum A, seek
01[0UU][1TT]TT - Set P, sel. drum A, seek
02[0UU][1TT]TT - Set R, sel. drum A, seek
03[0UU][1TT]TT - Set R and P, sel. drum A. seek
00[1UU][1TT]TT - Clear R and P, sel. drum B, seek
01[1UU][1TT]TT - Set P, sel. drum B, seek
02[1UU][1TT]TT - Set R, sel. drum B, seek
03[1UU][1TT]TT - Set R and P, sel. drum B, seek

AA = 02: See MTCU - Type 1 Commands. Except for the first octal digit, the commands are identical.

0XXXXX - Clear Recovery mode
1XXXXX - Set Recovery mode
2XXXXX - Clear Recovery mode
3XXXXX - Set Recovery mode

Note: It is illegal to try to read or write by iot.

AA = 05: 0AAAAA - Start Channel at specified address (16 bits)
2XXXXX - Continue

REFERENCES

(1) Boilen, S.; Fredkin, E.; Licklider, J. C. R.; McCarthy, J; "A Time-Sharing Debugging System for a Small Computer", Proc. SJCC 1963, Baltimore, Md.

(2) Bolt Beranek and Newman Inc. Report 975, "Progress Report No. 3, Hospital Computer Project", Cambridge, Mass. 1963.