Datapoint

2200 reference

COMPUTER TERMINAL CORPORATION

DATAPOINT 2200

REFERENCE MANUAL

Version I and Version II

The computer-oriented user will find this manual useful for evaluation of the Datapoint 2200 capabilities and limitations. However, only the hardware considerations are covered in this manual. The full utility of the Datapoint 2200 cannot be appreciated until the available software support for the machine has been reviewed.

There is a complete family of software packages available for the Datapoint 2200 including high-level languages, operating systems, source code and text editors, communications programs, utility programs, etc. Reference should be made to the latest issue of the Datapoint 2200 Software Catalog for more complete information.

TABLE OF CONTENTS

PART PAGE

TABLE OF CONTENTS
(Continued)

PART PAGE

LIST OF TABLES

LIST OF ILLUSTRATIONS

PART 1

GENERAL FEATURES

1.1 INTRODUCTION

In early 1970, Computer Terminal Corporation introduced the Datapoint 2200, Version I, an integrated data system consisting of an alphaumeric keyboard for data entry, a cathode ray screen for data display and two digital cassette recorders for bulk data storage. The terminal contained a general purpose digital computer for control and processing and the capability for extensive interfacing for communications and for interfacing with external devices or peripherals. The Version I Datapoint 2200 contained a maximum of 8,192 words of recirculating MOS memory. This integrated system of small computer, memory, and peripherals gained wide acceptance in a variety of data processing applications.

In early 1972, Computer Terminal Corporation introduced Version II of the Datapoint 2200 maintaining the original concept but incorporating many improvements and additions to augment its processing power.

Version II of the Datapoint 2200 appears physically identical to Version I and retains all the advantages of the first Datapoint. In fact, in keeping with the concept of upward compatibility programs generated for the Version I Datapoint will run, though substantially faster, on the Version II Datapoint. Improvements have been made at the basic processing unit and, through the incorporation of a random access memory, the speed has been vastly increased.

Memory size in the Version II is maximum of 16,384 words of 8 bit memory. One of the most impressive features of the random access memory aside from its size is the speed at which instructions are executed. Memory reference instructions require 3.2 or 4.8 micro seconds rather than up to 520 micro seconds for the Version I. For the same reasons jump and call instructions require 6.4 micro seconds instead of up to 520 micro seconds. In addition, the execution of all the instructions (except I/O instructions) has been reduced to less than 1/4 the time required in Version I.

Version II provides the programmer an advantage of interrupt capability. This hardware interrupt feature enables the programmer to service peripherals requiring attention without the necessity of periodic addressing and status checking in his main program. Other features include: an additional set of 7 registers and 4 control flip-flops (alpha and beta modes), program access to the stack through through the new Push and Pop instructions and faster cassette tapes. These changes have all been incorporated into the 2200 Version II without the need to modify or convert user software written for the Version I 2200, except for timing or delay loops.

The achievement of a small computer with integrated keyboard, display screen, random access storage, and communications capability at such low cost now makes possible computer sophistication for applications not previously practical – particularly in the computer terminal/source data entry/communications area.

This manual provides specific hardware specifications for Version I and Version II of the Datapoint 2200 with the following model codes:

VERSION I		
	2200 - 102	(2K Memory)
	2200 - 104	(4K Memory)
	2200 - 106	(6K Memory)
	2200 - 108	(8K Memory)
VERSION II		
	2200 - 114	(4K Memory)
	2200 - 118	(8K Memory)
	2200 - 122	(12K Memory)
	2200 - 126	(16K Memory)

1.2 SYSTEM ELEMENTS

There are four basic system elements in the basic Datapoint 2200 plus the capability of interface to a number of external peripheral devices.

This manual covers the basic elements; CRT, keyboard, processor, cassette tape decks.

1.3 CRT DISPLAY

The Datapoint 2200 CRT Display provides the following features:

a. 7" x 3½" viewing area;

b. 960 characters;

c. 80 character by 12 line format;

d. 4/32" x 3/32" character size;

e. Entire 94 character ASCII set;

f. 60 frames per second refresh rate (50 frames per second when using 50 cycle power);

g. 5 x 7 matrix character generation;

h. 5 x 7 solid, blinking cursor, alternates with character, nondestructive;

i. P31 green phosphor;

j. Single control line erasure, frame erasure, and page roll-up; and

k. Direct control of all CRT functions by the 2200 processor, providing tab, editing, form control, etc.

1.4 KEYBOARD

The integral keyboard provides a basic 41 key alphanumeric key group, an 11 key numeric group and five system control keys.

The keyboard provides a unique multi-key roll-over characteristic providing maximum ease of typing. Transfer of characters from the keyboard is under control of the 2200 processor. An audible click providing an acoustical feedback to the typist is available under processor control.

A programmable audio "beep" is also provided when it is desired to gain a typist's attention.

1.5 PROCESSOR

The integral processor provides all control functions and includes:

VERSION I	VERSION II
28 different instruction types;	31 different instruction types;
7 addressable registers;	14 addressable registers;
15 deep pushdown stack;	16 deep pushdown stack;
Up to 8,192 word memory;	Up to 16,384 word memory;

Both models: 8 bit memory word length;
Complete parallel I/O system;
Automatic power-up restart.

1.6 CASSETTE TAPE DECKS

Two read-write tape decks are provided for program and data storage. The deck accepts Norelco (Phillips)-type cassettes and provides:

a. 47 characters per inch density;

b. Bi-directional operation;

c. Processor controlled data transfer, direction control, and high-speed rewind.

1.7 GENERAL SPECIFICATIONS

The Datapoint 2200 has the following general characteristics:

a. 115 v. a. c., 60 cycle, 180 watts, power input (50 cycle 230 v. a. c., 60 optional);

b. 47 pounds weight;

c. 9-5/8" high, 18-1/2" wide, by 19-5/8" deep outside dimensions;

d. 0° to 50°C (32° to 122°F), 10 to 90 percent relative humidity operation environment.

1.8 2200/COMMUNICATIONS ADAPTOR

The 2200 is well-suited to tasks involving communications. Several models of adaptors are available for implementing communications systems. The following is a description of these adaptors:

2200-400 COMMUNICATIONS ADAPTOR
This adaptor provides an EIA RS-232 interface to external datasets or other devices requiring an asynchronous, start-stop serial format, full or half duplex.

2200-401 COMMUNICATIONS ADAPTOR WITH 103 MODEM
Full or half duplex communications up to 300 baud is provided by this adaptor. An internal modem is compatible with Bell System 103 series and automatic call control provides automatic dialing and answering under program control.

2200-402 COMMUNICATIONS ADAPTOR WITH 202 MODEM
Communication up to 1200 baud can be accomplished with this adaptor. An internal modem is compatible with the Bell 202 series and automatic call control provides automatic dialing and answering under program control.

2200-403 HIGH LEVEL KEYER
This adaptor provides interface to telegraph lines or channels with either neutral or polar keying.

2200-404 SYNCHRONOUS COMMUNICATIONS
This adaptor may be used to interface a Bell System 201 synchronous or other manufacturer's equivalent modem to provide high-speed synchronous communications.

2200-420 PARALLEL DATA
For other manufacturer's peripherals requiring parallel data communications this adaptor can be connected to most logic levels.

1.9 OTHER PERIPHERALS

A number of other peripherals are available for use with the Datapoint 2200, including:

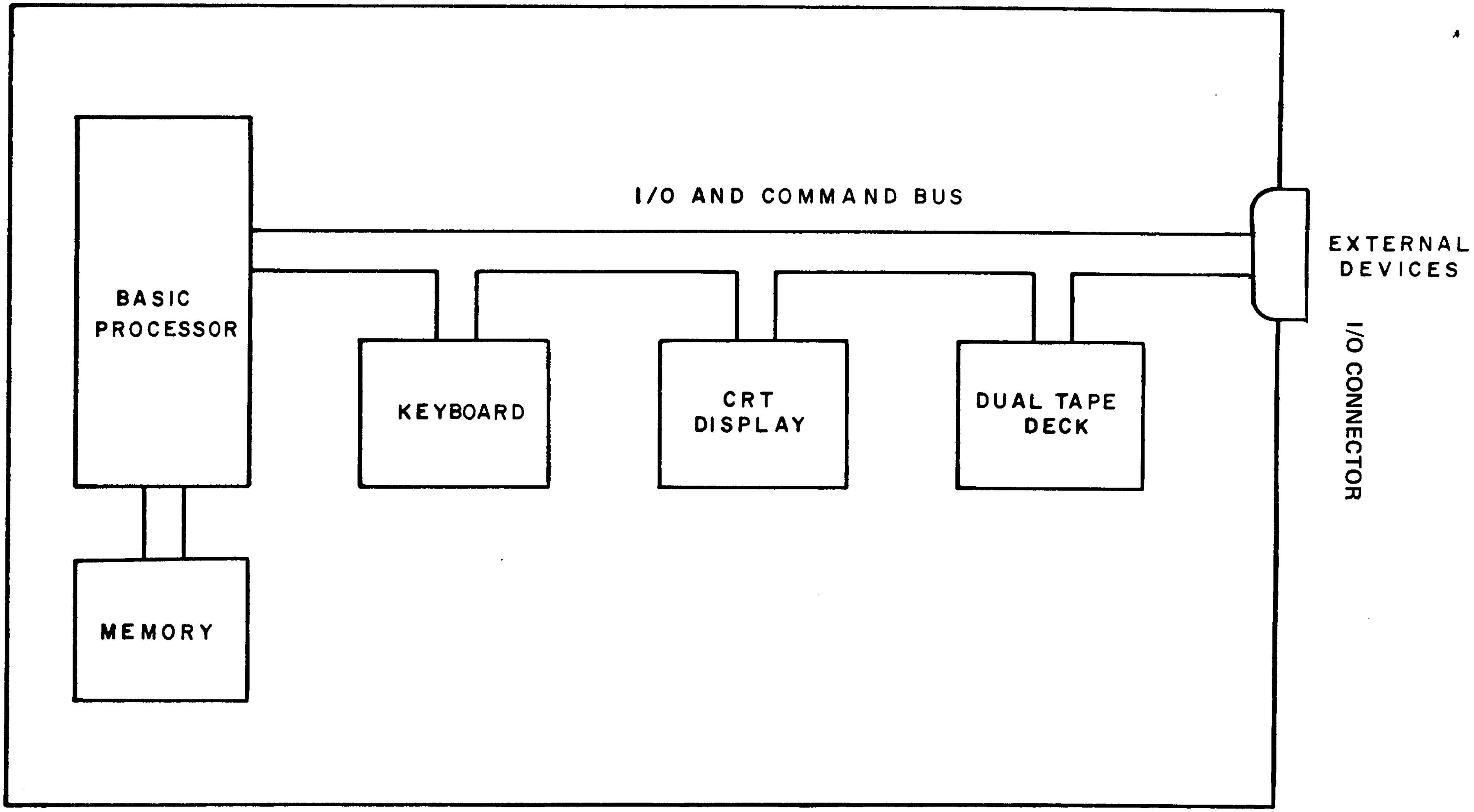
a. 2200/PRINTER A 132 column, 30 cps impact page printer.

b. 2200/LINE PRINTER A 132 column, 135 line per minute line printer.

c. 2200/TAPE Both 7 and 9 channel industry-compatible tapes will handle up to 8-1/2 inch reels.

d. 2200/DISC 2 megabyte removable cartridge disk.

e. 2200/CARD READER A 600 card/minute, 80 column card reader.



DATAPOINT 2200 — SYSTEM DIAGRAM
VERSION I & VERSION II

PART 2

BASIC PROCESSOR – VERSION I

2.1 PROCESSOR ORGANIZATION

The processor contained in the Datapoint 2200 is comprised of the Arithmetic/Logic Unit, two sets of 7 program accessible registers, 2K to 8K words of read/write memory, an instruction decoder, a 15-level hardware pushdown stack used in subroutine type operations, and an interrupt system.

2.2 ARITHMETIC/LOGIC UNIT

The Arithmetic/Logic Unit is capable of processing both binary integers and logical operands. All arithmetic and logical operations may take place between the A-register and any of the 7 program accessible registers or between the A-register and memory. The A-register always contains the result of an arithmetic or logic operation and the other register (or memory cell) is unaffected. Arithmetic and logic operations affect the Sign, Carry, Zero and Parity Flip-Flops.

2.3 PROCESSOR REGISTERS

Seven registers are present in the Datapoint 2200 Version I, giving the programmer a great deal of flexibility in creating a program.

A- The Accumulator register is used to hold the result of all arithmetic and logical instructions. All data transfers into or out of the computer take place through this register.

7	6	5	4	3	2	1	0	A Register

B, C, D, E - These are general purpose registers which may be used in conjunction with the Accumulator in arithmetic and logical operations. Each register may be loaded from or stored into memory or another register. When used in conjunction with the A and H, L registers, the B, C, D and E registers may function as indexes.

7	6	5	4	3	2	1	0	B,C,D,E, Registers

H, L - The H and L Registers are utilized to contain respectively the most significant portion (MSP) and least significant portion (LSP) of the address of a memory location being referenced. All memory reference instructions utilize these registers with the exception of CALL and JUMP commands. However, the H and L Registers may be used as general purpose registers when not being used as above.

H (8 bits)	L (8 bits)
MSP of address being referenced	LSP of address being referenced

P - The program register or "Location Counter" contains the address of the next instruction to be executed. This register is stored in the pushdown stack upon the execution of a "CALL" instruction and is loaded with the effective address upon execution of a "JUMP", "CALL" or "RETURN" instruction. The P register is 13 bits in length and is capable of addressing up to 8K of memory.

13	P (13 bits)	0

I - The I register is the register which holds the "operation code" of the instruction currently being executed. The contents of I are gated through a decoding network to determine what operation internal or external, is to be performed.

2.4 MEMORY

The basic Datapoint 2200 is supplied with 2048 eight-bit words of memory. Additional modules of 2048 words each may be incorporated with the total memory capacity of the processor being 8192 words. Each 2K memory is made up of 32 individual MOS shift registers with each one having a capacity of 512 bits or 64 eight-bit words. These registers are clocked at a rate of 1.2 MHz. Data is read out in it serial fashion with one word taking 8 microseconds. During this period of time, two clock pulse times are available for the processor to perform any necessary gating or testing functions.

The Datapoint 2200 memory might be likened to a drum type memory in some respects. It takes approximately 1/2 millisecond for the memory to completely circulate. Thus, if the current instruction referenced a memory location for data access, there would be a 1/2 millisecond delay before that instruction could be completed. However, unlike a drum memory the MOS memory may be stopped during instruction execution so that each succeeding instruction may be read from memory without delay (in 8 usec.).

Physically, instructions require a variable number of cycles for completion. In the first cycle, the instruction is fetched from memory and decoded. If the instruction involves no memory reference, it is then executed within 8 microseconds for a total completion time of 16 microseconds.

"Immediate" type instructions are the same as instructions requiring no memory reference and require a 16 usec. interval for the operand fetch and execute cycle. Jump and Call type instructions require a variable amount of time for execution, depending on the difference between the old and new locations.

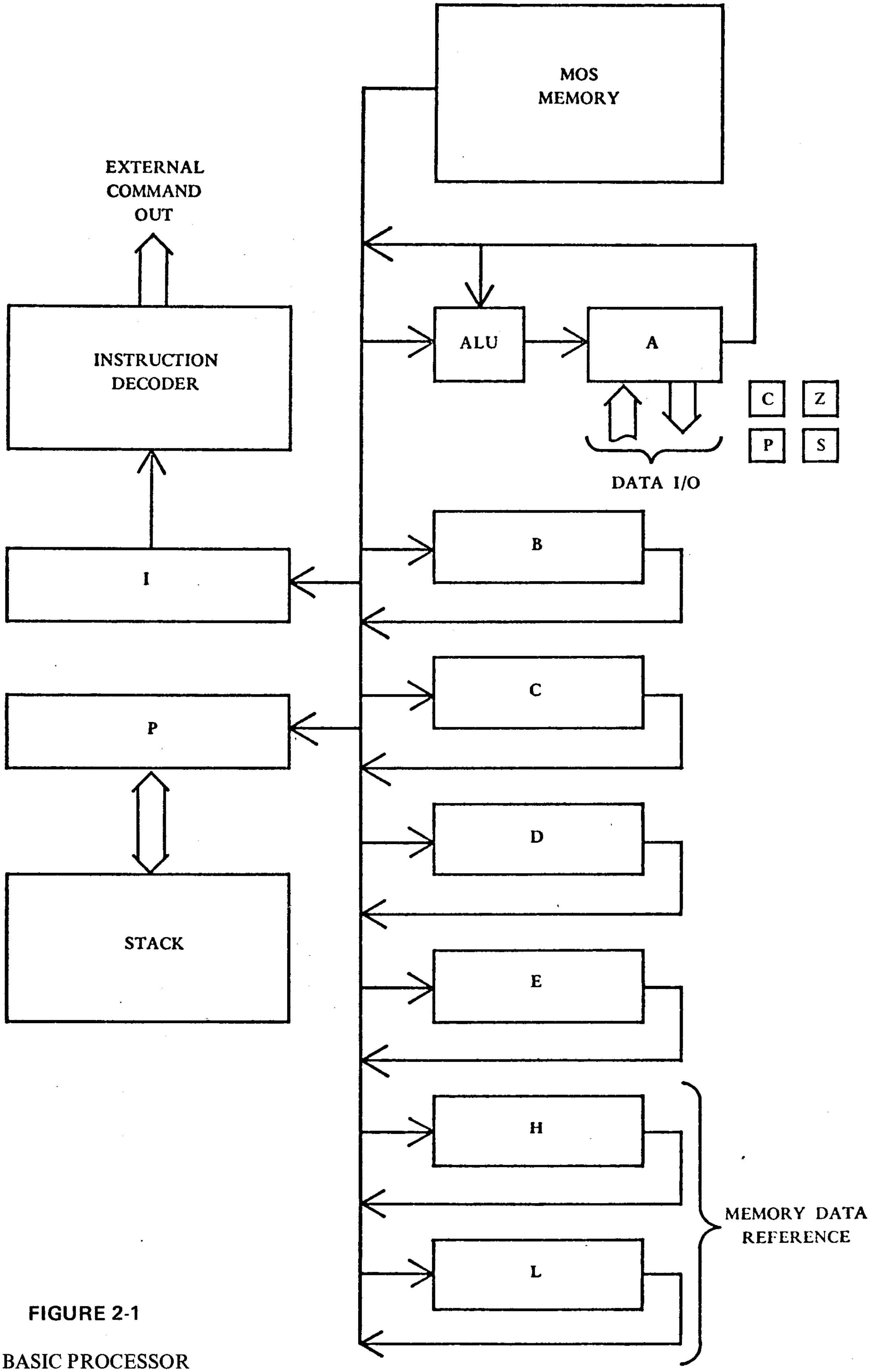


FIGURE 2-1

BASIC PROCESSOR
(VERSION I)

2.5 PUSHDOWN STACK

A unique feature of a machine this size is the incorporation into the processor's structure of a pushdown stack which is useful in any type of application which requires program subroutines. The stack automatically stores the contents of the P register upon execution of a "CALL" instruction and automatically restores P upon execution of a "RETURN". The stack is a group of "last-in/first-out" registers with a capacity of 15 CALLS. CALLS may be "nested", that is more than one CALL may be made before the execution of a RETURN. The execution of a "RETURN" will cause processor control to be given to the next instruction following the last executed CALL.

Pushdown Stack

Address of CALL 5	
Address of CALL 4	Maximum
Address of CALL 3	Capacity of
Address of CALL 2	15 CALLS
Address of CALL 1	

13 Bits

2.6 DATA FORMAT

Data is represented in the Datapoint 2200 in the form of 8-bit binary integers.

7	6	5	4	3	2	1	0

DATA WORD

2.7 CONTROL FLIP-FLOPS

Also contained in the basic processor are four control flops which reflect the state of the arithmetic logic unit and which can be tested through the execution of a conditional jump, call or return instruction. The flip-flop mnemonics with their associated functions are as follows:

C_f-Carry Flip-flop. Set when an arithmetic operation results in either a carry (add) or borrow (subtract). *The Carry Flip-flop also reflects the state of the most significant bit in the accumulator after completion of a shift right instruction. Likewise, it reflects the state of the accumulator least significant bit after completion of a shift left instruction.

Z_f-Zero Flip-flop. Set when the result of an arithmetic or logical operation is equal to zero.*

S_f-Sign Flip-flop. This flip-flop reflects the state of bit 7 in the accumulator after an arithmetic type operation.*

P_f-Parity Flip-flop. Indicates the parity or "number of one bits" contained in the accumulator. If this flip-flop is set (true), the A register contains an odd number of one bits; if it is reset (false), the A register contains an even number of one bits.*

*In the event of a compare instruction the contents of the accumulator are not changed; however, the control flip-flops reflect the equivalent of a subtract instruction.

2.8 INSTRUCTION FORMATS (GENERAL)

Instruction formats, dependent upon the operation to be performed, may be eight, sixteen or twenty-four bits in length.

Type-1 - register to register, memory reference, arithmetic, logical, shift instructions

OP CODE
8 bits

Type-2 - immediate mode instructions

OP CODE	OPERAND
8 bits	8 bits

Type-3 - JUMP & CALL instructions

OP CODE	ADDRESS
8 bits	16 bits

PART 3

BASIC PROCESSOR – VERSION II

3.1 PROCESSOR ORGANIZATION

The processor contained in the Datapoint 2200 Version II is comprised of the Arithmetic/Logic Unit, two sets of 7 program accessible registers, 2K to 16K words of read/write memory, an instruction decoder, a 16 level hardware push-down stack used in sub-routine type operations, and an interrupt system.

3.2 ARITHMETIC/LOGIC UNIT

The Arithmetic/Logic Unit is capable of processing both binary integers and logical operands. All arithmetic and logical operations may take place between the A-register and any of the 7 program accessible registers or between the A-register and memory. The A-register always contains the result of an arithmetic or logic operation and the other register (or memory cell) is unaffected. Arithmetic and logic operations affect the Sign, Carry, Zero and Parity Flip-Flops.

3.3 PROCESSOR REGISTERS

The Alpha and Beta modes of the processor each have 7 program accessible registers and 4 control flip flops. This gives the programmer the equivalent of 14 registers and 8 control flip flops. The registers and control flip flops of each mode are completely independent of each other allowing the registers of one mode to be altered without affecting the contents of the other.

To access the Beta mode registers and control flip flops, a select Beta mode instruction is executed. The Beta mode is then set until a select Alpha mode instruction is executed. The Alpha mode is automatically set when power is turned on, or the restart key depressed.

A- The Accumulator register is used to hold the result of all arithmetic and logical instructions. All data transfers into or out of the computer take place through this register.

7	6	5	4	3	2	1	0	A Register

B, C, D, E - These are general purpose registers which may be used in conjunction with the Accumulator in arithmetic and logical operations. Each register may be loaded from or stored into memory or another register. When used in conjunction with the A and H, L registers, the B, C, D and E registers may function as indexes.

7	6	5	4	3	2	1	0	B, C, D, E, Registers

H, L - The H and L Registers are utilized to contain respectively the most significant portion (MSP) and least significant portion (LSP) of the address of a memory location being referenced. All memory reference instructions utilize these registers with the exception of CALL and JUMP commands. However, the H and L Registers may be used as general purpose registers when not being used as above.

H (8 bits)	L (8 bits)
MSP of address being referenced	LSP of address being referenced

P - The program register or "Location Counter" contains the address of the next instruction to be executed. This register is stored in the pushdown stack upon the execution of a "CALL" instruction and is loaded with the effective address upon execution of a "JUMP", "CALL" or "RETURN' instruction. The P register is 14 bits in length and is capable of addressing up to 16K of memory.

13	P (14 bits)	0

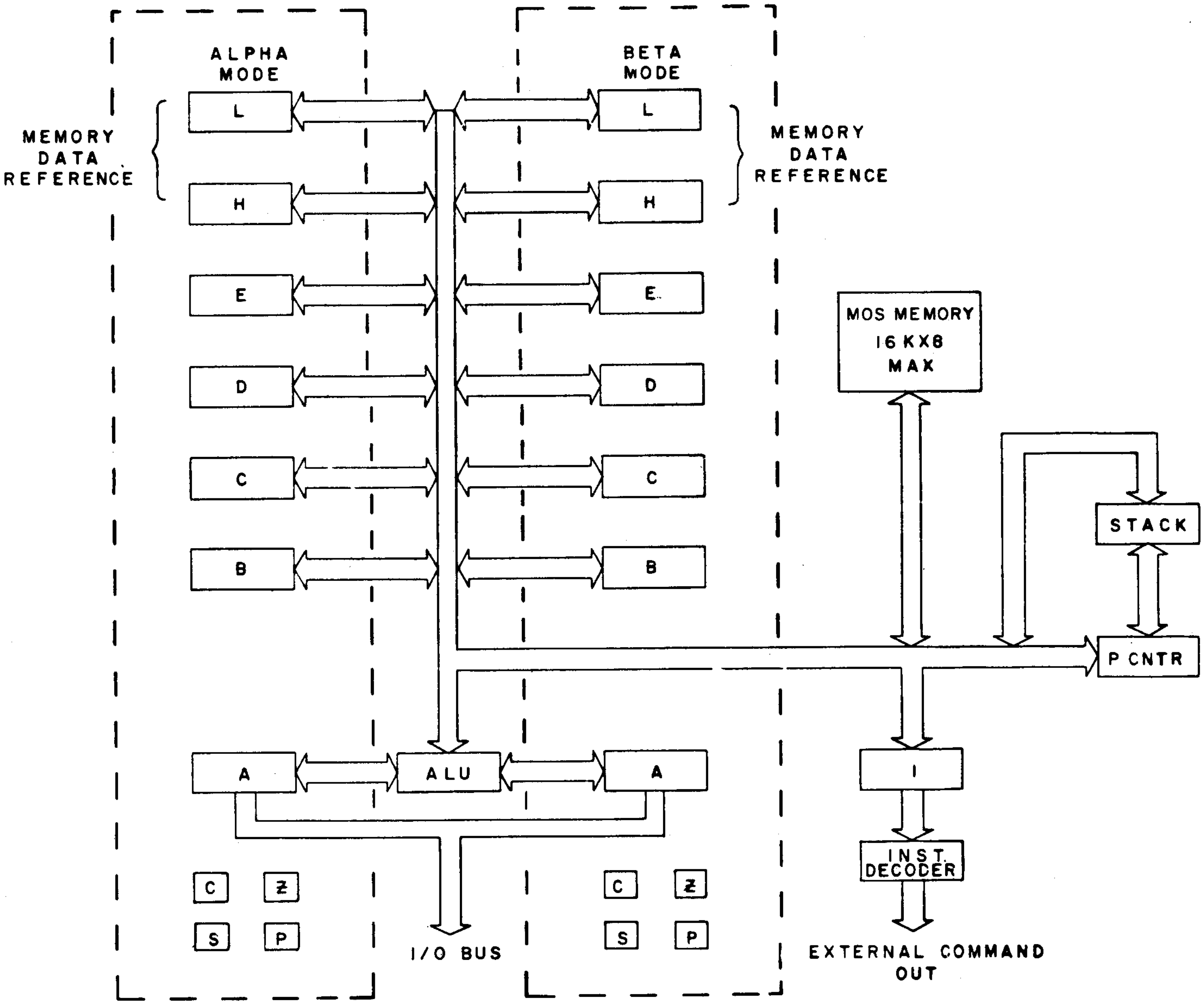
I - The I register is the register which holds the "operation code" of the instruction currently being executed. The contents of I are gated through a decoding network to determine what operation internal or external, is to be performed.

3.4 MEMORY

The basic (Version II) Datapoint 2200 is supplied with 4096 eight-bit words of memory. Additional modules of 4096 words each may be incorporated with the total memory capacity of the processor being 16384 words.

The Datapoint 2200 memory is a random access MOS memory with an access time of 500 nanoseconds. Each memory cycle takes 1.6 microseconds and each instruction takes 2, 3 or 4 cycles to complete. Due to the random access nature of the memory, the time required to read or write to the memory is the same regardless of the address.

Since the memory is a "Dynamic" MOS memory the contents of the memory must be rewritten or "refreshed" periodically. The refresh rate of the Datapoint 2200 memory is equal to once every 40 memory cycles. This means that one memory cycle out of every 40 will be used up to refresh the memory. The only effect this has



DATAPOINT 2200 (Version 2) Processor Organization

FIGURE 3-1
BASIC PROCESSOR (VERSION II)

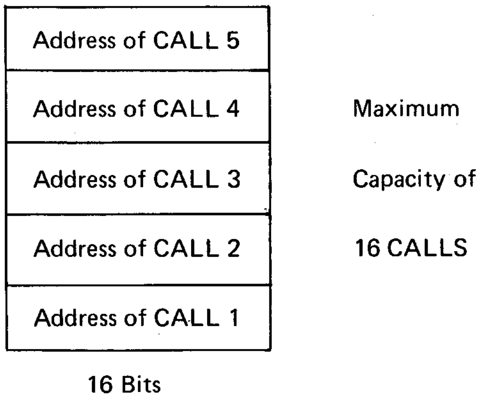
on processor operation is a slight slowing down of the processor (about 2.5%).

3.5 PUSHDOWN STACK

A unique feature of a machine this size is the incorporation into the processor's structure of a pushdown stack. This is useful for subroutine calling, saving the value of the H and L register pair while doing a memory reference to a different address, calculating an address and then jumping to it without having to overstore a JUMP instruction, making an abortive exit from a subroutine (returning control to a location other than the one after the CALL instruction), and saving the state of the machine if there is at least one free stack location.

Information may be transferred between either the P-counter and the stack or the H and L register pair and the stack. In the former case, only 14 bits are transferred with the two most significant bits always being zero; whereas, in the latter case, all 16 bits are transferred. The stack is actually a scratch pad of sixteen 16-bit words which is addressed by a four-bit up/down counter. Whenever a CALL or PUSH instruction is executed, the P-counter or H and L register pair is written into the stack word pointed out by the counter which is then incremented. The pointer end-arounds to 0 if it is incremented past 15. Whenever a RETURN or POP instruction is executed, the stack pointer is first decremented (ending around to 15 if it is decremented below 0) and then the P-counter or H and L register pair is loaded from the pointed location (this does not effect the contents of that stack location). Note that the above description implies that the maximum subroutine nesting depth is sixteen and will be less if data is also pushed onto the stack. That is, the seventeenth CALL or PUSH will overstore the value written in the first if no RETURN or POP instructions intervene.

Pushdown Stack



3.6 CONTROL FLIP-FLOPS

Also contained in the basic processor are eight control flip-flops (four in Alpha mode and four in Beta mode) which reflect the state of the arithmetic logic unit and which can be tested through the execution of a conditional jump, call or return instruction. The flip-flop mnemonics with their associated functions are as follows:

C_f-Carry Flip-flop. Set when an arithmetic operation results in either a carry (add) or borrow (subtract). *The Carry Flip-flop also reflects the state of the most significant bit in the accumulator after completion of a shift right instruction. Likewise, it reflects the state of the accumulator least significant bit after completion of a shift left instruction.

Z_f-Zero Flip-flop. Set when the result of an arithmetic or logical operation is equal to zero.*

S_f-Sign Flip-flop. This flip-flop reflects the state of bit 7 in the accumulator after an arithmetic type operation.*

P_f-Parity Flip-flop. Indicates the parity or "number of one bits" contained in the accumulator. If this flip-flop is set (true), the A register contains an odd number of one bits; if it is reset (false), the A register contains an even number of one bits.*

*In the event of a compare instruction the contents of the accumulator are not changed; however, the control flip-flops reflect the equivalent of a subtract instruction.

3.7 INTERRUPTS

Included in the Datapoint 2200 processor is a hardware interrupt feature. The interrupt signal occurs at a one millisecond rate. This one millisecond rate is accurate to within ±.005%.

To use the interrupts an "Enable Interrupts" instruction is executed. An interrupt can occur after the next instruction is executed. For example, in:

```
LAB
EI
RET
```

an interrupt cannot occur until after the return instruction has been executed. An interrupt is executed exactly as a

CALL Ø instruction would be. The current address is pushed onto the stack and the next instruction is picked up from location zero.

To prevent interrupts from occurring after they have been enabled,a "Disable Interrupt" instruction is issued. Interrupts are automatically disabled at power on, or whenever the restart key on the keyboard is depressed. If the interrupt system is in the "Disabled" mode, and the one millisecond time period expires, the one interrupt will be saved and will be executed the next time interrupts are enabled.

3.8 DATA FORMAT

Data is represented in the Datapoint 2200 in the form of 8-bit binary integers.

7	6	5	4	3	2	1	0

DATA WORD

3.9 INSTRUCTION FORMATS (GENERAL)

Instruction formats, dependent upon the operation to be performed, may be eight, sixteen or twenty-four bits in length.

Type-1 - register to register, memory reference, arithmetic, logical, shift instructions

OP CODE 8 bits

Type-2 - immediate mode instructions

OP CODE 8 bits	OPERAND 8 bits

Type-3 - JUMP & CALL instructions

OP CODE 8 bits	ADDRESS 16 bits

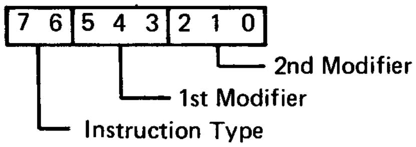
PART 4

INSTRUCTION REPERTOIRE

4.1 PRESENTATION FORMAT

This section gives a detailed description of each of the Datapoint 2200 instructions. The use and operations of each instruction is presented as follows:

FUNCTION:	Mnemonic Code
OPERATION:	Symbolic representation of instruction description.
OP CODE:	Operation Code, expressed in octal.
TIMING:	Execution time.
DESCRIPTION:	Definitión of function of the instruction.



INSTRUCTION FORMAT: Explanation of the function of each part of the instruction word.

NOTE

Considerations in instruction use and further definition of function.

Symbols and Abbreviations

The following symbols and abbreviations are used in the instruction format:

( )	the contents of
←	is replaced by
→	is transferred to
:	is compared with
A	8 bit arithmetic register (accumulator)
B C D E	8 bit general purpose registers
H	8 bit register used to specify most significant portion of operand address
L	8 bit register used to specify least significant portion of operand address
M	memory location designated by contents of H, L
r	one of the following register designators: A, B, C, D, E, H, L
r_s	designates operand source register (s=0-7)
r_d	designates operand destination register (d=0-7)
V	Logical "OR" operation
$\veebar$	Logical "exclusive-OR" operation
$\wedge$	Logical "AND" operation
STACK	Instruction counter (P) pushdown queue
P	Program counter
f_c	Flag flip-flop codes: C_f, Z_f, S_f, P_f
RR	Register to Register
IM	Immediate (from P+1)
MR	Memory Reference (Contents of memory location designated by H, L)
I	Instruction Register

TABLE 4-1

SOURCE AND DESTINATION CODES (s and d)

	s/d	SYMBOLIC CODE	
r_s/r_d	0	A	A Register
	1	B	B Register
	2	C	C Register
	3	D	D Register
	4	E	E Register
	5	H	H Register
	6	L	L Register
M	7	M	Memory location specified by contents of H&L

TABLE 4-2

FLIP-FLOP CODE (f_c)

c	SYMBOLIC CODE	NAME
0	C_f	Carry
1	Z_f	Zero
2	S_f	Sign
3	P_f	Parity

LOAD IMMEDIATE: Lr_d

OP CODE: 0d6 TIMING: 3.2 usec. (16 usec)*

OPERATION: $(P+1) \rightarrow r_d,\ P+2 \rightarrow P$

DESCRIPTION: Transfers the contents of the memory location immediately following the instruction, to the register specified by bits 3-5 of the instruction word.

INSTRUCTION FORMAT:

	P		P+1
7 6	5 4 3	2 1 0	7 0
0	d	6	OPERAND

d: is the destination designator
d=7: is not allowed

NOTE

1. The contents of P+1 are unchanged.
2. None of the Flag Flip-flops are affected.
3. Refer to Table 3-1 for destination codes.

LOAD: Lr_dM, Lr_dr_s, LMr_s

OP CODE: 3ds TIMING: 3.2 usec. (16 usec)

OPERATION: $(M) \rightarrow r_d$ $s=7,\ d \leqslant 6$ (Lr_dM)
$(r_s) \rightarrow r_d$ $s \leqslant 6,\ d \leqslant 6$ (Lr_dr_s)
$(r_s) \rightarrow M$ $s \leqslant 6,\ d=7$ (LMr_s)

DESCRIPTION: Transfers the operand from the source specified by bits 0-2 of the instruction word to the destination specified by bits 3-5 of the instruction word.

INSTRUCTION FORMAT:

	P	
7 6	5 4 3	2 1 0
3	d	s

d: designates the destination of data.

s: designates the source. If either s or d=7 a memory reference is indicated and the contents of registers H&L specify the address of the memory location.

NOTE

1. The data source is unaffected.
2. s & d both = 7 results in a Halt instruction.
3. None of the Flag Flip-flops are affected by execution of this instruction.
4. s=d results in a NOP, except as stated in Note 2.

*Note: Timing values in parentheses are for Version 1.

ADD IMMEDIATE: AD

OP CODE: 004 TIMING: 4.8 usec. (16 usec)

OPERATION: (A) + (P+1) → A, P+2 → P

DESCRIPTION: Adds to the contents of the A register the contents of the memory location immediately following the instruction, and retains the sum in the A register. Sets the C_f Flip-flop if ADD overflow occurs, otherwise resets C_f.

INSTRUCTION FORMAT:

P			P+1
7 6	5 4 3	2 1 0	7 0
0	0	4	OPERAND

NOTE

1. The Sign, Zero and Parity Flip-flops will indicate the status of the A register at completion.
2. The contents of P+1 are unchanged.
3. The Carry Flip-flop is cleared at the beginning of this instruction.

ADD WITH CARRY IMMEDIATE: AC

OP CODE: 014 TIMING: 4.8 usec. (16 usec)

OPERATION: (A) + (P+1) + (C_f) → A, P+2 → P

DESCRIPTION: Adds the C_f bit and the contents of the location immediately following the instruction to the contents of the A register, and retains the sum in the A register. If add overflow occurs, the C_f Flip-flop is set, otherwise C_f is reset.

INSTRUCTION FORMAT:

P			P+1
7 6	5 4 3	2 1 0	7 0
0	1	4	OPERAND

NOTE

1. The Sign, Zero and Parity Flip-flops will indicate the status of the A register at completion.
2. The contents of P+1 remain unchanged.

ADD: ADr$_s$ ADM

OP CODE: 20s TIMING: 3.2 (16) usec. if RR, 4.8 (520) usec. if MR

OPERATION: (A) + (r_s) → A or (A) + (M) → A

DESCRIPTION: This instruction is identical to ADD IMMEDIATE with the exception of operand source.

INSTRUCTION FORMAT:

P		
7 6	5 4 3	2 1 0
2	0	s

s: specifies the operand source. Refer to Table 3-1 for source codes.

ADD WITH CARRY: ACr$_s$ ACM

OP CODE: 21s TIMING: 3.2 (16) usec. if RR, (520) 4.8 usec. if MR

OPERATION: (A) + (C_f) + (r_s) → A or (A) + (C_f) + (M) → A

DESCRIPTION: This instruction is identical to ADD WITH CARRY IMMEDIATE with the exception of operand souce.

INSTRUCTION FORMAT:

P		
7 6	5 4 3	2 1 0
2	1	s

s: specifies the operand source. Refer to Table 3-1 for source codes.

NOTE: Timing values in parentheses are for Version I.

SUBTRACT IMMEDIATE: SU

OP CODE: 024 TIMING: 4.8 usec. (16 usec)

OPERATION: (A) - (P+1) → A, P+2 → P

DESCRIPTION: Subtracts the contents of the memory location immediately following the instruction from the contents of the A register, and retains the difference in the A register. The C_f Flip-flop is set if underflow occurs.

INSTRUCTION FORMAT:

P			P+1
7 6	5 4 3	2 1 0	7 0
0	2	4	OPERAND

NOTE

1. The contents of P+1 is unchanged.
2. The Zero, Sign, and Parity Flip-flops represent the status of the A register at the completion of this instruction.

SUBTRACT: SUr_s SUM

OP CODE: 22s TIMING: 3.2 (16) usec. if RR, 4.8 (520) usec. if MR

OPERATION: (A) - (r_s) → A or (A) - (M) → A

DESCRIPTION: This instruction is identical to SUBTRACT IMMEDIATE with the exception of operand source.

INSTRUCTION FORMAT:

P		
7 6	5 4 3	2 1 0
2	2	s

s: specifies the operand source. Refer to Table 3-1 for source codes.

SUBTRACT WITH BORROW IMMEDIATE: SB

OP CODE: 034 TIMING: 4.8 usec (16 usec)

OPERATION: (A) - (P+1) - (C_f) → A, P+2 → P

DESCRIPTION: Subtracts the contents of the memory location immediately following the instruction and the C_f bit, from the contents of the A register. Sets the C_f bit if underflow occurs, otherwise resets C_f.

INSTRUCTION FORMAT:

P			P+1
7 6	5 4 3	2 1 0	7 0
0	3	4	OPERAND

NOTE

1. The contents of P+1 are unchanged.
2. The Zero, Sign, and Parity Flip-flops represent the status of the A register at the completion of this instruction.

SUBTRACT WITH BORROW: SBr_s SBM

OP CODE: 23s TIMING: 3.2 (16) usec. if RR, 4.8 (520) usec. if MR

OPERATION: (A) - (r_s) - (C_f) → A or (A) - (M) - (C_f) → A

DESCRIPTION: This instruction is identical to SUBTRACT WITH BORROW IMMEDIATE with the exception of operand source.

INSTRUCTION FORMAT:

P		
7 6	5 4 3	2 1 0
2	3	s

s: specifies the operand source. Refer to Table 3-1 for source codes.

NOTE: Timing values in parentheses are for Version I.

AND IMMEDIATE: ND

OP CODE: 044 TIMING: 4.8 usec (16usec)

OPERATION: $(P+1) \wedge (A) \rightarrow A,\ P+2 \rightarrow P$

DESCRIPTION: Forms the logical product of the contents of the A register with the contents of the memory location immediately following the instruction, and places the results in the A register.

INSTRUCTION FORMAT:

	P		P+1
7 6	5 4 3	2 1 0	7 0
0	4	4	OPERAND

NOTE

1. The Carry Flip-flop will be reset upon completion of the operation.
2. The Zero, Sign, and Parity Flip-flops will represent the status of the A register upon completion of the operation.

SAMPLE OPERATION:

(A Reg)	0 0 0 0 1 1 1 1
(P+1)	0 1 1 0 0 1 1 0
(A Reg)	0 0 0 0 0 1 1 0

AND: NDr$_s$, NDM

OP CODE: 24s TIMING: 3.2 (16) usec. if RR, 4.8 (520) usec. if MR

OPERATION: $(A) \wedge (r_s) \rightarrow A$, or $(A) \wedge (M) \rightarrow A$

DESCRIPTION: This instruction is identical to AND IMMEDIATE with the exception of operand source.

INSTRUCTION FORMAT:

7 6	5 4 3	2 1 0
2	4	s

s: specifies the operand source. Refer to Table 3-1 for source codes.

OR IMMEDIATE: OR

OP CODE: 064 TIMING: 4.8 usec. (16 usec)

OPERATION: $(A) \vee (P+1) \rightarrow A,\ P+2 \rightarrow P$

DESCRIPTION: Forms the logical sum of the contents of the A register and the contents of the memory location immediately following the instruction, and places the result in the A register.

INSTRUCTION FORMAT:

	P		P+1
7 6	5 4 3	2 1 0	7 0
0	6	4	OPERAND

NOTE

1. The Carry Flip-flop will be reset at conclusion.
2. The Zero, Sign, and Parity Flip-flops will represent the status of the A register at completion of the operation.

SAMPLE OPERATION:

(A Reg)	0 0 0 0 1 1 1 1
(P+1)	0 1 1 0 0 1 1 0
(A Reg)	0 1 1 0 1 1 1 1

OR: ORr$_s$ ORM

OP CODE: 26s TIMING: 3.2 (16) usec. if RR 4.8 (520) usec. if MR

OPERATION: $(A) \vee (r_s) \rightarrow A$, or $(A) \vee (M) \rightarrow A$

DESCRIPTION: This instruction is identical to OR IMMEDIATE with the exception of operand source.

INSTRUCTION FORMAT:

	P	
7 6	5 4 3	2 1 0
2	6	s

s: specifies the operand source. Refer to Table 3-1 for source codes.

NOTE: Timing values in parentheses are for Version I.

EXCLUSIVE OR IMMEDIATE: XR

OP CODE: 054 TIMING: 4.8 (16) usec.

OPERATION: (A) ⊻ (P+1) → A, P+2 → P

DESCRIPTION: The logical difference of the contents of the A register and the contents of the memory location immediately following the instruction is formed, and the result replaces the contents of the A register.

INSTRUCTION FORMAT:

P			P+1
7 6	5 4 3	2 1 0	7 0
0	5	4	OPERAND

NOTE

1. The Carry Flip-flop will be reset at conclusion.
2. The Zero, Sign and Parity Flip-flops will represent the status of the A register upon completion of the operation.

SAMPLE OPERATION:

(A Reg)	0	0	1	1
(P+1)	0	1	0	1
(A Reg)	0	1	1	0

EXCLUSIVE OR: XRr_s XRM

OP CODE: 25s TIMING: 3.2 (16) usec. if RR, 4.8 (520) usec. if MR

OPERATION: (A) V (r_s) → A, (A) V (M) → A

DESCRIPTION: This instruction is identical to EXCLUSIVE OR IMMEDIATE with the exception of operand source.

INSTRUCTION FORMAT:

P		
7 6	5 4 3	2 1 0
2	5	s

s: specifies the operand source. Refer to Table 3-1 for source codes.

COMPARE IMMEDIATE: CP

OP CODE: 074 TIMING: 4.8 (16) usec.

OPERATION: (A) : (P+1), P+2 → P

DESCRIPTION: Compares the contents of the A register with the contents of the memory location immediately following the instruction. The flag flip-flops assume the same state as they would for a Subtract instruction.

INSTRUCTION FORMAT:

P			P+1
7 6	5 4 3	2 1 0	7 0
0	7	4	OPERAND

NOTE

1. The contents of the A register are unaffected.

COMPARE: CPr_s CPM

OP CODE: 27s TIMING: 3.2 (16) usec. if RR, 4.8 (520) usec. if MR

OPERATION: (A) : (r_s) or (A) : (M)

DESCRIPTION: This instruction is identical to COMPARE IMMEDIATE with the exception of operand source.

INSTRUCTION FORMAT:

P		
7 6	5 4 3	2 1 0
2	7	s

s: specifies the operand source. Refer to Table 3-1 for source codes.

NOTE: Timing values in parentheses are for Version I.

UNCONDITIONAL JUMP: JMP ,

OP CODE: 104 TIMING: 6.4 usec. (See Note for Version I)

OPERATION: (P+1, P+2) → P

DESCRIPTION: An unconditional transfer of control. The contents of P+1 represent the least significant portion of the address, while the contents of P+2 represent the most significant portion.

INSTRUCTION FORMAT:

P			P+1	P+2
7 6	5 4 3	2 1 0	7 0	7 0
1	0	4	LSP	MSP
	OP CODE		ADDRESS	

The three high order bits in the address are ignored, the remaining 13 bits specify the address to which control is to be transferred.

NOTE: *Timing is variable dependent upon cyclic difference between instruction and effective address locations. Version I Only.

JUMP IF CONDITION TRUE: JTf_C

OP CODE: 1(c+4)0 TIMING: 6.4 (Variable) usec. if condition true, 4.8 (24) usec. if condition false.

OPERATION: If (f_C=TRUE), (P+1 , P+2) → P. Otherwise, P+3 → P.

DESCRIPTION: Examines the designated flip-flop. If set, transfers control to the address designated by the contents of the two memory locations immediately following the instruction. If the selected flip-flop is reset, executes the next sequentially available instruction.

INSTRUCTION FORMAT:

P			P+1	P+2
7 6	5 4 3	2 1 0	7 0	7 0
1	c+4	0	LSP	MSP
	OP CODE		ADDRESS	

c: designates which flip-flop condition is to be tested. Refer to Table 3-2 for list of Flip-flop codes.

NOTE

1. The condition of the selected Flip-flop is unchanged by this instruction.

NOTE: Timing values in parentheses are for Version I.

JUMP IF CONDITION FALSE: JFf_c

OP CODE: 1c0 TIMING: 6.4 (Variable) usec. if condition false, 4.8 (24) usec. if condition true.

OPERATION: If (f_c=FALSE), (P+1, P+2) → P. Otherwise P+3 → P.

DESCRIPTION: Examines the designated flip-flop. If reset, transfers control to the address designated by the contents of the two memory locations immediately following the instruction. If the selected flip-flop is set, executes the next sequentially available instruction.

INSTRUCTION FORMAT:

P			P+1	P+2
7 6	5 4 3	2 1 0	7 0	7 0
1	c	0	LSP	MSP
OP CODE			ADDRESS	

c: designates which flip-flop (condition) is to be tested. Refer to Table 3-2 for list of flip-flop codes.

NOTE

1. The condition of the selected flip-flop is unchanged by this instruction.

SUBROUTINE CALL: CALL

OP CODE: 106 TIMING: 6.4 usec. (Variable)

OPERATION: P+3 → STACK, (P+1, P+2) → P

DESCRIPTION: Transfers the address of the next sequentially available instruction to the Pushdown Stack, and transfer control to the address specified by the contents of the two memory locations immediately following the Op Code.

INSTRUCTION FORMAT:

P			P+1	P+2
7 6	5 4 3	2 1 0	7 0 0	7 0
1	0	6	LSP	MSP
			ADDRESS	

NOTE

1. The Stack is open-ended in operation. If it is overfilled, the deepest address will be lost.

NOTE: Timing values in parentheses are for Version I.

CONDITIONAL SUBROUTINE CALL

IF CONDITION TRUE: CTf_c

OP CODE: 1(c+4)2

TIMING: 6.4 (Variable) usec. if condition true, 4.8 (24) if condition false.

OPERATION: If (f_c=TRUE), P+3 → STACK, (P+1, P+2) → P. Otherwise, P+3 → P.
DESCRIPTION: Examines the designated flip-flop. If set, transfers the address of the next sequentially available instruction to the pushdown stack, and transfers control to the address of the two memory locations immediately following the Op Code. If the selected flip-flop is reset, executes the next sequentially available instruction.

INST.RUCTION FORMAT:

			P+1	P+2
7 6	5 4 3	2 1 0	7 0	7 0
1	c+4	2	LSP	MSP

ADDRESS

c: designates which flip-flop (condition) is to be tested.

NOTE

1. The condition of the selected flip-flop is unchanged by this instruction.
2. The stack is open-ended in operation. If it is overfilled, the deepest address will be lost.
3. Refer to Table 3-2 for list of flip-flop codes.

NOTE: Timing values in parentheses are for Version I.

CONDITIONAL SUBROUTINE CALL

IF CONDITION FALSE: CFf_c

OP CODE: 1c2

TIMING: 6.4 (Variable) usec. if condition false, 4.8 (24) usec. if condition true.

OPERATION: If (f_c=FALSE), P+3 → STACK, (P+1, P+2) → P.
DESCRIPTION: Examines the designated flip-flop. If reset, transfers the address of the next sequentially available instruction to the pushdown stack, and transfers control to the address of the two memory locations immediately following the Op Code. If the selected flip-flop is set, executes the next sequentially available instruction.

INSTRUCTION FORMAT:

			P+1	P+2
7 6	5 4 3	2 1 0	7 0	7 0
1	c	2	LSP	MSP
			ADDRESS	

c: designates which flip-flop (condition) is to be tested.

NOTE

1. The condition of the selected flip-flop is unchanged by this instruction.
2. The stack is open-ended in operation. If it is overfilled, the deepest address will be lost.
3. Refer to Table 3-2 for list of flip-flop codes.

NOTE: Timing values in parentheses are for Version I.

SUBROUTINE RETURN: RETURN

OP CODE: 007 TIMING: 3.2 usec. (Variable)

OPERATION: (STACK) → P

DESCRIPTION: Transfer control to the address specified by the most recent entry in the Pushdown Stack. Deletes the most recent entry from the Stack.

INSTRUCTION FORMAT:

P

7 6	5 4 3	2 1 0
0	0	7

NOTE

1. The effect of attempting more "RETURN" than the Stack is capable of handling is undefined.

CONDITIONAL SUBROUTINE RETURN

IF CONDITION TRUE: RTf_c

OP CODE: 0(c+4)3 TIMING: 3-2 usec. (Variable if condition true, 16 usec. if condition false)

OPERATION: If (f_c=TRUE), Stack → P. Otherwise P+1 → P

DESCRIPTION: Examines the designated flip-flop. If set, transfers control to the address specified by the most recent entry in the pushdown stack. Deletes the most recent entry in the stack. If the selected flip-flop is reset, executes the next sequentially available instruction.

INSTRUCTION FORMAT:

7 6	5 4 3	2 1 0
1	c+4	3

c: designates which flip-flop (condition) is to be tested.

NOTE

1. The condition of the selected flip-flop is unchanged by this instruction.
2. The effect of attempting more "RETURN" than the stack is capable of handling is undefined.
3. Refer to Table 3-2 for list of flip-flop codes.

CONDITIONAL SUBROUTINE RETURN

IF CONDITION FALSE: RFf_c

OP CODE: 0c3 TIMING: 3.2 usec. (Variable if condition false, 16 usec. if condition true.)

OPERATION: If (f_c= FALSE), Stack → P. Otherwise, P+1 → P

DESCRIPTION: Examines the designated flip-flop. If reset, transfers control to the address specified by the most recent entry in the stack. If the selected flip-flop is set, executes the next sequentially available instruction.

INSTRUCTION FORMAT:

7 6	5 4 3	2 1 0
1	c	3

c: designates which flip-flop (condition) is to be tested.

NOTE

1. The condition of the selected flip-flop is unchanged by this instruction.
2. The effect of attempting more "RETURN" than the stack is capable of handling is undefined.
3. Refer to Table 3-2 for list of flip-flop codes.

Note: Timing values in parentheses are for Version 1.

SHIFT RIGHT CIRCULAR: SRC

OP CODE: 012 TIMING: 3.2 (16) usec.

OPERATION: $A_m \rightarrow A_{m-1}, A_o \rightarrow A_7, A_o \rightarrow C_f$

DESCRIPTION: Shifts the contents of the A register right in a circular fashion. Shifts the least significant bit into the most significant bit position. Upon completion of the operation, the Carry Flip-flop is equal to the most significant bit.

INSTRUCTION FORMAT:

P

7 6	5 4 3	2 1 0
0	1	2

NOTE

None of the flag flip-flops other than C_f is affected by this instruction.

SHIFT LEFT CIRCULAR: SLC

OP CODE: 002 TIMING: 3.2 (16) usec.

OPERATION: $A_m \rightarrow A_{m+1}, A_7 \rightarrow A_o, A_7 \rightarrow C_f$

DESCRIPTION: Shifts the contents of the A register left in a circular fashion. Shifts the most significant bit into the least significant bit position. Upon completion of the operation, the Carry Flip-flop is equal to the least significant bit.

INSTRUCTION FORMAT:

P

7 6	5 4 3	2 1 0
0	0	2

NOTE

None of the flag flip-flops other than C_f is affected by this instruction.

NO OPERATION: NOP

OP CODE: 300 TIMING: 3.2 (16) usec.

OPERATION: P+1 → P

DESCRIPTION: No instruction is executed.

INSTRUCTION FORMAT:

P

7 6	5 4 3	2 1 0
3	0	0

HALT: HALT

OP CODE: 000,001,377 TIMING: Execution Stops

OPERATION:

DESCRIPTION: The computer halts. When the START buttom on the console is depressed, operation resumes at P+1.

INSTRUCTION FORMAT:

P

7 6	5 4 3	2 1 0
0	0	0
0	0	1
3	3	7

INPUT: INPUT

OP CODE: 101 TIMING: 9.6 (16) usec.

OPERATION: (I/O Bus) → A

DESCRIPTION: Transfers the contents of the I/O Bus to the A register.

INSTRUCTION FORMAT:

P

7 6	5 4 3	2 1 0
1	0	1

NOTE: Timing values in parentheses are for Version I.

INSTRUCTIONS BELOW APPLY TO VERSION II ONLY

POP: POP

OP CODE: 060 TIMING: 4.8 usec.

OPERATION: Stack → H, L

DESCRIPTION: Transfers the most recent entry of the stack into the H & L registers. H = MSP, L = LSP

INSTRUCTION FORMAT:

7 6	5 4 3	2 1 0
0	6	0

PUSH: PUSH

OP CODE: 070 TIMING: 3.2 usec.

OPERATION: H,L → Stack

DESCRIPTION: Transfers the contents of the H & L registers into the PUSHDOWN stack. H = MSP,
L = LSP

INSTRUCTION FORMAT:

7 6	5 4 3	2 1 0
0	7	0

ENABLE INTERRUPTS: EI

OP CODE: 050 TIMING: 3.2 usec.

DESCRIPTION: Following the next instruction, will allow interrupts to occur until a DISABLE INTERRUPT instruction is executed.

INSTRUCTION FORMAT:

7 6	5 4 3	2 1 0
0	5	0

DISABLE INTERRUPTS: DI

OP CODE: 040 TIMING: 3.2 usec.

DESCRIPTION: Prevents interrupts from occurring until an ENABLE INTERRUPT instruction is executed.

INSTRUCTION FORMAT:

7 6	5 4 3	2 1 0
0	4	0

SELECT ALPHA MODE: ALPHA

OP CODE: 030 TIMING: 3.2 usec.

DESCRIPTION: Selects the ALPHA MODE registers and control Flip Flops.

INSTRUCTION FORMAT:

7 6	5 4 3	2 1 0
0	3	0

SELECT BETA MODE BETA

OP CODE: 020 TIMING: 3.2 usec.

DESCRIPTION: Selects the BETA MODE registers and control Flip Flops.

INSTRUCTION FORMAT:

7 6	5 4 3	2 1 0
0	2	0

EXTERNAL COMMAND: EX (exp)

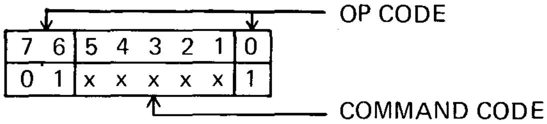
OP CODE: 121-177 depending on the specific command being executed.

TIMING: 9.6 (16) usec.

OPERATION: Performs I/O control functions according to (exp)

DESCRIPTION: These instructions perform the functions necessary for control of the I/O system and external devices. Many of these functions are specifically related to operation of particular devices. The device oriented commands for the Keyboard, CRT Display, Cassette Tapes, and Communications Interface are explained in the sections covering these devices.

INSTRUCTION FORMAT:



7 6	5 4 3 2 1	0
0 1	x x x x x	1

Table 3-3 is a list of External Commands used. For a detailed discussion of their use, reference should be made to Part 4 (Input/Output Operations) and to descriptions of the separate external devices.

TABLE 4-3

EXTERNAL COMMANDS

EX (exp)

COMMAND NUMBER	(exp)	OCTAL CODE	COMMAND	DESCRIPTION	DEVICE ADDRESS
1	ADR	121	Address	Selects device specified by A-register	ALL
2	STATUS	123	Sense Status	Connects selected device status to input lines	
3	DATA	125	Sense Data	Connects selected device data to input lines	
4	WRITE	127	Write Strobe	Signals selected device that output data word is on output lines	
5	COM1	131	Command 1	Outputs a control function to selected device	
6	COM2	133	Command 2	Outputs a control function to selected device	
7	COM3	135	Command 3	Outputs a control function to selected device	
8	COM4	137	Command 4	Outputs a control function to selected device	ALL
9	——	141	(Unassigned)	——	——
10	——	143	(Unassigned)	——	——
11	——	145	(Unassigned)	——	——
12	——	147	(Unassigned)	——	——

TABLE 4-3

EXTERNAL COMMANDS

EX (exp)

(Continued)

COMMAND NUMBER	(exp)	OCTAL CODE	COMMAND	DESCRIPTION	DEVICE ADDRESS
13	BEEP	151	Beep	Activates tone producing mechanism	341
14	CLICK	153	Click	Activates audible click producing mechanism	341
15	DECK1	155	Select Deck 1	Connects deck 1 to I/O bus	360
16	DECK2	157	Select Deck 2	Connects deck 2 to I/O bus	│
17	RBK	161	Read Block	Enables read circuitry and sets tape in forward motion	│
18	WBK	163	Write Block	Enables write circuitry and sets tape in forward motion	360
19	——	165	(Unassigned)	——	——
20	BSP	167	Backspace One Block	Backs up the selected tape one record	360
21	SF	171	Slew Forward	Sets selected tape deck in forward motion	│
22	SB	173	Slew Backward	Sets selected tape deck in backward motion	│
23	REWIND	175	Rewind	Rewinds the selected deck to beginning of tape	│
24	TSTOP	177	Stop Tape	Halts motion of the selected tape deck	360

TABLE 4-4

INSTRUCTION REPERTOIRE

OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC
000	HALT	050	* EI	120	JFS
001	HALT	051		121	EX ADR
002	SLC	052		122	CFS
003	RFC	053	RTZ	123	EX STATUS
004	AD	054	XR	124	
005		055		125	EX DATA
006	LA	056	LH	126	
007	RETURN	057		127	EX WRITE
010		060	* POP	130	JFP
011		061		131	EX COM1
012	SRC	062		132	CFP
013	RFZ	063	RTS	133	EX COM2
014	AC	064	OR	134	
015		065		135	EX COM3
016	LB	066	LL	136	
017		067		137	EX COM4
020	* BETA	070	* PUSH	140	JTC
021		071		141	
022		072		142	CTC
023	RFS	073	RTP	143	
024	SU	074	CP	144	
025		075		145	
026	LC	076		146	
027		077		147	
030	* ALPHA	100	JFC	150	JTZ
031		101	INPUT	151	EX BEEP
032		102	CFC	152	CTZ
033	RFP	103		153	EX CLICK
034	SB	104	JMP	154	
035		105		155	EX DECK1
036	LD	106	CALL	156	
037		107		157	EX DECK2
040	* DI	110	JFZ	160	JTS
041		111		161	EX RBK
042		112	CFZ	162	CTS
043	RTC	113		163	EX WBK
044	ND	114		164	
045		115		165	
046	LE	116		166	
047		117		167	EX BSP

* VERSION II ONLY

TABLE 4-4

INSTRUCTION REPERTOIRE

(Continued)

OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC
170	JTP	240	NDA	310	LBA
171	EX SF	241	NDB	311	
172	CTP	242	NDC	312	LBC
173	EX SB	243	NDD	313	LBD
174		244	NDE	314	LBE
175	EX REWND	245	NDH	315	LBH
176		246	NDL	316	LBL
177	EX TSTOP	247	NDM	317	LBM
200	ADA	250	XRA	320	LCA
201	ADB	251	XRB	321	LCB
202	ADC	252	XRC	322	
203	ADD	253	XRD	323	LCD
204	ADE	254	XRE	324	LCE
205	ADH	255	XRH	325	LCH
206	ADL	256	XRL	326	LCL
207	ADM	257	XRM	327	LCM
210	ACA	260	ORA	330	LDA
211	ACB	261	ORB	331	LDB
212	ACC	262	ORC	332	LDC
213	ACD	263	ORD	333	
214	ACE	264	ORE	334	LDE
215	ACH	265	ORH	335	LDH
216	ACL	266	ORL	336	LDL
217	ACM	267	ORM	337	LDM
220	SUA	270	CPA	340	LEA
221	SUB	271	CPB	341	LEB
222	SUC	272	CPC	342	LEC
223	SUD	273	CPD	343	LED
224	SUE	274	CPE	344	
225	SUH	275	CPH	345	LEH
226	SUL	276	CPL	346	LEL
227	SUM	277	CPM	347	LEM
230	SBA	300	NOP	350	LHA
231	SBB	301	LAB	351	LHB
232	SBC	302	LAC	352	LHC
233	SBD	303	LAD	353	LHD
234	SBE	304	LAE	354	LHE
235	SBH	305	LAH	355	
236	SBL	306	LAL	356	LHL
237	SBM	307	LAM	357	LHM

TABLE 4-4

INSTRUCTION REPERTOIRE

(Continued)

OP CODE	MNEMONIC	OP CODE	MNEMONIC
360	LLA	370	LMA
361	LLB	371	LMB
362	LLC	372	LMC
363	LLD	373	LMD
364	LLE	374	LME
365	LLH	375	LMH
366		376	LML
367	LLM	377	HALT

NOTE

OP Codes shown without Mnemonics are undefined.

PART 5

INPUT/OUTPUT OPERATIONS

5.1 GENERAL

The versatile input/output capability of the Datapoint 2200 permits it to communicate with external devices (such as the 2200 communications adaptor) through a parallel I/O system. The keyboard, CRT and tape decks that are an integral part of the Datapoint 2200 perform all operations over the same I/O system as external devices.

5.2 INPUT/OUTPUT INSTRUCTIONS

Two types of instructions provide for I/O operations. One is the INPUT command (see section 3) which, upon execution, transfers whatever is on the input bus to the A-register. The second is the EXTERNAL command which is sub-divided into 24 separate command operations (8 of which are available to devices physically external to the Datapoint 2200). Each EXTERNAL command produces a strobe pulse which may be used for control external to the processor. The actual control functions assigned to each external command are listed in Table 3-3.

5.3 INPUT/OUTPUT CABLE

The parallel I/O cable carries data, input strobe, external commands, and power between the 2200 processor and external devices connected to it. A complete I/O system is structured by connecting external devices in partyline fashion as shown in Figure 2-1. The I/O cable contains 8 input data lines, 8 output data lines, 1 input strobe line, 8 (of the 24) external command lines, 1 clock line, and 7 power and ground lines.

5.4 I/O DATA LINES

The data lines are broken into two groups. 8 lines are used for output and 8 lines are used for input.

The data output lines are connected (at all times) to the A-register in the processor and are used to perform three basic functions:

a. To transfer an address to select an external device (including the keyboard, c.r.t. and tape decks);

b. To transfer commands to an addressed device; and

c. To transfer data to an addressed device.

The data input lines are strobed into the A-register upon execution of the INPUT instruction and used to perform two basic functions:

a. To transfer status information from an addressed external device; and

b. To transfer data from an addressed external device.

As shown in Figure 4-1, input data or status from the data input lines is processed through input receivers and gated into the A-register. Once in the A-register data can be manipulated or stored as desired. Addresses, commands, or data that is to be transferred to an external device must first be loaded into the A-register. From the A-register it is transmitted through output devices onto the data output lines. The A-register is then used as a buffer register between the 2200 processor and external devices for all input and output data transfers.

5.5 INPUT STROBE

The INPUT STROBE carries a signal (8 usec pulse Version I, 4.8 usec pulse Version II) from the processor to the external device to indicate that whatever data is on the data input lines has been sampled and transferred into the A-register. The trailing edge of the pulse may be used by an external device to remove data from the data input line or to clear a status bit. The INPUT strobe is generated upon execution of the INPUT instruction.

In Version I, data transfer into the A-register occurs at the trailing edge of the INPUT STROBE. In Version II, this transfer occurs 400 nanoseconds prior to the leading edge of the input strobe.

5.6 EXTERNAL COMMAND STROBES

The eight EXTERNAL commands used by devices physically external to the Model 2200 are given function assignments as follows:

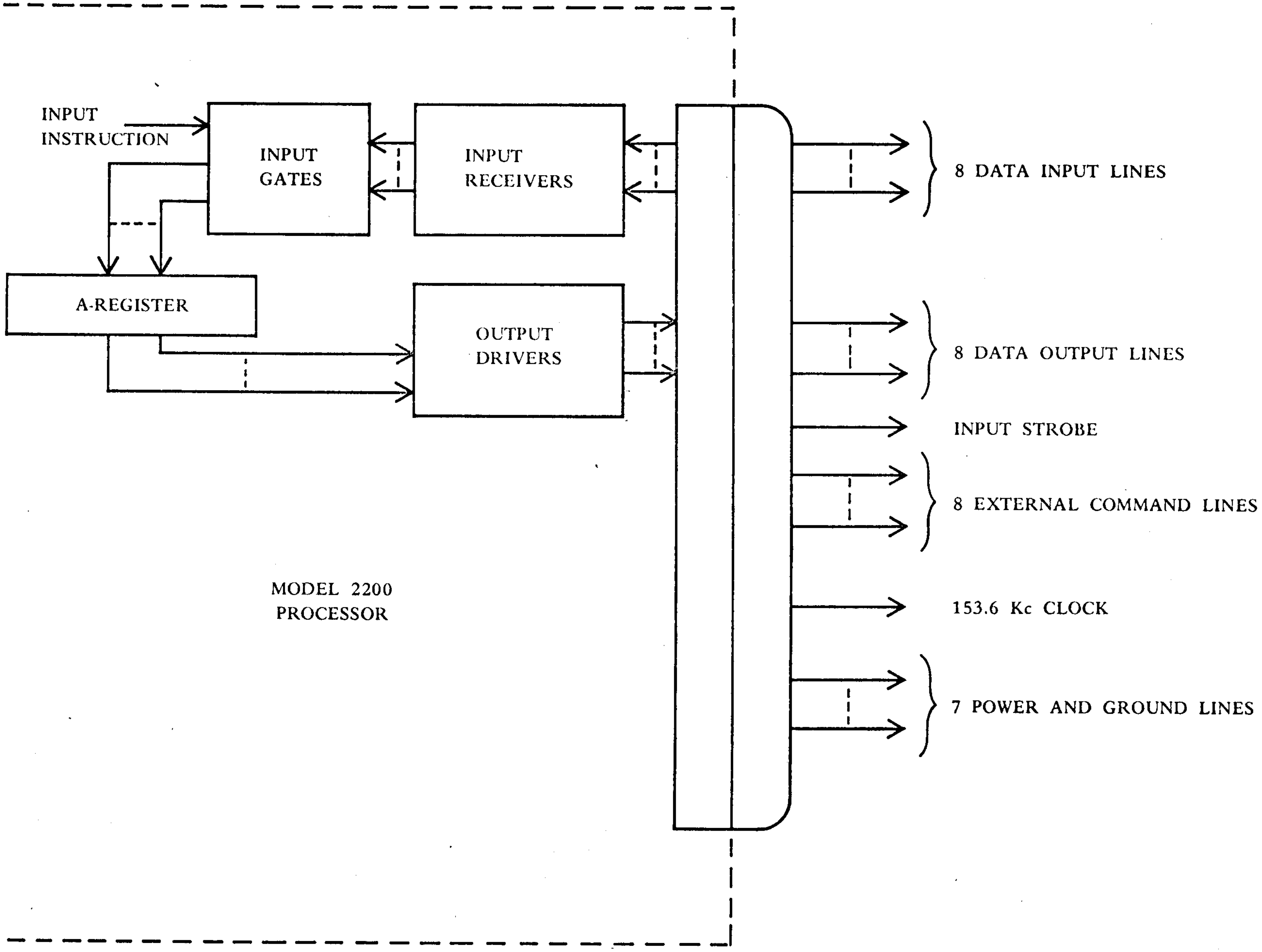


FIGURE 5-1

I/O SYSTEM, FUNCTIONAL DIAGRAM

TABLE 5-1

EXTERNAL COMMANDS

EX (exp)

COMMAND NUMBER	(exp)	OCTAL CODE	COMMAND	DESCRIPTION
1	ADR	121	Address	Selects device specified by A-register
2	STATUS	123	Sense Status	Connects selected device status lines to data input bus
3	DATA	125	Sense Data	Connects selected device data lines to data input bus
4	WRITE	127	Write Strobe	Signals selected device that output data is on data output lines
5	COM1	131	Command 1	Signals selected device that a control word is on data output lines
6	COM2	133	Command 2	Signals selected device that a control word is on data output lines
7	COM3	135	Command 3	Signals selected device that a control word is on data output lines
8	COM4	137	Command 4	Signals selected device that a control word is on data output lines

Execution of an EXTERNAL instruction provides a pulse 4.8 microseconds long. No functions are performed within the 2200 processor during execution of an EXTERNAL instruction. The interpretation of each of the EXTERNAL instructions is as follows:

a. Address. The address command (EX ADR) is a signal from the processor to all external devices to indicate that the information on the data output bus is to be interpreted as an external device address. Whenever an address command is executed all external devices should be disconnected from the I/O system except the device whose address appears in the A-register. (See paragraph 4.10 for discussion of address assignments).

b. Sense Status. The sense status (EX STATUS) command is a signal from the processor to the selected external device to place status information on the data input lines. (Note: External devices should be configured such that status is connected to the data input line whenever the device is first addressed. It is only necessary to use the EX STATUS instruction when it is desired to sense status after an EX DATA instruction has been used and a new address sequence has not been executed).

c. Sense Data. The sense data (EX DATA) command is a signal from the processor to the selected external device to place its data on the data input lines.

d. Write Strobe. The write strobe (EX WRITE) command is a signal from the processor that data is present on the data output lines for the selected external device.

e. Command 1 through Command 4. Command 1 through Command 4 (EX COM1, etc.) have meaning appropriate to the device selected. Reference should be made to a description of each device for specific function assignments.

5.7 CLOCK LINE

The clock line is crystal controlled 153.6 kilohertz squarewave that is available to external devices for timing purposes.

5.8 I/O BUS ELECTRICAL SPECIFICATIONS

All signals in the I/O System operate with a voltage swing of zero to +5 volts. Line drivers have a source impedance of approximately 470 ohms and line receivers have an input impedance in excess of 18,000 ohms and a decision threshold of +1.7 volts. Figure 4-2 illustrates a typical output line circuit.

All logic levels are True (logical 1) for zero (less than +1.7) volts and False (logical 0) for +5 (greater than 1.7) volts.

5.9 DATA TRANSFER OPERATION

a. Data Output. Figure 4-3 illustrates the sequence of events that occur when data is transferred from the

2200 processor to an external device. A typical program sequence to execute a transfer is as follows:

```
WDATA   LA 0322      Load device address into A-
                     register
        EX ADR
        INPUT        Load device status into A-register
        SRC          Shift desired status bit into C
                     flip-flop
        JFC EXIT     Exit if device not ready
        LAM          Load A-register with DATA
        EX WRITE     Write Data to device
```

Once a device is addressed it remains addressed until another device is addressed so that succeeding commands may be transmitted to a device without re-addressing the device. Transmitting a command to a device would follow a program sequence similar to a data transfer except that EX COM_n would replace EX WRITE.

b. Data Input. Figure 4-4 illustrates a sequence of events that occur when data is transferred from an external device to the 2200 processor. A typical program sequence is as follows:

```
RDATA   LA 0322      Load A-register with device
                     address
        EX ADR
        INPUT        Load device status into A-register
        SRC          Shift status bit into C flip-flop
        SRC
        JFC EXIT     Exit if device not ready
        EX DATA      Place data on input lines
        INPUT        Load A-register with data
```

5.10 DEVICE ADDRESS NUMBERING

Address assignments in the I/O system provides for up to 16 devices external to the 2200 processor. The address word is formulated such that the low-order four bits form the binary value for the address and the high-order four bits form the logical complement of the low order bits. For example device number 6 would have an address word as follows:

$$\underbrace{\underbrace{\boxed{1}\boxed{0}\boxed{0}\boxed{1}}_{\text{LOGICAL COMPLEMENT BINARY 6}}\underbrace{\boxed{0}\boxed{1}\boxed{1}\boxed{0}}_{\text{BINARY 6}}}_{\text{OCTAL 226}} \quad \text{A-Register}$$

This addressing system permits any device to be coded for its particular address with only a four-input gate strapped to those output lines that are set to one during the address command.

Device addresses used in the Model 2200 are given in the following table:

TABLE 5-2

DEVICE ADDRESS ASSIGNMENTS

DEVICE	NUMBER	BINARY	OCTAL
Cassette Tape Decks	0	11110000	360
CRT/Keyboard	1	11100001	341
Asynchronous Communication Adaptor	2	11010010	322
2200/Printers	3	11000011	303
2200/Tape	4	10110100	264
Synchronous Communications Adaptor	5	10100101	245
Parallel Communications Adaptor	6	10010110	226
Card Reader	7	10000111	207
Disk Controller	8	01111000	170
Additional Communications Adaptor	9	01101001	151
Additional Printer	10	01011010	132
Additional Tape/ Disk	11	01001011	113
Unassigned	12	00111100	074
"	13	00101101	055
"	14	00011110	036
"	15	00001111	017

5.11 I/O POWER AND GROUND LINES

The Model 2200 provides several power supply voltages for use by external devices. Table 4-3 below lists the characteristics of each power and ground line.

5.12 I/O SYSTEM CONNECTOR

Connection to the I/O system is made through an Amphenol 17-20500-1 connector. The mating I/O cable should have a 50-pin Amphenol 17-10500-1 connector.

Table 5-5 lists the pin assignments.

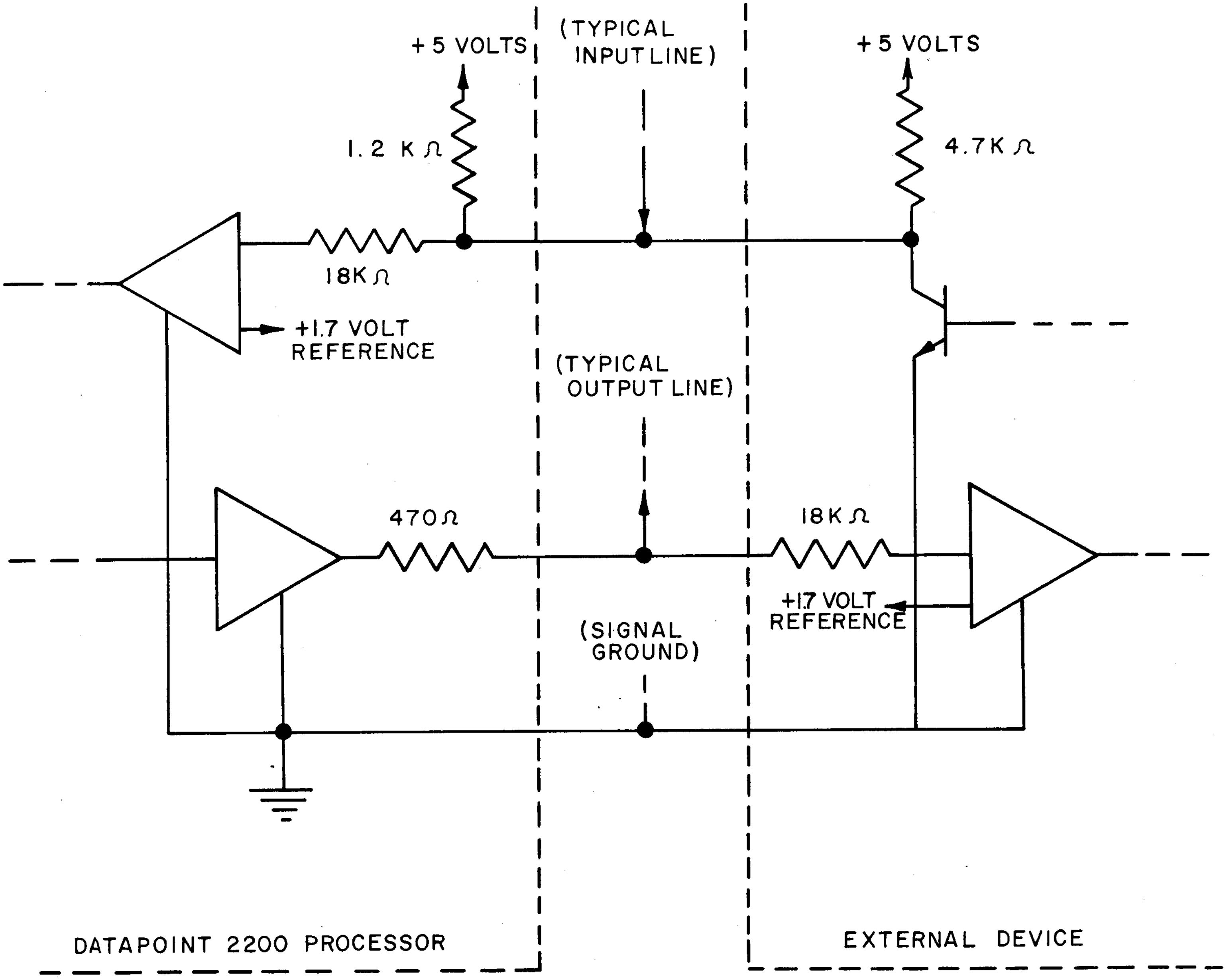


FIGURE 5-2

I/O CABLE, ELECTRICAL CHARACTERISTICS

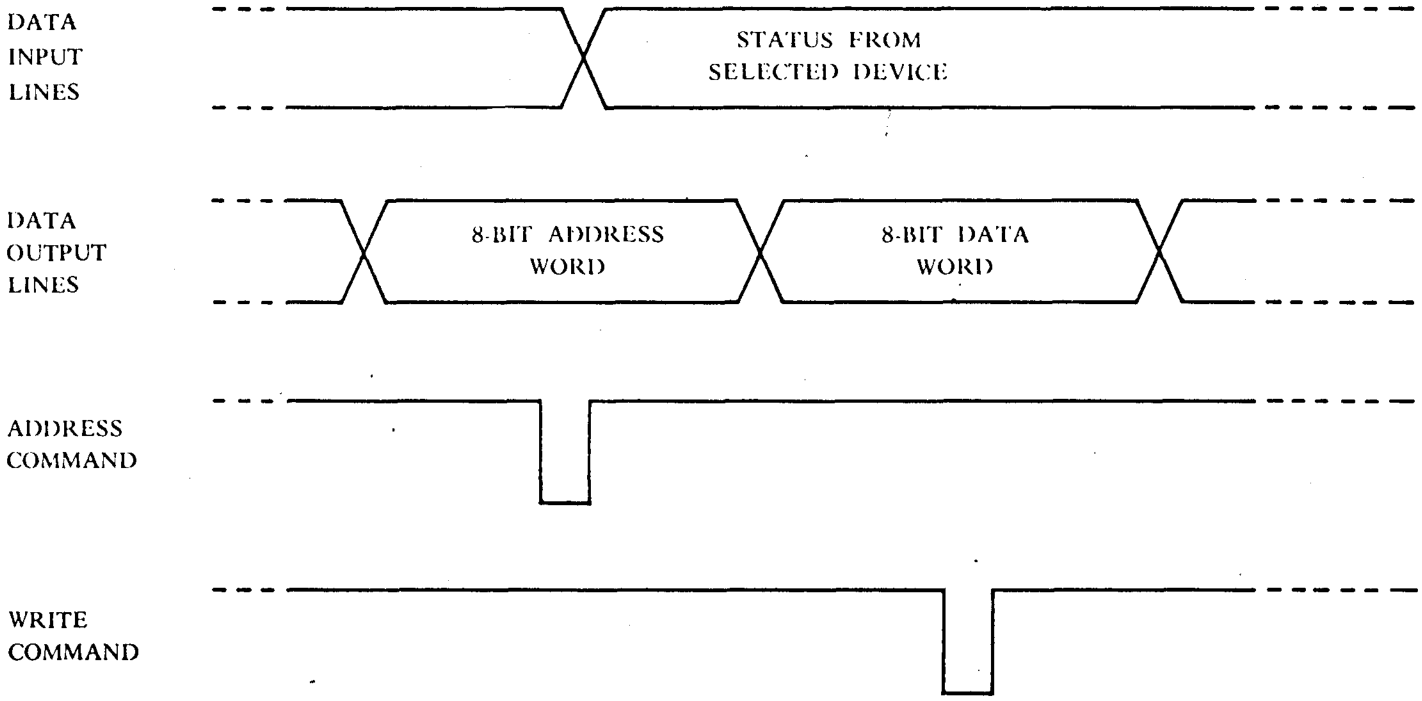


FIGURE 5-3

TYPICAL DATA OUTPUT
SEQUENCE

TABLE 5-3

I/O POWER AND GROUND LINES

VOLTAGE	MAX. CURRENT	REGULATION
-12 Volts	0.5 amps	±10%
- 5 Volts	0.1 amps	±5%
+ 5 Volts	3.4 amps	±5%
+12 Volts	0.5 amps	±10%
+24 Volts	0.1 amps	±5%
Power Ground	–	–
Signal Ground	–	–

TABLE 5-4

I/O CONNECTOR PIN ASSIGNMENTS

ASSIGNMENT		PIN NUMBER
Data output	0	44
	1	45
(A Bus Outputs)	2	46
	3	29
	4	30
	5	31
	6	32
	7	33
Data Input	0	1
	1	2
(A Bus Inputs)	2	3
	3	4
	4	5
	5	6
	6	7
	7	18
Input Strobe (Read)		12
Address Command		15
Sense Status Command		13
Sense Data Command		14
Write Command		19
Command 1		20
Command 2		21
Command 3		22
Command 4		23
153.6 KHz Clock		39
-12v		24
-5v		27
+5v		8, 9, 10, 11
+12v		25
+24v		26
Ground (Power & Signal)		40, 41, 42, 43

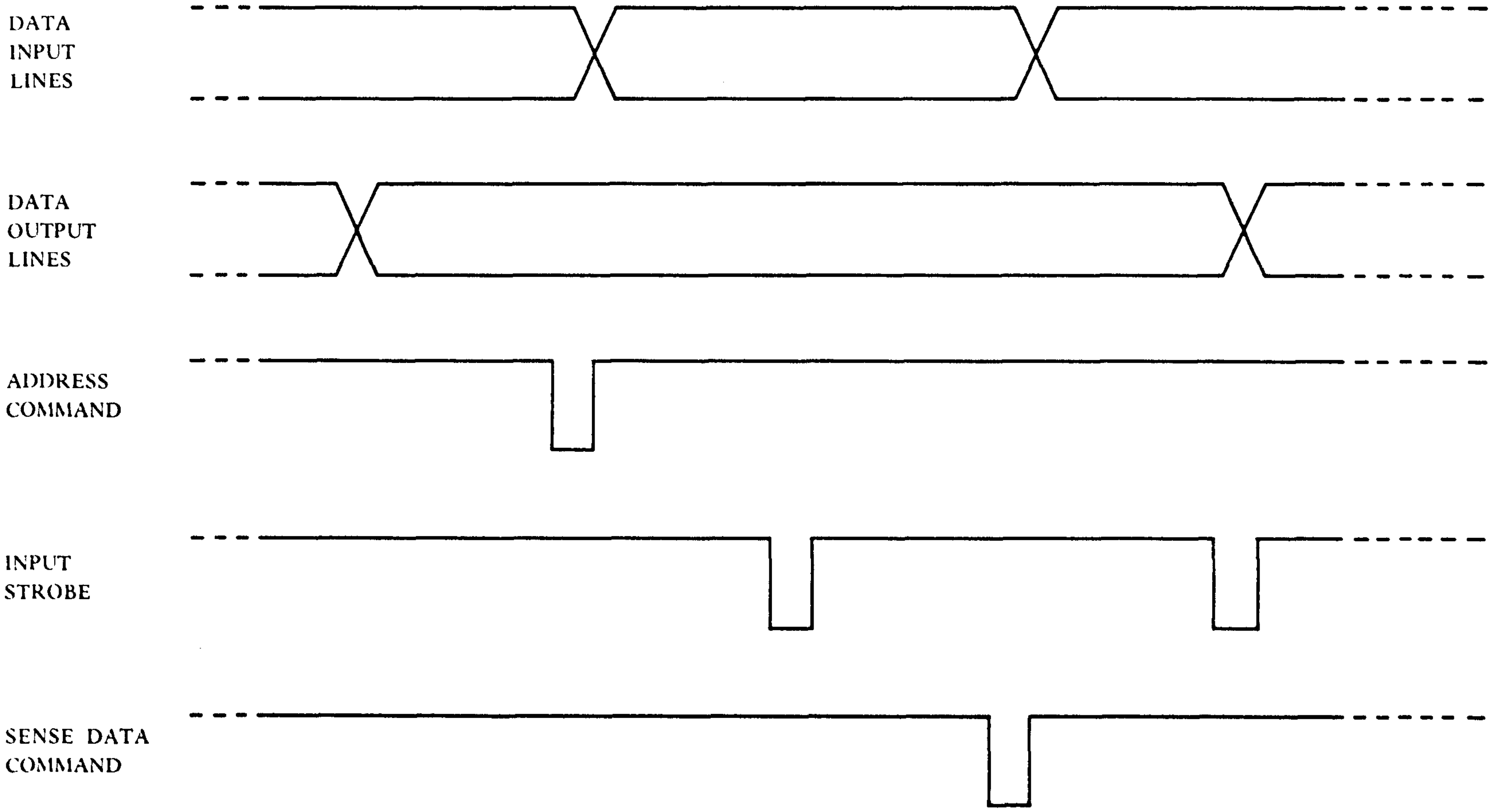


FIGURE 5-4

TYPICAL DATA INPUT SEQUENCE

PART 6

KEYBOARD

6.1 GENERAL DESCRIPTION

The keyboard on the Datapoint 2200 performs the functions of data entry and processor control. The keys are divided into three sections, each of which has it own function.

Section 1 consists of 41 standard alphabetic, numeric and special character keys found in the ASCII character set. Figure 5-1 illustrates the keyboard layout.

Section 2 consists of an 11 key matrix which is identical to a standard adding machine keyboard with the addition of a decimal point (period). The keys in this section are duplicates of certain keys found in Section 1 and are provided to facilitate entry of large amounts of numeric data. The 11-key matrix may be optionally supplied with control-key coding rather than numeric key coding and with keytops engraved to customer specifications.

The keys in Section 3 are special function keys which exert control over the processor. Their names and associated functions are as follows:

RUN Momentary contact switch, which when depressed, causes the processor to begin execution of the instruction located at the address in memory currently addressed by the program counter.

STOP Momentary contact switch which, when depressed, causes instruction execution to halt at the completion of the current instruction. Care should be taken when using this switch, because any tape operation which may be in progress will be aborted. If the auto-restart tab on the rear tape cassette is removed the STOP key performs the same function as the RESTART key.

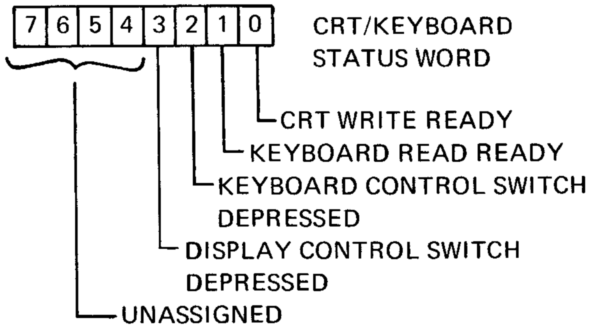
KEYBOARD Momentary contact switch which sets a status bit that may be tested at any time by the processor.

DISPLAY Momentary contact switch with a function similar to that of KEYBOARD switch. Either one or both of these switches may be depressed.

RESTART Momentary contact switch which causes the processor to halt, rewind the system or program tape mounted on Deck 1 (rear deck), load and execute the first record found on tape.

6.2 OPERATION

The keyboard is addressed by the processor by loading the A-register with 341_8 and executing an EX ADR command. (The crt display also uses this address. Data transfers to the processor are from the keyboard and transfers from the processor are to the display). Following the address sequence the c.r.t./keyboard status word can be loaded into the A-register by executing an INPUT instruction. Bit 1 of the A-register may be tested by the program to determine if a character is ready for transfer from the keyboard. Bits 2 and 3 will indicate if either the KEYBOARD or DISPLAY control switch is pressed.



The External Commands associated with the operation of the keyboard are as follows:

a. EX BEEP. This command produces a 1500 Hertz tone for a duration of about 100 msec. The tone could be used as an error oı ready signal to the keyboard operator.

b. EX CLICK. This command produces an audible click which could be used to acknowledge receipt of a valid character when a key is depressed.

c. EX COM1 (Command 1). Presents a control word contained in the A-register to the keyboard. Bit 5 of the control word controls the KEYBOARD switch light and bit 6 controls the DISPLAY switch light as follows:

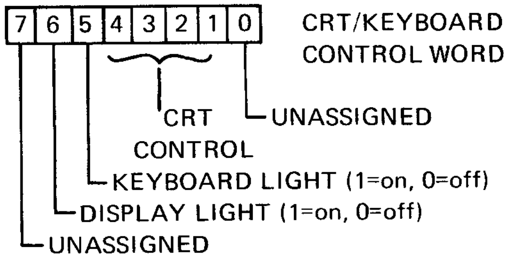

7 6 5 4 3 2 1 0
CRT/KEYBOARD CONTROL WORD
CRT CONTROL
UNASSIGNED
KEYBOARD LIGHT (1=on, 0=off)
DISPLAY LIGHT (1=on, 0=off)
UNASSIGNED

TABLE 6-1

KEYBOARD CODING (ASCII)

A-101	a -141	0-060	: -072
B-102	b -142	1-061	; -073
C-103	c -143	2-062	< -074
D-104	d -144	3-063	= -075
E-105	e -145	4-064	> -076
F-106	f -146	5-065	? -077
G-107	g -147	6-066	[-133
H-110	h -150	7-067	~ -176
I -111	i -151	8-070	] -135
J -112	j -152	9-071	Λ -136
		Space-040	— -137
K-113	k -153		
L-114	l -154	!-041	@ -100
M-115	m-155	"-042	{ -173
N-116	n -156	#-043	\ -134
O-117	o -157	$-044	' -140
P-120	p -160	%-045	| -174
Q-121	q -161	&-046	} -175
R-122	r -162	' -047	Enter-015
S-123	s -163	(-050	Cancel-030
T-124	t -164	) -051	Backspace-010
U-125	u -165	*-052	Del-177
V-126	v -166	+-053	
W-127	w-167	, -054	
X-130	x -170	- -055	
Y-131	y -171	. -056	
Z-132	z -172	/ -057	

SPECIAL NUMBER PAD OPTION

[.] - 016
[0] - 020
[1] - 021
[2] - 022
[3] - 023
[4] - 024
[5] - 025
[6] - 026
[7] - 027
[8] - 030
[9] - 031

PART 7

CRT DISPLAY

7.1 GENERAL DESCRIPTION

The display unit on the Datapoint 2200 consists of a CRT capable of displaying 12 lines of 80 characters each a character generator, 960 cells of refresh memory (power line frequency refresh rate), and a group of registers utilized to position the cursor. The maximum character transfer rate to the CRT depends on the software technique employed. It can vary from 60 characters per second (50 when used with 50 cycle power) to as much as 12,000 CPS (10,000 with 50 cycle power) using a special full screen transfer direct from memory.

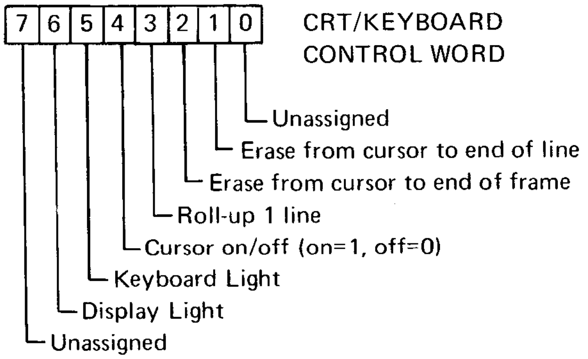
The character set utilized by the CRT display consists of the full ASCII set with both upper and lower case alphabetics and all numeric and special characters.

7.2 OPERATION

The CRT is addressed and status tested in the same manner as the keyboard (see paragraph 5.2). Bit 0 of the status word indicates that the CRT is ready to accept data or commands. Characters are transferred to the screen by loading the A-register with the character to be displayed and executing an EX WRITE. The character will be displayed at the current cursor location.

Control of the CRT is accomplished through the use of the three external commands - Command 1, Command 2, and Command 3. The functions performed by these commands are as follows:

a. EX COM1 (Command 1) Transfers a control word contained in the A-register to the CRT. The applicable bit assignments and their functions are as follows:


7 6 5 4 3 2 1 0
CRT/KEYBOARD CONTROL WORD
Unassigned
Erase from cursor to end of line
Erase from cursor to end of frame
Roll-up 1 line
Cursor on/off (on=1, off=0)
Keyboard Light
Display Light
Unassigned

The erase functions permit selective erasures on the screen by limiting erasures to those character positions following the current cursor position to the end of the line (or page).

The roll-up function causes all displayed characters (not the cursor) to move up one line. The top line on the screen is lost.

The cursor image may be turned on or off through the control word. The cursor position is the same in either case. The cursor image is automatically turned off whenever the processor is in the HALT state.

b. EX COM2 (Command 2) Positions the cursor to the horizontal character slot designated by the contents of the A-register. Character position $0\text{-}79_{10}(0\text{-}117_8)$ are valid.

c. EX COM3 (Command 3) Positions the cursor to the line designated by the contents of the A-register. Line number $0\text{-}11_{10}$ $(0\text{-}13_8)$ are valid.

In order to write a new character, the cursor must occupy that character's position on the screen. After the character has been written, the cursor should then be moved to the next horizontal (or vertical) position desired. The CRT Write Ready status bit must be true before positioning the cursor or displaying a character.

Both the CRT and keyboard utilize the standard ASCII character set. (See Table 5-1). Any invalid character code will appear as a blank space on the CRT screen.

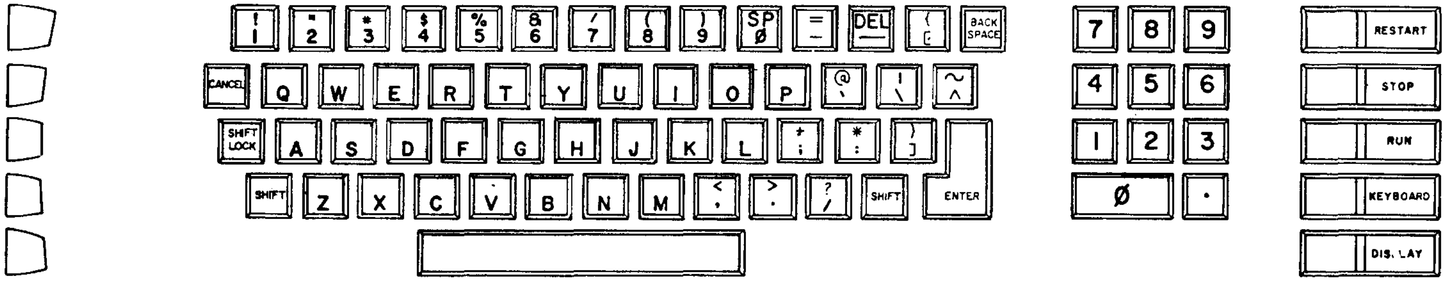


FIGURE 7-1
KEYBOARD LAYOUT

PART 8

CASSETTE TAPES

8.1 GENERAL DESCRIPTION

The Datapoint 2200 contains two cassette tape recording devices for storage of programs and data. Since the hardware RESTART (section 5.1) uses the rear deck (number one), programs will typically be on it while data areas will be the front deck (number two). However, once the machine is initially loaded, either deck may be used for both purposes.

Data on the Tape is organized by record (of any length). Records are written and read at 700 eight-bit characters per second with a tape speed of approximately 15 inches per second. See Table 7.1 for a list of the physical specifications.

8.2 OPERATIONS

Data is recorded or read in bit serial fashion on one track. Each eight bit character is framed by three sync bits on either side of the character:

-	0 1 0	x x x x x x x x	0 1 0	x x x x x x x x	0 1 0	-
	Sync Code	Character 1	Sync Code	Character 2	Sync Code	

The appearance of the correct sync code indicates that the character is valid. Any other sync code causes special action to be taken on data reads. Note that the sync codes are valid for tape motion in either direction so the tape may be read backwards although in the reverse direction the data bits will appear reversed (bit 0 will be bit 7, 1 will be 6, etc.)

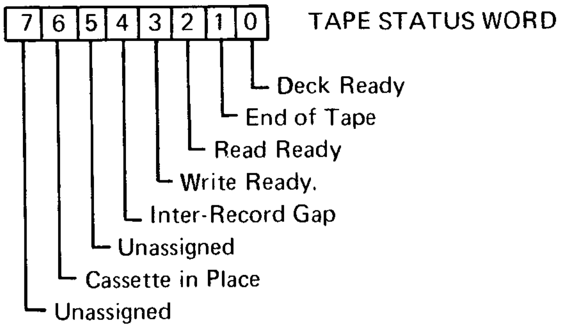
A record is a group of successive valid characters. An inter-record gap is indicated by the failure of the sync code to be zero one zero and the character position on all mark code (ones):

-	0 1 0	x x x x x x x x	0 1 0	x x x x x x x x	1 1 1	x x x	-
	Sync Code	Valid Character	Sync Code	Invalid Character	Sync Code	Inter Record Gap	

Only valid characters will be presented as data from the tape unit.

8.3 STATUS

The cassette tape unit is addressed by the processor by loading the A-register with 360_8 and executing the EX ADR instruction. Following this sequence, the tape unit status can be loaded into the A-register by executing an INPUT instruction. The bit assignments are as follows:



DECK READY — Deck ready will be set whenever the tape unit is ready to accept another command. (Only the TSTOP command should be issued if this bit is false). When Deck Ready is true the tape will be stopped, a cassette in the selected deck, and not wound to the clear leader at either end, and the head engaged. This bit should be checked after selecting a deck.

END OF TAPE — End of Tape indicates that the cassette has run onto leader (in either direction).

READ READY — Read Ready indicates that the selected deck has read another character.

WRITE READY — Write Ready indicates that the selected deck is ready to write another character.

INTER-RECORD GAP — Inter-Record Gap indicates the selected deck has come across an inter-record gap (invalid sync code).

CASSETTE IN PLACE — Cassette in Place indicates that a cassette is physically in place in the selected deck.

8.4 CONTROL

When the cassette tape unit is addressed the following instructions will control the action of the tape:

a. EX TSTOP causes any motion of either deck to be stopped and any read or write operations to be terminated. When everything has settled, the ready status bit will come true and operations may be resumed.

b. EX DECK1 causes deck one (rear) to be the currently selected deck. Before commanding a deck selection, care should be taken that the currently selected deck has completed all operations.

c. EX DECK2 causes deck two (front) to be the currently selected deck. Note the precaution in (b).

d. EX RBK causes the currently selected deck to be set in forward motion and, after 70 msec, for the read circuitry to be enabled. The read ready status bit will come true upon appearance on the tape of the first valid character. Upon appearance of an invalid sync code, the inter-record gap status bit comes true and tape motion is automatically stopped. Note that this will happen only after at least one valid character has been found. Once the read ready status bit comes true, the character must be taken within 2.8 milliseconds or it will be overwritten with the next one. The tape read hardware double-buffers incoming characters to allow the 2.8 msec character availabilitv.

e. EX BSP is similar tc EX RBK except that tape motion is in the reverse direction so the data bits will be reversed.

f. EX SF is similar to EX RBK except the tape is not stopped upon appearance of an inter-record gap, and if allowed to continue will start to read the next record on the tape. In this case, the read ready status bit will come true again after the first character of the next record is read. Only an EX TSTOP will stop the motion initiated by EX SF.

g. EX SB is similar to EX SF except that tape motion is in the reverse direction and the data bits are reversed.

h. EX WBK causes the currently selected deck to be set in forward motion and all status bits except the write ready to go false. A character must then be presented within 2.8 milliseconds (the first character will be accepted at once due to the buffering in the tape hardware and then there will be a pause while the tape comes up to speed), at which time the write ready will go false until the writing circuitry is ready to accept another character. An end of record is signaled to the hardware by withholding a character for a period of time longer than 2.8 milliseconds specified above. When this is done, the write ready will go false, an inter-record gap will be written, the tape motion will cease and the deck ready status bit will come true again.

i. EX REWIND causes the tape to be rewound to the beginning on the selected deck. Worst case rewind time is approximately 40 seconds.

j. PUNCH TABS on the Cassette Cartridge are used for "write protect" and "automatic restart". The punch tab on the left (as you face the terminal) inhibits the ability to write on tape, when punched. When the tab on the right is punched, it causes an automatic restart whenever a halt or power-up occurs.

TABLE 8-1

TAPE UNIT PHYSICAL SPECIFICATIONS

Density	47 characters/inch
Speed	7.5 ips
Recording Rate	350 c.p.s.
Capacity	130,000 characters (typical)
Start/Stop Time (Inter-Record Gap)	280 msec.
Start/Stop Distance (Inter-Record Gap)	2 inches
Rewind Speed	90 ips
Rewind Time (max 300 ft.)	40 sec.
Character Transfer Time	2.8 msec.

Datapoint

Datapoint Corporation
9725 Datapoint Drive
San Antonio, Texas 78284
512/696-4520