Burroughs

220 OPERATIONAL CHARACTERISTICS

BULLETIN 5020A
REVISED AUGUST, 1960

OPERATIONAL CHARACTERISTICS

OF THE

BURROUGHS 220

ELECTRONIC DATA PROCESSING SYSTEM

TABLE OF CONTENTS

Chapter	Title	Page
1	INTRODUCTION	1-1
2	THE DATA PROCESSOR	2-1
	General	2-1
	Representation of Information	2-1
	Word Format	2-1
	Numeric Information	2-1
	Alphabetic or Alphanumeric Information	2-2
	Instruction Words	2-2
	Partial-Word Fields	2-3
	Registers	2-3
	The A Register	2-3
	The R Register	2-3
	The D Register	2-3
	The B Register	2-3
	The C Register	2-3
	The P Register	2-4
	The IB Register	2-4
	The E Register	2-4
	The CD Register	2-4
	The T Register	2-4
	Information Flow	2-4
	The Operation Cycle	2-4
	Input Flow	2-5
	Output Flow	2-6
	Exceptional Conditions	2-6
	The Execute Phase	2-6
	Halt	2-6
	No Operation	2-7
	Clear Add	2-7
	Clear Add Absolute	2-7
	Clear Subtract	2-8
	Clear Subtract Absolute	2-8
	Add	2-9
	Add Absolute	2-9
	Subtract	2-10
	Subtract Absolute	2-10
	Multiply	2-11
	Divide	2-12
	Round	2-13
	Extract	2-14
	Compare Field A	2-14
	Compare Field R	2-14
	Add To Location	2-16
	Increase B, Branch	2-17
	Decrease B, Branch	2-18
	Floating Add	2-18
	Floating Add Absolute	2-18
	Floating Subtract	2-20
	Floating Subtract Absolute	2-20
	Floating Multiply	2-22
	Floating Divide	2-24
	Increase Field Location	2-26

TABLE OF CONTENTS (Cont)

Chapter	Title	Page
	Decrease Field Location	2-27
	Decrease Field Location, Load B	2-29
	Record Transfer	2-30
	Branch Unconditionally	2-31
	Branch, Overflow	2-31
	Branch, Repeat	2-32
	Branch, Sign A	2-32
	Branch, Comparison High	2-33
	Branch, Comparison Low	2-33
	Branch, Comparison Equal	2-34
	Branch, Comparison Unequal	2-34
	Branch, Field A	2-35
	Branch, Field R	2-36
	Set Overflow Remember	2-37
	Set Overflow Halt	2-37
	Interrogate Overflow Mode	2-37
	Store A	2-38
	Store R	2-38
	Store B	2-38
	Load R	2-39
	Load B	2-40
	Load B Complement	2-40
	Load Sign A	2-40
	Store P	2-41
	Clear A	2-41
	Clear R	2-41
	Clear A, R	2-41
	Clear B	2-41
	Clear A, B	2-41
	Clear R, B	2-41
	Clear A, R, B	2-41
	Clear Location	2-42
	Shift Right A	2-43
	Shift Right A And R	2-43
	Shift Right A With Sign	2-43
	Shift Left A	2-44
	Shift Left A And R	2-44
	Shift Left A With Sign	2-44
3	THE CONTROL CONSOLE	3-1
	General	3-1
	Program Control Switches	3-1
	The Reset and Transfer Switch	3-1
	The Keyboard	3-1
	The Supervisory Printer	3-1
	The Interval Timer	3-2
	The Execute Phase	3-2
	Keyboard Add	3-2
	Supervisory Print-Out	3-2
	Branch, Control Switch	3-3
4	THE MAGNETIC-TAPE SYSTEM	4-1
	General	4-1

TABLE OF CONTENTS (Cont)

Chapter	Title	Page
	The Magnetic-Tape Storage Unit	4-1
	The DATAFILE Unit	4-1
	Tape Format	4-2
	The Editing Process	4-3
	Magnetic-Tape Storage Units	4-3
	DATAFILE Units	4-4
	Writing on Newly Edited Tape	4-6
	Search	4-7
	Scan	4-8
	Reading	4-8
	Overwriting	4-9
	Positioning	4-9
	Interrogation	4-9
	Input Sign Control	4-9
	Parity Checking	4-9
	Digit-Count/Word-Count Checking	4-11
	Information Flow	4-11
	Input Flow	4-11
	Output Flow	4-11
	Use of BURROUGHS 205 Magnetic Tape	4-12
	Exceptional Conditions	4-12
	The Execute Phase	4-12
	Magnetic-Tape Search	4-12
	Magnetic-Tape Field Search	4-12
	Magnetic-Tape Lane Select	4-12
	Magnetic-Tape Rewind	4-12
	Magnetic-Tape Rewind, De-Activate	4-12
	Magnetic-Tape Scan	4-14
	Magnetic-Tape Field Scan	4-14
	Magnetic-Tape Read	4-15
	Magnetic-Tape Read, Record	4-16
	Magnetic-Tape Initial Write	4-18
	Magnetic-Tape Initial Write, Record	4-19
	Magnetic-Tape Overwrite	4-20
	Magnetic-Tape Overwrite, Record	4-21
	Magnetic-Tape Position Forward	4-21
	Magnetic-Tape Position Backward	4-21
	Magnetic-Tape Position At End Of Information	4-21
	Magnetic-Tape Interrogate, Branch	4-22
	Magnetic-Tape Interrogate End-Of-Tape, Branch	4-22
5	THE PAPER-TAPE SYSTEM	5-1
	General	5-1
	The Photoreader	5-1
	Word Format	5-1
	Instructions	5-2
	The Paper-Tape Punch	5-2
	The Printer	5-2
	The Mechanical Reader	5-2
	Information Flow	5-2
	Input Flow	5-2
	Output Flow	5-3
	Exceptional Conditions	5-3

TABLE OF CONTENTS (Cont)

Chapter	Title	Page
	The Execute Phase	5-3
	Paper-Tape Read	5-4
	Paper-Tape Read, Branch	5-6
	Paper-Tape Read, Inverse Format	5-8
	Paper-Tape Write	5-9
6	THE CARDATRON SYSTEM	6-1
	General	6-1
	Input Unit Operation	6-1
	Machine Characteristics	6-1
	The Information Band	6-1
	Format Bands	6-1
	Row Counter	6-2
	Core Shift Register	6-2
	The Numeric Toggle	6-2
	The Two Phases of Input Operation	6-2
	Transfer of Data to Information Band	6-3
	Transfer of Data to Core Storage	6-4
	Input Format Bands	6-5
	Format Selection	6-5
	Program-Controlled Format Bands	6-5
	Fixed Editing Function	6-5
	The Use of Input Format-Band Digits	6-7
	Insert Zero Digit (0)	6-7
	Alphanumeric-Transfer Digit (1)	6-7
	Insert 2 Digit (2)	6-7
	Delete Digit (3)	6-7
	Transferring the Sign-Position Digit	6-8
	Functions of Input Format-Band Digit Pairs	6-8
	Numeric Transfer: Digit Pair 31	6-8
	Alphabetic Transfer: Digit Pair 11	6-8
	Column Delete: Digit Pair 33	6-8
	Construction of an Input Format Band	6-9
	Using the Format-Band Coding Form	6-9
	Ordering of Editing and Transfer	6-9
	Active Segment	6-9
	Inactive Segment	6-9
	Writing an Input Format Band: Example	6-9
	The Editing Control Stream	6-10
	The Inactive Segment	6-14
	Output Unit Operation	6-14
	Information Transfer	6-15
	Designation of Format Bands	6-17
	Format Band Selection	6-17
	Instructions	6-17
	Functions of Output Format-Band Digits	6-17
	Insert Blank Digit (0)	6-17
	Transfer-Alphanumeric Digit (1)	6-18
	Transfer-Numeric Digit (2)	6-18
	Delete Digit (3)	6-18
	The Execute Phase	6-18
	Card Read	6-18

TABLE OF CONTENTS (Cont)

Chapter	Title	Page
	Card Write	6-25
	Card Read, Format Load	6-27
	Card Write, Format Load	6-29
	Card Read Interrogate, Branch	6-31
	Card Write Interrogate, Branch	6-31

APPENDICES

Appendices	Title	Page
A.	BURROUGHS 220 Instruction List in Operation-Code Order	A-1
B.	BURROUGHS 220 Instruction List, Alphabetically, by Abbreviation	A-3
C.	Alphanumeric Codes and Their Representation	A-6
D.	A Glossary of Terms	A-9
E.	A Glossary of Symbols	A-9
F.	Instruction Times, Alphabetically, by Abbreviation	A-10
G.	Magnetic-Tape Instruction Times	A-13

LIST OF TABLES

Table	Title	Page
4-1	Percentage of Useful Data in a Block	4-2
4-2	Number of Blocks and Words of Useful Data Recorded in 100 Feet of Flaw-Free Tape in One Lane	4-4
4-3	Defining the End-of-Tape Block	4-5
4-4	Defining the Control Block	4-6
4-5	Definition of Class I Parity Errors	4-10
6-1	Row Counter Settings, CARDATRON 220 Input Unit	6-3
6-2	Row Counter Settings, CARDATRON 220 Output Unit	6-16
6-3	CARDATRON Input Sign Control	6-21
G-1	220 Magnetic-Tape—TSU—Instruction Times in ms	A-14
G-2	220 Magnetic-Tape—DATAFILE—Instruction Times in ms. (Nominal)	A-15

LIST OF FIGURES

Figure	Title	Page
2-1	Digit-Position Identification in the Data Processor Word	2-1
2-2	The Fetch Phase (Flow Chart)	2-4
2-3	The Fetch Phase (Diagram)	2-5
2-4	Input Flow	2-5
2-5	Output Flow	2-6
4-1	Block Format (Not to Scale)	4-3
4-2	Example of Allocation of Storage on Execution of MRR	4-8
4-3	Input Flow	4-11
4-4	Output Flow	4-12
5-1	Input Flow	5-3
5-2	Output Flow	5-3
6-1	Input Information Path	6-4
6-2	Format-Band Coding Form (Input)	6-11
6-3	Output Information Path	6-15
6-4	Format-Band Coding Form (Output)	6-19

1

Introduction

This volume in the list of Burroughs Electronic Data Processing publications is intended as a manual for programmers. Its purpose is, as its title indicates, to acquaint the programmer with the operational characteristics of the BURROUGHS 220 Data Processing System.

The center of the BURROUGHS 220 System is the Data Processor which, with its magnetic-core storage, gives high computation speeds. Information is handled by the 220 in binary-coded decimal form. Thus the programmer need not translate the information to the binary number system and retranslate it to decimal form. Keeping track of the decimal point may be accomplished either through scaling or through use of the built-in automatic floating-point feature. Further reduction in programming and operating time can be achieved through the use of the 220's partial-word instructions.

The Control Console, the operation center for the BURROUGHS 220 System, is designed for ease of operation and maximum usefulness in checking out programs. It includes a keyboard for manual entry of data or instructions and an interval timer to check operating time. Register contents are displayed at all times; controls allow single-step manual operation. Thus, when checking out a program, the operator can examine intermediate results at any time. Program errors are pinpointed by indicators showing in what part of the system an unexpected stop occurred. The operator can inspect the contents of a storage location and enter new information into storage without affecting the contents of the registers. By depressing a button, successive storage locations can be displayed. Ten program switches, used singly or in combination, extend the programming flexibility of the 220 and reduce over-all programming time.

Two kinds of magnetic-tape storage are used with the BURROUGHS 220: the Magnetic-Tape Storage Unit and the DATAFILE* Unit. Both use dual-lane, addressable magnetic tape: i.e., the tape is divided lengthwise into two lanes to reduce the need for tape movement and the information written on the tape can be updated selectively. Magnetic tape may be "searched" for information in either a forward or a backward direction. Searching is the process of automatically comparing the search key, designated by the instruction, with the first word of each block on a lane of tape until a comparison is found. Once initiated, searching of magnetic tape is independent of Data Processor operation; computation can continue while information is being found. Tape can also be "scanned" independently. Scanning proceeds in a forward direction only and is the process of inspecting a designated part of the first ten words of each record to locate a particular category. Blocks may be from 10 to 100 words in length. From 1 to 10 blocks may be read with a single instruction.

The Paper-Tape System provides the BURROUGHS 220 System with complete facilities for input and output utilizing paper tape. The Photoreader reads punched paper tape at the rate of either 500 or 1000 characters per second and can stop on a single character with the next character in position to be read. Information from storage can be punched at the rate of 60 characters per second by the paper-tape punch. The punched tape produced is accepted for reading by the Photoreader or by the mechanical reader available with the Supervisory Printer. End-of-word characters are automatically supplied. The Supervisory Printer operates at the rate of 10 characters per second, printing numeric, alphabetic, and special characters. In addition to the Supervisory Printer, the 220 can use up to 10 Photoreaders and up to 10 paper-tape punches. Additional character-at-a-time printers. with the same characteristics as the Supervisory Printer. can be substituted for the punches.

A complete buffering and editing system connects card readers, card punches, and line printers to the BURROUGHS 220 Data Processor. This electronic card-handling system, the CARDATRON* System, allows the use of up to seven punched-card machines and/or printers—all operating simultaneously, independent of the Data Processor. A CARDATRON system consists of a control unit and up to seven input units and output units. Each card reader, punch, or line printer is connected to one of these units; each unit contains a magnetic-drum buffer. The card machine communicates only with its associated buffer unit and the Data Processor receives information from each buffer in a fraction of the time required for reading a punched card. Thus the Data Processor is not tied to the speeds of the card machines; instead, computation can continue during reading, punching, and printing. Editing, format control, and alphabetic translation of both input and output is handled automatically by the CARDATRON system and editing and format facilities may be program-controlled from the Data Processor. Thus there is no limit to the number of

* A Trade-mark of the Burroughs Corporation.

format choices available. Five separate formats are stored at the same time on the buffer drum of each input and output unit. Any one may be replaced by execution of one instruction. The CARDATRON system accepts and produces any combination of numeric, alphabetic, and special characters. And all 120 print positions are available for printing without restrictions of format or amount of alphabetic information.

The operation cycle of the BURROUGHS 220 is divided into two phases, a Fetch Phase and an Execute Phase. The Fetch Phase is described, with flow charts, in the beginning of Chapter 2, the chapter describing the Data Processor. Its description is not repeated since the Fetch Phase is the same for every operation. By far the greater part of most of the chapters of this volume is taken up with a discussion of the Execute Phase of each operation related to the system component being described.

Appendix D and Appendix E are A Glossary of Terms and A Glossary of Symbols, respectively. Since these are the terms and symbols used extensively throughout this manual, it would be advisable to study them before delving into the manual itself.

Extreme care has been taken to make this manual as error-free as possible; perfection, in such an endeavor, is seldom attained. The authors and publishers would appreciate, therefore, constructive criticism of the contents of this manual, whether in regard to errors in fact or to manner of presentation. Such comments should be addressed to:

Manager, Product Support, 220
Burroughs Corporation
6071 Second Avenue
Detroit 2, Michigan

2

The Data Processor

GENERAL

The BURROUGHS 220 Data Processor is a general-purpose, stored-program, sequentially-controlled, series-parallel, automatic, electronic, digital computer which employs a single-address operation code. The basic instruction list of the BURROUGHS 220 Data Processor consists of 41 instructions, but variations in the structure bring the number of distinct operations to 65 (this does not include instructions referring to input-output equipment, which will be enumerated in the chapters in which the various system components are described). Control and arithmetic units as well as power supplies and working storage are housed separately.

The Control Console is regarded as being an integral part of the Data Processor; however its details will be discussed in a separate chapter.

Depending on the size of magnetic-core working-storage in the system, one or two units are required to house that storage.

REPRESENTATION OF INFORMATION

The basic unit of information in the BURROUGHS 220, is the 8, 4, 2, 1-binary-coded decimal digit. An aggregate of eleven digit positions is called a word. More accurately, the word should be described as ten digits plus sign, since it is not intended that the sign-digit position in a word shall be used in the same unrestricted fashion as the other ten digits. The digit positions in a word are identified as shown in Figure 2-1.

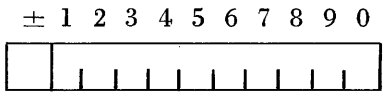


Figure 2-1. Digit-position identification in the Data Processor word

The logical structure of the BURROUGHS 220 is based on the fixed-length ten-digit-plus-sign word described above. Certain operations, however, permit manipulation of partial-word fields.[1]

Depending on the manner in which the word is referred to, its contents may be regarded as numeric, alphabetic or mixed numeric and alphabetic (otherwise, alphanumeric), or an instruction. For example, 0 5941 51 4148 may represent the number + 5941514148, the noun RAJAH, or an instruction which causes information on magnetic tape to be scanned.

Each word in magnetic-core storage is identified by a unique four-decimal-digit number called the address of the word. The word itself is said to be stored in the location whose address identifies it.

Although the BURROUGHS 220 uses only the ten decimal digits, each decade of four bits in the 8, 4, 2, 1 code might have been used to count to 15. The occurrence—for any reason—in the low-order digit position of certain of the control registers of a configuration corresponding to any one of the decimal numbers from ten to 15, is detected automatically. The detection of such a configuration results in a Digit Check ALARM STOP:[2] the Data Processor stops and an ALARM indicator light—on the Control Console—labelled DIGIT CHECK comes on.

WORD FORMAT

NUMERIC INFORMATION

Fixed-Point Numbers. Each word of ten digit positions plus sign-digit position may represent a number in the range −9999 99 9999 to +9999 99 9999. It is conventional to regard the so-called machine decimal point as being located between the sign and the high-order numeric digit; hence, it is customary to say that fixed-point machine numbers are restricted to the range $-1 < n < +1$.

If the result of an arithmetic operation yields a number outside the range specified above, arithmetic overflow is said to have occurred. The occurrence of arithmetic overflow causes the OVERFLOW indicator to be turned on. The status of the OVERFLOW indicator may be interrogated or ignored at the option of the programmer. (See overflow set instructions, page 2-37.)

Whenever it is necessary to distinguish the machine decimal point from a problem decimal point, the symbol ∧

[1] A partial-word field is any set of contiguous digit-positions within a Data Processor word; for example, digit positions 5, 6, and 7; or ±, 1, 2, 3, and 4; or ±.

[2] Optionally, an audible alarm signal may be caused to sound whenever the Data Processor stops on detecting an exceptional condition. All exceptional conditions will be described in context.

will be used to specify the location of the problem decimal point. A dot—the conventional symbol—will represent the machine decimal point.[3]

The convention for representing signs is the following: "0" in the sign-digit position means the number is positive (or, at least, non-negative, i.e., greater than or equal to zero); "1" in the sign-digit position means the number is negative (or, at least, non-positive, i.e., less than or equal to zero). An immediate consequence of this convention is that zero may be signed either positive or negative.[4]

Normally, the sign-digit position of a word which is numeric, and on which arithmetic operations are performed, is not different from 0 or 1. If arithmetic operands do have sign digits which differ from 0 or 1, the three high-order bits (8, 4, and 2) in the sign digit will be set to zero before arithmetic begins; in the result word, the three high-order bits will be zero.

Floating-Point Numbers. Each word of ten digit positions plus sign-digit position may represent a number in exponential or scientific notation; we call such a number a floating-point number. Digit-positions 1 and 2 hold the coded exponent—including its sign. A floating-point number with exponent greater than or equal to -50 and less than or equal to 0 is coded in the range 00 to 50, that is, the actual exponent is increased by 50; a floating-point number with exponent greater than 0 but less than 50 is coded in the range 51 to 99: again, the actual exponent is increased by 50.

Digit positions 3 through 0 hold the so-called mantissa of the floating-point number; the sign of the mantissa is the sign of the Data Processor word. The convention for representing signs of mantissas—and, hence, of floating-point numbers—is the same as the convention for fixed-point numbers. The decimal point of a floating-point number is regarded as being located between digit positions 2 and 3 of the floating-point word, as a result of which it is customary to assert that floating-point machine numbers, n, are restricted to the range $10^{-51} < n < 10^{+49}$.

Usually, floating-point numbers appear in normalized form; that is, the exponent and mantissa of a number are adjusted so that the highest-order digit of the mantissa is different from zero. Of course, in adjusting the mantissa, it may happen that the exponent will fall outside the permissible range, 00 through 99:

1. If the result of an arithmetic operation causes a floating-point exponent to exceed 99 (exponent) overflow is said to have occurred, and the OVERFLOW Indicator is set "on."
2. If the result of an arithmetic operation would have caused a floating-point exponent to be smaller than 00 (exponent) underflow is said to have occurred, and the arithmetic register (s) which was (were) to contain the result is (are) cleared.

The result of every floating-point arithmetic operation is normalized.

The following examples illustrate floating-point representation:

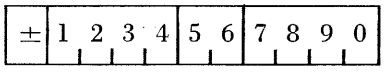
Number	Floating-point Representation
$+737_{\Delta}0\ 69\ 0000$	$+5373\ 70\ 6900$
$+_{\Delta}0004\ 88\ 7900$	$+4748\ 87\ 9000$
$-2344\ 9_{\Delta}1\ 8672$	$-5523\ 44\ 9186$
$-_{\Delta}8000\ 00\ 0236$	$-5080\ 00\ 0002$

ALPHABETIC OR ALPHANUMERIC INFORMATION

An alphabetic character is represented by two numeric digits in the BURROUGHS 220 code. The complete BURROUGHS 220 code is shown in Appendix C. Each word, therefore, has a capacity for five alphabetic characters; the sign-digit position of the word is used for a "flag" which indicates that the contents of the word are coded alphanumerically. The flag used is the digit 2.

INSTRUCTION WORDS

A BURROUGHS 220 instruction word may be regarded as composed of four parts, called the address (digit-positions 7, 8, 9, and 0), operation code (digit-positions 5 and 6), control (digit-positions 1, 2, 3, and 4), and sign-digit (±) parts which are illustrated below.



In the discussion of each BURROUGHS 220 instruction word which appears in this section, an instruction format is given followed by definitions of the symbols used. The following symbols, however, are common to many of the instructions. *To avoid unnecessary repetition, the definitions given here will apply to all instruction words in which these symbols are found:*

± : If ± is odd, B-register address-modification will occur; otherwise, there will be no such modification.

i : this digit not relevant to the execution of this instruction.

Op: operation code.

[3] This decimal point representation conforms to common usage and, therefore, its adoption here supersedes the representation set forth in any other Burroughs Electronic Data Processing publication.

[4] When an arithmetic operation yields a zero result the sign of the result can be predicted. See the Remarks section in the description of the appropriate operation and the corresponding flow chart.

Address Part. The four low-order digit positions in the instruction word are called the address part of the instruction because, in many instructions, they name the address of a location whose contents are used during the execution of the instruction; otherwise, the address part of the word is used to specify some quantity which is not an address; or the address part may not be relevant to the execution of the instruction.

In general, any part of an instruction word which is irrelevant to its execution may be coded in an arbitrary fashion.

Operation-Code Part. Digit-positions 5 and 6 in an instruction word are used to specify the numeric operation code of the instruction to be executed.

Control Part. Digit-positions 1 through 4 are used for a variety of purposes, among which may be listed the specification of partial-word boundaries, the designation of input and/or output units, the enumeration of tallies, etc., each of which specifies some control over the manner in which the instruction will be executed.

Sign-Digit Part. The sign-digit position of an instruction is used to specify whether the address part of the instruction is to be modified—in a manner which will be specified below—by the contents of the B register as the instruction is brought from storage to the control register: if the sign digit is an odd integer, B-register address-modification will occur; if the sign digit is an even integer, it will not.

On input from paper tape or punched cards the sign-digit position of an instruction may contain a flag which directs the instruction to the control register; simultaneously, the input device receives a signal that the execution of the input instruction is completed.

Additionally, the four-bit of the sign digit is used to specify that a partial-word key is desired and, hence, that MAGNETIC-TAPE FIELD SEARCH or MAGNETIC-TAPE FIELD SCAN is to be executed.

PARTIAL-WORD FIELDS

Several instructions allow operations to be performed on partial-word fields. It is necessary to identify the low-order digit position (i.e., the beginning) and the number of digits (i.e., the length) of the partial-word field. Digit-positions 1 and 2 of the instruction word are used for this purpose: digit-position 1 specifies the location of the low-order digit position;digit-position 2 specifies the number of digits in the partial-word field. Thus, for example, if these two digit positions contain 04, the four low-order digit positions of the referenced word are singled out; if these two digit positions contain 62, digit positions 5 and 6 are singled out of the referenced word.

The letter "s" is used to symbolize the low-order digit position, or starting point, the letter "L" the length, of a partial-word field. In order to describe completely the location of a partial-word field, we also require the address of the word in which the partial-word field is located. The complete address of a partial-word field is symbolized thus: aaaa:sL. In accordance with the usual convention (aaaa:sL) specifies the contents of the partial-word field. Thus, for example, (1000:04) is the address part of the word in location 1000; and (1426:00) specifies all but the sign digit of the word in location 1426.

REGISTERS

In the control and arithmetic sections of the Data Processor are several registers of interest to the programmer. The specific role of each of these registers is described in detail as each operation is described.

THE A REGISTER

The A register is a ten-digit-plus-sign-digit-position register. Its primary function is to store one of the operands as well as the result of an arithmetic operation, although it serves other purposes as well. It is frequently called the accumulator. The A register will be designated as rA, for short.

THE R REGISTER

The R register is a ten-digit-plus-sign-digit-position register. It is primarily an extension of the A register. It will be designated as rR.

THE D REGISTER

The D register is a ten-digit-plus-sign-digit-position register. The D register is primarily an intermediate buffer used during the transfer of information; in particular, the D register buffers all input to the Data Processor. It will be designated as rD.

THE B REGISTER

The B register is a four-digit-position register. Its primary purpose is to provide for automatic address-modification in a manner which will be specified; it is also used for tallying. The B register will be designated as rB.

THE C REGISTER

The C register is ten digits long. It is used to contain the instruction being executed. It is convenient to regard the C register as being divided into three parts:

1. The four high-order digit positions (1, 2, 3, and 4) of the C register contain what are called control digits.
2. Digit positions 5 and 6 always contain the operation code.
3. The four low-order digit positions (7, 8, 9, and 0) of the C register contain the address part of the instruction.

The C register is frequently called the control register. It will be designated as rC.

THE P REGISTER

The P register is a four-digit-position register. The P register controls the sequential operation of the Data Processor; it contains the address of the location from which the next instruction will be selected for execution. It is frequently called the program register. The P register will be designated as rP.

Four other registers, not in the control and arithmetic units, are worthy of note.

THE IB REGISTER

The IB register is a ten-digit-plus-sign-digit-position register in the storage control unit. It is used as a buffer between core storage and the control and arithmetic units. The IB register is frequently called the information buffer register. It will be designated as rIB.

THE E REGISTER

The E register is a four-digit-position register in the storage control unit. It is used for control purposes: the E register will always contain the address of a location to which access is being made under Data Processor or manual control. This register will be designated as rE.

The E register is used as a counter during execution of certain CARDATRON* instructions. When it is so used it is capable of counting *only* modulo the number of words in storage. For example, if there are only 6000 words of core storage, the E register counts modulo 6000.

THE CD REGISTER

The CD register is a ten-digit-plus-sign-position register in CARDATRON control unit 2. It is used for control purposes: the CD register will always contain a copy of the CARDATRON instruction which is being executed. The CD register will be designated as rCD.

THE T REGISTER

The T register is a ten-digit-position register in the magnetic-tape control unit. It is used for control purposes while magnetic-tape instructions are being executed. The T register will be designated as rT.

The contents of the A, R, D, B, P, C, and E Registers are displayed in 8, 4, 2, 1 binary-coded form on the control console with facilities for changing the contents of any one of them. This procedure is described in the *Handbook of Operating Procedures for the BURROUGHS 220.*

INFORMATION FLOW

The flow of information between core storage and the registers, between registers, between input-output equipment and the registers is parallel-serial. For example, all four bits of every digit are transferred in parallel; all 44 bits of a word are transferred between the information buffer register and core storage in parallel; all 16 bits of the P Register are transferred to the E Register in parallel: but information is transferred between the information buffer register and the D Register serially by digit. In the flow charts which follow this will be noted in detail.

There are three different types of information flow:

1. Operation cycle
   a. Fetch Phase
   b. Execute Phase
2. Input
3. Output

THE OPERATION CYCLE

The cycle of Data Processor operation is divided into two parts, the first of which is called the Fetch Phase, the second the Execute Phase. Each of these two names is descriptive of Data Processor operation for the duration of that part of the operation cycle: instructions are brought to the control unit during the Fetch Phase; they are executed during the Execute Phase. In normal operation the Fetch and Execute Phases follow each other alternately.

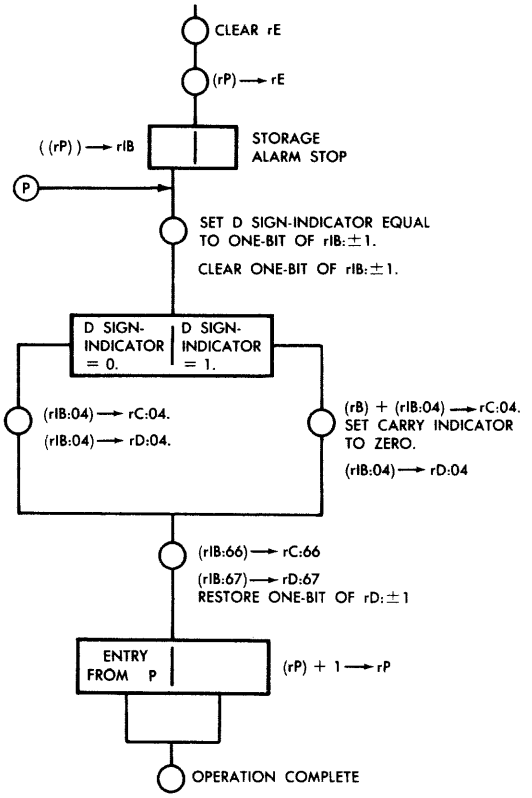


Figure 2-2. The Fetch Phase.

The Fetch Phase. Flow charts for the Fetch Phase are shown in Figures 2-2 and 2-3. A verbal description of this flow follows immediately:

* Trade-mark of the Burroughs Corporation.

1. At the beginning of the Fetch Phase, the contents of the P Register are transferred in parallel to the E Register.
2. The contents of the location whose address is in the E Register are transferred in parallel from core storage to the information buffer register. In this register the sign-digit position of the word is examined to determine whether B-register address-modification is intended.

If the one-bit of the sign digit is equal to 1 [in which case we may write (rIB:±1)/1=1], it is set to zero.

3. If the sign digit in rIB was an odd integer, B-register address-modification will occur as the ten low-order digits of the instruction word pass serially through the adder and into the C register. This transfer of information takes place in two parts: the address part of the word goes first; immediately after this transfer is completed the carry indicator in the adder is set to zero, so that, in case a five-digit sum was generated, there will be no overflow into the operation-code part of the instruction word. The second part of this transfer takes (rIB:66) into the C register.

If the sign digit in rIB was an even integer, no B-register address-modification will occur as the instruction is transferred to the C register.

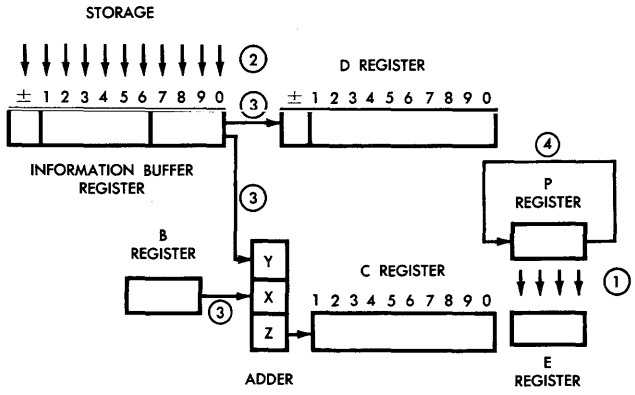


Figure 2-3. The Fetch Phase.

Simultaneously with the transfer to the C register, the instruction word, including the (possibly modified) sign digit, is transferred to the D register, but without address-modification, even if it is specified. When the transfer to the D register has been completed the one-bit of the sign digit will be restored in the sign-digit position in the D register. The word in the D register is, therefore, an exact copy of the instruction as it is represented in storage; it may be used for checking purposes by the system operator or the maintenance engineer.

4. Finally, the contents of the P register are counted up 1, unless the entry to the Fetch Phase was made as a result of interrupting a paper-tape or punched-card input instruction, in which case the counting up is inhibited. At the conclusion of the Fetch Phase the P register will contain the address of the location from which will come the next instruction selected for execution, if control continues in sequence.

The time for execution of the Fetch Phase is uniformly 90 microseconds.

The Execute Phase. During the Execute Phase, the instruction in the C register is executed. The nature of the Execute Phase and the time required to complete it are functions of the operation code, in particular, as well as the other digits in the C register. The detailed description of the Execute Phase for each operation comprises the bulk of this volume.

INPUT FLOW

Input information pulses to the Data Processor may be received from a Photoreader (paper tape), a CARDATRON Input Unit (punched cards), the Keyboard (manual), Magnetic-Tape Storage Unit, or DATAFILE* (magnetic tape) Unit. Figure 2-4 shows the information flow.

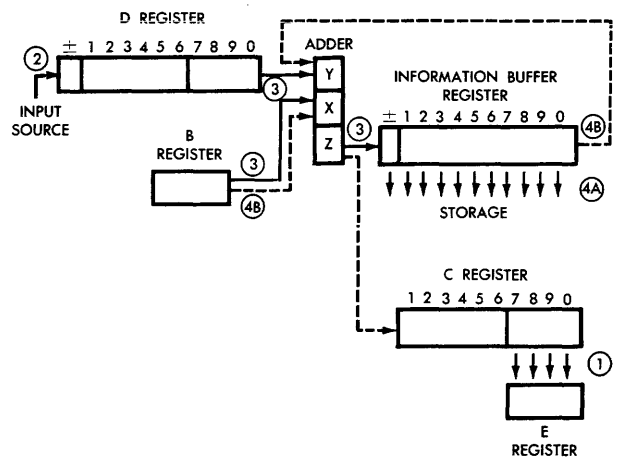


Figure 2-4. Input flow.

1. The address part of the C register is transferred in parallel to the E register to provide the address of the location in which will be stored the information which is destined for storage. The manner in which the contents of the E register are counted up or down for successive input words varies with the instruction being executed: on the one hand, (rC:04) may be counted up for each input word, after which it is transferred to rE; on the other hand, (rE) may be counted down. (See the flow chart in the description of the appropriate Execute Phase for details.)
2. The information is sent first to the D register where it is assembled as a word, digit by digit.

* Trade-mark of the Burroughs Corporation.

3. After each word is assembled it is transferred serially to the information buffer register (except in the case of manual operations). See *Handbook of Operating Procedures for the BURROUGHS 220* for details.
   a. If the word is to go to storage, B-register address modification will occur if it is specified.
   b. If the word is to go to the C register, B-register address-modification is postponed until the entry to the Fetch Phase is made.
4. a. If the word is to go to storage, the contents of the information buffer register are transferred, in parallel, to the location whose address is in the E register.
   b. If the word is to go to the C register, where it will be interpreted as a new instruction, entry to the Fetch Phase is made at connector P. (See Figure 2-2.) (4b) in Figure 2-4 shows the input flow in this case.

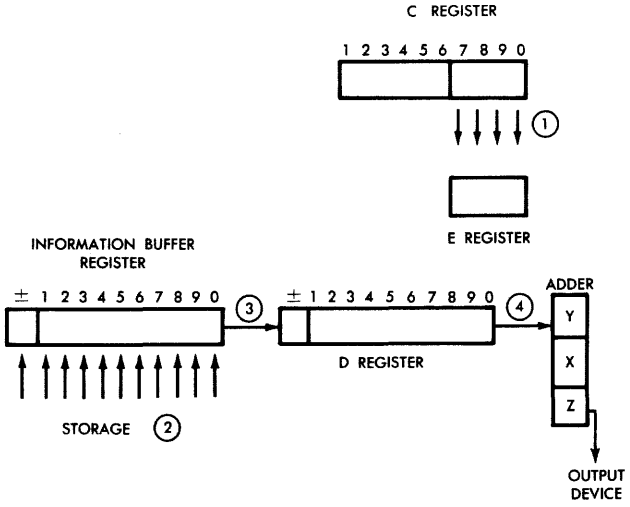


Figure 2-5. Output flow.

OUTPUT FLOW

Figure 2-5 shows the output flow.

1. The address part of the C register is transferred in parallel to the E register to provide the address of the location from which will be taken the first word destined for output. The manner in which the contents of the E register are counted up or down for successive output words varies with the instruction being executed: on the one hand (rC:04) may be counted up for each output word, after which it is transferred to rE; on the other hand, (rE) may be counted down. (See the flow chart in the description of the appropriate Execute Phase for details.)
2. The information is first transferred in parallel from storage to the information buffer register.
3. The contents of the information buffer register are then transferred serially to the D register.
4. The contents of the D register then pass serially through the adder to the output device.

EXCEPTIONAL CONDITIONS

1. The OVERFLOW Indicator may be turned on during the execution of several instructions (a list of them may be found on page 2-32).
2. In those operations which can manipulate partial-word fields, one must have specified sL, the partial-word boundaries. If $L>s+1$, $s\neq 0$, the partial-word will extend beyond the sign-digit position of the word. In this case field overflow is said to have occurred. This condition is detected by the Data Processor and results in a Program Check ALARM STOP.
3. If the low-order digit position of the IB, A, R, D, or B register contains a configuration equivalent to one of the decimal numbers 10 through 15, it will be sensed as an error. A Digit Check ALARM STOP will occur.
4. If an attempt is made to have access to a location not in the storage package (a nonexistent address) in the system, a Storage ALARM STOP will occur. For example, suppose the system has 3000 words of storage. An attempt to have access to location 4500 will produce the ALARM STOP.
5. If a nonexistent operation code is sensed in the operation-code part of the C register, a Program Check ALARM STOP will occur.
6. If a COMPARISON Indicator is "off" when it is interrogated (by a BRANCH, COMPARISON instruction; see page 2-33, ff), a Program Check ALARM STOP will occur.

THE EXECUTE PHASE

The remainder of the Data Processor chapter is devoted to descriptions of the Execute Phase of each operation with which the Data Processor is concerned exclusively.

HALT (HLT)

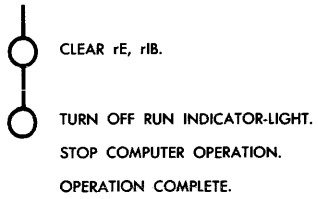
Operation Code. 00

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	i	i	i	i	O	p	i	i	i	i

Description of Operation. The Data Processor stops, ready to fetch the next instruction in sequence. The RUN indicator light is turned off.

Flow Chart.



Exceptional Conditions. None.

Remarks.

1. Although the address field is not used for addressing purposes, B-register address-modification will occur if it is specified. No Storage ALARM STOP, due to use of a nonexistent address, can occur, however.
2. If the Data Processor senses an unassigned operation code at the beginning of the Execute Phase, an ALARM STOP will occur; the Data Processor will stop with the PROGRAM CHECK alarm-indicator light on.

Register Status.

Register name	Contents after execution of HALT
A	Unchanged
R	Unchanged
D	± i i i i 0 0 i i i i
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 0 0 B[iiii]
E	Cleared

NO OPERATION (NOP)

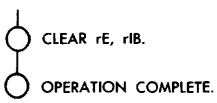
Operation Code. 01

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	i	i	i	i

Description of Operation. Perform no operation: at the time the operation code is sensed at the beginning of the Execute Phase, the execution is complete; the Data Processor proceeds to fetch the next instruction.

Flow Chart.



Exceptional Conditions. None.

Remarks. Although the address field is not relevant to the execution of this instruction, B-register address-modification will occur if it is specified. No Storage ALARM STOP, due to use of a nonexistent address, can occur, however.

Register Status.

Register name	Contents after execution of NOP
A	Unchanged
R	Unchanged
D	± i i i i 0 1 i i i i
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 0 1 B[iiii]
E	Cleared

CLEAR, ADD (CAD)

CLEAR, ADD ABSOLUTE (CAA)

Operation Code. 10

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	v	O	p	a	a	a	a

Definitions.

v: variation designator:
- v = 0: CLEAR, ADD will be executed.
- v = 1: CLEAR, ADD ABSOLUTE will be executed.

aaaa: address of base of location of augend.

Description of Operation.

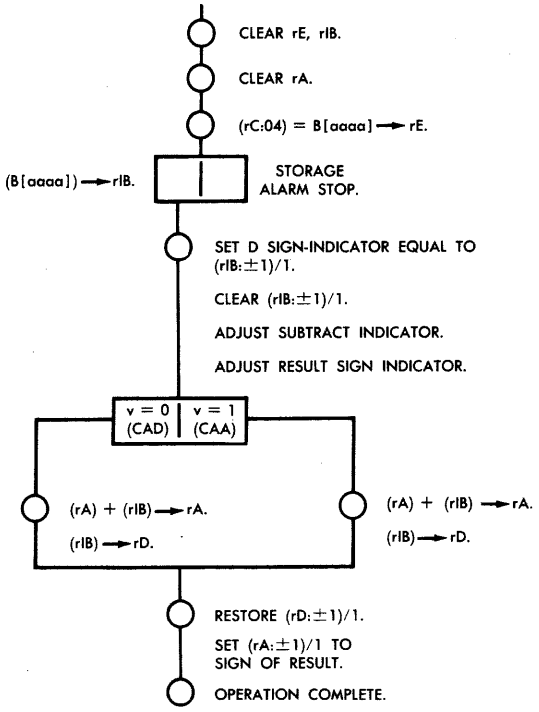
- v = 0: (B[aaaa]) replaces (rA).
- v = 1: |(B[aaaa])| replaces (rA).

Flow Chart. See page 2-8.

Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Remarks.

1. The CLEAR, ADD variation will be selected for execution if v ≠ 1.
2. CAD loads the A register with a word exactly as the word appears in storage.



3.

(B[aaaa]: ± 1)	$(rA: \pm 1)_a$	
	CAA	CAD
0	0	0
1	1	0
2	2	2
3	3	2
4	4	4
5	5	4
6	6	6
7	7	6
8	8	8
9	9	8

Register Status.

Register name	Contents after execution of CAD
A	(B[aaaa])
R	Unchanged
D	(B[aaaa])
B	Unchanged
P	$(rP)_b + 1$
C	i i i 0 l 0 B[aaaa]
E	B[aaaa]

Register name	Contents after execution of CAA
A	|(B[aaaa])|
R	Unchanged
D	(B[aaaa])
B	Unchanged
P	$(rP)_b + 1$
C	i i i l l 0 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Cleared
R	Unchanged
D	± i i i v l 0 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i v l 0 B[aaaa]
E	(B[aaaa])

CLEAR, SUBTRACT (CSU)

CLEAR, SUBTRACT ABSOLUTE (CSA)

Operation Code. 11

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	i	i	i	v	O	p	a	a	a	a

Definitions.

v: variation designator:
v = 0: CLEAR, SUBTRACT will be executed.
v = 1: CLEAR, SUBTRACT ABSOLUTE will be executed.

aaaa: address of base of location of subtrahend.

Description of Operation.

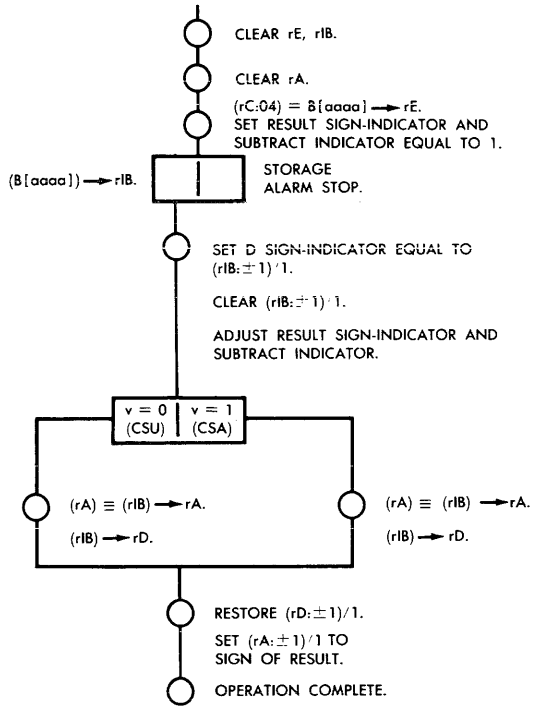
v = 0: (B[aaaa]) replaces (rA).
v = 1: |(B[aaaa])| replaces (rA).

Flow Chart. See page 2-9.

Exceptional Conditions. Storage ALARM STOP.

Remarks.

1. The CLEAR, SUBTRACT variation will be executed if $v \neq 1$.



2.

(B[aaaa]: ±1)	(rA: ±1)$_a$	
	CSU	CSA
0	1	1
1	0	1
2	3	3
3	2	3
4	5	5
5	4	5
6	7	7
7	6	7
8	9	9
9	8	9

Register Status.

Register name	Contents after execution of CSU
A	−(B[aaaa])
R	Unchanged
D	(B[aaaa])
B	Unchanged
P	(rP)$_b$ + 1
C	i i i | 0 | 1 1 | B[aaaa]
E	B[aaaa]

Register name	Contents after execution of CSA
A	−|(B[aaaa])|
R	Unchanged
D	(B[aaaa])
B	Unchanged
P	(rP)$_b$ + 1
C	i i i | 1 | 1 1 | B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Cleared
R	Unchanged
D	± | i i i | v | 1 1 | a a a a
B	Unchanged
P	(rP)$_b$ + 1
C	i i i | v | 1 1 | B[aaaa]
E	B[aaaa]

ADD (ADD)

ADD ABSOLUTE (ADA)

Operation Code. 12

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	v	O	p	a	a	a	a

Definitions.

v: variation designator:
v = 0: ADD will be executed.
v = 1: ADD ABSOLUTE will be executed.

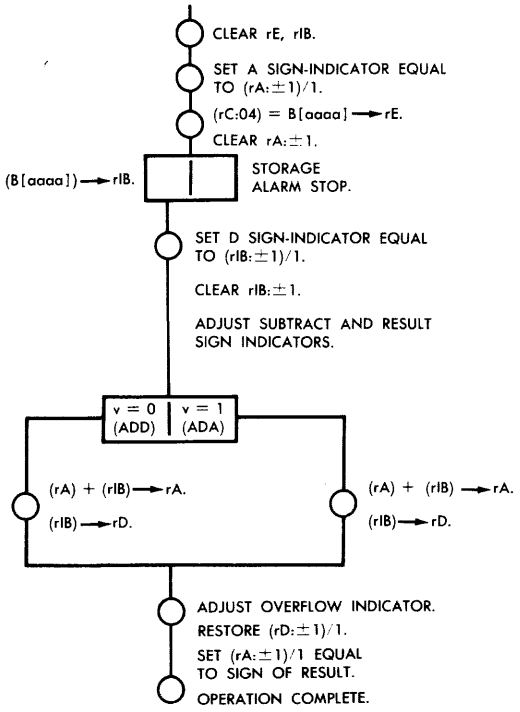
aaaa: address of base of location of augend.

Description of Operation.

v = 0: (rA) + (B[aaaa]) → rA.
v = 1: (rA) + |(B[aaaa])| → rA.

Flow Chart. See page 2-10.

Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.



Remarks.

1. The ADD variation will be executed if v ≠ 1.
2. The execution of both of these instructions can cause arithmetic overflow, in which case the OVERFLOW Indicator is set "on."
3. If the result of an ADD or ADA instruction is zero, the sign of the result is the same as the one-bit of the sign digit in the A register before execution of the instruction.

Register Status.

Register name	Contents after execution of ADD
A	$(rA)_b$ + B[aaaa])
R	Unchanged
D	(B[aaaa])*
B	Unchanged
P	$(rP)_b$ + 1
C	i i i 0 1 2 B[aaaa]
E	B[aaaa]

Register name	Contents after execution of ADA
A	$(rA)_b$ + (B[aaaa])
R	Unchanged
D	(B[aaaa])*
B	Unchanged
P	$(rP)_b$ + 1
C	i i i 1 1 2 B[aaaa]
E	B[aaaa]

* (rD:±1)/2 = (rD:±1)/4 = (rD:±1)/8 = 0; (rD:±1)/1 = (B[aaaa]:±1)/1; (rD:00) = (B[aaaa]:00)

Register name	Contents if Storage ALARM STOP occurs
A	(rA:±1) =0; $(rA:00)_a$ = $(rA:00)_b$
R	Unchanged
D	± i i i v 1 2 a a a a
B	Unchanged
P	$(rP)_b$ + 1
C	i i i v 1 2 B[aaaa]
E	(B[aaaa])

SUBTRACT (SUB)

SUBTRACT ABSOLUTE (SUA)

Operation Code. 13

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	v	O	p	a	a	a	a

Definitions.

v: variation designator:
- v = 0: SUBTRACT will be executed.
- v = 1: SUBTRACT ABSOLUTE will be executed.

aaaa: address of base of location of subtrahend.

Description of Operation.

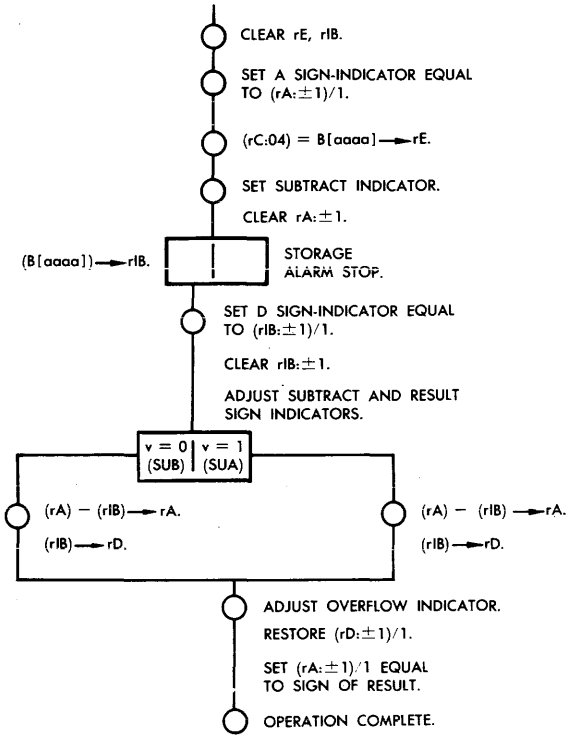
v = 0: (rA) − (B[aaaa]) → rA.
v = 1: (rA) − |(B[aaaa])| → rA.

Flow Chart. See page 2-11.

Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Remarks.

1. The SUBTRACT variation will be executed if v ≠ 1.
2. The execution of both of these instructions can cause arithmetic overflow, in which case the OVERFLOW Indicator is set "on."



3. If the result of a SUB or SUA instruction is zero, the sign of the result is the same as the one-bit of the sign digit in the A register before execution of the instruction.

Register Status.

Register name	Contents after execution of SUB
A	$(rA)_b - (B[aaaa])$
R	Unchanged
D	(B[aaaa])*
B	Unchanged
P	$(rP)_b + 1$
C	i i i | 0 | 1 3 | B[aaaa]
E	B[aaaa]

Register name	Contents after execution of SUA
A	$(rA)_b - \lvert(B[aaaa])\rvert$
R	Unchanged
D	(B[aaaa])*
B	Unchanged
P	$(rP)_b + 1$
C	i i i | 1 | 1 3 | B[aaaa]
E	B[aaaa]

* (rD:±1)/2 = (rD:±1)/4 = (rD:±1)/8 = 0; (rD:±1)/1 = (B[aaaa]:±1)/1: (rD:00) = (B[aaaa]:00).

Register name	Contents if Storage ALARM STOP occurs
A	$(rA:\pm 1)_a = 0$; $(rA:00)_a = (rA:00)$
R	Unchanged
D	± | i i i | v | 1 3 | a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i | v | 1 3 | B[aaaa]
E	(B[aaaa])

MULTIPLY (MUL)

Operation Code. 14

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	a	a	a	a

Definitions.

aaaa: address of base of location of multiplicand.

Description of Operation. The twenty-digit-long algebraic product, the contents of B[aaaa] multiplied by the contents of the A register, is generated. The ten low-order digits of the product replace the contents of the R register; the ten high-order digits of the product replace the contents of the A register. The sign of the product is inserted in the sign-digit position in both the A and R registers.

Flow Chart. See page 2-12.

Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Remarks. Execution time for MULTIPLY—exclusive of fetch time—is a function of the magnitude of the multiplier, $(rA)_b$. It may be calculated from the following formula:

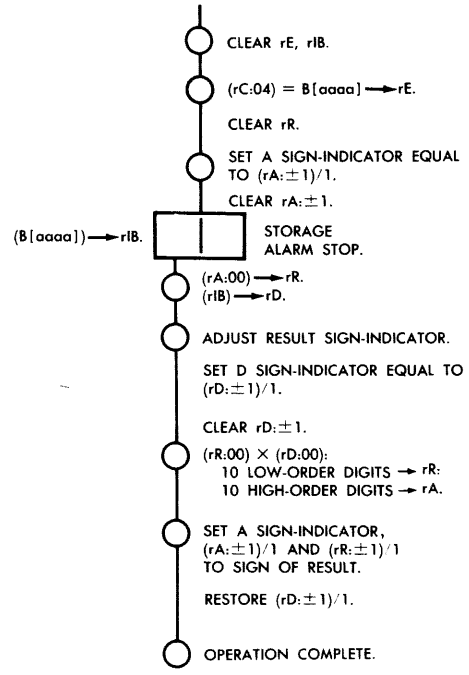
$$T = 90 + 5\sum_{k=0}^{9} M_k \ \mu s.$$

where

$M_k = 1$ if $(rA:k1) = 0$
$M_k = 13\,[(rA:k1)]+1$ if $1 \leq (rA:k1) \leq 5$
$M_k = 13\,[(11-(rA:k1)]$ if $6 \leq (rA:k1) \leq 9$

Hence, the following table:

(rA:k1)	0	1	2	3	4	5	6	7	8	9
M_k	1	14	27	40	54	66	65	52	39	26



Register Status.

Register name	Contents after execution of MUL
A	High-order digits of product
R	Low-order digits of product $(rR:\pm1)_a = (rA:\pm1)_a$
D	(B[aaaa])*
B	Unchanged
P	$(rP)_b + 1$
C	i i i i | 1 4 | B[aaaa]
E	B[aaaa]

* $(rD:\pm1)/2 = (rD:\pm1)/4 = (rD:\pm1)/8 = 0$;
$(rD:\pm1)/1 = (B[aaaa]:\pm1)/1$; $(rD:00) = (B[aaaa]:00)$.

Register name	Contents if Storage ALARM STOP occurs
A	$(rA:\pm1)_a = 0$; $(rA:00)_a = (rA:00)_b$
R	Cleared
D	± | i i i i | 1 4 | a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i i | 1 4 | B[aaaa]
E	B[aaaa]

DIVIDE (DIV)

Operation Code. 15

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	a	a	a	a

Definitions.

aaaa: address of base of location of divisor.

Description of Operation. The dividend is the twenty-digit number whose high-order digits are the contents of the A register and whose low-order digits are the contents of the R register. The sign of the A register is taken to be the sign of the dividend; the sign of the R register is not relevant.

The divisor is the contents of B[aaaa].

Before the arithmetic process is begun, the absolute value of the divisor is compared with the absolute value of the contents of the A register. Then:

1. If the absolute value of the divisor is greater than the absolute value of the contents of the A register, the process of division begins. The process terminates when a ten-digit quotient has been generated: The quotient, with sign, replaces the contents of the A register. The remainder replaces the contents of the R register; the sign of the remainder is the same as the sign of the dividend. On the other hand,
2. If the absolute value of the divisor is less than or equal to the absolute value of the contents of the A register, the OVERFLOW Indicator will be set "on," and the execution of the DIVIDE instruction will terminate, leaving the contents of the A and R registers unaltered. Depending upon the setting of the Overflow mode (see page 2-37), the operation either will halt or continue in sequence.

Flow Chart. See page 2-13.

Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Remarks. Execution time for DIVIDE—exclusive of fetch time—is a function of the magnitude of the quotient $(rA)_a$, if overflow does not occur. It may be calculated from the following formula:

$$T = 3895 + 60\sum_{k=0}^{4}[(rA:2k+1, 1)_a - (rA:2k, 0)_a]\ \mu s$$

$$= 3895 + 60\,[(rA:11) - (rA:21) + (rA:31) - (rA:41) + (rA:51) - (rA:61) + (rA:71) - (rA:81) + (rA:91) - (rA:01)\,].$$

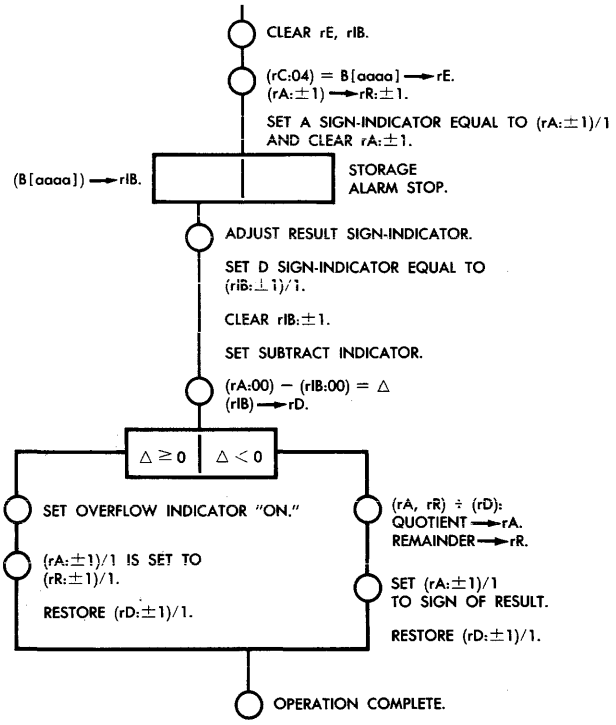
Register Status.

Register name	Contents after execution of DIV
A	Quotient
R	Remainder
D	(B[aaaa])*
B	Unchanged
P	$(rP)_b + 1$
C	i i i i | 1 5 | B[aaaa]
E	B[aaaa]

* (rD:±1)/2 = (rD:±1)/4 = (rD:±1)/8 = 0;
(rD:±1)/1 = (B[aaaa]:±1)/1; (rD:00) = (B[aaaa]:00).

Register name	Contents if the OVERFLOW Indicator is set "on"
A	One-bit $(rA:\pm1)_a$ = one-bit $(rA:\pm1)_b$ other bits $(rA:\pm)_a = 0$ $(rA:00)_a = (rA:00)_b$
R	$(rR:\pm1)_a = (rA:\pm1)_b$ $(rR:00)_a = (rR:00)_b$
D	(B[aaaa])
B	Unchanged
P	$(rP)_b + 1$
C	i i i i | 1 5 | B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	$(rA:\pm1) = 0$; $(rA:00)_a = (rA:00)_b$
R	$(rR:\pm1) = (rA:\pm1)_b$ $(rR:00)_a = (rR:00)_b$
D	± | i i i i | 1 5 | a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i i | 1 5 | B[aaaa]
E	B[aaaa]



ROUND (RND)

Operation Code. 16

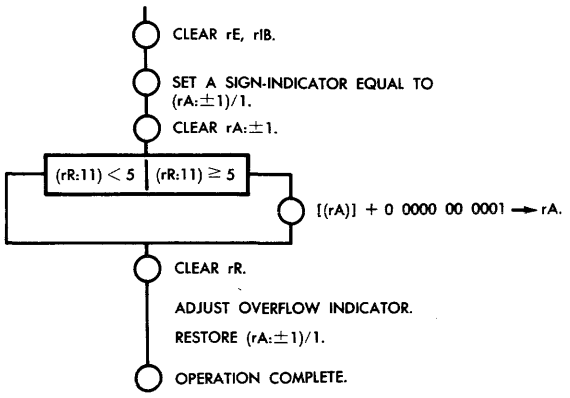
Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	i	i	i	i

Description of Operation.

If (rR:11) <5, clear rR.
If (rR:11) ≥5, |(rA)| is increased by + 0000 00 0001.
Then rR is cleared.

Flow Chart.



Exceptional Conditions. None

Remarks.

1. The execution of this instruction can cause arithmetic overflow, in which case the OVERFLOW Indicator is set to "on."

2. Although the address field is not relevant to the execution of this instruction, B-register address-modification will occur if it is specified. No Storage ALARM STOP, due to use of a nonexistent address, can occur, however.

Register Status.

Register name	Contents after execution of RND
A	$(rA)_b$, rounded
R	Cleared
D	± i i i i 1 6 i i i i
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 1 6 B[iiii]
E	Cleared

EXTRACT (EXT)

Operation Code. 17

Instruction Format.

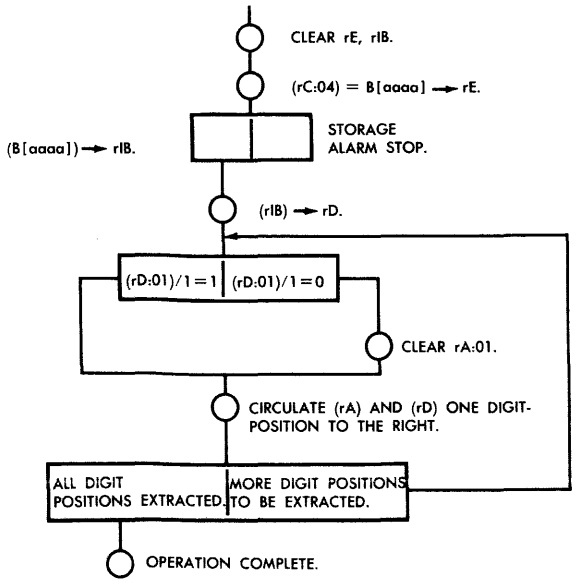
± 1 2 3 4 5 6 7 8 9 0
± i i i i Op a a a a

Definition.

aaaa: Address of base of location of extractor.

Description of Operation. Wherever the extractor—which is the contents of B[aaaa]—has an even digit, the corresponding digit in rA is replaced by 0; wherever the extractor has an odd digit, the corresponding digit in rA is not altered.

Flow Chart.



Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Register Status.

Register name	Contents after execution of EXT
A	See description of operation
R	Unchanged
D	(B[aaaa])
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 1 7 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± i i i i 1 7 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 1 7 B[aaaa]
E	B[aaaa]

COMPARE FIELD A (CFA)

COMPARE FIELD R (CFR)

Operation Code. 18

Instruction Format.

± 1 2 3 4 5 6 7 8 9 0
± s L f v Op a a a a

Definitions.

s: partial-word designator:
f = 0: s is not relevant.
f = 1: s designates the position, within the word, of the low-order digit of each partial-word operand.

L: partial-word designator:
f = 0: L is not relevant.
f = 1: L specifies the number of digits in each partial-word operand.

f: field designator:
f = 0: entire words will be used as operands.
f = 1: the contents of the partial-word fields defined by digits s and L will be used as operands.

v: variation designator:
v = 0: COMPARE FIELD A will be executed.
v = 1: COMPARE FIELD R will be executed.

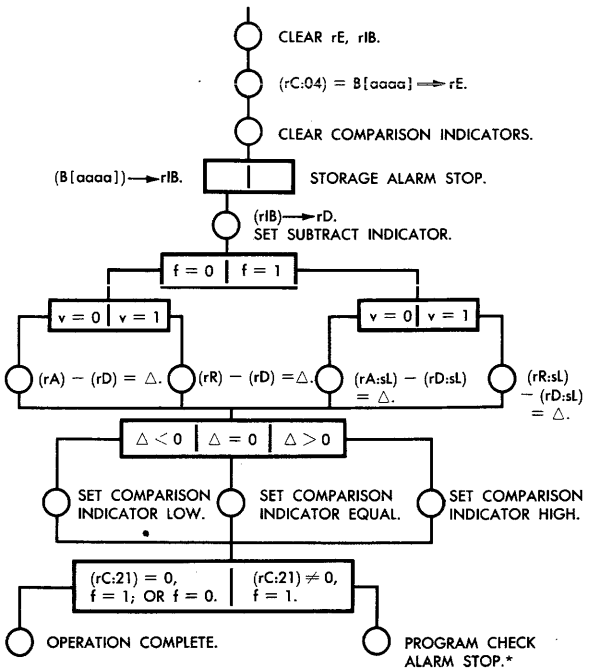
aaaa: address of base of location of comparator.

Description of Operation. v = 0: COMPARE FIELD A will be executed. Compare the contents of the specified field in the A register with the corresponding field in B[aaaa]. According as the contents of the specified part of the A register are greater than, equal to, or less than, the corresponding part of B[aaaa], set the COMPARISON Indicator to HIGH, EQUAL, or LOW.

v = 1: COMPARE FIELD R will be executed.

Except that the R register is used, this variation is identical with the one specified by v = 0.

Flow Chart.



*IF f = 1, AS EACH DIGIT OF THE PARTIAL-WORD DIFFERENCE IS GENERATED, L [i.e., (rC:21)] IS COUNTED DOWN. IF, AT THE START, L > s + 1, s ≠ 0, THEN AT THE END OF THE COMPARISON (rC:21) WILL BE DIFFERENT FROM ZERO, AND FIELD OVERFLOW WILL BE DETECTED.

Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. Program Check ALARM STOP due to field overflow.

Remarks.

1. The COMPARE FIELD A variation will be executed if v ≠ 1.
2. If f is an even integer, this has the same effect as f = 0; if f is an odd integer, this has the same effect as f = 1.
3. a. If the sign-digit position of a word is *not* included in the field specified as operand, then the comparison may be considered to be made with respect to the absolute value of each operand. On the other hand,

   b. If the sign-digit position of a word is included in the field specified as operand, then the comparison is algebraic, in which case the contents of the sign-digit position are regarded as having the following order:

   $3 < 2 < 1 < 0 < 7 < 6 < 5 < 4 < 8 < 9.$

   Hence, 3 9999 99 9999 < ... < 3 0000 00 0000 < 2 9999 99 9999 < ... < 2 0000 00 0000 < 1 9999 99 9999 < ... < 1 0000 00 0000 < 0 0000 00 0000 < ... < 0 9999 99 9999 < 7 0000 00 0000 < ... < 7 9999 99 9999 < 6 0000 00 0000 < ... < 6 9999 99 9999 < 5 0000 00 0000 < ... < 5 9999 99 9999 < 4 0000 00 0000 < ... < 4 9999 99 9999 < 8 0000 00 0000 < ... < 8 9999 99 9999 < 9 0000 00 0000 < ... < 9 9999 99 9999.

   c. One consequence of the ordering described above is that the sign digit must never be included in a comparison field if that field is alphanumeric; otherwise, the inverse of the natural alphabetic order will be determined.

   d. A second consequence of the ordering described above is that + 0000 00 0000 > − 0000 00 0000.

4. The order relationships of the content of the sign-digit position were determined as follows: the principal requirement is that negative numbers shall precede positive numbers; hence, 1 < 0. A second requirement—sorting—is that alphanumeric information shall precede numeric; hence 2 < 1 < 0. The remaining order relationships are just by-products of the way in which the required ordering of 2, 1, and 0 is achieved. The technique used is known as the "three's complement" method: the 1-bit and the 2-bit of the sign digit are complemented if, and only if, the 8-bit is 0. The results are displayed in the following table:

Decimal Digit in Sign Position	Binary Representation	Three's Complemented	Decimal Equivalent (= order)
3	0011	0000	0
2	0010	0001	1
1	0001	0010	2
0	0000	0011	3
7	0111	0100	4
6	0110	0101	5
5	0101	0110	6
4	0100	0111	7
8	1000	1000	8
9	1001	1001	9

Register Status.

Register name	Contents after execution of either CFA or CFR
A	Unchanged
R	Unchanged
D	(B[aaaa])
B	Unchanged
P	$(rP)_b + 1$
C	0 0 f v 1 8 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± s L f v 1 8 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	s L f v 1 8 B[aaaa]
E	B[aaaa]

Register name	Contents after Program Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	(B[aaaa])
B	Unchanged
P	$(rP)_b + 1$
C	0 * f v 1 8 B[aaaa]
E	B[aaaa]

* L — s — 1

ADD TO LOCATION (ADL)

Operation Code. 19

Instruction Format.

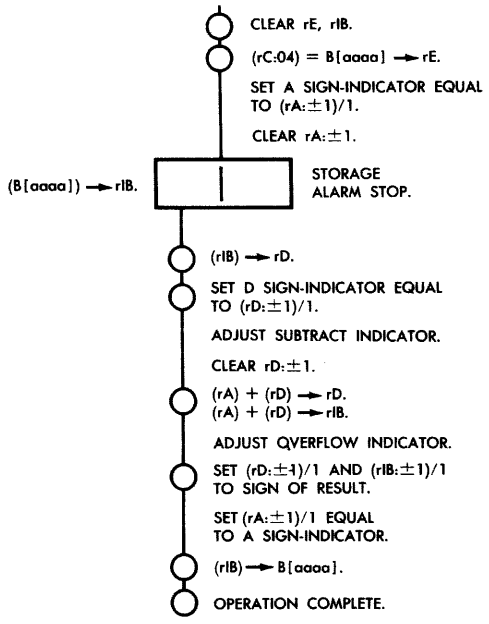
	±	1	2	3	4	5	6	7	8	9	0
	±	i	i	i	i	O	p	a	a	a	a

Definition.

aaaa: address of base of location of augend.

Description of Operation. (rA) + (B[aaaa]) → B[aaaa].

Flow Chart.



Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Remarks.

1. The execution of this instruction can cause arithmetic overflow, in which case the OVERFLOW Indicator is set to "on."
2. The arithmetic operation of ADL is essentially the same as that of ADD.

Register Status.

Register name	Contents after execution of ADL
A	Unchanged*
R	Unchanged
D	$(rA)_b + (B[aaaa])$
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 1 9 B[aaaa]
E	B[aaaa]

* Except that $(rA:\pm 1)/2 = (rA:\pm 1)/4 = (rA:\pm 1)/8 = 0$.

Register name	Contents if Storage ALARM STOP occurs
A	$(rA:\pm 1) = 0$, $(rA:00)_a = (rA:00)_b$
R	Unchanged
D	± i i i i 1 9 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 1 9 B[aaaa]
E	B[aaaa]

INCREASE B, BRANCH (IBB)

Operation Code. 20

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	n	n	n	n	O	p	a	a	a	a

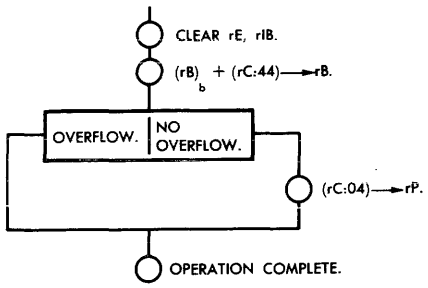
Definitions.

nnnn: modifier for the contents of the B register.

aaaa: address of base of location containing alternate instruction.

Description of Operation. Increase (rB) by nnnn. If no overflow occurs, set (rP) = B[aaaa] (i.e., prepare to branch to location containing alternate instruction); if overflow occurs, control continues in sequence. Overflow of the B register does *not* set the OVERFLOW Indicator "on."

Flow Chart.



Exceptional Conditions. None.

Remarks. The following statement embodies a method for determining whether branching will occur: if $(rB)_a \geq (rB)_b$, branching will occur; if $(rB)_a < (rB)_b$, control continues in sequence.

Register Status.

Register name	Contents after execution of IBB
A	Unchanged
R	Unchanged
D	± n n n n 2 0 a a a a
B	Low-order digits of sum, $S = (rB)_b + nnnn$
P	$(rP)_b + 1$, if $S > 9999$ (overflow) B[aaaa], if $S \leq 9999$ (no overflow)
C	n n n n 2 0 B[aaaa]
E	Cleared

Examples.

1. $(rP)_b =$ 0412
   $(rB)_b =$ 9984
   $(rC) =$ 0001 20 0396
   $(rB)_a =$ 9985
   $(rP)_a =$ 0396
   (Branch; no overflow)

2. $(rP)_b =$ 0412
   $(rB)_b =$ 9997
   $(rC) =$ 0005 20 0396
   $(rB)_a =$ 0002
   $(rP)_a =$ 0413
   (No branch; overflow)

DECREASE B, BRANCH (DBB)

Operation Code. 21

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	n	n	n	n	O	p	a	a	a	a

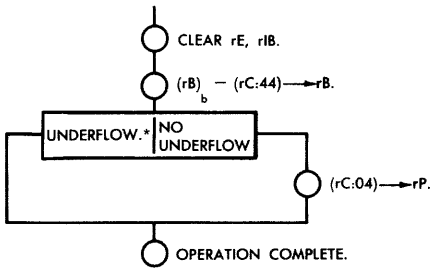
Definitions.

nnnn: modifier for the contents of the B register.

aaaa: address of base of location containing alternate instruction.

Description of Operation. Decrease (rB) by nnnn. If no underflow occurs set (rP) = B[aaaa] (i.e., prepare to branch to location containing alternate instruction); if underflow occurs, control continues in sequence.

Flow Chart.



*WHEN UNDERFLOW DOES OCCUR, THE COMPLEMENT OF THE ALGEBRAIC DIFFERENCE $(rB)_b - (rC:44)$, WILL APPEAR IN rB.

Exceptional Conditions. None.

Remarks. The following statement embodies a method for determining whether branching will occur: if $(rB)_a \leq (rB)_b$, branching will occur; if $(rB)_a > (rB)_b$, control continues in sequence.

Register Status.

Register name	Contents after execution of DBB
A	Unchanged
R	Unchanged
D	± n n n n 2 1 a a a a
B	Low-order digits of D = |rB$_b$| − |nnnn|
P	(rB)$_b$ + 1, if underflow B[aaaa], if no underflow
C	n n n n 2 1 B[aaaa]
E	Cleared

Examples.

1. $(rP)_b$ = 0412
   $(rB)_b$ = 0006
   (rC) = 0002 21 0396
   $(rB)_a$ = 0004
   $(rP)_a$ = 0396
   (Branch; no underflow.)

2. $(rP)_b$ = 0412
   $(rB)_b$ = 0002
   (rC) = 0005 21 0396
   $(rB)_a$ = 9997
   $(rP)_a$ = 0413
   (No branch; underflow.)

FLOATING ADD (FAD)

FLOATING ADD ABSOLUTE (FAA)

Operation Code. 22

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	n	i	i	v	O	p	a	a	a	a

Definitions.

n: normalizing limiter.

v: variation designator:
v = 0: FLOATING ADD will be executed.
v = 1: FLOATING ADD ABSOLUTE will be executed.

aaaa: address of base of location of augend.

Description of Operation. Both variations treat the operands like floating-point numbers:

v = 0: (rA) + (B[aaaa]) → rA.
v = 1: (rA) + |(B[aaaa])| → rA.

Let m be the number of digit positions through which the sum is shifted to obtain the sum in normalized form. If m > n, the Data Processor will halt at the end of the Execute Phase. n = 8, 9, or 0 (where 0 means 10) is meaningless and irrelevant.

Flow Chart. See page 2-19.

Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Remarks.

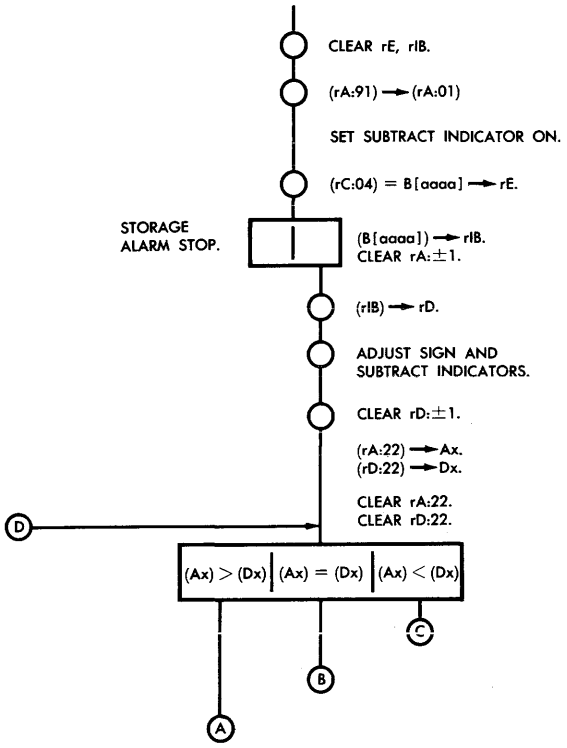
1. The FLOATING ADD variation will be selected for execution if v ≠ 1.
2. The execution of these instructions can cause arithmetic overflow, in which case the OVERFLOW Indicator will be turned "on." Arithmetic overflow occurs when the machine-coded exponent exceeds 99.

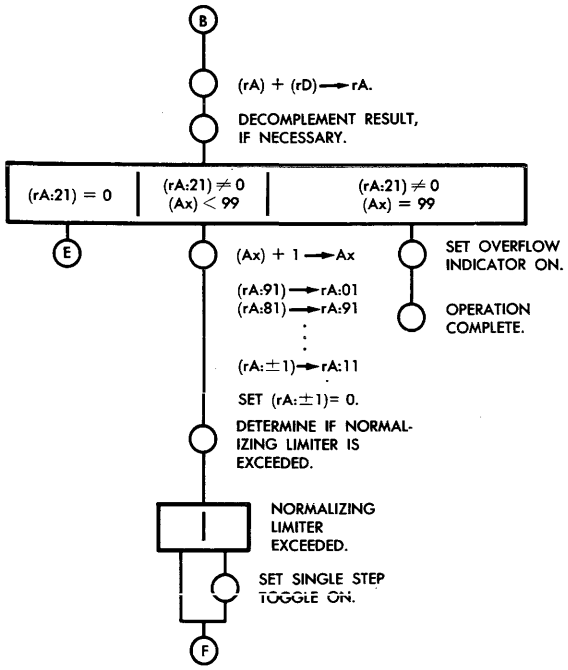
3. If the result of a FAD or FAA instruction is zero, the A register—which will contain the sum—is cleared. Hence, a zero sum has the form + 00 0000 0000, that is, 0×10^{-50}.

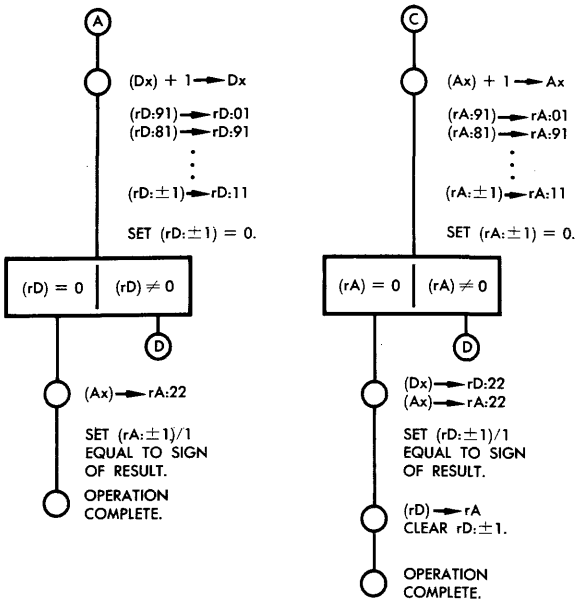
4. In adjusting the exponents so that addition can take place, low-order digits of one of the operands—the one whose exponent is smaller—are lost. Rounding does not occur; that is, the highest-order digit discarded is not examined to determine whether the low-order digit retained should be increased by 1.

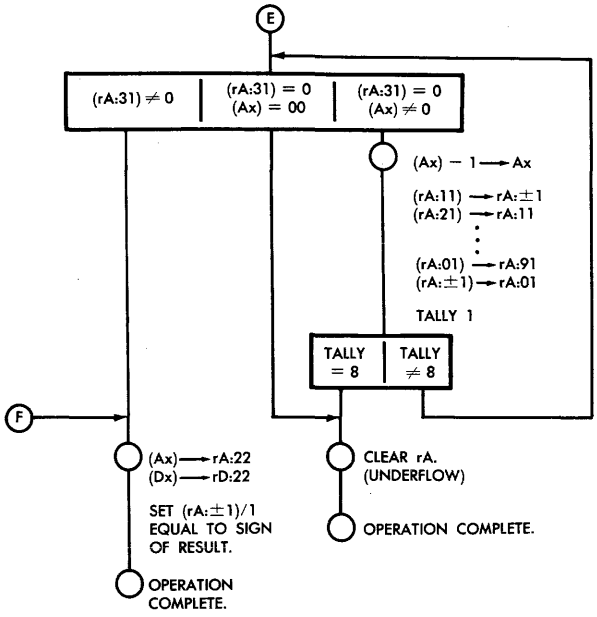
5. During the normalization process the number of digit positions, m, through which the sum is to be shifted is determined. If $m > n$, $n \neq 8$, 9, or 0, where n is the normalizing limiter, then $(rC:11)_a = 10 - (m - n)$; if $m = n$, $(rC:11)_a = 0$. If $n = 8$, 9, or 0, $(rC:11)_a = 0$.

After normalization, if $(rC:11) \neq 0$, the Single Step Toggle (SST) will be turned on. Whenever SST is turned on, the Data Processor will stop at the end of the cycle—Fetch Phase or Execute Phase—and wait for a signal to resume (automatic) operation.









Register Status.

Register name	Contents after execution of FAD
A	$(rA)_b + (B[aaaa])$
R	Unchanged
D	0 1 0 0 0 0 0 0 0 0 0
B	Unchanged
P	$(rP)_b + 1$
C	0 i i 0 2 2 B[aaaa]
E	B[aaaa]

Register name	Contents after execution of FAA
A	$(rA)_b + (B[aaaa])$
R	Unchanged
D	0 1 0 0 0 0 0 0 0 0 0
B	Unchanged
P	$(rP)_b + 1$
C	0 i i 1 2 2 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± n i i v 2 2 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	n i i v 2 2 B[aaaa]
E	B[aaaa]

Register name	Contents if OVERFLOW Indicator is "on"
A	Unnormalized sum without exponent
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	n i i v 2 2 B[aaaa]
E	B[aaaa]

Register name	Contents if HALT occurs
A	Nomalized sum
R	Unchanged
D	0 1 0 0 0 0 0 0 0 0 0
B	Unchanged
P	$(rP)_b + 1$
C	* i i v 2 2 B[aaaa]
E	B[aaaa]

* 10 − (m − n).

FLOATING SUBTRACT (FSU)

FLOATING SUBTRACT ABSOLUTE (FSA)

Operation Code. 23

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	n	i	i	v	O	p	a	a	a	a

Definitions.

n: normalizing limiter.

v: variation designator:
v = 0: FLOATING SUBTRACT will be executed.
v = 1: FLOATING SUBTRACT ABSOLUTE will be executed.

aaaa: address of base of location of subtrahend.

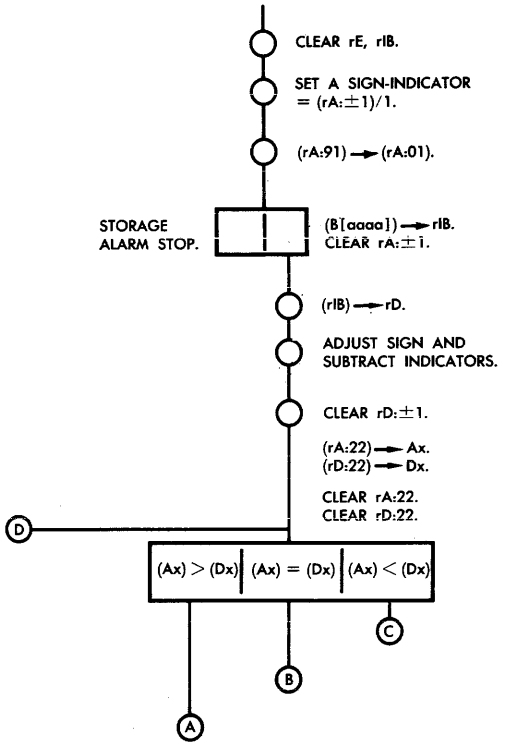
Description of Operation. Both variations treat the operands like floating-point numbers:

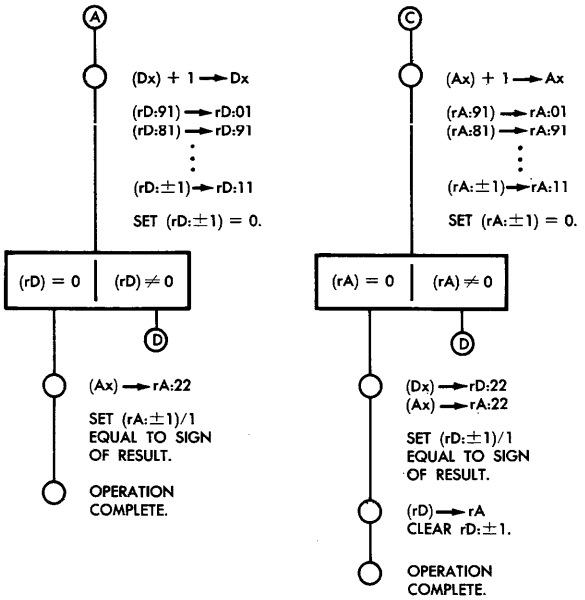
v = 0: (rA) − (B[aaaa]) → rA.
v = 1: (rA) − |(B[aaaa])| → rA.

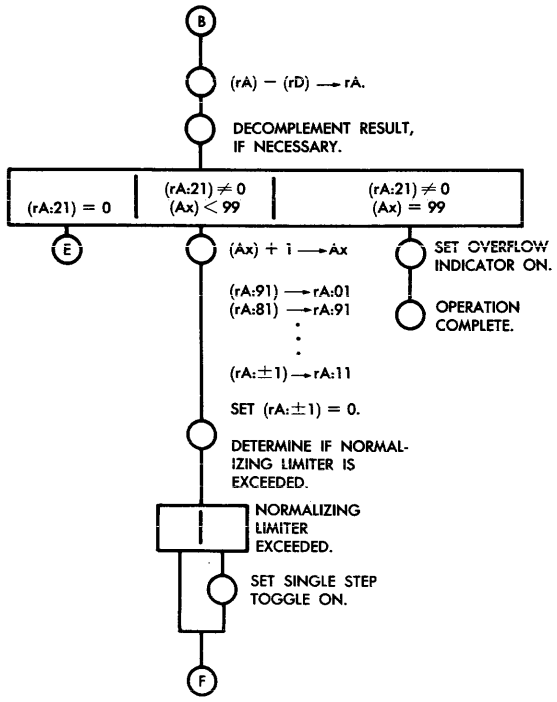
Let m be the number of digit positions through which the difference is shifted to obtain the difference in normalized form. If m > n, the Data Processor will halt at the end of the Execute Phase. n = 8, 9, or 0 (where 0 means 10) is meaningless and irrelevant.

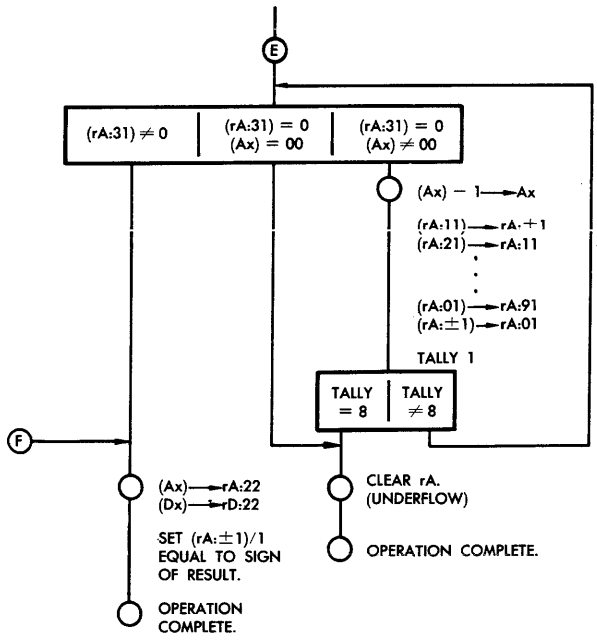
Flow Chart. See page 2-21.

Exceptional Conditions. ALARM STOP due to use of a nonexistent address.









Remarks.

1. The FLOATING SUBTRACT variation will be selected for execution if v ≠ 1.

2. The execution of these instructions can cause arithmetic overflow, in which case the OVERFLOW Indicator will be turned "on." Arithmetic overflow occurs when the machine-coded exponent exceeds 99.

3. If the result of a FSU or FSA instruction is zero, the A register—which will contain the difference—is cleared. Hence, a zero difference has the form + 00 0000 0000, that is 0×10^{-50}.

4. In adjusting the exponents so that subtraction can take place, low-order digits of one of the operands—the one whose exponent is smaller—are lost. Rounding does not occur; that is, the highest-order digit lost is not examined to determine whether the low-order digit retained should be increased by 1.

5. During the normalization process the number of digit positions, m, through which the difference is to be shifted is determined. If m > n, n≠ 8, 9, or 0, where n is the normalizing limiter, then $(rC:11)_a = 10 - (m - n)$; if m = n, $(rC:11)_a = 0$. If n = 8, 9, or 0, $(rC:11)_a = 0$.

After normalization, if (rC:11) ≠ 0, the Single Step Toggle (SST) will be turned "on." Whenever SST is turned on, the Data Processor will stop at the end of the cycle—Fetch Phase or Execute Phase—and wait for a signal to resume (automatic) operation.

Register Status.

Register name	Contents after execution of FSU
A	$(rA)_b - \|(B[aaaa])\|$
R	Unchanged
D	0 1 0 0 0 0 0 0 0 0 0
B	Unchanged
P	$(rP)_b + 1$
C	0 i i 0 2 3 B[aaaa]
E	B[aaaa]

Register name	Contents after execution of FSA
A	$(rA)_b - (B[aaaa])$
R	Unchanged
D	0 1 0 0 0 0 0 0 0 0 0
B	Unchanged
P	$(rP)_b + 1$
C	0 i i 1 2 3 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± n i i v 2 3 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	n i i v 2 3 B[aaaa]
E	B[aaaa]

Register name	Contents if OVERFLOW Indicator is "on"
A	Unnormalized difference without exponent
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	n i i v 2 3 B[aaaa]
E	B[aaaa]

Register name	Contents if HALT occurs
A	Normalized difference
R	Unchanged
D	0 1 0 0 0 0 0 0 0 0 0
B	Unchanged
P	$(rP)_b + 1$
C	* i i v 2 3 B[aaaa]
E	B[aaaa]

FLOATING MULTIPLY (FMU)

Operation Code. 24

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	a	a	a	a

Definition.

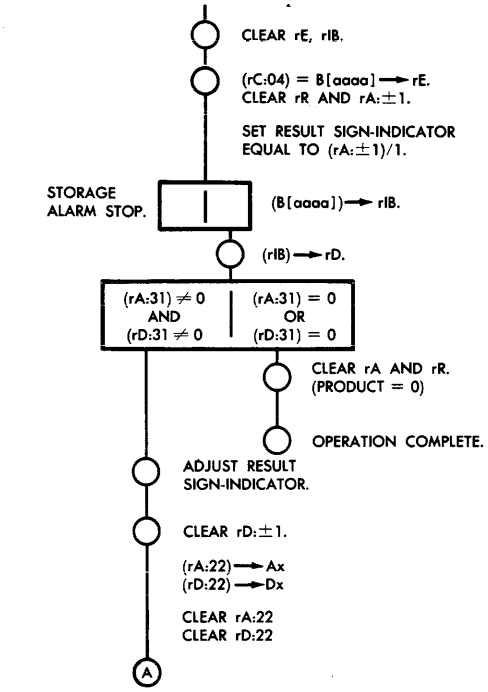
aaaa: address of base of location of multiplicand.

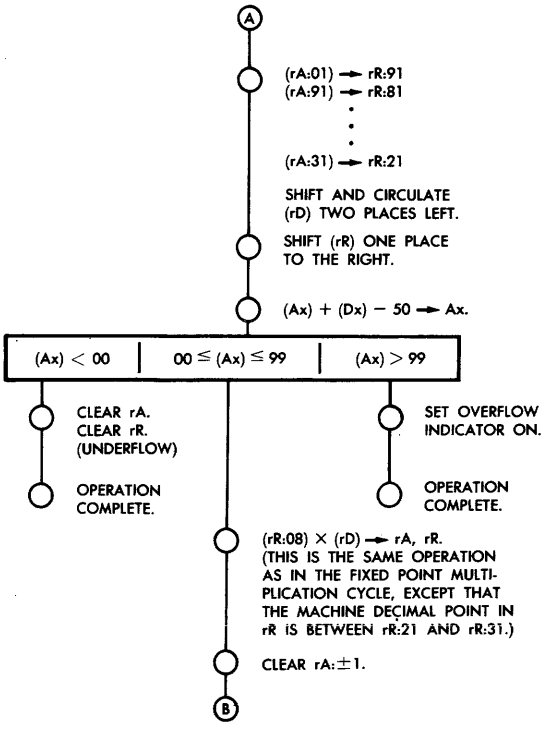
Description of Operation. This instruction treats the operands like floating-point numbers:

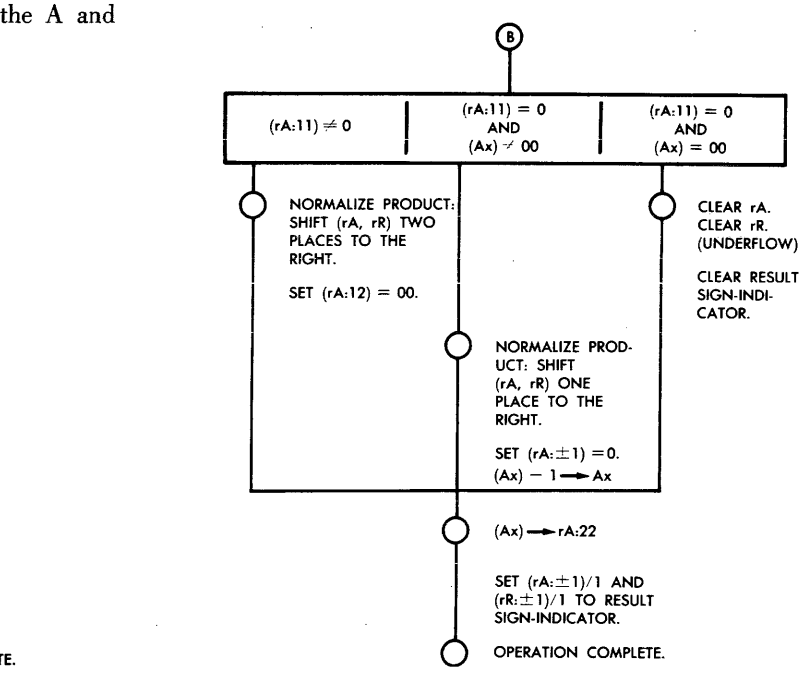
The floating-point product, (rA) × (B[aaaa]), is generated. The two-digit exponent and the eight high-order digits of the product's mantissa replace the contents of the A register. The seven or eight low-order digits of the product's mantissa are inserted in the high-order end of the R register with the three or two low-order digit positions of the R register set to zero. The sign of the product

is inserted in the sign-digit position in both the A and R registers.

Flow Chart.







Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. "Spurious exponent overflow" can occur in case exponent arithmetic—which takes place before the mantissas are multiplied—yields 100 for the adjusted sum of the exponents of (rA) and (B[aaaa]): if the adjusted sum of these exponents is 100, the OVERFLOW Indicator is turned "on," even though normalization of the product might yield a valid exponent, namely, 99, and the Execute Phase is terminated.

   For example, suppose (rA) = 0 75 1000 0000 = (aaaa); the instruction is 0 0000 FMU aaaa. The execution of this instruction should yield (rA) = 0 99 1000 0000. Instead, the OVERFLOW Indicator is turned "on."

Remarks.

1. Exponent overflow can occur, in which case the OVERFLOW Indicator is turned "on."
2. Exponent underflow causes both the A and R registers to be cleared, but is not otherwise indicated.
3. In case the mantissa of either operand is not normalized—in which case at least one of (rA:31) and (rD:31) is 0—it is assumed that the product will be zero. The operation is terminated as soon as this condition is determined to exist—85 μs are required —with the A and R registers cleared.

4. In normalizing the product, the A and R registers are shifted together one or two places to the right (see flow chart). Hence, the R register will contain either seven or eight low-order digits of the product's mantissa. It is not possible to predict which will be the case unless one knows the magnitudes of the multiplier and multiplicand.

Register Status.

Register name	Contents after execution of FMU
A	Exponent and eight high-order digits of product
R	Low-order digits of product as described
D	(B[aaaa])*
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 2 4 B[aaaa]
E	B[aaaa]

* However, $(rD:\pm1) = 0$; that is, the sign digit of the multiplicand may not be restored.

Register name	Contents if Storage ALARM STOP occurs.
A	Unchanged
R	Unchanged
D	± i i i i 2 4 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 2 4 B[aaaa]
E	B[aaaa]

Register name	Contents if (exponent) overflow occurs during exponent arithmetic
A	Cleared
R	$(rR:23) = 000$; $(rR:08)_a = (rR:08)_b$
D	$(rD:\pm1) = 0$; $(rD:88)_a = (rD:08)_b$; $(rD:02) = 00$
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 2 4 B[aaaa]
E	(B[aaaa])

Register name	Contents if (exponent) underflow occurs
A	Cleared
R	Cleared
D	$(rD:23) = 000$; $(rD:86)_a = (rD:06)_b$; $(rD:02) = 00$
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 2 4 B[aaaa]
E	(B[aaaa])

FLOATING DIVIDE (FDV)

Operation Code. 25

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	i	i	i	i	O	p	a	a	a	a

Definition.

aaaa: address of base of location of divisor.

Description of Operation. This instruction treats the operands like floating-point numbers:

Dividend: Exponent is (rA:22); high-order digits of mantisaa are (rA:08), low-order digits of mantissa are (rR:88).

Divisor: (B[aaaa]).

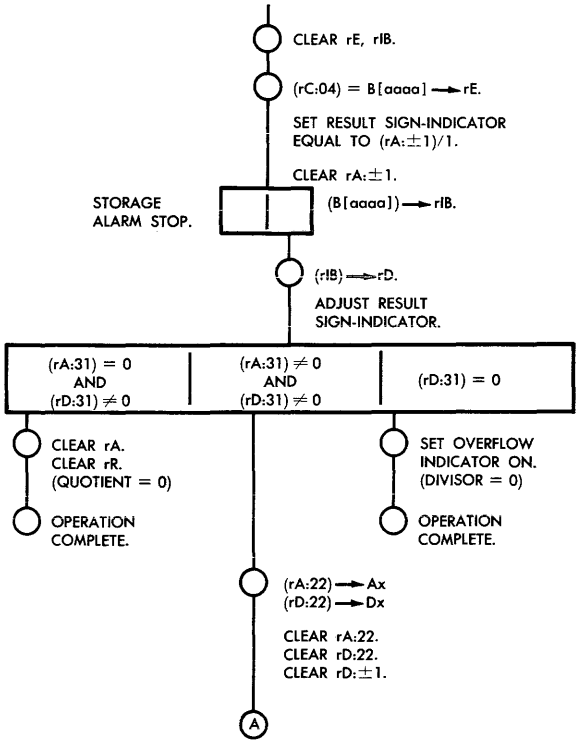
Quotient: Exponent is (rA:22); high-order digits of mantissa are (rA:08); low-order digit(s) of mantissa is (are) (rR:11) [(rR:22)].

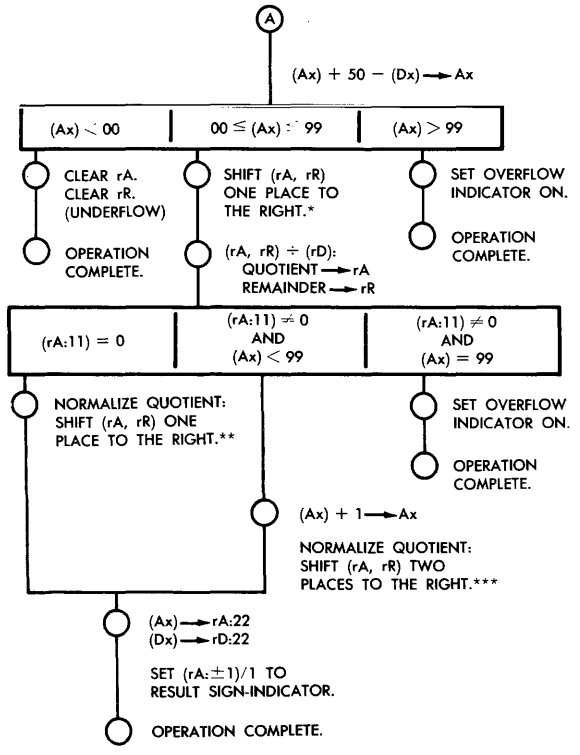
Remainder: The low-order digit positions of the R register.

Flow Chart. See page 2-25.

Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. "Spurious exponent underflow" can occur in case exponent arithmetic—which takes place before the mantissas are divided—yields the equivalent of −01 for the adjusted difference of the exponents of (rA)





*Exponent adjustment is made at the end of the operation if required.
**Then (rR:11) is the low-order digit of the quotient.
***Then (rR:22) are the two low-order digits of the quotient.

and (B[aaaa]): if the adjusted difference of these exponents is −01, the A and R registers are cleared, and the Execute Phase is terminated, even though normalization of the quotient might yield a valid exponent, namely, 00.

For example, suppose (rA) = 0 00 1000 0000 and (aaaa) = 0 51 1000 0000; the instruction is 0 0000 FDV aaaa. The execution of this instruction should yield (rA) = 0 00 1000 0000 (since the divisor is unity). Instead, the A and R registers are cleared.

Remarks.

1. Exponent overflow can occur, in which case the OVERFLOW Indicator is turned "on."
2. Exponent underflow causes both the A and R registers to be cleared, but is not otherwise indicated.
3. In case the mantissa of the dividend is not normalized—in which case (rA:31) = 0—but the mantissa of the divisor is normalized—in which case (rD:31) ≠ 0—it is assumed that the quotient will be zero. The operation is terminated as soon as this condition is determined to exist—85 μs are required—with the A and R registers cleared.
4. In case the divisor is not normalized—in which case (rD:31) = 0—it is assumed that the divisor is zero: the operation is terminated as soon as this condition is determined to exist—85 μs are required—with the OVERFLOW Indicator turned "on."

Register Status.

Register name	Contents after execution of FDV
A	Quotient
R	"Remainder"
D	(rD:23) = 000; (rD:08) = (B[aaaa]:08)
B	Unchanged
P	$(rP)_b + 1$
C	i i i i | 2 5 | B[aaaa]
E	B[aaaa]

Register name	Contents after Storage ALARM STOP
A	Unchanged
R	Unchanged
D	± | i i i i | 2 5 | a a a a

Register name	Contents after Storage ALARM STOP
B	Unchanged
P	$(rP)_b + 1$
C	i i i i | 2 5 | B[aaaa]
E	B[aaaa]

Register name	Contents if underflow occurs after exponent arithmetic
A	Cleared
R	Unchanged
D	(rD:23) = 000 (rD:08) = (B[aaaa]: 08)
B	Unchanged
P	$(rP)_b + 1$
C	i i i i | 2 5 | B[aaaa]
E	B[aaaa]

Register name	Contents if dividend = 0, divisor $\neq$ 0
A	Cleared
R	Cleared
D	B[aaaa]
B	Unchanged
P	$(rP)_b + 1$
C	i i i i | 2 5 | B[aaaa]
E	B[aaaa]

Register name	Contents if divisor = 0
A	(rA: $\pm$ 1) = 0 $(rA:00)_a = (rA:00)_b$
R	Unchanged
D	B[aaaa]
B	Unchanged
P	$(rP)_b + 1$
C	i i i i | 2 5 | B[aaaa]
E	B[aaaa]

INCREASE FIELD LOCATION (IFL)

Operation Code. 26

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	s	L	n	n	O	p	a	a	a	a

Definitions.

s: partial-word designator:
s designates the position, within the word, of the low-order digit of the partial-word operand.

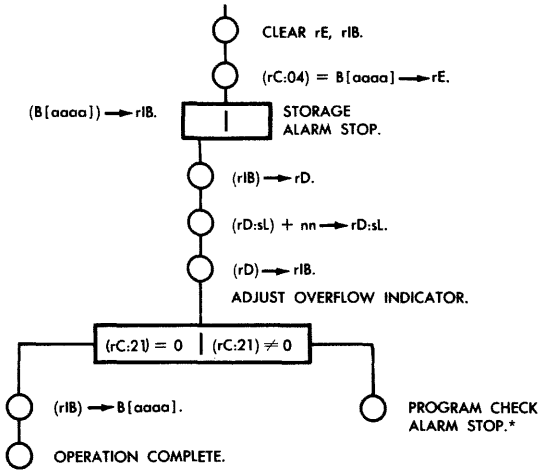
L: partial-word designator:
L specifies the number of digits in the partial-word operand.

nn: modifier for the partial-word operand.

aaaa: address of base of location of partial-word addend.

Description of Operation. Increase (B[aaaa]: sL) by nn. If overflow occurs—that is, if the result exceeds the capacity of the specified partial-word field—set the OVERFLOW Indicator to "on."

Flow Chart.



*As each digit of the partial-word sum is generated, L [i.e., (rC:21)] is counted down. If, at the start, L > s + 1, s ≠ 0, then at the end of the addition (rC:21) will be different from zero, and field overflow will be detected. See Example 6, page 2-27.

Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. Program Check ALARM STOP due to field overflow.

Remarks. If the sign-digit position of (B[aaaa]) is included in the partial-word field specified by sL, it does not have sign significance: the sign-digit position has numeric significance. See Examples 4, 5, and 6, page 2-27.

Register Status.

Register name	Contents after execution of IFL
A	Unchanged
R	Unchanged
D	$(B[aaaa])_a$
B	Unchanged
P	$(rP)_b + 1$
C	0 0 n n 2 6 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± s L n n 2 6 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	s L n n 2 6 B[aaaa]
E	B[aaaa]

Register name	Contents if Program Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	$(B[aaaa])_b$, as modified
B	Unchanged
P	$(rP)_b + 1$
C	0 * n n 2 6 B[aaaa]
E	(B[aaaa])

* L − s − 1

Examples.

1. (rC) = 0202 26 0396
   $(0396)_b$ = + 0000 00 0012
   $(rD)_a = (0396)_a$ = + 0000 00 0014
2. (rC) = 6314 26 0416
   $(0416)_b$ = − 2973 43 9216
   $(rD)_a = (0416)_a$ = − 2973 57 9216
3. (rC) = 6220 26 0534
   $(0534)_b$ = + 0002 90 2400
   $(rD)_a = (0534)_a$ = + 0002 10 2400

   OVERFLOW Indicator is set "on."
4. (rC) = 3412 26 0600
   $(0600)_b$ = 1 2490 00 9000
   $(rD)_a = (0600)_a$ = 1 2610 00 9000
5. (rC) = 1232 26 0657
   $(0657)_b$ = 9 5236 47 8888
   $(rD)_a = (0657)_a$ = 2 7236 47 8888

   OVERFLOW Indicator is set "on."
6. (rC) = 2530 26 0900
   $(0900)_b$ = 0 8429 90 5432
   $(rD)_a$ = 1 1429 90 5432

   Program Check ALARM STOP.
   $(rC)_a$ = 0230 26 0900
7. (rC) = 0104 26 0963
   $(0963)_b$ = − 1123 87 0003
   $(rD)_a = (0963)_a$ = − 1123 87 0007
8. (rC) = 6122 26 2440
   $(2440)_b$ = + 0000 81 0000
   $(rD)_a = (2440)_a$ = + 0000 83 0000
9. (rC) = 4115 26 4000
   $(4000)_b$ = − 1125 99 9999
   $(rD)_a = (4000)_a$ = − 1120 99 9999

   OVERFLOW Indicator is set "on."
10. (rC) = 0400 26 3000
    $(3000)_b$ = − 1233 00 2914
    $(rD)_a = (3000)_a$ = − 1233 00 2914

DECREASE FIELD LOCATION (DFL)

Operation Code. 27

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	s	L	n	n	O	p	a	a	a	a

Definitions.

s: partial-word designator:
s designates the position, within the word, of the low-order digit of the partial-word operand.

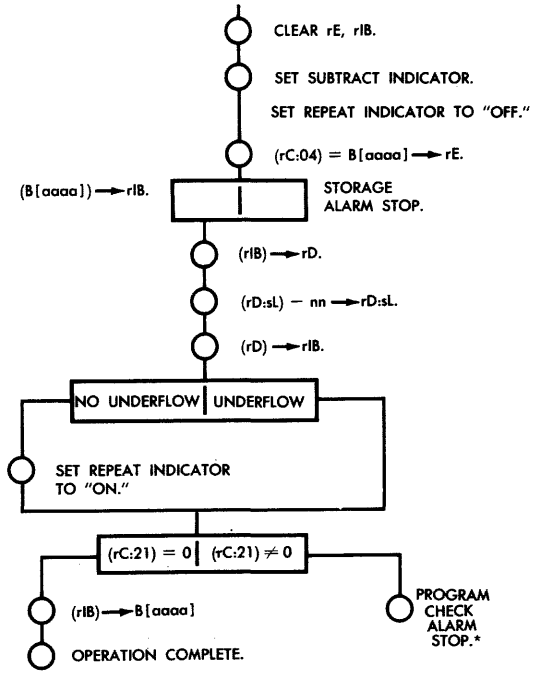
L: partial-word designator:
L specifies the number of digits in the partial-word operand.

nn: modifier for the partial-word operand.

aaaa: address of base of location of partial-word minuend.

Description of Operation. Decrease (B[aaaa]: sL) by nn. If overflow occurs—that is, if $(B[aaaa]: sL)_a > (B[aaaa]: sL)_b$—set the REPEAT Indicator to "off"; if not, set the REPEAT Indicator to "on."

Flow Chart.



*As each digit of the partial-word difference is generated, L [i.e., (rC:21)] is counted down. If, at the start, $L > s + 1$, $s \neq 0$, then at the end of the subtraction (rC:21) will be different from zero, and field overflow will be detected. See Example 6, page 2-29.

Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. Program Check ALARM STOP due to field overflow.

Remarks.

1. If the sign-digit position of (B[aaaa]) is included in the partial-word field specified by sL, it does not have sign significance: the sign-digit position has numeric significance. See Examples 5 and 6, page 2-29.
2. It is not necessary to follow this instruction immediately with a BRANCH, REPEAT instruction. See page 2-32.

Register Status.

Register name	Contents after execution of DFL
A	Unchanged
R	Unchanged
D	$(B[aaaa])_a$
B	Unchanged
P	$(rP)_b + 1$
C	0 0 n n 2 7 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± s L n n 2 7 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	s L n n 2 7 B[aaaa]
E	B[aaaa]

Register name	Contents if Program Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	$(B[aaaa])_b$, as modified
B	Unchanged
P	$(rP)_b + 1$
C	0 * n n 2 7 B[aaaa]
E	B[aaaa]

* L − s − 1

Examples.

1. (rC) = 0202 27 0396
   $(0396)_b$ = + 0000 00 0012
   $(rD)_a = (0396)_a$ = + 0000 00 0010
   REPEAT Indicator is set "on."
2. (rC) = 6314 27 0416
   $(0416)_b$ = − 2973 43 9216
   $(rD)_a = (0416)_a$ = − 2973 29 9216
   REPEAT Indicator is set "on."
3. (rC) = 6220 27 0534
   $(0534)_b$ = + 0002 10 2400
   $(rD)_a = (0534)_a$ = + 0002 90 2400
   REPEAT Indicator is set "off."
4. (rC) = 3412 27 0600
   $(0600)_b$ = 1 2490 00 9000
   $(rD)_a = (0600)_a$ = 1 2370 00 9000
   REPEAT Indicator is set "on."

5. $(rC) = 1232\ 27\ 0657$
   $(0657)_b = 0\ 5236\ 47\ 8888$
   $(rD)_a = (0657)_a = 7\ 3236\ 47\ 8888$

   REPEAT Indicator is set "off."

6. $(rC) = 2530\ 27\ 0900$
   $(0900)_b = 0\ 8429\ 90\ 5432$
   $(rD)_a = (0900)_a = 0\ 5429\ 90\ 5432$

   REPEAT Indicator is set "on."
   Program Check ALARM STOP.
   $(rC)_a = 0230\ 27\ 0900$

7. $(rC) = 0400\ 27\ 0900$
   $(0900)_b = +\ 2345\ 67\ 1212$
   $(rD)_a = (0900)_a = +\ 2345\ 67\ 1212$

   REPEAT Indicator is set "on."

DECREASE FIELD LOCATION, LOAD B (DLB)

Operation Code. 28

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	s	L	n	n	O	p	a	a	a	a

Definitions.

s: partial-word designator:
s designates the location, within the word, of the low-order digit of the partial-word operand.

L: partial-word designator:
L specifies the number of digits in the partial-word operand.

nn: modifier for the partial-word operand.

aaaa: address of base of location of partial-word operand.

Description of Operation. Execute DFL, and, in addition, load rB with modified partial-word field. That is, decrease (B[aaaa]:sL) by nn. If underflow occurs, set the REPEAT Indicator to "off;" if not, set the REPEAT Indicator to "on." Then load rB with the modified partial-word field.

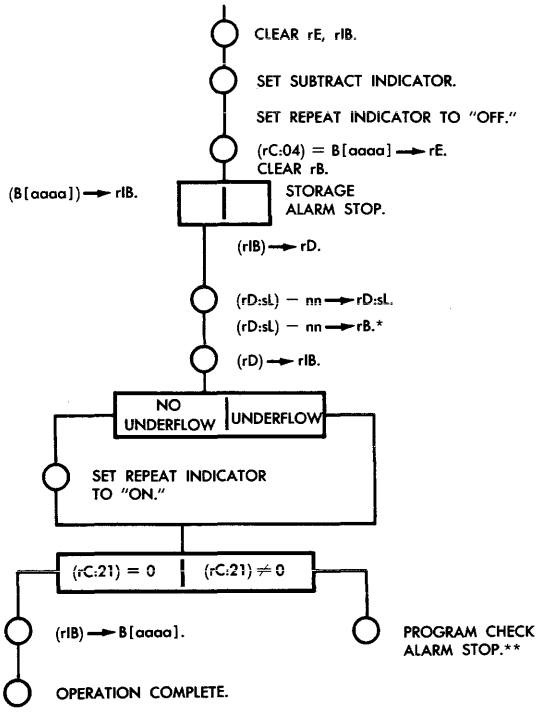
Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. Program Check ALARM STOP due to field overflow.

Remarks.

1. If the sign-digit position of (B[aaaa]) is included in the partial-word field specified by sL, it does not have sign significance: the sign-digit position has numeric significance. See Examples 5 and 6, page 2-30.
2. The examples for DFL will suffice to explain a large part of DLB. The reader is referred to page 2-28. Light is shed on the mechanism of the "load B" part of the operation by the examples on page 2-30.

Flow Chart.



*If L < 4, the L high-order digits of (rD:sL) − nn replaces the L high-order digits of rB; if L < 4, the four high-order digits of (rD:sL) − nn replace all of rB.

**As each digit of the partial-word difference is generated, L [i.e., (rC:21)] is counted down. If, at the start, L > s + 1, s ≠ 0, then at the end of the subtraction (rC:21) will be different from zero, and field overflow will be detected. See Examples 5 and 6, page 2−30.

Register Status.

Register name	Contents after execution of DLB
A	Unchanged
R	Unchanged
D	$(B[aaaa])_a$
B	See description of operation
P	$(rP)_b + 1$
C	0 0 | n n | 2 8 | B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged

Register name	Contents if Storage ALARM STOP occurs
D	± s L n n 2 8 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	s L n n 2 8 B[aaaa]
E	B[aaaa]

Register name	Contents if Program Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	$(B[aaaa])_b$, as modified
B	See description of operation
P	$(rP)_b + 1$
C	0 * n n 2 8 B[aaaa]
E	B[aaaa]

* L − s − 1

Examples.

NOTE: See also Examples, page 2-28, which elucidate DFL.

1. (rC) = 0402 28 0496
   $(0496)_b$ = + 1223 49 0148
   $(rD)_a = (0496)_a$ = + 1223 49 0146
   $(rB)_a$ = 0146
   REPEAT Indicator is set "on."
2. (rC) = 8205 28 0516
   $(0516)_b$ = − 3946 25 2014
   $(rD)_a = (0516)_a$ = − 3946 25 1514
   $(rB)_a$ = 1500
   REPEAT Indicator is set "on."
3. (rC) = 3310 28 0634
   $(0634)_b$ = + 0050 40 2222
   $(rD)_a = (0634)_a$ = + 9950 40 2222
   $(rB)_a$ = 9950
   REPEAT Indicator is set "off."
4. (rC) = 6650 28 0700
   $(0700)_b$ = − 1255 00 9753
   $(rD)_a = (0700)_a$ = − 1254 50 9753
   $(rB)_a$ = 1254
   REPEAT Indicator is set "on."
5. (rC) = 2420 28 0790
   $(0790)_b$ = 1 3540 44 2345
   $(rD)_a$ = 1 1540 44 2345
   $(rB)_a$ = 1 150
   REPEAT Indicator is set "on."
   Program Check ALARM STOP.
6. (rC) = 1310 28 2040
   $(2040)_b$ = 0 5998 74 0000
   $(rD)_a$ = 9 5998 74 0000
   $(rB)_a$ = 9 500
   REPEAT Indicator is set "off."
   Program Check ALARM STOP.

RECORD TRANSFER (RTF)

Operation Code. 29

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	i	n	n	i	O	p	a	a	a	a

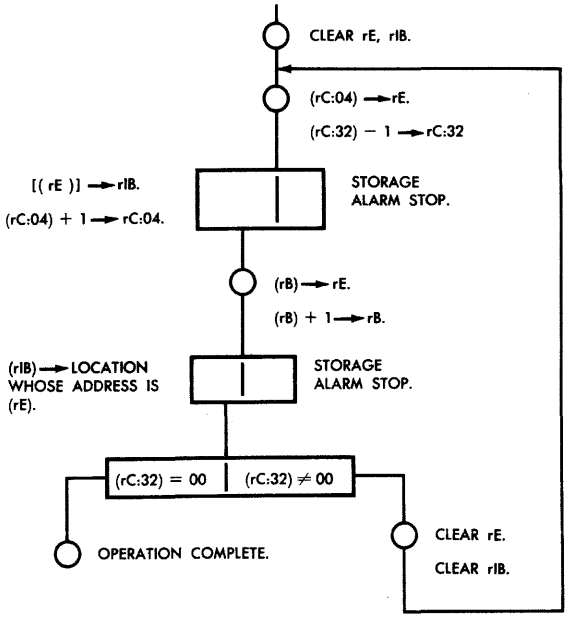
Definitions.

nn: specifies the number of words to be relocated: if nn = 00, 100 words will be relocated.

aaaa: address of base of location of first word to be relocated.

Description of Operation. Relocate the contents of nn consecutively addressed locations, beginning with the one whose address is B[aaaa]. Transfer the specified words, in succession, and one at a time, to the nn consecutively addressed locations, beginning with the one whose address is in the B register.

Flow Chart.



Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Remarks. After the execution of a RECORD TRANSFER operation, the B register will contain the sum of the address of the last location filled plus 1, that is, the address of "the next location to be filled."

Register Status.

Register name	Contents after execution of RTF
A	Unchanged
R	Unchanged
D	± i n n i 2 9 a a a a a
B	$(rB)_b + nn^*$
P	$(rP)_b + 1$
C	i 0 0 i 2 9 B[aaaa] + nn**
E	$(rB)_b + nn - 1^*$

Note: if nn = 00,
* $(rB)_b + nn = (rB)_b + 100$; $(rB)_b + nn - 1 = (rB)_b + 99$
**B[aaaa] + nn = B[aaaa] + 100

Register name	Contents after Storage ALARM STOP
A	Unchanged
R	Unchanged
D	± i n n i 2 9 a a a a
B	Indeterminate***
P	$(rP)_b + 1$
C	Indeterminate***
E	Indeterminate***

*** See description of operation

BRANCH, UNCONDITIONALLY (BUN)

Operation Code. 30

Instruction Format.

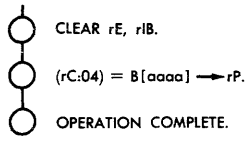
±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	a	a	a	a

Definitions.

aaaa: address of base of location of next instruction.

Description of Operation. Transfer control to location B[aaaa], i.e., prepare to take the next instruction from B[aaaa].

Flow Chart.



Exceptional Conditions. None.

Register Status.

Register name	Contents after execution of BUN
A	Unchanged
R	Unchanged
D	± i i i i 3 0 a a a a
B	Unchanged
P	B[aaaa]
C	i i i i 3 0 B[aaaa]
E	Cleared

BRANCH, OVERFLOW (BOF)

Operation Code. 31

Instruction Format.

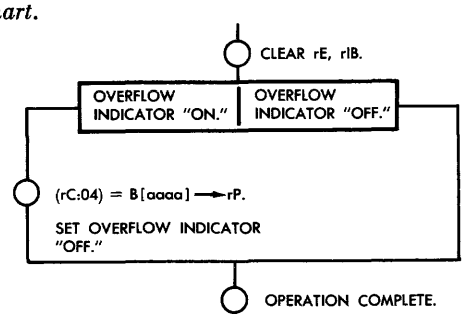
±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	a	a	a	a

Definitions.

aaaa: address of base of location of alternate instruction.

Description of Operation. If the OVERFLOW Indicator is "on," transfer control to location B[aaaa], i.e., prepare to take the next instruction from B[aaaa]. If the OVERFLOW Indicator is "off," control continues in sequence. (See page 2-37 for the use of the BOF instruction in conjunction with the SOR, SOH, and IOM instructions.)

Flow Chart.



Exceptional Conditions. None.

Remarks. The OVERFLOW Indicator may be set "on" by the following operations:

Code	*Name*
08	KEYBOARD ADD
12	ADD
12	ADD ABSOLUTE
13	SUBTRACT
13	SUBTRACT ABSOLUTE
15	DIVIDE
16	ROUND
19	ADD TO LOCATION
22	FLOATING ADD
22	FLOATING ADD ABSOLUTE
23	FLOATING SUBTRACT
23	FLOATING SUBTRACT ABSOLUTE
24	FLOATING MULTIPLY
25	FLOATING DIVIDE
26	INCREASE FIELD LOCATION

Register Status.

Register name	Contents after execution of BOF
A	Unchanged
R	Unchanged
D	i i i i 3 1 a a a a
B	Unchanged
P	B[aaaa], if OVERFLOW Indicator "on"; $(rP)_b + 1$, if OVERFLOW Indicator "Off"
C	i i i i 3 1 B[aaaa]
E	Cleared

BRANCH, REPEAT (BRP)

Operation Code. 32

Instruction Format.

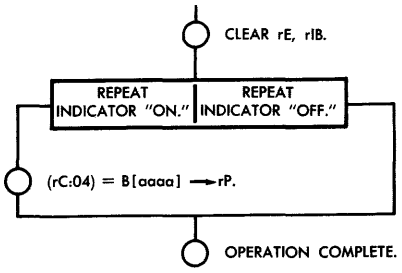
±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	a	a	a	a

Definitions.

aaaa: address of base of location of alternate instruction.

Description of Operation. If the REPEAT Indicator is "on," transfer control to location B[aaaa], i.e., prepare to take the next instruction from B[aaaa]. If the REPEAT Indicator is "off," control continues in sequence.

Flow Chart.



Exceptional Conditions. None.

Remarks.

1. The state of the REPEAT Indicator is not disturbed by the execution of a BRANCH, REPEAT instruction.
2. The REPEAT Indicator may be set "on" by the following operations:

Code	*Name*
27	DECREASE FIELD LOCATION
28	DECREASE FIELD LOCATION, LOAD B

Register Status.

Register name	Contents after execution of BRP
A	Unchanged
R	Unchanged
D	± i i i i 3 2 a a a a
B	Unchanged
P	B[aaaa], if REPEAT Indicator "on" $(rP)_b + 1$, if REPEAT Indicator "off"
C	i i i i n 3 2 B[aaaa]
E	Cleared

BRANCH, SIGN A (BSA)

Operation Code. 33

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	n	O	p	a	a	a	a

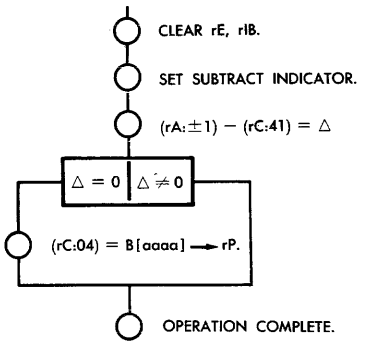
Definitions.

n: comparison digit.

aaaa: address of base of location of alternate instruction.

Description of Operation. If (rA:±1) = n, transfer control to location B[aaaa], i.e., prepare to take the next instruction from B[aaaa]. If (rA:±1) ≠ n, control continues in sequence.

Flow Chart.



Exceptional Conditions. None.

Register Status.

Register name	Contents after execution of BSA
A	Unchanged
R	Unchanged
D	± i i i n 3 3 a a a a
B	Unchanged
P	B[aaaa], if (rA:±1) = n $(rP)_b + 1$, if (rA:±1) ≠ n
C	i i i n 3 3 B[aaaa]
E	Cleared

BRANCH, COMPARISON HIGH (BCH)

BRANCH, COMPARISON LOW (BCL)

Operation Code. 34

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	v	O	p	a	a	a	a

Definitions.

v: variation designator:
- v = 0: BRANCH, COMPARISON HIGH will be executed.
- v = 1: BRANCH, COMPARISON LOW will be executed.

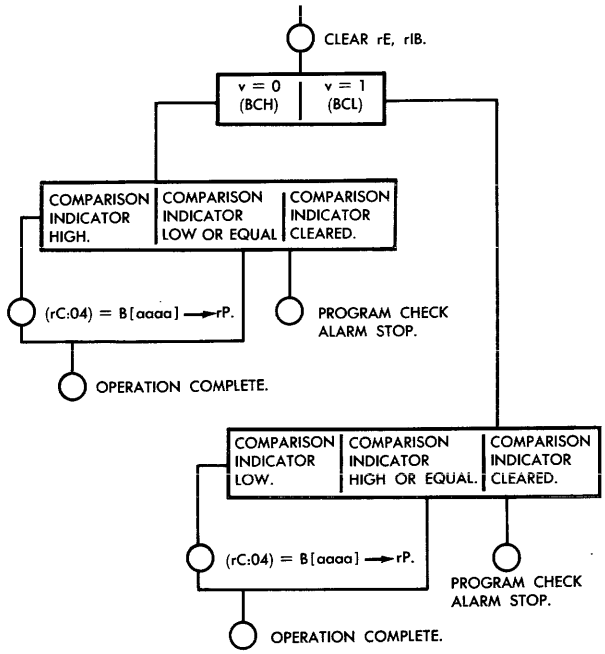
aaaa: address of base of location of alternate instruction.

Description of Operation.

v = 0: BRANCH, COMPARISON HIGH will be executed. If the COMPARISON Indicator is HIGH, transfer control to location B[aaaa], i.e., prepare to take the next instruction from B[aaaa]. If the COMPARISON Indicator is LOW or EQUAL, control continues in sequence.

v = 1: BRANCH, COMPARISON LOW will be executed. If the COMPARISON Indicator is LOW, transfer control to location B[aaaa]. If the COMPARISON Indicator is HIGH or EQUAL, control continues in sequence.

Flow Chart.



Exceptional Conditions. Program Check ALARM STOP caused by COMPARISON Indicator being "off" when interrogated.

Remarks.

1. The BRANCH, COMPARISON HIGH variation will be executed if v ≠ 1.
2. The state of the COMPARISON Indicator is not disturbed by the execution of these instructions.
3. The COMPARISON Indicator is set by the following operations:

Code	*Name*
18	COMPARE FIELD A
18	COMPARE FIELD R

Register Status.

Register name	Contents after execution if branching occurs
A	Unchanged
R	Unchanged
D	± i i i v 3 4 a a a a

Register name	Contents after execution if branching occurs
B	Unchanged
P	B[aaaa]
C	i i i v 3 4 B[aaaa]
E	Cleared

Register name	Contents after execution if branching does not occur
A	Unchanged
R	Unchanged
D	± i i i v 3 4 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i v 3 4 B[aaaa]
E	Cleared

Register name	Contents if Program Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± i i i v 3 4 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i v 3 4 B[aaaa]
E	Cleared

BRANCH, COMPARISON EQUAL (BCE)

BRANCH, COMPARISON UNEQUAL (BCU)

Operation Code. 35

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	v	O	p	a	a	a	a

Definitions.

v: variation designator:

v = 0: BRANCH, COMPARISON EQUAL will be executed.

v = 1: BRANCH, COMPARISON UNEQUAL will be executed.

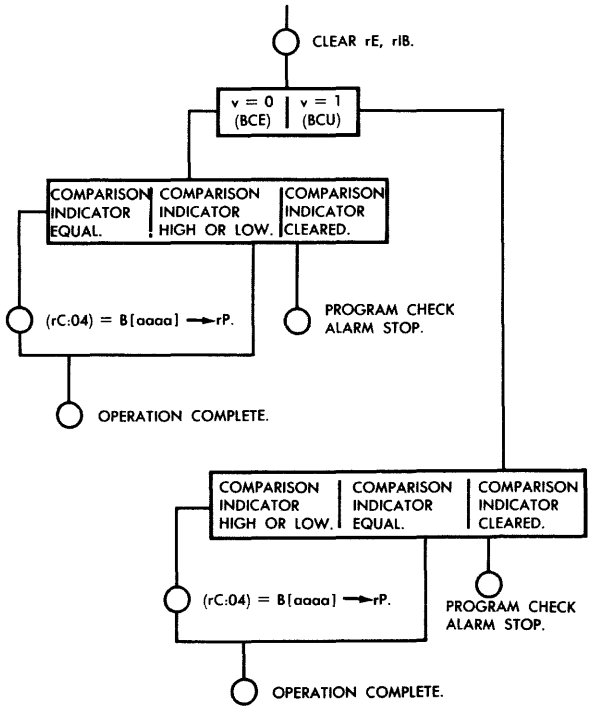
aaaa: address of base of location of alternate instruction.

Description of Operation.

v = 0: BRANCH, COMPARISON EQUAL will be executed. If the COMPARISON Indicator is EQUAL, transfer control to location B[aaaa], i.e., prepare to take the next instruction from B[aaaa]. If the COMPARISON Indicator is HIGH or LOW, control continues in sequence.

v = 1: BRANCH, COMPARISON UNEQUAL will be executed. If the COMPARISON Indicator is HIGH or LOW, transfer control to location B[aaaa]. If the COMPARISON Indicator is EQUAL, control continues in sequence.

Flow Chart.



Exceptional Conditions. Program Check ALARM STOP caused by COMPARISON Indicator being "off" when interrogated.

Remarks.

1. The BRANCH, COMPARISON EQUAL variation will be executed if v ≠ 1.
2. The state of the COMPARISON Indicator is not disturbed by the execution of these instructions.
3. The COMPARISON Indicator is set by the following operations:

Code	*Name*
18	COMPARE FIELD A
18	COMPARE FIELD R

Register Status.

Register name	Contents after execution if branching occurs
A	Unchanged
R	Unchanged
D	± i i i v 3 5 a a a a
B	Unchanged
P	B[aaaa]
C	i i i v 3 5 B[aaaa]
E	Cleared

Register name	Contents after execution if branching does not occur
A	Unchanged
R	Unchanged
D	± i i i v 3 5 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i v 3 5 B[aaaa]
E	Cleared

Register name	Contents if Program Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± i i i v 3 5 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i v 3 5 B[aaaa]
E	Cleared

BRANCH, FIELD A (BFA)

Operation Code. 36

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	s	L	n	n	O	p	a	a	a	a

Definitions.

s: partial-word designator:
s designates the position, within the word, of the low-order digit of the partial-word operand.

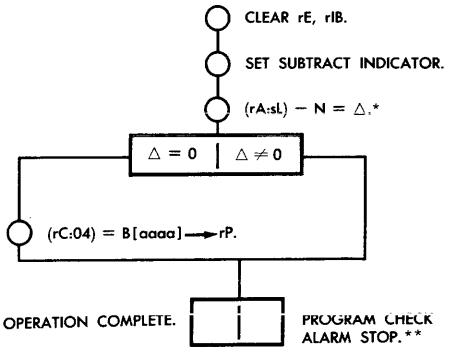
L: partial-word designator:
L specifies the number of digits in the partial-word operand.

nn: basis for comparator.

aaaa: address of base of location of alternate instruction.

Description of Operation. Beginning with the low-order digit of (rA:sL), successively higher-order digits are compared alternately with the low-order and high-order digit of nn. If equality exists for every digit position compared, transfer control to location B[aaaa]) i.e., take the next instruction from B[aaaa]. If inequality exists for any digit position, control continues in sequence.

Flow Chart.



*The subtrahend, N, is an L-digit number, constructed with the two-digit number, nn, as a basis: the low-order digit of N is the low-order digit of nn; the next-higher-order digit of N is the high-order digit of nn; the next-higher-order digit of N is the low-order digit of nn; and so forth, successively higher-order digits of N being, alternately the high-and low-order digits of nn.

**As each digit of N is generated, L [i.e., (rC:21)] is counted down. If, at the start, $L > s + 1$, $s \neq 0$, then at the end of the comparison (rC:21) is different from zero, and field overflow is detected.

Exceptional Conditions. Program Check ALARM STOP due to field overflow.

Remarks. Except for the difference in registers, BRANCH, FIELD A and BRANCH, FIELD R are identical in operation. See page 2-36.

Register Status.

Register name	Contents after execution of BFA
A	Unchanged
R	Unchanged
D	± s L n n 3 6 a a a a
B	Unchanged

Register name	Contents after execution of BFA
P	B[aaaa], if branch $(rP)_b + 1$, if no branch
C	0 0 n n 3 6 B[aaaa]
E	Cleared

Register name	Contents if Program Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± s L n n 3 6 a a a a
B	Unchanged
P	B[aaaa], if branch would have occurred $(rP)_b + 1$, if no branch would have occurred
C	0 0 * n n 3 6 B[aaaa]
E	Cleared

* L − s − 1

BRANCH, FIELD R (BFR)

Operation Code. 37

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	s	L	n	n	O	p	a	a	a	a

Definitions.

s: partial-word designator:
s designates the position, within the word, of the low-order digit of the partial-word operand.

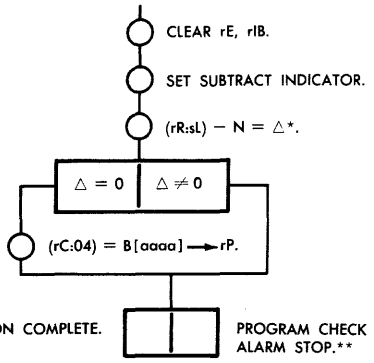
L: partial-word designator:
L specifies the number of digits in the partial-word operand.

nn: basis for comparator.

aaaa: address of base of location of alternate instruction.

Description of Operation. Beginning with the low-order digit of (rR:sL), successively higher-order digits are compared alternately with the low-order and high-order digit of nn. If equality exists for every digit position compared, transfer control to B[aaaa], i.e., take the next instruction from B[aaaa]. If inequality exists for any digit position, control continues in sequence.

Flow Chart.



*The subtrahend, N, is an L-digit number, constructed with the two-digit number, nn, as a basis: the low-order digit of N is the low-order digit of nn; the next-higher-order digit of N is the high-order digit of nn; the next-higher-order digit of N is the low-order digit of nn; and so forth, successively higher-order digits of N being, alternately the high-and low-order digits of nn.

**As each digit of N is generated, L [i.e., (rC:21)] is counted down. If, at the start, L > s + 1, s ≠ 0, then at the end of the comparison (rC:21) is different from zero, and field overflow is detected.

Exceptional Conditions. Program Check ALARM STOP due to field overflow.

Remarks. Except for the difference in registers, BRANCH, FIELD R and BRANCH, FIELD A are identical in operation. See page 2-35.

Register Status.

Register name	Contents after execution of BFR
A	Unchanged
R	Unchanged
D	± s L n n 3 7 a a a a
B	Unchanged
P	B[aaaa], if branch $(rP)_b + 1$, if no branch
C	0 0 n n 3 7 B[aaaa]
E	Cleared

Register name	Contents if Program Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± s L n n 3 7 a a a a
B	Unchanged
P	B[aaaa], if branch would have occurred $(rP)_b + 1$, if no branch would have occurred

Register name	Contents if Program Check ALARM STOP occurs
C	0 * n n 3 7 B[aaaa]
E	Cleared

* L − s − 1

SET OVERFLOW REMEMBER (SOR)

SET OVERFLOW HALT (SOH)

INTERROGATE OVERFLOW MODE (IOM)

Operation Code. 39

Instruction Format.

± 1	2	3	4	5	6	7	8	9	0
±	i	i	i	v	O	p	a	a	a	a

Definitions.

v: variation designator:
- v = 0: SET OVERFLOW REMEMBER will be executed.
- v = 1: SET OVERFLOW HALT will be executed.
- v = 2: INTERROGATE OVERFLOW MODE will be executed.

aaaa: address of base of location of alternate instruction.

Description of Operation.

v = 0: Sets the overflow circuitry such that a succeeding overflow condition turns the OVERFLOW Indicator "on," but *does not* cause the Data Processor to halt even if the instruction causing the overflow is not followed by a BOF instruction.

v = 1: Sets the overflow circuitry such that a succeeding overflow condition turns the OVERFLOW Indicator "on" and *does* cause the Data Processor to halt if the instruction causing the overflow is not followed by a BOF instruction.

v = 2: If the Data Processor is in the Overflow Remember Mode (set by a previous SOR instruction or the CLEAR switch), control continues in sequence.

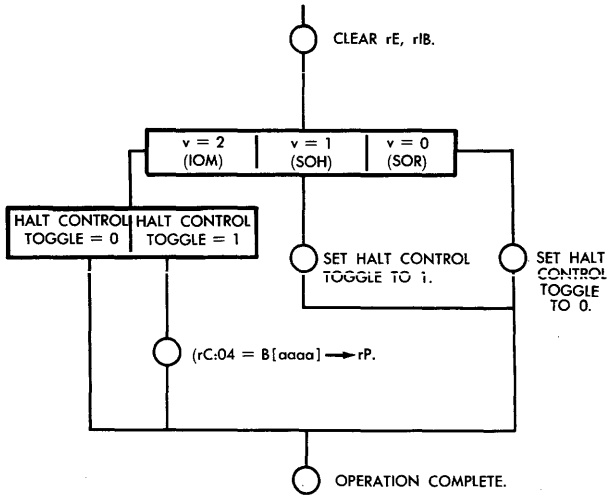
If the Data Processor is in the Overflow Halt Mode (set by a previous SOH instruction), control is transferred to B[aaaa].

Exceptional Conditions. None.

Remarks.

1. Depressing the CLEAR switch sets the Data Processor to the Overflow Remember Mode.
2. These instructions *will not* turn "off" the OVERFLOW Indicator.
3. An overflow condition followed by a SOR instruction *will* cause the Data Processor to halt.
4. aaaa is relevant only for v = 2.
5. If the Data Processor is in the Overflow Halt Mode (set by a previous SOH instruction), a BOF instruction immediately following an OVERFLOW condition will cause control to be transferred to the location specified by B[aaaa] of the BOF instruction.
6. If the Data Processor is in the Overflow Remember Mode (set by a previous SOR instruction or by depressing the CLEAR switch) a BOF instruction following an OVERFLOW condition will cause the same effect as 5, above, except that the BOF does not have to follow immediately after the operation that caused OVERFLOW. Control continues in sequence until the BOF instruction is encountered.

Flow Chart.



Register Status.

Register name	Contents after execution if branching occurs
A	Unchanged
R	Unchanged
D	± i i i v 3 9 a a a a
B	Unchanged
P	B[aaaa]
C	i i i v 3 9 B[aaaa]
E	Cleared

Register name	Contents after execution if branching does not occur
A	Unchanged
R	Unchanged
D	± i i i v 3 9 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i v 3 9 B[aaaa]
E	Cleared

STORE A (STA)

STORE R (STR)

STORE B (STB)

Operation Code. 40

Instruction Format.

± 1 2 3 4 5 6 7 8 9 0
± s L f v O p a a a a

Definitions.

s: partial-word designator:
f = 0: s is not relevant.
f = 1: s designates the position, within the word, of the low-order digit of each partial-word operand.

L: partial-word designator:
f = 0: L is not relevant.
f = 1: L specifies the number of digits in each partial-word operand.

f: partial-word designator:
f = 0: entire words will be used as operands.
f = 1: the contents of the partial-word fields defined by sL will be used as operands.

v: variation designator:
v = 0: STORE A will be executed.
v = 1: STORE R will be executed.
v = 2: STORE B will be executed.

aaaa: address of base of location in which the selected field will be stored.

Description of Operation. Store the specified field of the designated register in the corresponding field location in B[aaaa].

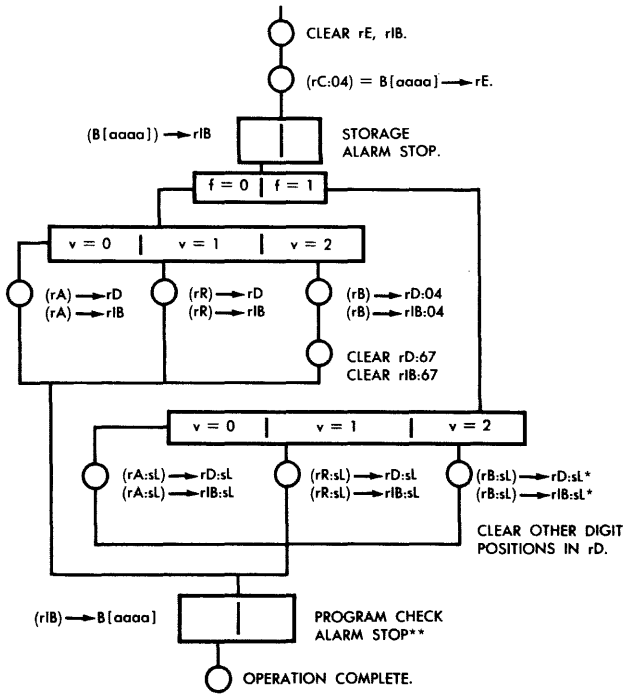
Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. Program Check ALARM STOP due to field overflow.

Remarks.

1. The STORE A variation will be executed if v ≠ 1 or 2.
2. f = 2, 4, 6, or 8 has the same effect as f = 0; f = 3, 5, 7, or 9 has the same effect as f = 1.
3. If, for example, the STORE A variation is being executed with f = 1 (partial-word selection), then (rA:sL) → (B[aaaa]:sL). The remainder of (B[aaaa]) is unaltered.

Flow Chart.



* The B register is regarded as if it were 11 digits long as indicated:

± 1 2 3 4 5 6 7 8 9 0	
0 0 0 0 0 0 0	r B

**As each digit is transferred from the designated register to the D register, L [i.e., (rC:21)] is counted down. If, at the start, L > s + 1, s ≠ 0, then at the end of the transfer (rC:21) will be different from zero and field overflow will be detected.

Register Status.

Register name	Contents after execution of all instructions
A	Unchanged
R	Unchanged
D	$(B[aaaa])_a$
B	Unchanged
P	$(rP)_b + 1$
C	0 0 f v 4 0 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± s L f v 4 0 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	s L f v 4 0 B[aaaa]
E	B[aaaa]

Register name	Contents if Program Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	See flow chart
B	Unchanged
P	$(rP)_b + 1$
C	0 * f v 4 0 B[aaaa]
E	B[aaaa]

* L − s − 1

Examples.

1. (rC) = 0412 40 1000
   $(1000)_b$ = 0 1234 56 7890
   (rB) = 1234
   $(rD)_a$ = 0 0000 00 1234
   $(1000)_a$ = 0 1234 56 1234

2. (rC) = 0002 40 1000
   $(1000)_b$ = 0 1234 56 7890
   (rB) = 1234
   $(rD)_a$ = 0 0000 00 1234
   $(1000)_a$ = 0 0000 00 1234

3. (rC) = 8412 40 2000
   $(2000)_b$ = 0 1234 56 7890
   (rB) = 2345
   $(rD)_a$ = 0 0000 00 2300
   $(2000)_a$ = 0 1234 00 2390

4. (rC) = 5312 40 3000
   $(3000)_b$ = 0 1234 56 7890
   (rB) = 1357
   $(rD)_a$ = 0 0000 00 0000
   $(3000)_a$ = 0 1200 06 7890

LOAD R (LDR)

Operation Code. 41

Instruction Format.

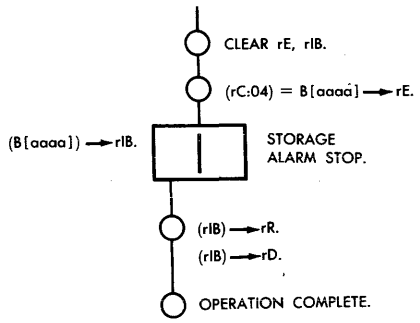
±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	a	a	a	a

Definitions.

aaaa: address of base of location of operand.

Description of Operation. Replace (rR) by (B[aaaa]).

Flow Chart.



Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Register Status.

Register name	Contents after execution of LDR
A	Unchanged
R	(B[aaaa])
D	(B[aaaa])
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 4 1 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± i i i i 4 1 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 4 1 B[aaaa]
E	B[aaaa]

LOAD B (LDB)

LOAD B, COMPLEMENT (LBC)

Operation Code. 42

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	v	O	p	a	a	a	a

Definitions.

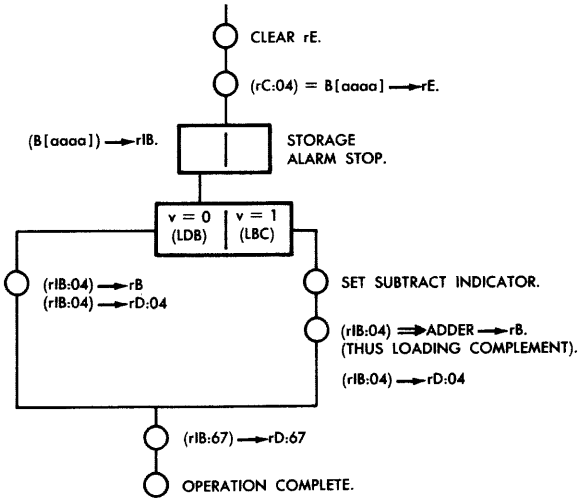
v: variation designator:
- v = 0: LOAD B will be executed
- v = 1: LOAD B, COMPLEMENT will be executed.

aaaa: address of base of location of operand.

Description of Operation.

v = 0: (B[aaaa]:04) replaces (rB).
v = 1: the 10's complement of (B[aaaa]) replaces (rB).

Flow Chart.



Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Remarks. The LOAD B variation will be executed if v ≠ 1.

Register Status.

Register name	Contents after execution of LDB
A	Unchanged
R	Unchanged
D	(B[aaaa])
B	(B[aaaa]:04)
P	$(rP)_b + 1$
C	i i i | 0 | 4 2 | B[aaaa]
E	B[aaaa]

Register name	Contents after execution of LBC
A	Unchanged
R	Unchanged
D	(B[aaaa])
B	10's complement of (B[aaaa]:04)
P	$(rP)_b + 1$
C	i i i | 1 | 4 2 | B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± | i i i | v | 4 2 | a a a a
P	Unchanged
B	$(rP)_b + 1$
C	i i i | v | 4 2 | B[aaaa]
E	B[aaaa]

LOAD SIGN A (LSA)

Operation Code. 43

Instruction Format.

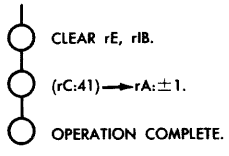
±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	n	O	p	i	i	i	i

Definition.

n: modifier for sign-digit position of A register.

Description of Operation. Replace the contents of the sign-digit position of the A register by n.

Flow Chart.



Exceptional Conditions. None.

Remarks. Although the address-field is not used for addressing purposes, B-register address-modification will occur if it is specified. No Storage ALARM STOP due to use of a nonexistent address can occur, however.

Register Status.

Register name	Contents after execution of LSA
A	(rA:±1) = n $(rA:00)_a = (rA:00)_b$
R	Unchanged
D	± i i i n 4 3 i i i i
B	Unchanged
P	$(rP)_b + 1$
C	i i i n 4 3 B[iiii]
E	Cleared

STORE P (STP)

Operation Code. 44

Instruction Format.

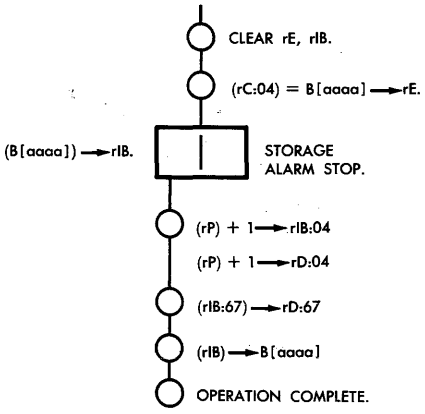
±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	i	i	i	i

Definition.

aaaa: address of base of location in which will be stored the augmented content of the P register.

Description of Operation. (rP) + 1 → (B[aaaa]:04).

Flow Chart.



Exceptional Conditions. Storage ALARM STOP due to use of a nonexistent address.

Register Status.

Register name	Contents after execution of STP
A	Unchanged
R	Unchanged
D	± i i i i 4 4 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 4 4 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± i i i i 4 4 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 4 4 B[aaaa]
E	B[aaaa]

CLEAR A (CLA)
CLEAR R (CLR)
CLEAR A, R (CAR)
CLEAR B (CLB)
CLEAR A, B (CAB)
CLEAR R, B (CRB)
CLEAR A, R, B (CLT)

Operation Code. 45

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	v	O	p	i	i	i	i

Definitions.

v: variation designator: if
(rC:41)/1 = 1:* CLEAR A will be executed: and/or
(rC:41)/2 = 1:* CLEAR R will be executed; and/or
(rC:41)/4 = 1:* CLEAR B will be executed.

* (rC:41)/1 is the one-bit of v;
(rC:41)/2 is the two-bit of v;
and (rC:41)/4 is the four-bit of v.

Description of Operation.

v	Clear Register	Abbreviation
0	None	———
1	A	CLA
2	R	CLR
3	A, R	CAR
4	B	CLB

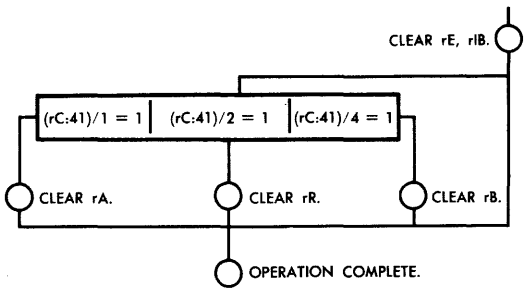
v	Clear Register	Abbreviation
5	A, B	CAB
6	R, B	CRB
7	A, R, B	CLT
8	None	———
9	A	———

Exceptional Conditions. None.

Remarks.

1. If v = 0 or if v = 8, no registers will be cleared; the effect is the same as that of a NO OPERATION instruction.

Flow Chart.



2. Although the address field is not used for addressing purposes, B-register address-modification will occur if it is specified. No Storage ALARM STOP due to use of a nonexistent address can occur, however.

Register Status.

Contents after execution of	Registers						
	A	R	D* v O p	B	P	C* v O p	E
CLA	Cl'r	Unch'g	1 4 5	Unch'g	$(rP)_b + 1$	1 4 5	Cl'r
CLR	Unch'g	Cl'r	2 4 5	Unch'g	$(rP)_b + 1$	2 4 5	Cl'r
CAR	Cl'r	Cl'r	3 4 5	Unch'g	$(rP)_b + 1$	3 4 5	Cl'r
CLB	Unch'g	Unch'g	4 4 5	Cl'r	$(rP)_b + 1$	4 4 5	Cl'r
CAB	Cl'r	Unch'g	5 4 5	Cl'r	$(rP)_b + 1$	5 4 5	Cl'r
CRB	Unch'g	Cl'r	6 4 5	Cl'r	$(rP)_b + 1$	6 4 5	Cl'r
CLT	Cl'r	Cl'r	7 4 5	Cl'r	$(rP)_b + 1$	7 4 5	Cl'r

* Control and address digits not shown here since they are irrelevant to the execution of these instructions.

CLEAR LOCATION (CLL)

Operation Code. 46

Instruction Format.

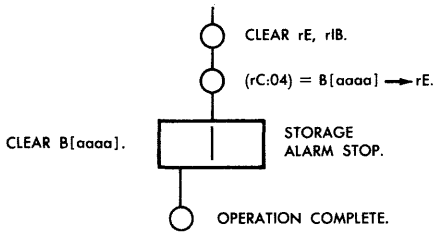
±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	a	a	a	a

Definitions.

aaaa: address of base of location to be cleared.

Description of Operation. Clear the contents of location B[aaaa].

Flow Chart.



Exceptional Conditions. None.

Register Status.

Register name	Contents after execution of CLL
A	Unchanged
R	Unchanged
D	± i i i i 4 6 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 4 6 B[aaaa]
E	B[aaaa]

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	± i i i i 4 6 a a a a
B	Unchanged
P	$(rP)_b + 1$
C	i i i i 4 6 B[aaaa]
E	B[aaaa]

SHIFT RIGHT A (SRA)

SHIFT RIGHT A AND R (SRT)

SHIFT RIGHT A WITH SIGN (SRS)

Operation Code. 48

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	v	O	p	i	i	n	n

Definitions.

v: variation designator:
- v = 0: SHIFT RIGHT A will be executed.
- v = 1: SHIFT RIGHT A AND R will be executed.
- v = 2: SHIFT RIGHT A WITH SIGN will be executed.

nn: specifies the number of digit positions (00 to 19) through which the operand will be shifted.

Description of Operation.

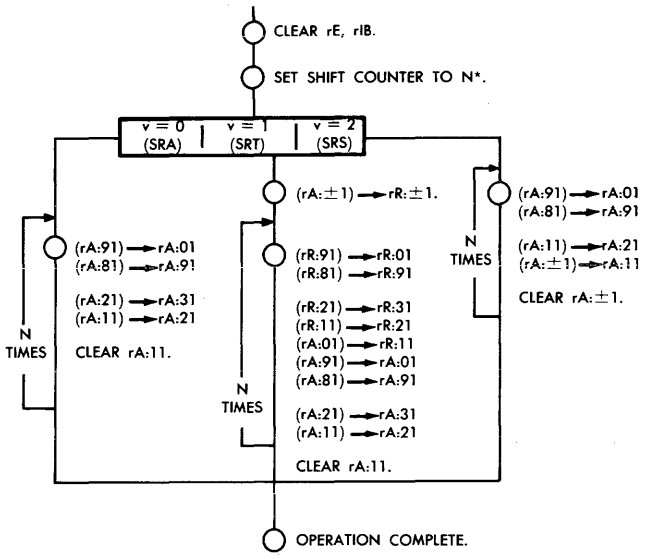
v = 0: SHIFT RIGHT A will be executed.
The contents of the A register, excluding the contents of the sign-digit position, are shifted to the right through the number of digit positions specified by nn. Digits shifted out of the low-order end of the A register are lost; as each digit is shifted out of the A register, a "0" is entered into digit-position 1 in the A register.

v = 1: SHIFT RIGHT A AND R will be executed.
The contents of the A and R registers, excluding the contents of the sign-digit positions of both registers, but regarded as one twenty-digit-long number, are shifted to the right through the number of digit positions specified by nn. Digits shifted out of the low-order end of the R register are lost; as each digit is shifted out of the R register, a "0" is entered into digit-position 1 in the A register.

Although the contents of the sign-digit position of neither register is shifted during the execution of this instruction, the contents of the sign-digit position of the R register are replaced by the contents of the sign-digit position of the A register; the sign digit of the A register is not altered.

v = 2: SHIFT RIGHT A WITH SIGN will be executed. The contents of the A register, including the contents of the sign-digit position, are shifted to the right through the number of digit positions specified by nn. Digits shifted out of the low-order end of the A register are lost; as each digit is shifted out of the A register, a "0" is entered into the sign-digit position of the A register.

Flow Chart.



*N is the least non-negative remainder obtained on dividing nn by 20.

Exceptional Conditions. None.

Remarks.

1. The SHIFT RIGHT A variation will be executed if v ≠ 1 or 2.
2. Although the address field is not used for addressing purposes, B-register address-modification will occur if it is specified. No Storage ALARM STOP due to use of a nonexistent address, can occur, however.
3. The number of digit positions through which the contents of the register(s) will be shifted is always less than or equal to 19, regardless of the value of nn. The number, N, of digit positions actually shifted is the least non-negative remainder obtained on dividing nn by 20.

Register Status.

Register name	Contents after execution of SRA
A	(rA:±1) is unchanged (rA:00) shifted as specified
R	Unchanged
D	± i i i 0 4 8 i i n n

Register name	Contents after execution of SRA
P	Unchanged
B	$(rP)_b + 1$
C	i i i | 0 | 4 8 | B[iinn]
E	Cleared

Register name	Contents after execution of SRT
A	(rA:±1) is unchanged (rA:00) shifted as specified
R	$(rR:\pm1)_a = (rA:\pm1)_b$ $(rR:00)_b$, shifted as specified
D	± | i i i | 1 | 4 8 | i i | n n
P	Unchanged
B	$(rP)_b + 1$
C	i i i | 1 | 4 8 | B[iinn]
E	Cleared

Register name	Contents after execution of SRS
A	(rA), shifted as specified
R	Unchanged
D	± | i i i | 2 | 4 8 | i i | n n
B	Unchanged
P	$(rP)_b + 1$
C	i i i | 2 | 4 8 | B[iinn]
E	Cleared

SHIFT LEFT A (SLA)

SHIFT LEFT A AND R (SLT)

SHIFT LEFT A WITH SIGN (SLS)

Operation Code. 49

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	v	O	p	i	i	n	n

Definitions.

v: variation designator:
v = 0: SHIFT LEFT A will be executed.
v = 1: SHIFT LEFT A AND R will be executed.
v = 2: SHIFT LEFT A WITH SIGN will be executed.

nn: specifies the number of digit positions through which the operand will be shifted.

Description of Operation.

v = 0: SHIFT LEFT A will be executed.
The contents of the A register, excluding the contents of the sign-digit position, are shifted to the left through the number of digit positions specified by nn. This is a circulating shift, that is, as each digit is shifted out of digit-position 1 in the A register it is entered into the low-order digit position of the A register.

v = 1: SHIFT LEFT A AND R will be executed.
The contents of the A and R registers, excluding the contents of the sign-digit positions of both registers, but regarded as one twenty-digit-long number, are shifted to the left through the number of digit positions specified by nn. This is a circulating shift, that is, as each digit is shifted out of digit-position 1 in the A register it is entered into the low-order digit position in the R register.

Although the contents of the sign-digit position of neither register is shifted during the execution of this instruction, the contents of the sign-digit position of the A register is replaced by the contents of the sign-digit position of the R register; the sign digit of the R register is not altered.

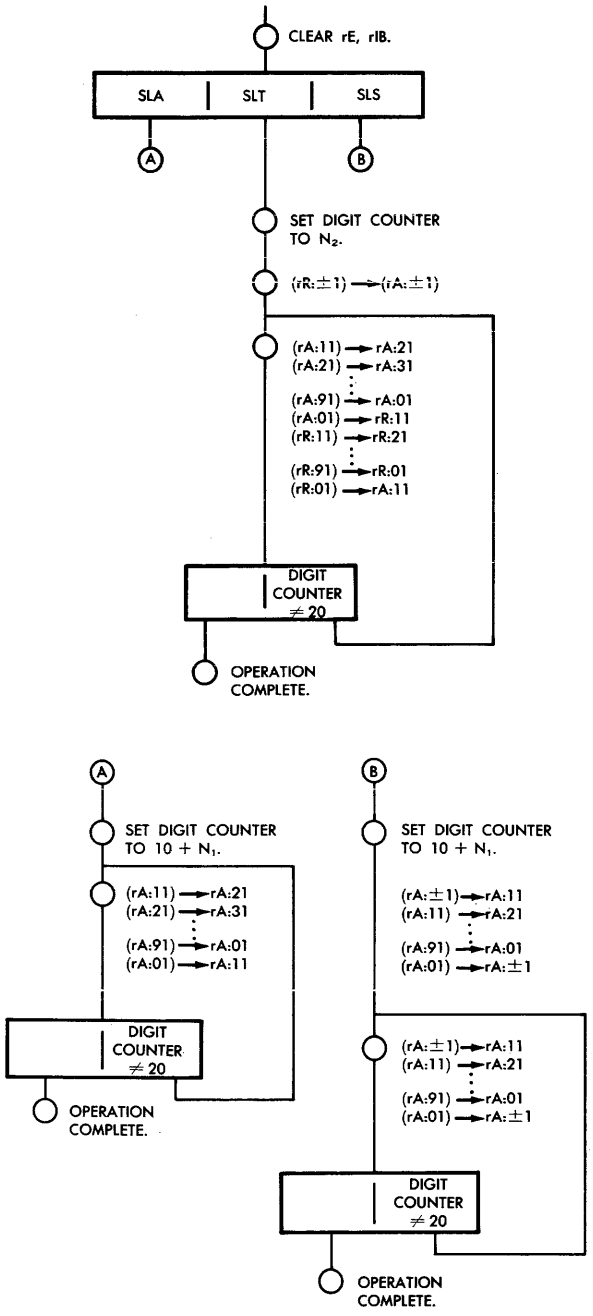
v = 2: SHIFT LEFT A WITH SIGN will be executed.
The contents of the A register, including the contents of the sign-digit position, are shifted to the left through the number of digit positions specified by nn. This is a circulating shift, that is, as each digit is shifted out of the sign-digit position in the A register it is entered into the low-order digit position in the A register.

Flow Chart. See page 2-45.

Exceptional Conditions. None.

Remarks.

1. The SHIFT LEFT A variation will be selected for execution if v ≠ 1 or 2.
2. Although the address field is not used for addressing purposes, B-register address-modification will occur if it is specified. No Storage ALARM STOP, due to use of a nonexistent address, can occur, however.
3. The Data Processor is capable only of shifting the contents of the A and R registers to the right. In order to shift to the left, therefore, it is necessary for the machine to simulate left shifts by executing circulating right shifts. This is of no other importance to the programmer/coder than that he knows how to determine execution times: for all practical purposes, if the instruction is SLS 0003, the contents of the A register will be shifted and circulated three places



to the left. In this particular case, as will be explained below, the number of digit positions through which the contents of the A register is shifted is eight, of which only seven enter into timing considerations. Examination of the flow chart may help to make clearer the description below.

The counter which determines how many digit positions are to be shifted is called the Digit Counter. The Digit Counter is capable of counting only to 20. Thus, for example, if the Digit Counter is set to 13, only seven pulses will be provided to accomplish shifting, one pulse each time the contents of the specified register(s) are shifted to the right. When the Digit Counter reaches 20, no more pulses are provided.

Suppose now that the instruction is SLA 00nn. The Data Processor first determines N_1, where N_1 = nn, modulo 10 (that is, N_1 is the least non-negative remainder obtained on dividing nn by 10). The Digit Counter is then set to $10 + N_1$. Hence, the number of digit positions actually shifted is $20 - (10 + N_1) = 10 - N_1$.

Example 1. SLA 0003. $N_1 = 3$. The Digit Counter is set to 13. The contents of (rA:00) will be shifted left, and circulated, three places by shifting right, and circulating, seven places.

Digit Counter	[rA]
13	± 1234 56 7890
14	± 0123 45 6789
15	± 9012 34 5678
16	± 8901 23 4567
17	± 7890 12 3456
18	± 6789 01 2345
19	± 5678 90 1234
20	± 4567 89 0123

Example 2. SLA 0049. $N_1 = 9$. The Digit Counter is set to 19. The contents of (rA:00) will be shifted left and circulated nine places by shifting right, and circulating, one place.

Digit Counter	[rA]
19	± 1234 56 7890
20	± 0123 45 6789

Suppose the instruction is SLS 00nn. As in the case of SLA, N_1 is determined. Also, because the operation is SLS, the Data Processor automatically shifts and circulates the contents of the A register one place to the right while it is determining N_1. The Digit Counter is set to $10 + N_1$, so that the number of additional shifts is $20 - (10 + N_1) = 10 - N_1$. In this way the proper kind of modulo-11 shift is achieved.

Example 3. SLS 0007. $N_1 = 7$. The Digit Counter is set to 17.

Digit Counter	[rA]
Original	± 1234 56 7890
17	0 ±123 45 6789
18	9 0±12 34 5678
19	8 90±1 23 4567
20	7 890± 12 3456

Because the SHIFT LEFT A AND R instruction handles sets of 20 digits, it provides a modulo-20 shift. The Data Processor first determines N_2, where $N_2 \equiv nn$, modulo 20. The Digit Counter is then set to N_2, so that the number of digit positions actually shifted is $20 - N_2$.

Example 4. SLT 0018. $N_2 = 18$.

Digit Counter	[rA]	[rR]
18	± 1234 56 7890	A BCDE FG HIJK
19	A K123 45 6789	A 0BCD EF GHIJ
20	A JK12 34 5678	A 90BC DE FGHI

4. The table below summarizes the times required to execute the various shifting instructions. Fetch time is included. As is noted in the headings, the entries in the table were computed using the following formulas:

SLA, SLS: $t = 160 - 5N_1$,
SLT: $t = 210 - 5N_2$.

N_1 and N_2 were defined in Remark 3.

nn	Time [μs]	
	SLA, SLS [$160 - 5N_1$]	SLT [$210 - 5N_2$]
00	160	210
01	155	205
02	150	200
03	145	195
04	140	190
05	135	185
06	130	180
07	125	175
08	120	170
09	115	165
10	160	160
11		150
12		150
13		145
14		140
15		135
16		130
17		125
18		120
19		115

Register Status.

Register name	Contents after execution of SLA
A	$(rA:\pm1)_a = (rA:\pm1)_b$ (rA:00) shifted as specified
R	Unchanged
D	± i i i 0 4 9 i i n n
B	Unchanged
P	$(rP)_b + 1$
C	i i i 0 4 9 B[iinn]
E	Cleared

Register name	Contents after execution of SLT
A	$(rA:\pm1)_a = (rR:\pm1)_b$ (rA:00, rR:00) shifted as specified
R	$(rR:\pm1)_a = (rR:\pm1)_b$ (rA:00, rR:00) shifted as specified
D	± i i i 1 4 9 i i n n
P	Unchanged
B	$(rP)_b + 1$
C	i i i 1 4 9 B[iinn]
E	Cleared

Register name	Contents after execution of SLS
A	(rA) shifted as specified
R	Unchanged
D	± i i i 2 4 9 i i n n
B	Unchanged
P	$(rP)_b + 1$
C	i i i 2 4 9 B[iinn]
E	Cleared

3

The Control Console

GENERAL

The Control Console of the BURROUGHS 220 Electronic Data Processing System is the control center for the System. By means of neon lamps, indicators, push-button indicators, and organ switches, the status of the System is indicated and supervisory control over its operation is provided.

The Control Console is divided into three sections, of which, normally, only the central section is of concern to the system operator and/or programmer. It is in this section that the contents of the A, R, D, B, P, C, and E registers are displayed (see Chapter 2). The other two sections, the side panels, mounted behind doors which usually are closed, are primarily for the maintenance or test engineer.

In this chapter we will discuss the ten PROGRAM CONTROL SWITCHES, the Reset and Transfer Switch, the numeric Keyboard, and the Supervisory Printer. For a more complete discussion of these and the other elements of the Control Console and for operating procedures and techniques, the reader is referred to the *Handbook of Operating Procedures for the BURROUGHS 220.*

PROGRAM CONTROL SWITCHES

The Control Console is equipped with ten clear-plastic organ switches. Each switch has a light behind it; the light is on when the switch is on. The status of a switch may be interrogated by the BRANCH, CONTROL SWITCH instruction. Although more than one switch may be on at any one time, only one switch may be interrogated with any one instruction.

Flexible manual-control facilities are provided by these switches.

THE RESET AND TRANSFER SWITCH

The manual Reset and Transfer Switch is located on the lower right portion of the Control Console. Depression of this switch will cause the contents of the address portion of the C register and the P register to be stored in memory location 0000 and program control will be transferred to memory location 0001. The contents of the P register will be stored in digit positions 7-0 and the contents of the address portion of the C register will be stored in digit positions 3-6. This switch will permit an operator to regain program control after a system halt without a loss of register information or a keyboard entry of a branching instruction.

THE KEYBOARD

A numeric Keyboard is an integral part of the Control Console. This keyboard may be activated under program control by use of the KEYBOARD ADD instruction (see page 3-2).

When the Data Processor is not in RUN status the keyboard may be activated by depressing the KEYBOARD switch.

THE SUPERVISORY PRINTER

Associated with the Control Console is a character-at-a-time alphanumeric printer. The printer, operating at a rate of 10 characters per second, is capable of printing both numeric and alphabetic information.

Controls are provided to suppress the printing of leading zeros as well as the translation of words specified as alphanumeric (an alphanumeric word is signaled by the presence of a 2 in the sign-digit position). See the *Handbook of Operating Procedures for the BURROUGHS 220.*

The following format controls are provided:

1. The right-hand margin may be set where required to provide an automatic carriage return. Carriage return always includes a single line feed.
2. A three-position switch permits the choice of "space," "tab," or "carriage return" action when the end-of-word is sensed.
3. Vertical form-control is provided so that fan-fold paper can feed past the tear line to a predetermined position or positions. These vertical tabs may be set at 2.75-, 5.5-, 8.25-, and 11-inch intervals.

The Supervisory Printer can print all the decimal digits and alphabetic characters; in addition, the following characters can be printed: -, $, &, *, □ , ., ', /, #, %, @, and "space." The BURROUGHS 220 codes for these characters are shown in Appendix C.

The Supervisory Printer is the same printer described in Chapter 5, The Paper-Tape System.

THE INTERVAL TIMER

Attached to the Control Console is a five-digit-position counter which can count to 9999.9 seconds by tenths of a second. The counter may be reset manually to zero.

The Interval Timer is connected, electrically, to the RUN Indicator: when the RUN Indicator is "on," the timer will count.

THE EXECUTE PHASE

The remainder of the Control Console chapter is devoted to descriptions of the Execute Phase of each operation which is controlled directly from the console.

KEYBOARD ADD (KAD)

Operation Code. 08

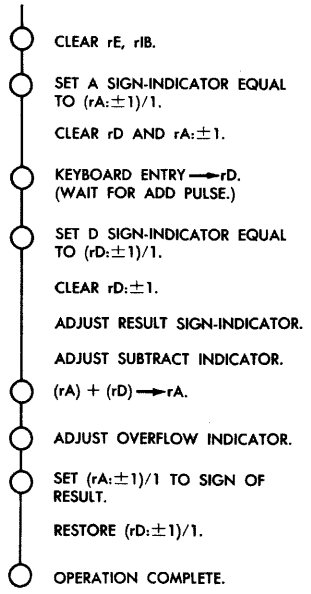
Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	i	i	i	i	O	p	i	i	i	i

Description of Operation. The D register is cleared, following which the Data Processor waits, prepared to accept information entered on the console keyboard. Each digit selected on the keyboard enters the low-order digit position of the D register, shifting the previous contents, including the contents of the sign-digit position, one place to the left; digits shifted out of the sign-digit position are lost. When the ADD key on the keyboard is depressed, the sum of the contents of the A register and the D register is generated. This sum replaces the contents of the A register.

Automatic control is resumed when the ADD key is depressed.

Flow Chart.



Exceptional Conditions. None.

Remarks.

1. Although the address-field is not used for addressing purposes, B-register address-modification will occur if it is specified. No Storage ALARM STOP, due to use of a nonexistent address, can occur, however.
2. The execution of this instruction can cause arithmetic overflow, in which case the OVERFLOW Indicator is set "on."
3. If (rA) + (rD) = 0, the sign of this result is the sign that was in the A register before execution.

Register Status.

Register name	Contents after execution of KAD
A	$(rA)_b$ + number keyed into rD
R	Unchanged
D	Number keyed in
B	Unchanged
P	$(rP)_b$ + 1
C	i i i i | 0 8 | B[iiii]
E	Cleared

SUPERVISORY PRINT-OUT (SPO)

Operation Code. 09

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	d	n	n	v	O	p	a	a	a	a

v = 0: d is not relevant to the execution of this instruction.

v = 1: d specifies the number of digit positions to the right of a decimal point provided by the Data Processor. d = 0 means 10.

nn: specifies the number of words to be printed. nn = 00 means 100.

v: variation designator:

v = 1: the Data Processor will insert a decimal point in the place specified by d.

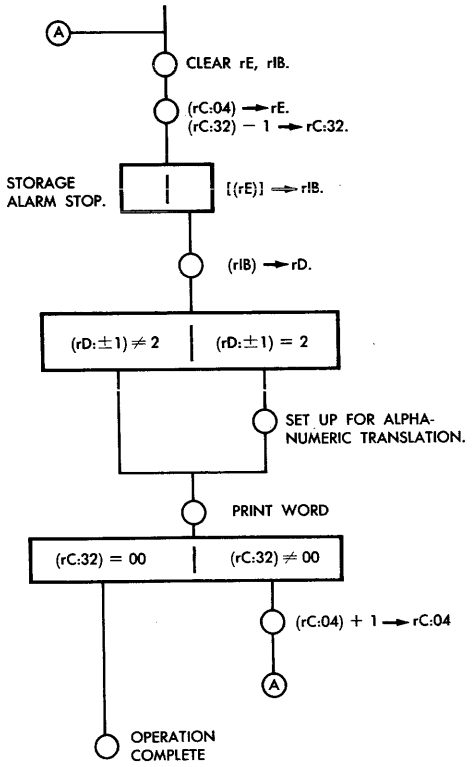
v = 0: no decimal point insertion.

aaaa: address of base of location of first word to be printed.

Description of Operation. Print, on the Supervisory Printer, nn words from consecutively addressed locations, beginning with (B[aaaa]). The next word is taken from B[aaaa] + 1. And so forth.

The abilities to control zero suppression and/or to print the contents of a location exactly as a word appears in core storage—for example, to prevent the translation of an alphanumeric word—are controlled by four switches, two of them on the Control Console (the HOLD PZT to ZERO and PUNCH SUPPRESS—LEADING ZEROS), and two of them on the Supervisory Printer, (the MAP MEMORY—NORMAL and ZERO SUPPRESS—NORMAL). The function of these switches is described in detail in the *Handbook of Operating Procedures for the BURROUGHS 220.*

Flow Chart.



Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. Digit Check ALARM STOP due to an impermissible configuration.
3. Paper Tape ALARM STOP due to Supervisory Printer being in a not-ready status.

Remarks. v ≠ 1 has the same effect as v = 0.

Register Status.

Register name	Contents after execution of SPO
A	Unchanged
R	Unchanged
D	(B[aaaa] + nn − 1)*
B	Unchanged
P	$(rP)_b$ + 1
C	d 0 0 v 0 9 B[aaaa] + nn − 1*
E	B[aaaa] + nn − 1*

* If nn = 00, B[aaaa] + nn = B[aaaa] + 100.

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	Last word printed
B	Unchanged
P	$(rP)_b$ + 1
C	d * * v 0 9 * * * *
E	* * * *

** See flow chart.
**** Address of location causing ALARM STOP.

BRANCH, CONTROL SWITCH (BCS)

Operation Code. 38

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	i	i	i	O	p	a	a	a	a

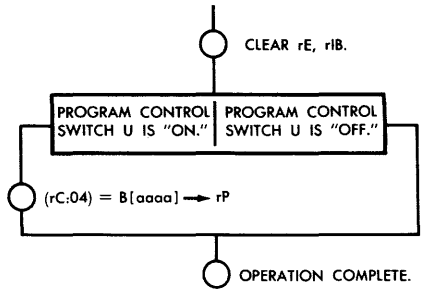
Definitions.

u: designates PROGRAM CONTROL SWITCH to be tested.

aaaa: address of base of location of alternate instruction.

Description of Operation. If PROGRAM CONTROL SWITCH u is "on," branch to location B[aaaa], i.e., prepare to take the next instruction from B[aaaa]. If SWITCH u is "off," control continues in sequence.

Flow Chart.



Exceptional Conditions. None.

Register Status.

Register name	Contents after execution of BCS
A	Unchanged
R	Unchanged
D	± u i i i 3 8 a a a a
B	Unchanged
P	B[aaaa], if SWITCH u is "on" $(rP)_h + 1$, if SWITCH u is "off"
C	u i i i 3 8 B[aaaa]
E	Cleared

4

The Magnetic-Tape System

GENERAL

Auxiliary storage capacity is provided through the medium of magnetic tape. The components of the magnetic-tape system are the Magnetic-Tape Control Unit, the Magnetic-Tape Storage Unit, and the DATAFILE* Unit.

Magnetic tape is "edited" prior to its initial use to insure a flawless writing medium. It may be re-edited at any time.

Additional protection of information recorded on magnetic tape is afforded by the use of a file protect ring: it is possible to write or edit only on reels—used by the tape storage unit—which are not equipped with a file protect ring. On the DATAFILE Unit there is a Write Lock Out Switch for each piece of tape plus a Master Lock Out switch for the entire unit; writing can occur only when the master switch and the appropriate individual switch are in the "write" position.

Information is recorded on tape in blocks whose lengths may vary from ten to one hundred words; no two adjacent blocks need be the same length. From one to ten blocks may be written or read with one instruction.

The first data word of a block is regarded as its keyword. It is possible to search[1] for a block with a given keyword. Once initiated, the searching operation is carried out independently of Data Processor control.

Any one of the first ten words may identify a category. It is possible to "scan" for blocks belonging to a specified category. Once initiated, the scanning operation is carried out independently of the Data Processor. (Search and scan are executed in distinct and different manners. The nature of the execution of each will be described in detail.)

Information is transferred between the Data Processor and magnetic tape at a nominal rate of 25,000 digits per second.

THE MAGNETIC-TAPE STORAGE UNIT

The Magnetic-Tape Storage Unit handles reels of magnetic tape, each reel containing up to 3500 feet of tape plus a 12 foot conductive leader at each end; the reels are 10-1/2 inches in diameter. Although the number of on-line tape storage units in a system cannot exceed ten, the number of reels of tape which can be used is limited only by the magnitude of the task to be accomplished.

Recording on magnetic tape is accomplished by a read-write mechanism (called the read-write head) over which tape is moved as it is wound on or unwound from the feed and take-up reels. Actually, there are two heads on each unit, fixed in position with respect to each other, to permit the recording of two parallel lanes of information on each piece of tape. The lanes are distinguished as the "even" and "odd" lanes, or lanes 0 and 1, respectively.

The tape drive mechanism is capable of moving tape in either of two directions, from feed reel to take-up reel—the forward direction—or from take-up reel to feed reel —the backward direction. The direction in which tape will be moved is a function of the operation being performed; it will be specified as each operation is described.

THE DATAFILE UNIT

The DATAFILE Unit holds fifty lengths of magnetic tape, each 270 feet in length (including a 6 foot conductive leader at each end), and each hanging freely in its own static-free bin. The bins are parallel to each other, so that the tapes are parallel to one another along their lengths.

In the DATAFILE, however, the performance of a specified operation which involves any one of the 100 lanes is accomplished by a single, movable read-write head mounted on the carriage assembly. The carriage assembly, which is driven across the width of the DATAFILE Unit on a horizontal support rod, can be positioned under any tape in the DATAFILE Unit. While the carriage assembly is moving, the read-write head is adjusted for either the even lane or the odd lane of the desired tape. Each lane in a DATAFILE Unit is distinguished uniquely by a two-digit number, in order, from 00 through 99. If we consider the tapes as numbered from 1 through 50, in order, from left to right, then tape number k contains lanes number $2k - 2$ and $2k - 1$.

Tape can be driven from the forward section of the bin to the rear section of the bin—the forward direction—or from the rear section of the bin to the forward section of the bin—the backward direction. Only the selected tape is driven.

* Trade-mark of the Burroughs Corporation.

[1] By "searching" is meant the ability to locate in the tape file a specified block without the necessity of passing a substantial part of the file through the Data Processor in order to identify the desired block. The search operation will be defined more explicitly in context.

Logically, to the Magnetic-Tape Control Unit, the DATA-FILE Unit and tape storage units are indistinguishable. For this reason, in the description which follows, the text will not distinguish between the units unless some characteristic—capacity, for example—requires mention of the difference.

TAPE FORMAT

Magnetic tape used in a BURROUGHS 220 system has provision for recording two lanes of information, parallel to one another, along the length of each tape. In each lane six channels are recorded: channels one through four are used to record decimal-digital information in the 8, 4, 2, 1 – binary code; channel five is used to provide an even-parity bit; and channel six is used for control purposes.

In each lane the recording density is approximately 208 digits per linear inch of tape. Tape is transported at the rate of 120 inches per second. The nominal transfer rate, therefore, is approximately 25,000 digits per second.

Conductive leaders, spliced to each end of a length of magnetic tape, serve to indicate the physical beginning and end of the tape to the Data Processor. Once the conductive leader is reached, the magnetic tape can be driven no further without reversing its direction.

Magnetic beginning-of-tape and end-of-tape are recorded during the editing process. During the editing process magnetic beginning-of-tape is overwritten by flaw markers to insure that information cannot be recorded too near the physical beginning of the tape. The magnetic end-of-tape area is sufficiently long to permit both the completion of any initial writing operation in progress when it is encountered as well as the initial writing of necessary end-of-tape control blocks before physical end-of-tape is encountered.

At least 6,500 hundred-word blocks will fit on a full 3,500-foot reel of tape—allowing for magnetic beginning and end-of-tape and a nominal number of flaws.

A block (of information) on magnetic tape is defined as the information which is recorded between two inter-block gaps (except for the last block, which is followed by blank tape in case all of the tape is not used, or by the magnetic end-of-tape). Associated with each block is a *preface word.* The preface word contains a two-digit number which specifies how many words are contained in the block with which the preface is associated. The manner of reading or writing the preface will be described.

The minimum block size is ten words; the maximum block size is 100 words. An attempt to write a block of information whose length is less than ten words will result in a Magnetic-Tape ALARM STOP unless the block is an end-of-tape block, where one word is designated. The format of the writing instructions precludes specification of a block length in excess of 100 words.

A typical k-word-long block layout is shown in figure 4-1. From the figure we obtain the information that a block of k words occupies space for 72 + 12k digits, including the space for one inter-block gap. However, only 11k digits represent data useful to a program. Table 4-1 shows how the percentage of useful information varies with block size.

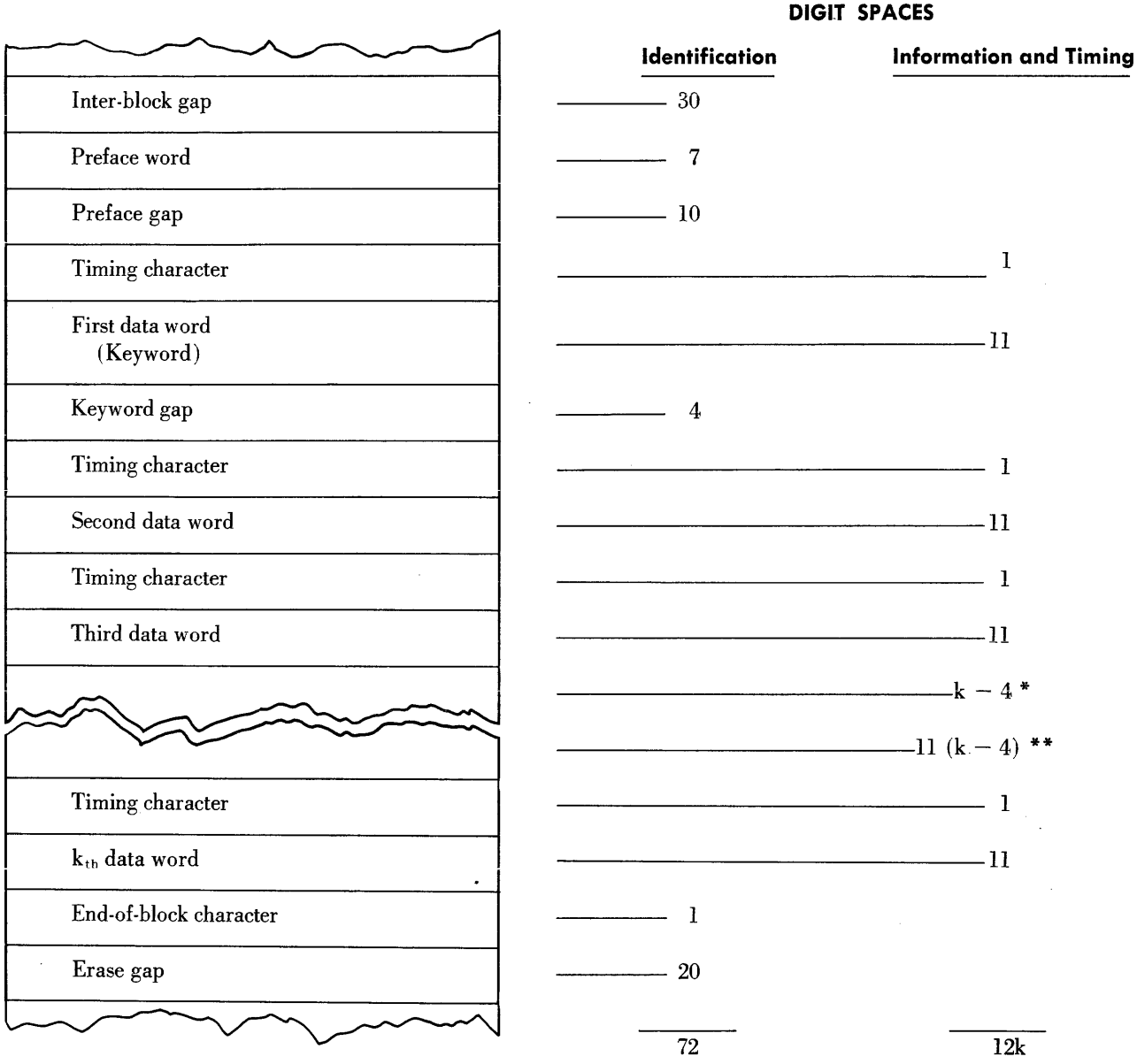
Since approximately 208 digits can be recorded per linear inch of tape, the number of digit spaces per 100 feet of tape in one lane is:

$$Q_{100} = 208 \ \frac{\text{digits}}{\text{inch}} \times 12 \ \frac{\text{inches}}{\text{foot}} \times 100 \text{ feet}$$

$$= 250{,}000 \text{ digit spaces.}$$

Table 4-1. Percentage of useful data in a block

Words per Block (k)	Useful Data (11k)	Total Digit Count (72 + 12k)	% of Useful Data
10	110	192	57.3
20	220	312	70.5
30	330	432	76.3
40	440	552	79.7
50	550	672	81.8
60	660	792	83.3
70	770	912	84.4
80	880	1,032	85.2
90	990	1,152	85.9
100	1,100	1,272	86.4



* Timing characters for words 4 through k − 1.
** Digits for words 4 through k − 1.

Figure 4-1. Block format (not to scale)

THE EDITING PROCESS

The editing procedure consists of an examination of magnetic tape to determine whether there are any imperfections—i.e., flaws—on it, and the subsequent "deletion" of the flaws by preceding them with flaw markers. A sufficient number of flaw markers are recorded preceding a flaw so that any block which is being written when the flaw-marked area is encountered can be completely written before the actual flaw is encountered. Before the next block is written, the remainder of the flaw-marked area and the flaw are skipped.

MAGNETIC-TAPE STORAGE UNITS

All Magnetic-Tape Storage Units are provided with the ability to edit tape. Prior to editing, however, it is necessary to distinguish two cases: new tape, and tape which has been used (i.e., which is to be re-edited).

If the tape has been used, it should bear a label which can be verified, under program control, as identifying a tape which it is safe to edit. After verification, necessary monitoring information and/or directions to the operator can be printed on the Supervisory Printer, following which the editing procedure can be initiated.

Table 4-2. Number of blocks and words of useful data recorded in 100 feet of flaw-free tape in one lane

Block size (k words)	Number of blocks [250,000/(72 + 12k)] = B	Number of words of useful data (W = B × k)
10	1,302.08	13,020
20	801.25	16,025
30	578.70	17,361
40	450.28	18,011
50	372.02	18,601
60	315.65	18,939
70	274.11	19,187
80	242.24	19,379
90	217.01	19,530
100	196.54	19,654

If the tape is new, it will not bear a label for interrogation, but it is possible to distinguish tape which has no information written on it. If a MAGNETIC-TAPE READ (MRD) instruction is issued to a tape which is blank, tape motion will be started and continued in an effort to find written information; thus, the tape will be moved to the end, unless there is manual intervention.

Tapes should be degaussed prior to each edit.

The initiation of the editing procedure begins with the simultaneous depression by the operator of the EDIT indicator and the EDIT button on the tape storage unit. The precise sequence of steps to initiate editing is described in the *Handbook of Operating Procedures for the BURROUGHS 220.* Once initiated, editing is carried out independently of both the Magnetic-Tape Control Unit and the Data Processor.

When the two EDIT buttons are depressed, tape movement starts in the forward direction. After reaching the physical end-of-tape, the direction of motion is reversed. Tape movement ceases and the editing process has been completed when the physical beginning-of-tape is reached. During these two passes the following are accomplished:

1. At the beginning of the forward pass, magnetic beginning-of-tape (flux changes in all information channels) is written in both lanes simultaneously for approximately 10 feet. This will appear as flaw markers after editing is completed (see 4 below).
2. During the remainder of the forward pass, the tape is "erased."
3. At the beginning of the backward pass, magnetic end-of-tape (flux changes in the parity channel only) is written in both lanes simultaneously for approximately 14 feet.
4. During the remainder of the backward pass, whenever a flaw is discovered in either lane, flaw markers (flux changes in all information channels) are written simultaneously in both lanes. Magnetic beginning-of-tape appears to be covered with flaws and is overwritten with flaw markers.

Of course, a tape label should be written after the editing process so that the tape may be identified, under program control, as one which has been edited and which is ready for initial writing.

DATAFILE UNITS

In DATAFILE Units, it is possible to edit all tapes automatically or to edit individual tapes. To edit all tapes, the Automatic Edit and Start Edit pushbuttons must be depressed simultaneously by the operator. Individual tapes are edited by depressing the Single Tape Edit and Start Edit pushbuttons simultaneously. Any tape to be edited must be in the "write" status.

Each tape to be edited is driven back and forth a total of eight times (four times per lane).

1. In the first forward pass, magnetic flux is laid down at digit rate in all six channels of the lane. The polarity of this flux is reversed each digit time. Writing takes place to within one inch of the conductive leader on each end of the tape.
2. During the first reverse pass, flux changes are read in all six channels of the lane. If a flux change is not read in every channel each digit time or if an extra flux change is read, a flaw is assumed. When a flaw is detected, the 2 channel is erased for a distance of 55 words beyond the flaw. The 2 channel then resets to "read," and reading continues until another flaw is detected or until the beginning conductive leader is sensed.

3. In the second forward pass, the first 55 words of the 2 channel only are erased. The tape is then driven to the end conductive leader.

4. During the second reverse pass, the control channel is erased completely. Flux changes are left in the parity channel for five feet. The remainder of the parity channel is then erased. As long as flux changes are sensed in the 2 channel, the 1, 4, and 8 channels are erased. If the 2 channel has no flux change (marking a flaw), the flux changes in the 1, 4, and 8 channels are retained for a distance of 110 words. Then, the 1, 4, and 8 channels are again erased. This procedure is continued until the beginning conductive leader is sensed.

After the beginning-of-tape conductive leader has been sensed, flux changes are written in the 1, 4, and 8 channels for five feet. The read-write head is then moved to the second lane of the tape and the above procedure is repeated.

Table 4-3. Defining the end-of-tape block

Operation	Description of effect on encountering End-of-Tape Block
MAGNETIC-TAPE INITIAL WRITE (MIW) MAGNETIC-TAPE INITIAL WRITE, RECORD (MIR)	Not applicable, since tape is newly edited.
MAGNETIC-TAPE OVERWRITE (MOW) MAGNETIC-TAPE OVERWRITE, RECORD (MOR)	If prefaces match, end-of-tape block is overwritten just like any other block. If prefaces do not match: 1. Terminate the operation; 2. Position tape so "next" block can be written; 3. Store (rP) in aaaa:04 and (rC:04) in aaaa:64; and 4. Transfer control to B[bbbb] (i.e., B[bbbb] → rP).
MAGNETIC-TAPE READ (MRD) MAGNETIC-TAPE READ, RECORD (MRR)	1. No words are transferred to storage; 2. Terminate the operation; 3. Position tape so "next" block can be read; 4. Store (rP) in aaaa:04 and (rC:04) in aaaa:64; and 5. Transfer control to B[bbbb] (i.e., B[bbbb] → rP).
MAGNETIC-TAPE SEARCH (MTS) MAGNETIC-TAPE FIELD SEARCH (MFS)	1. When a tape is moving forward: a. Terminate the operation; and b. Position tape so end-of-tape block can be read. 2. When tape is moving backward the end-of-tape block is treated as a normal block.
MAGNETIC-TAPE SCAN (MTC) MAGNETIC-TAPE FIELD SCAN (MFC)	1. Terminate the operation; and 2. Position tape so end-of-tape block can be read.
MAGNETIC-TAPE POSITION FORWARD (MPF) MAGNETIC-TAPE POSITION BACKWARD (MPB) MAGNETIC-TAPE POSITION AT END OF INFORMATION (MPE)	Treated as a normal block.
MAGNETIC-TAPE REWIND (MRW) MAGNETIC-TAPE REWIND, DEACTIVATE (MDA)	Rewind ignores all information written on tape.

Editing is performed independently of both the Data Processor and the Tape Control Unit. A "not ready" condition is established for the unit being edited.

WRITING ON NEWLY EDITED TAPE

The establishment of format on newly edited tape is accomplished by execution of MAGNETIC-TAPE INITIAL WRITE (MIW) or MAGNETIC-TAPE INITIAL WRITE, RECORD (MIR) instructions. These instructions write the preface associated with each block as well as the block itself. MIW is used to write from one to ten blocks, all of which contain the same number of words; MIR is used to write from one to ten blocks, no two of which need to contain the same number of words.

Two kinds of blocks are used to provide terminal-condition control: one is called an "end-of-tape block," the other a "control block." Except for its contents, each of these blocks is completely defined by the manner in which it is regarded during the execution of instructions. Table 4-3 defines the end-of-tape block; Table 4-4 defines the control block. The reader is cautioned that some of the defining conditions will become more meaningful only after he has a grasp of the manner in which instructions are executed.

The end-of-tape block is a one-word block, i.e., its preface is 01, which occupies the space of a ten-word block when it is written on tape. An end-of-tape block is the only block which can be written if the preface specified is less than ten. A block length of 02 through 09 will cause a Magnetic-Tape ALARM STOP to occur.

The first word of the end-of-tape block is the only word of the block which contains information relevant to the

Table 4-4. Defining the control block

Operation	Description of effect on encountering Control Block
MAGNETIC-TAPE INITIAL WRITE (MIW) MAGNETIC-TAPE INITIAL WRITE, RECORD (MIR)	Not applicable, since tape is newly edited.
MAGNETIC-TAPE OVERWRITE (MOW) MAGNETIC-TAPE OVERWRITE, RECORD (MOR)	If prefaces match, control block is overwritten. If prefaces do not match, a Magnetic-Tape ALARM STOP occurs.
MAGNETIC-TAPE READ (MRD) MAGNETIC-TAPE READ, RECORD (MRR)	1. All words but the control word, the last word, are transferred to memory; 2. Terminate the operation; 3. Position tape to read the next block; 4. Store (rP) in aaaa:04 and (rC:04) in aaaa:64; and 5. Transfer control to B[bbbb] (i.e., B[bbbb] → rP).
MAGNETIC-TAPE SEARCH (MTS) MAGNETIC-TAPE FIELD SEARCH (MFS)	Treated as a normal block.
MAGNETIC-TAPE SCAN (MTC) MAGNETIC-TAPE FIELD SCAN (MFC)	1. Terminate the operation; and 2. Position tape so control block can be read.
MAGNETIC-TAPE POSITION FORWARD (MPF) MAGNETIC-TAPE POSITION BACKWARD (MPB) MAGNETIC-TAPE POSITION AT END OF INFORMATION (MPE)	Treated as a normal block.
MAGNETIC-TAPE REWIND (MRW) MAGNETIC-TAPE REWIND, DEACTIVATE (MDA)	Rewind ignores all information written on tape.

program (the rest of the block written on tape is an erase block which automatically is supplied by the tape control unit). The format of the relevant word is the following: ± ii aaaa bbbb. The sign digit may be used to specify B-register address-modification of bbbb when the control word is read; ii are not relevant; aaaa is the address of the location in which will be stored information enabling the program to determine the circumstances in which the end-of-tape block is encountered (see Table 4-3), and B[bbbb] is the address of an alternate instruction, that is, the address of the location to which control is transferred when the block is encountered.

A control block is a uniquely identified block, namely any block, with the exception of an end-of-tape block, with a 7 in the sign-digit position of the keyword.[2] End-of-tape blocks cannot also be used as control blocks: if an end-of-tape block has a 7 in the sign-digit position of the control word, the 7 will not be recognized as designating a control block.

The last word of a control block is a control word which has the same format as the control word in an end-of-tape block. The control word in a control block serves the same function as the control word in an end-of-tape block. (See Table 4-4 for details.)

A MPE instruction should precede the first MIW or MIR instruction issued to a particular tape to position the tape at the end of the magnetic beginning-of-tape flaw markers. The writing of the first preface does not begin until an amount of *blank* tape equivalent to inter-block gap is passed. The writing of each subsequent preface is subject to the same restriction: after the end-of-block character and erase gap are written—or sensed, if another MIW or MIR instruction is issued—writing is inhibited until an amount of tape equivalent to inter-block gap is passed. In this manner, writing in the flaw-marked area at the beginning of tape is prevented; also, writing over a flaw is prevented, since a flaw-marked area preceding an imperfection is long enough to permit the completion of the writing of any maximum-length block if the flaw-marked area is encountered while writing is in process.

The magnetic end-of-tape condition is established at the completion of the operation if magnetic end-of-tape is encountered during the execution of a MIW or MIR instruction. The writing operation will be completed: during the editing process the amount of tape marked as magnetic end-of-tape is more than sufficient to permit the completion of any instruction which may be in progress at the time magnetic end-of-tape is encountered as well as the writing of the control information.

Every MIW and MIR instruction must either be preceded or followed by a MAGNETIC-TAPE INTERROGATE END-OF-TAPE, BRANCH (MIE) instruction to determine whether magnetic end-of-tape was encountered by the last MIW or MIR instruction. It is imperative that the MIW or MIR operation be completed before initiating the MIE instruction. The MIE, therefore, must be immediately preceded by a MAGNETIC-TAPE INTERROGATE, BRANCH (MIB) instruction.

Example:

Location	*Control*	*Op*	*Address*
1000	1000	MIB	1002
1001		BUN	1000
1002	1001	MIE	yyyy
1003	1101	MIW	aaaa

If magnetic end-of-tape was encountered, a control or end-of-tape block must be written to identify the end of information.

SEARCH

MAGNETIC-TAPE SEARCH (MTS) is an instruction which provides the ability to locate in a tape file a specified block. The block specified is the one whose first word, or keyword, is identical with a specified search key. The sign digit is excluded from the search key. Once initiated, searching is done under direction of the Magnetic-Tape Control Unit, independently of the Data Processor.

The searching operation is executed as follows: tape is moved first in the forward direction until a block is encountered whose keyword is greater than or equal to the search key.[3] The direction of tape motion is then reversed; tape moves backward until one of two events occurs:

1. A block is found whose keyword is identical with the search key. In this case the operation terminates with tape positioned so that the head can read the block whose keyword is identical with the search key.

2. A block is found whose keyword is less than the search key. In this case, the direction of tape motion is again reversed, and tape is moved in the forward direction. Tape movement stops with the head near the end of the block whose keyword is less than the search key. The head is in position to read the next block, that is, the first block whose keyword is greater than the search key.

It is apparent from this last statement that a searching operation may terminate with the tape positioned so the

[2] As will be seen, the sign-digit position of the keyword cannot be included in the search key; that is, the contents of the sign-digit position cannot be used as part of the keyword of a block. The sign-digit is also excluded from scan key.

[3] In case an end-of-tape block is encountered during a searching operation, the operation is modified as indicated in Table 4-3.

head can read a block whose keyword is different from the search key. Hence, it is necessary to verify, under program control, that the block sought has been found.

It should be noted, also, that the nature of the searching operation imposes one important requirement on the structure of a file: the blocks which are written in any lane must be arranged in order of increasing keywords, from beginning to end, if the searching operation is to be used. Techniques for partitioning a lane so that it may contain a part or all of more than one file must also order each file within partitions.

It is possible to execute a search instruction using a partial-word search key. The definition of the partial-word boundaries, sL, must be pre-set in the two high-order digit positions in the B-register [that is, sL = (rB:82)]. The instruction is MAGNETIC-TAPE FIELD SEARCH (MFS).

SCAN

MAGNETIC-TAPE SCAN (MTC) is an instruction which provides the ability to locate blocks having a code which categorizes the blocks as belonging to a certain class; for example, from an accounts receivable file we may wish to examine the record of every account which is delinquent.

The category code of a block may be in any one of the first ten words of the block. The sign-digit position of a word is excluded from the category code.

The scanning operation is executed as follows: tape is moved in the forward direction in an attempt to locate a block whose category code is identical with the scan key. The operation will terminate when either of two events occurs:

1. A block is found whose category code is identical with the scan key. The operation terminates with the head in position to read the block whose category code is identical with the scan key. Or,

2. A control block or end-of-tape block is encountered, in which case the operation terminates with the head in position to read the control or end-of-tape block.

Note that a scanning operation can terminate without having located a sought-for block. Hence, it is necessary to verify, under program control, that a sought-for block has been found.

It is possible to execute a scanning operation using a partial-word scan key. The definition of the partial-word boundaries, sL, must be pre-set in the two high-order digit positions in the B-register [that is, sL = (rB:82)]. The instruction is MAGNETIC-TAPE FIELD SCAN (MFC).

READING

It is possible to read from one to ten blocks with one instruction. If these blocks are all the same length, a MAGNETIC-TAPE READ (MRD) instruction is executed. The first word of the first block read (the keyword of that block) is stored in memory in the location whose address is specified in the MRD instruction. Succeeding words of the first, and all following blocks, are written into consecutively addressed locations following the location whose address is specified in the instruction.

If the blocks to be read are of variable lengths, i.e., their lengths are not known to the program in advance, a MAGNETIC-TAPE READ, RECORD (MRR) instruction is executed. This instruction causes the preface (which specifies how many words are contained in the block) associated with each block to be read into storage immediately preceding the first word of the associated block. Suppose, for example, that the instruction orders three blocks to be read, that the address specified in the instruction is 1001, and that the three blocks are, in order, 23, 37, and 17 words long. Figure 4-2 shows the allocation of storage after completion of the instruction.

Note that the preface occupies the two high-order digit positions of the location in which it is stored.

Note also that Figure 4-2 shows the allocation of storage required for the execution of a MAGNETIC-TAPE INITIAL WRITE, RECORD instruction.

Location	Contents
1001	0 2300 00 0000 (preface of first block)
1002	Keyword of first block
•	•
•	•
•	•
1024	23rd word of first block
1025	0 3700 00 0000 (preface of second block)
1026	Keyword of second block
•	•
•	•
•	•
1062	37th word of second block
1063	0 1700 00 0000 (preface of third block)
1064	Keyword of third block
•	•
•	•
•	•
1080	17th word of third block

Figure 4-2. Example of allocation of storage on execution of MRR

It is possible for a read instruction to be terminated without having read as many blocks as are specified in the instruction. This event may occur in case a control or end-of-tape block is encountered during the operation. The consequences of encountering a control or end-of-tape block during a reading operation are specified in Table 4-4 and Table 4-3, respectively.

OVERWRITING

The Magnetic-Tape System of the BURROUGHS 220 features the ability to modify information already recorded on tape. That is, the contents of any block may be altered, as required, without the need to copy the entire tape. This ability is provided by two instructions, MAGNETIC-TAPE OVERWRITE (MOW) and MAGNETIC-TAPE OVERWRITE, RECORD (MOR).

The execution of these instructions differs from the execution of the corresponding MIW and MIR instructions only in these essential respects:

1. It is not possible to execute MOW or MOR on newly-edited tape. If an attempt is made to overwrite on newly-edited tape, the tape will be moved to the physical end-of-tape marker, unless there is manual intervention.
2. Before overwriting begins, the preface written on tape is compared with the preface specified by the instruction. If the prefaces are identical, the block on tape will be overwritten; otherwise, a Magnetic-Tape ALARM STOP will occur, unless the mismatch occurs with an end-of-tape block. (See Table 4-2.)

The contents of a control block or end-of-tape block may be modified using the MOW or MOR instructions.

POSITIONING

MAGNETIC-TAPE POSITION FORWARD (MPF) and MAGNETIC-TAPE POSITION BACKWARD (MPB) permit tape to be moved forward or backward, respectively, passing over from one to ten blocks. Once initiated, these instructions are executed under direction of the Magnetic-Tape Control Unit, independently of the Data Processor.

MAGNETIC-TAPE POSITION AT END OF INFORMATION (MPE) moves tape, positioning it so that the read-write head is near the end of the last block written (that is, near the end of information), ready to initial write the next block.

MAGNETIC-TAPE REWIND (MRW) and MAGNETIC-TAPE REWIND, DE-ACTIVATE (MDA) allow tape to be rewound at the direction of the program. Once initiated, these instructions are executed under direction of the Magnetic-Tape Storage Unit or DATAFILE Unit, independently of both the Data Processor and the Magnetic-Tape Control Unit. On a Tape Storage Unit the MDA instruction differs from MRW in one essential respect: MDA causes interlock to be set; the interlock must be released manually by the system operator before the tape unit specified can be used by the program. If an instruction refers to a unit which is inoperative because the interlock has not been released, a Magnetic-Tape ALARM STOP will occur. On a DATAFILE Unit the MDA functions in the same manner as a MRW.

MAGNETIC-TAPE LANE SELECT (MLS) allows for the positioning of a read-write head in a specified lane without any tape motion.

INTERROGATION

MAGNETIC-TAPE INTERROGATE, BRANCH (MIB) permits the program to determine if a designated tape unit is ready to be used. If the unit queried is ready, control is transferred. If the unit is not ready, or turned off, control will continue in sequence.

INPUT SIGN CONTROL

MAGNETIC-TAPE READ and MAGNETIC-TAPE READ, RECORD may be coded to permit B-register address-modification of designated words read from magnetic tape.

B-register address-modification *can* occur only if (rC:41)/8=1. The word read from tape is examined in the D register to see if B-register address-modification is designated.

1. If (rD:±1)/8=1, B-register address-modification will occur; at the same time, (rD:±1)/8 is set to zero.
2. If (rD:±1)/8=0, no modification of the word read from tape will occur.

There is one exception to the above; if B-register address-modification of a control word (in a control or end-of-tape block) is intended, the sign digit of the control word should be odd. Then, when the control word is read, B-register address-modification will occur no matter how the instruction is coded.

PARITY CHECKING

As noted above, every digit is written on magnetic tape with an even-parity bit. A parity error can be detected, however, only during the portion of an operation in which the read-write head is in "read" status. For example, the head is in "read" status during an entire searching operation.

Generally speaking, the detection of parity errors falls into two classes:

1. The detection of errors in the part of the block which is relevant to the instruction being executed.

(For example, the block keyword when searching; the entire block when reading.) This is called a Class I error.

2. The detection of errors in a part of the block which is not relevant to the instruction being executed. (For example, all words other than the block keyword when searching.) This is called a Class II error.

In the BURROUGHS 220, only parity errors which fall into the first class are detected. The detection of Class II errors is ignored. Table 4-5 defines the Class I errors.

In conclusion, two statements can be made about parity errors:

1. The detection of a parity error has priority over any other kind of error which might occur simultaneously. For example, during the performing of a MAGNETIC-TAPE OVERWRITE instruction several kinds of errors may occur. Among these could be a parity error in the preface or a preface mismatch. (See page 4-20.) Should both of these errors occur in the same preface, the parity error has priority over the preface mismatch.

2. The initial detection of a parity error results, automatically, in two attempts to repeat the operation during which the error is detected in an attempt to eliminate the detected error. For example, if the error is detected during MAGNETIC-TAPE READ, two additional attempts will be made to read the block. In case the parity error cannot be eliminated, a Magnetic-Tape ALARM STOP will occur. Tape movement will stop with the head positioned to read the block in which the error was detected.

In MRD and MRR the block in question will have been read from tape and stored in core storage. If the error is a forbidden combination the 8 bit of the digit in error will be suppressed.

Table 4-5. Definition of Class I parity errors

Operation	Location of Error	Remarks
MAGNETIC-TAPE SEARCH MAGNETIC-TAPE FIELD SEARCH	1. Keyword 2. Preface of end-of-tape block	1. Correct keyword, but appears to be a. incorrect. b. correct (if the parity bit is "dropped"). 2. Incorrect keyword, but appears to be a. correct. b. incorrect. 3. End-of-tape block must be recognized.
MAGNETIC-TAPE SCAN MAGNETIC-TAPE FIELD SCAN	1. Category code. 2. Control block marker-digit in keyword of block. 3. Preface of end-of-tape block.	1. Correct category code, but appears to be a. incorrect b. correct (if the parity bit is "dropped"). 2. Incorrect category code, but appears to be a. correct. b. incorrect. 3. Control block must be recognized. 4. End-of-tape block must be recognized.
MAGNETIC-TAPE READ MAGNETIC-TAPE READ, RECORD	1. Entire block. 2. Preface. 3. Control block marker-digit in keyword of block.	1. Control block must be recognized. 2. End-of-tape block must be recognized.
MAGNETIC-TAPE OVER-WRITE MAGNETIC-TAPE OVER-WRITE, RECORD	1. Preface	1. Correct preface, but appears to be a. incorrect. b. correct (if parity bit is "dropped"). 2. Incorrect preface, but appears to be a. correct. b. incorrect. 3. Parity has priority over preface mismatch.

DIGIT-COUNT/WORD-COUNT CHECKING

Preceding every word written on tape is a timing character. During reading operations this character is used by the Magnetic-Tape Control Unit to verify that every word read is 11 digits long.

During reading operations the preface associated with the block being read is used by the Magnetic-Tape Control Unit to verify that all of the data words in the block have been read.

If, during a reading operation, the digit count and/or the word count fail to check, two additional attempts will be made to read the block in question. These re-tries are made automatically under direction of the Magnetic-Tape Control Unit. If the digit count or the word count fails to check in both of these attempts to read the block, a Magnetic-Tape ALARM STOP will occur, with the read-write head in position to read the block in which the error occurred.

The questionable block will have been read from tape and stored in core storage.

INFORMATION FLOW

The general nature of information flow during input and output was discussed in Chapter 2. In some details, however, the flow of information during execution of magnetic-tape operations is different from that indicated in Chapter 2. For this reason, the flow of information is described again below.

INPUT FLOW

Input flow is shown in Figure 4-3.

1. (rD:00)—the D register contains an image of the instruction brought from storage during the Fetch Phase—are transferred to the T register in the Magnetic-Tape Control Unit.
2. The address part of the C register—which specifies the location in storage where the first word read from tape will be stored—is copied into the four high-order digit positions of the C register, that is, (rC:04) → rC:44. This address is preserved in case a parity error is detected or the digit-count/word-count check fails and an attempt is made—under machine control—to re-read the block in which the error is detected: the Data Processor must have the address of the location into which is to be written the first word of the block read from tape.
3. The address part of the C register—which specifies the location in storage where the next word read from magnetic tape is to be stored—is transferred to the E register. At the same time, the address part of the C register is counted up 1.
4. The next word on tape is read and transferred to the D register.
5. The contents of the D register are then transferred to the IB register, with or without B-register modification as specified and indicated.
6. The contents of the IB register are transferred to storage. If all of the words of the block have not been read, there is a return to step 3 for the next word.

If all of the blocks, as specified by the instruction, have not been read, there is a return to step 2 to prepare for reading the next block.

Such modifications in the flow as are required when a control or end-of-tape block is encountered will be explained in the description of the Execute Phase of the appropriate instruction.

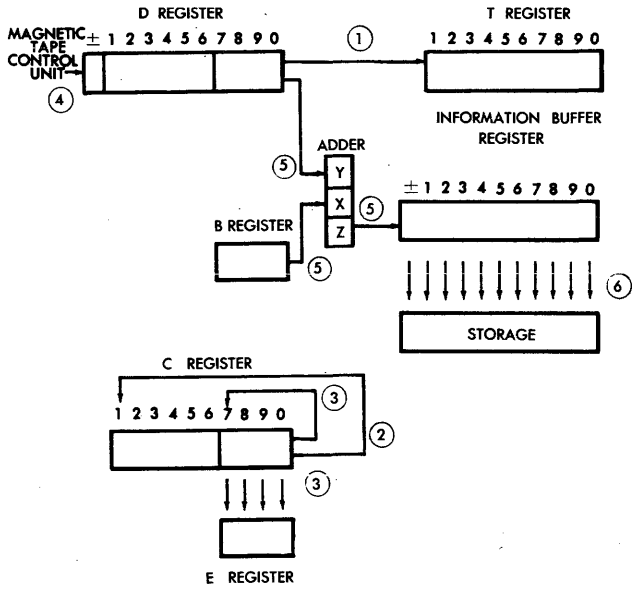


Figure 4-3. Input flow.

OUTPUT FLOW

Output flow is shown in Figure 4-4.

1. (rD:00)—the D register contains an image of the instruction brought from storage—are transferred to the T register.
2. The address part of the C register—which specifies the location in storage from which will be taken the first word to be written on tape—is copied into the four high-order digit positions of the C register, that is, (rC:04) → rC:44.
3. The address part of the C register—which specifies the location in storage from which will be taken the next word to be written on tape—is transferred to the E register. At the same time, the address part of the C register is counted up 1.
4. The word is transferred from storage to the IB register.

5. The contents of the IB register pass through the adder to the Magnetic-Tape Control Unit and thence to the designated tape unit, with the low-order digit of the word written first.

If all of the words of the block have not been written, there is a return to step 3 for the next word.

If all of the blocks, as specified by the instruction, have not been written, there is a return to step 2 to prepare for writing the next block.

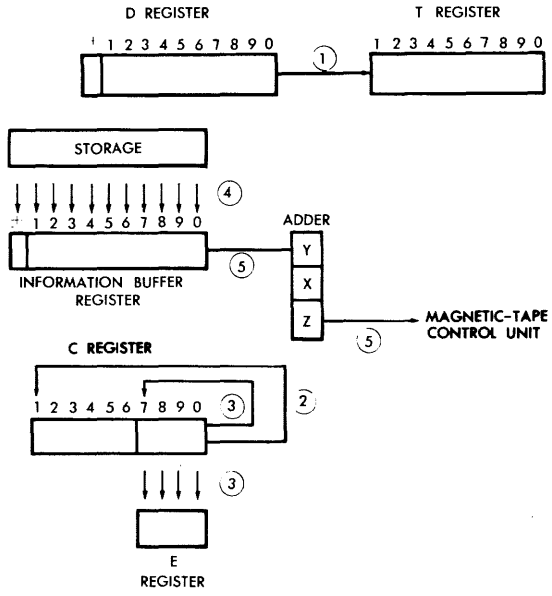


Figure 4-4. Output flow.

USE OF BURROUGHS 205 MAGNETIC TAPE

It is possible to read from magnetic tape which has been prepared by a BURROUGHS 205 system by using a 205/220 Magnetic-Tape Converter.

The reader may recall that a word is written on BURROUGHS 205 magnetic tape with sign digit first, low-order digit last. BURROUGHS 220 tape, on the other hand, is written with low-order digit first, sign digit last.

It is necessary, therefore, to invert the order of the digits in a word which may be written on BURROUGHS 205 tape before the word can be processed in the BURROUGHS 220. This may be accomplished, by means of an inversion program in the BURROUGHS 220, after the word has been read.

EXCEPTIONAL CONDITIONS

Following is a list summarizing conditions which can occur and are detected by the BURROUGHS 220. In addition to those enumerated below, a Storage ALARM STOP (nonexistent address) and Program Check ALARM STOP (field overflow) can occur.

A Magnetic-Tape ALARM STOP will occur:

1. Whenever a failure in parity is detected and attempts to eliminate the error have not been successful. Parity errors are defined by Table 4-5.
2. During a reading operation if either the digit count (per word) or the word count (per block) fails to check.
3. If an instruction specifies the writing of a block whose length is 2, 3, 4, 5, 6, 7, 8, or 9 words.
4. If the two prefaces are not identical during the execution of a MOW or MOR instruction and the block on tape is not an end-of-tape block.
5. If an attempt is made to write on a unit which is in not-write status.
6. If the next instruction is one which initially moves tape in the forward direction when the tape is positioned at physical end-of-tape.

If physical end-of-tape is sensed during the execution of MTS, MFS, MTC, MFC, or MPF instructions, an "operation complete" signal will not be generated. Manual intervention is required before processing can proceed.

If physical beginning-of-tape is sensed during the execution of MTS, MFS, or MPB instructions, an "operation complete" signal will not be generated. Manual intervention is required before processing can proceed.

7. If the unit referred to by an instruction is inoperative.

THE EXECUTE PHASE

The remainder of Chapter 4 is devoted to a description of the Execute Phase of all the instructions which refer to magnetic tape.

MAGNETIC-TAPE SEARCH (MTS)

MAGNETIC-TAPE FIELD SEARCH (MFS)

MAGNETIC-TAPE LANE SELECT (MLS)

MAGNETIC-TAPE REWIND (MRW)

MAGNETIC-TAPE REWIND, DE-ACTIVATE (MDA)

Operation Code. 50

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	u	h	h	v	0	p	a	a	a	a

Definitions.

u: designates tape unit.

hh: if u is a tape storage unit, lane 0 or 1 is selected according as hh is an even or odd number.
if u is a DATAFILE Unit, lane hh is selected.

v: variation designator:

v = 0: ± = 0 or 1: MAGNETIC-TAPE SEARCH will be executed.
± = 4 or 5: MAGNETIC-TAPE FIELD SEARCH will be executed.

v = 4: MAGNETIC-TAPE LANE SELECT will be executed.

v = 8: MAGNETIC-TAPE REWIND will be executed.

v = 9: MAGNETIC-TAPE REWIND, DE-ACTIVATE will be executed.

aaaa: v = 0, 1, 2, 3 address of base of location of search key.

v ≠ 0, 1, 2, 3 aaaa is not relevant to the execution of these instructions.

Description of Operation.

v = 0, ± ≠ 4 or 5: MAGNETIC-TAPE SEARCH will be executed. Search on unit u, in the lane specified by hh, for the block whose keyword is identical with (B[aaaa]:00), the search key. When the block sought has been found, the operation will terminate with the read-write head positioned to read the desired block. After MAGNETIC-TAPE SEARCH is initiated, it is executed under control of the Magnetic-Tape Control Unit, independently of the Data Processor.

v = 0, ± = 4 or 5: MAGNETIC-TAPE FIELD SEARCH will be executed. Search on unit u, in the lane specified by hh, for the block, the corresponding part of whose keyword is identical with (B[aaaa]:sL), the partial-word search key. The partial-word boundaries are defined by (rB:82) = sL. When the block sought has been found, the operation will terminate with the read-write head positioned to read the desired block.

After MAGNETIC-TAPE FIELD SEARCH is initiated, it is executed under control of the Magnetic-Tape Control Unit, independently of the Data Processor.

v = 4: MAGNETIC-TAPE LANE SELECT will be executed. On unit u, select the read-write head in the lane specified by hh. There is no tape movement. After MAGNETIC-TAPE LANE SELECT is initiated it is executed under control of the Magnetic-Tape Control Unit, independently of the Data Processor.

v = 9: MAGNETIC-TAPE REWIND, DE-ACTIVATE will be executed. Rewind unit u to the physical beginning-of-tape marker. At completion of the rewind, select the read-write head in the lane specified by hh. Then, de-activate the unit.

On a DATAFILE Unit, rewind tape containing lane hh on unit u. At completion of rewind, position head under lane hh. Unit is not de-activated.

After MAGNETIC-TAPE REWIND, DE-ACTIVATE is initiated, it is executed under control of the magnetic-tape unit, independently of both the Data Processor and Magnetic-Tape Control Unit. MAGNETIC-TAPE REWIND, DE-ACTIVATE will operate the same as a MAGNETIC-TAPE REWIND when the unit addressed is a DATAFILE Unit.

v = 8: MAGNETIC-TAPE REWIND will be executed. Except that the unit is *not* de-activated, this variation is the same as that executed when v = 9.

Exceptional Conditions.

1. Storage ALARM STOP caused by a nonexistent address.
2. Program Check ALARM STOP. The test for field overflow is made before initiating the searching operation: if field overflow is detected, there will be no tape movement.
3. Magnetic-Tape ALARM STOP caused by parity error, end-of-tape error, or unit not ready.

Remarks.

1. The MAGNETIC-TAPE SEARCH or MAGNETIC-TAPE FIELD SEARCH variation will be selected for execution if v = 0, 1, 2, or 3. Which of MTS or MFS will be selected is determined by the sign digit of the instruction.
2. The MAGNETIC-TAPE LANE SELECT variation will be selected for execution if v = 4, 5, 6, or 7.
3. The nature of the searching operation is defined as follows: Tape movement starts in the forward direction and continues in that direction until one of two events occurs:
   a. A block is encountered whose keyword is greater than or equal to the search key; or
   b. An end-of-tape block is encountered. When an end-of-tape block is encountered, the operation is terminated with the head in position to read the end-of-tape block.

When (a) occurs, the direction of tape motion is reversed. Tape continues to move in the backward direction until one of two events occurs:

   c. A block is encountered whose keyword is equal to the search key. In this case, the operation terminates with the head positioned to read the block whose keyword is identical with the search key.
   d. A block is encountered whose keyword is less than the search key.

When (d) occurs, the direction of tape motion is again reversed. Tape moves in the forward direction until the

read-write head is near the end of the block whose keyword is less than the search key: tape movement stops with the head in position to read the block following the one whose keyword is less than the search key. That is, tape movement ceases with the head in position to read the first block whose keyword is greater than the search key.

4. As a result of the definition of the searching operation, it can happen that an instruction to search will fail to find the sought-for block. It is necessary, therefore, to verify under program control that an instruction to search has in fact found the desired block.

5. As a result of the definition of the searching operation, an order structure is imposed on a file, or that part of it which is in one lane, if it is one in which searching is to be done: the file must be set up so that the blocks are in ascending sequence by keyword.

6. If the keyword of *every* block on a tape is greater than the search key, the conductive leader which marks the physical beginning-of-tape will be encountered during the backward pass after the first block has been passed. Tape movement—and, hence, searching—stops when the conductive leader is sensed. No "operation complete" signal is generated, however; so, the Magnetic-Tape Control Unit will appear to be busy if a MIB instruction is issued or if an attempt is made to use tape. Manual intervention is required in order to proceed with processing.

In a properly established file this event will not occur.

7. If the keyword of *every* block on tape is less than the search key, *and* if the tape is *not* bounded by an end-of-tape block, the conductive leader which marks the physical end-of-tape will be encountered during the forward pass after the last block has been passed. Tape movement—and, hence, searching—stops when the conductive leader is sensed. No "operation complete" signal is generated, however; so, the Magnetic-Tape Control Unit will appear to be busy if a MIB instruction is issued or if an attempt is made to use tape. Manual intervention is required in order to proceed with processing.

In a properly established file this event will not occur.

8. If the Magnetic-Tape Control Unit is directing a searching operation at the time another magnetic-tape instruction is issued—with the exception of MIB and MIE instructions—the Data Processor will wait for an "operation complete" signal from the Magnetic-Tape Control Unit before attempting to complete the execution of the instruction.

9. If a tape unit is rewinding at the time another magnetic-tape instruction is issued (with the exception of MIB and MIE instructions) the Data Processor will wait for an "operation complete" signal from the referenced tape unit before attempting to complete the execution of the instruction. If the rewinding instruction is MDA, manual intervention is required before processing can be continued on that unit.

10. Blank tape—that is, tape on which no information is recorded—is treated as inter-block gap.

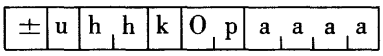
11. The sign digit cannot be included in a partial-word search key. If it is, a Program Check ALARM STOP will occur.

MAGNETIC-TAPE SCAN (MTC)

MAGNETIC-TAPE FIELD SCAN (MFC)

Operation Code. 51

Instruction Format.



Definitions.

±: if ± is odd, B-register address-modification will occur; otherwise, there will be no such modification.
± = 4 or 5: MAGNETIC-TAPE FIELD SCAN will be executed.

u: designates magnetic-tape unit.

hh: if u is a tape storage unit, lane 0 or 1 is selected according as hh is an even or odd number;
if u is a DATAFILE Unit, lane hh is selected.

k: specifies in which word of a block the category code will be found. If k = 0, the tenth word of the block contains the category code.

aaaa: address of base of location of scan key.

Description of Operation.

≠ 1 or 5: MAGNETIC-TAPE SCAN will be executed. an on unit u, in the lane specified by hh, for the block whose category code in the k-th word is identical with (B[aaaa]:00), the scan key. When the block sought has been found, the operation will terminate with the head positioned to read the sought-for block.

After MAGNETIC-TAPE SCAN is initiated, it is executed under control of the Magnetic-Tape Control Unit, independently of the Data Processor.

± = 4 or 5: MAGNETIC-TAPE FIELD SCAN will be executed. Scan on unit u, in the lane specified by hh, for the block the corresponding part of whose category code in the k-th word is identical with (B[aaaa]:sL), the scan key. The partial-word boundaries are defined by (rB:82) = sL. When the block sought has been found, the operation will terminate with the head positioned to read the sought-for block.

After MAGNETIC-TAPE FIELD SCAN is initiated, it is executed under control of the Magnetic-Tape Control Unit, independently of the Data Processor.

Exceptional Conditions.

1. Storage ALARM STOP caused by a nonexistent address.
2. Program Check ALARM STOP. The test for field overflow is made before initiating the searching operation: if field overflow is detected, there will be no tape movement.
3. Magnetic-Tape ALARM STOP caused by parity error, end-of-tape error or unit not ready.

Remarks.

1. The nature of the scanning operation is defined as follows: tape movement starts in the forward direction and continues in that direction until one of two events occurs:
   a. A block is encountered whose category code is identical with the scan key; or
   b. A control block or end-of-tape block is encountered. In either case, the operation terminates with the head positioned to read the block which caused the operation to terminate.
2. Note, therefore, that both MTC and MFC can terminate without having found a sought-for block. It is necessary, therefore, to verify under program control that a sought-for block has been found.
3. These two operations do *not* require an ordered file.
4. If there is no block on tape whose category code is identical with the scan key, *and* if the tape is *not* bounded by a control block or end-of-tape block, the conductive leader which marks the physical end-of-tape will be encountered after the last block has been passed. Tape movement—and, hence, scanning, —stops when the conductive leader is sensed. No "operation complete" signal is generated, however; so the Magnetic-Tape Control Unit will appear to be busy if a MIB instruction is issued or if an attempt is made to use tape. Manual intervention is required in order to proceed with processing.

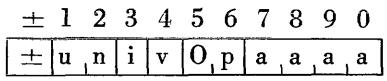
In a properly established file, this event will not occur.

5. If the Magnetic-Tape Control Unit is directing a scanning operation at the time another magnetic-tape instruction is issued—with the exception of MIB and MIE instructions—the Data Processor will wait for an "operation complete" signal from the Magnetic-Tape Control Unit before attempting to complete the execution of the instruction.
6. Blank tape—that is, tape on which no information is recorded—is treated as inter-block gap.
7. The sign digit cannot be included in a partial-word scan key. If it is, a Program Check ALARM STOP will occur.

MAGNETIC-TAPE READ (MRD)

Operation Code. 52

Instruction Format.



Definitions.

u: designates magnetic-tape unit.

n: specifies the number of blocks to be read. n = 0 means ten blocks.

v: variation designator:

(rC:41)/8 = 1: B-register address-modification of designated input will occur.

(rC:41)/1 = 1: control blocks are recognized as normal blocks.

aaaa: address of base of location in which is written the first word read from magnetic tape.

Description of Operation. Read n blocks from unit u. The lane read from is that specified by the last instruction referring to a lane on that unit: MTS, MFS, MRW, MDA, MLS, MTC, MFC. Words read from tape are stored into consecutively addressed storage locations beginning with B[aaaa].

Words read from tape are assembled in the D register before being sent to storage: if (rC:41)/8 = 1 and (rD:±1)/8 = 1, the word will be B-register address-modified as it goes from the D register to storage; at the same time, (rD:±1)/8 is set to 0.

If v is *even* a control block is recognized as such. If a control block is encountered during the execution of MRD, it will be sensed and the following events will occur:

1. Each word of the control block, except the last, will be read and written in the storage location designated.
2. The last word of the block read from tape—the control word—remains in the Data Processor's control section to provide information with which to terminate the operation. The control word has the following format: ± ii aaaa bbbb.
   a. The contents of the P register are preserved: (rP) → aaaa:04. This enables the program to determine the location of the instruction during whose execution the control block was encountered.

b. The contents of the address part of the C register are preserved: (rC:04) → aaaa:64. This enables the program to determine the location of the last word read from magnetic tape.

c. The address of the location of the next instruction selected for execution—an entry to the control routine—is transferred to the P register: B[bbbb] → rP.

3. After reading the control block, MRD is terminated.

If v is *odd* control blocks are not recognized as such. Instead, they are treated as normal blocks. Sign digits of 7 will be retained; the control word will remain intact.

An end-of-tape block is recognized as such. If an end-of-tape block is encountered during the execution of MRD, it will be sensed, and the following events will occur:

1. The control word is read from tape but remains in the Data Processor's control section to provide information with which to terminate the operation. The control word has the same format as the control word in a control block. The transfer of the remainder of the end-of-tape block from tape to core storage is inhibited. The events described under 2(a), 2(b), and 2(c), above, take place here also, but "end-of-tape" should be substituted for "control" wherever it occurs.

After reading the end-of-tape block, MRD is terminated.

After termination of MRD, tape is positioned so that the head can read the block following the one which caused the operation to terminate.

Exceptional Conditions.

1. Storage ALARM STOP caused by a nonexistent address.

2. Magnetic-Tape ALARM STOP caused by parity error, end-of-tape error, digit-count/word-count error, or unit not ready.

Remarks.

1. A description of block layout can be found on page 4-3; see also the description of MAGNETIC-TAPE INITIAL WRITE (page 4-18) where the assignment of storage locations is noted, as well.

2. This instruction does not read the preface word associated with each block. If the preface must be read, a MAGNETIC-TAPE READ, RECORD instruction must be executed.

3. Note that a MRD instruction may terminate without having read n blocks as specified.

4. Blank tape is treated as inter-block gap.

5. If an end-of-tape block is also identified as a control block, by having a 7 in the sign-digit position of the control word, the block will be recognized as end-of-tape. When the end-of-tape block is read, however, B-register address-modification of the control word will occur because its sign digit is odd.

6. During the execution of MRD, the timing character preceding each word written on tape is used to verify that 11 digits are read for each word; the preface and the end-of-block character are used to verify that the correct number of words has been read. Should there be a failure in either the digit count or the word count during MRD, two additional attempts to eliminate the failure will be made, automatically, under the direction of the Magnetic-Tape Control Unit. If neither of the additional attempts succeeds, a Magnetic-Tape ALARM STOP will occur with the tape positioned so that the read-write head can read the block in which the failure occurred.

7. After a Magnetic-Tape ALARM STOP caused by a parity error, the read-write head is in position to read the block in which the error was detected. The contents of the block will have been read from tape and written in storage, however.

8. The variation designator of the MRD instruction must be odd to read a 205 tape mounted on a 220 unit when using the 205/220 Magnetic-Tape Converter.

MAGNETIC-TAPE READ, RECORD (MRR)

Operation Code. 53

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	u	n	i	v	0	p	a	a	a	a

Definitions.

u: designates magnetic-tape unit.

n: specifies the number of blocks to be read. n = 0 means 10 blocks.

v: variation designator:

(rC:41)/8 = 1: B-register address-modification of designated input will occur.

(rC:41)/1 = 1: control blocks recognized as normal blocks.

aaaa: address of base of location in which will be written the preface associated with the first block read from tape.

Description of Operation. Read n blocks from unit u. The lane read from is that specified by the last instruction referring to a lane on that unit: MTS, MFS, MRW, MDA, MLS, MTC, MFC. Words read from tape are stored into

consecutively addressed storage locations, the preface associated with the first block going into B[aaaa].

Words read from tape are assembled in the D register before being sent to storage: if (rC:41)/8 = 1, and (rD:±1)/8 = 1, the word will be B-register address-modified as it goes from the D register to storage; at the same time, (rD:±1)/8 is set to 0.

If v is *even* a control block is recognized as such: if a control block is encountered during the execution of MRR, it will be sensed, and the following events will occur:

1. The preface and each word of the control block, except the last, will be read and written in the storage location designated.
2. The last word of the control block read from tape—the control word—remains in the Data Processor's control section to provide information with which to terminate the operation. The control word has the following format: ± ii aaaa bbbb.
   a. The contents of the P register are preserved: (rP) → aaaa:04. This enables the program to determine the location of the instruction during whose execution the control block was encountered.
   b. The contents of the address part of the C register are preserved: (rC:04) → aaaa:64. This enables the program to determine the location of the last word read from magnetic tape.
   c. The address of the location of the next instruction selected for execution—an entry to the control routine—is transferred to the P register: B[bbbb] → rP.
3. After reading the control block MRR is terminated.

If v is *odd* control blocks are not recognized as such. Instead they are treated as normal blocks. Sign digits of 7 will be retained; the control word will remain intact.

An end-of-tape block is recognized as such: if an end-of-tape block is encountered during the execution of MRR, it will be sensed, and the following will occur:

1. The control word is read from tape but remains in the Data Processor's control section to provide information with which to terminate the operation. This control word has the same format as the control word in a control block. The transfer of the remainder of the end-of-tape block from tape to core storage is inhibited. The events described under 2(a), 2(b), and 2(c), above, take place here also, but "end-of-tape" should be substituted for "control" wherever it occurs.

After reading the end-of-tape block, MRR is terminated.

After termination of MRR, tape is positioned so that the head can read the block following the one which caused the operation to terminate.

Exceptional Conditions.

1. Storage ALARM STOP caused by a nonexistent address.
2. Magnetic-Tape ALARM STOP caused by parity error, end-of-tape error, digit-count/word-count error, or unit not ready.

Remarks.

1. During MRR, the preface associated with each block is read from tape and stored in core storage. Otherwise, MRR and MRD are executed in identical fashion.
2. Suppose that a preface is kk and the preface is written in the location whose address is L. Then the first data word of the block will be in L + 1 and the last data word in L + kk.

For example, MRR says, "read three blocks; preface of first block to be written in 0984." The first two blocks are 25 and 30 words long, respectively; the third block is a control block and is 40 words long. The allocation of storage is as shown:

Location	Contents	Remarks
0984	0 2500 00 0000	Preface of first block
0985	1st word	First block
•	•	
•	•	
•	•	
1009	25th word	
1010	0 3000 00 0000	Preface of second block
1011	1st word	Second block
•	•	
•	•	
•	•	
1040	30th word	
1041	7 4000 00 0000	Preface and control-block marker
1042	1st word	Control block
•	•	
•	•	
•	•	
1080	39th word	

3. Note that a MRR instruction may terminate without having read n blocks as specified.

4. Blank tape is treated as if it were inter-block gap.

5. If an end-of-tape block is also identified as a control block, by having a 7 in the sign-digit position of the control word, the block will be recognized as end-of-tape. When the end-of-tape block is read, however, B-register address-modification of the control word will occur because its sign digit is odd.

6. During the execution of MRR, the timing character preceding each word written on tape is used to verify that 11 digits are read for each word; the preface and the end-of-block character are used to verify that the correct number of words has been read. Should there be a failure in either the digit count, or the word count during MRR, two additional attempts to eliminate the failure will be made, automatically, under the direction of the Magnetic-Tape Control Unit. If neither of the additional attempts succeeds, a Magnetic-Tape ALARM STOP will occur with the tape positioned so that the read-write head can read the block in which the failure occurred.

7. After a Magnetic-Tape ALARM STOP caused by a parity error, the read-write head is in position to read the block in which the error was detected. The contents of the block will have been read from tape and written in storage, however.

8. The variation designator of the MRR instruction must be odd to read a 205 tape mounted on a 220 unit when using the 205/220 Magnetic-Tape Converter.

MAGNETIC-TAPE INITIAL WRITE (MIW)

Operation Code. 54

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	n	k	k	O	p	a	a	a	a

Definitions.

u: designates magnetic-tape unit.

n: specifies number of blocks to be written. n = 0 means ten blocks.

kk: specifies number of words in each of the n blocks. kk = 00 means 100 words.

aaaa: address of base of location from which is read the first word of the first block to be written.

Description of Operation. Write—on newly edited tape—n blocks on unit u; each block is kk words long: the minimum-length block is ten words; the maximum-length block is 100 words. The lane written on is that specified by the last instruction referring to a lane on that unit: MTS, MFS, MRW, MDA, MLS, MTC, MFC. Words written on tape are taken from consecutively addressed locations beginning with B[aaaa]. This instruction causes the preface to be written immediately preceding the first word of the block with which the preface is associated.

If magnetic end-of-tape is encountered during the execution of MIW, the magnetic end-of-tape condition will be established at the completion of the operation.

Exceptional Conditions.

1. Storage ALARM STOP caused by a nonexistent address.

2. Magnetic-Tape ALARM STOP caused by improper block length, end-of-tape error, or unit in not-ready or not-write status.

Remarks.

1. MIW recognizes magnetic end-of-tape. Therefore, every MIW instruction must either be preceded or followed by a MAGNETIC-TAPE INTERROGATE END-OF-TAPE, BRANCH (MIE) instruction to sense the magnetic end-of-tape condition. The MIE instruction provides the branch to the end-of-tape control procedure (see page 4-22). The MIW operation must be completed before the MIE instruction is initiated. A MIB must be inserted before the MIE to insure this (see example, page 4-7).

2. After termination of a MIW instruction during which magnetic end-of-tape was encountered, tape is positioned at physical end. This fact is sensed by the MIE instruction which follows every MIW instruction. After execution of the MIE instruction, if the next instruction referring to the tape unit is one which causes tape movement, it *must* be one which causes tape to move initially in the backward direction; otherwise an ALARM STOP will occur. (Permissible instructions are MRW, MDA, MPB.)

   Normally, after encountering magnetic end-of-tape, terminal control blocks must be written to bound the file just written.

3. MIW is one of two magnetic-tape operations which recognize magnetic end-of-tape. The other operation is MAGNETIC-TAPE INITIAL WRITE, RECORD.

4. It is advisable to verify, under program control, that newly edited tape is being used. If unedited tape is used with MIW, or MIR, a Magnetic-Tape ALARM STOP will occur.

5. If the unit referred to by this instruction is in not-write status, a Magnetic-Tape ALARM STOP will occur.

6. To illustrate the allocation of storage, suppose a MIW instruction specifies that three blocks are to be written, each 18 words long; the first word of the first block is in location 5001.

Location	Contents	Remarks
5001	1st word	
•	•	
•	•	First block
•	•	
5018	18th word	
5019	1st word	
•	•	
•	•	Second block
•	•	
5036	18th word	
5037	1st word	
•	•	
•	•	Third block
•	•	
5054	18th word	

7. The format of a block as it appears on tape is shown in Figure 4-1.
8. The MIW instruction that writes the first block on newly edited tape should be preceded by a MRW and a MPE instruction to place the write head in position to record the first block.

MAGNETIC-TAPE INITIAL WRITE, RECORD (MIR)

Operation Code. 55

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	n	i	i	O	p	a	a	a	a

Definitions.

u: designates magnetic-tape unit.

n: specifies the number of blocks to be written. n = 0 means ten blocks.

aaaa: address of base of location from which is read the first word to be written on tape. This word contains the preface.

Description of Operation. Write—on newly-edited tape—n blocks on unit u. The lane written on is that specified by the last instruction referring to a lane on that unit: MTS, MFS, MRW, MDA, MLS, MTC, MFC. Words written on tape are taken from consecutively addressed locations beginning with B[aaaa].

The first word written is a preface word: (B[aaaa]:22) are interpreted as the preface. This preface specifies how many words in the consecutively addressed locations following B[aaaa] are in the first block. If more than one block is to be written, the first word following the last word of the previous block written will be interpreted as containing the preface of the next block to be written.

The minimum-length block is ten words; the maximum, 100 words.

If magnetic end-of-tape is encountered during the execution of MIR, the magnetic end-of-tape condition will be established. The instruction will be completed, however.

Exceptional Conditions.

1. Storage ALARM STOP caused by a nonexistent address.
2. Magnetic-Tape ALARM STOP caused by improper block length, end-of-tape error, or unit in not-ready or not-write status.

Remarks.

1. MIR recognizes magnetic end-of-tape. Therefore, every MIR instruction must either be preceded or followed by a MIE instruction to sense the magnetic end-of-tape condition. The MIE instruction provides the branch to the end-of-tape control procedure (see page 4-22). The MIR operation must be completed before initiating the MIE instruction. A MIB must be inserted before the MIE to insure this. (See example on page 4-7.)
2. After termination of a MIR instruction during which magnetic end-of-tape was encountered, tape is positioned at physical end. This fact is sensed by the MIE instruction which follows every MIR instruction. After execution of the MIE instruction, if the next instruction referring to the tape unit is one which causes tape movement, it *must* be one which causes tape to move initially in the backward direction; otherwise, a Magnetic-Tape ALARM STOP will occur. (Permissible instructions are MRW, MDA, MPB.)

   Normally after encountering magnetic end-of-tape, terminal control blocks must be written to bound the file just written.
3. Instruction formats for MIW and MIR are very much alike: the formats differ in digit positions 3 and 4 which are used to specify the preface in MIW and are noted as not relevant for MIR. However, even in a MIR instruction, digit positions 3 and 4 must not contain 02, 03, 04, 05, 06, 07, 08, or 09: if one of these configurations does appear, a Magnetic-Tape ALARM STOP will occur, with the read-write head positioned at its starting position.
4. MIR is one of two magnetic-tape operations which recognize the magnetic end-of-tape. The other operation is MAGNETIC-TAPE INITIAL WRITE.

5. It is advisable to verify, under program control, that newly-edited tape is being used. If unedited tape is used with MIR or MIW, a Magnetic-Tape ALARM STOP will occur.
6. If the unit referred to by this instruction is in not-write status, a Magnetic-Tape ALARM STOP will occur.
7. To illustrate the allocation of storage, suppose a MIR instruction specifies that three blocks are to be written, 15, 27, and 12 words long, in that order; the preface associated with the first block is in location 5001.

Location	Contents	Remarks
5001	0 1500 00 0000	Preface of first block
5002	1st word	First block
•	•	
•	•	
•	•	
5016	15th word	
5017	0 2700 00 0000	Preface of second block
5018	1st word	Second block
•	•	
•	•	
•	•	
5044	27th word	
5045	0 1200 00 0000	Preface of third block
5046	1st word	Third block
•	•	
•	•	
•	•	
5057	12th word	

8. The format of a block as it appears on tape is shown in Figure 4-1.

MAGNETIC-TAPE OVERWRITE (MOW)

Operation Code. 56

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	n	k	k	0	p	a	a	a	a

Definitions.

u: designates magnetic-tape unit.

n: specifies the number of blocks to be written. n = 0 means ten blocks.

kk: specifies number of words in each of the n blocks. kk = 00 means 100 words. Minimum number of words is 10; maximum is 100.

aaaa: address of base of location from which is read the first word of the first block to be written.

Description of Operation. Write n blocks on unit u; each block contains kk words; the lane written on is that specified by the last instruction referring to a lane on that unit: MTS, MFS, MRW, MDA, MLS, MTC, MFC. Words written on tape are taken from consecutively addressed locations beginning with B[aaaa].

Before each block is overwritten, its preface (already on tape) is compared with kk: the block is written if and only if the preface and kk are identical; otherwise—except when the block on tape is an end-of-tape block—a Magnetic-Tape ALARM STOP, due to this preface mismatch, occurs, with the read-write head positioned to read the block causing the ALARM STOP.

If a preface mismatch occurs with an end-of-tape block on tape, the ALARM STOP will not occur. Instead, the end-of-tape block is sensed, the control word is read from tape into the Data Processor control section, and terminating procedures are begun:[4]

1. The contents of the P register are preserved: (rP) → aaaa:04. This enables the program to determine the location of the instruction during whose execution the end-of-tape block was encountered.
2. The contents of the address part of the C register are preserved: (rC:04) → aaaa:64. This enables the program to determine the location in core storage of the last word written on tape.
3. The address of the location of the next instruction selected for execution—an entry to the end-of-tape routine—is transferred to the P register: B[bbbb] → rP.

After reading the end-of-tape block, MOW is terminated.

After termination of MOW, tape is positioned so that the head can write the block following the one which caused termination.

Exceptional Conditions.

1. Storage ALARM STOP caused by a nonexistent address.
2. Magnetic-Tape ALARM STOP caused by parity error, preface mismatch, end-of-tape error, or unit in not-ready or not-write status.

Remarks.

1. If the unit referred to by this instruction is in not-write status, a Magnetic-Tape ALARM STOP will occur.

[4] Recall that the format of the control word is ± ii aaaa bbbb.

2. The allocation of storage is as depicted for MIW.
3. Control blocks are not recognized as such during the execution of a MOW instruction.
4. Blank tape is treated as inter-block gap.
5. Note that a MOW instruction may terminate without having written n blocks as specified.
6. This instruction will not recognize magnetic end-of-tape.

MAGNETIC-TAPE OVERWRITE, RECORD (MOR)

Operation Code. 57

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	n	i	i	O	p	a	a	a	a

Definitions.

u: designates magnetic-tape unit.

n: specifies the number of blocks to be written. n = 0 means ten blocks.

aaaa: address of base of location from which is read the first word to be written on tape. This word contains the preface.

Description of Operation. Write n blocks on unit u; the lane written on is that specified by the last instruction referring to a lane on that unit: MTS, MFS, MRW, MDA, MLS, MTC, MFC. Words written on tape are read from consecutively addressed storage locations beginning with B[aaaa], except as noted below.

In storage the preface associated with each block is in the location which immediately precedes the location of the first word of the block (see description of storage allocation under MIR, page 4-19). For example, (B[aaaa]:22) is the preface associated with the block whose first word is in B[aaaa] + 1. Before each block is overwritten, its preface (already on tape) is compared with the preface read from storage: the block is overwritten only if the two prefaces are identical; otherwise—except when the block on tape is an end-of-tape block—a magnetic-tape ALARM STOP occurs, with the read-write head positioned to read the block causing the ALARM STOP.

If a preface mismatch occurs with an end-of-tape block on tape, the ALARM STOP will not occur. Instead, the end-of-tape block is sensed, the control word is read from tape into the Data Processor control section, and terminating procedures are begun:[5]

1. The contents of the P register are preserved: (rP) → aaaa:04. This enables the program to determine the location of the instruction during whose execution the end-of-tape block was encountered.
2. The contents of the address part of the C register are preserved (rC:04) → aaaa:64. This enables the program to determine the location in core storage of the last word written on tape.
3. The address of the location of the next instruction selected for execution—an entry to the end-of-tape routine—is transferred to the P register: B[bbbb] → rP.

After reading the end-of-tape block, MOR is terminated. After termination of MOR, tape is positioned so that the head can write the block following the one which caused termination.

Exceptional Conditions.

1. Storage ALARM STOP caused by a nonexistent address.
2. Magnetic-Tape ALARM STOP caused by parity error, preface mismatch, end-of-tape error, or unit in a not-ready or not-write status.

Remarks.

1. Instruction formats for MOW and MOR are very much alike: the formats differ in digit positions 3 and 4 which are used to specify the preface in MOW and are noted as not relevant for MOR. However, even in a MOR instruction, digit positions 3 and 4 must *not* contain 02, 03, 04, 05, 06, 07, 08, or 09: if one of these configurations does appear, a Magnetic-Tape ALARM STOP will occur, with the read-write head being positioned at its starting position.
2. If the unit referred to by this instruction is in not-write status, a Magnetic-Tape ALARM STOP will occur.
3. The allocation of storage is as depicted for MIR.
4. Control blocks are not recognized as such during the execution of a MOR instruction.
5. Blank tape—that is, tape on which no information is recorded—is treated as inter-block gap.
6. Note that a MOR instruction may terminate without having written n blocks as specified.
7. This instruction will not recognize magnetic end-of-tape.

MAGNETIC-TAPE POSITION FORWARD (MPF)

MAGNETIC-TAPE POSITION BACKWARD (MPB)

MAGNETIC-TAPE POSITION AT END OF INFORMATION (MPE)

Operation Code. 58

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	n	i	v	O	p	i	i	i	i

[5] Ibid

Definitions.

u: designates magnetic-tape unit.

n: specifies the number of blocks to be passed. n = 0 means ten blocks.

v: variation designator:

v = 0: MAGNETIC-TAPE POSITION FORWARD will be executed.

v = 1: MAGNETIC-TAPE POSITION BACKWARD will be executed.

v = 2: MAGNETIC-TAPE POSITION AT END OF INFORMATION will be executed.

Description of Operation.

v = 0: MAGNETIC-TAPE POSITION FORWARD will be executed. Move tape on unit u in the forward direction until n blocks have been passed. The lane in which counting is done is the lane referred to by the last instruction referring to a lane on that unit: MTS, MFS, MRW, MDA, MLS, MTC, MFC.

Control blocks and end-of-tape blocks are counted and passed without recognition of their control function.

After completion of MPF, the tape is positioned so that the read-write head can read or overwrite the block following the last block passed.

v = 1: MAGNETIC-TAPE POSITION BACKWARD will be executed. Except that tape is moved in the opposite direction, this variation is identical with that specified by v = 0.

After completion of MPB, the tape is positioned to permit reading the last block passed.

v = 2: MAGNETIC-TAPE POSITION AT END OF INFORMATION will be executed.

Tape movement begins in the forward direction, and continues until a blank area longer than inter-block gap is sensed, at which time the direction of tape movement is reversed. Tape is then moved in the backward direction until a block is sensed, at which time tape movement stops with the head in position to initial write a block following the one which caused the operation to terminate.

Exceptional Conditions.

1. Magnetic-Tape ALARM STOP caused by end-of-tape error or unit not ready.

Remarks.

1. If v ≠ 1 or 2, the MAGNETIC-TAPE POSITION FORWARD variation will be executed.
2. Both MPB and MPF treat blank tape as inter-block gap. MPE, on the other hand, recognizes blank tape uniquely as described above.
3. For purposes of counting blocks during execution of a MPF or MPB instruction, preface words are used to indicate that a block is being passed.
4. If the conductive leader which marks the physical end-of-tape is encountered during the execution of a MPF instruction, no "operation complete" signal will be generated. So, the Magnetic-Tape Control Unit will appear to be busy if a MIB instruction is issued or if an attempt is made to use the tape unit (by attempting to execute a MRD instruction, for example). Manual intervention is required in order to proceed with processing.

Similar remarks apply to the situation which results when the conductive leader which marks the physical beginning-of-tape is encountered during the execution of a MPB instruction.

MAGNETIC-TAPE INTERROGATE, BRANCH (MIB)

MAGNETIC-TAPE INTERROGATE END-OF-TAPE, BRANCH (MIE)

Operation Code. 59

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	i	i	v	0	p	a	a	a	a

Definitions.

u: designates magnetic-tape unit.

v: variation designator:

v = 0: MAGNETIC-TAPE INTERROGATE, BRANCH will be executed.

v = 1: MAGNETIC-TAPE INTERROGATE END OF TAPE, BRANCH will be executed.

aaaa: address of base of location of alternate instruction.

Description of Operation.

v = 0: MAGNETIC-TAPE INTERROGATE, BRANCH will be executed.

If unit u is ready, transfer control to B[aaaa], otherwise control continues in sequence.

v = 1: MAGNETIC-TAPE INTERROGATE END OF TAPE, BRANCH will be executed.

If the magnetic end-of-tape condition is established, transfer control to B[aaaa]; otherwise, control continues in sequence.

Remarks.

1. If v ≠ 1, the MAGNETIC-TAPE INTERROGATE, BRANCH variation will be executed.

2. Because operations independent of both tape-control unit and Data Processor (e.g., MRW, MDA) as well as operations independent only of the Data Processor (e.g., MTS) can be in process, it is necessary to distinguish four cases for MIB. These are shown in the table below.

Status of Tape-Control Unit	Status of Tape-Storage Unit	Branch ?
Ready	Ready	Yes
Ready	Not ready	No
Not ready	Ready	No
Not ready	Not ready	No

3. If the unit designated by MIB and MIE instructions is not ready or is not on line, the program will continue in sequence.

4. If control is transferred to B[aaaa] upon execution of the MIE instruction (i.e., if end-of-tape condition is established) the next instruction referring to that tape unit must be one which causes tape to move initially in the backward direction (MRW, MDA, MPB).

5. To write an end-of-tape block once the end-of-tape condition has been established:

Control	*Op*	*Address*
1101	MPB	0000
1002	MPE	0000
1101	MIW	aaaa

5

The Paper-Tape System

GENERAL

Paper-tape input and output facilities are provided by means of Photoreaders and punches. Interchangeable with the latter are character-at-a-time printers (physically and functionally these printers are identical with the Supervisory Printer). Each of the printers is capable of being driven off-line by a mechanical reader which may be attached to it.

THE PHOTOREADER

The paper-tape reader used by the BURROUGHS 220 reads seven-channel single-frame paper. The code is shown in Appendix C. The code is of the odd-parity type: provision is made for the detection of parity errors; in case a parity error is detected a Paper-Tape ALARM STOP will occur.

When reading in the numeric mode, the appearance of a code for a non-numeric character—the letter "A," for example—will be detected as an error and a Paper-Tape ALARM STOP will occur.

The Photoreader utilizes two sizes of single-flange plastic reels: a 5½-inch reel which holds 350 feet of tape, and a 7-inch reel which holds 700 feet of tape. Tape is moved at the rate of either 50 or 100 inches per second depending upon whether the SPEED switch is set to HI or LOW. Since information is punched ten characters per inch, the transfer rate is either 500 or 1,000 characters per second.

Start time is less than five milliseconds. The reader will stop on the stop character; that is, it will stop with the last character read still under the reading head.

As many as ten Photoreaders may be included in a BURROUGHS 220 system. Selection of a particular reader is under program control. In case the reader called for is inoperative, a Paper-Tape ALARM STOP will occur.

WORD FORMAT

Provision is made in the order structure for reading information with the sign digit punched first—this is normal format—as well as information with the sign digit punched last—this is inverse format.

A word, as it is punched on paper tape, is defined to be the information bounded by two end-of-word characters (except for the first word, which does not have an end-of-word character preceding it). As each character is read from paper tape it is transferred to rD:01, shifting the contents of the D register one place to the left and causing the contents of rD:±1 to be lost, i.e., the sign digit is shifted out of the D register.

In normal format, the sign digit is the first character of the word on paper tape; in inverse format, the sign digit is the character immediately preceding the end-of-word character.

Alphanumeric words are distinguished both on paper tape and in the Data Processor by having the symbol "2" punched as the sign digit. The recognition of this character causes automatic translation of the single-frame alphabetic characters in the word into the two-decimal-digit 220 code.[1] The sign digit is read and translated as a "2," (the digit 2 being used internally by the BURROUGHS 220 to flag alphanumeric words).

When reading inverse format, however, alphanumeric translation is not possible, because the sign digit is the last character sensed as the word is read. After the word in inverse format has been read, the contents of the D register are permuted so that the sign of the word is in its proper position. The sign digit is then checked; if it is 2—indicating that the word is supposed to be alphanumeric—a Paper-Tape ALARM STOP will occur.

The appearance of any digit, except 2, in the sign-digit position of a word on paper tape will cause the word to be translated numerically. Certain of the numeric digits, namely 6, 7, 8, and 9, in the sign-digit position are used for control purposes. Generally speaking, a 6 or 7 indicates that the word is to go to the C register, instead of to storage, without or with B-register address-modification, respectively; an 8 or 9 in the sign-digit position indicates that the word is to be B-register address-modified as it goes to storage. It is possible, it should be noted, to read from paper tape a word with any digit in the sign position, and to store that word exactly as it appeared on tape.

[1] Of course, when alphanumeric words are read and translated, the contents of the D register are shifted twice for each character read, once when the "zone" part of the character is sent to rD:01 from the decoding matrix, and second when the "numeric" part is sent to rD:01.

The reader is referred to the Execute Phase description of each input instruction for an *explicit* and *detailed* specification of sign-digit control. Except as noted in those specifications, the sign digits are read and translated exactly as they are punched in tape.

INSTRUCTIONS

There are two instructions for reading information in normal format. The first of these, PAPER-TAPE READ (see page 5-4), specifies that a fixed number of words (from 1 to 100) shall be read. However, the instruction may be coded to permit overriding of this specification: when so coded, the instruction says, in effect, "Read nn words, unless a control word is encountered, in which case terminate the reading operation."

The second instruction for reading information in normal format, PAPER-TAPE READ, BRANCH (see page 5-6), causes information to be read until a control word is encountered. The control word causes the reading operation to terminate.

There is one instruction for reading information in inverse format, namely, PAPER-TAPE READ, INVERSE FORMAT (see page 5-8). This instruction specifies that a fixed number of words—from 1 to 100—shall be read unless a control word is encountered. The control word causes the reading operation to terminate. With this instruction it is *not* possible to read into storage a word with a 2, 6, or 7 in the sign-digit position. (The reader is reminded that a 2 in the sign-digit position causes a Paper-Tape ALARM STOP.)

The control word mentioned in the preceding paragraphs is a word flagged with a 6 or 7 in the sign-digit position. The detection of such a word terminates the instruction being executed, disconnects the reader, and causes preparations to be made for transferring the word to the C register for execution. The reading operation stops with the control word in the information buffer register, after which control is transferred to the Fetch Phase circuits. During the Fetch Phase the control word is taken from the IB register to the C register—without B-register address-modification if the sign digit is 6, with such modification if the sign digit is 7—where it is regarded as an instruction. This is to say that a control word *must* be an instruction.

THE PAPER-TAPE PUNCH

The paper-tape punch used by the BURROUGHS 220 punches seven-channel single-frame tape at the rate of 60 characters per second. That is to say, it punches tape which can be used for input to a BURROUGHS 220 system. However, it punches tape in normal format, automatically supplying the end-of-word character after the last character has been punched.

As many as ten paper-tape punches may be included in a BURROUGHS 220 system. Selection of a particular punch is under program control. In case the punch called for is inoperative, a Paper-Tape ALARM STOP will occur.

Alphanumeric words—which are flagged internally with a 2 in the sign-digit position—are translated automatically to the single-frame code for punching in tape. The digit "2"—which also identifies the word in tape as alphanumeric—is punched as a "2."

Provision exists for the suppression of alphanumeric translation, in which case the contents of the word are punched as 11 decimal digits.

There is also provision for the suppression of high-order zeros. This feature is provided by the SUPPRESS LEADING ZEROS switch on the Control Console. When the SUPPRESS LEADING ZEROS switch is "on," the punching of zeros preceding the first non-zero digit in the word will be suppressed. This means that no zeros will be suppressed if the sign digit is different from 0.

THE PRINTER

In place of any paper-tape punch there may be substituted a character-at-a-time printer. Naturally, this printer is logically equivalent to the punch which it replaces. The printer is physically and functionally identical with the Supervisory Printer described in Chapter 3.

THE MECHANICAL READER

Any printer—including the Supervisory Printer—in a BURROUGHS 220 Paper-Tape System may have attached to it a mechanical reader which is capable of reading BURROUGHS 220 paper tape. The reader provides off-line facilities for printing the contents of paper tape at the printer's rate of 10 characters per second.

Except for its ability to check for parity errors, the mechanical reader and its circuits are logically equivalent to the Data Processor's circuits when the printer is driven on-line.

INFORMATION FLOW

The general nature of information flow during input and output was discussed in Chapter 2. In some details, however, the actual flow of information is different from that indicated in Chapter 2. For this reason, the flow of information is described again below.

INPUT FLOW

Input flow is shown in Figure 5-1.

1. The address part of the C register—which specifies the location in storage where the next word read

from paper tape is to be stored—is transferred to the E register. At the same time, the address part of the C register is counted up 1.

2. The word is read from paper tape through the translator and into the D register.

3. a. If the instruction is PRI, the contents of the D register are permuted so that the sign is in its normal position.

   b. The contents of the D register are then sent to the information buffer register, with or without B-register address-modification as specified, after which the D register is cleared.

4. a. If the word is to go to storage, the contents of the information buffer register are stored in the location whose address is in the E register.

   b. If the word is to go to the C register, the reader is disconnected, and preparations are made to enter the Fetch Phase, during which the contents of the information buffer register are transferred, with or without B-register address-modification as specified, to the C register, where the word is regarded as an instruction.

If the instruction has not been terminated, there is a return to step 1 for the next word.

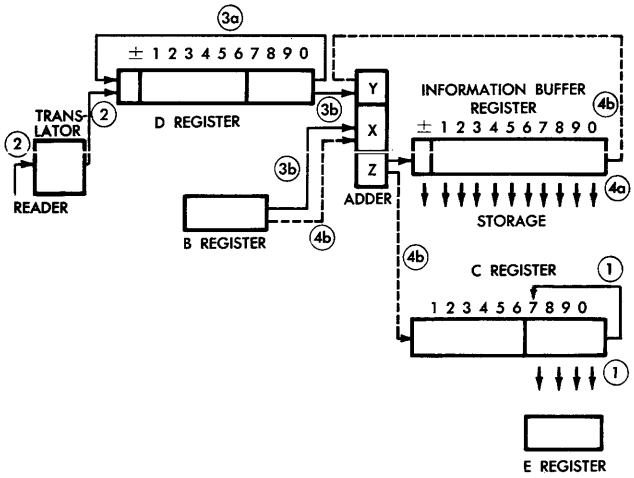


Figure 5-1. Input flow.

OUTPUT FLOW

Output flow is shown in Figure 5-2.

1. The address part of the C register—which specifies the location in storage from which the next word to be written will be taken—is transferred to the E register. At the same time, the address part of the C register is counted up 1.

2. The word is transferred from storage to the information buffer register.

3. The contents of the information buffer register are transferred to the D register.

4. a. The contents of the D register are permuted so that the (rD:±1) appear in rD:01.

   b. The contents of rD:01 pass through the adder to the translator and thence to the printer or punch.

4a. and 4b. are repeated until the entire word has been printed or punched.

If the instruction has not terminated, there is a return to step 1 for the next word.

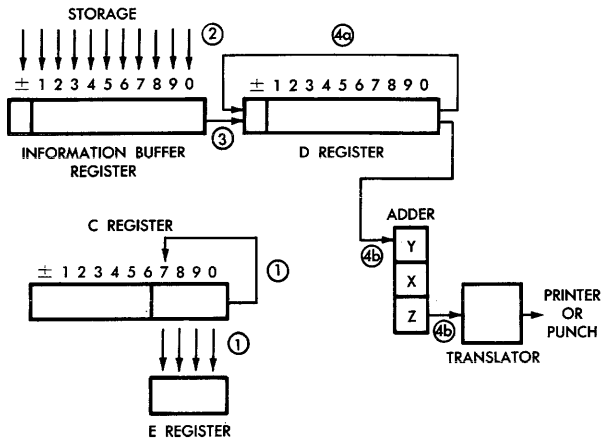


Figure 5-2. Output flow.

EXCEPTIONAL CONDITIONS

Following is a list defining exceptional conditions which can occur and are detected. In addition to those enumerated below, a Storage ALARM STOP, due to use of a nonexistent address, can occur.

1. The seven-channel code uses an odd-parity check. Whenever a character is sensed which has an even number of channels punched, the fact is signaled by a Paper-Tape ALARM STOP.

2. The detection of a code for a non-numeric character when reading in the numeric mode is signaled by a Paper-Tape ALARM STOP.

3. When executing PAPER-TAPE READ, INVERSE FORMAT, the appearance in the sign-digit position of a "2" is detected and a Paper-Tape ALARM STOP occurs.

4. If the input or output unit designated by the instruction is inoperative, a Paper-Tape ALARM STOP will occur.

THE EXECUTE PHASE

The remainder of the Paper-Tape System chapter is devoted to a discussion of the instructions involving input and output on paper tape.

PAPER-TAPE READ (PRD)

Operation Code. 03

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	u	n	n	v	0	p	a	a	a	a

Definitions.

u: designates paper-tape reader unit.

nn: specifies the number of words to be read. nn = 00 means 100 words.

v: variation designator:

- v = 1 specified word goes to C register for execution; reader is turned off.
- v = 8, 9: specified input will be B-register address-modified.

aaaa: address of base of location into which is written first word read from paper tape.

Description of Operation.

v ≠ 1: Read nn words from unit u into consecutively addressed locations beginning with B[aaaa]. If v = 8 or 9, designated input will be B-register address-modified.

v = 1: Read nn words, or until a word with sign digit equal to 6 or 7 is encountered, from unit u into consecutively addressed locations beginning with B[aaaa]. An input word with sign digit equal to 6 or 7 goes to the C register, without or with B-register address-modification, respectively, for immediate execution; the reader is turned off.

Input Sign-Control:

If v = 1 and:

1. If the input sign digit is 6, the word goes to rC for immediate execution; the reader is turned off.
2. If the input sign digit is 7, the word goes to rC, with B-register address-modification, for immediate execution; the reader is turned off.

If v is equal to 8 or 9 and:

1. If the input sign digit is 8, the word is B-register address-modified; the sign goes into the specified storage location as 0.
2. If the input sign digit is 9, the word is B-register address-modified; the sign goes into the specified storage location as 1.

Flow Chart. See page 5-5.

Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. Paper-Tape ALARM STOP due to a parity error, detection of an inadmissible character, or the designated unit in not-ready status.

Remarks.

v \ s	0	1	2	3	4	5	6	7	8	9
0			A							
1			A				C	C B		
2			A							
3			A							
4			A							
5			A							
6			A							
7			A							
8			A						B_0	B_1
9			A						B_0	B_1

s = Sign of input word. (Read across.)

v = Variant digit of PRD instruction. (Read down.)

A: Convert each character in the word to a two-digit (alphanumeric) code.

B: Prior to storing, B-modify the input word.

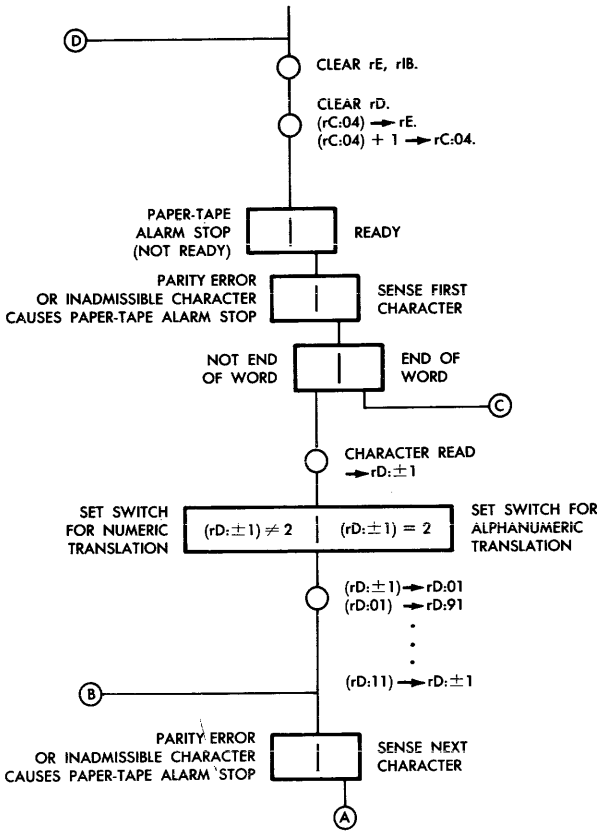
B_i: Same as B. Sign of input is equal to i, where i = 0 or 1.

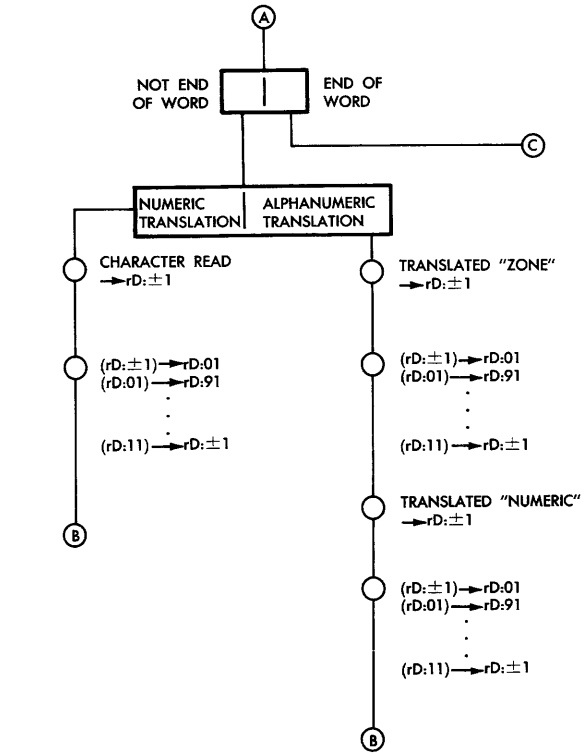
C: Turn off Photoreader. Transfer input to C register and execute.

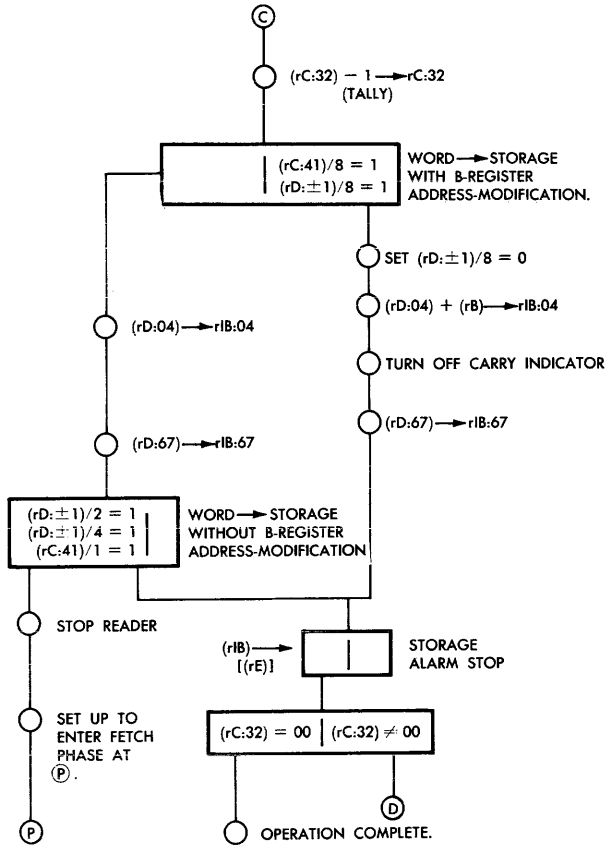
Register Status.

Register name	Contents after execution of PRD; no override of nn
A	Unchanged
R	Unchanged
D	Last word read
B	Unchanged
P	$(rP)_b + 1$
C	u | 0 0 | v | 0 3 | B[aaaa] + nn*
E	B[aaaa] + nn − 1*

* If nn = 00, add 100.







Register name	Contents after execution with override of nn
A	Unchanged
R	Unchanged
D	Last word read
B	Unchanged
P	$(rP)_b + 1$
C	u | * | * | v | 0 | 3 | * | * | * | *
E	* * *

* *, * * *, * * * * : See description of operation.

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	Last word read
B	Unchanged

Register name	Contents if Storage ALARM STOP occurs
P	$(rP)_b + 1$
C	u | * | * | v | 0 | 3 | * | * | * | *
E	Address causing ALARM STOP

** nn minus number of words read.

**** Sum: address causing ALARM STOP + 1.

PAPER-TAPE READ, BRANCH (PRB)

Operation Code. 04

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	i	i	v	O	p	a	a	a	a

Definitions.

u: designates paper-tape reader unit.

v: variation designator:

v = 8, 9: specified input will be B-register address-modified.

v = 0: no B-register address-modification of input.

aaaa: address of base of location into which is written first word read from paper tape.

Description of Operation. Read from unit u, into consecutively addressed locations beginning with B[aaaa], until a word with the sign digit equal to 6 or 7 is encountered. An input word with the sign digit equal to 6 or 7 goes to the C register, without or with B-register address-modification, respectively, for immediate execution; the reader is turned off.

Input sign control:

1. If the input sign digit is 6, the word goes to rC for immediate execution; the reader is turned off.
2. If the input sign digit is 7, the word goes to rC, with B-register address-modification, for immediate execution; the reader is turned off.
3. If v is equal to 8 or 9 and:
   a. If the input sign digit is 8, the word is B-register address-modified; the sign goes into the specified storage location as 0.
   b. If the input sign digit is 9, the word is B-register address-modified; the sign goes into the specified storage location as 1.

Flow Chart. See page 5-7.

Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. Paper-Tape ALARM STOP due to a parity error, detection of an inadmissible character, or the designated unit in not-ready status.

Remarks.

s / v	0	1	2	3	4	5	6	7	8	9
0			A				C	C B		
1			A				C	C B		
2			A				C	C B		
3			A				C	C B		
4			A				C	C B		
5			A				C	C B		
6			A				C	C B		
7			A				C	C B		
8			A				C	C B	B_0	B_1
9			A				C	C B	B_0	B_1

s = Sign of input word. (Read across.)

v = Variant digit of PRB instruction. (Read down.)

A: Convert each character in the word to a two-digit (alphanumeric) code.

B: Prior to storing, B-modify input word.

B_i: Same as B. Sign of input is equal to i, where i = 0 or 1.

C: Turn off Photoreader. Transfer input to C register and execute.

Register Status.

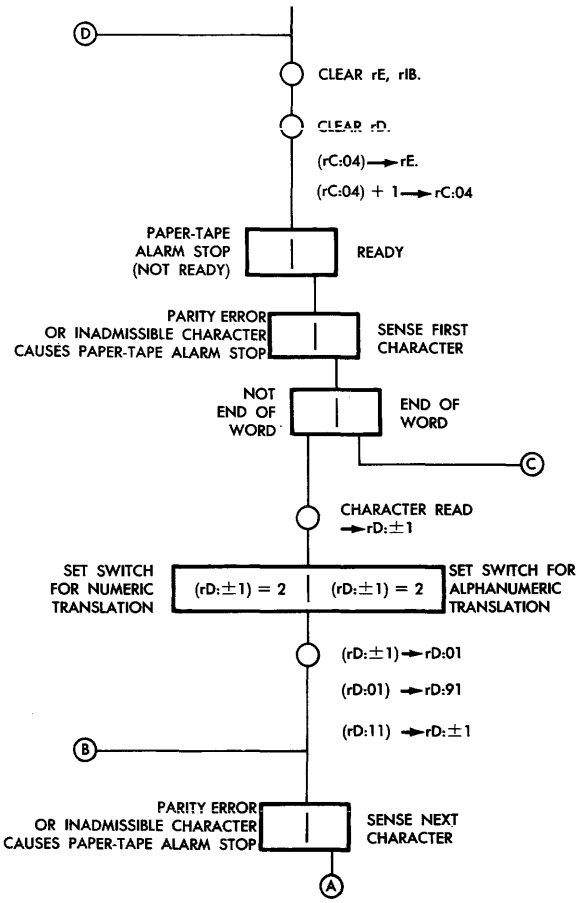
Register name	Contents after execution of PRB
A	Unchanged
R	Unchanged
D	Last word read
B	Unchanged

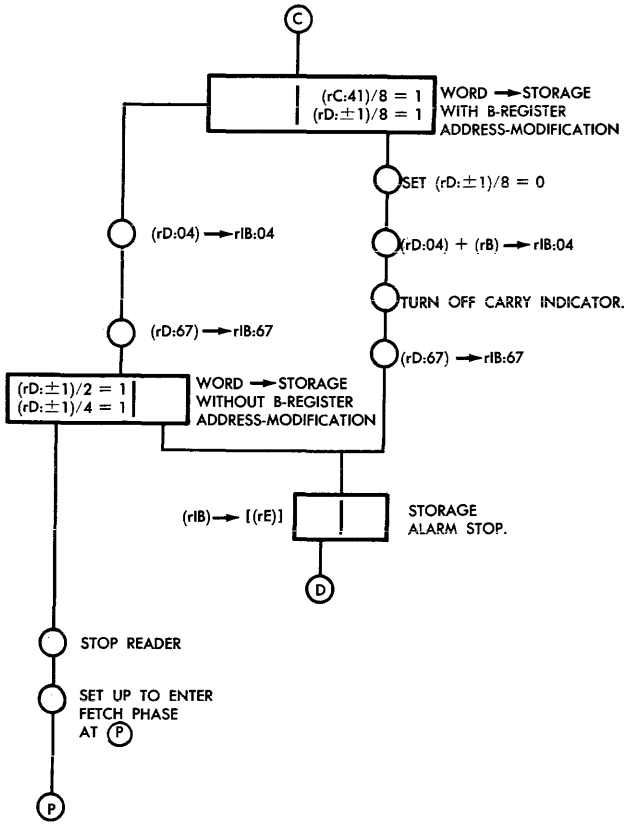
Register name	Contents after execution of PRB
P	$(rP)_b + 1$
C	u i i v 0 4 * * * *
E	Address of last location filled

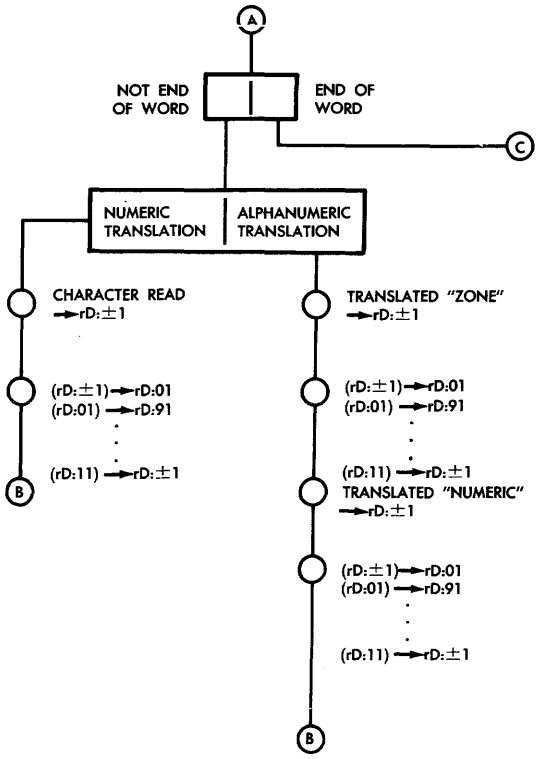
**** Sum: address of last location filled + 1.

Register name	Contents after Storage ALARM STOP
A	Unchanged
R	Unchanged
D	Last word read
B	Unchanged
P	$(rP)_b + 1$
C	u i i v 0 4 * * * *
E	Address causing ALARM STOP

**** Sum: address causing ALARM STOP + 1.







PAPER-TAPE READ, INVERSE FORMAT (PRI)

Operation Code. 05

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	u	n	n	v	O	p	a	a	a	a

Definitions.

u: designates paper-tape reader unit.

nn: specifies the number of words to be read. nn = 00 means 100 words.

v: variation designator:

v = 8, or 9: designated input will be B-register address-modified.

aaaa: address of base of location into which is written first word read from paper tape.

Description of Operation. Read nn words, or until a word with the sign digit equal to 6 or 7 is encountered, from unit u into consecutively addressed locations beginning with B[aaaa]. An input word with the sign digit equal to 6 or 7 goes to the C register, without or with B-register address-modification, respectively, for immediate execution; the reader is turned off.

Input sign control:

1. If the input sign digit is 6, the word goes to rC for immediate execution; the reader is turned off.
2. If the input sign digit is 7, the word goes to rC, with B-register address-modification, for immediate execution; the reader is turned off.
3. If v is equal to 8 or 9 and:
   a. If the input sign digit is 8, the word is B-register address-modified; the sign goes into the specified storage location as 0.
   b. If the input sign digit is 9, the word is B-register address-modified; the sign goes into the specified storage location as 1.

Flow Chart. See page 5-9.

Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. Paper-Tape ALARM STOP due to a parity error, detection of an inadmissible character, use of a 2 in the sign-digit position of a word, or the designated unit in not-ready status.

Remarks.

1. Because the sign digit is read last, alphanumeric translation is not possible.

2.

v \ s	0	1	2	3	4	5	6	7	8	9
0			E				C	C B		
1			E				C	C B		
2			E				C	C B		
3			E				C	C B		
4			E				C	C B		
5			E				C	C B		
6			E				C	C B		
7			E				C	C B		
8			E				C	C B	B_0	B_1
9			E				C	C B	B_0	B_1

s = Sign of input word. (Read across.)

v = Variant digit of PRI instruction. (Read down.)

A: Convert each character in the word to a two-digit (alphanumeric) code.

B: Prior to storing, B-modify the input word.

B_i: Same as B. Sign of input is equal to i, where i = 0 or 1.

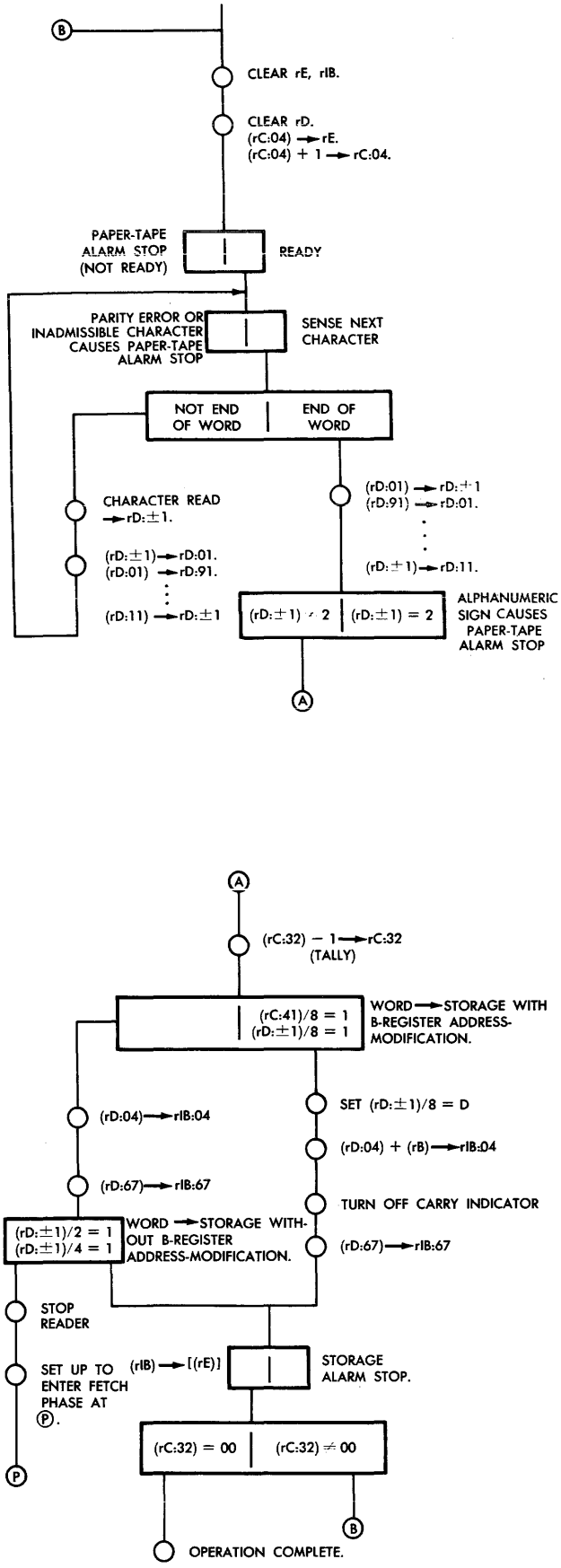
C: Turn off Photoreader. Transfer input to C register and execute.

E: Error Stop.

Register Status.

Register name	Contents after execution of PRI, no override of nn
A	Unchanged
R	Unchanged
D	Last word read
B	Unchanged
P	$(rP)_b + 1$
C	u 0 0 v 0 5 B[aaaa] + nn*
E	B[aaaa] + nn − 1*

* If nn = 00, add 100.



Register name	Contents after execution with override of nn
A	Unchanged
R	Unchanged
D	Last word read
B	Unchanged
P	$(rP)_b + 1$
C	u | * | * | v | 0 | 5 | * | * | * | *
E	* * *

* *, * * *, * * * * See description of operation.

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	Last word read
B	Unchanged
P	$(rP)_b + 1$
C	u | * | * | v | 0 | 5 | * | * | * | *
E	Address causing ALARM STOP

** nn minus number of words read.

**** Sum: address causing ALARM STOP + 1.

PAPER-TAPE WRITE (PWR)

Operation Code. 06

Instruction Format.

	±	1	2	3	4	5	6	7	8	9	0
	±	u	n	n	i	O	p	a	a	a	a

Definitions.

u: designates output unit.

nn: specifies number of words to be punched. nn = 00 means 100 words.

aaaa: address of base of location from which is read first word to be punched.

Description of Operation. Write nn words on unit u, taking the words from consecutively addressed locations beginning with B[aaaa].

Flow Chart. See page 5-10.

Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address.
2. Paper-Tape ALARM STOP; unit designated is in not-ready status.

Remarks.

1. Alphanumeric translation may be suppressed by setting the ALPHA SUPPRESS switch on the writing unit to "on."
2. The writing of leading zeros may be suppressed by setting the ZERO SUPPRESS switch on the writing unit to "on."

Register Status.

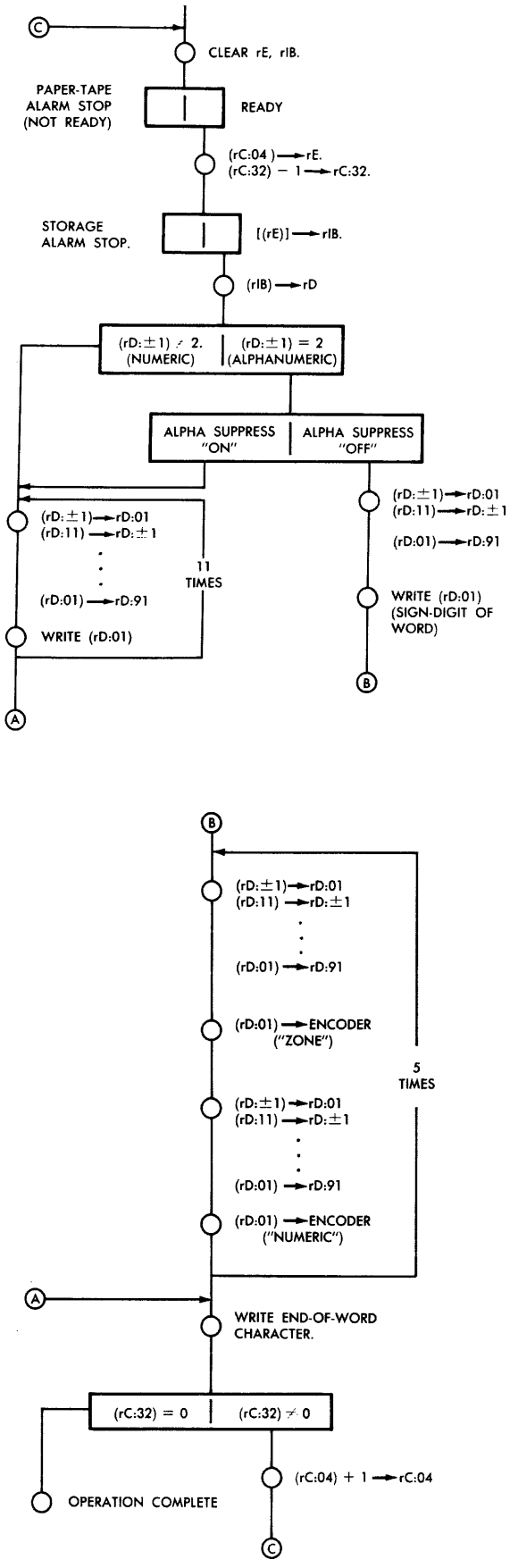
Register name	Contents after execution of PWR
A	Unchanged
R	Unchanged
D	Last word punched
B	Unchanged
P	$(rP)_b + 1$
C	u | 0 0 | i | 0 6 | B[aaaa] + nn*
E	B[aaaa] + nn − 1*

* If nn = 00, add 100.

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	Last word punched
B	Unchanged
P	$(rP)_b + 1$
C	u | * * | i | 0 6 | * * * *
E	Address causing ALARM STOP

** nn minus number of words punched.

**** Sum: address causing ALARM STOP + 1.



6

The Cardatron System

GENERAL

The BURROUGHS 220 CARDATRON System is that facility which couples the Data Processor with the peripheral card-handling machines. The system is designed to accommodate volume punched-card input and output and, if required, to supply a complete printed report of each operation. Alphabetic, numeric or alphanumeric information may be processed with equal proficiency.

Three elements are basic to the system. The Control Unit (housed in two cabinets), the Input Unit and the Output Unit. A full CARDATRON complement can include, in addition to the Control Unit, a maximum of seven Input and Output Units in any combination. For example, the operations at a typical installation might call for three Input Units, to each of which is attached a card reader, and four Output Units, to each of which is attached a card punch or a line-at-a-time printer.

The CARDATRON Input and Output Units serve as buffers for card input from the card readers and for Data Processor output to the punches and printers. Card reading, punching and line printing operations may be carried out independently of Data Processor control; while these operations are taking place, the Data Processor is free to process other portions of the program. The CARDATRON buffering functions, therefore, permit independent and simultaneous multiple-input-output operations.

A distinctive feature of the CARDATRON System is its editing ability, which complements the editing and format control facilities of the plugboard panel on the card-handling machine. During the input phase, for example, each card can select one from among six different editing modes, five of which are under program control. These modes will not only edit the contents of a card, but will automatically translate card information into Data Processor code.

It is the purpose of this section to describe the communication of the CARDATRON System with the peripheral card-handling equipment and the Data Processor; and to examine the coding techniques and programming methods which afford maximum efficiency of operation.

INPUT UNIT OPERATION

MACHINE CHARACTERISTICS

The principal components of the Input Unit are a buffer drum and the associated control circuitry including a Core Shift Register which effects the translation of card code to CARDATRON code. The operation of this Core Shift Register is automatic and does not come under the control of the programmer. The buffer is a magnetic storage drum which is two inches in diameter and revolves at a speed of 21,600 rpm. (One complete revolution requires 2.78 ms.) Information is stored around the drum on one information band and editing digits on format bands. Each of these bands contains 320 digit positions, 319 of which are available for operational use.

THE INFORMATION BAND

Information read from a single card is stored on the information band of the buffer drum. The execution of a CARD READ (CRD) instruction transfers this information to core storage in the Data Processor. All editing of the information is under control of one of the format bands. Information is recorded on the information band in binary-coded decimal notation; four bits are available for each digit position. Two digit positions on the information band are required to record the information from one card column; one position receives the digit supplied by the numeric punch on the card (rows 9 through 1), and the second position receives the digit supplied by the zone punch (rows 0, 11, and 12). Therefore, 160 of the 319 digit positions will always be used in transferring information from an 80-column punched card. Of the 29 words (319 positions) on the band, eleven positions are used to shift the information through the system, permitting a maximum of 28 Data Processor words to be used for active information. The 29th word on the format band (eleven lowest order digits) will be filled with zeros.

FORMAT BANDS

There are five format bands on the buffer drum. The digits 0, 1, 2 and 3 in the format bands perform the editing function; there are two bits available for each digit position. The digit positions on the format band correspond to those on the information band and are numbered consecutively in the same order. Each format-band digit controls the transfer of information to and from the corresponding digit position on the information band.

The card being read on the buffer drum has preselected the particular format band to be used in editing the information on that card. The section is made by a punch in a predesignated column. This punch can indicate one (and

in certain cases more than one) of eight possible format choices: five of these are the format bands, and the other three are the fixed editing functions.

These fixed editing functions control the following operations: transfer of information with a standard numeric format; rejection of cards from which the information is not to be transferred to the Data Processor; and inhibition of the automatic card-reading operation, used when loading or changing format-band digits.

Each card column requires two digit positions on the information band, and these two digit positions are edited by the corresponding digit positions on the selected format band. It is important that the reasons for this be given extensive consideration. A primary function of the CARDATRON System is data editing and format control. The system will accommodate alphanumeric information, translate alphabetic and numeric characters from the 12-row card code to two-digit binary-coded decimal machine code, and effect the proper placement of these characters in the Data Processor word.

An alphabetic character requires two punches in a given card column and therefore two positions on the information band (one for each punch); it is also represented in the Data Processor word by two digits. A numeric character, though it requires only one punch on a card, is also represented on the information band by two digits; it may be reduced, under format control, to one digit in the Data Processor word. Therefore, all card columns require two positions on the information band. Within the Data Processor, alphabetic material is always coded in two digits; numeric information may be represented by either one or two digits, depending on whether the card field is treated as numeric or alphanumeric.

ROW COUNTER

CARDATRON System operations are controlled by a 4-bit decade called the row counter. This counter has 16 decimal values, 0 through 15. The setting of the row counter is shown by the RC indicators in the 0 register on the Input Unit control panel[1]; these indicators show what part of the card cycle is in progress. During the reading of a card, the row counter is synchronized with the digit pulses from the card reader, so that when card row 9 is being read, for example, the row counter is set at 9. This setting assigns a digital value to the punches being sensed at a particular time. (See Table 6-1 for the various row counter settings and their interpretations.)

CORE SHIFT REGISTER

The Core Shift Register of the Input Unit consists of 80 magnetic cores which are connected to the 80 TO CARDATRON hubs on the card reader. A punch in a card column pulses the TO CARDATRON hub for that card column, which sets the corresponding core. Each core, therefore, is set by a punch from the corresponding card column. The Core Shift Register is set in parallel; all punches in one 80-column row are sensed and the cores set simultaneously. The configuration of set cores duplicates the configuration of punches in any one card row. The cores then transfer the information in serial fashion by shifting the information off to the right and on to the buffer drum. The value of the digits written on the buffer drum is determined by the row counter setting (Table 6-1).

THE NUMERIC TOGGLE

Since there are 160 format-band digits controlling the transfer of 80 card columns, and since there are only 80 cores in the core register, it follows that not all of the format-band digits can be active and cause a shift in the Core Shift Register at any one row time. There are two format-band digits for every card column, one to transfer the numeric punch and one to transfer the zone punch. The row counter indicates whether a numeric or zone row on the card is being read. Working in conjunction with the row counter is the numeric (NU) toggle (an internal electronic switch) which, by its set or reset condition, designates which format digits transfer numeric punches and which transfer zone punches to the information band. The NU toggle changes state each time one of the format digits which affects the transfer of information from a card column is encountered in the format band. Thus the CARDATRON System keeps track of the transfer digits in the active segment of the format band so that they are interpreted alternately with numeric or zone significance. When the row counter is set at a numeric row value, the state of the NU toggle will cause only the format-band digits which transfer numeric punches to be active; when the row counter indicates zone time, only the format-band digits controlling the transfer of zone punches will be active.

THE TWO PHASES OF INPUT OPERATION

The transfer of information through the Input Unit occurs in two phases. The first phase involves reading the information from a card onto the drum (Figure 6-1). This process takes place automatically, provided that the card machine is ready to begin a card reading cycle. The Input Unit stores this information on the buffer drum as long as necessary.

The second phase of operation occurs when a CARD READ (CRD) instruction is executed by the Data Processor, causing the Input Unit to transfer the information from the buffer drum to core storage. As soon as this information leaves the Input Unit, automatic read-in of the next card follows immediately, unless Reload Lockout has been imposed.

[1] See *Handbook of Operating Procedures for the BURROUGHS 220.*

TRANSFER OF DATA TO INFORMATION BAND

As a card feeds into the reader, it is read row by row: the punches sensed in each row pulse the appropriate TO CARDATRON hubs, causing the corresponding cores of the Core Shift Register to be set. Each row is transferred separately to the Core Shift Register which duplicates the configuration of punches in that card row. When that information is shifted onto the buffer drum, its digital value is assigned by the row counter. If, for instance, a punch occurs in column 72 of the card being read, the core corresponding to column 72 will be set. If the row counter indicates 9 time, then the digital value transferred to the buffer drum from the set core will be 9.

While the card reader prepares to read the next row, the Core Shift Register shifts the information off in serial fashion, one digit at a time, shifting 80 positions to the right for each row read. The information from the row is transmitted to the information band; its placement in the 319 digit positions on the band is determined by the format digits in the equivalent digit positions on the format band selected. The shifting of the register is synchronized with the rotation of the buffer drum, so that with each shift one digit position on the information band passes under the read-write head. When a zero appears in the format band, the core register does not shift; no information is transferred, and so a zero is placed in the corresponding position on the information band. Thus zeros are inserted between information digits at this state, expanding the 80 columns of card information as much as desired within the 319 positions available. When a 2

Table 6-1. Row Counter settings, CARDATRON 220 Input Unit

9-Edge-First Machine		12-Edge-First Machine	
Row Counter Setting	**Significance**	**Row Counter Setting**	**Significance**
10	Buffer unloaded, ready for reader.	13	Buffer unloaded, ready for reader.
9	Read 9 row of card.	14	Read 12 row of card.
8	Read 8 row of card.	15	Read 11 row of card.
7	Read 7 row of card.	0	Read 0 row of card.
6	Read 6 row of card.	1	Read 1 row of card.
5	Read 5 row of card.	2	Read 2 row of card.
4	Read 4 row of card.	3	Read 3 row of card.
3	Read 3 row of card.	4	Read 4 row of card.
2	Read 2 row of card.	5	Read 5 row of card.
1	Read 1 row of card.	6	Read 6 row of card.
0	Read 0 row of card.	7	Read 7 row of card.
15	Read 11 row of card.	8	Read 8 row of card.
14	Read 12 row of card.	9	Read 9 row of card.
13[a]	Buffer loaded, ready for Data Processor.	10[b]	Buffer loaded, ready for Data Processor.
12	Transfer information to Data Processor if CRD or erase format band if CRF.	11	Transfer information to Data Processor if CRD or erase format band if CRF.
11	Erase information band if CRD.	12	Erase information band if CRD.

[a] To execute a CRF instruction, RC must be set to 11 or 13, and FS[2] to 7.

[b] To execute a CRF instruction, RC must be set to 12 or 10, and FS[2] to 7.

[2] Ibid

appears in the format band, a 2 is transferred by the same procedure to the corresponding position on the information band.

The information band on the buffer drum is loaded automatically, unless this operation is restricted by Reload Lockout instructions from either the Data Processor or the preceding card. When all the rows on a card have been sensed and the information from that card has been transmitted through the Core Shift Register, all the information from that card has been accumulated on the information band in binary-coded decimal form.

TRANSFER OF DATA TO CORE STORAGE

Execution of A CARD READ (CRD) Instruction. Execution of a CARD READ instruction transfers the contents of the information band to a series of consecutively addressed locations in core storage in one revolution of the buffer drum. Again the format-band digits take control of the transfer. As each digit from the information is transmitted through the control units, the format digits determine whether the material is to be translated as alphabetic or numeric information. The same digits govern the final editing of the information just before it enters the D register. Specified digits may be deleted or zeros or 2's may be inserted in the information stream before it reaches core storage. Those digits, which are to be transferred to storage along with the 0's (often called scaling 0's) that have been used to fill out Data Processor words during transfer of data to the information band, now begin their passage through the registers of the Data Processor to core storage.

Flow of Information through The Data Processor. Digits from the first word transferred from the information band on the Input Unit buffer drum enter the Data Processor through the sign position of the D register (see Figure 6-1). The D register shifts right until it is filled with the 11 digits of a Data Processor word. The right-most five digits of this word are shifted through the adder, (where, if called for, B-register address-modification occurs) to the five low-order positions of the IB register. The six left-most digits are shifted directly to the six high-order positions of the IB register. The second word is shifted into the D register immediately following the first, and prepares to make the same transfer.

Meanwhile, the first word, which is in the IB register, is transferred to core storage, clearing the IB register. As soon as the first word is stored, the second word fills the IB register. The third word is then transferred from the information band to the D register and so forth. This process continues until the last word to be transferred is in the D register. The last word is shifted into core storage, through the adder and IB register, by the eleven zeros in the inactive segment of the format band. At this point, all the words have been transferred and stored in sequence, and the information band has been completely read. The Data Processor is free to continue processing while the card machine automatically proceeds to read the information from the next card onto the buffer drum.

Information is read from cards through the Core Shift Register which transfers it onto the information band on the buffer drum. The information is read from right-to-left: that is, by column, from column 80 to column 1. It is stored, however, in left-to-right order, by word, in *descending* sequence from the address given in the CARD READ instruction so that the left-most card field enters

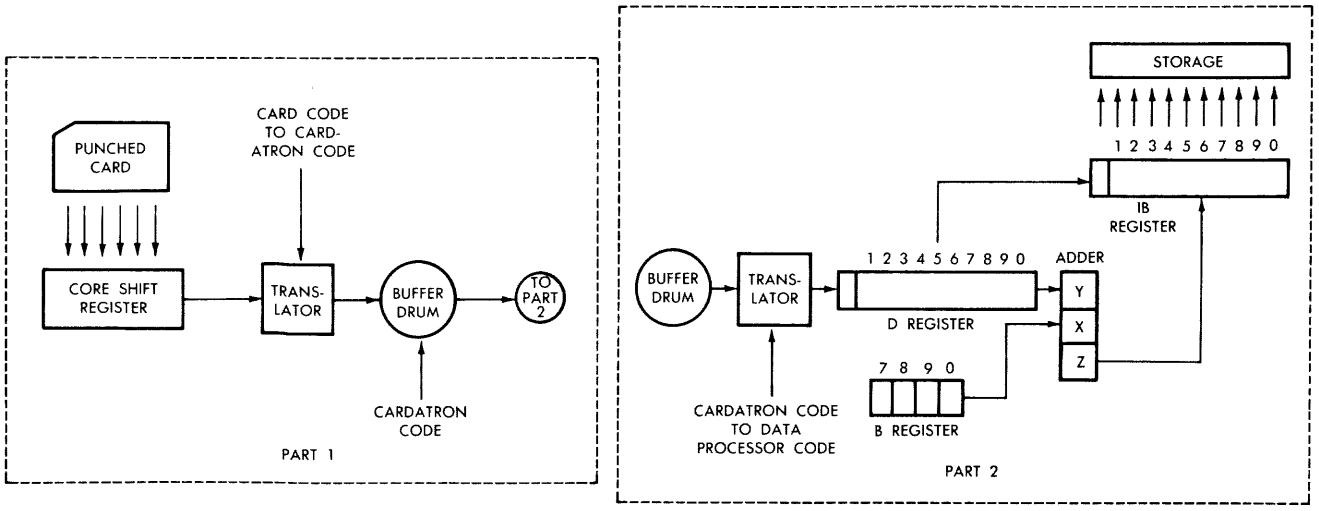


Figure 6-1. Input information path.

the lowest memory location. This procedure facilitates the use of B-register address-modification on input. (Output operations transfer the information in right-to-left order, by word, so that the information will reappear, after processing, in the same relative sequence as that in which it entered the system.)

If an attempt is made to execute a CARD READ instruction before the transfer of information from card to buffer has been completed, the Data Processor will wait until that transfer is completed; the execution of the CARD READ instruction will then proceed. The status of an Input Unit may be ascertained by executing a CARD READ INTERROGATE, BRANCH instruction.

Use of The INTERROGATE Instruction. The Data Processor cannot execute a CARD READ (CRD) instruction until the transfer of information from card reader to the information band is complete. Therefore, some computation time may be lost while the Data Processor waits until the buffer drum is ready to be unloaded. The CARD READ INTERROGATE, BRANCH (CRI) instruction has been provided to avoid this delay; with this instruction, the Input Unit can be interrogated to find out if it is ready. If it is ready, the information can be transferred and the next card can be read. If it is not ready, processing of information which is already in storage continues.

Altering the Flow of Information. A 6 or 7 punch in the sign-digit position of a word transferred from the buffer drum can be used to change the path of the information transfer. Under sign control, the information in a given card word can be sent to the C register after its passage through the D register, the adder, and the IB register. The 6 or 7 in the sign position inhibits the transfer into core storage, and the information passes directly into the C register. Such input words can be used to control the input and storage of subsequent information.

INPUT FORMAT BANDS

FORMAT SELECTION

The cards which are read by a card-handling device for input through the CARDATRON System must contain a punch in a predesignated column indicating which format band or fixed editing function will govern the loading process, if each card is to select its own format. These punches are interpreted (under control of the plugboard panel wiring) before the information is read from the card, causing the proper selection of format by the Input Unit.

PROGRAM-CONTROLLED FORMAT BANDS

The first five of eight possible format choices are the format bands on the buffer drum. Each is identified on the plugboard panel of the card reader by one of the digits 1, 2, 3, 4, or 5, and can be selected by a punch in a designated card column. These five format bands are program-controlled in that the editing digits are loaded onto the format band by the execution of a CARD READ, FORMAT LOAD (CRF) instruction. This instruction transfers 319 format digits (29 Data Processor words) from Data Processor storage to a designated input format band. The format bands cannot be loaded directly from cards; the format digits must first be stored in the Data Processor as 29 Data Processor words.

FIXED EDITING FUNCTIONS

In addition to the five format bands, three special editing functions are available for input only; these too are chosen by a punch in the column and are identified by the numbers 6, 7, and 8. These fixed editing functions cannot be modified to the point of changing format or characters by an instruction from the program stored in the Data Processor.

Numeric Format (6). Use of format 6 causes all information from a card to be treated as numeric and transferred according to a set pattern. The primary purpose of format 6 is to load the object program, including format bands, into core storage. Two format-band digits are required to transfer information from a card column; the pair of digits 31 transfers numeric information. Each 31 combination of format digits will transfer one Data Processor digit.

When format 6 is used, cards must be punched with 11-column fields beginning at the right. Format 6 affects the information from the card as follows:

All overpunches are deleted; that is, only the numeric punches are sensed. (Under numeric interpretation, a blank column will be treated as a 0 punch.) The sign is transferred from the lowest numbered column (the least-significant-digit position) of each card field.

Any card read in under this editing pattern must have the following format: the card is divided into eight fields comprised of columns 1 through 3, 4 through 14, 15 through 25, 26 through 36, 37 through 47, 48 through 58, 59 through 69, and 70 through 80. The numeric part of each of these eight fields, with the exception of the first, is translated as an 11-digit Data Processor word.

Normally these columns will be wired to the corresponding sets of TO CARDATRON hubs on the plugboard panel; however, it is possible, if the pragrammer desires, to use other plugboard panel wiring to rearrange the order of the words. For example, card columns 37 through 47 could be wired to TO CARDATRON hubs 4 through 14. Other card fields could be transposed on input in the same way.

The eight card fields are transferred from the card in right-to-left order and are stored in the Data Processor in descending sequence. The simulated format digits will transfer eight full Data Processor words to storage. To the numeric part of the three-column field wired to TO CARDATRON hubs 1 through 3 are added eight high-order 0's to form a complete Data Processor word. An additional five words of zeros are supplied automatically by the fixed numeric format function, effecting a transfer of a total of thirteen full Data Processor words. In many cases, the programmer will not want the five words of 0's occupying space in Data Processor storage; a signal can be supplied by the input card to inhibit the transfer to storage of the 0's or of unwanted information.

Reject Format (7). The reject function allows a card to pass through the card reader without transfer of the information in it to core storage. The contents of the card are actually transferred to the information band under control of format 6. However, transfer of information to the Data Processor is inhibited, because the selection of format 7 sets the row counter so that the Input Unit by-passes the transfer of digits from the information band to the Data Processor, clears the information band, and immediately starts the next card-read cycle. The next card feeds into the card reader automatically, and its contents fill the information band, unless Reload Lockout has been imposed. A card which selects Reject Format has no effect on the program, since its contents never reach the Data Processor.

Reload Lockout (8). When Reload Lockout is selected, automatic feed and read-in of the card following is inhibited. There are two possible situations when Reload Lockout may be used: (1) when one or more format bands are to be changed, and (2) when the programmer wants the information band to be read twice so that its contents will appear in two storage areas. Reload Lockout may be imposed either from the stored program or from a card punched to designate format 8.

From the stored program Reload Lockout must be imposed by a CARD READ (CRD) instruction. The imposition of Reload Lockout by a CARD READ instruction does not inhibit the transfer of the existent contents of the information band, but inhibits rather the automatic signal back to the card machine to begin reading the next card. It may then be reimposed by either the next CRD or a CARD READ, FORMAT LOAD (CRF) instruction. Either of these instructions, when properly coded, may also release Reload Lockout.

Reload Lockout may be selected by coding a 1 or 9 in the r-digit position of the CRD instruction (± uivr 60 aaaa). It may be reimposed, if desired, by coding an odd digit in the f-digit position of the CRF instruction (± uiif 62 aaaa). Reload Lockout will be released by the CRD instruction if the r-digit is 0 or 8; if r is an even digit in the CRF instruction, Reload Lockout will be released.

The selection of the format band on which the editing control stream is written and the imposition or release of Reload Lockout[3] are effected by the CRF instruction as described in the table which follows:

f	Format Band	Reload Lockout
0	1	Released
1	1	Imposed
2	2	Released
3	2	Imposed
4	3	Released
5	3	Imposed
6	4	Released
7	4	Imposed
8	5	Released
9	5	Imposed

If Reload Lockout is imposed to permit a format band to be changed, the CRD instruction used to impose Lockout will be followed by a CRF instruction. The CRF instruction causes the format digits on the designated band to be replaced by new format digits from the specified locations in core storage.

If only one format band is to be loaded, the variant-digit f in the CRF instruction is coded even. On execution of the instruction, the designated format band is loaded, Reload Lockout is released, and the contents of the card that was locked out are read onto the information band. The next CRD instruction will cause the new contents of the information band to be transferred to core storage.

If more than one format band is to be loaded, the f digit in each CRF instruction is coded odd, to reimpose Reload Lockout, until the instruction which loads the last format band is reached. The f digit is then coded even.

[3] Format band 8 must be selected in conjunction with the selection of some other format band; if it is not, a No Format ALARM STOP will occur. (The No Format ALARM appears on the CARDATRON Input Unit Panel and a CARDATRON ALARM appears on the Control Console of the Data Processor.)

The effect of the imposition of Reload Lockout by selection of format 8 from the punched card is exactly the same as that which would occur if a 1 or a 9 had been punched in the r-digit position of the CARD READ instruction. This instruction will transfer the contents of the information band to core storage and will impose Reload Lockout just as though it had been so coded.

THE USE OF INPUT FORMAT BAND DIGITS

The transfer of information from card to buffer, and from buffer to Data Processor storage is effected under control of the format band. *Two format-band digits are required to transfer and edit the information from each card column.* To read an 80-column card, therefore, a minimum of 160 format digits must be used in the active segment of the format band.

INSERT ZERO DIGIT (0)

Use of this digit on a format band causes a 0 to be written in a designated position on the information band. In no way does it affect the operation of the CARDATRON System; the 0 does not replace nor transfer information from the punched card; it only causes 0's to be written into the Data Processor word. As demonstrated in Figure 6-2, the programmer, if he desires, can construct a complete 11-digit word of 0's. Assume, for example, that in a given card, columns 36, 37, 38 and 39 contain numeric information. The format band transfers this information so that it is reproduced in storage as single numeric digits.

Two 0's in the corresponding positions in the format band will produce two 0's in the Data Processor word between the digits supplied by card columns 37 and 38:

Card Column	36	37		38	39
Contents of card field	6	7		8	9
Format-band digits	31	31	00	31	31
Portion of Data Processor word	6	7	00	8	9

ALPHANUMERIC-TRANSFER DIGIT (1)

A 1 digit in the format band will transfer either a numeric punch or a zone punch into the corresponding position in the information band. If, for example, the first two format-band positions contain 1's (these are the two format-band positions which edit the contents of card column 80), the 1 digit in format position 1 will transfer the numeric portion of column 80 to the information band, and the 1 in format position 2 will transfer the zone portion of that column. A pair of 1's, therefore, is required to transfer both punches from one card column. Only the 1 digit will transfer information from a card to a Data Processor word. In the transfer from buffer drum to core storage, the transfer is effected under control of the same 1 digits which controlled the transfer from card to buffer drum. The following example will illustrate the effect of the 1 digit on card information.

Card Column	80
Format-band digits	11
Alphabetic contents of card field	A
Portion of Data Processor word	41

Further, then, if card column 79 contained both numeric and zone punches for alphabetic information, and format positions 3 and 4 contained 1's:

Card Column	79
Format-band digits	11
Alphabetic contents of card field	B
Portion of Data Processor word	12

INSERT 2 DIGIT (2)

On input, a 2 digit inserts a 2 in the corresponding position on the information band. In the transfer from buffer drum to core storage, a 2 is inserted into the corresponding position in the Data Processor word. In principle, then, the function of format digit 2 is the same as that of the 0 digit: there is a one-for-one transfer of 2's into storage. In addition, a 2 permits the insertion, in the sign-digit position of a word, of the flag which identifies it as an alphanumeric word. It is this digit which effects compatibility between the CARDATRON System and the Paper-Tape and Supervisory Printer systems.

DELETE DIGIT (3)

Format digit 3 deletes the numeric or zone punch which it is editing. The primary uses of this digit are the contraction of punched card information and the filling-out of the inactive segment of the format band. Data from the card column coming into the information band under control of format digit 3 is transferred to the information band in the normal fashion. However, with the execution of a CARD READ instruction, the 3 in the format band inhibits the D register shift and prevents the information under its control from reaching the D register. For this corresponding digit position, therefore, nothing is transferred to core storage.

Card Columns	50	51	52
Format-band digits	31	33	11
Alphanumeric contents of card field	6	A	B
Portion of Data Processor word	6		42

Note that in column 50 the 1 digit transferred the numeric portion of the column, a 6, into its corresponding position in the Data Processor word. The 3, however, deleted the zone portion of the column, so that nothing was transferred to its corresponding position in the Data Processor word. In column 51, the digit pair 33 deleted both the numeric and zone portions of the column, so that nothing was transferred to the Data Processor word for this digit position. The digit pair 11 in column 52 transferred the alphabetic information (B) to core storage and translated it into Data Processor code for that character (42).

TRANSFERRING THE SIGN-POSITION DIGIT

The eleventh digit position, the high-order digit, of the Data Processor word is the sign-digit position of that word. This digit may be supplied by an overpunch in the card column at zone time or by a 1 or 0 punch at numeric time. The sign is transferred under control of the same format digits which edit any numeric information. Assume, for example, that the format calls for an overpunch in the sign position, but there is not enough numeric information to fill all eleven digits of the Data Processor word, as illustrated in the following example:

Card Columns	76	77	78	79	80
Contents of card field	4	7	7	9	5
Format-band digits	1000001	31	31	31	31
Data Processor word	1000004	7	7	9	5

The digit pairs 31 transferred the numeric information in columns 77—80; the 1 corresponding to the numeric portion of column 76 transferred that portion of the column. 0's in the remaining format positions transferred 0's into the next five most-significant-digit positions, thus forming a complete Data Processor word. The format digit 1 corresponding to the zone portion of column 76 transferred that portion at sign time. Owing to the insertion of the 0's, the D register did not interpret this punch as information belonging to the sixth most-significant-digit position, but as the sign digit of this Data Processor word.

The sign digit may also be supplied by a 1 or 0 punch in the numeric portion of a given card column. If the card field does not contain sufficient information to form a complete Data Processor word, 0's may be inserted to fill in the remaining digit positions, and the designated column will be transferred at sign time.

Card Columns	75	76	77	78	79	80
Contents of card field	1	4	7	7	9	5
Format-band digits	3100000	31	31	31	31	31
Data Processor word	100000	4	7	7	9	5

The 1 digit transferred the numeric portion of column 75 at sign time; the zone portion was deleted by the digit 3. After the transfer of the information in column 76, the transfer of the 0's was not recognized as numeric activity by the D register. Thus, the word in storage was completed by the 0's, but the transfer of the numeric punch in column 75 was delayed until sign time.

FUNCTIONS OF INPUT FORMAT-BAND DIGIT PAIRS

In understanding the applications of format-band digits, it is necessary to remember that card column information can be accommodated only by pairs of format digits; one digit to transfer the numeric portion of the column; one digit to transfer the zone portion.

NUMERIC TRANSFER: DIGIT PAIR 31

As shown in previously cited examples, the format digit 1 will transfer one digit of numeric information to the information band and then to core storage. To effect the transfer of numeric characters, therefore, only the numeric portion of the card column need be transferred by the format digit 1; the zone portion is deleted by the digit 3. Thus the digit pair 31 is used to edit numeric information in a given card column.

ALPHABETIC TRANSFER: DIGIT PAIR 11

The Data Processor recognizes certain pairs of decimal digits as being the equivalents of certain alphabetic characters, and under specified conditions will translate them as such. For example: A = 41, B = 42 and so forth. Since each portion of the card column, zone and numeric, will contain a digit of numeric information, there must be a punch in both portions if alphabetic information is to be obtained. In the editing control stream, then, the digit pair 11 is required to transfer one column of alphabetic information to the information band and core storage. The right-most digit transfers the numeric portion of the column; the next digit transfers the zone portion.

COLUMN DELETE: DIGIT PAIR 33

If the entire contents of a card column are to be deleted and not transferred to the Data Processor word, the digit

pair 33 is entered in the editing control stream in the positions corresponding to that column. Thus both the zone and numeric portions of the column are deleted and the information is not transferred to core storage.

CONSTRUCTION OF AN INPUT FORMAT BAND

For the purposes of most applications, the construction of an input format band can be considered as five separate steps:

1. Determining the editing requirements. These are established when the punched-card format is known and the processing requirements, including the form needed for the Data Processor words, have been stated.
2. Filling out the CARDATRON format-band coding form (Figure 6-2). This furnishes a written representation of the format digits that will do the required editing.
3. Preparing the cards that will contain the format digits to be read into core storage.
4. Loading the format-band digits into storage.
5. Transferring the format-band digits from storage to the designated format band on the Input Unit buffer drum.

USING THE FORMAT-BAND CODING FORM

ORDERING OF EDITING AND TRANSFER

On the format-band coding form (Figure 6-2), the numbered squares for the format-band digits read from right to left. This right-to-left order is characteristic of the BURROUGHS 220 CARDATRON System: the digits are written on the coding form from right to left; information on a card is both edited and transferred from right to left, beginning with column 80.

ACTIVE SEGMENT

Those digits required for editing the contents of a card and inserting 0's and 2's into the Data Processor words make up the active segment of a format band. There are always at least 160 digits in the active segment because two digits are needed to edit each of the 80 columns on a card. There will be more than 160 digits in the active segment if the editing requirements call for the insertion of 0's or 2's among the information digits read from the card. The active segment begins with digit position 1, on the right side of the coding form.

INACTIVE SEGMENT

The rest of the total of 319 available digits comprise the inactive segment of the format band. Eleven of these must be 0's (usually the last eleven); the other positions of the inactive segment are occupied by 3's.

WRITING AN INPUT FORMAT BAND: EXAMPLE

This section describes the use of an input format-band coding form by following, step by step, the process of writing an input format band. The completed sample form is shown in Figure 6-2. A sample problem, such as the Payroll Adjustment Card format and format band shown in Figure 6-2, will illustrate the principles and techniques of coding a format band in a typical but simple situation.

At the top of the CARDATRON format coding form, there is a horizontal row of squares numbered 1—120, from left to right. The first 80 of these numbers represent the 80 columns in a punched Payroll Adjustment input card. The card format is laid out along this row, in parallel fashion, normally from left to right, digit-for-digit, as shown in Figure 6-2. The symbols representing the information in the card are:

A = Alphabetic character

N = Numeric character

D = Digit delete

In left to right order, that is, reading from column 1 through 80, the card itself contains the following information:

1. Two deleted digits, columns 1 and 2.
2. Employee number; numeric information, columns 3 through 10.
3. Employee's name; alphabetic information, columns 11 through 25.
4. Department; alphabetic information, columns 26 and 27.
5. Standard rate of pay; numeric information, columns 28 through 30.
6. Overtime rate; numeric information, columns 31 through 33.
7. Deleted digits; columns 34 through 52.
8. Department number; numeric information, columns 53 and 54.
9. One deleted digit, column 55.
10. Insurance premium; numeric information, columns 56 through 59. Numeric sign, displayed in entire card column.

11. Year-to-date gross; numeric information, columns 60 through 66; sign is overpunched in column 60.
12. Year-to-date net; numeric information, columns 67 through 73; sign is overpunched in column 67.
13. Code; alphanumeric information, columns 74 through 77.
14. Three deleted digits, columns 78 through 80.

In the center of the lower half of the coding form, there is a block labelled STORAGE DISPLAY. Here, the card format is filled in just as it will appear in core storage. It is divided into two segments, each representing a column of Data Processor words (the card format), and their locations in storage (labelled LOC.), with space provided for remarks. The programmer should begin the entries in the bottom row of the right-hand segment, since the information is transferred into storage in descending order.

Beginning with the bottom row of the right-hand segment, the right-most field of information on the punched card, (in this case it is the alphanumeric code) is entered in the space provided. Note that this does not include the three deleted digits in columns 78—80. Since two numeric characters on a punched card are required to transfer one alphabetic character to core storage, every alphabetic character transferred will occupy two digit positions in the Data Processor word. Therefore, the format for this particular word in core storage will be:

1012 00000NAAAAN

The code digits for the alphabetic characters AA in the punched card occupy digit positions 6—9 in the Data Processor word, and are represented in STORAGE DISPLAY as AA AA. The programmer chose to assign this word to storage location 1012; the ensuing words are stored sequentially in descending order.

The next two words, representing this employee's year-to-date net and year-to-date gross, are stored in locations 1011 and 1010, respectively. Since these words contain only numeric information, there is a one-for-one digit transfer from punched card to core storage. They will appear in storage, then, as:

1010 ±000NNNNNNN

1011 ±000NNNNNNN

Provision must be made for including the sign digit, in both cases an overpunch, since these quantities may have either a positive or negative value.

For this particular application, the programmer has chosen to insert a complete word of zeros in location 1009. Most of the other Data Processor words in the STORAGE DISPLAY are partially constructed of zeros. Data Processor words, then, may contain zeros in any amount, and words wholly or partially containing zeros may be placed in the STORAGE DISPLAY wherever desired.

The insurance-premium Data Processor word contains three digits of numeric information. Note, however, that a numeric character with the designation "sign" occupies column 56 of the card format section. This means that the sign digit of the word appears in the numeric portion of the punched card as a 1 if negative, and as a blank if positive.

The next word, the department number, contains two digits of numeric information and is stored in location 1007. Obviously, the deleted digits which follow this word in the punched card are not shown in the STORAGE DISPLAY section.

Storage locations 1006 and 1005 are occupied by the overtime rate and the standard rate of pay respectively. Each contain three digit positions of numeric information, and here again, there is a one-for-one transfer to core storage.

The department code, which occupies columns 26 and 27 in the punched card, is alphabetic information and occupies the four least-significant-digit positions in storage location 1004. As in the case of the code number in location 1012, it is necessary to remember here that two numeric characters are required to transfer one alphabetic character to core storage. Therefore, the word in location 1004 shows the configuration: 2000000AAAA. The same is true of the employee's name, which appears in columns 11 through 25 of the punched card. These 15 digits are alphabetic information; owing to this same requirement of a two-for-one transfer, the name appears in core storage as three words (ten digits plus sign), occupying locations 1003, 1002, and 1001.

Note that in each of the four Data Processor words which contain only alphabetic information, the digit 2 appears in the sign position of each word. Whereas the digit 2 in the sign position will not directly affect operation within the CARDATRON system, it serves to demonstrate the fact that a format digit 2 will insert a 2 in the stored program. Further, the 2 in the sign position is required to effect compatibility between the CARDATRON system and the Paper-Tape and Supervisory Printer systems, if an alphabetic translation of the material is desired. Therefore, unless a 2 is provided in the sign-digit position, there can be no supervisory printout in alphabetic characters.

THE EDITING CONTROL STREAM

In the central portion of the coding form, there are five parallel rows of numbered squares, beginning with 1 and

Burroughs Corporation **BURROUGHS 220** **CARDATRON FORMAT BAND CODING FORM**

INPUT ☒ BAND NO. 1

OUTPUT ☐ UNIT NO. 1

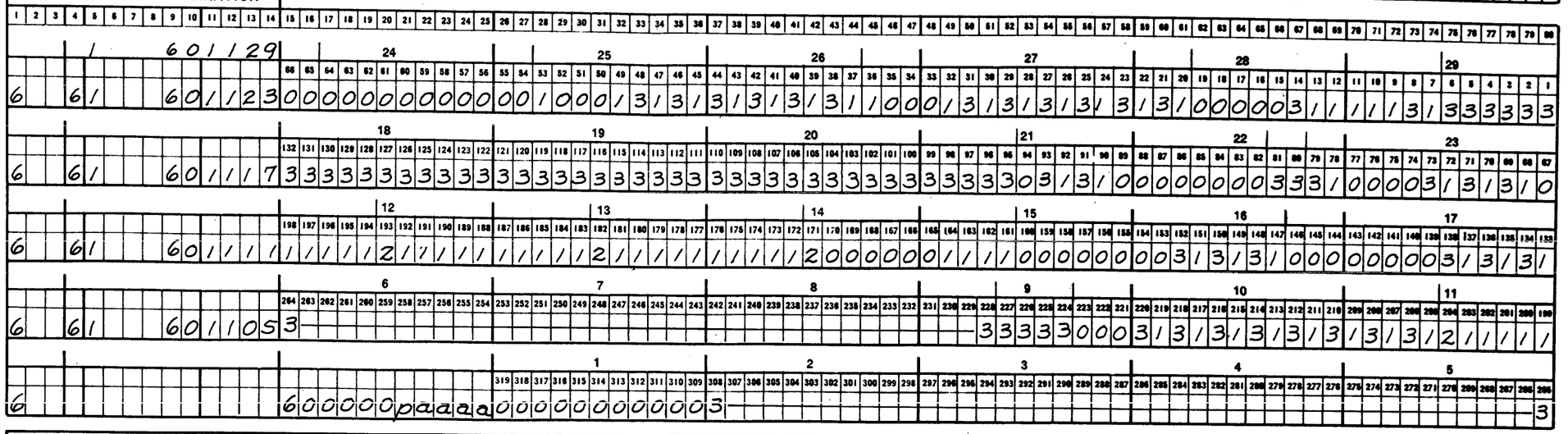
TITLE: SAMPLE PROBLEM: PAYROLL ADJUSTMENT CARD ROUTINE NO. _____ CODER: J. WALDRIP & R. SOLT

D	EMPLOYEE NUMBER	NAME	DEPT	STANDARD RATE OF PAY	O.T PAY		DEPT. NO.	INS. PREM. ± SIGN	Y.T.D. + GROSS	Y.T.D. NET	CODE		A = ALPHA N = NUMERIC D = DELETE
D	NNNNNNNNN	AAAAAAAAAAAAAAAAAAAA	A	NNNNN	NNNNN	DDDDDDDDDDDDDDDDDDDDDDDDDDDDDD	NN	D N	NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	NNNNNNNNNNNNNNNNNNNNNNNNNNNNNN	AAAA	DDD	
1	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51	52	53 54	55	56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80							81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120

PAPER-TAPE PREPARATION

#	±	1	2	3	4	5	6	7	8	9	0
	0	0	0	0	0	0	0	0	0	0	0
	319	318	317	316	315	314	313	312	311	310	309
1	3	3	3	3	3	3	3	3	3	3	3
	308	307	306	305	304	303	302	301	300	299	298
2	3	3	3	3	3	3	3	3	3	3	3
	297	296	295	294	293	292	291	290	289	288	287
3	3	3	3	3	3	3	3	3	3	3	3
	286	285	284	283	282	281	280	279	278	277	276
4	3	3	3	3	3	3	3	3	3	3	3
	275	274	273	272	271	270	269	268	267	266	265
5	3	3	3	3	3	3	3	3	3	3	3
	264	263	262	261	260	259	258	257	256	255	254
6	3	3	3	3	3	3	3	3	3	3	3
	253	252	251	250	249	248	247	246	245	244	243
7	3	3	3	3	3	3	3	3	3	3	3
	242	241	240	239	238	237	236	235	234	233	232
8	3	3	3	3	3	3	3	3	3	3	3
	231	230	229	228	227	226	225	224	223	222	221
9	3	3	3	3	3	3	3	3	0	0	0
	220	219	218	217	216	215	214	213	212	211	210
10	3	/	3	/	3	/	3	/	3	/	3
	209	208	207	206	205	204	203	202	201	200	199
11	/	3	/	3	/	2	/	/	/	/	/
	198	197	196	195	194	193	192	191	190	189	188
12	/	/	/	/	/	2	/	/	/	/	/
	187	186	185	184	183	182	181	180	179	178	177
13	/	/	/	/	/	2	/	/	/	/	/
	176	175	174	173	172	171	170	169	168	167	166
14	/	/	/	/	/	2	0	0	0	0	0
	165	164	163	162	161	160	159	158	157	156	155
15	0	/	/	/	/	0	0	0	0	0	0
	154	153	152	151	150	149	148	147	146	145	144
16	0	0	3	/	3	/	3	/	0	0	0
	143	142	141	140	139	138	137	136	135	134	133
17	0	0	0	0	0	3	/	3	/	3	/
	132	131	130	129	128	127	126	125	124	123	122
18	3	3	3	3	3	3	3	3	3	3	3
	121	120	119	118	117	116	115	114	113	112	111
19	3	3	3	3	3	3	3	3	3	3	3
	110	109	108	107	106	105	104	103	102	101	100
20	3	3	3	3	3	3	3	3	3	3	3
	99	98	97	96	95	94	93	92	91	90	89
21	3	3	3	3	3	0	3	/	3	/	0
	88	87	86	85	84	83	82	81	80	79	78
22	0	0	0	0	0	0	0	3	3	3	/
	77	76	75	74	73	72	71	70	69	68	67
23	0	0	0	0	3	/	3	/	3	/	0
	66	65	64	63	62	61	60	59	58	57	56
24	0	0	0	0	0	0	0	0	0	0	0
	55	54	53	52	51	50	49	48	47	46	45
25	0	0	/	0	0	0	/	3	/	3	/
	44	43	42	41	40	39	38	37	36	35	34
26	3	/	3	/	3	/	3	/	/	0	0
	33	32	31	30	29	28	27	26	25	24	23
27	0	/	3	/	3	/	3	/	3	/	3
	22	21	20	19	18	17	16	15	14	13	12
28	/	3	/	0	0	0	0	0	3	/	/
	11	10	9	8	7	6	5	4	3	2	1
29	/	/	/	3	/	3	3	3	3	3	3

PUNCHED CARD PREPARATION



STORAGE DISPLAY

LOC.	#	±	1	2	3	4	5	6	7	8	9	0	REMARKS
1000			0	0	0	N	N	N	N	N	N	N	EMPLOYEE NUMBER
1001			2	A	A	A	A	A	A	A	A	A	NAME
1002			2	A	A	A	A	A	A	A	A	A	
1003			2	A	A	A	A	A	A	A	A	A	
1004			2	0	0	0	0	0	0	A	A	A	DEPARTMENT CODE
1005			0	0	0	0	0	0	0	N	N	N	STANDARD RATE
1006			0	0	0	0	0	0	0	N	N	N	O.T. RATE
1007			0	N	N	0	0	0	0	0	0	0	DEPARTMENT NO.
1008		±	0	0	0	0	N	N	N	0	0	0	INSURANCE PREMIUM
1009			0	0	0	0	0	0	0	0	0	0	INSERT ZEROS
1010		±	0	0	0	N	N	N	N	N	N	N	Y.T.D. GROSS
1011		±	0	0	0	N	N	N	N	N	N	N	Y.T.D. NET
1012			0	0	0	0	N	A	A	A	A	A	CODE

DIGIT CHECK FOR ACTIVE SEGMENT

INPUT	OUTPUT
NUMBER OF 0's = 63	NUMBER OF 0's = _____
NUMBER OF 1's = 76	NUMBER OF 1's = _____
NUMBER OF 2's = 4	NUMBER OF 2's = _____
NUMBER OF 3's = 84	NUMBER OF 3's = _____
SUM* = 227	SUM* =
$\frac{0's + 1's + 2's}{11}$ = WORDS PER CARD	$\frac{0's + 1's + 2 \times 2's}{2}$ = COLUMNS PRINTED OR PUNCHED
$\frac{63+76+4}{11} = 13$	_____ =
$\frac{1's + 3's}{2}$ = CARD COLUMNS	$\frac{1's + 2's + 3's}{11}$ = WORDS
$\frac{76+84}{2} = 80$	_____ =

*If the active segment starts in format band digit position 1, the sum of 0's, 1's, 2's, and 3's should be identical with the format band digit position containing the last digit of the active segment.

Figure 6-2. Format-Band Coding Form (Input).

BAND NO: 1 TITLE: SAMPLE PROBLEM

UNIT NO: 1 ROUTINE:

INPUT: ✓ CODER: WALDRIP-SOLT

OUTPUT: DATE:

REMARKS: PAYROLL ADJUSTMENT CARD

INPUT:

FORMAT BAND DIGITS	EDITING FUNCTION	CORE REGISTER SHIFTS	INFORMATION TRANSFER	D, IB REGISTERS SHIFT
0	INSERT 0	NO	NO	YES
1	TRANSFER DIGIT	YES	YES	YES
2	INSERT 2	NO	NO	YES
3	DELETE DIGIT	YES	NO	NO

TYPE OF CARD COLUMN	ABBR	FORMAT BAND DIGITS	EDITED RESULTS IN WORD
ALPHABETIC OR	A	11	TRANSLATED INTO TWO-DIGIT CODE.
SPECIAL CHARACTER	D	33	CHARACTER DELETED.
NUMERIC	N	31	TRANSLATED INTO SINGLE DIGIT.
	A	11	TRANSLATED INTO TWO-DIGIT CODE; DIGIT PRECEDED BY INSERTED 8.
	D	33	DIGIT DELETED.
BLANK	A	11	TRANSLATED AS 00.
	D	33	COLUMN DELETED.
NONE		0	0 INSERTED; NO EFFECT ON CARD CONTENTS.
		2	2 INSERTED; NO EFFECT ON CARD CONTENTS.

NOTES:

1. Always start the active segment of the editing control stream in digit position 1 on the format band.
2. The inactive segment follows the active segment and contains 3's, except as noted below.
3. Eleven 0's are needed following the active segment to shift the last word out of the D register and into storage. These 0's may appear anywhere in the inactive segment (usually they should occupy digit positions 309 through 319).

OUTPUT:

FORMAT BAND DIGITS	EDITING FUNCTION	D, IB REGISTERS SHIFT	INFORMATION TRANSFER	CORE REGISTER SHIFTS
0	INSERT BLANK	NO	NO	YES
1	TRANSFER ALPHA	YES	YES	YES
2	TRANSFER NUMERIC	YES	YES	YES
3	DELETE DIGIT	YES	YES	NO

TYPE OF OUTPUT	ABBR	FORMAT BAND DIGITS	PUNCHED OR PRINTED RESULTS
ALPHABETIC OR	A	11	TWO-DIGIT CODE TRANSLATED AS ALPHABETIC OR SPECIAL CHARACTER.
SPECIAL CHARACTER		33	TWO DIGITS DELETED.
NUMERIC	N	2	ONE DIGIT TRANSLATED NUMERICALLY.
	SN	01	1 THROUGH 9 TRANSLATED NUMERICALLY; 0 TRANSLATED AS BLANK.
	SZ	10	OVERPUNCH SIGN WITH NO DIGIT UNDERPUNCH.
	A	11	OVERPUNCH SIGN WITH DIGIT UNDERPUNCH.
	D	3	ONE DIGIT DELETED.
BLANK	B	00	BLANK COLUMN PRODUCED.

NOTES:

1. Always start the active segment of the editing control stream with digit position 1 corresponding to the numeric part of column 120. Always write a format band as if the output contained 120 printing or punching positions.
2. The inactive segment follows the active segment and contains 3's.
3. The editing control stream digit pair 01 does not always produce the same effect as the digit 2.

proceeding right to left to 319. In these squares, the editing control stream digits are entered, in right-to-left order exactly as they will appear in the format band on the buffer drum. For every column in the card format (which will appear in the information band on the buffer drum), there must be corresponding editing digits in the format band. The functions of these editing control stream digits can probably be demonstrated most clearly by showing the word-for-field correspondence between the format band and the card format.

Alphabetic, numeric, and alphanumeric information as well as deleted digits in the punched card are transferred by pairs of digits in the format band: one digit to transfer the numeric portion of the column, and the next digit to transfer the zone portion of the column. (The exceptions among the format digits are 0's and 2's. A 0 will insert a 0 in the information band; a 2 will insert a 2.) The card columns are edited by the control stream digits as follows:

According to the card format established on page 6-9 for the sample problem, the three digits in card columns 78—80 are transferred by the pairs of 3's which occupy format-digit positions 1—6. The digit pair 33 deleted column 80 of the card. The 3 in position 1 of the format band deleted the numeric portion of column 80, and the 3 in position 2 of the format band deleted the zone portion of column 80. The digit pairs 33 in format-band positions 3—6 deleted columns 79 and 78 in the same fashion.

The first card field contains alphanumeric information and occupies columns 77—74. The first digit to be transferred is numeric and is located in column 77; the format digit 1 in position 7 transfers the numeric portion of the column; the digit 3 in position 8 deletes the zone portion of the card. The digit pair 31 in the format band, then, is required to transfer one numeric digit to the information band. The characters in columns 76 and 75 are alphabetic; since alphabetic characters appear in Data Processor code as two decimal digits (for example A = 41, B = 42), the digit pair 11 will be required to transfer these digits to the information band. Thus, the digits in positions 9 through 12 on the format band will transfer card columns 76 and 75: the 1 in position 9 will transfer the numeric portion of column 76, and the 1 in position 10 will transfer the zone portion of that column. In the same fashion, the digit pair 11 in format band positions 11 and 12 will transfer the information in column 75. As previously described, the digit pair 31 in positions 12 and 14 will transfer the numeric information in column 74.

Format-band positions 15 through 19 are filled with zeros; in the first card field there are only four columns of information, which are transferred to the information band by the format digits in positions 7 through 14. In core storage, however, eleven digit positions (ten digits plus sign) must be filled in order to form a complete Data Processor word. Since there is a one-for-one transfer of zeros, that is, a zero in the format band will transfer a zero to the information band, the programmer need only insert as many zeros as are required to complete the Data Processor word. This word was assigned to storage location 1012 and has the configuration:

1012 00000NAAAAN

The second card field, the year-to-date net, occupies card columns 73 through 67 with a sign overpunch (a 1 or X punch in column 67). The numeric information in the card columns is transferred under control of the format-digit pairs 31 in positions 20 through 32. Note, however, that although a 1 digit is in position 32 of the format band, position 33 contains a 0. So, also, do positions 34 and 35. In this card field, the card format calls for an overpunch in column 67 to supply the sign digit in the Data Processor word. The 1 digit in format position 32 accommodates the numeric portion of column 67; the 0's in positions 33—35 fill in the remainder of the Data Processor word and delay the transfer of the zone portion of column 67 which is brought in at sign time under control of the 1 digit in position 36 of the format band. The next card field, the year-to-date gross, is transferred in exactly the same fashion by the format digits in positions 37—53.

The eleven 0's in format positions 54—64 insert an equal number of 0's in the information band on the buffer drum; these, in turn, are transferred to core storage where they are stored as a complete word of 0's in location 1009.

The insurance premium, including numeric sign, occupies card columns 59—56. However, to obtain the desired scale factor in the Data Processor word, the three 0's in format positions 65, 66 and 67 cause 0's to be inserted in the three least-significant-digit positions of storage location 1008. The next three digit pairs 31 transfer the numeric information in card columns 59—57 which occupy digit positions 5, 6, and 7 in location 1008. The 0's in format positions 73—77 transfer an equal number of 0's to the four high-order positions of the Data Processor word; then, the digit pair 31 in positions 78 and 79 transfers a 1 or 0 punch from the numeric portion of column 56 to the sign position of this location.

Column 55 of the punched card is deleted by the digit pair 33 in format positions 80 and 81. The following word, the department number, is transferred, in the manner previously described, by the digit pairs 31 in positions

90—93 of the format band. The 0's are supplied in the normal fashion. Card columns 52 through 34, however, are deleted by the digit pairs 33 in format positions 95—132; these deletions do not affect, in any way, the operation of the CARDATRON System, and the next card fields in order are transferred in regular sequence to core storage.

The card fields labelled "Dept." and "Name" both contain alphabetic information: the digit-pairs 11 in positions 151—203 effect the necessary transfer of these alphabetic characters to the information band. Since two numeric are required to carry one alphabetic, twice as many digit positions in core storage will be occupied, as there are columns in the card format. Thus, the employee's name appears in core storage as three Data Processor words, occupying locations 1003, 1002, and 1001 in that order. The 2's in the sign positions of these words were transferred directly by the 2's in the format band positions 171, 182, 193, and 204.

In the prescribed manner, the employee number in the last card field is transferred to storage location 1000 by the digit pairs 31 in format positions 205—220. Card columns 1 and 2 are deleted by the digit-pairs 33 in positions 224—227; these digits are a function of the editing control stream, and are not connected with the inactive segment of the format band.

THE INACTIVE SEGMENT

Positions 228—319 of the format band comprise what is known as the inactive segment. It is that portion of the 319 available format digits which is not used in the editing control stream. It is filled in entirely with the digit 3, except for the last eleven positions which contain 0's.

The section labelled PUNCHED CARD PREPARATION contains provisions in columns 4 through 14 for entering "bootstrap" instructions. The four least-significant-digits, columns 11 through 14, contain the addresses of the storage locations where the contents of the next card will be stored. Columns 9 and 10 contain the operation code: in this case, a 60, which is a CARD READ (CRD) instruction. Column 5 contains the unit-designate digit of the control portion of the instruction; this instruction has designated Input Unit 1. A 6 occupies the sign position in column 4; under control of this sign digit, the instruction is transferred to the C register for immediate execution. The 6 in column 1 signifies that the editing control stream is assigned to format-band 6, that is, on each of the cards containing format digits, a 6 punch in column 1 is used for selecting numeric format. The format-band words are stored in descending order in storage locations 1129 through 1101. The last instruction in this section, 6 0000 Op aaaa, is used as a "bootstrap" to continue. If, for example, there are additional cards to be read, it may be a CARD READ or a CARD READ, FORMAT LOAD instruction; or it may be a BRANCH UNCONDITIONALLY (BUN) instruction if control is to be turned over to a stored program.

The left-most section of the format-band coding form is labelled PAPER-TAPE PREPARATION, and shows the format-digit layout for paper-tape input. Note that the format digits enter in descending order beginning with position 1 at the bottom of the column. Just as in the punched card, the editing control digits for the first information field form word number 29, and proceed through word 1 which is the final word of 0's. The unnumbered 11-digit word positions at the top and bottom of the form are for paper-tape read and transfer-control instructions respectively.[4]

In the lower right-hand corner of the coding form, a section is provided for checking the digit count of the active segment of an input format band. Whereas this count cannot guarantee complete accuracy, it is an aid in detecting clerical errors.

OUTPUT UNIT OPERATION

The Output Unit, like the Input Unit, is made up of two essential parts, control circuitry and a magnetic drum (the buffer). On the drum are six bands for storing information. One band, called the information band, is used to store information transferred from the Data Processor to the card-handling machine. The remaining five bands, called format bands, are used to contain editing control streams. Each of the six bands on the buffer can accommodate 319 digits.

Corresponding to each digit position in the information band is a digit position in each of the five format bands. Each of these digit positions is identified uniquely and in order by one of the numbers from 1 to 319. Digit position 1 is regarded as the origin of a band. Whereas each digit position in the information band consists of a full four-bit decade—and hence is capable of storing any one of the decimal digits—each digit position in a format band is comprised of only two positions: it is possible to write only digits 0, 1, 2 and 3 in the format band. As is the case with an input format band, the four bit or the eight bit will be ignored if an attempt is made to write as a format-band digit a decimal digit greater than or equal to four.

[4] A careful study of this paper-tape format section is advised. A 2 in the sign position of any of the 29 words will cause alphabetic translation of each such word as it is read into the Data Processor. This can be prevented by having the operator set the PZT switch, on the left-hand Maintenance Panel of the console, to the UP position. Doing so will ensure numeric translation of all words. (See *Handbook of Operating Procedures for the BURROUGHS 220* for a discussion of the PZT switch.)

INFORMATION TRANSFER

The transfer of information from the Data Processor to a card-handling machine takes place in two phases. During the first phase, information is transferred from the Data Processor to the information band in response to a CARD WRITE instruction. During the second phase, the contents of the information band are transferred to a card-handling machine. This is accomplished, independently of the Data Processor, under control of an Output Unit. After information has been transferred from the Data Processor to the information band, execution of the program is resumed automatically.

Phase 1: From Data Processor to Buffer. (See Figure 6-3.) Each CARD WRITE instruction specifies the format band which is to edit the stream of information to be transferred from the Data Processor.

If an attempt is made to execute another CARD WRITE instruction before the transfer of information from the buffer to the card-handling machine has been completed—that is, before completion of Phase 2 of the information transfer associated with the preceding CARD WRITE instruction—the Data Processor will wait until the Output Unit is in a state of readiness. The status of an Output Unit may be ascertained by executing a CARD WRITE INTERROGATE, BRANCH instruction.

Phase 2: From Buffer to Card-Handling Machine. (See Figure 6-3.) After the information band has been loaded by the execution of a CARD WRITE instruction, its contents are transferred to the card-handling machine. With the completion of this transfer, the final editing, the deletion of unwanted digits and conversion to card code, is completed.

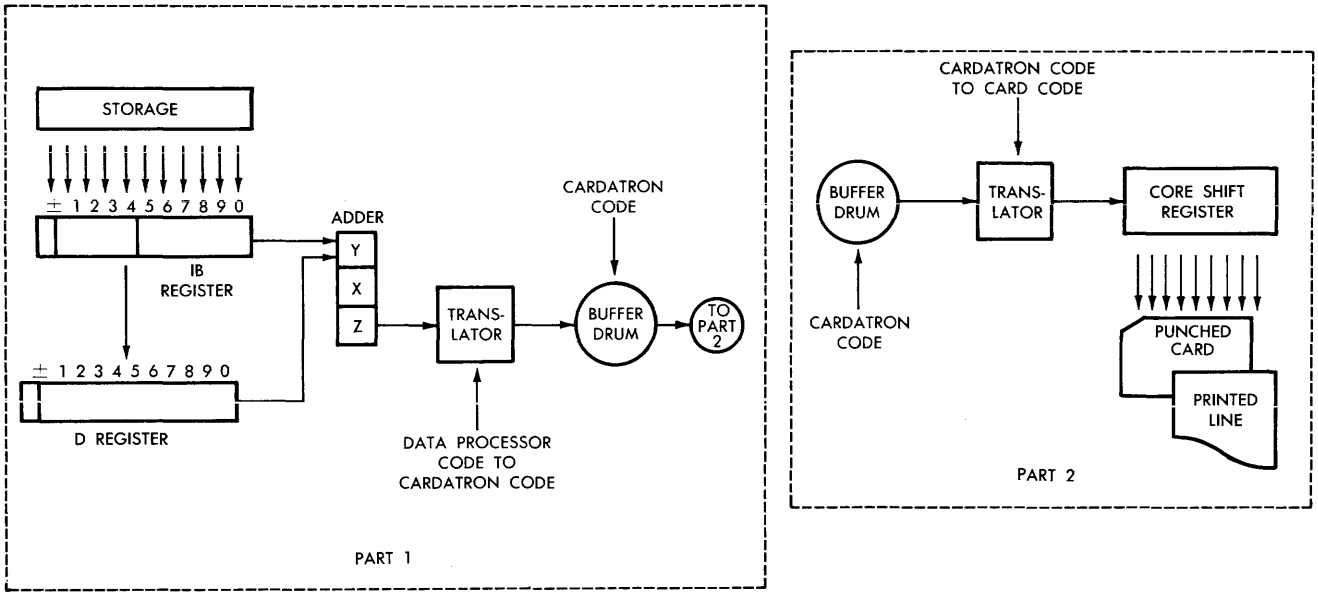


Figure 6-3. Output information path.

Up to 28 full words can be transferred from sequentially addressed storage locations in the Data Processor. The first digit transferred is taken from the storage location specified by the address portion of the CARD WRITE instruction. The remaining digits are transferred from core storage locations in order of sequentially descending addresses.

The editing control stream on the format band selected by the CARD WRITE instruction performs a part of the editing process when the information is transferred to the information band: 0's are inserted. In addition, part of the conversion from Data Processor code to card code is accomplished.

The purpose of Phase 2 of the output cycle is to present information from the buffer to the card-handling machine in the proper sequence for printing or punching. The conversion of information from serial presentation to parallel presentation takes place when the contents of the output Core Shift Register are transferred to the card-handling machine.

As each digit is transferred from the information band it is set into a decade of comparison toggles. When the toggles are set, their value is compared with the value of the Output Unit's row counter setting; at the same time the corresponding editing control stream digit in the format band is examined. Three cases need to be considered:

1. Each time there is agreement between the row counter and the comparison toggles the input position—the first core—of the Core Shift Register[5] is set and the entire Core Shift Register is shifted one place to the right, except when the editing control stream digit is a 3.
2. When the editing control stream digit is a 3, the corresponding digit in the information band is deleted by the simple expedient of not setting and not shifting the Core Shift Register.

Table 6-2. Row Counter settings, CARDATRON 220 Output Unit

9-Edge-First Machine		12-Edge-First Machine	
Row Counter Setting	**Significance**	**Row Counter Setting**	**Significance**
13	Print 12 row; buffer unloaded, ready for Data Processor.	10	Punch 9 row; buffer unloaded, ready for Data Processor.
12	Erase information band if CWR. Erase format band if CWF.	11	Erase information band if CWR. Erase format band if CWF.
11	Transfer information from Data Processor to buffer.	12	Transfer information from Data Processor to buffer.
10	Buffer loaded, ready for printer.	13	Buffer loaded, ready to punch.
9	Transfer 9's to cores.	14	Transfer 12's to cores.
8	Transfer 8's to cores. Print 9 row.	15	Transfer 11's to cores. Punch 12 row.
7	Transfer 7's to cores. Print 8 row.	0	Transfer 0's to cores. Punch 11 row.
6	Transfer 6's to cores. Print 7 row.	1	Transfer 1's to cores. Punch 0 row.
5	Transfer 5's to cores. Print 6 row.	2	Transfer 2's to cores. Punch 1 row.
4	Transfer 4's to cores. Print 5 row.	3	Transfer 3's to cores. Punch 2 row.
3	Transfer 3's to cores. Print 4 row.	4	Transfer 4's to cores. Punch 3 row.
2	Transfer 2's to cores. Print 3 row.	5	Transfer 5's to cores. Punch 4 row.
1	Transfer 1's to cores. Print 2 row.	6	Transfer 6's to cores. Punch 5 row.
0	Transfer 0's to cores. Print 1 row.	7	Transfer 7's to cores. Punch 6 row.
15	Transfer 11's to cores. Print 0 row.	8	Transfer 8's to cores. Punch 7 row.
14	Transfer 12's to cores. Print 11 row.	9	Transfer 9's to cores. Punch 8 row.

[5] The output Core Shift Register is comprised of 120 magnetic cores that are connected to the 120 FROM CARDATRON hubs on the plugboard panel of a line printer; or, 80 of the cores are connected to the 80 FROM CARDATRON hubs on the plugboard panel of a card punch. If the Output Unit is connected to a punch, cores numbered 1 through 80 in the Core Shift Register will have been cable-connected to the punch.

3. When the row counter setting and the comparison toggles do not agree, the Core Shift Register is shifted one place to the right without setting the input position, except when the editing control stream digit is a 3. In the latter case, the effect is as described in 2.

Thus, for each row counter setting the information band is scanned for digits corresponding in value to the row counter setting (see Table 6-2): the configuration of such digits in the information band is represented by the set positions in the Core Shift Register, with the exception of digits that have been deleted. At this time a digit impulse is emitted from the card-handling machine. The purpose of this impulse is to so synchronize the Output Unit and the card-handling machine that information corresponding to the set positions of the Core Shift Register can be transferred to the card-handling machine. The value of this information is determined by the digit time of the card-handling machine's cycle during which this transfer occurs.

Each position in the Core Shift Register is connected to and thereby controls, by means of control panel wiring, the energizing of a corresponding punch or print magnet in the card-handling machine. If the machine is a punch, all the positions in the card that are to be punched in a given row will be punched at the time the information in the cores is transferred.

If the card-handling machine is a printer, each of the bits of information in the Core Shift Register sets the corresponding type wheel each time the digit pulse is emitted. When all of the information has been transferred from the information band, the line printer will print one line.

DESIGNATION OF FORMAT BANDS

Each of the five format bands on the buffer of an Output Unit is identified uniquely by one of the digits from 1 to 5. No special format bands are used by the Output Unit. All of the format bands are under program control; that is, the entire contents of a designated format band may be changed by executing a CARD WRITE, FORMAT LOAD instruction.

FORMAT BAND SELECTION

The desired format band is designated by a digit in the CARD WRITE or CARD WRITE, FORMAT LOAD instruction.

INSTRUCTIONS

CARD WRITE (CWR) is an instruction that transfers up to 28 words from sequentially addressed storage locations in the Data Processor to the information band of a designated Output Unit. The instruction also designates the format band whose contents are to edit the information transferred. In addition, it must be specified whether the Suppress-12 mode is elected (see page 6-25). The first digits transferred are taken from the storage location specified by the address part of the CARD WRITE instruction. The remaining digits are taken from storage locations in order of sequentially *descending* addresses.

CARD WRITE, FORMAT LOAD (CWF) is an instruction that provides the ability to load the editing control stream digits onto the buffer drum of an Output Unit. It transfers 319 digits, stored in 29 consecutively addressed storage locations, to the designated format band of a designated Output Unit. The contents of the location specified by the address part of the CARD WRITE, FORMAT LOAD instruction will occupy the first 11 digit positions on the format band. Suppose the instructions were CWF 1249. Then (1249:01) goes to format-band digit position 1; (1249:91) goes to format-band digit position 2; . . . ; and (1249:±1) goes to format-band digit position 11. The word in location 1248 will occupy format-band digit positions 12 through 22. And so forth. Finally, (1221:09) will occupy the last nine digit positions on the format band.

CARD WRITE INTERROGATE, BRANCH (CWI) is an instruction that permits the program to determine if a designated Output Unit is ready to be used. If the unit queried is ready, control is transferred, that is, the contents of the P register are replaced by the address part of the CWI instruction; otherwise, control continues in sequence.

FUNCTIONS OF OUTPUT FORMAT-BAND DIGITS

Processed material passes through the two output phases under control of an editing control stream, just as it must during input. There are, however, basic differences between the activities of the input and output editing control streams; a proper understanding of the functions of the output format bands must necessarily include an understanding of the separate digits which comprise them. It is the purpose of this section to describe and exemplify (see Figure 6-4), the functions of these digits.

INSERT BLANK DIGIT (0)

The primary function of the 0 digit is the insertion of a blank in a portion of a card column. A single 0 will accommodate either the numeric or the zone portion of the column; to insert an entire blank column in the output card, or a blank print position, the digit pair 00 must be entered in the corresponding format-band digit positions. Assume, for example, that a Data Processor word contains digits for the alphabetic characters A, B, C, D and E, and the program calls for a printout of this particular word with blank print positions following the second and fourth characters:

Data Processor word	41	42		43	44		45
Format-band digits	11	11	00	11	11	00	11
Printed line	A	B	b*	C	D	b*	E

* b = Blank column

The numeric characters in the Data Processor word were transferred by the digit pair 11, and translated into their alphabetic equivalents by the line printer. The digit pairs 00 forced the blank spaces between B and C, and D and E.

TRANSFER-ALPHANUMERIC DIGIT (1)

The format digit 1 will transfer either the numeric or zone portion of a card column of alphabetic information. The transfer of an entire card column, then, must be effected by the use of the digit pair 11, since alphabetic characters are represented in Data Processor code by the decimal digit pairs 41, 42, and so forth. The right-hand format digit 1 causes the digit in the corresponding position on the information band to be transferred with numeric interpretation; the next digit of the transfer pair is given zone interpretation. If the format digit pair 01 is used to edit a given card column, only the numeric portion of that column will be transferred; the position on the information band corresponding to the zone portion will remain blank. If the transfer pair 10 is used, only the zone portion will be transferred, and the numeric portion will remain blank.

Data Processor word	1	3	41
Format-band digits	10	01	11
Printed line or punched column	Print or Punch 11 (over punch)	3	A

TRANSFER NUMERIC DIGIT (2)

Format digit 2 will transfer a single numeric character from core storage to its corresponding position on the information band. This digit will, by itself, accommodate the information at both zone and numeric time and will effect the complete transfer of any numeric digit from 0 through 9. A 2 will not transfer decimal-digit pairs or alphabetic characters, this can be achieved only under control of the digit pair 11; note, however, that a 0 is regarded as any other decimal digit and can be transferred only by a 2, and not by a format digit 0 as is the case on input. For example, if a printout of a word containing both alphabetic and numeric information is desired:

Data Processor word	1	42	43	5	7	9	4	0	3
Format-band digits	2*	11	11	2*	2*	2*	2*	2*	2*
Printed line	1	B	C	5	7	9	4	0	3

* Format digit 2 will translate a 0 as a 0 to both the punch and the printer. The digits 01, however, will translate a 0 as a blank.

DELETE DIGIT (3)

The format digit 3 will delete the digit in the corresponding position in the Data Processor word. A single 3 will delete a single Data Processor digit and has no effect on the print position or on the punched-card column. It will not, however, delete such information as alphabetic characters or pairs of decimal digits both of which would occupy a single card column. To effect such a deletion, the digit pair 33 must be used in the format band. The following example illustrates the use of the delete digit.

Data	1	4	9	2	A	D	3	7	0
Data Processor word	1	4	9	2	41	44	3	7	0
Format-band digits	2	3	2	2	33	33	2	2	3
Printed line*	1		9	2			3	7	

* In the actual printout, this line will appear as: 1 9 2 3 7; the delete digit does not leave spaces in the printed line.

The desired numeric characters are transferred by the format digit 2. The digits 4 and 0 in the Data Processor word are deleted by the corresponding 3's in the format band; the digit pairs 41 and 44 in the Data Processor word are deleted by the format digit pairs 33.

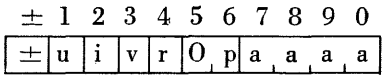
THE EXECUTE PHASE

The remainder of the CARDATRON System chapter will be devoted to a discussion of the Execute Phase of the CARDATRON instructions.

CARD READ (CRD)

Operation Code. 60

Instruction Format.



Definitions.

u: designates Input Unit.

Burroughs Corporation **BURROUGHS 220** **CARDATRON FORMAT BAND CODING FORM**

INPUT ☐ BAND NO. 5

OUTPUT ☒ UNIT NO. 3

TITLE: EXAMPLE : OUTPUT ROUTINE NO. ________ CODER: F. BAUER & R. REITMEYER

Field		1		3		4		5		6		7		8		9		10
Code	BLANK	S ZWWWWWWWWWWW	BLANK	AAWW	BLANK	WWWWW	BLANK	WWWWWW	BLANK	SSSS ZWWWMAWAW	BLANK	AAAAA	BLANK	AAAAA	BLANK	WWWWWWWWWWW	BLANK	S ZWWWWWWWWWW
Positions	1–7	8–19	20–29	30–33	34–38	39–43	44–48	49–54	55–59	60–68	69–73	74–78	79–83	84–88	89–93	94–104	105–109	110–120

PAPER-TAPE PREPARATION

#	Positions	±	1	2	3	4	5	6	7	8	9	0
1	319–309	0	0	0	3	3	3	3	3	3	3	3
2	308–298	3	3	3	3	3	3	3	3	3	3	3
3	297–287	3	3	3	3	3	3	3	3	3	3	3
4	286–276	3	3	3	3	3	3	3	3	3	3	3
5	275–265	3	3	3	3	3	3	3	3	3	3	3
6	264–254	3	3	3	3	3	3	3	3	3	3	3
7	253–243	3	3	3	3	3	3	3	3	3	3	3
8	242–232	3	3	3	3	3	3	3	3	3	3	3
9	231–221	3	3	3	3	3	3	3	3	3	0	0
10	220–210	0	0	0	0	0	0	0	0	0	0	0
11	209–199	0	0	0	1	0	2	2	2	2	2	2
12	198–188	2	2	2	2	3	3	3	3	3	3	3
13	187–177	3	3	3	3	0	0	0	0	0	0	0
14	176–166	0	0	0	0	0	0	0	0	0	0	0
15	165–155	0	0	3	1	1	1	1	3	3	3	3
16	154–144	2	2	0	0	0	0	0	0	0	0	0
17	143–133	0	2	2	2	3	3	3	3	3	3	2
18	132–122	2	0	0	0	0	0	0	0	0	0	0
19	121–111	2	3	3	3	3	3	2	2	2	2	2
20	110–100	0	0	0	0	0	0	0	0	0	0	1
21	99–89	0	0	1	0	1	0	1	2	1	1	2
22	88–78	1	1	2	0	0	0	0	0	0	0	0
23	77–67	0	0	3	1	1	1	1	1	1	1	1
24	66–56	1	1	0	0	0	0	0	0	0	0	0
25	55–45	0	3	1	1	1	1	1	1	1	1	1
26	44–34	1	0	0	0	0	0	0	0	0	0	0
27	33–23	2	2	2	2	2	2	2	2	2	2	2
28	22–12	0	0	0	0	0	0	0	0	0	0	1
29	11–1	0	2	2	2	2	2	2	2	2	2	2

PUNCHED CARD PREPARATION

Card	Col. 1	Cols. 4–14	Cols. 15–25	Cols. 26–36	Cols. 37–47	Cols. 48–58	Cols. 59–69	Cols. 70–80
(above card 1)		03000601249	24	8 — 25	26	9 — 27	28	10 — 29
1	6	63000601243	11000000000	31111111111	10000000000	22222222222	00000000000	01222222222
(segments)			18	5 — 19	20 — 6	21	22	7 — 23
2	6	63000601237	20000000000	23333322222	00000000001	00101012112	11200000000	00311111111
(segments)			2 — 12	13	14	3 — 15	16	4 — 17
3	6	63000601231	22223333333	33330000000	00000000000	00311113333	22000000000	02223333332
(segments)			6	7	8	9	10	1 — 11
4	6	63000601225	3 —			— 3 (pos. 223)	00000000000	00010222222
(segments)				1	2	3	4	5
5	6		6 0paaaa0003					— 3

STORAGE DISPLAY

LOC.	#	±	1	2	3	4	5	6	7	8	9	0	REMARKS
2017	1	±	N	N	N	N	N	N	N	N	N	N	
2018	2	D	D	D	D	D	D	D	D	D	D	D	DELETE THIS WORD
2019	3	D	A	A	A	A	D	D	D	D	N	N	
2020	4	±	N	N	D	D	D	D	D	D	N	N	
2021	5	N	D	D	D	D	D	N	N	N	N	N	
2022	6	±	N	N	N	N	A	A	N	A	A	N	
2023	7	D	A	A	A	A	A	A	A	A	A	A	
2024	8	D	A	A	A	A	A	A	A	A	A	A	
2025	9	±	N	N	N	N	N	N	N	N	N	N	
2026	10	±	N	N	N	N	N	N	N	N	N	N	

DIGIT CHECK FOR ACTIVE SEGMENT

	INPUT	OUTPUT
NUMBER OF 0's —		112
NUMBER OF 1's —		34
NUMBER OF 2's —		47
NUMBER OF 3's —		29
SUM* —		222

INPUT: $\frac{0\text{'s} + 1\text{'s} + 2\text{'s}}{11}$ — WORDS PER CARD

OUTPUT: $\frac{0\text{'s} + 1\text{'s} + 2 \times 2\text{'s}}{2}$ — COLUMNS PRINTED OR PUNCHED

$\frac{112+34+94}{2} = 120$

INPUT: $\frac{1\text{'s} + 3\text{'s}}{2}$ — CARD COLUMNS

OUTPUT: $\frac{1\text{'s} + 2\text{'s} + 3\text{'s}}{11}$ — WORDS

$\frac{34+47+29}{11} = 10$

*If the active segment starts in format band digit position 1, the sum of 0's, 1's, 2's, and 3's should be identical with the format band digit position containing the last digit of the active segment.

Figure 6-4. Format-Band Coding Form (Output).

BAND NO: 5 TITLE: EXAMPLE: OUTPUT REMARKS:

UNIT NO: 3 ROUTINE:

INPUT: ___ CODER: BAUER-REITMEYER

OUTPUT: ✓ DATE:

INPUT:

FORMAT BAND DIGITS	EDITING FUNCTION	CORE REGISTER SHIFTS	INFORMATION TRANSFER	D, IB REGISTERS SHIFT
0	INSERT 0	NO	NO	YES
1	TRANSFER DIGIT	YES	YES	YES
2	INSERT 2	NO	NO	YES
3	DELETE DIGIT	YES	NO	NO

TYPE OF CARD COLUMN	ABBR	FORMAT BAND DIGITS	EDITED RESULTS IN WORD
ALPHABETIC OR SPECIAL CHARACTER	A	11	TRANSLATED INTO TWO-DIGIT CODE.
	D	33	CHARACTER DELETED.
NUMERIC	N	31	TRANSLATED INTO SINGLE DIGIT.
	A	11	TRANSLATED INTO TWO-DIGIT CODE; DIGIT PRECEDED BY INSERTED 8.
	D	33	DIGIT DELETED.
BLANK	A	11	TRANSLATED AS 00.
	D	33	COLUMN DELETED.
NONE		0	0 INSERTED; NO EFFECT ON CARD CONTENTS.
		2	2 INSERTED; NO EFFECT ON CARD CONTENTS.

NOTES:

1. Always start the active segment of the editing control stream in digit position 1 on the format band.
2. The inactive segment follows the active segment and contains 3's, except as noted below.
3. Eleven 0's are needed following the active segment to shift the last word out of the D register and into storage. These 0's may appear anywhere in the inactive segment (usually they should occupy digit positions 309 through 319).

OUTPUT:

FORMAT BAND DIGITS	EDITING FUNCTION	D, IB REGISTERS SHIFT	INFORMATION TRANSFER	CORE REGISTER SHIFTS
0	INSERT BLANK	NO	NO	YES
1	TRANSFER ALPHA	YES	YES	YES
2	TRANSFER NUMERIC	YES	YES	YES
3	DELETE DIGIT	YES	YES	NO

TYPE OF OUTPUT	ABBR	FORMAT BAND DIGITS	PUNCHED OR PRINTED RESULTS
ALPHABETIC OR SPECIAL CHARACTER	A	11	TWO-DIGIT CODE TRANSLATED AS ALPHABETIC OR SPECIAL CHARACTER.
		33	TWO DIGITS DELETED.
NUMERIC	N	2	ONE DIGIT TRANSLATED NUMERICALLY.
	SN	01	1 THROUGH 9 TRANSLATED NUMERICALLY; 0 TRANSLATED AS BLANK.
	SZ	10	OVERPUNCH SIGN WITH NO DIGIT UNDERPUNCH.
	A	11	OVERPUNCH SIGN WITH DIGIT UNDERPUNCH.
	D	3	ONE DIGIT DELETED.
BLANK	B	00	BLANK COLUMN PRODUCED.

NOTES:

1. Always start the active segment of the editing control stream with digit position 1 corresponding to the numeric part of column 120. Always write a format band as if the output contained 120 printing or punching positions.
2. The inactive segment follows the active segment and contains 3's.
3. The editing control stream digit pair 01 does not always produce the same effect as the digit 2.

v: variation designator:

v = 0: Input control words will be recognized as such.

v = 1: Input control words will not be recognized as such.

r: r = 8: designated input will be B-register address-modified.

r = 0: no B-register address-modification of input.

r = 9: designated input will be B-register address-modified; Reload Lockout will be imposed.

r = 1: no B-register address-modification of input; Reload Lockout will be imposed.

aaaa: address of base of location into which is written first word transferred from the information band.

Description of Operation. Transfer the contents of the information band from Input Unit u to core storage. The first 11 digits transferred from the information band are stored in location B[aaaa]; the next 11 digits transferred are stored in location B[aaaa] − 1; the next 11 digits transferred are stored in location B[aaaa] − 2; and so forth. Therefore, the card fields from right to left are in core storage locations in ascending order. The editing control stream used to edit the contents of the information band is the one on the format band which was selected by the card whose contents are on the information band.

If r = 1 or 9, Reload Lockout will be imposed, that is, the transfer to the information band of the contents of the next card in the card reader will be inhibited.

If r = 0 or 8, Reload Lockout will be released, if it had been imposed.

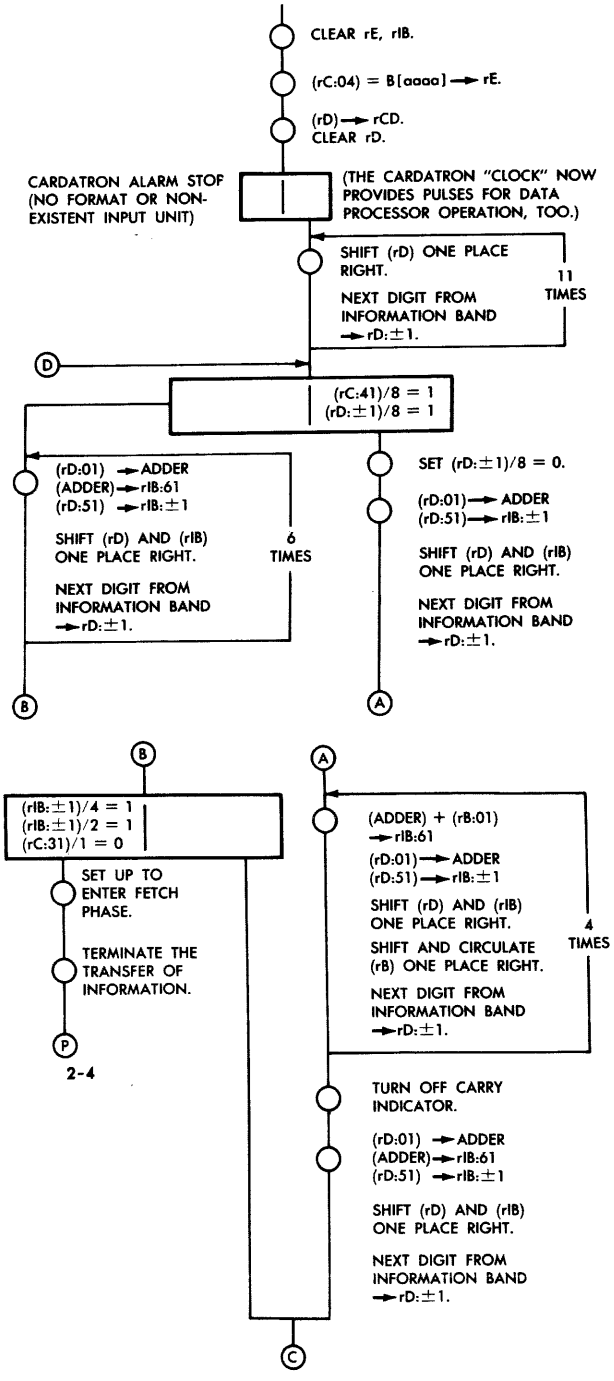
Table 6-3. CARDATRON input sign control

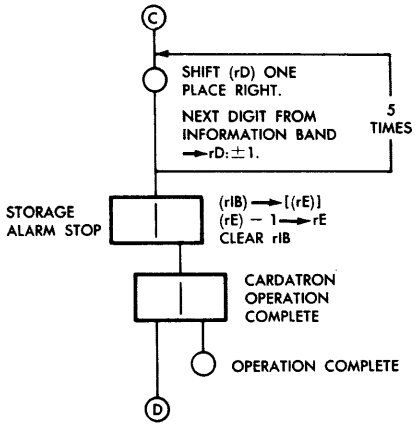
Sign Digit	Where Sensed	r = (rC:41)	v = (rC:31)	Effect
0	rD	Any	Any	None. (Sign digit stored as 0.)
1	rD	Any	Any	None. (Sign digit stored as 1.)
2	rD	Any	Any	None. (Sign digit stored as 2.)
3	rD	Any	Any	None. (Sign digit stored as 3.)
4	rD	Any	Any	None. (Sign digit stored as 4.)
5	rD	Any	Any	None. (Sign digit stored as 5.)
6	rIB	Any	0	This is a control word: terminate the transfer of information from the information band; release the Input Unit to read the next card; enter the Fetch Phase at connector P (see page 2-4). The control word is then treated as an instruction (because the sign digit is even, there will be no B-register address-modification).
			1	None. (Sign digit stored as 6.)
7	rIB	Any	0	Except that B-register address-modification will occur during the Fetch Phase — because the sign digit is odd — the effect is the same as that when the sign digit is 6 and v = 0.
			1	None. (Sign digit stored as 7.)
8	rD	≠ 8, ≠ 9	Any	None. (Sign digit stored as 8.)
		8, 9	Any	B-register address-modification occurs; the sign digit is stored as 0.
9	rD	≠ 8, ≠ 9	Any	None. (Sign digit stored as 9.)
		8, 9	Any	B-register address-modification occurs; the sign digit is stored as 1.

If Reload Lockout was imposed by the card whose contents are being transferred by this CRD instruction, the execution of this CRD instruction will set Reload Lockout. (See page 6-6.) The contents of the card following the one which imposed Reload Lockout will not be automatically transferred to the information band after completion of this CRD instruction. Another CARDATRON cycle—the execution of either a CRD or CRF instruction—is required to release Reload Lockout.

Input sign-control is effected as described in Table 6-3.

Flow Chart.





Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address. (See Remark 8.)
2. CARDATRON ALARM STOP due to designating a nonexistent input unit.
3. Digit Check ALARM STOP. (See Remark 9.)
4. No Format ALARM STOP on Input Unit control panel.

Remarks.

1. r = 3, 5, or 7 has the same effect as r = 1; r = 2, 4, or 6 has the same effect as r = 0.
2. v = 2, 4, 6, or 8 has the same effect as v = 0; v = 3, 5, 7, or 9 has the same effect as v = 1.
3. The inactive segment of a format band must contain eleven 0's. (Usually these 0's will occupy format-band digit positions 309 through 319.) If the inactive segment does not contain these eleven 0's, the last word transferred from the information band will not be written in core storage: it will be lost to the program.
4. If Reload Lockout is imposed, it is done by selecting format band 8. Format band 8 must be selected in conjunction with the selection of some other format band: if it is not, a No Format ALARM STOP will occur on the Input Unit control panel.
5. Once Reload Lockout is imposed, it can be removed only by specification in a CRD or CRF instruction: removal of Reload Lockout is achieved by making r an even digit.
6. At the beginning of the Execute Phase of a CRD instruction B[aaaa] = (rC:04) is transferred to the E register. The address of the core storage location into which will be written each successive word transferred from the information band is obtained by decreasing by 1 the contents of the E register after each word is stored.

Location	Contents of numeric part of card column stored in digit position										
	±	1	2	3	4	5	6	7	8	9	0
1013	70	71	72	73	74	75	76	77	78	79	80
1012	59	60	61	62	63	64	65	66	67	68	69
1011	48	49	50	51	52	53	54	55	56	57	58
1010	37	38	39	40	41	42	43	44	45	46	47
1009	26	27	28	29	30	31	32	33	34	35	36
1008	15	16	17	18	19	20	21	22	23	24	25
1007	4	5	6	7	8	9	10	11	12	13	14
1006		.....			inserted 0's	.....			1	2	3
1005		.....			inserted 0's	.....					
1004		.....			inserted 0's	.....					
1003		.....			inserted 0's	.....					
1002		.....			inserted 0's	.....					
1001		.....			inserted 0's	.....					

Since the E register counts modulo the size of storage, care must be exercised in the choice of B[aaaa]. For example, suppose the Data Processor has 4000 words of core storage; suppose also that B[aaaa] is chosen so that some word is transferred to location 0000. After this word is stored, 1 will be subtracted from 0000, the contents of the E register, modulo 4000. The E register will then contain 3999.

7. The selection of numeric format, format band 6, allows a maximum of 13 numeric words to be transferred from the information band to core storage when the next CRD instruction is executed: all overpunches will be deleted.

Suppose, for example, that the instruction is 0 1000 CRD 1013; suppose also that the control panel is wired 80-80. The execution of the specified CRD instruction will load storage as displayed in the table above:

If it is desired to prevent the insertion of zeros as shown above, it is necessary to supply the card with a word having a 6 or 7 in the sign-digit position. Suppose, for example, that it is desired to transfer only four words from a card which has selected numeric format; suppose also that the control panel is wired 80-80. Then these four words must occupy the fields comprised of columns 70—80, 59—69, 48—58, 37—47, inclusive. The field comprised of columns 26—36, inclusive, must be an instruction, and must have a 6 or 7 in the sign-digit position (column 26). Let the CRD instruction be as specified above. The execution of that instruction will load storage as displayed below:

8. Once the transfer of information from the information band to the Data Processor has begun, it can terminate only when all of the information has been transferred to the D register. Examination of the flow chart will disclose that the information transfer is already under way when a Storage ALARM STOP can occur (it occurs before the first word is written in core storage). Hence, none of the words will be stored; in other words, the information is "lost," that is, the card whose contents were on the information band must be re-read.

Location	Contents of numeric part of card column stored in digit position										
	±	1	2	3	4	5	6	7	8	9	0
1013	70	71	72	73	74	75	76	77	78	79	80
1012	59	60	61	62	63	64	65	66	67	68	69
1011	48	49	50	51	52	53	54	55	56	57	58
1010	37	38	39	40	41	42	43	44	45	46	47

No other locations in storage will be affected.

9. A Digit Check ALARM STOP can occur after the execution of a CRD instruction only as a result of detecting the impermissible configuration when the attempt is made to shift the digit out of rD:01. Examination of the flow chart will disclose that it is possible to write in core storage a word with an impermissible configuration in some digit position. This can happen because:

   a. The contents of the D register are transferred to the IB register in two parts, the high-order part of the word being split off with rD:51. Hence, if the impermissible configuration first occurs in any digit position in rD:56 before the transfer to rIB of (rD) is begun, it will be transferred to rIB; and

   b. Since the contents of the IB register are transferred to storage in parallel, no check is made in rIB for the impermissible configuration.

Further examination of the flow chart discloses that the contents of the D register are shifted to the right as each additional digit is transferred from the information band to the D register. Some digits are transferred to rD—whose contents are shifted to the right as each digit is entered in rD:±1—while the word in rIB is being written in storage: the transfer to storage is completed before the D register is completely filled with the next word from the information band.

Again it should be noted that once the transfer of information to the Data Processor has begun, it can terminate only when all of the information has been transferred to rD. Thus, digits from the information band will "pile up" in rD:±1 immediately after the impermissible configuration is detected. In no case can the ALARM STOP be prevented from occurring. In any event:

a. It is not possible to predict with certainty what will be in the D register;

b. The card whose contents were on the information band must be re-read; and

c. The E register will provide a clue to the location in storage of the word with the impermissible configuration. Suppose the word is in aaaa: then (rE) = aaaa − 1.

Register Status.

Register name	Contents after execution of CRD
A	Unchanged
R	Unchanged
D	*
B	Unchanged
P	$(rP)_b + 1$
C	u i v r 6 0 B[aaaa]
E	Address of last location filled − 1

Register name	Contents if Storage ALARM STOP occurs
A	Unchanged
R	Unchanged
D	*
B	Unchanged
P	$(rP)_b + 1$
C	u i v r 6 0 B[aaaa]
E	B[aaaa]

* (rD:67) = 0000000; (rD:04) are the four high-order digits of the last word transferred from the buffer.

Register name	Contents if No Format or CARDATRON ALARM STOP occurs
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	u i v r 6 0 B[aaaa]
E	B[aaaa]

Register name	Contents if Digit Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	**
B	Unchanged
P	$(rP)_b + 1$
C	u i v r 6 0 B[aaaa]
E	Address of last location filled − 1

** Depends on when the impermissible configuration is detected. See flow chart.

CARD WRITE (CWR)

Operation Code. 61

Instruction Format.

± 1	2	3	4	5	6	7	8	9	0
±	u	i	c	f	O	p	a	a	a	a

Definitions.

u: designates Output Unit.

c: specifies which of the T RELAYS is/are selected for additional control of the line printer.

f: specifies which format band is used to edit the output; in addition, if

f is even: the Suppress-12 mode of printing or punching is selected.

f is odd: the Suppress-12 mode is not selected.

aaaa: address of base of location from which is taken the first digit transferred to the information band.

Description of Operation. Transfer to the information band of Output Unit u the contents of up to 28 core storage locations as follows: the contents of B[aaaa] are transferred first; (B[aaaa] − 1) are transferred next; . . . ; (B[aaaa] − 27) are transferred last. The number of digits transferred is determined by the editing control stream on the format band which is specified by f.

The format band whose contents are used to edit the information transferred to the information band is specified in the table below:

f	Format Band	Suppress-12 Mode?
0	1	Yes
1	1	No
2	2	Yes
3	2	No
4	3	Yes
5	3	No
6	4	Yes
7	4	No
8	5	Yes
9	5	No

The Suppress-12 mode has to do with overpunching the sign digits of numeric words, that is, words whose signs are either 0 or 1, in accordance with either of the two conventions for overpunching signs. The Suppress-12 mode also has to do with printing the sign digits of numeric words (in the tables below it is assumed that the printer is a 407 with Type A print wheels).

Two cases need to be considered:

Case 1: the sign digit is translated by the editing control stream digit pair 10. In this case the sign digit occupies a column or print position by itself.

Suppress-12 Mode?	± = 0		± = 1	
	Punch	Print	Punch	Print
Yes	Blank	Blank	11	—
No	12	&	11	—

Case 2: the digit preceding the sign digit and the sign digit are translated by the editing control stream digit pair 11. In this case the sign digit shares a column or print position with the digit which precedes it.

Suppress-12 Mode?	± = 0		± = 1	
	Punch	Print	Punch	Print
Yes	Blank	a	11	b
No	12	c	11	b

a. The digit preceding the sign digit.

b. The alphabetic character whose code is 11−n, where n is the digit preceding the sign digit.

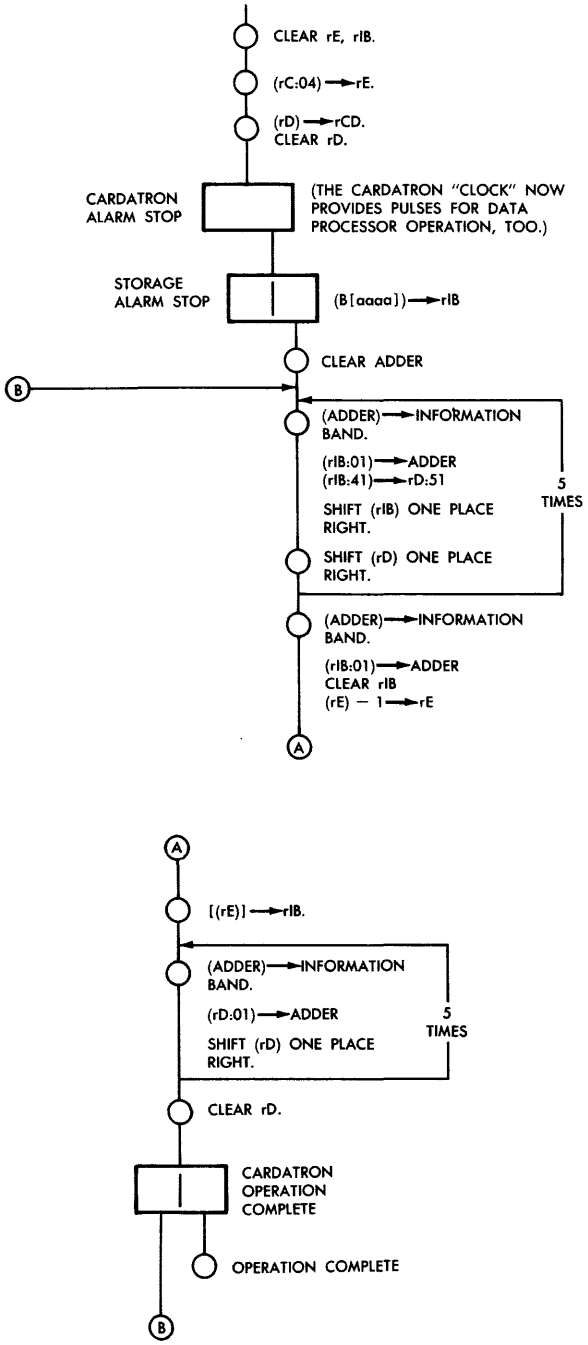
c. The alphabetic character whose code is 12−n, where n is the digit preceding the sign digit.

The digit c is used to transfer the T RELAYS whose points are available on the control panel of the printer (see the *Handbook of Operating Procedures for the BURROUGHS 220*). After the edited contents of the information band have been transferred to the printer, all of the relays will have been returned to their normal (N) position: they are returned to normal just prior to the transfer from the Core Shift Register of the last row of information. The selection of relays specified by c is shown in the following table:

c	Relays Selected
0	None
1	1
2	2
3	3
4	4
5	5
6	2 and 4
7	3 and 5
8	None
9	1

As soon as the transfer of information from core storage to the information band is completed, the Output Unit automatically directs the transfer of the contents of the information band to the card-handling machine.

Flow Chart.



Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address. (See Remark 2.)
2. CARDATRON ALARM STOP due to designating a nonexistent output unit.
3. Digit Check ALARM STOP. (See Remark 3.)

Remarks

1. At the beginning of the Execute Phase of a CWR instruction B[aaaa] = (rC:04) is transferred to the E register. The address of the core storage location from which is read each successive word transferred to the information band is obtained by decreasing by 1 the contents of the E register after each word is read.

   Since the E register counts modulo the size of storage, care must be exercised in the choice of B[aaaa]. For example, suppose the Data Processor has 4000 words of core storage; suppose also the B[aaaa] $\leq$ 0028: hence, some word will be read from location 0000. After this word is read, 1 will be subtracted from 0000, the contents of the E register, modulo 4000. The E register will then contain 3999.

2. Examination of the flow chart will disclose that control is vested in the CARDATRON Unit—by virtue of the fact that the CARDATRON "clock" is providing pulses for Data Processor operation, too—when a nonexistent address is detected. The detection of the nonexistent address inhibits the transfer of any information in the Data Processor. But the CARDATRON Unit must complete its cycle of operation, even through no information is transferred from the Data Processor to the buffer. The ALARM STOP will occur after Part 1 of the CARDATRON cycle is completed. *Part 2 of the CARDATRON cycle will also occur:*

   a. If the Output Unit is connected to a punch, a card will be fed. The card will contain a mixture of blank columns and 0-punches, the configuration being determined by the contents of the format band selected by the CWR instruction.

   b. If the Output Unit is connected to a printer, a blank line will be produced.

3. The detection of an impermissible configuration occurs when (rIB:01) and (rD:01) are shifted to the right. Such an event can occur only after control is vested in the CARDATRON Unit (see the flow chart). The detection of the impermissible configuration immediately stops the transfer of information within the Data Processor, and from the Data Processor to the buffer. But the CARDATRON Unit must complete its cycle of operation, even though no more information is transferred to it. The Digit Check ALARM STOP will occur after Part 1 of the CARDATRON cycle is completed. *Part 2 of the CARDATRON cycle will also occur.* The nature of the output depends both on the editing control stream and on how much information was transferred from the Data Processor to the buffer.

Register Status.

Register name	Contents after execution of CWR
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	u | i | c | f | 6 1 | B[aaaa]
E	B[aaaa] − 28

Register name	Contents if Digit Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	*
B	Unchanged
P	$(rP)_b + 1$
C	u | i | c | f | 6 1 | B[aaaa]
E	*

* Depends on when the impermissible configuration is detected. See flow chart.

Register name	Contents if Storage or CARDATRON ALARM STOP occurs
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	u | i | c | f | 6 1 | B[aaaa]
E	B[aaaa]

CARD READ, FORMAT LOAD (CRF)

Operation Code. 62

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	i	i	f	O	p	a	a	a	a

Definitions.

u: designates Input Unit.

f: specifies which format band will be written. If f is odd, Reload Lockout is imposed.

aaaa: address of base of location from which is read the first word transferred to the format band specified by f.

Description of Operation. Transfer to the format band specified by f—Input Unit u—the contents of 29 core storage locations as follows: the contents of B[aaaa] occupy format-band digit positions 1 through 11; (B[aaaa] − 1) occupy positions 12 through 22: . . . ; (B[aaaa] − 27) occupy positions 298 through 308; (B[aaaa] − 28) occupy positions 309 through 319.

The selection of the format band on which the editing control stream is written and the imposition or release of Reload Lockout are accomplished as described in the table below:

f	Format Band	Reload Lockout
0	1	Released
1	1	Imposed
2	2	Released
3	2	Imposed
4	3	Released
5	3	Imposed
6	4	Released
7	4	Imposed
8	5	Released
9	5	Imposed

Flow Chart. See page 6-28.

Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address (See Remark 3.)
2. CARDATRON ALARM STOP due to designating a nonexistent input unit.
3. Digit Check ALARM STOP. (See Remark 4.)

Remarks.

1. A format-band digit position contains only two bits. Hence, a format-band digit may be only one of 0, 1, 2, or 3. If an attempt is made to write as a format-band digit some decimal digit greater than or equal to 4, the 4-bit or the 8-bit will be dropped

resulting in the transfer of the setting of the 1-bit and/or the 2-bit when that decimal digit is transferred to the selected format band. For example: a 4 or 8 would be transferred as 0, a 5 or 9 as 1, etc.

2. At the beginning of the Execute Phase of a CRF instruction B[aaaa] = (rC:04) is transferred to the E register. The address of the core storage location from which is read each successive word transferred to the format band is obtained by decreasing by 1 the contents of the E register after each word is read.

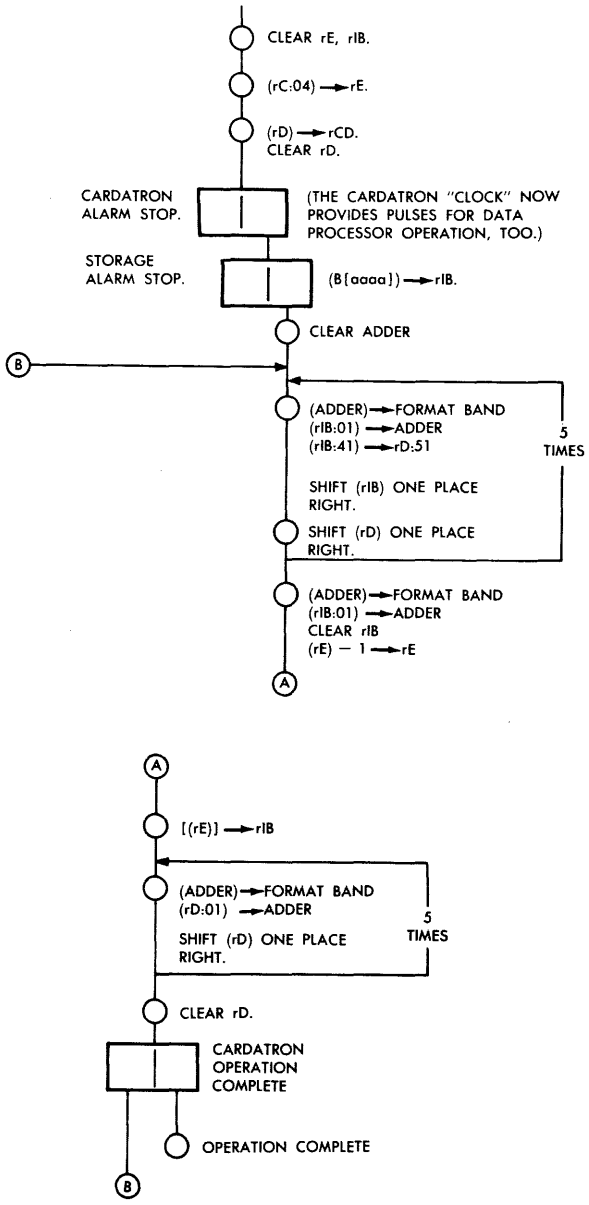
Since the E register counts modulo the size of storage, care must be exercised in the choice of B[aaaa]. For example, suppose the Data Processor has 4000 words of core storage; suppose also that B[aaaa] $\leq$ 0028; then some word will be read from location 0000. After this word is read, 1 will be subtracted from 0000, the contents of the E register, modulo 4000. The E register will then contain 3999.

3. Examination of the flow chart will disclose that control is vested in the CARDATRON Unit—by virtue of the fact that the CARDATRON "clock" is providing pulses for Data Processor operation, too—when a nonexistent address is detected. The detection of the nonexistent address inhibits the transfer of information in the Data Processor. But the CARDATRON Unit must complete its cycle of operation, even though no information is transferred from the Data Processor to the buffer.

The ALARM STOP will occur after the CARDATRON cycle is completed. It will appear that the designated format band had all 0's written on it.

4. The detection of an impermissible configuration occurs when (rIB:01) and (rD:01) are shifted to the right. Such an event can occur only after control is vested in the CARDATRON Unit (see the flow chart). The detection of the impermissible configuration immediately stops the transfer of information within the Data Processor, and from the Data Processor to the buffer. But the CARDATRON Unit must complete its cycle of operation, even though no more information is transferred to it.

The Digit Check ALARM STOP will occur after the CARDATRON cycle is completed. It will appear that the designated format band contains 0's following the last digit transferred from the Data Processor.



Register Status.

Register name	Contents after execution of CRF
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	u | i | i | f | 6 | 2 | B[aaaa]
E	B[aaaa] − 29

Register name	Contents if Digit Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	*
B	Unchanged
P	$(rP)_b + 1$
C	u | i i | f | 6 2 | B[aaaa]
E	*

* Depends on when the impermissible configuration is detected. See flow chart.

Register name	Contents if Storage or CARDATRON ALARM STOP occurs
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	u | i i | f | 6 2 | B[aaaa]
E	B[aaaa]

CARD WRITE, FORMAT LOAD (CWF)

Operation Code. 63

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	i	i	f	O	p	a	a	a	a

Definitions.

u: designates Output Unit.

f: specifies which format band will be written.

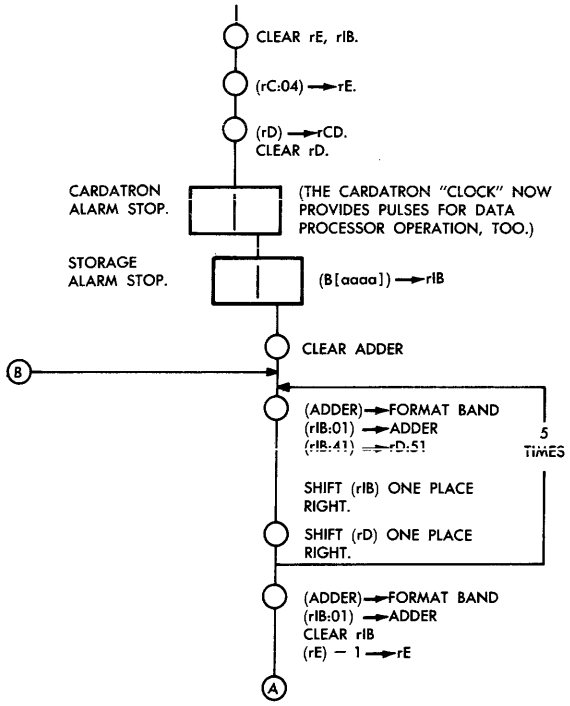
aaaa: address of base of location from which is read the first word transferred to the format band specified by f.

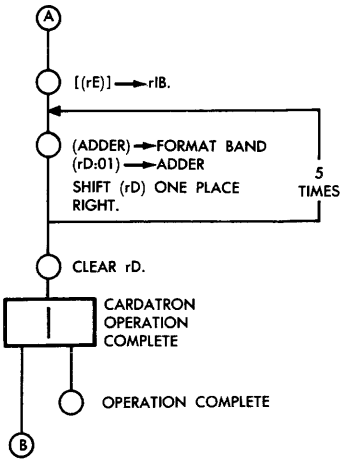
Description of Operation. Transfer to the format band specified by f—on Output Unit u—the contents of 29 core storage locations as follows: the contents of B[aaaa] occupy format-band digit positions 1 through 11; (B[aaaa] − 1) occupy positions 12 through 22; . . . ; (B[aaaa] − 27) occupy positions 298 through 308; (B[aaaa] − 28:08) occupy positions 309 through 316.

The selection of the format band on which the editing control stream is written is accomplished as described in the table below:

f	Format Band
0, 1	1
2, 3	2
4, 5	3
6, 7	4
8, 9	5

Flow Chart.





Exceptional Conditions.

1. Storage ALARM STOP due to use of a nonexistent address. (See Remark 3.)
2. CARDATRON ALARM STOP due to designating a nonexistent output unit.
3. Digit Check ALARM STOP. (See Remark 4.)

Remarks.

1. A format-band digit position contains only two bits. Hence, a format-band digit may be only one of 0, 1, 2, or 3. If an attempt is made to write as a format-band digit some decimal digit greater than or equal to 4, the 4-bit or the 8-bit will be dropped resulting in the transfer of the setting of the 1-bit and/or the 2-bit when the decimal digit is transferred to the selected format band. For example: a 4 or 8 would be transferred as 0, a 5 or 9 as 1, etc.
2. At the beginning of the Execute Phase of a CWF instruction B[aaaa] = (rC:04) is transferred to the E register. The address of the core storage location from which is read each successive word transferred to the format band is obtained by decreasing by 1 the contents of the E register after each word is read from storage.

Since the E register counts modulo the size of storage, care must be exercised in the choice of B[aaaa]. For example, suppose the Data Processor has 4000 words of core storage; suppose also that B[aaaa] $\leq$ 0028: then some word will be read from location 0000. After this word is read, 1 will be subtracted from 0000, the contents of the E register, modulo 4000. The E register will then contain 3999.

3. Examination of the flow chart will disclose that control is vested in the CARDATRON Unit—by virtue of the fact that the CARDATRON "clock" is providing pulses for Data Processor operation, too—when a nonexistent address is detected. The detection of the nonexistent address inhibits the transfer of information in the Data Processor. But the CARDATRON Unit must complete its cycle, even though no information is transferred from the Data Processor to the buffer.

The ALARM STOP will occur after the CARDATRON cycle is completed. It will appear that the designated format band had all 0's written on it.

4. The detection of an impermissible configuration occurs when (rIB:01) and (rD:01) are shifted to the right. Such an event can occur only after control is vested in the CARDATRON Unit (see the flow chart). The detection of the impermissible configuration immediately stops the transfer of information with the Data Processor, and from the Data Processor to the buffer. But the CARDATRON Unit must complete its cycle of operation, even though no more information is transferred to the buffer.

The Digit Check ALARM STOP will occur after the CARDATRON cycle is completed. It will appear that the designated format band contains 0's following the last digit transferred from the Data Processor.

Register Status.

Register name	Contents after execution of CWF
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	u | i i | f | 6 3 | B[aaaa]
E	B[aaaa] − 29

Register name	Contents if Digit Check ALARM STOP occurs
A	Unchanged
R	Unchanged
D	*
B	Unchanged
P	$(rP)_b + 1$
C	u | i i | f | 6 3 | B[aaaa]
E	*

* Depends on when the impermissible configuration is detected. See flow chart.

Register name	Contents if Storage or CARDATRON ALARM STOP occurs
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	u | i i | f | 6 3 | B[aaaa]
E	B[aaaa]

CARD READ INTERROGATE, BRANCH (CRI)

Operation Code. 64

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	i	i	i	O	p	a	a	a	a

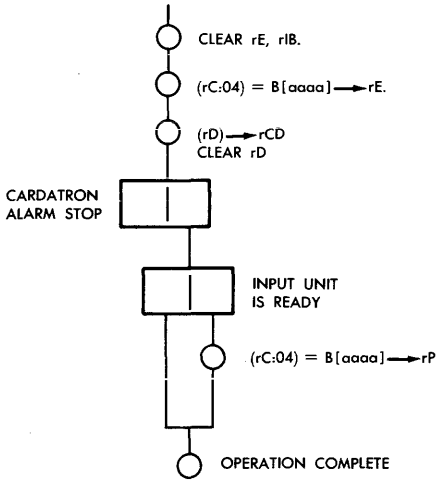
Definitions.

u: designates Input Unit.

aaaa: address of base of location of alternate instruction.

Description of Operation. If Input Unit u is ready, transfer control to B[aaaa], that is, prepare to take the next instruction from B[aaaa]; otherwise, control continues in sequence.

Flow Chart.



Exceptional Conditions. CARDATRON ALARM STOP due to designating a nonexistent input unit.

Register Status.

Register name	Contents after execution of CRI
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	No branch: $(rP)_b + 1$ Branch: B[aaaa]
C	u | i | i | i | 6 | 4 | B[aaaa]
E	B[aaaa]

Register name	Contents if CARDATRON ALARM STOP occurs
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	u | i | i | i | 6 | 4 | B[aaaa]
E	B[aaaa]

CARD WRITE INTERROGATE, BRANCH (CWI)

Operation Code. 65

Instruction Format.

±	1	2	3	4	5	6	7	8	9	0
±	u	i	i	i	O	p	a	a	a	a

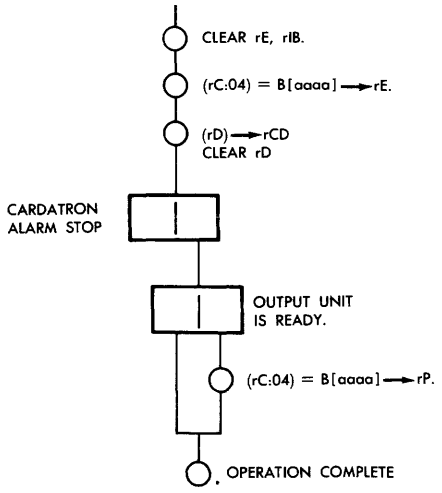
Definitions.

u: designates Output Unit.

aaaa: address of base of location of alternate instruction.

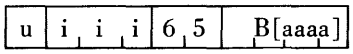
Description of Operation. If Output Unit u is ready, transfer control to B[aaaa], that is, prepare to take the next instruction from B[aaaa]; otherwise, control continues in sequence.

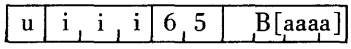
Flow Chart.



Exceptional Conditions. CARDATRON ALARM STOP due to designating a nonexistent output unit.

Register Status.

Register name	Contents if CARDATRON ALARM STOP occurs
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	$(rP)_b + 1$
C	
E	B[aaaa]

Register name	Contents after execution of CWI
A	Unchanged
R	Unchanged
D	Cleared
B	Unchanged
P	No branch: $(rP)_b + 1$ Branch: B[aaaa]
C	
E	B[aaaa]

Appendices

A. BURROUGHS 220 INSTRUCTION LIST IN OPERATION-CODE ORDER

	Instruction Format			Operation		
±	1234	56	7890	Abr	Name	Page
±	iiii	00	iiii	HLT	HALT	2-6
±	iiii	01	iiii	NOP	NO OPERATION	2-7
±	unnv	03	aaaa	PRD	PAPER-TAPE READ	5-4
±	uiiv	04	aaaa	PRB	PAPER-TAPE READ, BRANCH	5-6
±	unnv	05	aaaa	PRI	PAPER-TAPE READ, INVERSE FORMAT	5-8
±	unni	06	aaaa	PWR	PAPER-TAPE WRITE	5-9
±	iiii	08	iiii	KAD	KEYBOARD ADD	3-2
±	dnnv	09	aaaa	SPO	SUPERVISORY PRINT-OUT	3-2
±	iii0	10	aaaa	CAD	CLEAR ADD	2-7
±	iiil	10	aaaa	CAA	CLEAR ADD ABSOLUTE	2-7
±	iii0	11	aaaa	CSU	CLEAR SUBTRACT	2-8
±	iiil	11	aaaa	CSA	CLEAR SUBTRACT ABSOLUTE	2-8
±	iii0	12	aaaa	ADD	ADD	2-9
±	iiil	12	aaaa	ADA	ADD ABSOLUTE	2-9
±	iii0	13	aaaa	SUB	SUBTRACT	2-10
±	iiil	13	aaaa	SUA	SUBTRACT ABSOLUTE	2-10
±	iiii	14	aaaa	MUL	MULTIPLY	2-11
±	iiii	15	aaaa	DIV	DIVIDE	2-12
±	iiii	16	iiii	RND	ROUND	2-13
±	iiii	17	aaaa	EXT	EXTRACT	2-14
±	sLf0	18	aaaa	CFA	COMPARE FIELD A	2-14
±	sLf1	18	aaaa	CFR	COMPARE FIELD R	2-14
±	iiii	19	aaaa	ADL	ADD TO LOCATION	2-16
±	nnnn	20	aaaa	IBB	INCREASE B, BRANCH	2-17
±	nnnn	21	aaaa	DBB	DECREASE B, BRANCH	2-18
±	nii0	22	aaaa	FAD	FLOATING ADD	2-18
±	niil	22	aaaa	FAA	FLOATING ADD ABSOLUTE	2-18
±	nii0	23	aaaa	FSU	FLOATING SUBTRACT	2-20
±	niil	23	aaaa	FSA	FLOATING SUBTRACT ABSOLUTE	2-20
±	iiii	24	aaaa	FMU	FLOATING MULTIPLY	2-22
±	iiii	25	aaaa	FDV	FLOATING DIVIDE	2-24
±	sLnn	26	aaaa	IFL	INCREASE FIELD LOCATION	2-26
±	sLnn	27	aaaa	DFL	DECREASE FIELD LOCATION	2-27

Instruction Format				Operation		
±	1234	56	7890	Abr	Name	Page
±	sLnn	28	aaaa	DLB	DECREASE FIELD LOCATION, LOAD B	2-29
±	inni	29	aaaa	RTF	RECORD TRANSFER	2-30
±	iiii	30	aaaa	BUN	BRANCH, UNCONDITIONALLY	2-31
±	iiii	31	aaaa	BOF	BRANCH, OVERFLOW	2-31
±	iiii	32	aaaa	BRP	BRANCH, REPEAT	2-32
±	iiin	33	aaaa	BSA	BRANCH, SIGN A	2-32
±	iii0	34	aaaa	BCH	BRANCH, COMPARISON HIGH	2-33
±	iii1	34	aaaa	BCL	BRANCH, COMPARISON LOW	2-33
±	iii0	35	aaaa	BCE	BRANCH, COMPARISON EQUAL	2-34
±	iii1	35	aaaa	BCU	BRANCH, COMPARISON UNEQUAL	2-34
±	sLnn	36	aaaa	BFA	BRANCH, FIELD A	2-35
±	sLnn	37	aaaa	BFR	BRANCH, FIELD R	2-36
±	uiii	38	aaaa	BCS	BRANCH, CONTROL SWITCH	3-3
±	iii0	39	aaaa	SOR	SET OVERFLOW REMEMBER	2-37
±	iii1	39	aaaa	SOH	SET OVERFLOW HALT	2-37
±	iii2	39	aaaa	IOM	INTERROGATE OVERFLOW MODE	2-37
±	sLf0	40	aaaa	STA	STORE A	2-38
±	sLf1	40	aaaa	STR	STORE R	2-38
±	sLf2	40	aaaa	STB	STORE B	2-38
±	iiii	41	aaaa	LDR	LOAD R	2-39
±	iii0	42	aaaa	LDB	LOAD B	2-40
±	iii1	42	aaaa	LBC	LOAD B COMPLEMENT	2-40
±	iiin	43	iiii	LSA	LOAD SIGN A	2-40
±	iiii	44	aaaa	STP	STORE P	2-41
±	iii1	45	iiii	CLA	CLEAR A	2-41
±	iii2	45	iiii	CLR	CLEAR R	2-41
±	iii3	45	iiii	CAR	CLEAR A, R	2-41
±	iii4	45	iiii	CLB	CLEAR B	2-41
±	iii5	45	iiii	CAB	CLEAR A, B	2-41
±	iii6	45	iiii	CRB	CLEAR R, B	2-41
±	iii7	45	iiii	CLT	CLEAR A, R, B	2-41
±	iiii	46	aaaa	CLL	CLEAR LOCATION	2-42
±	iii0	48	iinn	SRA	SHIFT RIGHT A	2-43
±	iii1	48	iinn	SRT	SHIFT RIGHT A AND R	2-43
±	iii2	48	iinn	SRS	SHIFT RIGHT A WITH SIGN	2-43
±	iii0	49	iinn	SLA	SHIFT LEFT A	2-44
±	iii1	49	iinn	SLT	SHIFT LEFT A AND R	2-44
±	iii2	49	iinn	SLS	SHIFT LEFT A WITH SIGN	2-44
0	uhh0	50	aaaa	MTS	MAGNETIC-TAPE SEARCH	4-12
4	uhh0	50	aaaa	MFS	MAGNETIC-TAPE FIELD SEARCH	4-12
±	uhh4	50	iiii	MLS	MAGNETIC-TAPE LANE SELECT	4-12

Instruction Format				Operation		
±	1234	56	7890	Abr	Name	Page
±	uhh8	50	iiii	MRW	MAGNETIC-TAPE REWIND	4-12
±	uhh9	50	iiii	MDA	MAGNETIC-TAPE REWIND, DE-ACTIVATE	4-12
0	uhhk	51	aaaa	MTC	MAGNETIC-TAPE SCAN	4-14
4	uhhk	51	aaaa	MFC	MAGNETIC-TAPE FIELD SCAN	4-14
±	univ	52	aaaa	MRD	MAGNETIC-TAPE READ	4-15
±	univ	53	aaaa	MRR	MAGNETIC-TAPE READ, RECORD	4-16
±	unkk	54	aaaa	MIW	MAGNETIC-TAPE INITIAL WRITE	4-18
±	unii	55	aaaa	MIR	MAGNETIC-TAPE INITIAL WRITE, RECORD	4-19
±	unkk	56	aaaa	MOW	MAGNETIC-TAPE OVERWRITE	4-20
±	unii	57	aaaa	MOR	MAGNETIC-TAPE OVERWRITE, RECORD	4-21
±	uni0	58	iiii	MPF	MAGNETIC-TAPE POSITION FORWARD	4-21
±	unil	58	iiii	MPB	MAGNETIC-TAPE POSITION BACKWARD	4-21
±	uni2	58	iiii	MPE	MAGNETIC-TAPE POSITION AT END OF INFORMATION	4-21
±	uii0	59	aaaa	MIB	MAGNETIC-TAPE INTERROGATE, BRANCH	4-22
±	uiil	59	aaaa	MIE	MAGNETIC-TAPE INTERROGATE END OF TAPE, BRANCH	4-22
±	uiir	60	aaaa	CRD	CARD READ	6-18
±	uicf	61	aaaa	CWR	CARD WRITE	6-25
±	uiif	62	aaaa	CRF	CARD READ, FORMAT LOAD	6-27
±	uiif	63	aaaa	CWF	CARD WRITE, FORMAT LOAD	6-29
±	uiii	64	aaaa	CRI	CARD READ INTERROGATE, BRANCH	6-31
±	uiii	65	aaaa	CWI	CARD WRITE INTERROGATE, BRANCH	6-31

B. BURROUGHS 220 INSTRUCTION LIST, ALPHABETICALLY, BY ABBREVIATION

Instruction Format				Operation		
±	1234	56	7890	Abr	Name	Page
±	iiil	12	aaaa	ADA	ADD ABSOLUTE	2-9
±	iii0	12	aaaa	ADD	ADD	2-9
±	iiii	19	aaaa	ADL	ADD TO LOCATION	2-16
±	iii0	35	aaaa	BCE	BRANCH, COMPARISON EQUAL	2-34
±	iii0	34	aaaa	BCH	BRANCH, COMPARISON HIGH	2-33
±	iiil	34	aaaa	BCL	BRANCH, COMPARISON LOW	2-33
±	uiii	38	aaaa	BCS	BRANCH, CONTROL SWITCH	3-3
±	iiil	35	aaaa	BCU	BRANCH, COMPARISON UNEQUAL	2-34
±	sLnn	36	aaaa	BFA	BRANCH, FIELD A	2-35
±	sLnn	37	aaaa	BFR	BRANCH, FIELD R	2-36
±	iiii	31	aaaa	BOF	BRANCH, OVERFLOW	2-31
±	iiii	32	aaaa	BRP	BRANCH, REPEAT	2-32

Instruction Format			Operation		
±	1234 56 7890		Abr	Name	Page
±	iiin	33 aaaa	BSA	BRANCH, SIGN A	2-32
±	iiii	30 aaaa	BUN	BRANCH, UNCONDITIONALLY	2-31
±	iiil	10 aaaa	CAA	CLEAR ADD ABSOLUTE	2-7
±	iii5	45 iiii	CAB	CLEAR A, B	2-41
±	iii0	10 aaaa	CAD	CLEAR ADD	2-7
±	iii3	45 iiii	CAR	CLEAR A, R	2-41
±	sLf0	18 aaaa	CFA	COMPARE FIELD A	2-14
±	sLf1	18 aaaa	CFR	COMPARE FIELD R	2-14
±	iiil	45 iiii	CLA	CLEAR A	2-41
±	iii4	45 iiii	CLB	CLEAR B	2-41
±	iiii	46 aaaa	CLL	CLEAR LOCATION	2-42
±	iii2	45 iiii	CLR	CLEAR R	2-41
±	iii7	45 iiii	CLT	CLEAR A, R, B	2-41
±	iii6	45 iiii	CRB	CLEAR R, B	2-41
±	uiir	60 aaaa	CRD	CARD READ	6-18
±	uiif	62 aaaa	CRF	CARD READ, FORMAT LOAD	6-27
±	uiii	64 aaaa	CRI	CARD READ INTERROGATE, BRANCH	6-31
±	iiil	11 aaaa	CSA	CLEAR SUBTRACT ABSOLUTE	2-8
±	iii0	11 aaaa	CSU	CLEAR SUBTRACT	2-8
±	uiif	63 aaaa	CWF	CARD WRITE, FORMAT LOAD	6-29
±	uiii	65 aaaa	CWI	CARD WRITE INTERROGATE, BRANCH	6-31
±	uicf	61 aaaa	CWR	CARD WRITE	6-25
±	nnnn	21 aaaa	DBB	DECREASE B, BRANCH	2-18
±	sLnn	27 aaaa	DFL	DECREASE FIELD LOCATION	2-27
±	iiii	15 aaaa	DIV	DIVIDE	2-12
±	sLnn	28 aaaa	DLB	DECREASE FIELD LOCATION, LOAD B	2-29
±	iiii	17 aaaa	EXT	EXTRACT	2-14
±	niil	22 aaaa	FAA	FLOATING ADD ABSOLUTE	2-18
±	nii0	22 aaaa	FAD	FLOATING ADD	2-18
±	iiii	25 aaaa	FDV	FLOATING DIVIDE	2-24
±	iiii	24 aaaa	FMU	FLOATING MULTIPLY	2-22
±	niil	23 aaaa	FSA	FLOATING SUBTRACT ABSOLUTE	2-20
±	nii0	23 aaaa	FSU	FLOATING SUBTRACT	2-20
±	iiii	00 iiii	HLT	HALT	2-6
±	nnnn	20 aaaa	IBB	INCREASE B, BRANCH	2-17
±	sLnn	26 aaaa	IFL	INCREASE FIELD LOCATION	2-26
±	iii2	39 aaaa	IOM	INTERROGATE OVERFLOW MODE	2-37
±	iii2	08 iiii	KAD	KEYBOARD ADD	3-2
±	iiil	42 aaaa	LBC	LOAD B COMPLEMENT	2-40
±	iii0	42 aaaa	LDB	LOAD B	2-40
±	iiii	41 aaaa	LDR	LOAD R	2-39

Instruction Format				Operation		
±	1234	56	7890	Abr	Name	Page
±	iiin	43	iiii	LSA	LOAD SIGN A	2-40
±	uhh9	50	iiii	MDA	MAGNETIC-TAPE REWIND, DE-ACTIVATE	4-12
4	uhhk	51	aaaa	MFC	MAGNETIC-TAPE FIELD SCAN	4-14
4	uhh0	50	aaaa	MFS	MAGNETIC-TAPE FIELD SEARCH	4-12
±	uii0	59	aaaa	MIB	MAGNETIC-TAPE INTERROGATE, BRANCH	4-22
±	uiil	59	aaaa	MIE	MAGNETIC-TAPE INTERROGATE END OF TAPE, BRANCH	4-22
±	unii	55	aaaa	MIR	MAGNETIC-TAPE INITIAL WRITE, RECORD	4-19
±	unkk	54	aaaa	MIW	MAGNETIC-TAPE INITIAL WRITE	4-18
±	uhh4	50	iiii	MLS	MAGNETIC-TAPE LANE SELECT	4-12
±	unii	57	aaaa	MOR	MAGNETIC-TAPE OVERWRITE, RECORD	4-21
±	unkk	56	aaaa	MOW	MAGNETIC-TAPE OVERWRITE	4-20
±	unil	58	iiii	MPB	MAGNETIC-TAPE POSITION BACKWARD	4-21
±	uii2	58	iiii	MPE	MAGNETIC-TAPE POSITION AT END OF INFORMATION	4-21
±	uni0	58	iiii	MPF	MAGNETIC-TAPE POSITION FORWARD	4-21
±	univ	52	aaaa	MRD	MAGNETIC-TAPE READ	4-15
±	univ	53	aaaa	MRR	MAGNETIC-TAPE READ, RECORD	4-16
±	uhh8	50	iiii	MRW	MAGNETIC-TAPE REWIND	4-12
0	uhhk	51	aaaa	MTC	MAGNETIC-TAPE SCAN	4-14
0	uhh0	50	aaaa	MTS	MAGNETIC-TAPE SEARCH	4-12
±	iiii	14	aaaa	MUL	MULTIPLY	2-11
±	iiii	01	iiii	NOP	NO OPERATION	2-7
±	uiiv	04	aaaa	PRB	PAPER-TAPE READ, BRANCH	5-6
±	unnv	03	aaaa	PRD	PAPER-TAPE READ	5-4
±	unnv	05	aaaa	PRI	PAPER-TAPE READ, INVERSE FORMAT	5-8
±	unni	06	aaaa	PWR	PAPER-TAPE WRITE	5-9
±	iiii	16	iiii	RND	ROUND	2-13
±	inni	29	aaaa	RTF	RECORD TRANSFER	2-30
±	iii0	49	iinn	SLA	SHIFT LEFT A	2-44
±	iii2	49	iinn	SLS	SHIFT LEFT A WITH SIGN	2-44
±	iiil	49	iinn	SLT	SHIFT LEFT A AND R	2-44
±	iiil	39	aaaa	SOH	SET OVERFLOW HALT	2-37
±	iii0	39	aaaa	SOR	SET OVERFLOW REMEMBER	2-37
±	dmnv	09	aaaa	SPO	SUPERVISORY PRINT-OUT	3-2
±	iii0	48	iinn	SRA	SHIFT RIGHT A	2-43
±	iii2	48	iinn	SRS	SHIFT RIGHT A WITH SIGN	2-43
±	iiil	48	iinn	SRT	SHIFT RIGHT A AND R	2-43
±	sLf0	40	aaaa	STA	STORE A	2-38
±	sLf2	40	aaaa	STB	STORE B	2-38
±	iiii	44	aaaa	STP	STORE P	2-41

Instruction Format ± 1234 56 7890	Abr	Operation Name	Page
± sLf1 40 aaaa	STR	STORE R	2-38
± iii1 13 aaaa	SUA	SUBTRACT ABSOLUTE	2-10
± iii0 13 aaaa	SUB	SUBTRACT	2-10

C. ALPHANUMERIC CODES AND THEIR REPRESENTATION

Symbol		Computer and Magnetic Tape	Code: Paper Tape: Channel							CARDATRON Buffer	Punched Card
BURROUGHS 220	046		X	0	√ (Parity Check)	8	4	2	1		
(space)	(space)	00			√					0 - 0	(blank)
•	•	03	X	0		8		2	1	1 - 11	12 8-3
□	□	04	X	0	√	8	4			1 - 12	12 8-4
&	&	10	X	0	√					1 - 0	12
$	$	13	X		√	8		2	1	2 - 11	11 8-3
*	*	14	X			8	4			2 - 12	11 8-4
-	-	20	X							2 - 0	11
/	/	21		0	√				1	4 - 1	0 1
,	,	23		0	√	8		2	1	4 - 11	0 8-3
%	%	24		0		8	4			4 - 12	0 8-4
#	#	33				8		2	1	0 - 11	8-3
@	@	34			√	8	4			0 - 12	8-4
A	A	41	X	0					1	1 - 1	12 1
B	B	42	X	0				2		1 - 2	12 2
C	C	43	X	0	√			2	1	1 - 3	12 3
D	D	44	X	0			4			1 - 4	12 4

Symbol		Computer and Magnetic Tape	Code							CARDATRON Buffer	Punched Card
			Paper Tape Channel								
BURROUGHS 220	046		X	0	Parity Check √	8	4	2	1		
E	E	45	X	0	√		4		1	1 - 5	12 5
F	F	46	X	0	√		4	2		1 - 6	12 6
G	G	47	X	0			4	2	1	1 - 7	12 7
H	H	48	X	0		8				1 - 8	12 8
I	I	49	X	0	√	8			1	1 - 9	12 9
J	J	51	X		√				1	2 - 1	11 1
K	K	52	X		√			2		2 - 2	11 2
L	L	53	X					2	1	2 - 3	11 3
M	M	54	X		√		4			2 - 4	11 4
N	N	55	X				4		1	2 - 5	11 5
O	O	56	X				4	2		2 - 6	11 6
P	P	57	X		√		4	2	1	2 - 7	11 7
Q	Q	58	X		√	8				2 - 8	11 8
R	R	59	X			8			1	2 - 9	11 9
S	S	62		0	√			2		4 - 2	0 2
T	T	63		0				2	1	4 - 3	0 3
U	U	64		0	√		4			4 - 4	0 4
V	V	65		0			4		1	4 - 5	0 5
W	W	66		0			4	2		4 - 6	0 6
X	X	67		0	√		4	2	1	4 - 7	0 7
Y	Y	68		0	√	8				4 - 8	0 8
Z	Z	69		0		8			1	4 - 9	0 9

Symbol		Computer and Magnetic Tape	Code							CARDATRON Buffer	Punched Card
			Paper Tape Channel								
BURROUGHS 220	046		X	0	Parity Check √	8	4	2	1		
0 (zero)	0 (zero)	80		0						4 - 0	0
1	1	81							1	0 - 1	1
2	2	82						2		0 - 2	2
3	3	83			√			2	1	0 - 3	3
4	4	84					4			0 - 4	4
5	5	85			√		4		1	0 - 5	5
6	6	86			√		4	2		0 - 6	6
7	7	87					4	2	1	0 - 7	7
8	8	88				8				0 - 8	8
9	9	89			√	8			1	0 - 9	9
Tab	Skip	26		0	√	8	4	2		0 - 14	0 8-6
Carriage Return	CR	16	X		√	8	4	2		2 - 14	11 8-6
Form Out	PI 6	15	X		√	8	4		1	Not assigned	11 8-5
End-of-Word	PI 7	35				8	4		1		8-5
Blank	SP 1	02	X	0	√	8		2			12 8-2
Tape Feed	Tape Feed	*	X	0	√	8	4	2	1		12 8-7
Not assigned	End Card 1	36				8	4	2			8-6
	End Card 2	27		0		8	4	2	1		0 8-7
	Correct Tab	37			√	8	4	2	1		8-7

* Not represented.

Symbol		Computer and Magnetic Tape	Code: Paper Tape: Channel							CARDATRON Buffer	Punched Card
					Parity Check						
BURROUGHS 220	046		X	0	√	8	4	2	1		
Not assigned	Error	17	X			8	4	2	1	Not assigned	11 8-7
	PI 1	32			√	8		2			8-2
	PI 2	12	X			8		2			11 8-2
	PI 3	22		0		8		2			0 8-2
	PI 4	25		0	√	8	4		1		0 8-5
	PI 5	05	X	0		8	4		1		12 8-5
	SP 2	06	X	0		8	4	2			12 8-6

D. A GLOSSARY OF TERMS

ABSOLUTE VALUE: The absolute value of a number, n, is the number which results from making the sign of n positive. For example, if n = +6, the absolute value of n is +6; if n = −7, the absolute value of n is +7. In symbols, we would write: |+6| = +6; |−7| = +7.

In general, then, |n| = +n, if n is greater than or equal to zero; and |n| = −n, if n is less than zero.

ADDRESS OF BASE OF LOCATION • • • : Because the address part of an instruction word can be augmented by the contents of the B register as the instruction is fetched from storage to the C register, that address frequently is not the address of the location referenced by the instruction which is executed. In such cases the address part of the instruction in storage is the base on which is constructed—by addition of the contents of the B register—the actual address used by the instruction which is executed. In this context it is convenient to call the address part of the stored instruction the "address of the base of the location of • • •."

BLOCK: A "block" of information on BURROUGHS 220 magnetic tape is the information recorded between two inter-block gaps. It is so stated because a block may contain any number of Data Processor words from a minimum of 10 words to a maximum of 100 words.

B-REGISTER ADDRESS-MODIFICATION: The B register and the address part of a word are four digits long. B-register address-modification is the addition of the contents of the B register to the address part of a word; only the four low-order digits of the sum are retained however. Moreover, there can be no carry into digit-position 5 in the word which is modified.

FIELD: A field is a set of contiguous digit-positions.

ITEM: An item is a piece of information within a record.

PARTIAL-WORD FIELD: A field, all of whose digit positions lie in the same word.

RECORD: A record is the amount of information on any one subject; this is a function of the problem, not of the machine.

E. A GLOSSARY OF SYMBOLS

(aaaa): Contents of location whose address is aaaa.

$(aaaa)_a$: The subscript "a" identifies the contents after execution.

$(aaaa)_b$: The subscript "b" identifies the contents before execution.

[aaaa]: The enclosure, in brackets, of an address, is an indication that the address may be or will be modified during program execution.

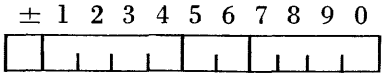
<aaaa>: It is convenient to distinguish B-register address-modification from other types. Although the contents of the sign-digit position of an instruction word specify such modification, the sign-digit is not sufficiently distinctive on a handwritten coding sheet.

B[aaaa]: In the description of each operation it is necessary to indicate that B-register address-modification may or may not occur. This symbol indicates, therefore, that where B-modification is not intended it represents the contents of location aaaa; where B-modification is intended it represents the contents of location (aaaa + the contents of the B-register).

rA, etc.: A lower-case "r" will precede the names of the various registers in the control and arithmetic units in order to distinguish them. rA, for example, is the A register.

rA:sL: In order to specify a partial-word field, three parameters are required: the location of the word containing the field, and the two boundaries of the field. In the example shown, the location is the A register. The letter s designates the position of the right-hand boundary, i.e., the low-order digit of the partial-word field. The left-hand boundary is defined by specifying that there are L digits in the partial-word field. For example, rA:04 specifies the address part of the A register; (rA:04) is the address itself, i.e., the four digits in the specified field. And rD:±1 is the sign-digit position in the D register.

(rA:±1)/1: This symbol specifies the one bit of the A register's sign digit. Similarly, (rA:±1)/2 means the two bit of the A register's sign digit, and so forth.



This diagram identifies the digit positions in a computer word. The digit position identified by the symbol "±" is called the sign-digit position. The contents of digit positions 1, 2, 3, and 4 are sometimes called control digits; the contents of digit positions 5 and 6 comprise the operation code; and the remaining four digit-positions comprise the address part of the word, when these terms are meaningful.

→: "(rP) → rE" means "the contents of the P register go into (i.e., replace the contents of) the E register." Alternatively one may write "(rP) → (rE)" which is to be read "the contents of the P register replace the contents of the E register".

⇒: "(rB) Adder → (rC:04)" means "the contents of the B register go through the adder and replace the contents of the address part of the C register."

|nnnn|: the absolute value of nnnn. See Appendix D, A Glossary of Terms, for a definition of absolute value.

The following symbols are used in the flow charts, which elaborate on the execution of the BURROUGHS 220 operations:

2-60: This is connector A. If it is an output connector, the corresponding input connector will be found on page 2-60; if it is an input connector, the corresponding output connector will be found on page 2-60.

 : This indicates action. All of the action associated with this symbol is specified beside the symbol. All of the action associated with the symbol occurs simultaneously.

 : This is the symbol for a branch point; it is used to indicate alternative courses of action.

F. INSTRUCTION TIMES, ALPHABETICALLY, BY ABBREVIATION

Abr	Op Code	Total time (Fetch + Execute) in μs / Remarks
ADA	12	185 / No decomplement
		245 / Decomplement
ADD	12	185 / No decomplement
		245 / Decomplement
ADL	19	255 / No decomplement
		315 / Decomplement
BCE	35	105 / No branch
		125 / Branch
BCH	34	105 / No branch
		125 / Branch
BCL	34	105 / No branch
		125 / Branch
BCS	38	105 / No branch
		125 / Branch

Abr	Op Code	Total time (Fetch + Execute) in μs / Remarks
BCU	35	105 / No branch
		125 / Branch
BFA	36	165 / No branch
		185 / Branch
BFR	37	165 / No branch
		185 / Branch
BOF	31	105 / No branch
		125 / Branch
BRP	32	105 / No branch
		125 / Branch
BSA	33	175 / No branch
		195 / Branch
BUN	30	125
CAA	10	185
CAB	45	100
CAD	10	185
CAR	45	100
CFA	18	240
CFR	18	240
CLA	45	100
CLB	45	100
CLL	46	115
CLR	45	100
CLT	45	100
CRB	45	100
CRD	60	5855 / Minimum
		7270 / Average
		9110 / Maximum
CRF	62	8635 / Minimum
		10050 / Average
		12030 / Maximum
CRI	64	335 / Minimum, no branch
		355 / Average, no branch
		375 / Maximum, no branch
		355 / Minimum, branch
		375 / Average, branch
		395 / Maximum, branch
CSA	11	185
CSU	11	185
CWF	63	8635 / Minimum
		10050 / Average
		12030 / Maximum
CWI	65	335 / Minimum, no branch
		355 / Average, no branch
		375 / Maximum, no branch
		355 / Minimum, branch
		375 / Average, branch
		395 / Maximum, branch
CWR	61	8635 / Minimum
		10050 / Average
		12030 / Maximum
DBB	21	130 / No branch
		150 / Branch
DFL	27	250
DIV	15	/ Quotient is $(rA)_a$
		180 / Minimum, overflow
		1285 / Minimum, $(rA)_a = \pm$ 0909 09 0909
		3985 / Average, $(rA)_a = \pm$ 5555 55 5555
		6685 / Maximum, $(rA)_a = \pm$ 9090 90 9090
		/ Execution time—exclusive of fetch time—is a function of the magnitude of the quotient $(rA)_a$, if overflow does not occur. It may be calculated with the aid of the following formula:
		$T = 3895 + 60\sum_{k=0}^{4} [(rA: 2k+1, 1)_a - (rA:2k, 1)_a]$
		$= 3895 + 60 [(rA:11) - (rA:21) + (rA:31) - (rA:41) + (rA:51) - (rA:61) + (rA:71) - (rA:81) + (rA:91) - (rA:01)]$
DLB	28	250
EXT	17	235
FAA	22	280 / Sum = 0
		325 / Minimum, underflow
		410 / Maximum, underflow
		215 / Sum $\neq$ 0, no decomplement, minimum
		360 / Sum $\neq$ 0, no decomplement, maximum
		280 / Sum $\neq$ 0, decomplement, minimum
		415 / Sum $\neq$ 0, decomplement, maximum

Abr	Op Code		Total time (Fetch + Execute) in μs / Remarks
FAD	22	280	/ Sum = 0
		325	/ Minimum, underflow
		410	/ Maximum, underflow
		215	/ Sum $\neq$ 0, no decomplement, minimum
		360	/ Sum $\neq$ 0, no decomplement, maximum
		280	/ Sum $\neq$ 0, decomplement, minimum
		415	/ Sum $\neq$ 0, decomplement, maximum
FDV	25	175	/ Quotient or divisor = 0
		260	/ Underflow
		175	/ Overflow, minimum
		6775	/ Overflow, maximum
		4075	/ Quotient $\neq$ 0, average
		6775	/ Quotient $\neq$ 0, maximum
FMU	24	175	/ Product = 0
		255	/ Overflow
		255	/ Underflow, minimum
		2915	/ Underflow, maximum
		1735	/ Product $\neq$ 0, average
		2915	/ Product $\neq$ 0, maximum
FSA	23	280	/ Difference = 0
		325	/ Underflow, minimum
		410	/ Underflow, maximum
		215	/ Difference $\neq$ 0, no decomplement, minimum
		360	/ Difference $\neq$ 0, no decomplement, maximum
		280	/ Difference $\neq$ 0, decomplement, minimum
		415	/ Difference $\neq$ 0, decomplement, maximum
FSU	23	280	/ Difference = 0
		325	/ Underflow, minimum
		410	/ Underflow, maximum
		215	/ Difference $\neq$ 0, no decomplement, minimum
		360	/ Difference $\neq$ 0, no decomplement, maximum
		280	/ Difference $\neq$ 0, decomplement, minimum
		415	/ Difference $\neq$ 0, decomplement, maximum
HLT	00	100	
IBB	20	130	/ No branch
		150	/ Branch
IFL	26	250	
IOM	39	105	/ No branch
		125	/ Branch
KAD	08		/ Manual operation
LBC	42	180	
LDB	42	180	
LDR	41	175	
LSA	43	105	
MDA	50		/ See Appendix G.
MFC	51		/ See Appendix G.
MFS	50		/ See Appendix G.
MIB	59		/ See Appendix G.
MIE	59		/ See Appendix G.
MIR	55		/ See Appendix G.
MIW	54		/ See Appendix G.
MLS	50		/ See Appendix G.
MOR	57		/ See Appendix G.
MOW	56		/ See Appendix G.
MPB	58		/ See Appendix G.
MPE	58		/ See Appendix G.
MPF	58		/ See Appendix G.
MRD	52		/ See Appendix G.
MRR	53		/ See Appendix G.
MRW	50		/ See Appendix G.
MTC	51		/ See Appendix G.
MTS	50		/ See Appendix G.
MUL	14		/ Multiplier is $(rA)_b$
		230	/ Minimum, $(rA)_b$ = $\pm$ 0000 00 0000
		2095	/ Average, $(rA)_b$ = $\pm$ 0123 45 6789
		3480	/ Maximum, $(rA)_b$ = $\pm$ 5555 55 5555
			/ Execution—exclusive of fetch time—is a function of the magnitude of the multiplier $(rA)_b$. It may be calculated from the following formula:
			$T = 90 + 5\sum_{k=0}^{9} M_k$,

Abr	Op Code	Total time (Fetch + Execute) in μs / Remarks
	where	
	$M_k = 1$	if (rA:kl) = 0
	$M_k = 13\,[(rA:kl)] + 1$	if $1 \leq$ (rA:kl) ≤ 5
	$M_k = 13\,[11-(rA:kl)]$	if $6 \leq$ (rA:kl) ≤ 9

Hence the following table:

(rA:kl)	0	1	2	3	4	5	6	7	8	9
M_k	1	14	27	40	54	66	65	52	39	26

Abr	Op Code	Total time (Fetch + Execute) in μs / Remarks
NOP	01	100
PRB	04	/ Photoreader speed, nominally 1000 characters per second
PRD	03	/ Photoreader speed, nominally 1000 characters per second
PRI	05	/ Photoreader speed, nominally 1000 characters per second
PWR	06	/ Punch speed, nominally 60 characters per second / Printer speed, nominally 7-- 10 characters per second
RND	16	105 / (rR:11) < 5 165 / (rR:11) $\geq$ 5
RTF	29	$130 + 60(nn)$ / $01 \leq nn \leq 99$ 6130 / nn = 00
SLA	49	$160 - 5N_1$ / nn $\equiv N_1$, mod 10
SLS	49	$160 - 5N_1$ / nn $\equiv N_1$, mod 10
SLT	49	$210 - 5N_2$ / nn $\equiv N_2$, mod 20
SOH	39	105
SOR	39	105
SPO	09	/ Printer speed, nominally 7 - 10 characters per second
SRA	48	$110 + 5N$ / nn $\equiv$ N, mod 20
SRS	48	$110 + 5N$ / nn $\equiv$ N, mod 20
SRT	48	$110 + 5N$ / nn $\equiv$ N, mod 20
STA	40	190
STB	40	190
STP	44	185
STR	40	190
SUA	13	185 / No decomplement 245 / Decomplement
SUB	13	185 / No decomplement 245 / Decomplement

G. MAGNETIC-TAPE INSTRUCTION TIMES

Magnetic-tape instruction times include Data Processor instruction Fetch time, and complete Execute time including that during which the Data Processor and other units associated in the execution of the instruction are busy together or separately. Table G-1 lists instruction times when the TSU is used; Table G-2 lists times for use of the DATAFILE Unit.

TSU INSTRUCTION TIMES (See Table G-1)

Times are computed on the following basis:

1. Data Processor Fetch time: 09 ms (column 2)
   Data Processor Execute time (column 3)
2. Tape transit time at full speed (.04 ms/digit space) (column 3 or 4)
3. Start time: 3 ms.

Start time = 3 ms + unknown time difference "x" which results from difference in position of head relative to preface of first block to be reached. The formulas in Table G-1 and G-2 are computed with "x" = 1 ms.

In case of instructions involving a lane change, "x" may be nearly one block tape-transit time as blocks may not start at the same point on the two lanes.

4. Stop time: 3 ms.
5. Turnaround time: 5 ms.
6. +2 ms by TCU to lock up and complete instruction.
7. Add 70-73 ms for lane change in an instruction.

DATAFILE INSTRUCTION TIMES (See Table G-2)

Computations are basically the same as those for the TSU; however, nominal stop, start, and turnaround times are larger.

1. Data Processor Fetch time: .09 ms (column 2)
   Execute time: (column 3)
2. Nominal tape speed and format
3. Start time: 4 ms + x; x = 1 (assumed)
4. Stop time: 4 ms
5. Turnaround time: 8 ms
6. +2 ms for TCU locking
7. Additional 8 ms on any search or scan instruction over TSU time.
8. Lane change on the same tape 40 ms
9. DATAFILE bin changes: 200 + 25B (B = number of bins passed over). If the head passes over the center of the file (48/49 ⇄ 50/51), add 100 ms

Using the nominal figures, times for the DATAFILE operations on one tape exceed equivalent TSU times by 5 ms or 13 ms.

Table G-1. 220 Magnetic-Tape—TSU—Instruction Times in ms

Instructions		Fetch Time	Execute Time = A + B + C			Comments
			A	B	C	
Op	Mnemonic	DP*	DP* TCU* TSU*	TCU* TSU*	TSU*	
50 50	MTS MFS	.09 ms	0.16 ms	16+N(3.2+.48k)+C		See Page A-16 for re-try times
50	MLS	.09	0.16	3 + C		
50 50	MRW MDA	.09 .09	0.16 0.16	50	100L	L = Length, in feet, of tape to be rewound
51 51	MTC MFC	.09 .09	0.16 0.16	16+(N+2) (3.2+.48k) +C		
52 53 56 57	MRD MRR MOW MOR	.09 .09 .09 .09	6+N(3.2+.48k)	12		
54 55	MIW MIR	.09 .09	6+N(3.2+.48k)	14		
58	MPF	.09	0.13	16+N(3.2+.48k)		
58	MPB	.09	0.13	6+N(3.2+.48k)		
58	MPE	.09	0.13	23+N(3.2+.48k)		
59 59	MIB MIE	.09 .09	a. 0.16 b. 0.14 c. 0.01			a. TCU Not busy, TSU at end of tape. Branch b. TCU Not busy, TSU not at end of tape. No Branch c. TCU Busy. No Branch

* These units are busy for times indicated. Omission of any of these three units from a column signifies that it is free for other purposes. N = Number of blocks traversed (except in MTC). In MTC, N = number of blocks traversed to block scanned for; the next block is traversed twice, resulting in the (N+2) coefficient. k = Number of words/block. C = 0, no lane change. C = 70, lane change.

Table G-2. 220 Magnetic-Tape—DATAFILE—Instruction Times in ms (nominal)

Instructions		Fetch Time	Execute Time = A + B + C			Comments
			A	B	C	
Op	Mnemonic	DP*	DP* TCU* TSU*	TCU* TSU*	TSU*	
50 50	MTS MFS	.09 .09	0.16	N(3.2+.48k) +29+ C+d(200+25B+q)		Re-try time (See Page A-16.)
50	MLS	.09 .09 .09	a. 0.16 b. 0.16 c. 0.16	a. 10 b. 40 c. 50	200+ 25B +q	a. No lane change b. Lane change on same tape c. Carriage movement to another tape
50 50	MRW MDA	.09 .09	0.16 0.16	50	100L	L = Length, in feet, of tape to be rewound
51 51	MTC MFC	.09 .09	0.16 0.16	(N+2) (3.2+.48k) +29+ C+d(200+25B+q)		
52 53 56 57	MRD MRR MOW MOR	.09	7+N(3.2+.48k)	15 (±1)		
54 55	MIW MIR	.09 .09	7+N(3.2+.48k)	17 (±1)		
58	MPF	.09	0.13	N(3.2+.48k) +21		
58	MPB	.09	0.13	N(3.2+.48k) +8		
58	MPE	.09	0.13	N+ (3.2+.48k) +23+5		
59 59	MIB MIE	.09 .09	a. 0.16 b. 0.14 c. 0.01			a. TCU Not busy; TSU at end of tape; Branch b. TCU Not busy; TSU not at end of tape; No Branch c. TCU Busy; No Branch

* These units are busy for times indicated. Omission of any of these three units from a column signifies that it is free for other purposes.

N = Number of blocks traversed (except in MTC).

In MTC, N = number of blocks traversed to block scanned for, the next block traversed twice, resulting in the (N+2) coefficient.

k = number of words/block.

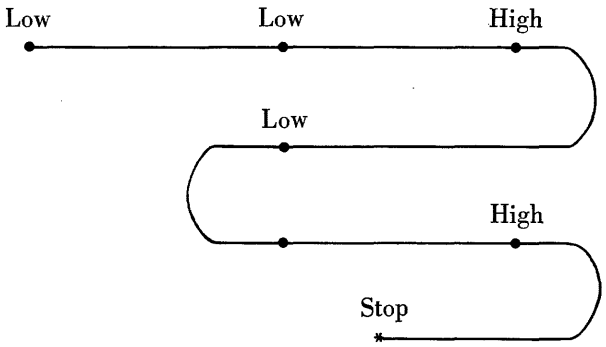
B = number of bins to be traversed by carriage to reach designated position.

	LANE CHANGE	MOVE TO ANOTHER TAPE	CARRIAGE MOVES OVER 48/49–50/51
YES:	C = 40	d = 1	q = 100
NO:	C = 0	d = 0	q = 0

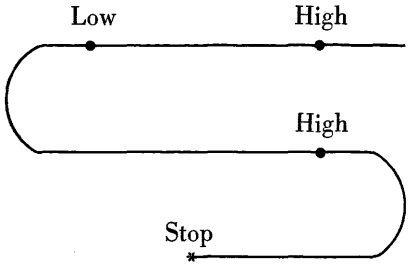
RECHECK OF KEY

MTS and MFS instructions incorporate the recheck-of-key or retry.

Forward Search Retry



Backward Search



Extra Time for Retry Pattern:

TSU: 14+g (3.2 + .48k) ms

DF : 18-1/2+g (3.2 + .48k) ms

g = 2 full pattern (forward)

g = 1 backward pattern

Burroughs Corporation ELECTRODATA DIVISION

BURROUGHS 220

A Summary of Operations in Alphabetic Order

Instruction Format	OPERATION	
± 1234 56 7890	ABR	NAME
± 1110 12 aaaa	ADD	ADD
± 1111 12 aaaa	ADA	ADD ABSOLUTE
± 1111 19 aaaa	ADL	ADD TO LOCATION
± 1110 35 aaaa	BCE	BRANCH, COMPARISON EQUAL
± 1110 34 aaaa	BCH	BRANCH, COMPARISON HIGH
± 1111 34 aaaa	BCL	BRANCH, COMPARISON LOW
± 1111 35 aaaa	BCU	BRANCH, COMPARISON UNEQUAL
± u111 38 aaaa	BCS	BRANCH, CONTROL SWITCH
± sLnn 30 aaaa	BFA	BRANCH, FIELD A
± sLnn 37 aaaa	BFR	BRANCH, FIELD R
± 1111 31 aaaa	BOF	BRANCH, OVERFLOW
± 1111 32 aaaa	BRP	BRANCH, REPEAT
± 111n 33 aaaa	BSA	BRANCH, SIGN A
± 1111 30 aaaa	BUN	BRANCH, UNCONDITIONALLY
± uivr 60 aaaa	CRD	CARD READ
± u111 62 aaaa	CRF	CARD READ, FORMAT LOAD
± u111 64 aaaa	CRI	CARD READ INTERROGATE, BRANCH
± u1cf 61 aaaa	CWR	CARD WRITE
± u111 63 aaaa	CWF	CARD WRITE, FORMAT LOAD
± u111 65 aaaa	CWI	CARD WRITE INTERROGATE, BRANCH
± 1111 45 1111	CLA	CLEAR A
± 1115 45 1111	CAB	CLEAR A, B
± 1110 10 aaaa	CAD	CLEAR, ADD
± 1111 10 aaaa	CAA	CLEAR, ADD ABSOLUTE
± 1113 45 1111	CAR	CLEAR A, R
± 1117 45 1111	CLT	CLEAR A, R, B
± 1114 45 1111	CLB	CLEAR B
± 1111 46 aaaa	CLL	CLEAR LOCATION
± 1112 45 1111	CLR	CLEAR R
± 1116 45 1111	CRB	CLEAR R, B
± 1110 11 aaaa	CSU	CLEAR, SUBTRACT
± 1111 11 aaaa	CSA	CLEAR, SUBTRACT ABSOLUTE
± sLf0 18 aaaa	CFA	COMPARE FIELD A
± sLf1 18 aaaa	CFR	COMPARE FIELD R
± nnnn 21 aaaa	DBB	DECREASE B, BRANCH
± sLnn 27 aaaa	DFL	DECREASE FIELD LOCATION
± sLnn 28 aaaa	DLB	DECREASE FIELD LOCATION, LOAD B
± 1111 15 aaaa	DIV	DIVIDE
± 1111 17 aaaa	EXT	EXTRACT
± n1l0 22 aaaa	FAD	FLOATING ADD
± n1l1 22 aaaa	FAA	FLOATING ADD ABSOLUTE
± 1111 25 aaaa	FDV	FLOATING DIVIDE
± 1111 24 aaaa	FMU	FLOATING MULTIPLY
± n1l0 23 aaaa	FSU	FLOATING SUBTRACT
± n1l1 23 aaaa	FSA	FLOATING SUBTRACT ABSOLUTE
± 1111 00 1111	HLT	HALT
± nnnn 20 aaaa	IBB	INCREASE B, BRANCH
± sLnn 26 aaaa	IFL	INCREASE FIELD LOCATION
± 1111 08 1111	KAD	KEYBOARD ADD
± 1110 42 aaaa	LDB	LOAD B
± 1111 42 aaaa	LBC	LOAD B COMPLEMENT
± 1111 41 aaaa	LDR	LOAD R
± 111n 43 1111	LSA	LOAD SIGN A
4 uhhk 51 aaaa	MFC	MAGNETIC-TAPE FIELD SCAN
4 uhh0 50 aaaa	MFS	MAGNETIC-TAPE FIELD SEARCH
± unkk 54 aaaa	MIW	MAGNETIC-TAPE INITIAL WRITE
± un11 55 aaaa	MIR	MAGNETIC-TAPE INITIAL WRITE, RECORD
± u110 59 aaaa	MIB	MAGNETIC-TAPE INTERROGATE, BRANCH
± u111 59 aaaa	MIE	MAGNETIC-TAPE INTERROGATE END OF TAPE, BRANCH
± uhh4 50 1111	MLS	MAGNETIC-TAPE LANE SELECT
± unkk 56 aaaa	MOW	MAGNETIC-TAPE OVERWRITE
± un11 57 aaaa	MOR	MAGNETIC-TAPE OVERWRITE, RECORD
± u112 58 1111	MPE	MAGNETIC-TAPE POSITION AT END OF INFORMATION
± un11 58 1111	MPB	MAGNETIC-TAPE POSITION BACKWARD
± un10 58 1111	MPF	MAGNETIC-TAPE POSITION FORWARD
± univ 52 aaaa	MRD	MAGNETIC-TAPE READ
± univ 53 aaaa	MRR	MAGNETIC-TAPE READ, RECORD
± uhh8 50 1111	MRW	MAGNETIC-TAPE REWIND
± uhh9 50 1111	MDA	MAGNETIC-TAPE REWIND, DE-ACTIVATE
0 uhhk 51 aaaa	MTC	MAGNETIC-TAPE SCAN
0 uhh0 50 aaaa	MTS	MAGNETIC-TAPE SEARCH
± 1111 14 aaaa	MUL	MULTIPLY
± 1111 01 1111	NOP	NO OPERATION
± unnv 03 aaaa	PRD	PAPER-TAPE READ
± u11v 04 aaaa	PRB	PAPER-TAPE READ, BRANCH
± unnv 05 aaaa	PRI	PAPER-TAPE READ, INVERSE FORMAT
± unn1 06 aaaa	PWR	PAPER-TAPE WRITE
± u111 07 aaaa	PWI	PAPER-TAPE WRITE INTERROGATE, BRANCH
± 1nn1 29 aaaa	RTF	RECORD TRANSFER
± 1111 16 1111	RND	ROUND
± 1110 49 11nn	SLA	SHIFT LEFT A
± 1111 49 11nn	SLT	SHIFT LEFT A AND R
± 1112 49 11nn	SLS	SHIFT LEFT A WITH SIGN
± 1110 48 11nn	SRA	SHIFT RIGHT A
± 1111 48 11nn	SRT	SHIFT RIGHT A AND R
± 1112 48 11nn	SRS	SHIFT RIGHT A WITH SIGN
± sLf0 40 aaaa	STA	STORE A
± sLf2 40 aaaa	STB	STORE B
± 1111 44 aaaa	STP	STORE P
± sLf1 40 aaaa	STR	STORE R
± 1110 13 aaaa	SUB	SUBTRACT
± 1111 13 aaaa	SUA	SUBTRACT ABSOLUTE
± dnnv 09 aaaa	SPO	SUPERVISORY PRINT-OUT

Burroughs Corporation ELECTRODATA DIVISION

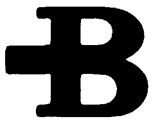
BURROUGHS 220 Alphanumeric codes

SYMBOL		CODE							
			Paper Tape						
			Channel						
Burroughs 220	046	Computer and Magnetic Tape	X	0	Check	8	4	2	1	Punched Card
(Space)	(Space)	00			✓					(Blank)
.	.	03	X	0		8		2	1	12 8-3
¤	¤	04	X	0	✓	8	4			12 8-4
&	&	10	X	0	✓					12
$	$	13	X		✓	8		2	1	11 8-3
*	*	14	X			8	4			11 8-4
—	—	20	X							11
/	/	21		0	✓				1	0 1
,	,	23		0	✓	8		2	1	0 8-3
%	%	24		0		8	4			0 8-4
#	#	33				8		2	1	8-3
@	@	34			✓	8	4			8-4
A	A	41	X	0					1	12 1
B	B	42	X	0				2		12 2
C	C	43	X	0	✓			2	1	12 3
D	D	44	X	0			4			12 4
E	E	45	X	0	✓		4		1	12 5
F	F	46	X	0	✓		4	2		12 6
G	G	47	X	0			4	2	1	12 7
H	H	48	X	0		8				12 8
I	I	49	X	0	✓	8			1	12 9
J	J	51	X		✓				1	11 1
K	K	52	X		✓			2		11 2
L	L	53	X					2	1	11 3
M	M	54	X		✓		4			11 4
N	N	55	X				4		1	11 5
O	O	56	X				4	2		11 6
P	P	57	X		✓		4	2	1	11 7
Q	Q	58	X		✓	8				11 8
R	R	59	X			8			1	11 9
S	S	62		0	✓			2		0 2
T	T	63		0				2	1	0 3
U	U	64		0	✓		4			0 4
V	V	65		0			4		1	0 5
W	W	66		0			4	2		0 6
X	X	67		0	✓		4	2	1	0 7
Y	Y	68		0	✓	8				0 8
Z	Z	69		0		8			1	0 9
0 (Zero)	0 (Zero)	80		0						0
1	1	81							1	1
2	2	82						2		2
3	3	83			✓			2	1	3
4	4	84					4			4
5	5	85			✓		4		1	5
6	6	86			✓		4	2		6
7	7	87					4	2	1	7
8	8	88				8				8
9	9	89			✓	8			1	9
TAB	SKIP	26		0	✓	8	4	2		0 8-6
CARRIAGE RETURN	CR	16	X		✓	8	4	2		11 8-6
FORM OUT	PI 6	15	X		✓	8	4		1	11 8-5
END-OF-WORD	PI 7	35				8	4		1	8-5
BLANK	SP 1	02	X	0	✓	8		2		12 8-2
TAPE FEED	TAPE FEED	*	X	0	✓	8	4	2	1	12 8-7
Not assigned	END CARD 1	36				8	4	2		8-6
	END CARD 2	27		0		8	4	2	1	0 8-7
	CORRECT TAB	37			✓	8	4	2	1	8-7
	ERROR	17	X			8	4	2	1	11 8-7
	PI 1	32			✓	8		2		8-2
	PI 2	12	X			8		2		11 8-2
	PI 3	22		0		8		2		0 8-2
	PI 4	25		0	✓	8	4		1	0 8-5
	PI 5	05	X	0		8	4		1	12 8-5
	SP 2	06	X	0		8	4	2		12 8-6

*Not represented.

40
50

12-0
11-0



Burroughs Corporation

Printed in U.S.A.

BULLETIN 5020A