

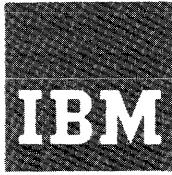
# IBM

## Programmed Instruction Course

1401 DPS-BASIC PROGRAMMING  
Symbolic Programming System  
Student Materials

Education Development - Endicott, New York

## **Reference Materials**



**Systems Reference Library**

**IBM 1401 Symbolic Programming Systems: SPS-1 and SPS-2  
Specifications and Operating Procedures**

The following excerpt is reprinted from page 12  
of IBM Systems Reference Library, form C24-1480.

**Comments (Columns 40-55)**

This field is reserved for programmer's notes or comments about a particular entry. A source program that contains a complete set of comments can be more easily understood and traced by all persons concerned with a given program. The comments have no effect on the object program as it is assembled or executed. Columns 56-75 of source program cards must be left blank, or incorrect processing will occur.

**COMMENTS CARD**

To provide the programmer with the ability to insert more extensive descriptive information in the program listing than is possible by using the comments field on a program entry card, a comments card may be included in the source program deck.

Comments cards will not be assembled nor will they affect the assembling procedure. When encountered by the processor, they will be reproduced unaltered in the SPS output deck, and will be bypassed when the object program is being loaded.

*The Programmer:*

1. Indicates with an asterisk in the first position of the label field (column 8) that the card is a comments card.
2. May write the comment beginning at any position (columns 9-55). Comments extending beyond position 55 may cause an error during processing.

*The Processor:* Reproduces (unaltered) the comment in proper sequence in the program listing.

LINE	COUNT	LABEL	OPERATION	(A) OPERAND				(B) OPERAND				COMMENTS		
				ADDRESS	±	CHAR. ADJ.	IND.	ADDRESS	±	CHAR. ADJ.	IND.			
3	5	8	13	14	16	17	23	27	28	34	36	39	40	55
01	0	*PAYROLL												LISTING ROUTINE PROGRAMMED FOR THE 1401

Figure 12. Typical Entry on an SPS Comments Card

The following excerpt is reprinted from page 20 of IBM Systems Reference Library, form C24-1480.

#### **END — End**

*General Description:* An END statement is a signal to the processor that the last card in the source program has been processed. If the programmer specifies in the (A) operand the actual or symbolic address at which the object program is to begin execution, an END statement will produce an instruction that will start program execution immediately after loading. If the (A) operand is blank, the 1401 will halt when the last instruction has been loaded.

#### *The Programmer:*

1. writes END in the operation field.
2. may write a symbolic blank, or actual machine address (left-justified) in the (A) operand. An asterisk operand is not permissible.

*The Processor* clears the read area (positions 001-080) of core storage and assembles an instruction that branches to the address specified in the (A) operand after loading is completed.

# 1401 Instruction Set

## Reference Manual

### CONTENTS

IBM card character code . . . . .	2	1401 BCD character code . . . . .	3
-----------------------------------	---	-----------------------------------	---

### LOGIC OPERATIONS

Branch (unconditional) . . . . .	5	Branch if indicator on . . . . .	9
Branch if character equal . . . . .	6	Halt . . . . .	10
Branch if word mark and/or zone . . . . .	7	Halt and branch . . . . .	11
Compare . . . . .	8		

### DATA TRANSFER OPERATIONS

Move numerical . . . . .	12	Clear storage . . . . .	18
Move zone . . . . .	13	Clear storage and branch . . . . .	19
Set word marks . . . . .	14	Move characters to word mark . . . . .	20
Set word mark . . . . .	15	Move characters and suppress zeros . . . . .	21
Clear word marks . . . . .	16	Move characters and edit . . . . .	22
Clear word mark . . . . .	17	Load characters to A-field word mark . . . . .	23

### ARITHMETIC OPERATIONS

Zero and add . . . . .	24	Subtract (two fields) . . . . .	28
Zero and subtract . . . . .	25	Subtract (one field) . . . . .	29
Add (two fields) . . . . .	26	Multiply . . . . .	30
Add (one field) . . . . .	27	Divide . . . . .	31

### INPUT/OUTPUT OPERATIONS

Read a card . . . . .	33	Control carriage . . . . .	37
Punch a card . . . . .	34	Combination input/output . . . . .	38
Select stacker . . . . .	35	Input/output and branch . . . . .	39
Write a line . . . . .	36		

### INDEX

Operation codes . . . . .	40
---------------------------	----

IBM CARD CHARACTER CODE		DIGIT PUNCH												
		NO	0	1	2	3	4	5	6	7	8	9	3-8	4-8
ZONE PUNCH	12	&	plus zero	A	B	C	D	E	F	G	H	I	.	□
	11	-	minus zero	J	K	L	M	N	O	P	Q	R	\$	*
	0			/	S	T	U	V	W	X	Y	Z	,	%
	NO	blank	zero	0	1	2	3	4	5	6	7	8	9	# @

Translation from card code to BCD code (and vice versa) is automatic. During card reading, digit punches are automatically translated into digit bits, and zone punches into zone bits, which are put into the read area of storage. (Conversely, during card punching, the BCD characters in the punch area are automatically translated into card code.)

The 9-row of the card is read first; for any punch in the 9-row, 81-bits are generated and put into the corresponding read area position. Then the 8-row is read; punches in the 8-row are translated into 8-bits. Punches in the 7-row are translated into 421-bits, and so on.

Certain symbols are made up of combinations of the 3-punch and the 8-punch, or 4-punch and 8-punch. Each punch of a combination is read separately, and translated into the corresponding bits. For example, the 3-8 punch combination is translated into 821-bits. Other digit punch combinations that can be read by the 1402 are 5-8, 6-8, and 7-8; these combinations become 841, 842, and 8421-bit configurations, and are shown on the BCD chart on the opposite page.

Unless the 1401 is equipped with a special device, other punch combinations cannot be read. For example, the 3-5 combination is not valid, because the "5" would be translated into 41-bits, the "3" into 21-bits, and together, they would form a 421-bit configuration, which represents "7", not "3" or "5". The 1401 System checks for such invalid punch combinations, and stops automatically if one is detected.

Punches in the 0-row may be either digit or zone punches. When a 0-punch is read, a check is made to see whether any digit punch was previously read in that column. If so, the 0-punch is treated as a zone, and is translated into an A-bit. If not, the 0-punch is treated as a digit, and translated into 82-bits. Punches in the 11-row are translated into B-bits, and punches in the 12-row are translated into BA-bits.

1401 BCD CHARACTER CODE		DIGIT BITS															
		NO BITS	1	2	21	4	41	42	421	8	81	82	821	84	841	842	8421
ZONE BITS	B AND A	&	A	B	C	D	E	F	G	H	I	plus zero ⊕	.	⊖	⊗	⊘	⊙
	B ONLY	-	J	K	L	M	N	O	P	Q	R	minus zero ⊖	\$	*	]	;	Δ
	A ONLY	‡	/	S	T	U	V	W	X	Y	Z	‡	,	%	√	∖	‡‡
	NO BITS	blank	1	2	3	4	5	6	7	8	9	zero 0	#	@	:	>	✓

Characters that are shaded print as blank spaces on a standard 1403

For arithmetic and certain other operations, only the numerical portions of characters are acted upon by the 1401. The numerical portion of a character that is to be processed arithmetically may be any value from 0 to 9; the numerical portion is equal to the value of the digit bits of the character. The numerical portion of the letter "A", for instance, is 1, because it is made up of a 1-bit (besides the BA-bits). This means that in an arithmetic operation, the letter "A" will be treated as a "1".

The numerical portion of a character that is made up of more than one digit bit may be obtained by adding the digit-bit values. For example, the letter "P" is made up of 421-bits (and an A-bit); its numerical portion is 7 (4+2+1). The numerical portion of the letter "E" is 5 (4+1), since it is made up of BA41-bits. (Characters with 82-bits are an exception to this rule; 82-bits equal zero as far as the 1401 is concerned.)

Zone bits serve as signs for numbers, as well as zones for letters and symbols. The sign of a number is identified by the zone bits in the low-order position.

The standard plus sign is BA-bits. However, numbers with no zone bits, with an A-bit only, or with BA-bits, in the low-order position, are regarded as "positive" numbers. For example, the numbers "25", "2V", and "2E" all are regarded as "+25".

The standard minus sign is a B-bit. Only numbers with a B-bit in the low-order position are regarded as "negative" numbers. So, the number "2N" is regarded as "-25".





OPERATION - BRANCH (UNCONDITIONAL)

OP CODE - B

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X		

PURPOSE OF INSTRUCTION

To cause a branch, under all conditions, to the instruction whose address is specified by the (A) operand.

**OPERATION - BRANCH IF CHARACTER EQUAL**

**OP CODE - B**

**INSTRUCTION PARTS**

<b>OP CODE</b>	<b>(A) OPERAND ADDRESS</b>	<b>(B) OPERAND ADDRESS</b>	<b>d CHARACTER</b>
<b>X</b>	<b>X</b>	<b>X</b>	<b>X</b>

**PURPOSE OF INSTRUCTION**

To cause a branch to the instruction whose address is specified by the (A) operand, if the character in the position whose address is given by the (B) operand is the same as the d-character.

Note: A word mark in the position being tested is not regarded as part of the character.

**OPERATION - BRANCH IF WORD MARK AND/OR ZONE**

**OP CODE - BWZ**

**INSTRUCTION PARTS**

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	X

**PURPOSE OF INSTRUCTION**

To cause a branch to the instruction whose address is specified by the (A) operand, if the position whose address is given by the (B) operand contains the word mark and/or zone-bit **condition** that is represented by the d-character.

**WORD MARK AND/OR ZONE BIT CONDITION**

**d-CHARACTER**

Word mark	1
Both B-bit and A-bit (12-zone)	B
B-bit, but no A-bit (11-zone)	K
A-bit, but no B-bit (0-zone)	S
Neither B-bit nor A-bit (no zone)	2
Either a word mark or a 12-zone	C
Either a word mark or an 11-zone	L
Either a word mark or a 0-zone	T
Either a word mark or no zone	3

**OPERATION - COMPARE**

**OP CODE - C**

**INSTRUCTION PARTS**

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

**PURPOSE OF INSTRUCTION**

To turn on a "compare" indicator by comparing a field of characters whose right-most address is given by the (B) operand with another field of characters whose right-most address is given by the (A) operand.

**Note:** Only characters are compared. Word marks identify the high-order positions of the fields being compared, and thereby stop the compare operation, but the word marks themselves are not compared.

Result of Comparing Value of Characters in B-Field to Value of Characters in A-Field	Compare Indicator That is Turned On	Code Name Of Indicator
B is not equal to A	Unequal	/
B is equal to A	Equal	S
B is less than A	Low	T
B is more than A	High	U

**Note:** Only the "unequal" compare indicator is standard on all 1401 Systems. The other indicators are available as an optional feature.

OPERATION - BRANCH IF INDICATOR ON

OP CODE - B

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X		X

PURPOSE OF INSTRUCTION

To cause a branch to the instruction whose address is specified by the (A) operand, if the indicator represented by the d-character is on.

INDICATOR	d-CHARACTER
Channel 9 (form overflow)	9
Channel 12 (form overflow)	@
Last card	A
Sense Switch B	B
Sense Switch C	C
Sense Switch D	D
Sense Switch E	E
Sense Switch F	F
Sense Switch G	G
Unequal compare	/
Equal compare	S
Low Compare	T
High compare	U

**OPERATION - HALT**

**OP CODE - H**

**INSTRUCTION PARTS**

<b>OP CODE</b>	<b>(A) OPERAND ADDRESS</b>	<b>(B) OPERAND ADDRESS</b>	<b>d CHARACTER</b>
<b>X</b>			

**PURPOSE OF INSTRUCTION**

To cause the 1401 DPS to stop acting on instructions, and to turn on the red Stop key light on the Console.

Note: Pressing the Start key causes the 1401 to restart at the next sequential instruction.

**OPERATION - HALT AND BRANCH**

**OP CODE - H**

**INSTRUCTION PARTS**

<b>OP CODE</b>	<b>(A) OPERAND ADDRESS</b>	<b>(B) OPERAND ADDRESS</b>	<b>d CHARACTER</b>
<b>X</b>	<b>X</b>		

**PURPOSE OF INSTRUCTION**

To cause the 1401 DPS to stop acting on instructions, to turn on the red Stop key light on the console, and then to branch if the Start key is pressed.

Note: Pressing the Start key causes the 1401 to restart at the instruction whose address is specified by the (A) operand.

OPERATION - MOVE NUMERICAL

OP CODE - MN

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

#### PURPOSE OF INSTRUCTION

To transfer the numerical bits in the position whose address is specified by the (A) operand to the position whose address is specified by the (B) operand.

The status of the zone bits and the word mark in the position specified by the (B) operand is not altered.

None of the bits in the position specified by the (A) operand are altered. This position will have exactly the same bits after the operation as it had before the operation.



OPERATION - MOVE ZONE

OP CODE - MZ

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

#### PURPOSE OF INSTRUCTION

To transfer the zone bits in the position whose address is specified by the (A) operand to the position whose address is specified by the (B) operand.

The status of the numerical bits and the word mark in the position specified by the (B) operand is not altered.

None of the bits in the position specified by the (A) operand are altered. This position will have exactly the same bits after the operation as it had before the operation.

**OPERATION - SET WORD MARKS (2 POSITIONS)**

**OP CODE - SW**

**INSTRUCTION PARTS**

<b>OP CODE</b>	<b>(A) OPERAND ADDRESS</b>	<b>(B) OPERAND ADDRESS</b>	<b>d CHARACTER</b>
<b>X</b>	<b>X</b>	<b>X</b>	

**PURPOSE OF INSTRUCTION**

To set word marks in the positions specified by the (A) and (B) operands.

Note: The numerical and zone bits are not changed in the positions where the word marks are set.

**OPERATION - SET WORD MARK (1 POSITION)**

**OP CODE - SW**

**INSTRUCTION PARTS**

<b>OP CODE</b>	<b>(A) OPERAND ADDRESS</b>	<b>(B) OPERAND ADDRESS</b>	<b>d CHARACTER</b>
<b>X</b>	<b>X</b>		

**PURPOSE OF INSTRUCTION**

To **set** a word mark in the position specified by the (A) operand.

**Note:** The numerical and zone bits are not changed in the position where the word mark is set.

OPERATION - CLEAR WORD MARKS (2 POSITIONS)

OP CODE - CW

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

#### PURPOSE OF INSTRUCTION

To remove the word marks that are in the locations specified by the (A) and (B) operands.

Note: The numerical and zone bits are not changed in the positions from which the word marks are removed.

OPERATION - CLEAR WORD MARK (1 POSITION)

OP CODE - CW

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X		

PURPOSE OF INSTRUCTION

To remove the word mark that is in the position specified by the (A) operand.

Note: The numerical and zone bits are not changed in the position from which the word mark is removed.

## OPERATION - CLEAR STORAGE

OP CODE - CS

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X		

### PURPOSE OF INSTRUCTION

To set to "blank" as few as 1 or as many as 100 storage positions.

The (A) operand specifies the right-most position to be cleared ("blanked out").

The 1401 then proceeds to the left and continues to clear one position at a time, until it comes to a position whose address ends in "00". The operation automatically stops after the position ending in "00" has been cleared.

For example, if the 1401 is instructed to clear storage starting with position 0299, it will clear the 100 positions from 0299 down to 0200, inclusive. If the (A) operand address is "0301", then 2 positions will be cleared, 0301 and 0300. If the starting address is "0300", only position 0300 will be cleared because the 1401 will immediately come to the "00" address that stops the operation.

Note: This operation removes zone bits, numerical bits, and word marks. Each position that is cleared will contain just a check bit (C-bit) after the operation has been performed.

**OPERATION - CLEAR STORAGE AND BRANCH**

**OP CODE - CS**

**INSTRUCTION PARTS**

<b>OP CODE</b>	<b>(A) OPERAND ADDRESS</b>	<b>(B) OPERAND ADDRESS</b>	<b>d CHARACTER</b>
<b>X</b>	<b>X</b>	<b>X</b>	

**PURPOSE OF INSTRUCTION**

To clear storage beginning with the address specified by the (B) operand, and then to branch to the instruction whose address is specified by the (A) operand.

The operation of clearing storage works exactly as described for "Clear storage", starting with the specified address and stopping when a position is reached whose address ends in "00". However, in this instruction, the first position to be cleared is indicated by the (B) operand, not by the (A) operand.

The (A) operand indicates the "branch-to" address.

OPERATION - MOVE CHARACTERS TO WORD MARK

OP CODE - MCW

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

#### PURPOSE OF INSTRUCTION

To transfer characters, one at a time, starting from the address specified by the (A) operand, to the address specified by the (B) operand.

This operation stops when a word mark is detected in either an A-field position or a B-field position.

Word marks are not moved by this operation.



OPERATION - MOVE CHARACTERS AND SUPPRESS ZEROS

OP CODE - MCS

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

#### PURPOSE OF INSTRUCTION

To transfer characters, one at a time, starting from the address specified by the (A) operand, to the address specified by the (B) operand.

The transfer of characters stops only when a word mark is detected in an A-field position.

Those transferred characters that are non-significant are replaced with blanks.

Zone bits (if there are any) are removed from the low-order character as it is transferred to the B-field. In other words, the "sign" of the field will not be transferred.

Word marks in the B-field are removed.

**OPERATION - MOVE CHARACTERS AND EDIT**

**OP CODE - MCE**

**INSTRUCTION PARTS**

<b>OP CODE</b>	<b>(A) OPERAND ADDRESS</b>	<b>(B) OPERAND ADDRESS</b>	<b>d CHARACTER</b>
<b>X</b>	<b>X</b>	<b>X</b>	

**PURPOSE OF INSTRUCTION**

To transfer characters, one at a time, starting from the address specified by the (A) operand to the address specified by the (B) operand, under the control of an edit control word previously loaded into the B-field.

The zone bits of the low-order character are removed during its transfer to the B-field.

A word mark is required in the B-field to stop the transfer of characters. This word mark in the B-field is removed during the operation.

OPERATION - LOAD CHARACTERS TO A-FIELD WORD MARK

OP CODE - LCA

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

PURPOSE OF INSTRUCTION

To transfer characters, one at a time, starting from the address specified by the (A) operand, to the address specified by the (B) operand.

This operation stops when a word mark is detected in an A-field position. The word mark in the A-field is transferred, with the character, to the B-field. Original word marks in the B-field positions are removed.

## OPERATION - ZERO AND ADD

OP CODE - ZA

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

### PURPOSE OF INSTRUCTION

To set every position of the B-field to zero, and then to move the numerical portion of each A-field character to the B-field. If the A-field is shorter than the B-field, then the excess B-field positions will contain zeros when the operation is finished. The contents of the A-field are not changed.

The B-field must have a word mark, to indicate how many positions are to be set to zero. The A-field must have a word mark if it has fewer positions than the B-field. The A-field does not have to have a word mark if the fields are equal in length.

A standard plus sign or minus sign is generated and moved to the low-order position of the B-field. Thus, the result of a "zero and add" operation is always signed (whether or not the A-field is signed), and the sign is always a standard plus or minus sign.

A standard plus sign (BA-bits) is generated if the A-field is positive, that is, if the low-order position of the A-field has no zone bits, or an A-bit, or BA-bits.

A standard minus sign (B-bit) is generated if the A-field is negative, that is, if the low-order position of the A-field has a B-bit.

(In arithmetic instructions, the (A) operand specifies the address of the low-order position of the A-field, and the (B) operand specifies the address of the low-order position of the B-field.)

## OPERATION - ZERO and SUBTRACT

OP CODE - ZS

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

### PURPOSE OF INSTRUCTION

To set every position of the B-field to zero, and then to move the numerical portion of each A-field character to the B-field. If the A-field is shorter than the B-field, then the excess B-field positions will contain zeros when the operation is finished.

The B-field must have a word mark to indicate how many positions are to be set to zero. The A-field must have a word mark if it has fewer positions than the B-field. The A-field does not have to have a word mark if the fields are equal in length.

A standard plus sign or minus sign is generated and moved to the low-order position of the B-field. Thus, the result of a "zero and subtract" operation is always signed (whether or not the A-field is signed), and the sign is always a standard plus or minus sign. The sign of the result will be plus if the A-field is negative, and minus if the A-field is positive.

A standard plus sign (BA-bits) is generated if the A-field is negative, that is, if the low-order position of the A-field has a B-bit.

A standard minus sign (B-bit) is generated if the A-field is positive, that is, if the low-order position of the A-field has no zone bits, or an A-bit, or BA-bits.

## OPERATION - ADD (2 FIELDS)

OP CODE - A

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

### PURPOSE OF INSTRUCTION

To add the number in the A-field to the number in the B-field. The result is produced in the B-field, and it replaces the original number in that field. For example, if the A-field contains "20" and the B-field contains "080" before an "add" operation, then after the operation the A-field will still contain "20", but the B-field will contain "100".

The B-field must have a word mark in its high-order position. The A-field must have a word mark if it is shorter than the B-field; it does not have to have a word mark if it is the same length as the B-field.

If the numbers are both positive, or both negative, the 1401 will find the sum of the numbers. ~~The zone bits (if any) in the low-order position of the B-field will not be changed.~~

If one number is positive and one is negative, the 1401 will find the difference between the numbers, and it will generate a standard sign for the result.

The result will be signed plus (BA-bits) if:

1. the larger number is positive, or if
2. the numbers have equal numerical values and the original B-field is positive.

The result will be signed minus (B-bit) if:

1. the larger number is negative, or if
2. the numbers have equal numerical values and the B-field is negative.

OPERATION - ADD (1 FIELD)

OP CODE - A

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X		

#### PURPOSE OF INSTRUCTION

To add the number in the A-field to itself, that is, to double the value of the number in the A-field. The result is placed into the A-field, and it replaces the original number.

The A-field must have a word mark.

The zone bits (if any) in the low-order position of the A-field will not be changed.

**OPERATION - SUBTRACT (2 FIELDS)**

**OP CODE - S**

**INSTRUCTION PARTS**

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

**PURPOSE OF INSTRUCTION**

To subtract the number in the A-field from the number in the B-field. The result is produced in the B-field, and it replaces the original number in that field.

The B-field must have a word mark in its high-order position. The A-field must have a word mark if it is shorter than the B-field; it does not have to have a word mark if it is the same length as the B-field.

If the numbers are both positive, or both negative, the 1401 will find the difference between the numbers, and it will generate a standard sign for the result.

The result will be signed plus (BA-bits) if:

1. the numbers are positive and the larger number is in the B-field, or if
2. the numbers are negative and the larger number is in the A-field, or if
3. the numbers are positive and equal in numerical value.

The result will be signed minus (B-bit) if:

1. the numbers are positive and the larger number is in the A-field, or if
2. the numbers are negative and the larger number is in the B-field, or if
3. the numbers are negative and equal in numerical value.

If one number is positive and one is negative, the 1401 will find the sum of the numbers. The zone bits (if any) in the low-order position of the B-field will not be changed.



OPERATION - SUBTRACT (1 FIELD)

OP CODE - S

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X		

PURPOSE OF INSTRUCTION

To subtract the number in the A-field from itself, that is, to reduce the value of the number in the A-field to zero.

The A-field must have a word mark.

If the number is positive (BA-bits, A-bit, or no zone bits in the low-order position), the result will be signed plus (BA-bits).

If the number is negative (B-bit in the low-order position), the result will be signed minus (B-bit).

## OPERATION - MULTIPLY

OP CODE - M

### INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

### PURPOSE OF INSTRUCTION

To develop a product by multiplying two numbers. One of the numbers is in the left-most positions of a "work area" in which the product will be developed; this work area serves as the B-field. The other number is in the A-field.

Both the A-field and the B-field (work area) must have word marks in their high-order positions.

The work area must have one more position than the sum of the positions in the numbers to be multiplied. For example, if the numbers to be multiplied are "0379" (4 positions) and "0009625" (7 positions), then the work area must have 12 positions ( $4 + 7 + 1$ ).

Prior to the "multiply" operation, one of the numbers to be multiplied is placed into the left-most positions of the work area. Usually, either a "move" (MCW) or a "zero and add" (ZA) operation is used to do this.

In the "Multiply" instruction, the (A) operand is the address of the low-order position of the A-field. The (B) operand is the address of the low-order position of the work area (B-field).

After the "multiply" operation, the number in the A-field is unchanged. The number that had been placed into the work area has been reduced to zero; the work area now contains the product in its right-most positions. The product is always signed with a standard plus or minus sign.

The sign of the product will be plus (BA-bits) if the numbers to be multiplied are both positive, or both negative.

The sign of the product will be minus (B-bit) if one of the numbers to be multiplied is positive and one is negative.

## OPERATION - DIVIDE

OP CODE - D

INSTRUCTION PARTS

CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X	X	

### PURPOSE OF INSTRUCTION

To develop a quotient and a remainder by dividing one number (dividend) by another number (divisor). The dividend is in the right-most positions of a "work area" in which the quotient and remainder will be developed; this work area serves as the B-field. The divisor is in the A-field.

Both the A-field and the B-field (work area) must have word marks in their high-order positions.

The work area must have one more position than the sum of the positions in the divisor and dividend. For example, if "0360" (dividend) is to be divided by "012" (divisor), then the work area must have 8 positions (4 + 3 + 1).

Prior to the "divide" operation, the dividend is placed into the right-most positions of the work area, using a "zero and add" (ZA) operation.

In the "Divide" instruction, the (A) operand is the address of the low-order position of the A-field, which contains the divisor. The (B) operand is the address of the high-order character of the dividend in the work area.

After the "divide" operation, the divisor in the A-field is unchanged. The quotient is in the left-most positions of the work area, and the remainder is in the right-most positions of the work area. Both the quotient and the remainder will be signed with standard plus or minus signs.

The sign of the quotient will be plus (BA-bits) if the factors (dividend and divisor) are both positive, or both negative.

The sign of the quotient will be minus (B-bit) if one of the factors is positive and one is negative.

The sign of the remainder will be the same as the sign of the dividend.



OPERATION - READ A CARD

OP CODE - R

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X			

#### PURPOSE OF INSTRUCTION

To move a card past the reading mechanism in the 1402 Card Read-Punch, and to cause the data punched in the card to be placed into storage positions 0001-0080 in the 1401 Processing Unit. Word marks in the read area are not changed.

After it has been read, the card is moved on and 10 milliseconds (10-thousandths of a second) later it falls into a stacker. The card automatically falls into the NR (Normal Read) stacker, unless another stacker (either stacker 1 or stacker 8/2) is selected before the 10 milliseconds have elapsed.

Ordinarily, if a "select stacker" instruction is to be given, it is given right after the "Read a card" instruction.

However, other processing or decision-making instructions may precede the "Select stacker" instruction. Ten milliseconds is a fairly "long" period of time, so far as the 1401 is concerned; as a rule of thumb, about 40 "add" or "move" operations or about 90 "branch if character equal" operations can be performed in that period of time. So, there is certainly enough time available in which, for example, to determine what control punches the card contains and to select a particular stacker depending on the punches in the card.

While a card is being read, the data punched in it is checked to determine whether it is "valid" data. If any column is punched with invalid data, the card is automatically stacked in the NR stacker, whether or not a "Select stacker" instruction is given in the program, and the 1401 System automatically stops. An example of invalid data in a card is "multiple-digit" punching; a 1401 System cannot properly read a column that is punched with both the digit "4" and the digit "7", for example, unless the 1401 is equipped with a special feature.

If the card that is read is the last card of the input file, and if sense switch "A" (on the 1401 console) is on, then the Last Card indicator will turn on after the card has been read.

**OPERATION - PUNCH A CARD**

**OP CODE - P**

**INSTRUCTION PARTS**

<b>OP CODE</b>	<b>(A) OPERAND ADDRESS</b>	<b>(B) OPERAND ADDRESS</b>	<b>d CHARACTER</b>
<b>X</b>			

**PURPOSE OF INSTRUCTION**

To move a card past the punching mechanism in the 1402 Card Read-Punch, and to cause the data from storage positions 0101-0180 in the 1401 Processing Unit to be punched into the card. Word marks are not punched.

After it has been punched, the card stops. It is not moved to the stackers until the next time a card is punched. So, during the execution of a "Punch a card" instruction, the previously-punched card falls into a stacker at the same time that another card is being punched. The card automatically falls into the NP (Normal Punch) stacker, unless another stacker (either stacker 4 or stacker 8/2) is selected.

There is no specific time limit for selecting a stacker. Ordinarily, the "Select stacker" instruction is given right after the "Punch a card" instruction. This causes the desired stacker to be selected, even though the actual stacking of the card does not take place until the next "punch a card" operation.

At the same time that an output card is being moved to the stackers, that card is checked to make sure that it is punched correctly. If any column is not punched correctly, the card is automatically stacked in the NP stacker, whether or not a "Select stacker" instruction is given in the program, and the 1401 System automatically stops. An example of an incorrectly punched card is one that has punches in columns whose corresponding punch area positions contained blanks; this indicates that an operator accidentally put cards that were not blank into the punch hopper. (Output cards must not be "pre-punched" unless the 1401 is equipped with a special feature.)

At the end of the job, it is customary to clear the punch area to blanks and to instruct the 1401 to punch a card. No holes will be punched into this last card, of course, since there are only blanks in the punch area, but this "dummy punch" operation will cause the last real output data card to be checked and stacked.

## OPERATION - SELECT STACKER

OP CODE - SS

### INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X			X

### PURPOSE OF INSTRUCTION

To select a stacker for a card that has just been read or just been punched. Either stacker 1 or stacker 8/2 may be selected for a card that has just been read. (If neither one is selected, the card will fall into the NR stacker.) Either stacker 4 or stacker 8/2 may be selected for a card that has just been punched. (If neither one is selected, the card will fall into the NP stacker.)

The desired stacker is specified by the d-character of the instruction.

To select stacker:	1	8/2	4	8/2
for a card that has just been:	read	read	punched	punched
use the d-character:	1	2	4	8

The "Select stacker" instruction **is ordinarily given** right after the "read a card" or "punch a card" operation. The instruction is effective only for one card; the next time reading or punching is done, the stacking mechanism returns to its "normal" (NR or NP) setting. Thus, the 1401 must act on a "Select stacker" instruction after each "read a card" operation and after each "punch a card" operation, if stackers are to be selected for all input and output cards.

OPERATION - WRITE A LINE

OP CODE - W

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X			

#### PURPOSE OF INSTRUCTION

To print a line of data. The data to be printed comes from storage positions 0201-0332, if the 1403 Printer has 132 print positions; or from storage positions 0201-0300 if the 1403 has 100 print positions. Word marks are not printed.

After the data has been printed, the paper is automatically moved up to the next line on the form (single spaced), unless a "Control carriage" instruction that was given prior to the "Write a line" instruction specified double or triple spacing, or skipping, after printing.



OPERATION - CONTROL CARRIAGE

OP CODE - CC

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X			X

PURPOSE OF INSTRUCTION

To space or to skip the paper forms in the 1403 Printer. Thirty different spacing or skipping operations are possible. The d-character of the instruction indicates exactly which operation is desired.

The thirty d-characters are listed in the chart below:

		d - CHARACTER	
		Forms movement is to occur:	
		Immediately	After next "Write a line" Operation
Space:	1 line	J	/
	2 lines	K	S
	3 lines	L	T
Skip to:	Channel 1	1	A
	Channel 2	2	B
	Channel 3	3	C
	Channel 4	4	D
	Channel 5	5	E
	Channel 6	6	F
	Channel 7	7	G
	Channel 8	8	H
	Channel 9	9	I
	Channel 10	0	+
	Channel 11	#	·
	Channel 12	@	⌘

OPERATIONS - COMBINATION INPUT/OUTPUT

OP CODES - (See table below.)

INSTRUCTION PARTS	OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
	X			

PURPOSE OF INSTRUCTION

To perform two or three of the operations of printing, reading, and punching, at the same time. There are four combinations, and their operation codes are listed below.

Combination Operation	Operation Code
Write a line, Read a card, and Punch a card	WRP
Write a line, and Read a card	WR
Write a line, and Punch a card	WP
Read a card, and Punch a card	RP

Combination operations take less time than the same operations done separately; for instance, the combination "RP" causes a card to be read while another card is punched, in less time than two separate "R" and "P" operations. In this instance, both operations are done in the time that is otherwise required just to punch a card!

Carriage control may be done for "WRP", "WR" and "WP" instructions, in the same way as for a separate "W" instruction.

Stacker selection may be done after "WR" and "WP" instructions, in the same way as after separate "R" and "P" instructions. In the combination instructions that involve both reading and punching ("WRP" and "RP"), stacker selection is possible only for the card that has been punched. (During these operations, punching continues for more than 10 milliseconds after reading is finished, so the 1401 cannot act on a read stacker selection instruction within the 10-millisecond time limit.)

OPERATIONS - INPUT /OUTPUT AND BRANCH

OP CODES - (See list below.)

INSTRUCTION PARTS

OP CODE	(A) OPERAND ADDRESS	(B) OPERAND ADDRESS	d CHARACTER
X	X		X (with CC and SS op codes only)

PURPOSE OF INSTRUCTION

To perform an input/output operation, and then to branch unconditionally to the instruction whose address is specified by the (A) operand.

A branch address may be specified for any of the operations listed below.

OPERATION

OPERATION CODE

Control carriage	CC
Punch a card	P
Read a card	R
Read a card, and Punch a card	RP
Select stacker	SS
Write a line	W
Write a line, and Punch a card	WP
Write a line, and Read a card	WR
Write a line, Read a card, and Punch a card	WRP

## INDEX OF OPERATION CODES

A	Add (2 fields) . . . . .	page 26
A	Add (1 field) . . . . .	27
B	Branch (unconditional). . . . .	5
B	Branch if character equal . . . . .	6
B	Branch if indicator on . . . . .	9
BWZ	Branch if word mark and/or zone . . . . .	7
C	Compare. . . . .	8
CC	Control carriage . . . . .	37
CC	Control carriage and branch . . . . .	39
CS	Clear storage . . . . .	18
CS	Clear storage and branch . . . . .	19
CW	Clear word marks (2 positions) . . . . .	16
CW	Clear word mark (1 position) . . . . .	17
D	Divide . . . . .	31
H	Halt . . . . .	10
H	Halt and branch . . . . .	11
LCA	Load characters to A-field word mark . . . . .	23
M	Multiply . . . . .	30
MCE	Move characters and edit . . . . .	22
MCS	Move characters and suppress zeros . . . . .	21
MCW	Move characters to word mark . . . . .	20
MN	Move numerical . . . . .	12
MZ	Move zone . . . . .	13
P	Punch a card . . . . .	34
P	Punch a card and branch . . . . .	39
R	Read a card . . . . .	33
R	Read a card and branch . . . . .	39
RP	Read and punch . . . . .	38
RP	Read, punch, and branch . . . . .	39
S	Subtract (2 fields). . . . .	28
S	Subtract (1 field) . . . . .	29
SS	Select stacker . . . . .	35
SS	Select stacker and branch . . . . .	39
SW	Set word marks (2 positions) . . . . .	14
SW	Set word mark (1 position). . . . .	15
W	Write a line . . . . .	36
W	Write a line and branch . . . . .	39
WP	Write and punch . . . . .	38
WP	Write, punch, and branch . . . . .	39
WR	Write and read . . . . .	38
WR	Write, read, and branch . . . . .	39
WRP	Write, read, and punch . . . . .	38
WRP	Write, read, punch, and branch . . . . .	39
ZA	Zero and add . . . . .	24
ZS	Zero and subtract . . . . .	25
* DC	<i>Define character</i>	
* DCW	<i>Define " " word mark</i>	
DS	<i>Define Symbols</i>	
CTL	<i>Control</i>	

**Supplies**



INTERNATIONAL BUSINESS MACHINES CORPORATION  
**IBM 1401 SYMBOLIC PROGRAMMING SYSTEM**  
 CODING SHEET

FORM X24-1152-3  
 PRINTED IN U.S.A.

Program \_\_\_\_\_

Page No. 1 of 2

Programmed by \_\_\_\_\_

Date \_\_\_\_\_

Identification 05W  
76 80

LINE	COUNT	LABEL	OPERATION	(A) OPERAND				(B) OPERAND				d	COMMENTS
				ADDRESS	±	CHAR. ADJ.	IND.	ADDRESS	±	CHAR. ADJ.	IND.		
3	5 6 7 8	13 14 16 17		23			27	28	34		38	39 40	55
0 1 0	1 1	ED.1	D.6W*				-	-					
0 2 0	1 0	ED.NET	D.6W*			\$	)	0.					
0 3 0	1 9	ED.NAME	D.6W*				.s	.s					
0 4 0	1 3	ED.AMT	D.6W*			*	,	φ.					
0 5 0													
0 6 0	1 9	ED.EMP	D.6W*				.s	.s					
0 7 0	0 7	ED.C&TD	D.6W*			\$	φ	φ.					
0 8 0	1 0	ED.YTD	D.6W*			\$	)	φ.					
0 9 0	1 0	ED.SUM	D.6W*			\$	φ	.s					
1 0 0	0 8	ED.RATE	D.6W*			φ	/	/					
1 1 0													
1 2 0	2 0	ED.WD.1	D.6W*				.s	.s					
1 3 0	0 9	ED.WD.2	D.6W*			φ	,	φ.					
1 4 0	1 2	ED.WD.3	D.6W*			φ	,	φ.					
1 5 0	0 8	ED.WD.4	D.6W*			φ	/	/					
1 6 0													
1 7 0													
1 8 0													
1 9 0													
2 0 0													



INTERNATIONAL BUSINESS MACHINES CORPORATION  
**IBM 1401 SYMBOLIC PROGRAMMING SYSTEM**  
**CODING SHEET**

Program \_\_\_\_\_  
 Programmed by \_\_\_\_\_

Date \_\_\_\_\_

Page No. 

--	--

 of \_\_\_\_\_  
 Identification 

--	--	--	--	--	--

LINE	COUNT	LABEL	OPERATION	(A) OPERAND				(B) OPERAND				d	COMMENTS
				ADDRESS	±	CHAR. ADJ.	IND.	ADDRESS	±	CHAR. ADJ.	IND.		
3	5 6 7 8	13 14 16 17	23	27	28	34	38	39	40	55			
0	1	0											
0	2	0											
0	3	0											
0	4	0											
0	5	0											
0	6	0											
0	7	0											
0	8	0											
0	9	0											
1	0	0											
1	1	0											
1	2	0											
1	3	0											
1	4	0											
1	5	0											
1	6	0											
1	7	0											
1	8	0											
1	9	0											
2	0	0											

AREA-DEFINITION CHARACTER COUNT → 1      5      10      15      20      25      30      32



INTERNATIONAL BUSINESS MACHINES CORPORATION  
**IBM 1401 SYMBOLIC PROGRAMMING SYSTEM**  
 CODING SHEET

Program \_\_\_\_\_

Page No. 1 of 2

Programmed by \_\_\_\_\_

Date \_\_\_\_\_

Identification 76 80

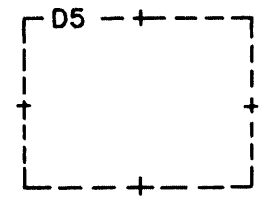
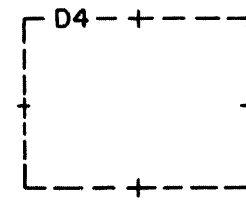
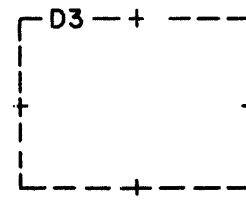
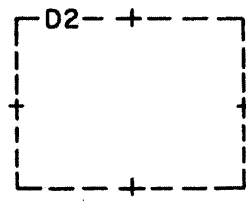
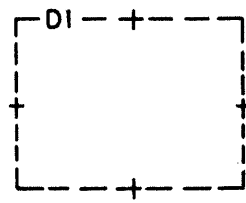
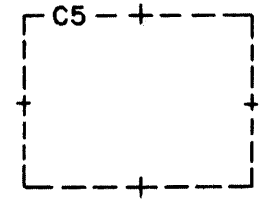
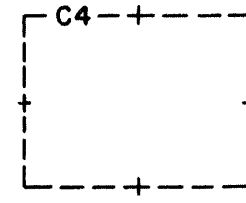
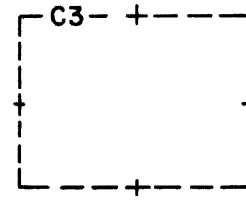
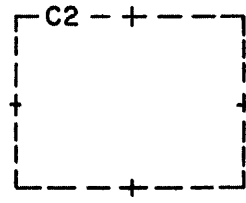
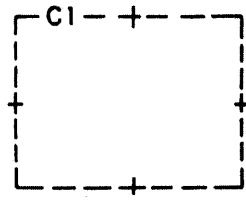
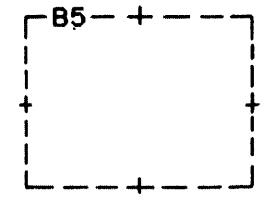
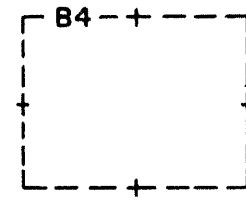
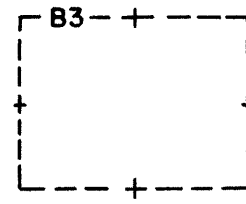
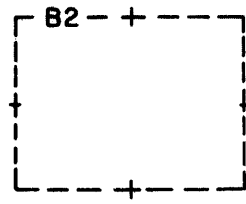
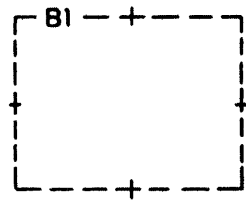
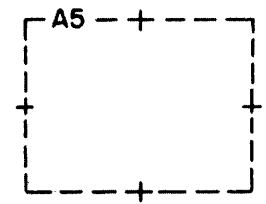
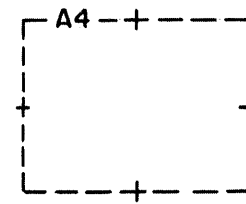
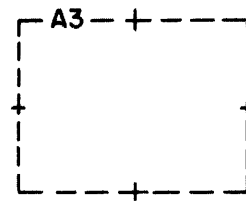
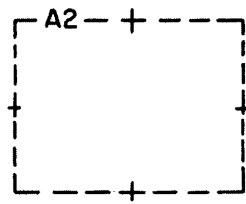
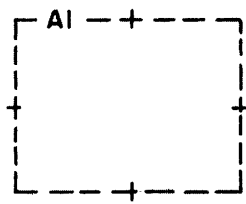
LINE	COUNT	LABEL	OPERATION	(A) OPERAND				(B) OPERAND				d	COMMENTS
				ADDRESS	±	CHAR. ADJ.	IND.	ADDRESS	±	CHAR. ADJ.	IND.		
3	5	6 7 8	13 14 16 17	23		27	28	34		38	39	40	55
0	1	0											
0	2	0											
0	3	0											
0	4	0											
0	5	0											
0	6	0											
0	7	0											
0	8	0											
0	9	0											
1	0	0											
1	1	0											
1	2	0											
1	3	0											
1	4	0											
1	5	0											
1	6	0											
1	7	0											
1	8	0											
1	9	0											
2	0	0											



# IBM Flowcharting Worksheet

PRINTED IN U.S.A.  
X20-8021-1

Programmer: _____	Program No.: _____	Date: _____	
Chart ID: _____	Chart Name: _____	Program Name: _____	Page: _____







# IBM PRINTER SPACING CHART

6 Lines Per Inch

The chart consists of a grid of small squares, approximately 30 squares wide and 100 squares high. The grid is divided into columns labeled 5, 6, 7, 8, 9, 10, 11, 12, and 13. The labels 5, 6, 7, 8, 9, 10, 11, 12, and 13 are positioned above the first row of squares in their respective columns. A vertical dashed line runs between the 9th and 10th columns. The grid is used to measure the spacing of printer output.

5 6 7 8 9 10 11 12 13

56789012345678901234567890123456789012345678901234567890123456789012

NOTE: This form is subject to inaccuracies from variations in humidity. Dimensions of form should be calculated from measurements shown and not scaled from this chart.

620805MSP

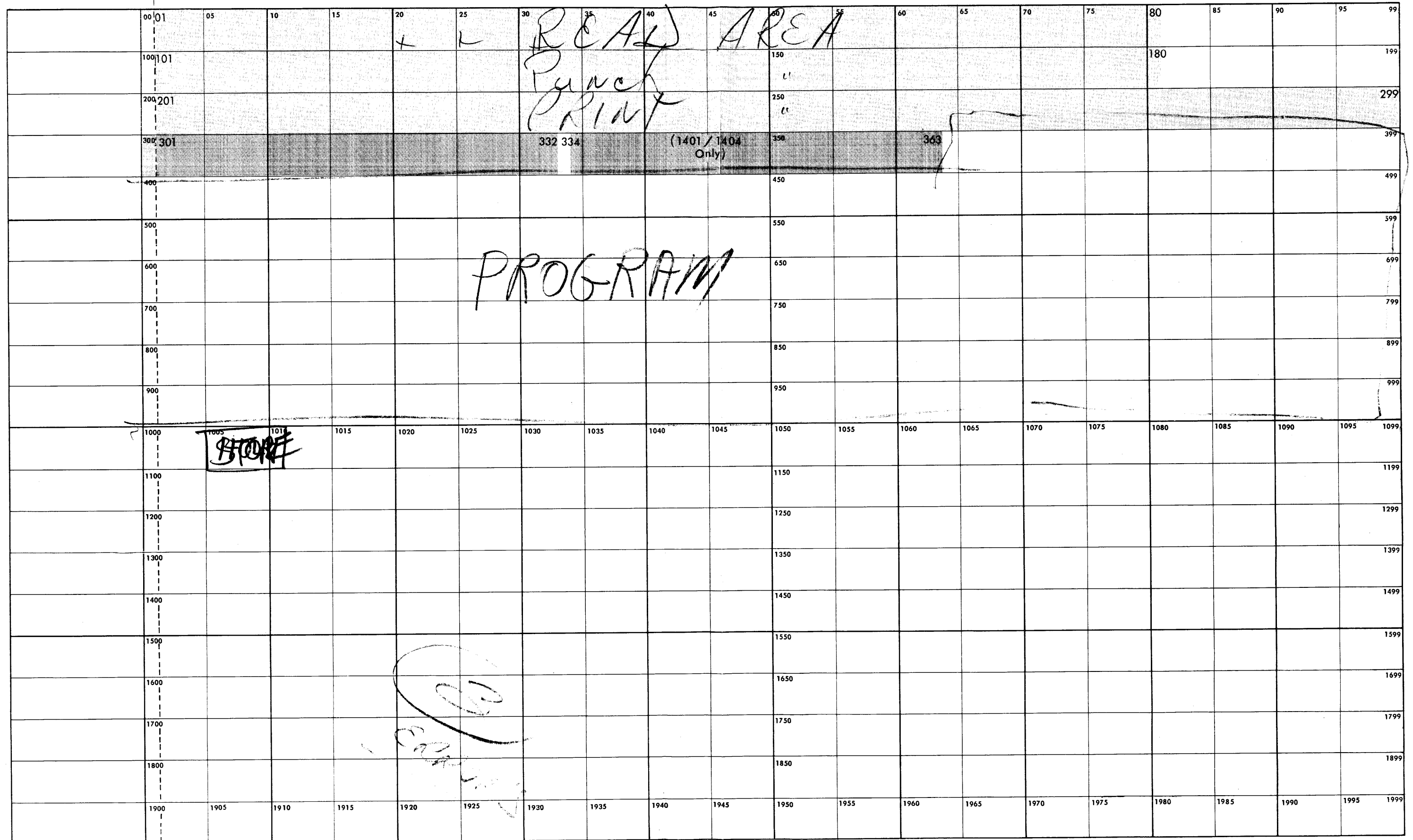
56789012345678901234567890123456789012345678901234567890123456789012

STORAGE SCHEMATIC—2000 POSITIONS

APPLICATION \_\_\_\_\_

DATE \_\_\_\_\_

(Shaded portion indicates IBM 1401 input-output areas)



# STORAGE LAYOUT

APPLICATION \_\_\_\_\_

IBM 1240 - 1401 - 1410 - 1420 - 1440 - 1460

DATE \_\_\_\_\_

	SYMBOLIC	
	DATA	
	LOCATION	0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95
	WORD MARK	
	SYMBOLIC	
	DATA	
	LOCATION	0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95
	WORD MARK	
	SYMBOLIC	
	DATA	
	LOCATION	0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95
	WORD MARK	
	SYMBOLIC	
	DATA	
	LOCATION	0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95
	WORD MARK	
	SYMBOLIC	
	DATA	
	LOCATION	0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95
	WORD MARK	
	SYMBOLIC	
	DATA	
	LOCATION	0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95
	WORD MARK	
	SYMBOLIC	
	DATA	
	LOCATION	0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95
	WORD MARK	
	SYMBOLIC	
	DATA	
	LOCATION	0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95
	WORD MARK	
	SYMBOLIC	
	DATA	
	LOCATION	0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90 95
	WORD MARK	

1401 BCD CHARACTER CODE		DIGIT BITS															
		NO BITS	1	2	21	4	41	42	421	8	81	82	821	84	841	842	8421
ZONE BITS	B AND A	A	A	B	C	D	E	F	G	H	I	plus zero	-	-	-	-	
	B ONLY	-	J	K	L	M	N	O	P	Q	R	minus zero	\$	*	-	-	
	A ONLY	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	NO BITS	-	1	2	3	4	5	6	7	8	9	zero	0	(0)	-	-	

Characters that are shaded print as blank spaces on a standard 1403

IBM CARD CHARACTER CODE		DIGIT PUNCH													
		NO	0	1	2	3	4	5	6	7	8	9	3-8	4-8	
ZONE PUNCH	12	&	plus zero	A	B	C	D	E	F	G	H	I	-	II	
	11	-	minus zero	J	K	L	M	N	O	P	Q	R	\$	*	
	0	-	-	/	S	T	U	V	W	X	Y	Z	,	%	
	NO	blank	zero	0	1	2	3	4	5	6	7	8	9	#	(0)

### 1401 STORAGE

SIZE	NUMBER OF POSITIONS	RANGE OF ADDRESSES
1	1,400 (1.4K)	0000 - 1399
2	2,000 (2K)	0000 - 1999
3	4,000 (4K)	0000 - 3999
4	8,000 (8K)	0000 - 7999
5	12,000 (12K)	00000 - 11999
6	16,000 (16K)	00000 - 15999

Printed in U.S.A.

R29-0044 2

**IBM**

**International Business Machines Corporation**  
**Data Processing Division**  
**112 East Post Road, White Plains, N.Y. 10601**  
**[USA Only]**

**IBM World Trade Corporation**  
**821 United Nations Plaza, New York, New York 10017**  
**[International]**