

Copy #44

SPACE OPERATOR'S MANUAL

BY

F. O. UNDERWOOD
DEPARTMENT 280

F. O. Underwood

JUNE 4, 1958

COMPANY CONFIDENTIAL

This document contains information of a proprietary nature. ALL INFORMATION CONTAINED HEREIN SHALL BE KEPT IN CONFIDENCE. No information shall be divulged to persons other than IBM employees authorized by the nature of their duties to receive such information or individuals or organizations who are authorized by Data Processing Division, Product Development management in accordance with existing policy regarding the release of company information.

F. O. Underwood

June 4, 1958

"STORED PROGRAM ACCOUNTING AND CALCULATING EQUIPMENT"
(SPACE)

MANUAL OF OPERATION

The SPACE machine performs a task by following a series of "instructions." These "instructions" must first be determined in detail by the operator, key punched into instruction cards, and then given to the machine. The machine will accumulate all of the instructions for a problem into its memory before the actual task is given to it.

When the task is given to the machine, it will process the data in just the way that the operator has told (instructed, programmed) the machine to do.

The SPACE machine, in its basic form, has the ability to perform approximately 30 basic functions, such as "feed a card", "print a line", "add", "subtract", etc. Any of these functions, or operations, in combination with information relative to the location of data upon which the operation is to be performed, constitutes an "instruction."

The amount of information contained in an instruction can vary, that is, when a "Feed card" instruction is given, it is not required that any data locations be specified, and if an "Add" instruction is given, then two data locations must be specified. This gives rise to the concept known as "variable-length instructions."

The SPACE machine must be instructed in its own language, which is simply a kind of "shorthand" notion. The "operation" part of an instruction is coded by the use of the 48 standard alphanumeric and special symbols, and the data locations in memory are specified by a three-digit "address."

The present operation code structure is mnemonic to as great an extent as possible. The operations and their related "operation codes" are shown in the following chart:

SPACE OPERATION CODES

1	FEED	B	CARRIAGE SKIP BEFORE
2	PRINT		PRINT
3	FEED AND PRINT	A	CARRIAGE SKIP AFTER
4	PUNCH		PRINT
5	FEED AND PUNCH	P	CARRIAGE SPACE BEFORE
6	PRINT AND PUNCH		PRINT
7	FEED, PRINT AND PUNCH	Q	CARRIAGE SPACE AFTER
8	FEED RELEASE (OPT)		PRINT
9	PUNCH RELEASE (OPT)		
+	ADD	\$	FLOAT DOLLAR SIGN (OPT)
R	RESET ADD	*	FILL ASTERISKS (OPT)
-	SUBTRACT	F	TEST CHARACTER
S	RESET SUBTRACT	G	TEST COMPARISON
T	TRANSFER	H	TEST TAPE CHANNEL
Z	TRANSFER AND ZERO SUPPRESS	W	TEST SWITCH
D	TRANSFER DIGIT	M	TEST DIGIT
X	TRANSFER ZONE	Y	TEST ZONE
E	EDIT	/	CLEAR
C	COMPARE	L	LOAD
N	NO OPERATION		
J	JUMP (PROGRAM SKIP)		
.	STOP		
,	WORD MARK SET		
□	WORD MARK CLEAR		

STORAGE

Before describing the writing of instructions in detail, the storage, or memory, arrangement of the SPACE machine will be described.

It has been determined that a memory capacity of 1400 alphanumeric characters provides sufficient power to match the functional ability of a standard 407, and in fact surpass it for many applications.

Three areas of this memory are reserved for input and output data; 80 character positions receive the 80 columns of card information from the 800 CPM card reader, another 80 character positions are reserved for assembly of summary punch data, and another 120 character positions are reserved for assembly of printer information.

Each of the 1400 character positions has a three-digit "address." The first thousand positions of the memory have the addresses 000-999. The addresses of the remaining 400 positions of memory require the use of an alphabetic or special character in the hundreds position of the address. This will be discussed later.

The coding of characters in memory requires the use of six information bits and one "check" bit per character. Each of the bits has a value, or weight;

1-2-4-8-A-B-C

The first four bits are employed in the coding of the numeric portion of a character, and the A and B bits are used to code the zone part of the character. The "C" bit is the "check" bit. Each bit position is two-valued; that is, a "yes" or "no" condition can be indicated for each of the bits.

To illustrate, the alphabetic character "S", coded in the card as a numeric "2" and a "zero" zone would be coded in storage as follows:

1	2	4	8	A	B	C
No	Yes	No	No	Yes	No	Yes

or simply as a 2-A-C. The "A" bit corresponds to a "zero" zone, the "B" bit corresponds to an "11" zone, and if both the "A" and "B" bits are "Yes" a "12" zone is indicated. The value of the "C" bit is automatically chosen by the machine's checking circuitry so that the total number of "Yes" conditions will always be ODD. As another illustration, the letter E (12-5) would be coded in storage as 1-4-A-B-C.

It is possible to display the coded information (one character at a time) on the console of the SPACE machine by a group of lights, a light being "ON" for each bit which is "YES".

The "Character Code Display" illustrates the code. If any displayed character contains a Word Mark, the "C" bit will be just opposite to the condition shown.

A numeric "zero" will be coded as an 2-8-C. It is then possible to place a "zone-zero" over a "numeric-zero" in storage, to obtain the code combination 2-8-A. Since this is possible, addresses above the first thousand are indicated by placing a "zone-zero" (setting the A-bit to "YES") over the numeric part of the hundreds digit of the address. Therefore the addresses of the eleventh one-hundred positions of storage have the code 2-8-A in the hundreds position of the address.

This particular combination has, at present, no symbol. For the SPACE machine, a new symbol is being provided for this code combination, the symbol being a "plus" sign (+). With the present key punch equipment, this code is punched in a card with an "S", with an "8" punch added in the same column.

It can be seen, then, that the 1400 character positions of memory have the following addresses:

1st hundred positions	000-099
2nd hundred positions	100-199
3rd hundred positions	200-299
--	
--	
--	
9th hundred positions	800-899
10th hundred positions	900-999
11th hundred positions	00-099
12th hundred positions	100-199
13th hundred positions	S00-S99
14th hundred positions	T00-T99

As previously mentioned, the input-output devices are assigned fixed locations in memory. The assignments are such that a correlation is achieved between input-output columns and memory addresses.

The card input area is assigned to memory locations 001 through 080, inclusive, corresponding to card columns 1-80.

The Punch output area is assigned to memory locations 101 through 180 inclusive, corresponding to punch columns 1-80.

The Printer area is assigned to memory locations 201 through 320, inclusive, corresponding to Print positions 1-120.

The remainder of storage may be used as desired, as working storage, "constant" storage, accumulation of totals, or as instruction storage.

One of the most important characteristics of the SPACE machine is the "variable-length" word storage. This means that storage is not divided into increments larger than a single character and that fields of any size are easily handled, and may be placed anywhere in storage. The location, or address, of any data word or field in storage is given as the location of the units, or rightmost character of the field.

When it is necessary to define the length of the data word (it will be seen later that it is not always necessary to do so) the location of the high-order (or leftmost) character is "marked". This "mark" is called a "word-mark" and is provided by an eighth bit in the character code.

To illustrate, assume that a nine-digit total is stored in memory, in locations 087 through 095, inclusive. The address of the word is 095 (location of units position) and position 087 will have the 8th, or word mark bit set to "YES".

The word marks are installed, where necessary, at the time the machine is given its instructions, and provision is made for altering the word marks during the execution of the program. More will be said later about word marks.

It should be kept in mind that data is normally operated upon in a right-to-left direction, as is necessary when performing arithmetic operations.

Instructions, however, are read from storage in a left-to-right direction (opposite to data), so that the address of an instruction will be the address of the left-most character. This left-most character will always be the Operation code, and will always contain a word mark (8th bit set to "YES"), which is automatically set when instruction cards are loaded into the machine.

THE INSTRUCTION FORMAT

The most common form of instruction consists of an Operation code, followed by two three-digit addresses. A two-address instruction is required to move data from one storage location to another, to perform the arithmetic operations of addition and subtraction, to compare two fields (Group Control) and to perform the editing function (to be described later).

Such an instruction would appear as:

⌘ 072 423

This is an "Add" instruction, as indicated by the operation code "+", and would cause the word whose units digit is in location 072 to be added algebraically to the word whose units digit is in location 423. The result would appear in location 423. It is common to refer to this operation as:

$A + B = B'$

Because of this, the two addresses in the instruction are referred to as the "A" and "B" addresses.

The one instruction causes the entire "A" word to add to the entire "B" word, digit-by-digit. The detection of the word mark in the high-order position of the B field will indicate the end of the operation and cause the machine to advance to the next instruction.

Not all instructions are of the "two-address" form. Some instructions may have only one address, and others have none at all.

There are three classes of instructions for the SPACE machine:

- a) Machine control
- b) Transfers, arithmetic, editing
- c) Test, or "branching" instructions.

MACHINE CONTROL

This first class of instructions involves those operations which cause card feeding, punching, printing, and carriage operation. These operations do not require data addresses to be included in the instruction. Therefore, an instruction of this class usually consists of a single character which is the operation code.

Included in this class are the operations 1 through 7. We may, if we wish, include the "NO OP" code (N) and the "Stop" code (.) in this class.

The carriage instructions are also included, but these require at least two characters per instruction; one for the operation code, and another character which specifies the "SKIP TO" channel or the number of line spaces to be taken.

After any of the instructions of this class have been executed, the machine will go to the next instruction in location sequence. If, however, it is desired to "Jump" to some other instruction, the address of that instruction may be written in the "machine control" instruction immediately following the "operation code" character.

Note that in the case of a carriage control instruction, the digit specifying the channel or number of line spaces is always the last character in the instruction.

TRANSFERS, ARITHMETIC, EDITING, AND COMPARE

The second class of instructions are the Transfers, Arithmetic, Editing, and Compare instructions which always consist of an operation code plus two data addresses. The operations in this class are the following:

- (T) TRANSFER
- (+) ADD
- (-) SUBTRACT
- (R) RESET ADD
- (S) RESET SUBTRACT
- (Z) TRANSFER AND ZERO SUPPRESS
- (C) FIELD COMPARE
- (E) EDIT
- (L) LOAD
- (D) TRANSFER DIGIT
- (X) TRANSFER ZONE

All but the last two operations operate on complete words, word marks being required in at least one of the data fields being operated on. The last two operations are "single digit" operations, and no word marks are required in the data fields.

In order to understand this class of instructions clearly, the rules for word marks must first be understood. They are as follows:

- 1) Word marks are not transferred with data, except when the "LOAD" instruction is used.
- 2) When transferring data from one location to another, only one of the fields need have the defining word mark, since the "Transfer" instruction implies that both fields are of the same length.

The same is true of the "Compare" instruction.

- 3) For an arithmetic operation, the "B" field must have a defining word mark, and the "A" field must have a word mark only when it is shorter than the "B" field.
- 4) The "LOAD" instruction is the same as "Transfer" except that it will transfer the word mark of the A field to the B field, and will cause any other word marks in the B field to be cleared.
- 5) Predetermined word marks that normally are expected to remain in that location throughout the run are set into storage, where required, during the loading operation (to be described later).
- 6) Two operation codes are provided to permit the setting and clearing of word marks during program execution. These are:

Word mark set (,)
Word mark clear (\square)

Any number of addresses may be written with either of these operation codes, permitting any number of word marks to be set or cleared, with one instruction, at the specified addresses.

Since the machine processes data in a "serial-by-digit" manner, the word marks serve both to indicate the end of a word, and the end of the execution of an instruction. When the machine calls a new instruction from memory, the detection of the word mark of the next adjacent instruction indicates that the complete instruction has been read, causing the machine to go into an execution cycle.

BRANCHING

The third class of instructions are the "branching" instructions which cause various tests to be made, and, as a result of the test, permits a choice to be made as to which instruction will be executed next.

A test on a character in memory requires an eight-character instruction; one for the operation code, three for the "conditional" (branching) instruction address, three for the data address, and one to specify the particular character being tested for.

A test of a Console switch or Tape channel requires a five-character instruction; one for the operation code, three for the "conditional" instruction address, and one to specify the Tape channel or Console switch is to be tested.

Since a "Compare" instruction does not include provision for branching, it must be followed by a "Test Compare" instruction, which consists of the operation code and a three-digit "conditional" instruction address.

The following Operation Code description explains each of the operations in detail. The symbols (I), (A), (B) indicate three-digit addresses included in an instruction. "d" indicates a single-digit character.

- 1 FEED
- 1 (d) FEED AND STACKER SELECT
- 1 (I) FEED AND PROGRAM SKIP
- 1 (I) (d) FEED, STACKER SELECT AND PROGRAM SKIP

This instruction causes the program to stop, activate the card feed, and then resume with the next instruction in sequence after a new card has been read. The card read on the previous cycle will enter the first (non-select) stacker.

If a three-digit Instruction address follows the operation code, the next instruction will be taken from that address, effecting a "Program Skip", after card reading time.

Pocket selection for the card read on the previous cycle is specified by including, as the last character of the instruction, a digit specifying the selected pocket. A digit "1" will cause the card to enter the first select pocket, and a digit "2" will cause the card to enter the second select (or "merge") pocket.

2 PRINT
2 (I) PRINT AND PROGRAM SKIP

This instruction causes the program to stop, and the print area to be scanned to the printer. The program will resume immediately after printing is complete, with the next instruction in sequence unless the operation code is followed by an address. In this case, the next instruction will be taken from that address so indicated.

3 PRINT AND FEED
3 (d) PRINT, FEED AND STACKER SELECT
3 (I) PRINT AND PROGRAM SKIP
3 (I) (d) PRINT, FEED, STACKER SELECT AND PROGRAM SKIP

This instruction combines the operations of FEED (1) and PRINT (2). Since the basic machine cannot simultaneously feed and print, the Printer is given priority and operates first.

4 PUNCH
4 (d) PUNCH AND STACKER SELECT
4 (I) PUNCH AND PROGRAM SKIP
4 (I) (d) PUNCH, STACKER SELECT AND PROGRAM SKIP

This instruction is similar to FEED (1) but causes the Punch to operate, and the P_unch area of memory to be scanned to the punch magnets. The program will stop until after punch time. The next instruction will then be the next in sequence, unless an address follows the operation code, in which case the next instruction will be taken from the specified address.

Pocket selection for the card previously read at the Punch brush station is effected by including a single character (d) as the last character of the instruction; "4" for the first select pocket, and "8" for the second select (or "merge") pocket. If no digit (d) is included, then the card will enter the first (non-select) pocket.

- 5 FEED AND PUNCH
- 5 (d) FEED, PUNCH AND STACKER SELECT
- 5 (I) FEED, PUNCH AND PROGRAM SKIP
- 5 (I) (d) FEED, PUNCH, STACKER SELECT AND PROGRAM SKIP

This instruction combines the FEED (1) and PUNCH (4) operations, and the two processes will overlap.

In order to select Stack for both feeds, the (d) digit will be the sum of the pocket numbers, that is, if the card from the Punch feed is to enter the merge pocket and card from the Read feed is to enter the first select pocket, the (d) digit is 9 ($8 + 1 = 9$).

If both cards are to enter the merge pocket, the (d) digit is punched "ZERO" ($8 + 2 = 0$).

- 6 PRINT AND PUNCH
- 6 (d) PRINT, PUNCH AND STACKER SELECT
- 6 (I) PRINT, PUNCH AND PROGRAM SKIP
- 6 (I) (d) PRINT, PUNCH, STACKER SELECT AND PROGRAM SKIP

This instruction combines the functions of PRINT (2) and PUNCH (4). Since these processes cannot overlap, the Printer takes priority.

- 7 PRINT, FEED, PUNCH
- 7 (d) PRINT, FEED, PUNCH, STACKER SELECT
- 7 (I) PRINT, FEED, PUNCH, PROGRAM SKIP
- 7 (I) (d) PRINT, FEED, PUNCH, STACKER SELECT, PROGRAM SKIP

This instruction combines the functions of FEED (1), PRINT (2), and PUNCH (4).

Again, the Printer has priority and the Feed and Punch processes are overlapped.

- B (d) CARRIAGE SKIP TO (d) BEFORE PRINT
- B (I) (d) CARRIAGE SKIP TO (d) BEFORE PRINT, AND PROGRAM SKIP

This instruction causes the carriage to start a skip to the channel specified by the digit (d). If the carriage is in motion because of some previous instruction to the carriage, the program will stop until the carriage comes to rest. At this point, the new carriage action will be initiated, and the program will advance to the next instruction in memory. If, however, an address is included in this instruction, the next instruction will be taken from the specified address.

A (d) CARRIAGE SKIP TO (d) AFTER PRINT
A (I) (d) CARRIAGE SKIP TO (d) AFTER PRINT, AND PROGRAM SKIP

This instruction will initiate a skip to (d) immediately after the next print instruction has been given and executed. The program will not halt when this instruction is given, but will proceed to the next in sequence, or will go to the instruction located at address (I) if included.

P (d) SPACE BEFORE PRINT
P (I) (d) SPACE BEFORE PRINT AND PROGRAM SKIP

This instruction causes the carriage to start spacing a number of lines specified by the digit (d). A maximum of three line spaces are permitted.

In order to maintain the highest possible machine speed, this instruction should be given as early in the "Compute" cycle as possible.

If the carriage is moving because of some previous instruction to the carriage, the program will halt until the carriage stops, at which time the new carriage space operation will be initiated, and the program will advance. If an address is included in the instruction, the next instruction will be taken from that address.

Q (d) CARRIAGE SPACE AFTER PRINT
Q (I) (d) CARRIAGE SPACE AFTER PRINT AND PROGRAM SKIP

This instruction will initiate a number of line spaces, specified by the (d) character, immediately after the next "Print" instruction has been given and executed. Since a single line space is automatically taken after each print cycle, the (d) character need specify only 2 or 3 line spaces.

T (A) (B) TRANSFER

This instruction causes the data field at the (A) address to be stored at the (B) address. Since both fields must be of the same length, only one of the fields need have the defining word mark.

The word marks, themselves, are not affected by the transfer operation, nor is the data at the (A) address.

Upon the completion of this operation, the memory addressing registers contain the addresses of the fields immediately to the left of the original fields, so that, if the next instruction is one which instructs the machine to perform an operation on these fields, then this next instruction need not specify these addresses. Therefore, if a series of fields in, say, the card area are to be transferred to a series of locations in working storage, and word marks are used in storage to define the word lengths, the necessary instructions will appear as follows:

T 080 540 T T T T ---

+ (A) (B) ADD

This instruction causes the numeric data at the A address to add algebraically to the numeric data at the B address.

The B field requires a word mark, but the A field requires a word mark only if it is shorter than the B field.

The sign of a field is carried as a "B" bit in the units position of that field. The "B" bit is set to "NO" for a positive or zero amount, and set to "YES" for a negative amount, the data always being in true form.

If the sign of the B field should happen to change as a result of the arithmetic operation, the machine will take an automatic recomplementing cycle so as to store the data in true form.

Upon the completion of this operation, the memory addressing registers contain the addresses of the fields immediately to the left of the original fields, so that it is possible to add a series of A fields to a series of B fields without carrying addresses in successive instructions.

- (A) (B) SUBTRACT

This instruction is the same as the ADD (+) instruction, except that the A field is algebraically subtracted from the B field.

R (A) (B) RESET ADD

This instruction is similar to the ADD (+) instruction, except that the B field is, in effect, set to zero before the A field data is added to it. This is not the same as a "Transfer" (T) instruction.

S (A) (B) RESET SUBTRACT

This instruction is similar to the Reset Add (R) instruction, except the A field is subtracted (algebraically) from the B field (all zeros).

Z.(A) (B) TRANSFER AND ZERO SUPPRESS

This instruction is similar to the "Transfer" (T) instruction except for one important difference; upon completion of the operation, the B field will be found to contain blanks, instead of zeros, to the left of the most significant digit.

E (A) (B) EDIT

This instruction causes the numeric data at field (A) to be edited under control of the EDIT CONTROL WORD in field B, generating the edited data in the B field. Since this destroys the control word, it must be stored elsewhere in working storage and transferred to the B address prior to the EDIT instruction.

As with TRANSFER (T), the A and B addresses of the EDIT instruction must specify addresses of the low-order (rightmost) positions of the fields.

The EDIT CONTROL WORD is composed of certain symbols which control the insertion of commas, decimal points, blanks, conditional minus and CR symbols, and certain other characters.

The edit symbols are as shown below, and have the specified function:

ZERO SUPPRESS

If a data word is to be zero suppressed, in conjunction with some other editing function such as comma or blank insertion, etc., and/or if a dollar sign is to be floated or asterisks are to fill left, then the edit control word must contain this symbol in every position to be zero-suppressed.

@ CHARACTER POSITION

This symbol is written in the control word to define the character positions of the field, and will be replaced by characters of the data word.

& SIGN TEST

This symbol causes a test to be made for the sign of the data field, and is normally written to the right of the "CR" or "-" symbol in the control word, if these symbols are to be printed under the condition that the data field is negative.

- MINUS SIGN

This symbol will cause a minus sign to be printed, unconditionally.

If placed to the left of the & symbol, a minus sign will print only if the data word is negative. A blank will be inserted if the sign of the data word is positive.

CR CREDIT SYMBOLS

These two symbols will cause "CR" to print, unconditionally.

If placed to the left of the "&" symbol, the "CR" will print only if the data word is negative. Blanks will be inserted if the sign of the data word is positive.

. (period) DECIMAL POINT

This symbol, in the edit control word, will cause zero suppression to turn off, and .00 will print for zero balance fields.

, COMMA

This symbol will cause the printing of a comma only if the data word contains a significant digit to the left of this position.

BLANK

A blank space in the control word will cause a blank space to appear in the edited data. It does not eliminate any characters.

All other numeric, alphabetic and special characters, when appearing in an edit control word, will be printed.

Several examples of editing appear on the next page.

- 1) DATA 00120305
CONTROL WORD ###, ###. @@
EDITED DATA 1, 203. 05
- 2) DATA 00020305
CONTROL WORD ###, ###. @@
EDITED DATA 203. 05
- 3) DATA 00123456
CONTROL WORD ##### b @@ B CR & *
EDITED DATA 1234 56 CR *
..
- 4) DATA 01238
CONTROL WORD # @-@@-5@
EDITED DATA 1-23-58

D (A) (B) TRANSFER DIGIT

This instruction causes the numeric portion (1-2-4-8 bits) of the single character at the A address to be written at the B address. The zone information at both addresses is not affected.

Since this is a "single-character" operation, no word marks are required.

X (A) (B) TRANSFER ZONE

This instruction is similar to "Transfer Digit" (D) except that the zone portion (A-B bits) are transferred.

C (A) (B) COMPARE

This instruction causes the word at the A address to be compared with the word at the B address. Since both fields must be of the same length, only one of the fields need have the defining word mark.

The result of the test ("equal" or "unequal") is stored in the machine for later use by a "Test Compare" (G) instruction, to be explained.

F (I)(B) (d) TEST-CHARACTER

This instruction tests the contents of the B address for the presence of the (d) character. If the character (d) is not contained at B, then the next instruction to be executed will be the next adjacent instruction to the right in memory. If the character (d) is contained at B, then the next instruction to be executed will be that which is stored at the (I) address.

Therefore, this is a "JUMP ON 'YES'" instruction.

G (I) TEST COMPARE

This instruction will test the result of the last "Compare" instruction executed. If the result was equal, the next instruction will be the next in sequence, and if "unequal" the next instruction will be taken from the (I) address.

H (I) (d) TEST-CARRIAGE TAPE CHANNEL

This instruction tests the carriage tape for the presence of a punch in the tape channel specified by the digit (d). The basic machine only permits the testing for channel 9 (d = 9) and channel 12 (d = &).

When a channel 9 (or channel 12) punch is sensed by the carriage brushes, an indication of the fact is given to the machine, and this indication is retained until a punch in channel 1 is sensed.

When the channel test instruction is given, and if the machine indicates that the particular channel has been reached, the next instruction will be taken from the (I) address, otherwise the next in sequence.

W (I) (d) TEST-SWITCH

This instruction tests the setting of one of the switches, specified by the (d) digit, on the console.

The basic machine is provided with only one console switch, (#1) which serves as an "END OF FILE" switch. This switch is sensed as being "ON", only when it is turned to the "ON" position and the last card of a file, or deck, has passed the second read station.

When sensed as being "ON", the next instruction will be taken from the (I) address, otherwise the next in sequence.

M (I) (B) (d) TEST-DIGIT

This instruction is the same as TEST-CHARACTER (F) except that the zone portion (A-B bits) is ignored.

Y (I) (B) (d) TEST-ZONE

This instruction is the same as TEST-CHARACTER (F) except that the numeric portion (1-2-4-8 bits) is ignored.

L (A) (B) LOAD

This instruction is the same as TRANSFER (T) except that the length of the A word must be defined by a word mark. The word mark of the A field is transferred to the B field and all other word marks in the B field are cleared. This instruction is most commonly used to load instruction into memory.

J (I) JUMP (UNCONDITIONAL PROGRAM SKIP)

This instruction causes the next instruction to be taken from the (I) address, rather than the next in sequence.

N NO OPERATION

This operation code may be substituted for the operation code of any instruction to make that instruction ineffective.

. STOP
. (I) STOP, PROGRAM SKIP

This instruction will cause the program to stop, and the "STOP-PROCESS" light to turn on.

Depressing the "PROGRAM RESTART" key will cause the program to resume either from the next instruction in sequence, or from the (I) address, if included.

, (A) (A) (A)--- WORD MARK SET

This instruction may include any number of addresses, and will cause a word mark to be set at each of the specified addresses without disturbing data.

□ (A) (A) (A)--- WORD MARK CLEAR

Same as WORD MARK SET (,) except that word marks are cleared at the specified addresses.

/ (I) (B) CLEAR

This instruction is used to clear an area of storage (100 characters, maximum) of data and word marks.

It will cause the clearing in all positions XXX down through X00. That is, if the (B) address is, say, 563, then all positions, 563 through 500 will be cleared. An (I) address must be included, and the next instruction will be taken from that address.

8 FEED RELEASE (OPTIONAL)
8 (I) FEED RELEASE AND PROGRAM SKIP

This instruction permits the Card Feed clutch to engage at the next clutch time, and allows the program to continue until actual card reading time. This is useful when tabbing at an 800 CPM rate. It provides a calculate time of approximately 29 ms, which would otherwise only be about 9 ms.

9 PUNCH RELEASE (OPTIONAL)
9 (I) PUNCH RELEASE AND PROGRAM SKIP

This instruction causes the Punch clutch to engage at the next punch cycle, and permits the program to continue until actual punching time. This is useful when doing a 604-type operation at 250 CPM.

It permits a calculate time of about 50 milliseconds, which would otherwise be about .28 milliseconds.

\$ (A) DOLLAR SIGN FLOAT (OPTIONAL)

In order to float a dollar sign, the data word must first be zero-suppressed by a previous instruction. The DOLLAR SIGN FLOAT (\$) instruction may then be given, with the (A) address being the location of the units position of the word, or, as is more usual, the location of the decimal point.

The effect of this operation is to place a dollar sign one position to the left of the most significant digit in the word.

* (A) ASTERISK FILL (OPTIONAL)

This instruction will cause asterisks to fill all blank positions to the left of the most significant digit of a previously zero-suppressed word.

LOADING

The loading of the instructions into the memory of the SPACE machine is preceded by several steps;

- 1) Formulation of the problem.
- 2) Storage assignment for constants and totals.
- 3) Preparation of "Flow chart".
- 4) Writing the detailed instructions.
- 5) Key-punching the instruction cards.

The instructions may be written in several degrees of detail. It is a simple matter to determine instruction addresses for a short program, so that such a program can be written in complete detail.

However, for larger programs, which involve a fair amount of branching instructions and address modification, the program need not be written in complete detail. Instead, it is possible to utilize a previously written "assembly program" which treats the incomplete program as data, producing a new set of punched cards constituting the "complete" program.

In any event, the "complete" program must then be loaded into storage. This is done by placing the instruction cards in the feed and depressing the "LOAD" key on the Reader. The data cards may be placed in the hopper, on top of the instruction cards.

When the "LOAD" key is depressed, the cards will feed until the first card passes second reading, loading the input area with the data from the first card.

Normally, the instruction cards contain not only the instructions to be loaded, but also, instructions as to where the instructions are to be stored, what constants are to be stored, and where, and also instructions as to where word marks are to be placed.

The use of a "standard" loading card is recommended, such a card is shown in the accompanying illustration. It can be seen that the format of this card follows, quite closely, the format of the "Program Chart", and requires one card per instruction.

This card is designed to facilitate the automatic assembly routines, and contains enough information to permit the printing of a final "Program Chart" by the use of the SPACE machine, itself.

Columns 1-7 contain the LOAD instruction which loads the instruction contained in columns 23-30 to a specified place in memory.

Columns 8-12 contain the next instruction to be executed, namely, Feed a card, select this card into pocket #1, and execute the next instruction from address 001.

A deck of these cards must be preceded by a card which sets word marks in the input area, punched as shown in the accompanying "Storage Layout Chart."

The arrangement of the word mark addresses is such that the "Word mark set" instruction will appear as a single instruction, that is, word marks are not set into positions 008, 013, 023, 031, etc., until the instruction has been read from the corresponding addresses. The repeated "001" address is simply a "filler" so that everything works out all right.

Following the loading of instructions the word marks must be cleared with a card punched in columns 1-7 as follows:

/ (I) 080

Where (I) is the address of the first instruction of the program, and is normally a "Feed" instruction.

In any event, the first instruction must be located in working storage.

There are other ways of punching Load cards, permitting up to four instructions per card, but such methods are limited in flexibility.

SPACE - OPERATION CODE SUMMARY

1	FEED	+	ADD
2	PRINT	-	SUBTRACT
3	FEED-PRINT	R	RESET ADD
4	PUNCH	S	RESET SUBTRACT
5	FEED-PUNCH		
6	PRINT-PUNCH	C	COMPARE
7	FEED-PRINT-PUNCH	E	EDIT
8	FEED RELEASE (OPT)	J	JUMP
9	PUNCH RELEASE (OPT)	N	NO OPERATION
A	SKIP AFTER PRINT	/	CLEAR
B	SKIP BEFORE PRINT	,	WORD MARK SET
P	SPACE BEFORE PRINT	⌘	WORD MARK CLEAR
Q	SPACE AFTER PRINT	L	LOAD
T	TRANSFER	F	TEST CHARACTER
D	TRANSFER DIGIT	G	TEST COMPARE
X	TRANSFER ZONE	H	TEST CHANNEL
Z	TRANSFER, ZERO SUPPRESS	W	TEST SWITCH
		M	TEST DIGIT
		Y	TEST ZONE
\$	DOLLAR SIGN FLOAT (OPT)		
*	ASTERISK FILL (OPT)		

	A	B	C	D	E	F	G	H	I	&	.	□	
C	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	C
B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	B
A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	A
B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	B
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	4
2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1

	J	K	L	M	N	O	P	Q	R	-	\$	*	
C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	C
B	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	B
A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	A
B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	B
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	4
2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1

	/	S	T	U	V	W	X	Y	Z	+	,	%	
C	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	C
B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	B
A	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	A
B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	B
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	4
2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1

	1	2	3	4	5	6	7	8	9	0	#	@	
C	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	C
B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	B
A	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	A
B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	B
4	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	4
2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2
1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	1

CHARACTER CODE DISPLAY

2 JUNE '58 - Jau

STORAGE LAYOUT CHART

FIRST INSTRUCTION CARD - BASIC SPACE -	1001	0010	0505	1075	1078	1001	0080	1013	0230	3103	3035	0404	4304	6100	5100	11																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
	SET WORD MARKS															FEED																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																											
NORMAL INSTRUCTION CARD - BASIC SPACE -	LOAD	FEED	OPERATION	ADDRESS	INSTRUCTION NO	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)	(J)	(K)	(L)	(M)	(N)	(O)																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
	L(A)	(B)	1100	11																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
LAST INSTRUCTION CARD - BASIC SPACE -	CLEAR	(I)	0800																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																								
																		FEED																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																									
NORMAL INSTRUCTION CARD - EXPANDED SPACE -	LOAD	FEED	OP	(A)	(B)	(C)	(D)	(E)	(F)	(G)	(H)	(I)	(J)	(K)	(L)	(M)	(N)	(O)	(P)	(Q)	(R)	(S)	(T)	(U)	(V)	(W)	(X)	(Y)	(Z)	(AA)	(AB)	(AC)	(AD)	(AE)	(AF)	(AG)	(AH)	(AI)	(AJ)	(AK)	(AL)	(AM)	(AN)	(AO)	(AP)	(AQ)	(AR)	(AS)	(AT)	(AU)	(AV)	(AW)	(AX)	(AY)	(AZ)	(BA)	(BB)	(BC)	(BD)	(BE)	(BF)	(BG)	(BH)	(BI)	(BJ)	(BK)	(BL)	(BM)	(BN)	(BO)	(BP)	(BQ)	(BR)	(BS)	(BT)	(BU)	(BV)	(BW)	(BX)	(BY)	(BZ)	(CA)	(CB)	(CC)	(CD)	(CE)	(CF)	(CG)	(CH)	(CI)	(CJ)	(CK)	(CL)	(CM)	(CN)	(CO)	(CP)	(CQ)	(CR)	(CS)	(CT)	(CU)	(CV)	(CW)	(CX)	(CY)	(CZ)	(DA)	(DB)	(DC)	(DD)	(DE)	(DF)	(DG)	(DH)	(DI)	(DJ)	(DK)	(DL)	(DM)	(DN)	(DO)	(DP)	(DQ)	(DR)	(DS)	(DT)	(DU)	(DV)	(DW)	(DX)	(DY)	(DZ)	(EA)	(EB)	(EC)	(ED)	(EE)	(EF)	(EG)	(EH)	(EI)	(EJ)	(EK)	(EL)	(EM)	(EN)	(EO)	(EP)	(EQ)	(ER)	(ES)	(ET)	(EU)	(EV)	(EW)	(EX)	(EY)	(EZ)	(FA)	(FB)	(FC)	(FD)	(FE)	(FF)	(FG)	(FH)	(FI)	(FJ)	(FK)	(FL)	(FM)	(FN)	(FO)	(FP)	(FQ)	(FR)	(FS)	(FT)	(FU)	(FV)	(FW)	(FX)	(FY)	(FZ)	(GA)	(GB)	(GC)	(GD)	(GE)	(GF)	(GG)	(GH)	(GI)	(GJ)	(GK)	(GL)	(GM)	(GN)	(GO)	(GP)	(GQ)	(GR)	(GS)	(GT)	(GU)	(GV)	(GW)	(GX)	(GY)	(GZ)	(HA)	(HB)	(HC)	(HD)	(HE)	(HF)	(HG)	(HH)	(HI)	(HJ)	(HK)	(HL)	(HM)	(HN)	(HO)	(HP)	(HQ)	(HR)	(HS)	(HT)	(HU)	(HV)	(HW)	(HX)	(HY)	(HZ)	(IA)	(IB)	(IC)	(ID)	(IE)	(IF)	(IG)	(IH)	(II)	(IJ)	(IK)	(IL)	(IM)	(IN)	(IO)	(IP)	(IQ)	(IR)	(IS)	(IT)	(IU)	(IV)	(IW)	(IX)	(IY)	(IZ)	(JA)	(JB)	(JC)	(JD)	(JE)	(JF)	(JG)	(JH)	(JI)	(JJ)	(JK)	(JL)	(JM)	(JN)	(JO)	(JP)	(JQ)	(JR)	(JS)	(JT)	(JU)	(JV)	(JW)	(JX)	(JY)	(JZ)	(KA)	(KB)	(KC)	(KD)	(KE)	(KF)	(KG)	(KH)	(KI)	(KJ)	(KL)	(KM)	(KN)	(KO)	(KP)	(KQ)	(KR)	(KS)	(KT)	(KU)	(KV)	(KW)	(KX)	(KY)	(KZ)	(LA)	(LB)	(LC)	(LD)	(LE)	(LF)	(LG)	(LH)	(LI)	(LJ)	(LK)	(LM)	(LN)	(LO)	(LP)	(LQ)	(LR)	(LS)	(LT)	(LU)	(LV)	(LW)	(LX)	(LY)	(LZ)	(MA)	(MB)	(MC)	(MD)	(ME)	(MF)	(MG)	(MH)	(MI)	(MJ)	(MK)	(ML)	(MN)	(MO)	(MP)	(MQ)	(MR)	(MS)	(MT)	(MU)	(MV)	(MW)	(MX)	(MY)	(MZ)	(NA)	(NB)	(NC)	(ND)	(NE)	(NF)	(NG)	(NH)	(NI)	(NJ)	(NK)	(NL)	(NM)	(NO)	(NP)	(NQ)	(NR)	(NS)	(NT)	(NU)	(NV)	(NW)	(NX)	(NY)	(NZ)	(OA)	(OB)	(OC)	(OD)	(OE)	(OF)	(OG)	(OH)	(OI)	(OJ)	(OK)	(OL)	(OM)	(ON)	(OO)	(OP)	(OQ)	(OR)	(OS)	(OT)	(OU)	(OV)	(OW)	(OX)	(OY)	(OZ)	(PA)	(PB)	(PC)	(PD)	(PE)	(PF)	(PG)	(PH)	(PI)	(PJ)	(PK)	(PL)	(PM)	(PN)	(PO)	(PP)	(PQ)	(PR)	(PS)	(PT)	(PU)	(PV)	(PW)	(PX)	(PY)	(PZ)	(QA)	(QB)	(QC)	(QD)	(QE)	(QF)	(QG)	(QH)	(QI)	(QJ)	(QK)	(QL)	(QM)	(QN)	(QO)	(QP)	(QQ)	(QR)	(QS)	(QT)	(QU)	(QV)	(QW)	(QX)	(QY)	(QZ)	(RA)	(RB)	(RC)	(RD)	(RE)	(RF)	(RG)	(RH)	(RI)	(RJ)	(RK)	(RL)	(RM)	(RN)	(RO)	(RP)	(RQ)	(RR)	(RS)	(RT)	(RU)	(RV)	(RW)	(RX)	(RY)	(RZ)	(SA)	(SB)	(SC)	(SD)	(SE)	(SF)	(SG)	(SH)	(SI)	(SJ)	(SK)	(SL)	(SM)	(SN)	(SO)	(SP)	(SQ)	(SR)	(SS)	(ST)	(SU)	(SV)	(SW)	(SX)	(SY)	(SZ)	(TA)	(TB)	(TC)	(TD)	(TE)	(TF)	(TG)	(TH)	(TI)	(TJ)	(TK)	(TL)	(TM)	(TN)	(TO)	(TP)	(TQ)	(TR)	(TS)	(TT)	(TU)	(TV)	(TW)	(TX)	(TY)	(TZ)	(UA)	(UB)	(UC)	(UD)	(UE)	(UF)	(UG)	(UH)	(UI)	(UJ)	(UK)	(UL)	(UM)	(UN)	(UO)	(UP)	(UQ)	(UR)	(US)	(UT)	(UU)	(UV)	(UW)	(UX)	(UY)	(UZ)	(VA)	(VB)	(VC)	(VD)	(VE)	(VF)	(VG)	(VH)	(VI)	(VJ)	(VK)	(VL)	(VM)	(VN)	(VO)	(VP)	(VQ)	(VR)	(VS)	(VT)	(VU)	(VV)	(VW)	(VX)	(VY)	(VZ)	(WA)	(WB)	(WC)	(WD)	(WE)	(WF)	(WG)	(WH)	(WI)	(WJ)	(WK)	(WL)	(WM)	(WN)	(WO)	(WP)	(WQ)	(WR)	(WS)	(WT)	(WU)	(WV)	(WW)	(WX)	(WY)	(WZ)	(XA)	(XB)	(XC)	(XD)	(XE)	(XF)	(XG)	(XH)	(XI)	(XJ)	(XK)	(XL)	(XM)	(XN)	(XO)	(XP)	(XQ)	(XR)	(XS)	(XT)	(XU)	(XV)	(XW)	(XX)	(XY)	(XZ)	(YA)	(YB)	(YC)	(YD)	(YE)	(YF)	(YG)	(YH)	(YI)	(YJ)	(YK)	(YL)	(YM)	(YN)	(YO)	(YP)	(YQ)	(YR)	(YS)	(YT)	(YU)	(YV)	(YW)	(YX)	(YY)	(YZ)	(ZA)	(ZB)	(ZC)	(ZD)	(ZE)	(ZF)	(ZG)	(ZH)	(ZI)	(ZJ)	(ZK)	(ZL)	(ZM)	(ZN)	(ZO)	(ZP)	(ZQ)	(ZR)	(ZS)	(ZT)	(ZU)	(ZV)	(ZW)	(ZX)	(ZY)	(ZZ)	(00)	(01)	(02)	(03)	(04)	(05)	(06)	(07)	(08)	(09)	(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)

