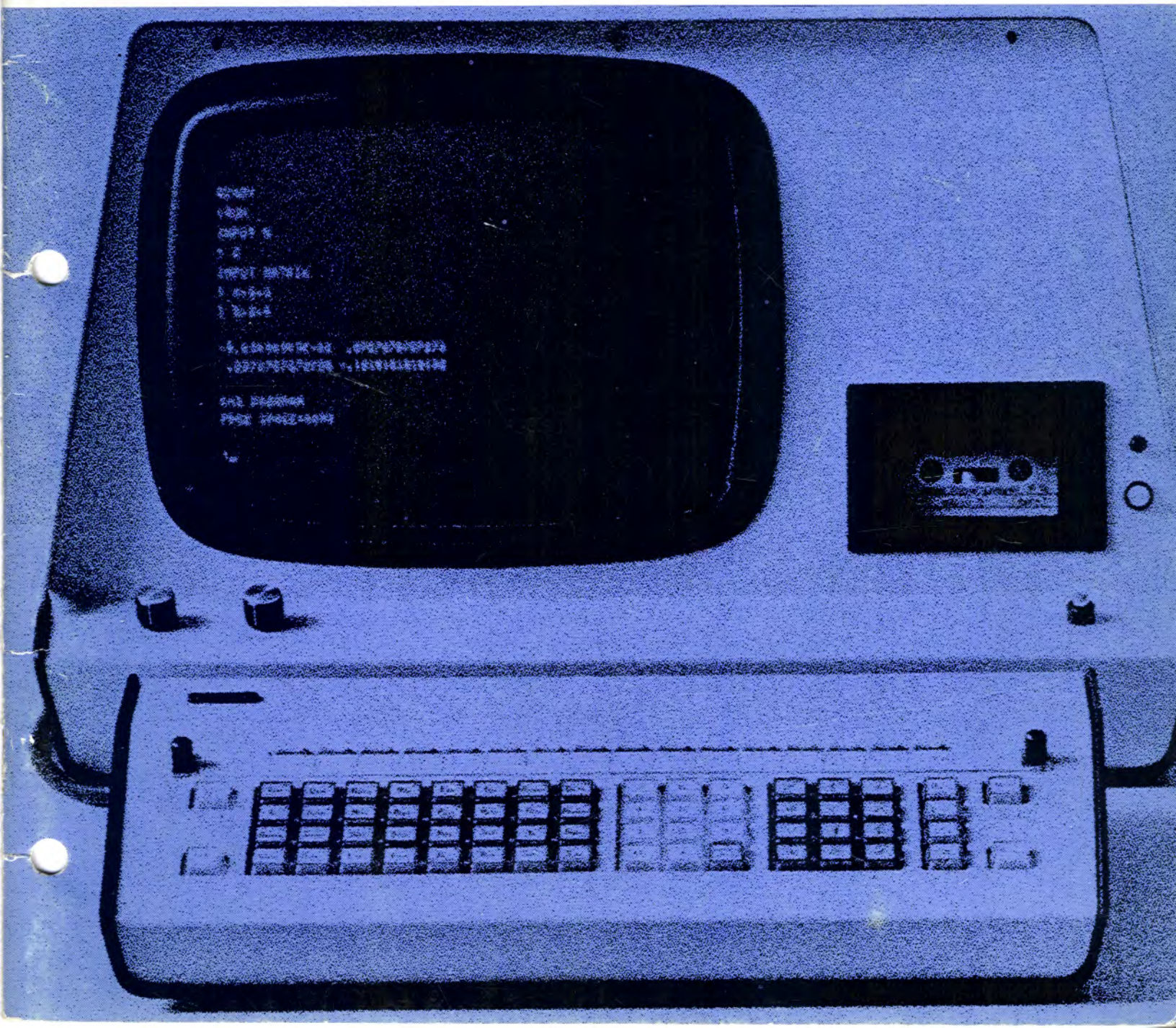
WANG

2203
HIGH-SPEED
PAPER TAPE READER

SYSTEM 2200



2203
High-Speed Paper Tape Reader

Reference Manual

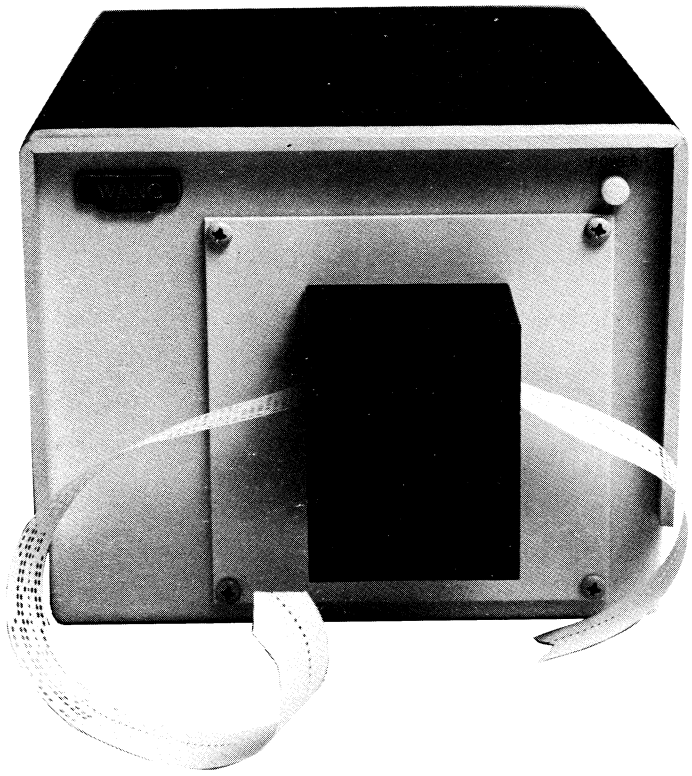
©Wang Laboratories, Inc., 1973

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343-6769, TELEX 94-7421

PREFACE

Your purchase of the WANG Model 2203 High-Speed Optical Paper Tape Reader represents perhaps one of the best investments you have ever made.

The 2203 adds a degree of flexibility to your WANG System 2200 which has hitherto been unavailable in its price range. The 2203's ability to read non-standard tapes as well as non-standard codes, coupled with the System 2200's ability to convert any code to ASCII (the code used internally by the 2200B) under 2200B software control add up to compatibility with virtually any off-line system you care to work with. Add to that the 2203's quiet, efficient and reliable operation, and you have an off-line system for data input which is tough to beat at any price.



HOW TO USE THIS MANUAL

The 2203 Manual has been designed and written as a handy instruction and reference manual for the 2203 High-Speed Paper Tape Reader only. It is assumed that the user has some familiarity with the System 2200B calculator and with the 2200B BASIC instruction set. Particularly in the discussion of bit and byte manipulation, space limitations have dictated a necessarily general discussion of some complex concepts. Further discussion of these concepts is undertaken in the System 2200B BASIC Reference Manual, to which the interested user is referred.

TABLE OF CONTENTS

Section I
General Information

UNPACKING AND INSPECTION

INSTALLATION

PREPARING THE MODEL 2203 TO READ PAPER TAPE

READING STANDARD AND NON-STANDARD TAPES

SECTION I GENERAL INFORMATION

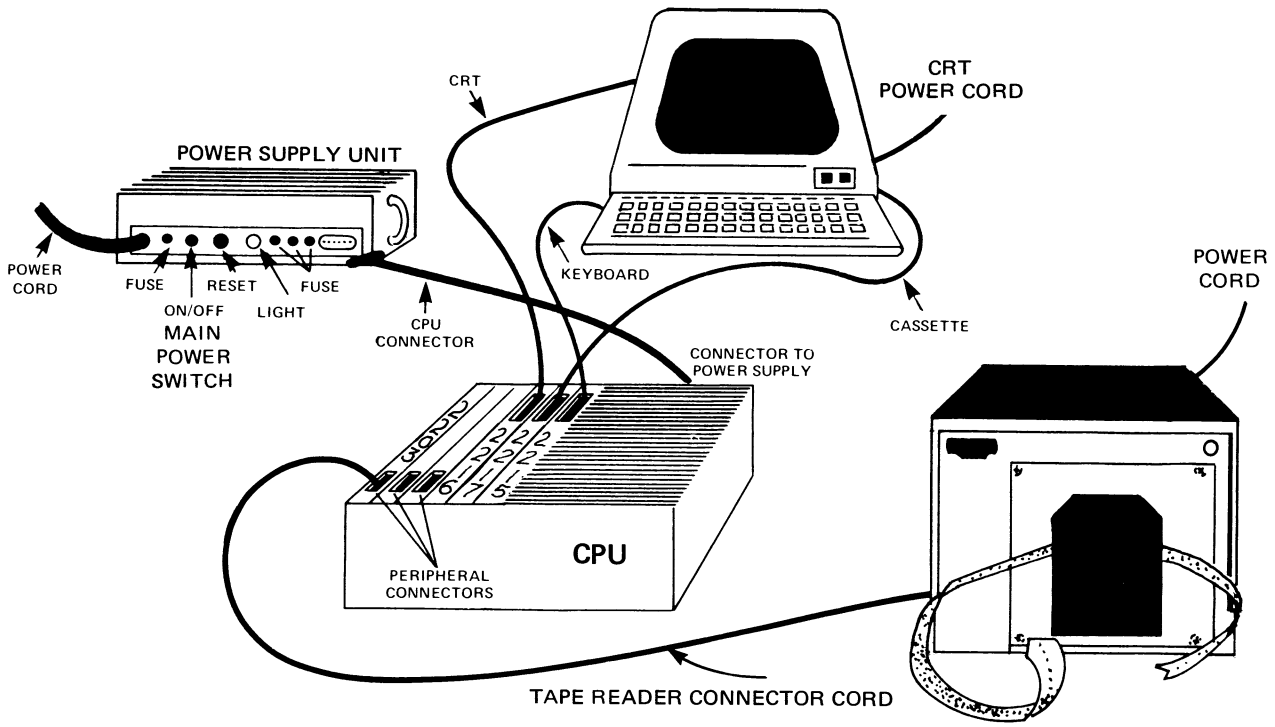
UNPACKING AND INSPECTION

(Call your Wang Service Representative immediately if there are any problems with your system.) Carefully unpack the equipment and inspect all units for damage. If equipment has been damaged, notify the shipping agency at once. Check all equipment received against the purchase order (decals specifying model numbers can be found on all Wang equipment, usually on the back of the unit).

INSTALLATION

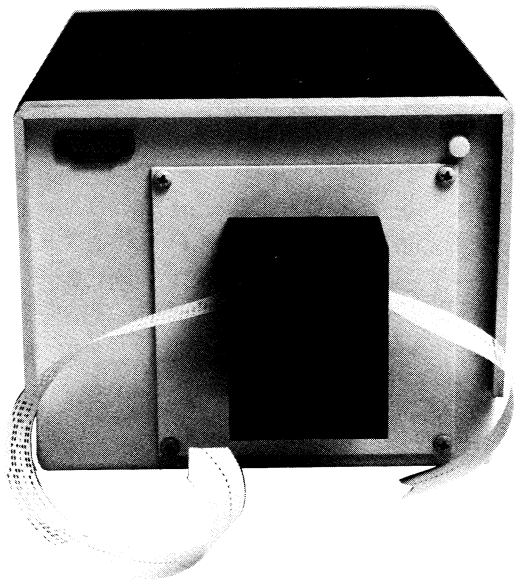
1. Turn ON/OFF switches on all equipment OFF.
2. Plug the 2203 connector cord into the appropriate peripheral connector on the CPU chassis (Labeled 2203).
3. Plug the 2203 power cord into a wall outlet (Power requirements of the 2203 are 115/230 volts and 50/60 Hz.).
4. Plug the main power cord of the CPU chassis into the Power Supply Unit; plug the power cord of the Power Supply Unit into a wall outlet.
5. Turn ON/OFF switches on all equipment ON. The system is now ready to use.

TYPICAL SYSTEM CONFIGURATION



PREPARING THE MODEL 2203 TO READ PAPER TAPE

1. Switch ON the POWER switch located on the rear panel of the 2203.
2. Flip up the hinged reader-head cover (see photo).
3. Insert the punched paper tape into the reader-head, in the direction indicated by the photo and the diagram below. The sprocket holes should be positioned so that the three(3) data channels 1-3 are on the inside of the tape (i.e., closer to the unit) and the five(5) data channels 4-8 are on the outside of the tape (i.e., further away from the unit) (see diagram). On the 2203, forward reading is from right to left, and reverse reading is from left to right.



4. After the tape is inserted, flip the reader-head cover back down to its read position. The 2203 is now ready to read your tape.

> **NOTE:**
>
> *It is important that the paper tape be properly inserted in the reader-head, since improper insertion will probably result in data loss or incorrect reading.*

The System 2200B BASIC statements which control tape reading either ignore unpunched leader code automatically, or can be programmed to do so. Thus, it isn't generally necessary to position the tape exactly on the starting frame.

READING STANDARD AND NON-STANDARD TAPES

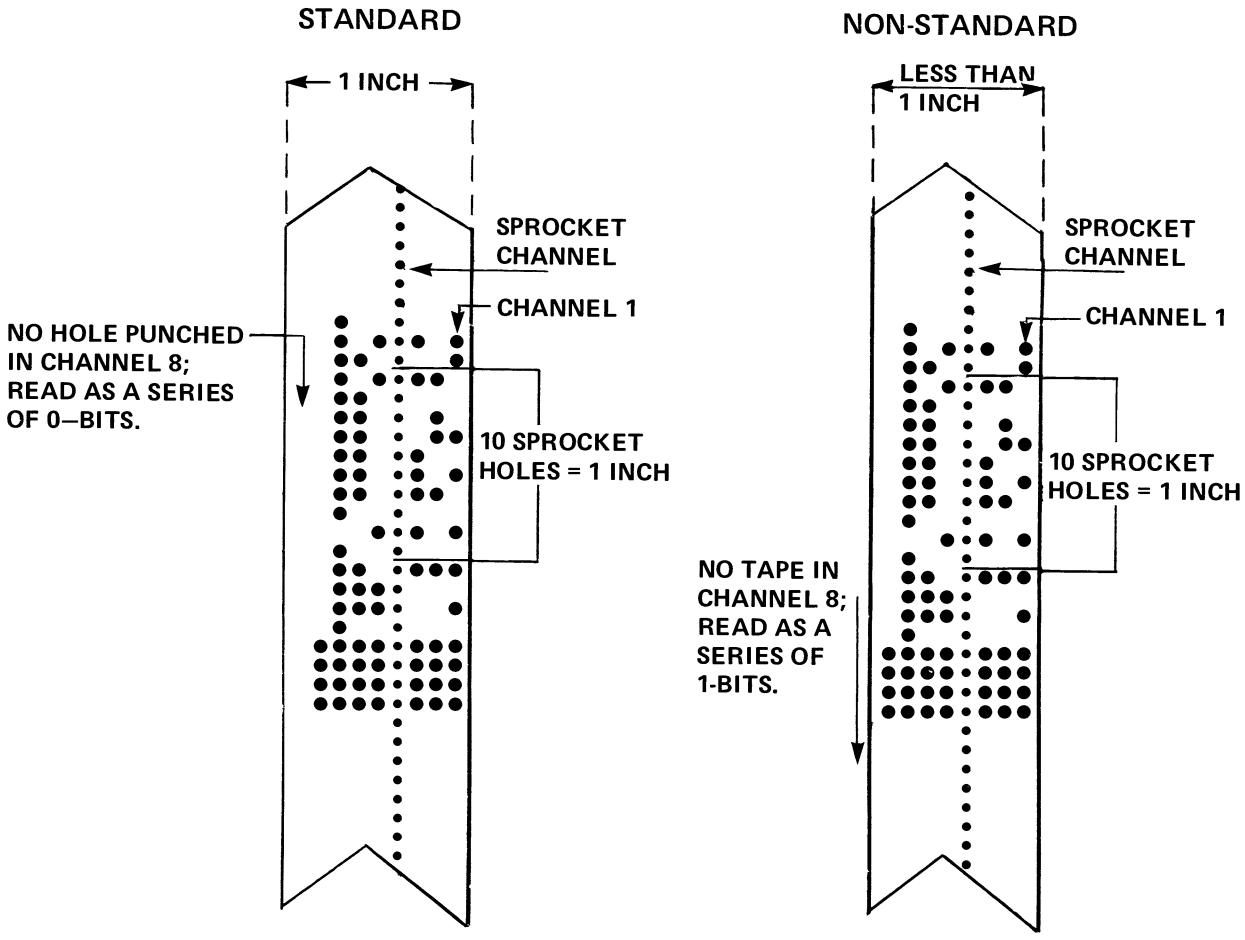
The Model 2203 High-Speed Optical Paper Tape Reader reads tape at the rate of approximately 300 cps (characters per second), forward or reverse. The rate may be slowed somewhat by drag on the tape with no data loss.

Standard punched paper tape has eight data channels with a single sprocket channel located between data channels 3 and 4 (see diagram). The 2203 optically scans these channels, reading a 1-bit where light passes through (hole punched) and 0-bit where no light passes through (no hole punched). Thus, for example, an ASCII Rubout, which is signified by all holes punched, is read as seven 1-bits. (Although channel 8 is called a data channel, it is reserved in ASCII code for the parity bit; since the System 2200 does not utilize a parity bit, however, the eighth channel contains no significant information, and is **ignored** when the 2203 is under LOAD or DATALOAD control. The eighth bit is read only when the 2203 is under DATALOAD BT control).

In addition to standard 8-channel, 1-inch tape, the 2203 can read narrower, non-standard tapes which conform to the following specifications:

1. The sprocket channel must be aligned with the data channels.
2. The sprocket hole must conform to the 2203 pin feed mechanism, which handles 10 holes per inch.
3. The ratio of the diameter of the data hole to the diameter of the sprocket hole must conform to punched paper tape standards.

On narrower tapes, a missing data channel (i.e., no tape) is read as a series of 1-bits.



Section II
Tape Reader Operation

READING DATA FROM PUNCHED PAPER TAPE

DEVICE SELECTION

THE LOAD STATEMENT AND COMMAND

THE DATALOAD STATEMENT

THE DATALOAD BT STATEMENT

SECTION II TAPE READER OPERATION

READING DATA FROM PUNCHED PAPER TAPE

Although the System 2200B utilizes the standard ASCII character set, it provides the capability, under software control, to convert data input on a character-by-character or bit-by-bit basis from any code into any other code. Thus, the 2200B can be made compatible with virtually any data-generating system.

Three BASIC command statements are utilized to access the Model 2203. Each BASIC statement has a unique function; these functions are briefly summarized below and then discussed in greater detail.

LOAD	Used to load program text lines from a paper tape and store them in memory. The tape must conform to the System 2200B paper tape format for programs.
DATALOAD	Used to load data from a paper tape and store them in designated variables or arrays. The tape must conform to System 2200B paper tape format for data.
DATALOAD BT (BLOCK TAPE)	Used to load data from a tape, without regard for control characters, and store it in a designated alphanumeric variable or array. All 8 bits of each character are read. Also used for reading tape in reverse direction. This statement is utilized most generally for tapes with a non-standard code or format.

DEVICE SELECTION

Because there are a large number of devices available for entering information into the System 2200, it is necessary, when executing a LOAD, DATALOAD, or DATALOAD BT statement, to have some means of identifying the Model 2203 as the device from which data is to be read. In the System 2200 there are three ways of designating, or selecting, a particular device for input.

Each peripheral device in the System 2200 is assigned a unique device address which can be used to identify that device. The device address of the Model 2203 is 618; thus a LOAD /618 command would instruct the system to begin loading program text from the 2203. The device address must be preceded by a "/".

Alternatively, the device address may be assigned to a particular file number with a SELECT statement. SELECT #1 618, for example, assigns the address of the 2203 (618) to file #1. This file number can then be used in place of the device address when accessing the Model 2203 (in this case, a LOAD #1 command would cause program text to be loaded from the 2203). The file number must be an integer from 1 to 6. It is always preceded by a "#".

Finally, if neither the device address nor the file number is given, the system automatically uses the device address of the Console Tape device. Unless otherwise specified, the Console Tape device is normally the tape cassette unit with device address 10A. However, it is possible to designate the 2203 as Console Tape device with a SELECT TAPE 618 statement. In this case, a LOAD command with no parameters causes program text to be loaded from the 2203.

Consult your **System 2200 Reference Manual** for further discussion of device selection and the SELECT statement.

SECTION II TAPE READER OPERATION

LOAD (Loading Program Text in ASCII from Punched Paper Tape)

In order to load programs or program segments from a paper tape and store them in memory, either the LOAD command or the LOAD statement may be used. The LOAD command and the LOAD statement have similar general forms; the difference is that in the LOAD command the 1st and 2nd line number parameters are not included.

General Form: LOAD [#n, /618,] [1st line number] [,2nd line number]

Where:

{#n /618} = the file number or device address of the 2203. Either the device address or the file number to which the address has been assigned (where #n is an integer from 1 to 6) may be specified. If neither is specified, the address of the device currently selected as Console Tape device is used.

1st line number (LOAD **statement** only) = the line number of the first statement line to be deleted from the program currently stored in memory prior to storing the new program. When loading is completed, execution continues automatically at the statement line whose number is equal to the 1st line number. If this line number does not appear in the new program, an error is indicated.

2nd line number (LOAD **statement** only) = the line number of the last statement line to be deleted from the program currently stored in memory prior to storing the new program.

NOTE:

Square Brackets must not be included in the actual BASIC program statements; they are used here only to indicate that a particular parameter is optional.

LOAD Command

The LOAD command is used in Immediate Mode; when it is keyed in, followed by a device address or file number if necessary, it instructs the 2203 to begin reading the program or program segment from paper tape and append it to the last program in memory. No parameters (i.e., 1st and 2nd line numbers) can be specified in the LOAD command, and no statement lines are cleared from memory prior to loading the new program. If line numbers of the new program being loaded from tape are identical to those of the program currently stored in memory, the new lines replace the old lines in memory. After the new program has been loaded, a RUN and an EXECUTE (CR/LF) command must be keyed in in order to begin program execution.

LOAD Command Examples:

```
LOAD /618

SELECT #3 618
LOAD #3
```

LOAD Statement

The LOAD statement is used to automatically load and execute a new program segment during the execution of a program. The LOAD statement provides the user with 3 options: either both the 1st and 2nd line number parameters may be specified, in which case all statement lines in memory between and including these two lines are cleared before the new program is loaded; or the 1st line number parameter only may be specified, in which case all of memory beginning at this line is cleared; or neither parameter may be specified, in which case the entire memory is cleared prior to loading the new program.

The first statement line number of the program segment being loaded from tape should coincide with the first statement line number specified in the LOAD statament (i.e., if 10 LOAD /618, 100, 200 is specified, the statement line numbers of the program on tape should begin at 100). When the new program has been loaded, program execution automatically begins with the first statement line number specified in the LOAD statement. If no statement line numbers are specified in the LOAD statement, execution begins with the first (lowest) statement line number in memory.

NOTE:

The LOAD statement can be used, in conjunction with a technique called "chaining", to make possible the running of a program which is too long to be completely stored in memory. For a detailed explanation of this technique, refer to Appendix B, "Chaining".

LOAD Statement Examples:

10 LOAD /618

No parameters specified, all memory cleared prior to loading program text from paper tape. After loading, program execution begins automatically at first statement line in memory.

10 SELECT #3 618
20 LOAD #3, 100, 200

Clear System 2200B memory from statement line 100 through line 200 inclusive, then load program from paper tape. After loading, execution begins automatically at statement line 100.

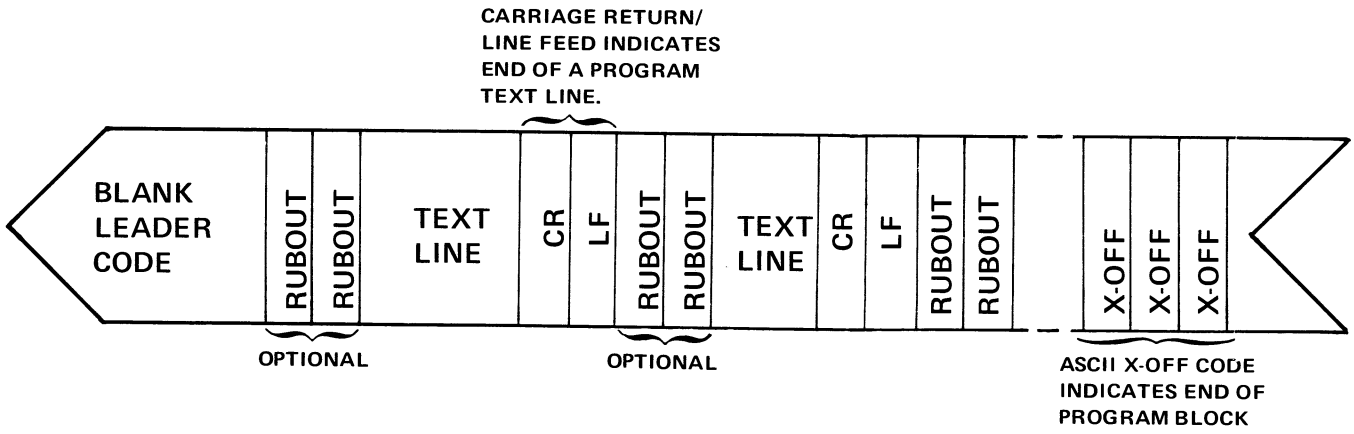
10 SELECT TAPE 618
20 LOAD 100

Clear System 2200B memory beginning at statement line 100, then load program from paper tape. After loading, program execution begins automatically at statement line 100. Since the 2203 was selected as Console Tape device in this example, (SELECT TAPE 618), there is no need for a device address or file number in the LOAD statement.

It must be emphasized that the LOAD statement or command is used **only** for the loading of program text lines from a tape which is punched in ASCII code and which has been formatted according to the specifications of the System 2200B. In general the LOAD statement or command is used to load programs which have previously been saved from a System 2200B.

SECTION II TAPE READER OPERATION

The System 2200B tape format for program text is:



Under LOAD control, only the first 7 bits of each character on tape are read; the eighth (parity) bit is ignored by the system (i.e., automatically set equal to 0 when loaded). The 2203 ignores all unpunched frames and Rubouts. The Rubouts indicated in the program tape format are, therefore, not required by the 2203. They are, however, required when programs are loaded directly from Teletype, and thus are normally punched when programs are saved on paper tape via Teletype.

DATALOAD (Loading Data in Free Format ASCII from Punched Paper Tape)

Data is loaded from a paper tape and stored in memory with the DATALOAD statement.

General Form: DATALOAD [#n, /618,] argument list

Where:

{#n /618} = the file number or device address of the 2203. Either the device address or the file number to which the address has been assigned (where #n is an integer from 1 to 6) may be specified. If neither is specified, the address of the device currently selected as Console Tape device is used.

argument list = the numeric, alphanumeric, scalar or array variables (or array elements) to which the data is to be assigned. Numeric or alphanumeric values may be assigned to alphanumeric variables, but values assigned to numeric variables must be legitimate BASIC numbers. Variables in the argument list must be separated by commas. An entire array is specified by the array name followed by left and right parentheses, e.g., A ( ), B$ ( ).

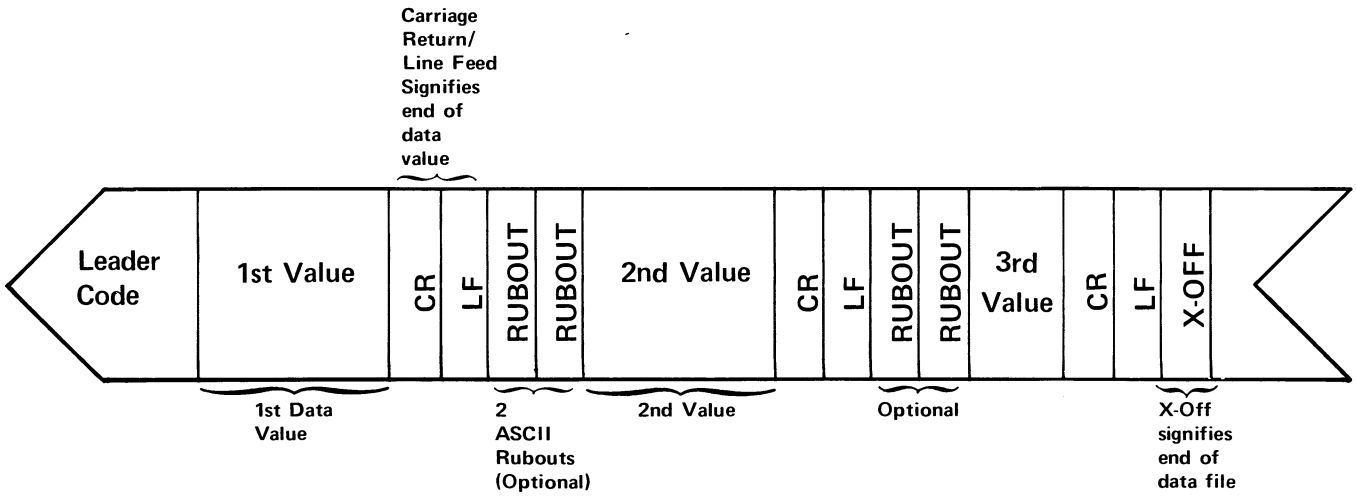
SECTION II TAPE READER OPERATION

The DATALOAD statement instructs the 2203 to begin loading data from the tape and sequentially assign them to the variables or arrays specified in the argument list. The eighth bit of each character is ignored (i.e., automatically set equal to 0 when loaded). Values are read from the tape until all variables have been satisified or until the end-of-file is encountered (i.e., an X-Off character is read). When an end-of-file is read, all remaining unsatisfied variables specified in the argument list are left with their current values. An IF END THEN statement in the program causes a valid transfer when the end-of-file is encountered.

> **NOTE:**
>
> *Arrays are filled row by row.*

Again, it must be emphasized that the DATALOAD statement is used **only** for loading data from paper tapes which are punched in standard ASCII code and formatted according to System 2200B specifications. In general, the DATALOAD statement is used for loading data which was originally saved **from** the System 2200B.

The System 2200B tape format for data is:



Alphanumeric and numeric values (in ASCII code) can be stored in alphanumeric variables, but only legitimate BASIC numbers in free-form format (e.g., +34.73, .07E+10, –73.41) should be stored in numeric variables. Carriage Return (CR) and Line Feed (LF) characters are always interpreted as terminators and cannot therefore be loaded and stored from tape as data under DATALOAD control.

Although the indicated Rubouts are **not** required for the 2203 DATALOAD they are required when loading data from a Teletype, and are therefore normally punched when saving data on paper tape via Teletype.

DATALOAD Examples:

```
10  SELECT #3 618
20  DATALOAD #3, N( ), A$, B$
```

Load data from paper tape into numeric array N( ) and alphanumeric variables A$ and B$.

```
10  DATALOAD /618,A1$(3), X, A$
```

Load and store data in alphanumeric array element A1$(3), numeric variable X, and alphanumeric variable A$.

```
10  SELECT TAPE 618
20  DATALOAD X, Y, A$, B$
```

Since the 2203 has been selected as Console Tape device, no address or file no. is necessary in the DATALOAD statement.

DATALOAD SAMPLE PROGRAM

PROGRAM	COMMENTS
10 DIM A$(5,5)25	Dimension A$ as a 2-dimensional array of 25 (5X5) elements, with a maximum of 25 characters per element.
.	
.	
40 SELECT #3 618	Assign device address 618 (i.e., the 2203) to file #3.
.	
60 DATALOAD#3, A$(),A,B	Load data from 2203 into array A$ (25 values) and then load data into numeric variables A, B (1 value each).
.	
70 IF END THEN 150	Skip to statement 150 when end-of-file (X-Off) is read, to stop program execution.
.	
.	
100 B$=A$(1,5)	Perform operations utilizing values assigned to variables from paper tape.
.	
140 GOTO 60	If end-of-file not read, loop back to read in more data.
150 END	

DATALOAD BT (BLOCK TAPE) (Loading Data or Program Text in any Format from Punched Paper Tape)

If data is to be processed from off-line equipment which does not utilize standard ASCII code or the System 2200B paper tape format, the DATALOAD BT statement is used in place of DATALOAD. All 8 bits of each character are read.

The DATALOAD BT statement has two basic functions:

1. To read and store data from a non-formatted tape;
2. To read data in a forward or reverse direction, depending upon whether the "R" parameter is or is not specified.

SECTION II TAPE READER OPERATION

The DATALOAD BT statement has the following general form:

General Form:

DATALOAD BT [R] [([N=expression] *,[L= {XX / alpha variable}] *,[S= {XX / alpha variable}])][{#n, / /618,}] {alpha variable / alpha array name}

Where:

R = reverse direction and read the data in reverse order. If the 'R' parameter is not specified, the 2203 automatically reads the tape in the forward direction (i.e., right to left).

N = the number of characters on the tape to be read (1 character = 1 byte = 8 bits). The value of the expression is truncated to an integer value; the value must be $\geqslant 1$.

L = the character which is to be read as the Leader code. If the initial frames contain this character, they are skipped over, and data is read beginning with the first character immediately succeeding the Leader code which is not equal to it. If no Leader code is specified, data is read beginning with the first character encountered (even if this is a leading blank). A two-digit HEX code or an alpha variable may be used for the Leader code. If an alpha variable is used, the first character of its value designates the Leader code.

S = the character which is to be read as the Stop code. When read, this character (8 bits) instructs the 2203 to stop reading data. Control then passes back to the System 2200B, which is positioned to begin execution of the next program statement. The Stop code can be designated by a 2-digit HEX code or an alpha variable. If an alpha variable is used, only the first character of its value is used to specify the Stop code.

#n
/618 = the file number or device address of the 2203. Either the device address (618) or the file number to which the address has been assigned (where #n is an integer from 1 to 6) may be specified. If neither is specified, the address of the device currently selected as Console Tape device is used.

alpha variable
alpha array name = a single alpha variable, alpha array name, or alpha array element can be specified to receive the read-in data. An entire alpha array is indicated by an alpha variable followed by left and right parentheses, as A$( ).

*When several parameters are included, commas must separate parameters.

SECTION II TAPE READER OPERATION

The DATALOAD BT statement causes data or program text to be loaded from a paper tape and stored in a single alpha variable or alpha array specified in the argument list. The entire 8 bits of each character are read and loaded into the alphanumeric string variable or array; the high-order (eighth) bit is **not** zeroed. The 2203 continues loading data or programs until one of the following conditions is met:

1. The designated Stop character is encountered; or
2. The alpha variable or array is full; or
3. The number of characters specified by N are read.

If data is stored in an alphanumeric array, any array elements which are unfilled when reading is terminated retain their original values.

> **NOTE:**
>
> *For a complete listing of all alphanumeric and special characters and their corresponding ASCII (HEX) codes and 8-bit forms, see Appendix C.*

DATALOAD BT Examples:

1. **100 DATALOAD BT /618, A$**

   Load data into alpha numeric variable A$. Since no other parameters are specified, data is loaded until A$ has been filled.

2. **100 SELECT #1 618**
   200 DATALOAD BTR (L = FF, S = OD) #1, A$(5)

   Reverse tape direction, begin reading data with 1st character not equal to HEX (FF). Stop reading when HEX code 0D is read (or when the array element A$(5) is filled). Store characters in array element A$(5).

3. **100 SELECT TAPE 618**
   200 DATALOAD BT (N = 100), A$( )

   Read 1st 100 characters from tape and store in array A$. Since the 2203 has been selected as Console Tape device, no address or file number is necessary in the DATALOAD BT statement.

4. **90 T$ = "JOHN JONES"**
   100 SELECT #3 618
   200 DATALOAD BT (N=100, L=T$, S=OD) #3, A1$

Set alpha variable T$ equal to "John Jones." Then, load data from paper tape, beginning with the first character *not* equal to an ASCII "J" (the first character of the value of T$, specified as Leader code). Continue reading data from tape *either* until 100 characters have been read *or* until a character whose code is read (or until the alpha variable A1$ is filled). Store this information in a single alphanumeric variable, A1$.

DATALOAD BT
SAMPLE PROGRAM

The DATALOAD BT statement can be used to load and store data from an unformatted paper tape. The data can then be converted into a form usable by the System 2200B. Suppose, for example, that you have punched a number of data values on paper tape separated with a single space between each pair of values. The values are punched in ASCII code, but since your tape isn't formatted according to System 2200B specifications (i.e., Carriage Return and Line Feed characters separating values), it's not possible to use the DATALOAD statement. You can, however, use the DATALOAD BT statement, which causes every character in a specified character string to be loaded and stored into an alphanumeric variable or array. Since arithmetic operations can be performed only on values stored in numeric variables or arrays, however, it will be necessary to transfer the values out of the alpha variable, convert them to numeric values, and store them in a numeric array.

Your approach might be to load each value from the paper tape into an alphanumeric string variable, T$, then use the CONVERT function to transfer it into a numeric array, A( ). Since you know that each value on tape is followed by a single space character, you can read a single value from the tape simply by specifying the HEX code for space (20) as Stop character in the DATALOAD BT statement. This causes the string of characters up to the space character to be read and stored in the designated alpha variable. The CONVERT function could then be used to convert and transfer each value into the numeric array. A "dummy value" (say, 999) could be punched at the end of the paper tape to indicate the end of data.

A program which does this might be written as follows:

	PROGRAM	COMMENTS
10	DIM T$64, A(10)	Dimension alphanumeric string variable T$ to 64 characters (max. allowable) and numeric array A for 10 elements (a numeric value to be stored in each element).
20	I=0	Initialize counter at 0.
30	T$=" "	Initialize T$ to contain only blanks. When a number of values are to be stored and transferred from an alpha variable or array under DATALOAD BT, the variable or array **must** be re-initialized to contain blanks every time a new value is to be stored in it.
40	I=I+1	Counter incremented.

50	DATALOAD BT (S=20)/618, T$	Read a string of characters from tape and store in T$ until a space character (HEX(20)) is read.
60	CONVERT T$ TO A(I)	The string of ASCII characters representing each number read into T$ is converted into a 2200 internal decimal number and transferred into the numeric array element A(I). The values read in must, however, be ASCII characters representing legitimate BASIC numbers (e.g., 4.2, –57.936, 1.2E –07).
70	IF A(I) = 999, THEN 90	When dummy value is read, stop reading tape and exit read loop.
80	GOTO 30	If dummy value not read, loop back to re-initialize T$ and read next value.
90	END	

Section III
Appendices

APPENDIX A – CLEANING THE 2203

A Tape Reader Cleaner Kit is included with each Model 2203 High-Speed Paper Tape Reader. The kit includes a brush, rectangular pieces of lint-free plastic wiper strips, and a bottle of tape reader cleaner (inhibited 1–1–1 Trichloroethane).

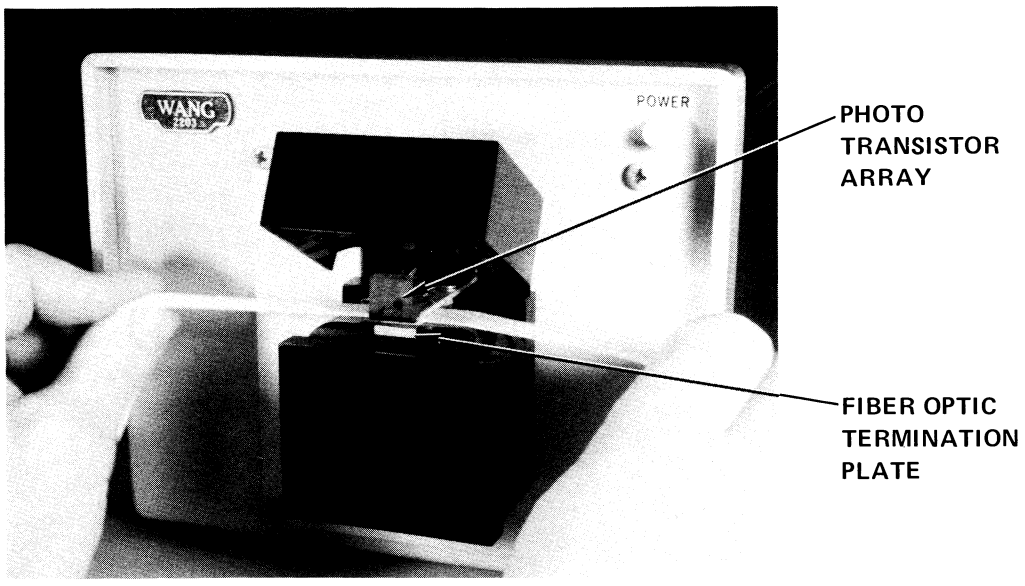
The frequency with which cleaning is required depends upon the cleanliness of the environment and the condition of the tapes being used, particularly in the area of tape splices. However, it is recommended that the 2203 be cleaned weekly for the first three or four weeks. Thereafter, the cleaning schedule should be adjusted according to the amount of dirt picked up by the lens wiper.

CLEANING PROCEDURE

1. Use the brush to remove visible dust from both inside and outside the reader cover housing.
2. Saturate approximately one inch of the center of the lens wiper strip with tape reader cleaner solution.
3. Stretch the lens wiper strip to decrease its thickness so that it fits between the photo-transistor array and the fiber optic termination plate, inside the reader-head.
4. Decrease the stretch to a point where the wipe contacts the upper and lower surfaces with a wiping action.
5. Move the wipe back and forth in the direction of tape travel 30 or 40 times.

> **NOTE:**
>
> *Movement should be relatively easy. If movement is difficult, stretch the wiper a little more.*

6. Remove the wiper and inspect for excessive dirt.
7. Saturate the center section of another wiper strip with solvent.
8. Insert and move the strip back and forth 10 or 20 times.
9. Remove the wiper strip and inspect it for dirt. If no dirt has been collected, consider the reader head to be clean. If dirt is present on the wiper, repeat step 8 until no dirt appears.



APPENDIX B – CHAINING

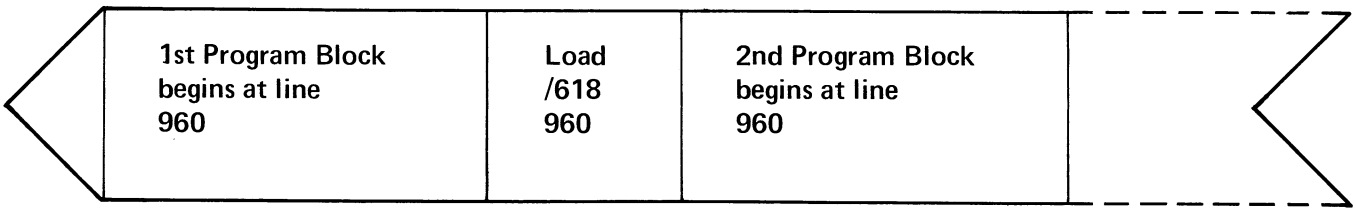
One important use of the LOAD statement is in the technique known as chaining. Chaining makes possible the execution of programs too large to be stored in memory. Although this technique is most commonly used with tape cassettes, it is readily adaptable for use with the paper tape reader. Suppose, for example, that you have written a program too large for the 8K memory of your machine (remember that 700 bytes are reserved for "housekeeping" and cannot be utilized for storage of program text or data; since an 8K machine actually has 8192 bytes of storage space, this leaves about 7500 bytes available for storing program text). Suppose, further, that your program incorporates a large number of subroutines which are referred to throughout the program. The following procedure would allow you to "chain" the program and run it despite the fact that it is too large for memory.

1. Store the subroutines which are common throughout the program in memory (assume they occupy lines 10-950 and take up 2400 bytes of memory).
2. Assume that 2100 bytes are required for the storage of variable data.
3. Write the main program and save it on punched paper tape. The program must be broken up into a series of segments, or 'blocks', designed to fit into the remaining memory space (in this case, 3000 bytes). Assuming that your main program occupies 6000 bytes, you will need to break it down into two blocks of 3000 bytes each (i.e., 2400 + 2100 + 3000 = 7500 bytes).

> **NOTE:**
>
> *Since program text lines have varying lengths, it is often difficult to estimate how many bytes of memory a program requires. Refer to Appendix B of your* **System 2200 BASIC Programming Manual**, *"Estimating Program Memory Requirements".*

4. Number the program text lines in each program block with the same sequence of line numbers, starting with the line number immediately following the last line of programming in memory (in this case, since your subroutines occupy lines 10-950 in memory, you would begin numbering the text lines in each block on tape at 960).
5. At the end of the first block, as the final statement in that block, add a LOAD /618, 960 statement.
6. Insert the tape in the Paper Tape Reader and key a LOAD command (in Immediate Mode).

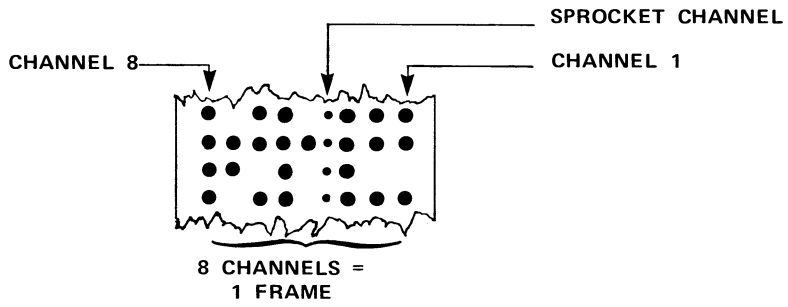
The calculator begins execution of the program immediately after it is stored in memory. It executes through the main program, diverging to subroutines as instructed, and comes finally to the LOAD statement. It then clears memory starting at line 960, and loads the next block of program text into the same space just occupied by the first block. This segment is executed until a second LOAD statement is encountered, and the process is repeated. In this way, extremely long programs can be broken down into segments which will fit into memory, be executed, and then discarded to make room for the next segment. Operations commonly used throughout the program can, on the other hand, be written as subroutines and permanently stored in memory.



APPENDIX C – ASCII CODES

ASCII character code is a standard coding system in which each number, letter, and symbol in the ASCII character set is assigned a unique 8-bit binary code (although in fact only 7 bits are actually used for the code, since the 8th bit is reserved in ASCII for parity). In the System 2200B, the 8th bit is automatically set equal to 0 when read in under LOAD and DATALOAD control; it is actually read only under DATALOAD BT control). This 8-bit binary code is, in turn, translated into a pattern of punches on paper tape.

Consider the following section of a paper tape:



NOTE:

In order to read a paper tape properly, always position it so that the three data channels 1-3 are to the right of the sprocket channel and the five data channels 4-8 are to the left of the sprocket channel.

Each channel in which a hole is punched is read by the 2203 as a "1"; each channel in which no hole is punched is read as a "0". Thus each frame on the paper tape really represents a binary value, and each individual channel represents a binary digit, or bit (BInary digiT), either 1 or 0, depending on whether that channel is or is not punched.

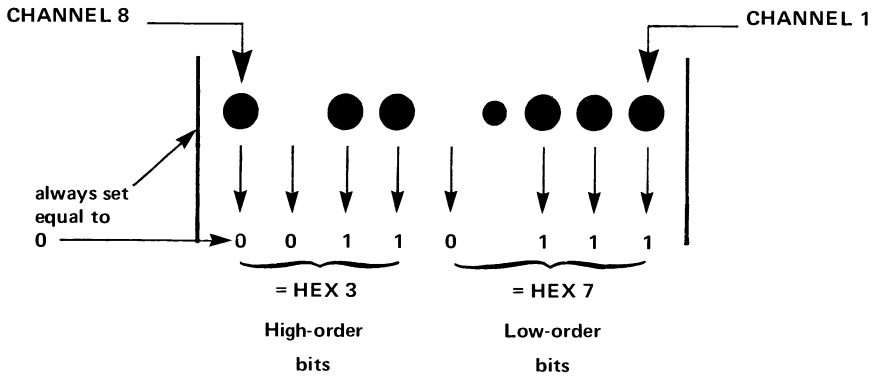
The binary value punched in the first frame illustrated above is 10110111; remember, however, that the 8th bit is always set equal to 0 whether it is punched or not, under LOAD and DATALOAD control. Thus the binary value actually read into the system is 00110111. The table of ASCII codes on page 22 shows that 00110111 is the binary code in ASCII for the number 7. Therefore, this particular frame of a paper tape contains the number 7 in ASCII code. You can read any frame on tape in the same fashion, by checking the binary code in that frame and finding the character which corresponds to that code in the ASCII table.

Despite its value as a code which is readily translatable into a pattern of punches on tape, however, the notation of the binary system is somewhat ponderous to work with. For this reason, the Hexadecimal system has been developed as a means of expressing binary values in a shorthand notation. The Hexadecimal system is to the base 16. Every 4-bit binary value can be expressed by one of the 16 Hexadecimal digits. Thus every 8-bit binary value can be expressed as a 2-digit HEX number, and therefore every 8-bit binary code in ASCII is expressible as a 2-digit HEX number. Frequently in dealing with ASCII codes it is more efficient to refer to the 2-digit HEX code for a particular character than its equivalent 8-bit binary code. The 16 HEX digits with their binary as well as decimal equivalences are listed below (for a more detailed discussion of the Hexadecimal counting system and the System 2200 HEX function, consult your **System 2200 BASIC Programming Manual**).

APPENDIX C – ASCII CODES

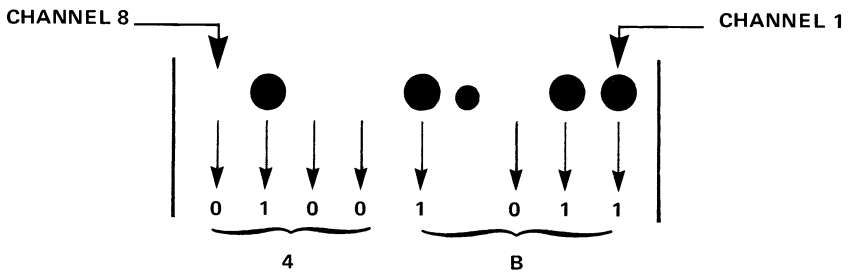
BINARY	HEXADECIMAL	DECIMAL EQUIVALENCE
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

The 2-digit HEX code for each frame of tape is related to the 8-bit binary code in the following way:



As you can see, the right 4 bits of each 8-bit binary value (tape channels 1-4) are represented by the right digit of the 2-bit HEX code; they are referred to as the "low-order" bits. The left 4 bits (channels 5-8) are represented by the left digit of the HEX code; they're called the "high-order" bits. The binary value in the above frame, then, is expressed in HEX shorthand as 37. The table on page 22 shows that HEX 37, like its binary equivalent 00110111, is the code in ASCII for 7. Often you'll find it convenient when reading a tape to convert each frame into a HEX code and then check the table for the characters corresponding to these HEX codes.

Letters and symbols are handled in a manner exactly analogous to numbers. The letter K, for example, is assigned the binary code 01000001 (HEX 4B) in ASCII. A frame punched with the letter K, then, would look like this:



APPENDIX C – ASCII CODES

BASIC SYMBOL	HEX CODE	8-BIT FORM Tape Channels 8 1
LINE FEED (LF)	0A	00001010
CARRIAGE RETURN (CR)	0D	00001101
X–ON	11	00010001
X–OFF	13	00010011
SPACE	20	00100000
!	21	00100001
DOUBLE QUOTE	22	00100010
#	23	00100011
$	24	00100100
%	25	00100101
&	26	00100110
SING. QUOTE	27	00100111
(	28	00101000
)	29	00101001
*	2A	00101010
+	2B	00101011
, (comma)	2C	00101100
–	2D	00101101
. (decimal point)	2E	00101110
/ (slash)	2F	00101111
0	30	00110000
1	31	00110001
2	32	00110010
3	33	00110011
4	34	00110100
5	35	00110101
6	36	00110110
7	37	00110111
8	38	00111000
9	39	00111001

APPENDIX C – ASCII CODES

BASIC SYMBOL	HEX CODE	8-BIT FORM Tape Channels 8 1
:	3A	00111010
;	3B	00111011
<	3C	00111100
=	3D	00111101
>	3E	00111110
?	3F	00111111
@	40	01000000
A	41	01000001
B	42	01000010
C	43	01000011
D	44	01000100
E	45	01000101
F	46	01000110
G	47	01000111
H	48	01001000
I	49	01001001
J	4A	01001010
K	4B	01001011
L	4C	01001100
M	4D	01001101
N	4E	01001110
O	4F	01001111
P	50	01010000
Q	51	01010001
R	52	01010010
S	53	01010011
T	54	01010100
U	55	01010101
V	56	01010110
W	57	01010111

APPENDIX C – ASCII CODES

BASIC SYMBOL	HEX CODE	8-BIT FORM Tape Channels 8 1
X	58	01011000
Y	59	01011001
Z	5A	01011010
[	5B	01011011
]	5D	01011101
↑ (up arrow)	5E	01011110
a	61	01100001
b	62	01100010
c	63	01100011
d	64	01100100
e	65	01100101
f	66	01100110
g	67	01100111
h	68	01101000
i	69	01101001
j	6A	01101010
k	6B	01101011
l	6C	01101100
m	6D	01101101
n	6E	01101110
o	6F	01101111
p	70	01110000
q	71	01110001
r	72	01110010
s	73	01110011
t	74	01110100
u	75	01110101
v	76	01110110
w	77	01110111
x	78	01111000
y	79	01111001
z	7A	01111010

APPENDIX D – MODEL 2203 SPECIFICATIONS

Type of Reader

Optical

Tape Size

Standard 8-channel, one-inch paper tape, as well as narrower tapes which conform to the following specifications:

a. The sprocket holes must conform to the 2203 pin-feed mechanism, which handles 10 sprocket holes per inch;

b. The sprocket holes must align with the data holes;

c. The ratio between diameter of sprocket hole and diameter of data hole must conform to punched tape standards.

Tape Code

Standard 8-level ASCII is used by the System 2200; any other code can be read in and converted to internal numbers or ASCII characters under 2200B software control.

Reading Speed

300 characters per second (average).

Size:

Height	6.5 in. (16.5 cm.)
Width	7.5 in. (19.1 cm.)
Depth	11.5 in. (29.2 cm.) [13.5 in. counting reader head]
Weight	13 lbs. (5.9 kg.)

Power Requirements

115/230 volts ± 10%, 50/60 Hz

Cable

9 ft. (2.74 m.)

Operating Environment

50°F (10°C) – 104°F (40°C)

20% – 80% Relative Humidity

INDEX

To help us to provide you with the best manuals possible, please make your comments and suggestions concerning this publication on the form below. Then detach, fold, tape closed and mail to us. All comments and suggestions become the property of Wang Laboratories, Inc. For a reply, be sure to include your name and address. Your cooperation is appreciated.

TITLE OF MANUAL:

COMMENTS:

Fold

Fold

(Please tape. Postal regulations prohibit the use of staples.)

WANG

Fold

FIRST CLASS
PERMIT NO. 16
Tewksbury, Mass.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

— POSTAGE WILL BE PAID BY —

WANG LABORATORIES, INC.
836 NORTH STREET
TEWKSBURY, MASSACHUSETTS 01876

Attention: Marketing Department

Fold

Cut along dotted line.

Printed in U.S.A.

WANG LABORATORIES (CANADA) LTD.
180 Duncan Mill Road
Don Mills, Ontario M3B 1Z6
TELEPHONE (416) 449-7890
TELEX 06-966546

WANG EUROPE, S.A.
Buurtweg 13
9412 Ottergem
Belgium
TELEPHONE: 053/74514
TELEX: 26077

WANG ELECTRONICS LTD.
1, Olympic Way, 4th Floor
Wembley Park,
Middlesex, England
TELEPHONE: 01/903/6755
TELEX: 922161

WANG FRANCE SARL
47, Rue de la Chapelle
Paris 18, France
TELEPHONE 203.27.94 or 203.25.94

WANG LABORATORIES GMBH
Moselstrasse No. 4
6000 Frankfurt am Main
West Germany
TELEPHONE (0611) 252061-64

WANG SKANDINAVISKA AB
Fredsgaten 17
S-172-23
Sundbyberg 1, Sweden
TELEPHONE 08-98-12-45

WANG NEDERLAND B.V.
Damstraat 2
Utrecht, Netherlands
TELEPHONE 030-930947

WANG PACIFIC LTD.
61, King Yip Street, 1st Floor
Kwun Tong Kowloon, Hong Kong
TELEPHONE 3-434231/2

WANG INDUSTRIAL CO., LTD.
110-118 Kuang-Fu N. Rd.
Taipei, Taiwan
Republic of China
TELEPHONE 784181-3

WANG GESELLSCHAFT MBH
Grinzinger Allee 16
1190 Vienna 19
Austria
TELEPHONE (0222) 32.42.43

WANG COMPUTER PTY. LTD.
25 Bridge Street
Pymble, NSW 2073
Australia
TELEPHONE 449-6388

WANG INTERNATIONAL TRADE, INC.
836 North Street
Tewksbury, Massachusetts 01876
TELEPHONE (617) 851-4111
TWX 710-343-6769
TELEX 94-7421

WANG COMPUTER SERVICES
836 North Street
Tewksbury, Massachusetts 01876
TELEPHONE (617) 851-4111
TWX 710-343-6769
TELEX 94-7421

24 Mill Street
Arlington, Massachusetts 02174
TELEPHONE (617) 648-8550

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851 4111 TWX 710 343 6769 TELEX 94 7421

Printed in U.S.A.
700-3191
11-73-1.5M
Price $7.50