E. C. Greanias
C. J. Hoppel
M. Kloomok
J. S. Osborne

Design of Logic for Recognition of Printed Characters by Simulation

Abstract: A logic for the recognition of printed characters employing a "proportional parts" method has been tested extensively by simulation. The proportional parts method relates the identity of a character to the relative size and position of character elements detected by optical scanning along closely spaced vertical lines. The video information from each scan is coded to designate the number, size, and position of inked areas detected by the scanner. The time sequence or order of occurrence of this coded information is tested by logical circuitry against prescribed sequences for character recognition. A set of coding definitions, a prescribed sequence for the recognition of a given character, and the methods of simulative testing of the recognition logic on the IBM Type 650 Magnetic Drum Data Processing Machine are discussed.

Inventors have been trying to develop machines to read printed matter for more than thirty years. During this period patents have been issued on at least twenty different schemes for character recognition. The usual procedure in such developments has been to build a model of each device, test it, observe its limitations, and then build an improved model. This approach has proved to be costly and time consuming, and the devices that have been developed could not recognize realistic characters with the accuracy required for accounting purposes. In general, the early recognition schemes did not provide means to detect relatively subtle differences between degraded characters which are obvious to the human eye and must be seen for accurate recognition. To sense such characters requires many more logical operations on highly detailed information than the early schemes employed.

The advent of digital computers with adequate capacity now makes it possible to rapidly test, modify and retest the logic of such recognition schemes by simulation. In this manner the feasibility of a system can be well established before any construction is undertaken. By repetitive modification and simulation the final design of the recognition criteria can be determined in considerable detail. The recognition system described here was developed by simulative testing and modification equivalent to extensive machine testing before an experimental machine was constructed.

Recognition logic fundamentals

The particular recognition logic developed in this work has been called a proportional parts system. It relates the identity of a character to the relative size and position of character elements detected by serial scanning with a fine spot of light along closely spaced vertical lines. Scanning proceeds from top to bottom and from right to left across the characters. For the logical analysis, the light signals reflected from the paper are first quantized into black and white cells. The binary data from each vertical scan are then coded in three decimal digits which designate the number, position, and size of the black areas in the scan. The first digit relates to the number and proportions of the black areas detected. The second digit indicates the change in altitude of the top of the character relative to a selected earlier scan. The third digit designates the distance between extreme black areas on the scan.

Individual characters can be represented by a succession of coded scan data, and the basic recognition criteria for each character can be defined as a sequence of these codes. Recognition sequences are defined in terms of the allowable codes that will satisfy each stage of the sequence, the maximum number of scans that can occur before the next stage of the sequence is satisfied, and specific inhibiting codes that must not occur between particular stages of some sequences. Recognition sequences are implemented by the use of two basic logical elements, delay units and inhibitor units. Delay units are employed at all entry and intermediate stages of the prescribed recognition sequences. These are triggered whenever an acceptable combination of events occurs, and remain "on" for a specific number of scans characteristic of the individual delay units. Entry delay units are triggered by the occurrence of specified code numbers, whereas intermediate delay units are triggered when the preceding delay units are "on," the specified new codes occur and no inhibiting signals are present. Inhibitor units are used selectively to prevent the triggering of an

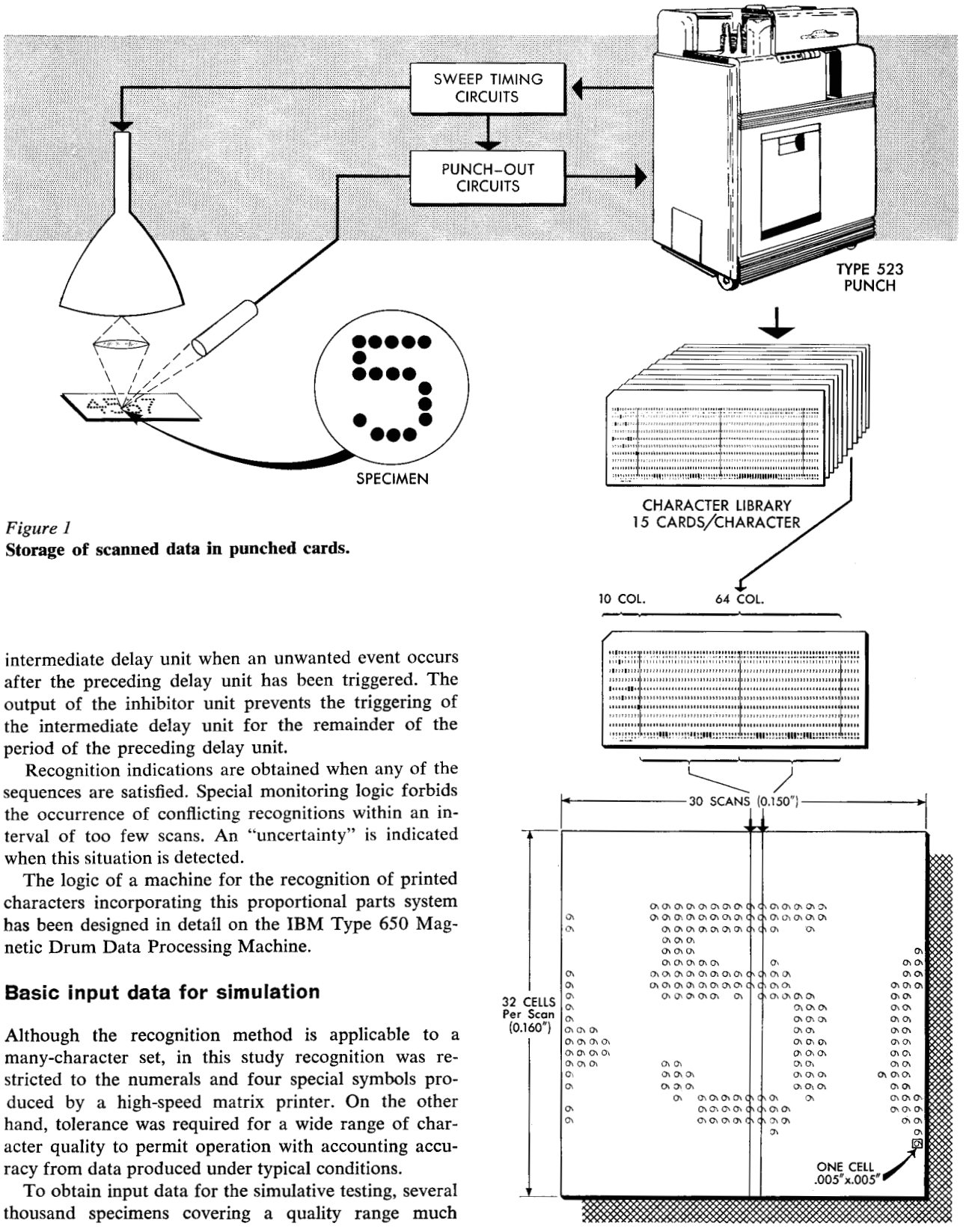


Figure 1
Storage of scanned data in punched cards.

intermediate delay unit when an unwanted event occurs after the preceding delay unit has been triggered. The output of the inhibitor unit prevents the triggering of the intermediate delay unit for the remainder of the period of the preceding delay unit.

Recognition indications are obtained when any of the sequences are satisfied. Special monitoring logic forbids the occurrence of conflicting recognitions within an interval of too few scans. An "uncertainty" is indicated when this situation is detected.

The logic of a machine for the recognition of printed characters incorporating this proportional parts system has been designed in detail on the IBM Type 650 Magnetic Drum Data Processing Machine.

Basic input data for simulation

Although the recognition method is applicable to a many-character set, in this study recognition was restricted to the numerals and four special symbols produced by a high-speed matrix printer. On the other hand, tolerance was required for a wide range of character quality to permit operation with accounting accuracy from data produced under typical conditions.

To obtain input data for the simulative testing, several thousand specimens covering a quality range much

greater than the normal output of properly adjusted printers were test-scanned with a laboratory flying spot scanner. Each vertical scan was sub-divided into 32 sample intervals and scans were made every 0.005 inches. The equipment was adjusted so that the threshold circuits would always designate a cell as "black" if the reflected light signal exceeded a selected threshold for 40% of the sample interval and a cell would always be designated as "white" if the light signal exceeded the threshold for less than 15% of the interval.

The movement of the flying spot over the specimen was precisely timed and synchronized with the operation of an IBM Type 523 Summary Punch. Test scanning was done at moderate speeds, approximately one millisecond per vertical scan. Waiting periods were provided between scans to synchronize the scanning with the action of the punch. The technique for permanent storage of the scanned data in punched cards is illustrated in Figure 1.

Ten columns of each card are reserved for information identifying the character, the scan number, the type font, the paper and the ribbon. An additional 64 columns of the card are used to store the binary data representing two scans (32 bits per scan). A "9" punch represents a black cell, a blank column represents a white cell. In order to monitor the test data manually, "print-outs" of each specimen are obtained by listing the cards on an IBM Type 402 Accounting Machine. A listing of two typical characters is illustrated in Figure 2. After careful consideration, 3000 representative specimens were selected for the actual simulation. Half of these specimens were used for developing optimum character recognition sequences, and the remaining half were run against the final sequences as a check.

Coding video data as first step in simulation

The first step of the particular recognition process being described is the encoding of the 32 binary bits representing each vertical scan into one three-digit number. The first digit describes the number of black areas in the scan and their relative size and disposition. For example, the single black areas encountered in the stem of the numeral "1" or in the left overhang of the numeral "7" are coded as ones. Two black areas, each of short duration, such as those occurring in the center of the "0" are assigned the code two. When two black areas occur and the upper area is longer than the lower, as in the left edge of the "5" or the right edge of the "2," a code of four is assigned. The reverse condition, where the lower black area is longer than the upper as in the center of the "7," is an example of the code five.

The second digit of the three-digit code provides a measure of the difference in altitude of the top of the character at any given scan as compared with data taken three scans previously. When the top of the character is straight, as in the "7", a code of zero is assigned. A sloping drop off as in the "4" is coded as two. Five values are defined for the second digit.

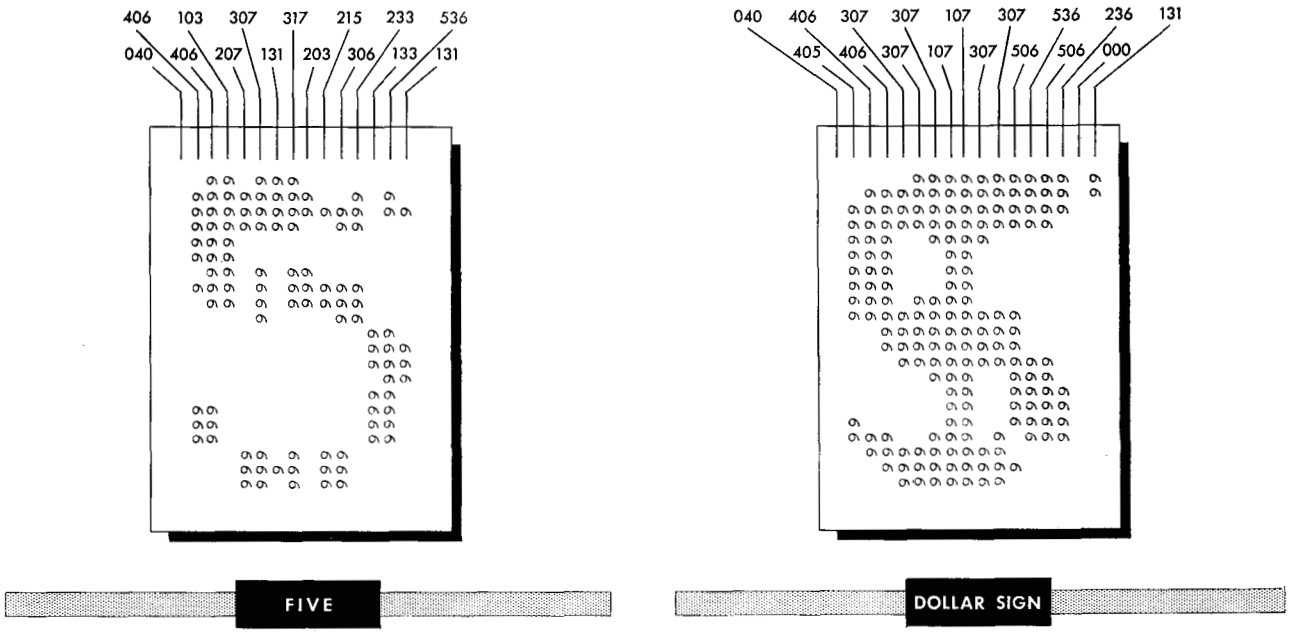


Figure 2
Print-out of scan data for two sample characters with coding indicated.

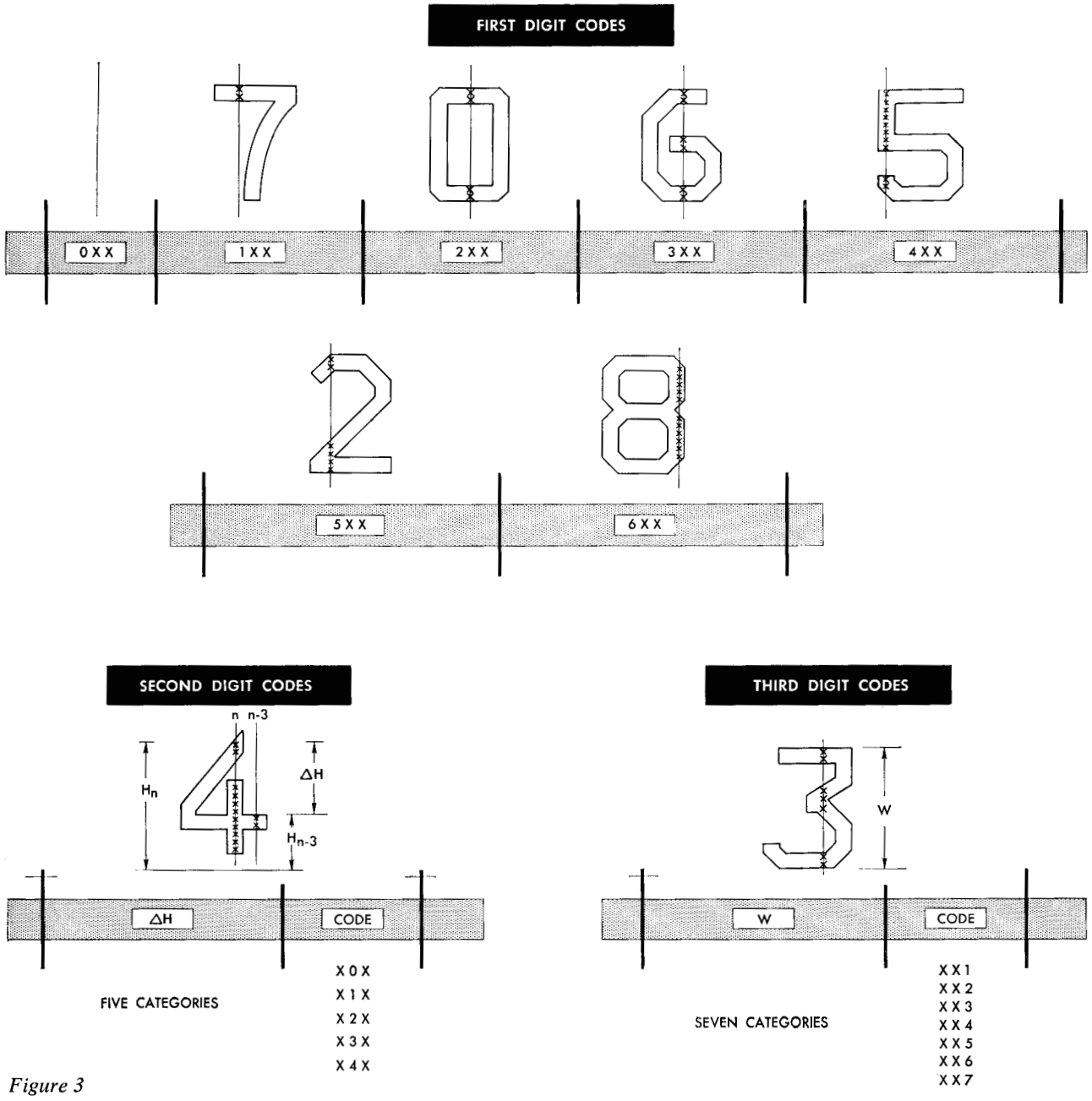


Figure 3
Illustration of coding.

The third digit of the three-digit code represents the distance between the upper and lower black areas on any scan. This code ranges from a one for the left edge of the "7" to a seven for the center of a "3" or "8." Examples of coding are illustrated in Figures 2 and 3.

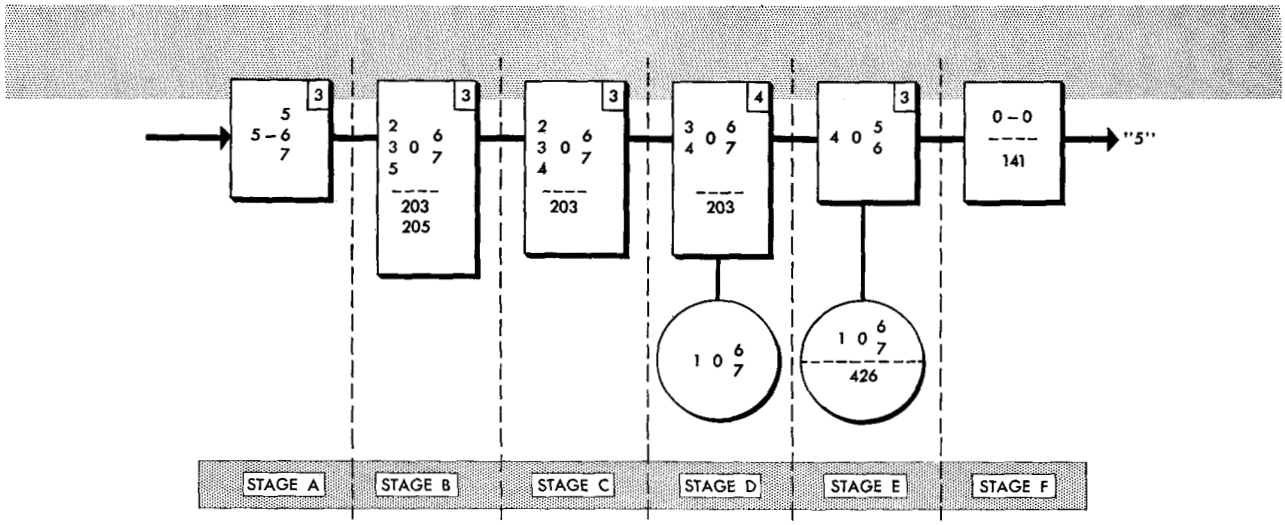
A program was written for the Type 650 which applied the coding definitions to the scan data obtained by the flying spot scanner. This coding program reads 15 cards of raw data for each character and generates a two-card set, each card containing 15 scan codes and identifying information at the rate of 80 characters per hour.

The permanent record of coded scan information contained in punched cards serves the following purposes:

(1) This record is used as input data for the simulation of logical sequences.

(2) A listing of this record is used by the sequence designer to define the logical sequences.

(3) Subsequent revisions of the logical sequences are made by trial and error. Reasons for conflicts and failures exhibited by the results are determined by referring to the listing of coded information.



ALLOWABLE CODE VALUES

505	203	203	203	405	000
506	205	206	306	406	010
507	206	207	307	AND NOT	020
515	207	306	406	106	030
516	306	307	407	107	040
517	307	406	AND NOT	426	141
525	506	407	106		
526	507		107		
527					
535					
536					
537					
545					
546					
547					

Figure 4
Recognition sequence for the "5".

Simulation of recognition sequences

As previously defined, prescribed sequences of coded scan information form the basic criteria for character recognition. The actual test sequence of coded scan information is compared to a prescribed sequence. On the basis of the comparison the scanned character is either accepted or rejected. The basic implementation (by means of delay units and inhibitors) to explore scanned data for a given sequence was discussed earlier. In the following paragraph is described a program which simulates this basic implementation on the Type 650.

The logical sequence consists of a chain of delay units with inhibitors at certain stages as illustrated in Figure 4. In this figure delay units are denoted by rectangles and inhibitors by circles. The various stages are denoted by capital letters.

The triggering or retriggering of a delay unit at an intermediate stage of a sequence requires the satisfaction of the following conditions:

(1) The coding for the scan under consideration must be identical to one of several allowable codes. The allowable codes for each stage of the sequence are shown in the rectangular enclosures representing delay units.
(2) The delay unit of the previous stage must be "on."
(3) The inhibitor (if it exists) of the given stage must be "off."

Condition (1) is necessary and sufficient for the triggering of a delay unit at the first stage since there are no previous delay units and no inhibitors at the first stage.

Each delay unit is represented in the Type 650 by four words of storage—a *recognition* word, a *delay entry* word, an *amount of delay* word and a *delay output* word:

Recognition Word: Normally zero. Whenever the scan coding matches the prescribed coding for any delay unit a (+1) is entered into the corresponding recognition word.

Delay Entry Word: Normally zero. When a delay unit is triggered, the number of the scan is placed in the delay entry word.

Amount of Delay Word: When a delay unit is triggered, it remains "on" for a given number of scans. This constant is placed in the amount of delay word at the start of the simulation. In Figure 4 the amount of delay is indicated in the corner of the rectangle representing each delay unit. Note that there is no amount of delay specified for the last delay unit in a sequence.

Delay Output Word: A zero denotes that the delay unit is "off," a (+1) denotes that the delay unit is "on."

The triggering of an inhibitor unit requires the satisfaction of the following conditions:

(1) The coding for the scan under consideration must be identical to one of the several inhibiting codes. The inhibiting codes are indicated in the circular enclosures representing inhibitor units.

(2) The delay unit of the previous stage must be "on."

When these conditions are satisfied the inhibitor is considered to be "on" and remains "on" until Condition (2) is no longer satisfied.

Each inhibitor is represented in the Type 650 by two words of storage—a *recognition* word and an *inhibitor output* word:

Recognition Word: Normally zero. Whenever the scan coding matches the prescribed coding for any inhibitor unit a (+1) is entered into the corresponding recognition word.

Inhibitor Output Word: A zero denotes that the inhibitor is "off," a (−1) denotes that the inhibitor is "on."

The following paragraphs describe the order of events which take place in the Type 650 after the program has been loaded into the machine.

Phase 1: The coded scan information for 600 scans (20 characters) is read into storage and then tested sequentially with no reference made to character separation. Thus the sequence formed by the final scans of one character and the initial scans of the succeeding character is tested against the prescribed sequence. Any extraneous recognitions due to this combination of events can be detected in the simulation.

Phase 2: All recognition words are set to zero. Then tests are performed to determine whether any of the delay units will be "on" for the present scan cycle. The contents of the delay entry and amount of delay words are added and the present scan number is subtracted from this sum. The sign of this result is then tested by means of a Branch on Minus Instruction. A non-negative result means the delay unit will be "on" for the present scan cycle and a (+1) is stored in the delay output word. A negative result indicates that both the delay unit and the inhibitor of the next stage (if it exists) should be "off." Zeros are stored in the delay output and inhibitor output words.

Phase 3: The coding is compared by means of Table Lookup Instructions to the specified code values of the various delay units and inhibitors stored in separate tables in the Type 650 drum storage. If a match is detected a (+1) is stored in the corresponding recognition word(s).

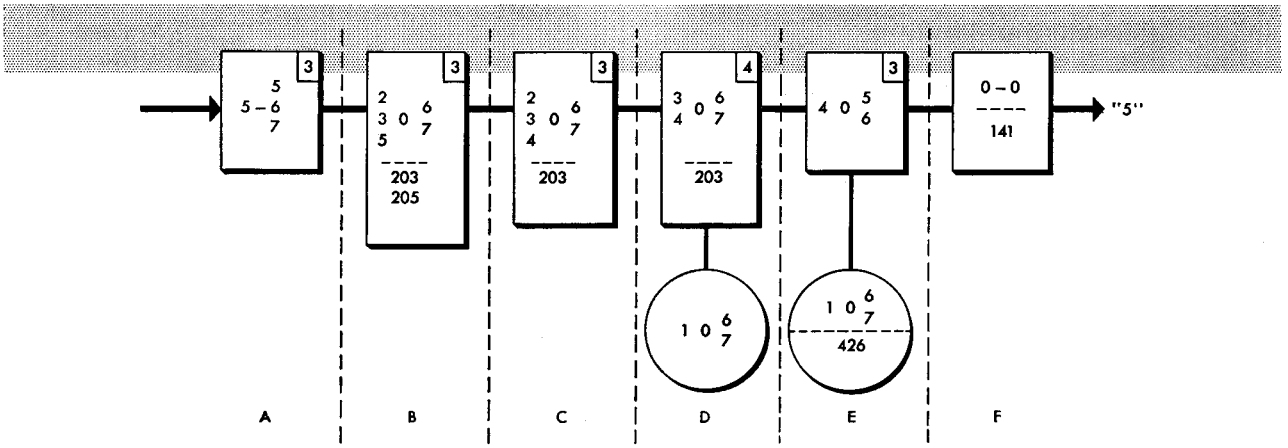
Phase 4: Tests are performed to determine whether the triggering conditions are satisfied for any of the inhibitor units. The contents of the delay output word of the previous delay unit are added to the contents of the given inhibitor recognition word and the constant (+2) is subtracted from this sum. A zero result indicates that the triggering conditions are satisfied and a (−1) is inserted into the given inhibitor output word. A non-zero result indicates that the triggering conditions are not satisfied and the inhibitor output word is not altered. The test for a zero result is made by means of the Branch on Non-Zero Instruction.

Phase 5: Tests are performed to determine whether the triggering conditions are satisfied for any delay unit. For an initial delay unit the corresponding recognition word is examined for a non-zero condition by means of the Branch on Non-Zero Instruction. If a non-zero condition is detected the present scan number is stored in the delay entry word, otherwise the contents of the delay entry word are not altered. For each intermediate delay unit, the contents of the delay output word of the previous delay unit, the inhibitor output word of the given stage (if it exists), and the recognition word of the given delay unit are added and the constant (+2) is subtracted from this sum. A zero result indicates that the triggering conditions are satisfied. In this case the present scan number is stored in the delay entry word of the given delay unit. In the event that the result is other than zero the contents of the corresponding delay entry word are not altered.

The tests for triggering of the final delay units are performed in the same fashion. However, if the result is zero (that is, the triggering conditions are satisfied, indicating a recognition), the 650 program automatically calls for the punching of a recognition card (discussed below), resets all inhibitor output words and delay entry words to zero and returns to Phase 2 to test the next coded scan. Should the result be non-zero (no recognition) the program returns to Phase 2 to test the next coded scan without altering the inhibitor output and delay entry words.

When all 600 scans have been processed the program prepares for a return to Phase 1. In order to present the data to the Type 650 in the same manner as it would be presented to the proposed character recognition machine, the information contained in the inhibitor output and delay entry words is preserved. The scan numbers in the delay entry words, however, are reduced by a factor of 600 to be compatible with the new scan number which will return to one. Hence, the Type 650 effectively processes a continuous flow of scanned data.

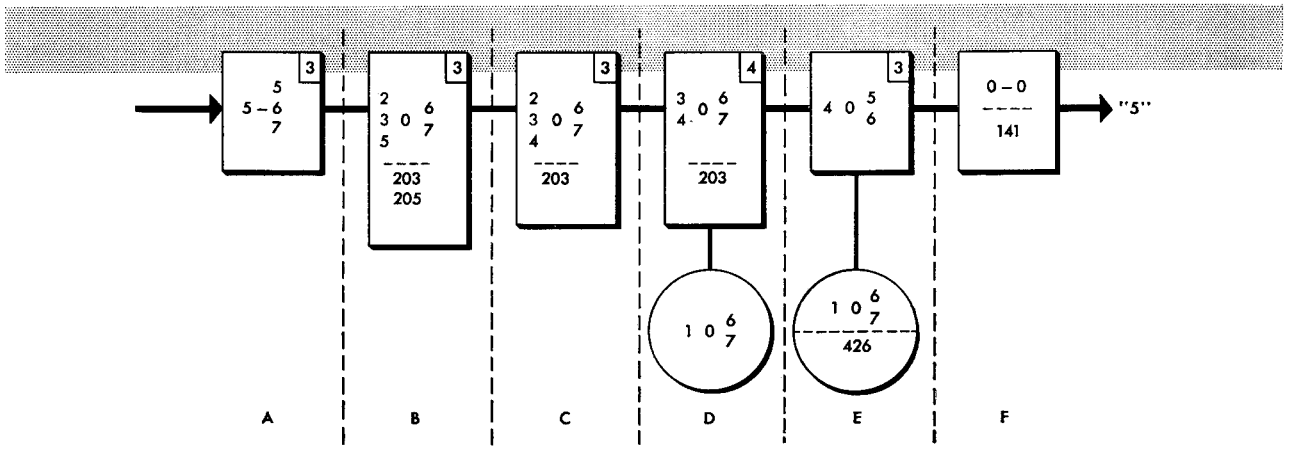
The application of this process to the coded scan information of a "5" and a "$" respectively are illustrated in Figures 5 and 6. The prescribed sequence is shown at the top of each figure and the coded scan information to be



SCAN NO.	SCAN CODE	PHASE 2 DELAY OUTPUT WORDS					PHASE 3 RECOGNITION WORDS: DELAY UNITS						PHASE 3 RECOGNITION WORDS: INHIBITORS		PHASE 4 INHIBITOR OUTPUT WORDS		PHASE 5 DELAY ENTRY WORDS					
		A	B	C	D	E	A	B	C	D	E	F	D	E	D	E	A	B	C	D	E	F
1	236	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	236	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	132	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	123	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	424	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
7	141	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
8	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
9	000	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
10	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
11	000	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
12	131	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	536	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0
14	133	+1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0
15	233	+1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0
16	306	+1	0	0	0	0	0	+1	+1	+1	0	0	0	0	0	0	13	16	0	0	0	0
17	215	0	+1	0	0	0	0	0	0	0	0	0	0	0	0	0	13	16	0	0	0	0
18	203	0	+1	0	0	0	0	+1	+1	+1	0	0	0	0	0	0	13	16	18	0	0	0
19	317	0	+1	+1	0	0	0	0	0	0	0	0	0	0	0	0	13	16	18	0	0	0
20	131	0	0	+1	0	0	0	0	0	0	0	0	0	0	0	0	13	16	18	0	0	0
21	307	0	0	+1	0	0	0	+1	+1	+1	0	0	0	0	0	0	13	16	18	21	0	0
22	207	0	0	0	+1	0	0	+1	+1	0	0	0	0	0	0	0	13	16	18	21	0	0
23	103	0	0	0	+1	0	0	0	0	0	0	0	0	0	0	0	13	16	18	21	0	0
24	406	0	0	0	+1	0	0	0	+1	+1	+1	0	0	0	0	0	13	16	18	21	24	0
25	406	0	0	0	+1	+1	0	0	+1	+1	+1	0	0	0	0	0	13	16	18	21	25	0
26	040	0	0	0	0	(+1)	0	0	0	0	0	(+1)	0	0	0	0	13	16	18	21	25	(26)
27	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
28	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
29	000	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
30	000	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0

Figure 5

Simulated recognition of a "5" on the IBM Type 650.



		PHASE 2					PHASE 3								PHASE 4		PHASE 5					
SCAN NO.	SCAN CODE	DELAY OUTPUT WORDS					RECOGNITION WORDS								INHIBITOR OUTPUT WORDS		DELAY ENTRY WORDS					
							DELAY UNITS						INHIBITORS									
		A	B	C	D	E	A	B	C	D	E	F	D	E	D	E	A	B	C	D	E	F
1	133	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	134	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	335	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	315	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	305	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	141	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
7	141	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
8	141	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
9	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
10	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
11	131	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	000	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	0
13	236	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	506	0	0	0	0	0	+1	+1	0	0	0	0	0	0	0	0	14	0	0	0	0	0
15	536	+1	0	0	0	0	+1	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0
16	506	+1	0	0	0	0	+1	+1	0	0	0	0	0	0	0	0	16	16	0	0	0	0
17	307	+1	+1	0	0	0	0	+1	+1	+1	0	0	0	0	0	0	16	17	17	0	0	0
18	307	+1	+1	+1	0	0	0	+1	+1	+1	0	0	0	0	0	0	16	18	18	18	0	0
19	107	+1	+1	+1	+1	0	0	0	0	0	0	0	+1	+1	−1	−1	16	18	18	18	0	0
20	107	0	+1	+1	+1	0	0	0	0	0	0	0	+1	+1	−1	−1	16	18	18	18	0	0
21	307	0	+1	+1	+1	0	0	+1	+1	+1	0	0	0	0	−1	−1	16	18	21	18	0	0
22	307	0	0	+1	+1	0	0	+1	+1	+1	0	0	0	0	−1	−1	16	18	21	18	0	0
23	307	0	0	+1	0	0	0	+1	+1	+1	0	0	0	0	−1	0	16	18	21	18	0	0
24	406	0	0	+1	0	0	0	0	+1	+1	+1	0	0	0	−1	0	16	18	21	18	0	0
25	406	0	0	0	0	0	0	0	+1	+1	+1	0	0	0	0	0	16	18	21	18	0	0
26	405	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	0	16	18	21	18	0	0
27	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	16	18	21	18	0	0
28	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	16	18	21	18	0	0
29	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	16	18	21	18	0	0
30	040	0	0	0	0	0	0	0	0	0	0	+1	0	0	0	0	16	18	21	18	0	0

Figure 6
Rejection of a "$" by the recognition sequence for a "5".

considered is listed in column two. The contents of the various delay output, recognition, inhibitor output, and delay entry words for Phases 2 through 5 scan by scan are shown in the columns so designated in the figures.

In Figure 5 the sequence of coded scan information for a "5" is effectively compared to the prescribed sequence and a character recognition occurs at scan 26. At this scan the E delay output word is "on" and the scan code is one of the allowable codes of the F delay unit. This is all that is required for triggering delay unit F since there is no inhibitor in this stage.

When delay unit F is triggered, a recognition card is punched containing the identification of the character being tested, the prescribed sequence against which the character was tested and the scan number at which recognition was made.

Figure 6 depicts the manner in which inhibitors can prevent a "$" from being recognized as a "5." At scan 19 delay units A through D are "on" and it appears as though a character recognition is imminent. However, since the code value of scan 19 is one of the values of the inhibitors in Stages D and E, both inhibitors are effectively turned "on" by storing (−1) in the corresponding inhibitor output words. Thus the character recognition is prevented. Comparison of the "5" and the "$" indicates that, except for the center vertical line and the curvature in the upper left corner, the "5" and the "$" are quite similar in physical appearance. The prescribed sequence uses these two distinguishing characteristics to prevent a "$" from being recognized as a "5." The 106 and 107 codes, characteristic of the center vertical line, are used for inhibition in Stages D and E, while the 426 code which is descriptive of the curvature in the upper left corner is used for inhibition in Stage E.

Table 1 **Interpretative Operation Codes**

Operation Code	*Explanation of Operations*
01	Sets up Delay Unit A
02	Sets up Delay Unit B
03	Sets up Delay Unit C
04	Sets up Delay Unit D
05	Sets up Delay Unit E
06	Sets up Delay Unit F
07	Sets up Delay Unit G
08	Sets up Delay Unit H
09	Sets up an Inhibitor in Stage B
10	Sets up an Inhibitor in Stage C
11	Sets up an Inhibitor in Stage D
12	Sets up an Inhibitor in Stage E
13	Sets up an Inhibitor in Stage F
14	Sets up an Inhibitor in Stage G
15	Sets up an Inhibitor in Stage H

The interpretative instructions consist of a two-digit operation code, a four-digit "X" address, and a four-digit "Y" address.

In using operation codes 01 through 08 which set up delay units, the X address specifies the initial address of the 650 table where the allowable code values associated with the respective delay unit are stored. The Y address specifies the amount of delay of the delay unit. It is set to zero for the last delay unit.

In using operation codes 09 through 15 which set up inhibitors, the X address again specifies the initial address of the 650 table where the inhibiting code values associated with the respective inhibitors are stored. The Y address is always set to zero.

Up to this point we have discussed a Type 650 program in terms of one recognition sequence. The program is general in that it allows for the simulation of any sequence of eight stages or less. Inhibitors can be inserted at any intermediate or final stage. Interpretative instructions are used in setting up the simulation of a given logical sequence. These instructions consist of a two-digit operation code, a four-digit "X" address and a four-digit "Y" address. A list of the possible operation codes is shown in Table 1. The interpretative instructions used to set up the logical sequence of Figure 4 are shown in Table 2. These instructions are read by the Type 650. Upon interpretation the general program is altered to simulate the given logical sequence.

It is noted in Table 1 that the "X" addresses of the interpretative instructions specify the initial addresses of the Type 650 tables where the allowable code values associated with the various delay units and inhibitors are stored. The allowable code values must be stored in the allotted Type 650 tables prior to processing. This is accomplished by first listing the code values for each delay unit and inhibitor in ascending order as shown in Figure 4. Then these values are punched into load cards, 7 code values per card, along with the initial address of the Type 650 table to which they are assigned. These load cards are read by the Type 650 and, under direction of the program, the code values are stored in the tables assigned.

Results and conclusions

As described herein, the simulation consisted of the collection of basic scan data, the reduction of this data by the application of coding definitions and the testing of logical sequences against the coded data. The coding of raw data proceeded at the rate of 80 characters per hour. Each pass of the 1500 sample characters against one recognition sequence took between 1.5 and 2 hours, depending on the complexity of the sequence. The Type 650 affords a highly effective simulative tool; the logical elements of the proposed recognition machine are easily accommodated by the logical instructions available to the programmer. Each modification in the main program to accommodate a modified sequence was achieved in less than 1 minute through the use of the interpretative instructions. Furthermore, results are produced at a rate easily matching the requirements of a staff of sequence designers. Computers will assume even more of the design responsibilities in future simulations of this type, and prospects for generating and modifying recognition sequences entirely automatically appear quite promising.

Caution should be exercised in predicting the ultimate

Table 2 **Interpretative Instructions for the "5" Sequence**

Operation Code	*X Address*	*Y Address*	*Description*
01	1400	0003	Set up delay unit A. Allowable code values are stored in table starting with location 1400. Amount of delay is 3 scans.
02	1450	0003	Set up delay unit B. Allowable code values are stored in table starting with location 1450. Amount of delay is 3 scans.
03	1500	0003	Set up delay unit C. Allowable code values are stored in table starting with location 1500. Amount of delay is 3 scans.
04	1550	0004	Set up delay unit D. Allowable code values are stored in table starting with location 1550. Amount of delay is 4 scans.
05	1600	0003	Set up delay unit E. Allowable code values are stored in table starting with location 1600. Amount of delay is 3 scans.
06	1650	0000	Set up delay unit F. Allowable code values are stored in table starting with location 1650. This is the last delay unit in the sequence.
11	1700	0000	Set up an inhibitor in stage D. Inhibiting code values are stored in table starting with location 1700.
12	1750	0000	Set up an inhibitor in stage E. Inhibiting code values are stored in table starting with location 1750.

performance of a machine on the basis of the results described herein since performance is obviously dependent on the quality distribution of the specimens. To insure an adequate distribution the inked ribbons used to produce these specimens were operated for 150% of nominal life, and identical samples were taken at equal intervals throughout this period. In addition, three standard types of paper were used: white, clear manila, and a manila with small dark fibers which tended to introduce random spots in the character background.

Two sets of basic input data were used, each containing 100 or more specimens of every numeral and symbol. The first set was used to develop the recognition sequences by repetitive testing and modification. A total of fifty-four runs, or an average of almost four runs per sequence, were required to arrive at "final" criteria for all of the characters. An additional pass was then made with the first set of data against these fourteen final sequences. At this point only 26 of the specimens in the set could not be recognized with certainty. All of the uncertainties occurred on samples that had been produced after 90% of nominal ribbon life, and rejections were three times as frequent among the samples printed on the dark fiber paper. Perhaps of even greater importance is the fact that none of the specimens were mis-recognized (where one character is substituted for another). As a matter of fact, only four substitutional errors occurred during all of the fifty-four runs with unperfected sequences. The second set of specimens was used to check the final sequences. In this set only 5 of the specimens could not be recognized and again there were no substitutional errors.

As the development of recognition criteria progressed, limits had to be established on the degree of degradation that would be tolerated in each character. If these limits were too loose, conflicts with wrong characters would be likely to occur, and if the limits were too severe, many reasonable characters would be rejected. To properly set the tolerances a quantitative measure of character quality was defined. Each of the quantized print-outs was compared with an ideal outline for that character and a count was made of all the black cells that should have appeared white and all the white cells that should have appeared black. This total "wrong cell" count was then expressed as a percentage of the total number of black cells in the ideal character. One hundred percent minus this quantity was taken to be the "percent quality" of the character. Percent qualities were determined for all specimens. In the final sequences none of the characters with quality greater than 65% failed. Figure 7 (overleaf) contains twenty examples of less than ideal characters that were recognized, with a very heavy and very light specimen shown for each.

Although the results do not give a precise measure of the ultimate performance of the proposed machine, they do indicate that a very practical machine of reasonable size can be built incorporating the proportional parts system of recognition logic. The detailed criteria that have been designed offer a good start towards the final development of such a machine.

Received June 12, 1956

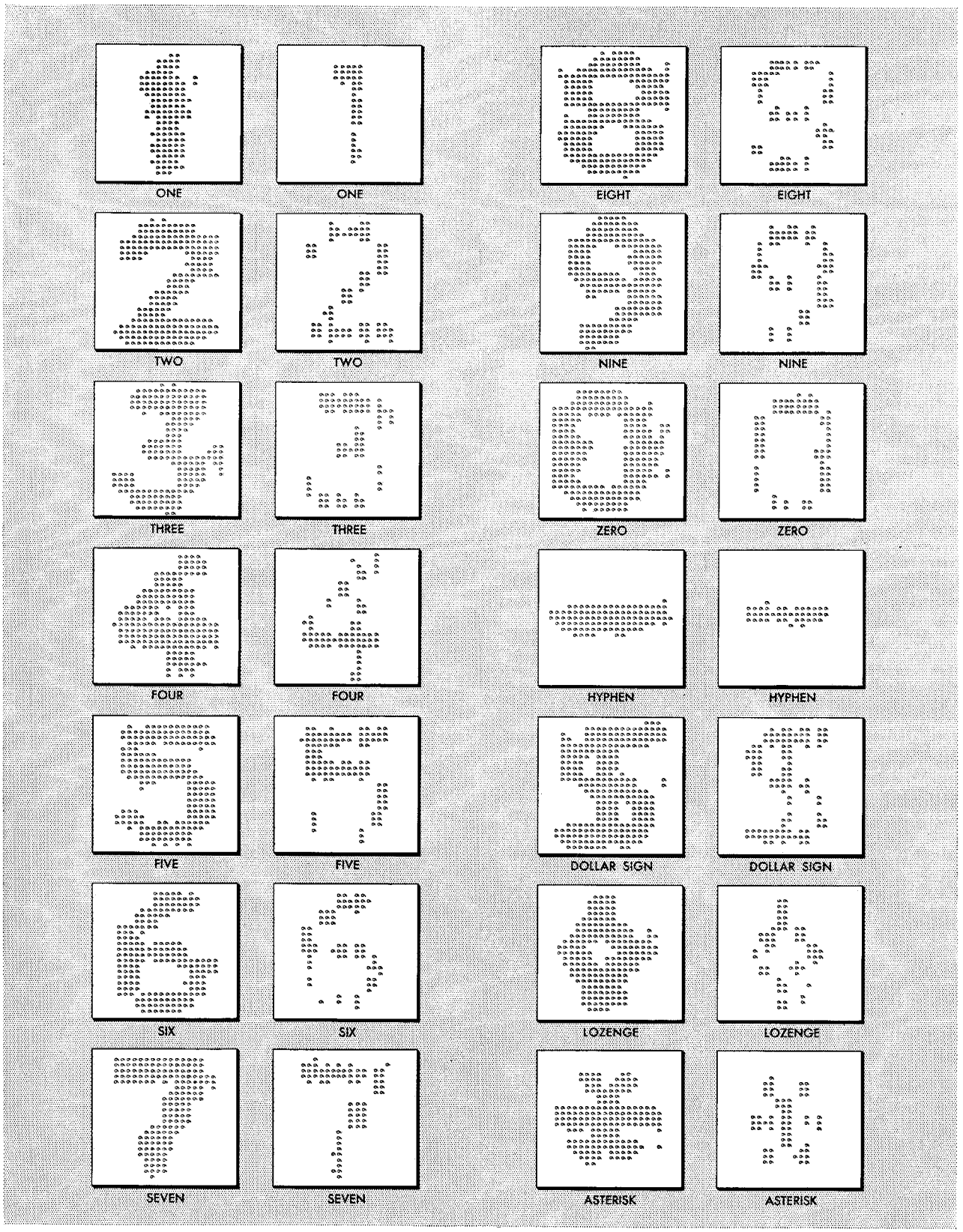


Figure 7

Examples of characters recognized by the proportional parts logic.