Nov. 2, 1965 L. W. MARSH, JR 3,215,985

CONTROL SYSTEM FOR HIGH SPEED PRINTERS

Filed March 8, 1962 5 Sheets-Sheet 1

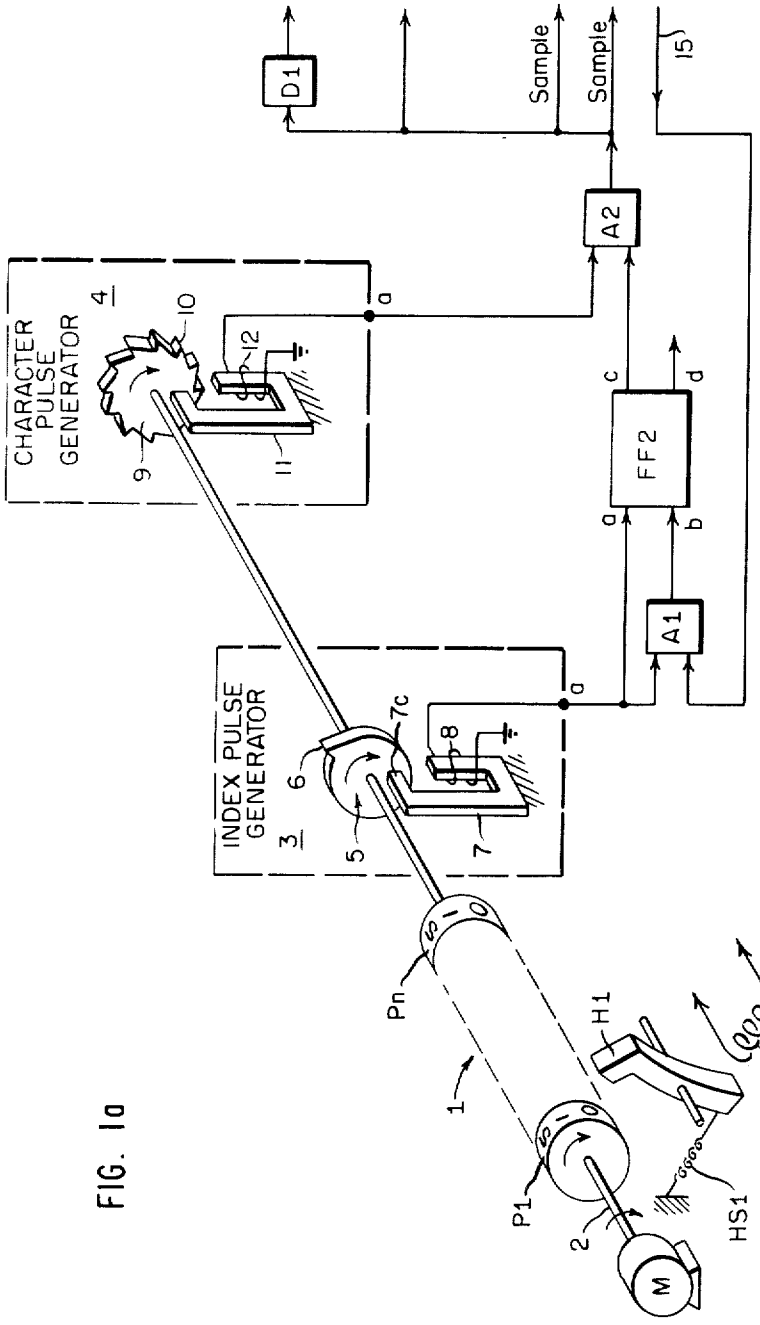


FIG. 1a

INVENTOR.
LYNN W. MARSH
BY
Kenway, Jenney & Hildreth
ATTORNEYS

Nov. 2, 1965 L. W. MARSH, JR 3,215,985

CONTROL SYSTEM FOR HIGH SPEED PRINTERS

Filed March 8, 1962 5 Sheets-Sheet 2

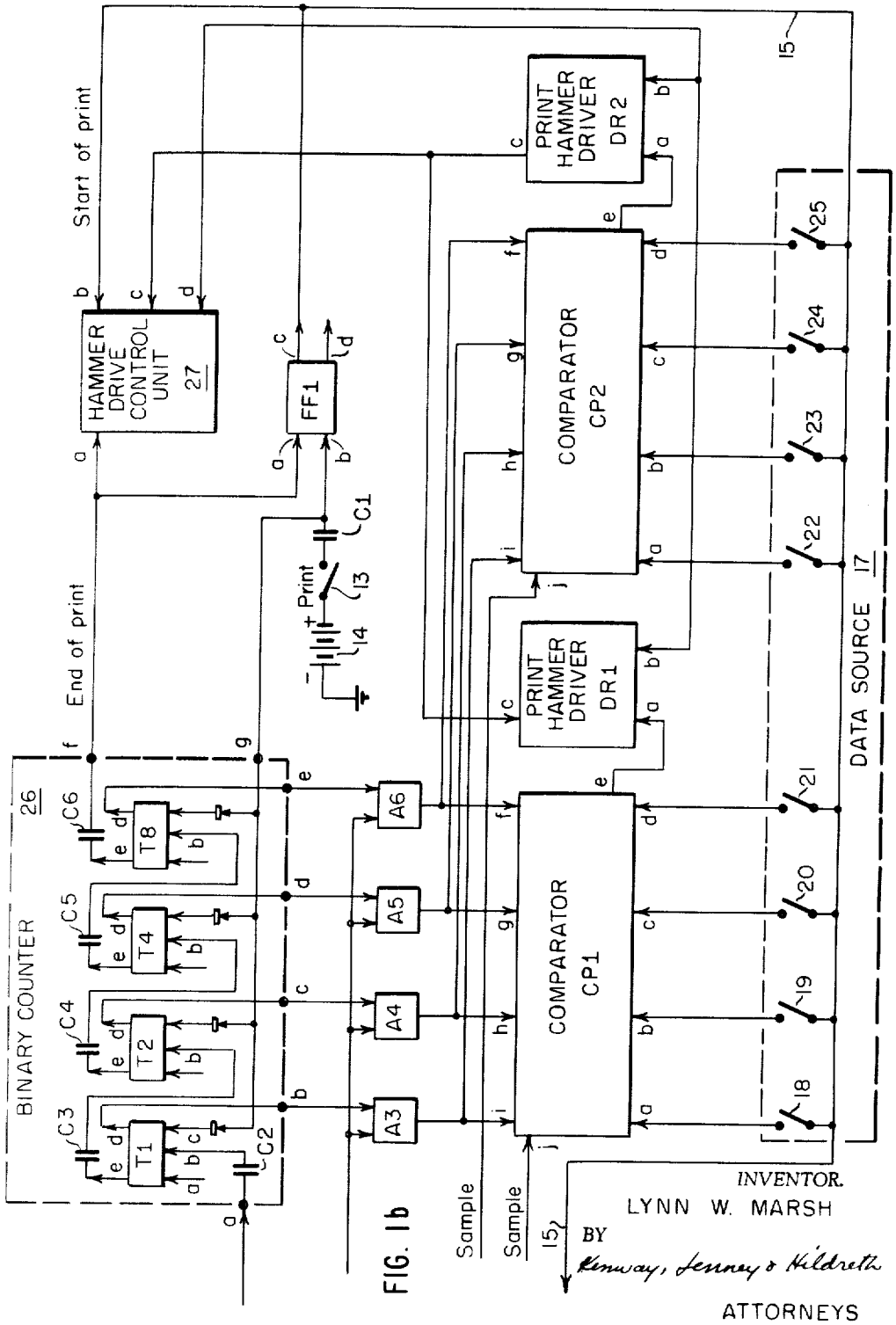


FIG. 1b

INVENTOR.
LYNN W. MARSH
BY
Kenway, Jenney & Hildreth
ATTORNEYS

Nov. 2, 1965 L. W. MARSH, JR 3,215,985

CONTROL SYSTEM FOR HIGH SPEED PRINTERS

Filed March 8, 1962 5 Sheets-Sheet 3

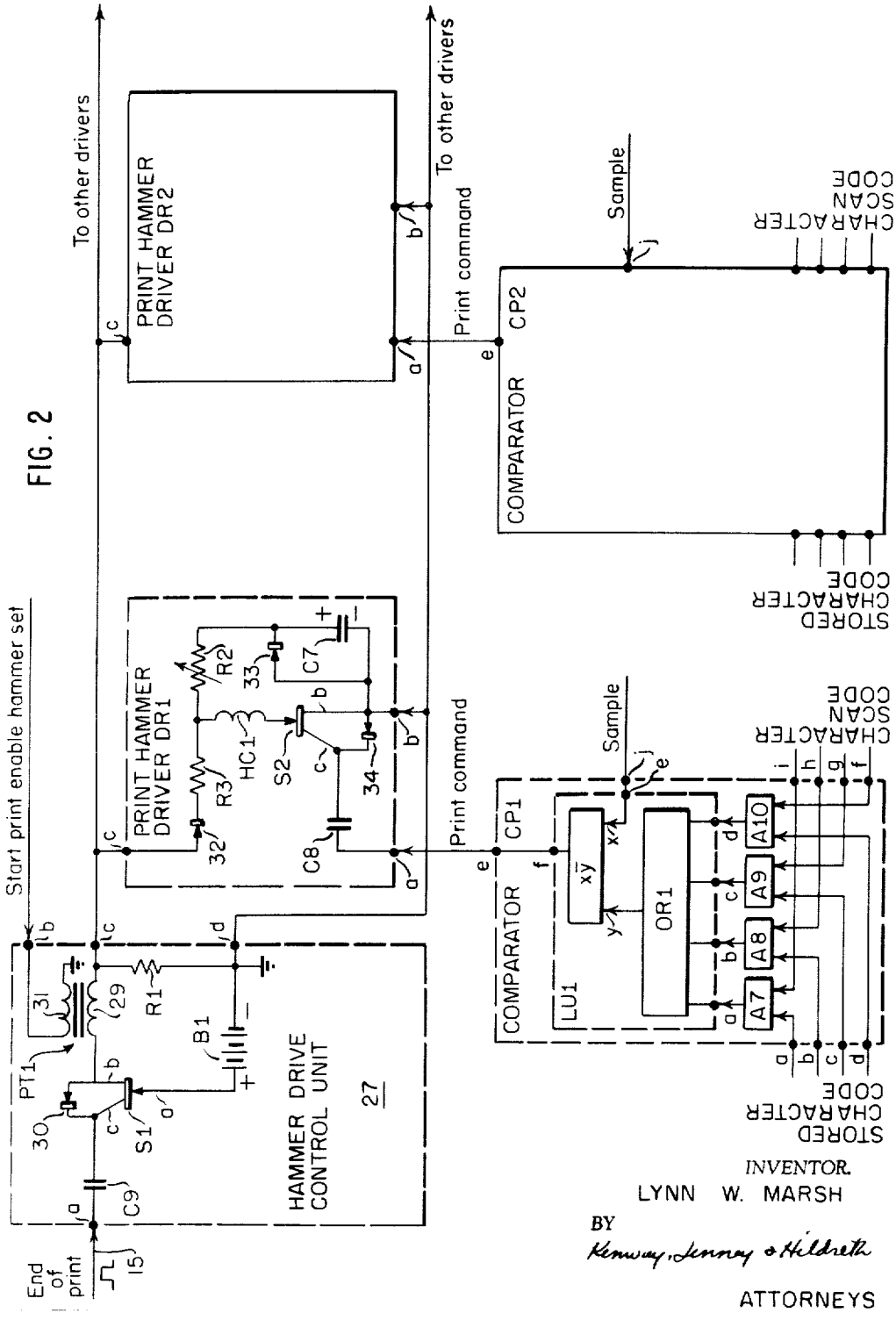


FIG. 2

INVENTOR.
LYNN W. MARSH

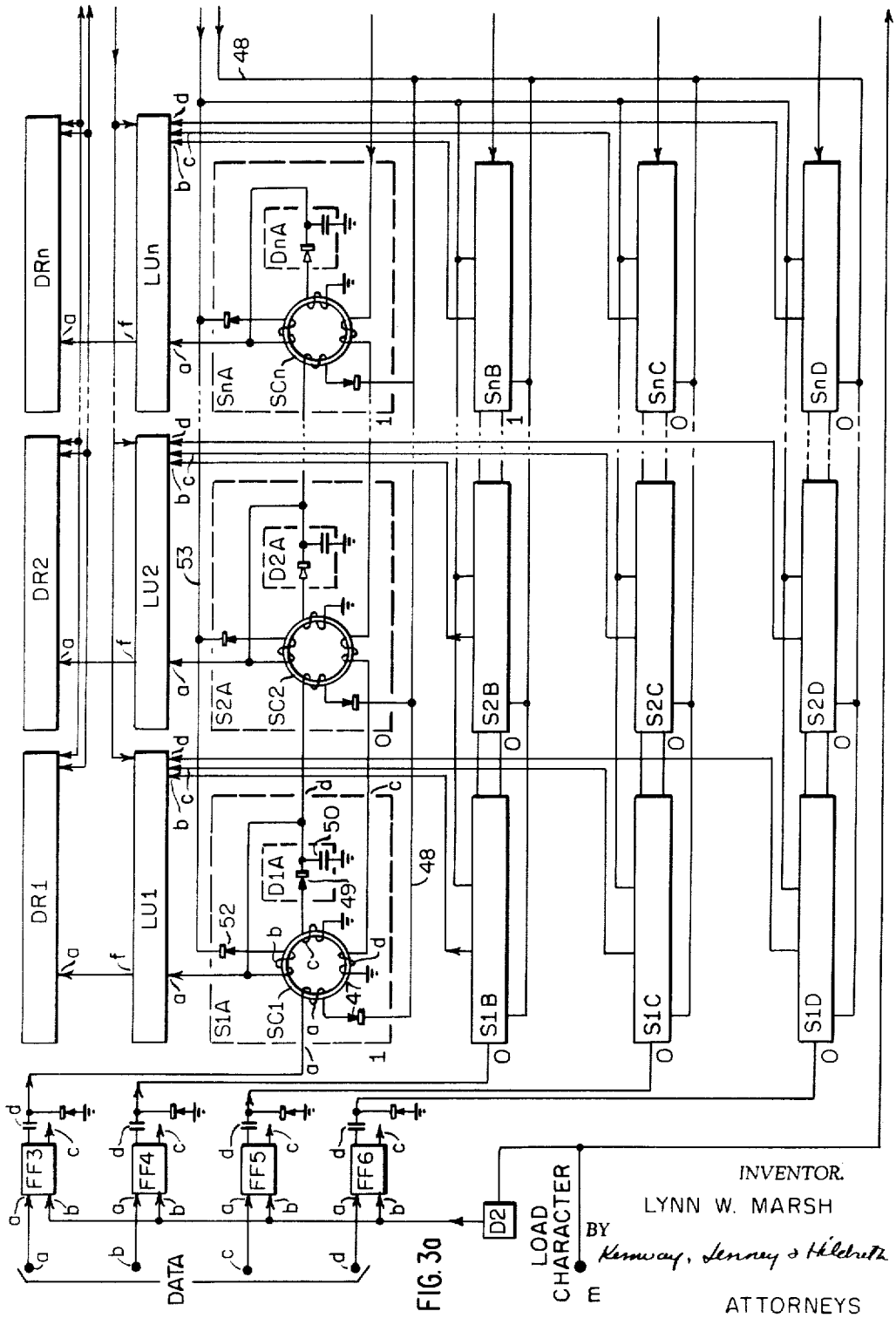
BY
Kenway, Jenney & Hildreth

ATTORNEYS

Nov. 2, 1965 L. W. MARSH, JR 3,215,985

CONTROL SYSTEM FOR HIGH SPEED PRINTERS

Filed March 8, 1962 5 Sheets-Sheet 4



INVENTOR.
LYNN W. MARSH
BY Kenway, Jenney & Hildreth
ATTORNEYS

Nov. 2, 1965 L. W. MARSH, JR 3,215,985

CONTROL SYSTEM FOR HIGH SPEED PRINTERS

Filed March 8, 1962 5 Sheets-Sheet 5

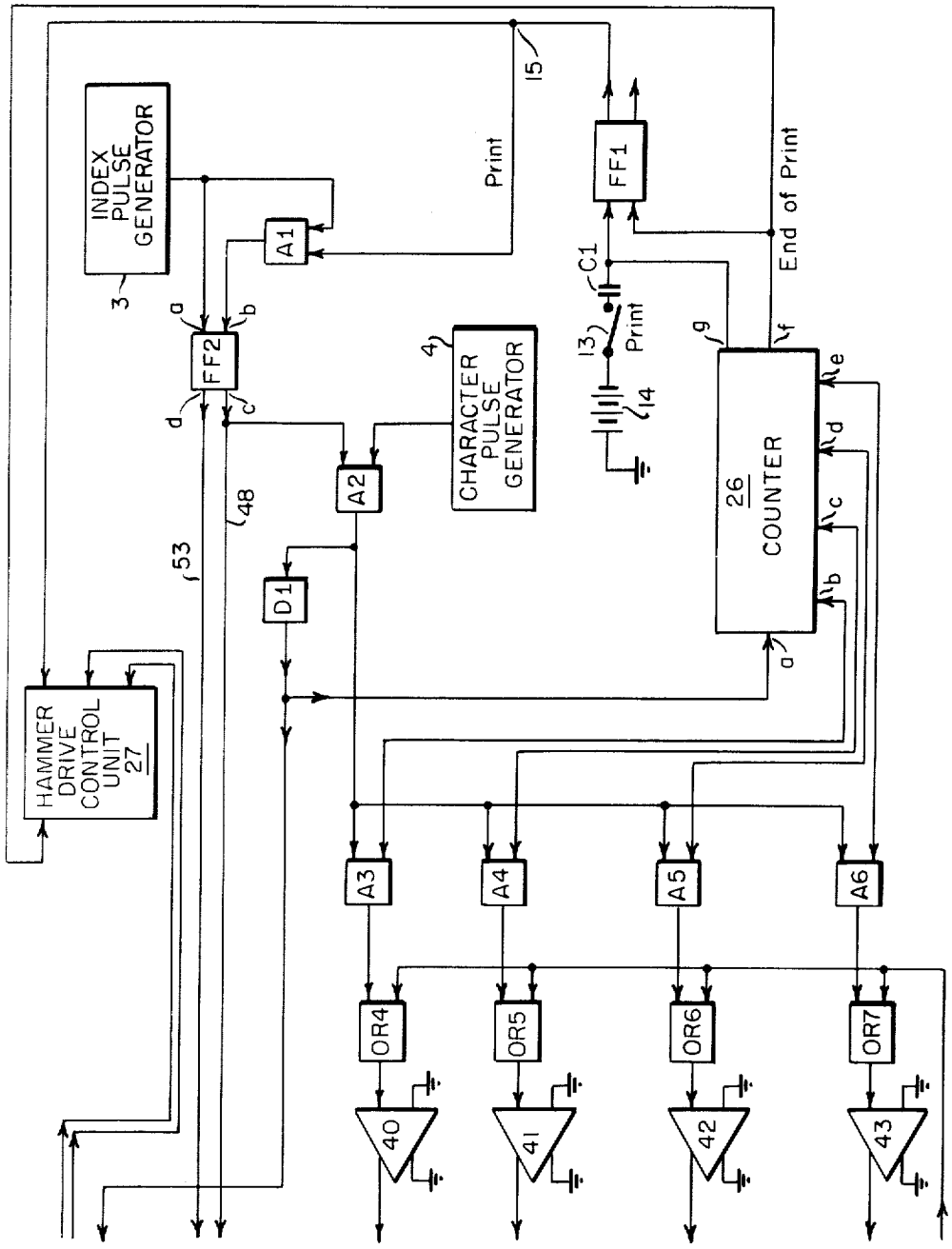


FIG. 3b

INVENTOR.
LYNN W. MARSH
BY
Kenway, Jenney & Hildreth
ATTORNEYS

United States Patent Office 3,215,985 Patented Nov. 2, 1965

3,215,985

CONTROL SYSTEM FOR HIGH SPEED PRINTERS

Lynn W. Marsh, Jr., Marblehead, Mass., assignor to Anelex Corporation, Boston, Mass., a corporation of New Hampshire

Filed Mar. 8, 1962, Ser. No. 178,445
5 Claims. (Cl. 340—172.5)

My invention relates to high speed printers, and particularly to an improved system for actuating the print hammers of a high speed printer in response to data representing the characters of a line to be printed.

Various high speed printers have been devised for printing a line of characters stored in the form of a selected code. For example, one widely used printer comprises a series of constantly rotating print rolls, one for each column in the line to be printed. Each print roll has a font of characters formed on its periphery, and each cooperates with a print hammer which, when actuated, strikes a record sheet and a transfer sheet against the print roll to record the character then in position on the record sheet. In order to actuate each hammer when the character in printing position on the associated print roll corresponds to the stored character, it is necessary to compare each character, as it comes into position, with the stored character. United States Letters Patent No. 2,805,620, issued September 10, 1957 to Leo Rosen, Howard C. Barlow and Ray L. Bowman for Control Means For High Speed Printing Apparatus, for example, shows various ways in which such a comparison may be made. Prior to my invention, the apparatus required to make comparisons for each character has been extremely complex, particularly in printers in which as many as one hundred and twenty columns per line are provided. A primary object of my invention is to reduce the amount and complexity of the apparatus required to make the character comparisons in a high speed printer.

Briefly, the control system of my invention comprises a novel ordered sequence comparator in which a code generated in a predetermined sequence is compared with the code representing each character for a line to be printed, not bit-by-bit as in former comparators, but in terms of predetermined code sets, such that each generated code sequence is registered as the same as any character code in the same set. Ambiguity is prevented by selecting the order in which the comparison sequence is generated, and by the provision of a novel final detection circuit capable of responding only once during each comparison sequence. As will appear, the control system of my invention may be adapted for use either in systems in which input character codes are available throughout the printing of a line, or in systems in which character codes are presented only transiently, one after another until a complete line has been stored.

My invention will best be understood by reference to the accompanying drawings, together with the following detailed description.

In the drawings,

FIGS. 1*a* and 1*b*, when arranged horizontally side by side with FIG. 1*a* at the left, comprise a schematic wiring diagram of a print control system, for high speed printers in which data representing a line of characters is available throughout the printing of the line, in accordance with one embodiment of my invention;

FIG. 2 is a schematic wiring diagram showing the internal details of the comparators, print hammer drivers, and the hammer drive control unit shown in block diagram form in FIG 1; and

FIGS. 3*a* and 3*b*, when arranged horizontally side by side with FIG. 3*a* at the left, comprise a schematic wiring diagram of a print control system, for printers in which the data for each character to be printed is available

transiently, in accordance with a second embodiment of my invention.

In the drawings, certain conventions have been adopted to facilitate a concise explanation of my invention. Conventional elements, such as logical gates, bistable multivibrators, and delay lines, have been shown in block diagram form. Units requiring detailed description, which are duplicated in the system, are shown in circuit detail at least once, within a dotted rectangle arbitrarily provided with external reference terminals, and similarly designated blocks having the same reference terminals elsewhere in the system are to be assumed to be of the same internal construction. The various components of the system are arranged on the drawings in an order selected to illustrate their functions in the system; in practice, the components would be arranged in the manner dictated by conventional wiring practice.

Referring now to FIG. 1, the first embodiment of my invention to be described is shown adapted to control a conventional printer of the type comprising a constantly rotating array of print wheels **1**, mounted on a shaft **2** connected to the output shaft of a constant speed motor M for rotation in the sense shown by the curved arrows. As schematically indicated, any selected number of print wheels **1** may be provided, each corresponding to a character column in a line to be printed. Thus, the first print wheel P1 may be be separated from the last print wheel P*n* by any selected number of intermediate print wheels. Each print wheel such as P1 is engraved or otherwise formed with indicia comprising a character font around its periphery, and is associated with a print hammer such as H1, which is pivoted from a suitable support, not shown, and restrained by a suitable spring such as HS1. In response to the energization of a hammer coil such as HC1, provided for each print hammer, the print hammer such as H1 strikes against a record sheet and a transfer ribbon, not shown, which are fed between the hammers and the print wheels, to record the character then adjacent the hammer on the record sheet. Since this apparatus is well known in the art, it will not be described in detail; a fuller description may be found in U.S. Patent No. 2,805,620, referred to above. It should be noted in this regard that the details of the printer do not form a part of, nor are they essential to, my invention; it is only necessary in the illustrated embodiment that the characters in a font for each column be available for printing in a fixed time sequence, and any printer in which such a sequence is enforced may readily be adapted to the apparatus of my invention.

In the illustrated embodiment, the position of the characters on the print wheels **1** is registered by an index pulse generator **3** and a character pulse generator **4**, both controlled by the print wheel drive shaft **2**. The index pulse generator **3** comprises a ferromagnetic disc **5** mounted on the shaft **2** and provided with a single projecting tooth **6**. The tooth **6** cooperates with a magnetic pickoff comprising a ferromagnetic frame **7** mounted on a suitable support, not shown. As schematically indicated, the frame comprises part of a magnetic path which includes the body of the disc and two air gaps, one a constant gap between the end **7***a* of the frame **7** and the body of the disc **5**, and a variable gap between the other end of the frame and the periphery of the disc **5**. Flux may be provided in this path by magnetically polarizing the tooth **6** with respect to the body of the disc **5**, by magnetizing the frame **7**, or both. An index pulse is induced in a coil **8** wound on the frame **7** as the tooth **6** passes the frame, thus momentarily reducing the second air gap. The location of the tooth **6** on the periphery of the disc **5** with respect to the characters on the print wheels **1** and the frame **7** is such that an index pulse is generated

just before a predetermined first character in each font passes before the associated print hammer, so that the index pulse indicates the beginning of a scan of the characters on the print wheels.

The character pulse generator **4** may be similar to the index pulse generator, except that it is arranged to emit a pulse just before each character arrives in printing position. As shown, it may comprise a ferromagnetic disc **9** provided with a plurality of projecting teeth such as **10**, the disc being fixed on the shaft **2** for rotation therewith and adapted to cooperate with a relatively fixed magnetic frame **11** on which a pick-off coil **12** is wound. As described for the index pulse generator, the teeth **10**, the frame **11**, or both, are magnetically polarized. The relative arrangement of the parts is such that, just before each character arrives in printing position, a character pulse is induced in the coil **12**. The reason for anticipating the arrival of the characters by a small amount is that time must be allowed for the electrical system, to be described, to function, and for the hammer to move the short distance necessary to print the character as it arrives in position.

By conventional apparatus, which it is unnecessary to describe in detail, the record sheet and transfer ribbon are stepped forward after the printing of each line, and during this period data for a new line is supplied. After the record sheet is in position and the new data has been stored, a print cycle is initiated in which the stored data is decoded and the corresponding characters are printed. Since the apparatus of my invention is involved only in the control of the print hammers during the print cycle, the apparatus for initiating this cycle has been schematically shown as a manually operable switch **13**, closed when it is desired to print a line of characters to supply a print pulse to the system from a suitable power supply, here schematically shown as a battery **14**.

When the switch **13** is closed, a positive pulse from the battery **14** is coupled through a suitable capacitor **C1** to an input terminal *b* of a conventional bistable multivibrator, or flip-flop, **FF1**. This flip-flop, as well as other flip-flops to be described, may be of any suitable known construction having complementary output terminals set to a first state in response to a pulse applied to one of two complementary input terminals and to an opposite state in response to a pulse applied to the other input terminal. A suitable circuit is shown, for example, in Fig. 198 on page 203 of TM11–690, Basic Theory and Application of Transistors, published in March, 1959 by Headquarters, Department of the Army. The output terminals *c* and *d* of the flip-flop **FF1** may be connected to the collectors of the transistors **Q1** and **Q2** in the publication. The bias voltages will be assumed to be such that in one state of the flip-flop **FF1**, its output terminals *c* and *d* are at ground and a positive voltage, respectively, and in the other state, at the positive potential and ground, respectively.

The other input terminal *a* of the flip-flop **FF1** is connected to an output terminal *f* of a binary counter **26**, to be described. The connections are such that when the switch **13** is closed, the pulse applied to the input terminal *b* of the flip-flop **FF1** drives it to a logical 1 state in which its output terminal *c* is positive. After completion of the printing operation, the binary counter **26** applies an "end-of-print" pulse to the input terminal *a* of the flip-flop **FF1**, to reset it to a logical 0 state in which the terminal *c* is at ground potential.

When the flip-flop **FF1** is in its logical 1 state, a positive voltage level from its output terminal *c* is applied over the lead **15** to energize a data source schematically indicated at **17**. The data source **17** may be any conventional storage unit, provided with gating means for entering and extracting character data in any selected digital code. As here schematically shown, however, it

may comprise a series of groups of switches, such as the group comprising the switches **18**, **19**, **20** and **21**, which may be manually set in a binary code sequence corresponding to a selected character to be printed, with the lowest ordered bit represented by the switch **18**; thus, the sequence 1101 would be represented by **18** closed, **19** open, and **20** and **21** closed. One such group of switches is provided for each character in the line, and as many characters as are desired may be provided for; for simplicity, however, I have shown apparatus for handling only two characters; the switches **22**, **23**, **24** and **25** provide storage for the second character.

During the print cycle, when the flip-flop **FF1** is in its logical 1 state, data stored in the data source **17** is applied to comparators, one for each character to be printed. Thus, the character code represented by the state of the switches **18**, **19**, **20** and **21** is applied to the input terminals *a*, *b*, *c* and *d*, respectively, of a comparator **CP1**, and the character code represented by the state of the switches **22**, **23**, **24** and **25** is applied to the input terminals *a*, *b*, *c* and *d* of a comparator **CP2**. The internal construction of these comparators will be described in detail below.

The output terminal *c* of the flip-flop **FF1** is also connected over the lead **15** to one input terminal of a conventional AND gate **A1**. This AND gate, as well as other AND gates to be described, may be of any conventional construction, and, for example, may be of the type shown in FIG. 208A on page 213 of the above cited TM11–690. For convenience, it will be assumed that a positive input voltage provides a positive output pulse in this gate, so that an n-p-n transistor, with appropriate bias potentials, would be employed instead of the p-n-p type shown in the publication.

The output of the AND gate **A1** is connected to one input terminal *a* of a second flip-flop **FF2**, which may be of the same construction as the flip-flop **FF1**. The output terminal *a* of the index pulse generator **3** is connected to the other input terminal of the AND gate **A1** and to the input terminal *b* of the flip-flop **FF2**. The output terminal *c* of the flip-flop **FF2** is connected to one input terminal of an AND gate **A2**, and the output terminal *a* of the character pulse generator **4** is connected to the other input terminal of the AND gate **A2**. During the paper feed and character loading portion of the printer cycle, when the flip-flop **FF1** is in its 0 state, the shaft **2** makes one or more complete revolutions, and the index pulse generator **3** emits at least one index pulse. This pulse will set the flip-flop **FF2** to a state in which its output terminal *c* will be at a potential, representing a 0 logical level, which will not permit the AND gate **A2** to pass character pulses emitted by the character pulse generator **4**. Repeated index pulses will not affect the state of the flip-flop **FF2**, so that the AND gate **A2** will remain cut off until the print cycle is initiated by the closing of the switch **13**.

When the switch **13** is closed, the flip-flop **FF1** goes to its 1 state and the AND gate **A1** is enabled to pass the next index pulse to set the flip-flop **FF2** to its opposite state; the concurrent application of the index pulse to the flip-flop on the opposite terminal will have no effect, as is known in the art and will be apparent from a consideration of the circuit detailed in Fig. 198 of TM11–690, cited above. With the flip-flop **FF2** in its opposite state, a logical 1 in the form of a positive voltage will be applied to the AND gate **A2** to enable it to emit a pulse for each applied character pulse until the next index pulse. Since the pulses emitted from the AND gate **A2** are thus keyed to the preceding index pulse, they identify the characters coming into printing position on the print wheels in the sequence in which they appear, and one set of character pulses so emitted represents one complete sequential scan of the character fonts. This sequence is used to generate a character scanning code having a unique code

sequence for each character on the print wheels, in a manner which will be made clear below.

The output terminal of the AND gate A2 is connected to one input terminal of each of the AND gates A3, A4, A5 and A6. The other input terminal of each of these gates is connected to a different one of the output terminals *b*, *c*, *d* and *e* of a scan code generator, here shown as a binary counter **26**. While any conventional code generator capable of emitting a predetermined sequence of digital code sequences could be employed for this purpose, one suitable binary counter is shown which comprises four trigger circuits **T1**, **T2**, **T4** and **T8**, whose reference numerals indicate the relative binary order of the output of each trigger circuit. These trigger circuits are interconnected to form a binary counter which will generate code sequences in descending binary order on its output terminals *b*, *c*, *d* and *e* in response to successive positive-going input pulses applied to its input terminal *a*. The trigger circuits may be constructed in the manner shown in detail in Fig. 15.54 on pages 15–55 of Hunter, Handbook of Semiconductor Electronics, published in 1950 by the McGraw-Hill Book Company, Inc., although any other suitable circuit could be employed, if so desired, without departing from the scope of my invention. The details of these circuits and their mode of operation are explained in the reference publication; it is sufficient for the understanding of my invention to point out that in response to a positive-going pulse applied to the central input terminal *b* of a trigger circuit such as **T1**, its complementary output terminals *e* and *d* will switch from one state to an opposite state, and that a succeeding pulse will switch these terminals back to their initial state. The binary zero state is assumed to be the one in which terminal *e* of the trigger circuit **T1** is at ground potential and terminal *d* is at a positive potential. In the binary 1 state, terminal *e* will be at a positive potential and terminal *d* will be at ground potential. In passing from the 1 state to the **0** state of the trigger circuit, a positive-going pulse will be transmitted through a capacitor **C3** from the output terminal *e* of the trigger circuit **T1** to the central input terminal of the trigger circuit **T2**, causing it to reverse its state. The output of the trigger circuit **T2** is similarly connected to the input of the trigger circuit **T4**, and the trigger circuit **T4** is connected to the trigger circuit **T8**, in the manner shown in the drawing. It will be apparent to those skilled in the art that with these connections, a series of input pulses will cause the trigger circuits to change state, **T1** on each pulse, **T2** on every other pulse, **T4** on every fourth pulse, and **T8** on every eighth pulse. Thus, the output terminals *e* of the trigger circuits will cycle through the binary sequence 0000, 0001, 0010, 0011, etc., while the output terminals *d* will cycle through the complementary sequence 1111, 1110, 1101, 1100, etc. At the seventeenth pulse, the counter will be reset to zero and the count will begin anew. As shown in FIG. 1*b*, the complementary output terminals *d* of the trigger circuits are connected to the output terminals *b*, *c*, *d* and *e* of the counter. The reason for the selection of the complementary sequence for the character scan sequence will be made apparent below.

The output terminal *e* of the trigger circuit **T8** is connected through a suitable coupling capacitor **C6** to an output terminal *f* of the counter. At each seventeenth input pulse, a positive-going pulse will be applied to the terminal *f*, which pulse is used as an "end-of-print" signal in a manner set forth in detail below.

It is desired to produce this end-of-print pulse after the end of each character scan. Accordingly, a connection is made through suitable isolating diodes, as shown, to the input terminal *c* of each of the trigger circuits to apply a pulse through the capacitor **C1** at the beginning of each print cycle, to set all of the trigger circuits to their logical zero states. Thus, the counter will always start at zero and the first input pulse applied to input terminal *a* of the counter will step the counter to its second state, so that the output pulse will appear at terminal *f* on the sixteenth input pulse. After the first cycle of operation, the pulse applied to terminals *c* of the triggers will have no effect, because the counter will be in the zero state when their pulse is applied.

While the counter and the data source have been described in terms of a four-bit code, it will be apparent that characters requiring larger or smaller codes could be employed without requiring more than an obvious extension or contraction of the system.

It will be seen from the above description that at each character pulse gated to the AND gates A3, A4, A5 and A6 by the AND gate A2, the code stored in the counter **26** will be gated to the input terminals *f*, *g*, *h* and *i* of the comparators **CP1** and **CP2**. As each code sequence in the scan is thus applied to the comparators, it is compared with the character code stored in the comparator. At the same time, a sample pulse is applied to the input terminal *j* of each comparator, and, if the character then coming into position on the print wheels, which corresponds to the scan code sequence, is a member of the set complementing each 1 in the character code for the stored character in a given comparator, the sample pulse is gated to the output terminal *e* of the comparator. The output terminal *e* of each comparator is connected to an input terminal *a* of a print hammer driver circuit, such as **DR1** and **DR2**, to be described. The print hammer drivers are controlled by a hammer drive control unit **27**, to be described, which controls the print hammer drivers in such a way that, at the first pulse applied to the input terminal *a* of a print hammer driver such as **DR1** during a given print cycle, the associated print hammer coil such as **HC1** is energized to print the character then in position. However, succeeding pulses during the same print cycle will not cause energization of the print hammer coil. The necessity for this mode of operation will be made apparent below.

The output of the AND gate A2 is also applied to the input of a suitable delay line **D1**. This unit, as well as other delay lines to be described, may be of any conventional construction for example, it may be of the type shown and described in my United States application Serial No. 106,840, filed May 1, 1961, for Pulse Retiming System, which is assigned to the assignee of this application. It is sufficient for the understanding of my invention to note that the delay line **D1** will produce a positive-going output pulse a predetermined time after a positive-going input pulse is applied.

The output pulses produced by the delay line **D1** are applied to the input terminal *a* of the counter **26**, so that the counter is set to the next binary state after its current state has been gated to the comparators. The reasons for this mode of operation are, first, that it is necessary to use the zero state of the counter in the character scan sequence, as will appear, and second, that it would be undesirable to gate the counter output during a change of its state.

Refer now to FIG. 2, in which the details of the comparators, the print hammer driver, and the hammer drive control unit **27** are shown. Considering first the comparators, all of these units may be identical; therefore, only the unit **CP1** is shown in detail. As shown, each bit of the stored character code (from the data source **17** in FIG. 1*b*) is applied to one input terminal of a different one of four AND gates **A7**, **A8**, **A9** and **A10**. The corresponding bit of the character scan code (from the counter **26** in FIG. 1) is applied to the other terminal of the associated AND gate. Thus, each of the AND gates **A7**, **A8**, **A9** and **A10** will produce a positive output potential, corresponding to a logical 1 level, if and only if both of the input bits are positive, corresponding to logical 1 levels.

The outputs of the AND gates **A7**–**A10** are applied to a logic unit **LU1**, which is typical of the other similarly designated units to be described. In this unit, the output

terminals of the AND gates A7, A8, A9 and A10 are connected to the input terminals of an OR gate OR1. This OR gate may be of any suitable construction capable of producing a positive output pontetial in response to a positive potential applied to any one or more of its four input terminals, such as the OR gate shown in Fig. 206A on page 211 of the above-cited TM11–690, which may be provided with additional input terminals as pointed out on page 211.

The output of OR gate OR3 is connected to the y input terminal of an $x\bar{y}$ gate **28**. A suitable circuit for the gate **28** is shown in Fig. 15.43(a) on pages 15–46 of the above-cited Handbook of Semiconductor Electronics. It will suffice for the understanding of my invention to note that this circuit will produce a positive potential on output terminal f of the logic unit LU1 if and only if a positive potential is applied to its input terminal x but not to its input terminal y. As shown, the sample pulses (from the AND gate A2 in FIG. 1) are applied to the x terminal of unit **28**, so that a print command pulse will appear at output terminal e of the comparator CP1 if and only if a sample pulse has been applied to the input terminal j of the comparator, and no two corresponding bits of the stored character and character scan codes are both logical 1's.

Assume for simplicity that there are 16 characters on each print wheel, and identify their order of appearance in printing position following an index pulse by the numerals from 1 to 16. Let the stored characters be coded from 0 to 15 in the ascending binary sequence. For each character, the corresponding codes will then be given by the following table:

Character No.	Stored Character Code	Character Scan Code
1	0000	1111
2	0001	1110
3	0010	1101
4	0011	1100
5	0100	1011
6	0101	1010
7	0110	1001
8	0111	1000
9	1000	0111
10	1001	0110
11	1010	0101
12	1011	0100
13	1100	0011
14	1101	0010
15	1110	0001
16	1111	0000

To illustrate the mode of comparison, suppose that character number 12 was stored for the print wheel corresponding to comparator CP1. The input terminals a, b and d of the comparator CP1 and then be at a positive potential, and the input terminal c would be open, representing the character code 1011. As the scan progressed following the first index pulse in a given print cycle, the AND gates A7, A8, A9 and A10 would have the inputs and outputs given by the following table, in which C represents a stored character code input bit, S represents a scan code input bit, and O represents the output of the AND gate, with positive potentials being indicated by logical 1, and ground potential or an open terminal indicated by logical 0:

Character number	Gate											
	A10			A9			A8			A7		
	C	S	O	C	S	O	C	S	O	C	S	O
1	1	1	1	0	1	0	1	1	1	1	1	1
2	1	1	1	0	1	0	1	1	1	1	0	0
3	1	1	1	0	1	0	1	0	0	1	1	1
4	1	1	1	0	1	0	1	0	0	1	0	0
5	1	1	1	0	0	0	1	1	1	1	1	1
6	1	1	1	0	0	0	1	1	1	1	0	0
7	1	1	1	0	0	0	1	0	0	1	1	1
8	1	0	1	0	0	0	1	0	0	1	0	0
9	1	0	0	0	1	0	1	1	1	1	1	1
10	1	0	0	0	1	0	1	1	1	1	0	0
11	1	0	0	0	1	0	1	0	0	1	1	1
12	1	0	0	0	1	0	1	0	0	1	0	0
13	1	0	0	0	0	0	1	1	1	1	1	1
14	1	0	0	0	0	0	1	1	1	1	0	0
15	1	0	0	0	0	0	1	0	0	1	1	1
16	1	0	0	0	0	0	1	0	0	1	0	0

It will be seen that none of the AND gates A7, A8, A9 and A10 will produce an output in response to the scan codes corresponding to characters 12 and 16, and that at least one of the AND gates will produce an output in response to every other scan code. Referring to FIG. 2, it will be recalled that logic unit LU1 will emit a print command pulse when and only when a sample pulse is applied to its input terminal x and none of the AND gates A7, A8, A9 and A10 supply a pulse, through the OR gate OR1, to the input terminal y of the unit LU1. Accordingly, a print command pulse will be produced only when the 12th and 16th characters come into position. As will appear, the print hammer drivers such as DR1 will energize their hammer drive coils such as HC1 only once during a print cycle, so that only the 12th character on the print wheel will be printed in the example here considered.

The operation of the comparator with other stored character codes may be seen from the following table, in which the scan codes which will cause a print command pulse to be generated for each stored character code which may be supplied to a comparator are indicated by an x in the column designated by the decimal value of the binary number corresponding to the stored character.

Character Number	Character Scan Code	Stored Character Code (Decimal Value)															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1111	X															
2	1110	X	X														
3	1101	X		X													
4	1100	X	X	X	X												
5	1011	X				X											
6	1010	X	X			X	X										
7	1001	X		X		X		X									
8	1000	X	X	X	X	X	X	X	X								
9	0111	X								X							
10	0110	X	X							X	X						
11	0101	X		X						X		X					
12	0100	X	X	X	X					X	X	X	X				
13	0011	X				X				X				X			
14	0010	X	X			X	X			X	X			X	X		
15	0001	X		X		X		X		X		X		X		X	
16	0000	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Since it is readily implemented with a simple binary counter, the descending binary code sequence is preferred for scanning in the comparator of my invention. However, a large number of other scan code sequences is possible, which sequences may be obtained by reordering the descending binary sequence in such a way that each scan

code sequence applied to the comparators will produce an output with one and only one character code in addition to those previously scanned. A typical rearrangement is shown in the following table, using a three bit code for conciseness; eight characters on the print wheels are assumed.

Character Number	Character Scan Code	Stored Character Code							
		000	001	010	011	100	101	110	111
1	111	X							
2	011	X				X			
3	101	X		X					
4	110	X	X						
5	011	X		X		X		X	
6	010	X	X			X	X		
7	100	X	X	X	X				
8	000	X	X	X	X	X	X	X	X

It will be seen that a stored character represented by binary zero (000) will respond to any scan code, and any scan code other than 111 will also produce a response with at least one other character code. Thus, a principle of reordering is that if 000 is not excluded, 111 must be the first scan code. Also, any character code will respond to the scan code 000, so that if this scan code is used, it must be last in the scan sequence. If desired, the scan code may be truncated, as where fewer characters than a full binary set are used, by taking away the last sequences in the scan code first.

More specifically, a scan code sequence in which there are m 0's will produce a response with 2^m-1 character code sequences, including the zero sequence 000 . . . 0. Thus, if such a sequence is designated as Sm, and the response is to be unambiguous, Sm must be preceded by a set of 2^m-2 other scan code sequences besides the sequence 11 . . . 1 and not including the zero sequence, which include fewer than m 0's and which would produce a response with all but one of the sequences with which Sm would produce a response. For example, if the sequences are ranked in sets in accordance with the number of 0's as S_0, S_1, . . . S_m . . . S_n, then they might be arranged in that order with the individual members of the sets S_0, S_1, etc., arranged in random order to form a class of scan code sequences operative in the comparator of my invention. However as illustrated by the descending binary sequence, it is not necessary to exhaust all of the sequences containing less than m 0's before introducing a sequence Sm, but only to scan for all of its possible 1-complementing sequences except the full complement, which contains m 1's. Thus, the first scan code sequence containing two 0's must be preceded by $2^2-2=2$ scan code sequences each containing one 0. For example, the sequence 1011011 must be preceded by the sequences 1011111 and 1111011, in either order, to exclude all of the character codes except 0100100. However, the next S_2 sequence may be added following only one additional S_1 sequence, if it is related to one of the previous S_1 sequences. Thus, 1111101 could be added to the sequence, followed by 1011101. A scan code beginning as follows might then be formed:

Character Number	Character Scan Code	Stored Character Code					
		0	32	4	36	2	34
1	1111111	X					
2	1011111	X	X				
3	1111011	X		X			
4	1011011	X	X	X	X		
5	1111101	X				X	
6	1011101	X	X		X	X	X

In which the responses are indicated by the X's and the character codes causing the responses are represented by their decimal equivalents, as before. In this example, the character codes are scanned in the order 0, 32, 4, 36, 2, 34 . . . .

In general, sequences other than the normal descending binary will require a pattern generator other than the simple binary counter. One suitable pattern generator for generating any desired code sequence may be made by mounting a series of toothed discs, such as the character pulse generator disc **9** in FIG. 1, on the shaft **2**, each provided with a pickoff coil such as **12** in FIG. 1, and each having selected teeth removed to produce a sequence of logical 1 and 0 levels as each tooth position passes the pickoff coil.

Referring again to FIG. 2, the hammer drive control unit **27** and the print hammer driver **DR1** will next be described. Since the other print hammer drivers, such as **DR2**, are the same as **DR1**, and are controlled in the same way by the hammer drive control unit **27**, they will not be described in detail.

The hammer drive control unit **27** comprises a silicon controlled rectifier **S1**, having its load terminals *a* and *b* connected in series with the secondary winding **29** of a pulse transformer **PT1**, a suitable current limiting resistor **R1**, and a suitable source of voltage such as a battery **B1**. A conventional diode **30** is connected between the base terminals *b* and *c* of the controlled rectifier **S1** as shown. A suitable coupling capacitor **C9** is connected between an input terminal *a* of the control unit and the base terminal *c* of the controlled rectifier **S1**, which serves to couple an applied positive-going end-of-print pulse to the base terminal *c*. The pulse transformer **PT1** is provided with a primary winding **31**, connected between input terminal *b* of the unit **27** and ground; when the switch **13** in FIG. 1 is first closed, this winding **31** induces a blocking pulse in the winding **29**, for purposes to be described.

As is well known in the art, a controlled rectifier such as **S1** will not conduct current in response to a forward biased voltage across its load terminals unless its control base terminal such as *c* is biased positively with respect to its load base terminal such as *b*. Thereafter, the controlled rectifier will continue to conduct, regardless of the bias potential, until the current in the load circuit is interrupted. Assuming that the controlled rectifier **S1** is not conducting, a positive pulse applied between the input terminal *a* of the unit **27** and ground will switch it to its conducting state, causing load current to flow through the secondary winding **29** of the pulse transformer **PT1** and the resistor **R1**. This current will continue to flow until a "start print" pulse is applied to the secondary winding **29** of the pulse transformer **PT1** by the primary winding **31** in response to the closing of the switch **13** in FIG. 1*b*. This pulse will momentarily reverse-bias the controlled rectifier, causing it to switch back to its non-conducting state. Thus, during the interval between an end of print pulse applied to the input terminal *a* of unit **27** and an ensuing start print pulse applied to the input terminal *b*, a voltage will appear across the output terminals *c* and *d* of the unit **27**, which voltage is used as a supply for the print hammer drivers.

A print hammer driver such as **DR1** is provided for each hammer drive coil such as **HC1**. As shown, the hammer drive coil **HC1** is connected in series with the load circuit of a silicon controlled rectifier **S2**, of the same type as the silicon controlled rectifier **S1** in the hammer drive control unit **27**. The hammer coil **HC1** is at times energized, under conditions to be set forth, through a circuit extending from the positive terminal of an electrolytic capacitor **C7** through a variable current trimming resistor **R2**, the hammer coil **HC1**, the controlled rectifier **S2** in its conducting state, and back to the negative terminal of the capacitor **C7**. A conventional diode **33** may be connected across the capacitor **C7** as shown, to protect it against transient negative voltages.

A charging circuit for the capacitor **C7** extends from the output terminal *c* of the hammer drive control unit **27** through a conventional diode **32**, which serves to isolate the print hammer driver **DR1** from the other print

hammer drivers, a current limiting resistor R3 which serves to time the charging of the capacitor C7, the current trimming resistor R2, the capacitor C7, and thence to the grounded terminal *d* of the hammer drive control unit 27.

A control circuit for the controlled rectifier S2 is provided in the same manner as for the controlled rectifier S1. Thus, a blocking diode 34 is connected across the base terminals *b* and *c* of the controlled rectifier S2, to prevent the control base terminal *c* from going negative during conduction between the input terminal *a* and the base terminal *b*, and also to permit the control terminal *c* to be biased positively with respect to the base terminal *b* when it is desired to switch the controlled rectifier S2 to its conducting state. A capacitor C8 is connected between the base terminal *c* of the controlled rectifier and the output terminal *e* of the comparator CP1, to couple a positive pulse to the base terminal *c* for each print command pulse emitted by the comparator.

It is desirable to discharge the capacitor C7 much more rapidly than it is practical to charge it. Accordingly, in the hammer driver circuit, the resistance of the current trimming resistor R2 is preferably much less than that of the current limiting resistor R3, so that the discharge time of the capacitor C7 may be quite short with respect to the charging time.

The operation of the print hammer driver DR1 may best be considered beginning with the conditions prevailing just before the start of the print cycle. At this time, the controlled rectifier S1 in the hammer drive control unit 27 is conducting, and charging current is supplied to charge the capacitor C7, through the charging circuit previously traced.

When a print cycle is initiated by closing the switch 13 in FIG. 1*b*, a "start print" signal is supplied to the hammer drive control unit 27 in the manner described above, and the output terminal *c* of the hammer drive control unit will go to ground potential. Discharge of the capacitor C7 is blocked at this time by the diode 32 and the controlled rectifier S2 in its non-conducting state.

At some time during the rotation of the print wheel shaft 2 in FIG. 1*a*, the character code stored at the input terminals *a*, *b*, *c* and *d* of the comparator CP1 in FIG. 2 will match the scan code applied to its input terminals *f*, *g*, *h* and *i*, and a print command pulse will be applied to the input terminal *a* of the print hammer driver DR1 in the manner described above. This print command pulse will switch the controlled rectifier S2 to its conducting state, and the capacitor C7 will be discharged through the hammer drive coil HC1, causing the stored character to be printed. With the capacitor C7 discharged, any succeeding print command pulses emitted by the comparator CP1 will be ineffective.

At the end of the scan sequence, the counter 26 in FIG. 1*b* will emit an "end-of-print" pulse, which will reset the controlled rectifier S1 in the hammer drive control unit 27 to its conducting state. Thereafter, charging current will be supplied to the capacitor, such as capacitor C7, of the print hammer driver circuits, until the next print cycle is initiated.

The overall operation of the embodiment of my invention just described will next be described with reference to FIGS. 1*a*, 1*b*, and 2. Since the operation of the apparatus for each column to be printed is the same, only the printing of a single character in the first column, using the print wheel P1, will be described. Using the binary scan and character codes given above, assume that it is desired to print the fourth character on the wheel P1, for which the corresponding code is 0011. This code is first stored in the data source 17 by closing the switches 18 and 19. It is assumed that the motor M is in operation, and that the print wheels 1 are rotating. Next, the switch 13 in FIG. 1*b* is closed, causing the flip-flop FF1 to produce a positive voltage level on its output terminal *c*. This voltage performs three functions. First, referring to FIG. 2, it produces a pulse output across the secondary winding 29 of the pulse transformer PT1, which cuts off the controlled rectifier S1, and restores the output terminal *c* of the hammer driver control unit 27 to ground potential. This action terminates the charging of the print hammer driver DR1 as previously described. Second, the application of a positive voltage to the lead 15 applies the data stored on switches 18, 19, 20 and 21 to the input terminals *a*, *b*, *c* and *d* of the comparator CP1, so that the terminals *c* and *d* are open and terminals *a* and *b* are at a positive potential. Third, the AND gate A1 is enabled to pass the next index pulse.

When the tooth C on the index pulse generator disc 5 passes the frame 7, a pulse is applied to input terminal *a* of the flip-flop FF2, which has no effect because the flip-flop is already in its logical 0 state, and the pulse is also applied through the AND gate A1 to the input terminal *b* of the flip-flop FF2, causing it to shift to its logical 1 state. This action enables the AND gate A2 to pass the next character pulse.

As the character pulses are generated, the AND gate A2 performs three functions. First, a sample pulse is applied to the input terminal *j* of the comparator CP1. Second, during each character pulse the AND gates A3, A4, A5 and A6 are enabled to apply the potentials appearing on the output terminals *b*, *c*, *d* and *e* of the binary counter 26 to the input terminals *i*, *h*, *g* and *f*, respectively, of the comparator CP1. Third, each character pulse is applied to the delay line D1. A delayed time thereafter, the delay line D1 then applies a pulse to the input terminal *a* of the binary counter 26 to shift it to the next state. Referring now to FIG. 2, in the example given, the terminals *c* and *d* of the comparator CP1 are unenergized, and the input terminals *a* and *b* are energized. Thus, the AND gates A9 and A10 are cut off and the AND gates A7 and A8 are enabled to produce a pulse for any logical one in the scan code applied to terminals *f* and *g*. As the fourth character comes into position, the scan code 1100 will be generated, and terminals *h* and *i* of the comparator CP1 will both be at ground potential. Accordingly, there will be no input to the OR gates in the logical unit LU1, so that the sample pulse simultaneously applied to the *x* terminal of the $x\bar{y}$ circuit will cause a print command pulse to be applied to the input terminal *a* of the print hammer driver DR1. The capacitor C7 will then discharge through the hammer drive coil HC1 in the manner previously described causing the fourth character on the print wheel P1 to be printed. Print command pulses will also be emitted on the eighth, twelfth, and sixteenth scan code sequences, but, with the capacitor C7 discharged, these signals will be ineffective.

After the sixteenth character pulse, the binary counter 26 emits an end-of-print pulse, which performs two functions. First, it is applied to the input terminal *a* of the hammer drive control unit 27 in FIG. 2 to restore the controlled rectifier S1 to its conducting state and resume the charging of the capacitors in the print hammer drivers. Second, it is applied to the input terminal *a* of the flip-flop FF1 to restore it to its 0 state, thus disabling the data source 17 and the AND gate A1 in FIG. 1*a*.

The next index pulse emitted by the generator 3 will restore the flip-flop FF2 to its 0 state, disabling the AND gate A2. Since the flip-flop FF1 cannot be reset until the switch is again opened and then closed, additional index pulses and character pulses which may occur will be ignored. When the switch 13 is opened, the apparatus will be restored to its original condition.

Referring now to FIGS. 3*a* and 3*b*, a modification of the apparatus of my invention for use with transient data presented serially in parallel form is shown. Data may be supplied to the system by applying character codes, one code sequence at a time, to the input terminals *a*, *b*, *c* and *d* in FIG. 3*a*. Thus, for example, assume that the

fifth character on the print wheel P1, having the character code 0100, was to be stored, a positive pulse would momentarily be applied to input terminal *b* in FIG. 3*a*. It is assumed that together with each character code an additional pulse will be applied to an auxiliary input terminal *m*, which will be termed a "load character" pulse. Suitable apparatus for supplying data in this form is well known, and need not be described. If desired, it could be supplied by a manually set switches in the manner shown in FIG. 1*b*.

Before describing the details of the system of FIGS. 3*a* and 3*b*, the arrangement of the components and their functions will be briefly described. In general, the apparatus is adapted to be used with a printer of the type described in connection with FIGS. 1*a* and 1*b*, provided with a series of print wheels, each having a hammer controlled by one of the print hammer drivers DR1, DR2, . . . DR*n*, which may be identical with the print hammer drivers DR1 shown in FIG. 2. These print hammer drivers are controlled by a hammer drive control unit 7, shown in FIG. 3*b*, which is shown in more detail in FIG. 2. The index pulse generator 3 and character pulse generator 4 are the same as those units shown in the previously described emobdiment, as is the counter 26.

The print command signal for each print hammer driver is provided by an associated logic unit, such as LU1, LU2 and LU*n*. These logic units may be of the construction shown for unit LU1 in FIG. 2. In the system shown in FIG. 3*a*, the data for each character to be printed is shifted through a series of storages until it is stored in a storage unit associated with the proper print hammer driver. Thus, for the first print hammer driver DR1 there is provided the first storage unit S1A for the first bit of the character code, and storage units S1B, S1C and S1D for the remaining bits of the character code. Associated with the print hammer driver DR2 is a similar series of storage units, S2A, S2B, S2C and S2D. Likewise, the *n*th print hammer driver DR*n*, as well as intermediate units, is provided with a series of storages S*n*A, S*n*B, S*n*C and S*n*D. As it will appear, considering only the first bit of the character code, which corresponds to the *n*th column to be printed, it is first stored in the unit S1A, then shifted into the storage unit S2A, and finally into the storage S*n*A, from which it is transferred to the logical unit LU*n*, together with the remaining bits of the character code stored in the units S*n*B, S*n*C and S*n*D, to actuate the print hammer driver DR*n* when the appropriate scan code is generated. The manner in which this function is carried out will be made clear below.

The input terminals *a*, *b*, *c* and *d* in FIG. 3*a* are connected to the input terminals *a* of a series of flip-flops FF3, FF4, FF5 and FF6. The application of a positive pulse to these input terminals of the flip-flops will cause them to shift to their logical 1 states in which their output terminals *d* are at ground potential. These terminals are each connected to one terminal of an output coupling capacitor, such as the capacitor 45 shown for the flip-flop FF3, and the other terminal of each capacitor is connected to ground through a suitable clamping diode such as the diode 46. Thus, setting of the flip-flops to their logical 1 states will not produce an output pulse, because the diode 46 will not permit the output to go below ground, but resetting them to their logical 0 states will produce positive-going output pulses.

The terminal *m* to which the load character pulse is applied is connected to the input terminal of a delay line D2, which may be of the same type as the delay line D1 in FIG. 1*a*. The output of the delay line D2 is connected to the input terminals *b* of the flip-flops FF3, FF4, FF5 and FF6. Thus, each flip-flop is reset a delayed time after each character code is presented to it, causing it to emit a positive pulse if a logical one, or a positive pulse, was presented to the input terminal *a* of that flip-flop by the applied character code.

The input terminal *m* is also connected to one input terminal of each of a group of OR gates OR4, OR5, OR6 and OR7, which may be the same as OR gate OR1 in FIG. 2, except that they include only two input terminals. The output of these OR gates are connected to a series of driver amplifiers 40, 41, 42 and 43. The functions of these amplifiers will be described after describing the storage units in somewhat more detail.

Referring now to the storage unit S1A, this unit comprises a switch core SC1 of saturable ferromagnetic material, which is provided with four windings *a*, *b*, *c* and *d*. An energizing circuit for the winding *a* of the switch core SC1 extends from the output of the flip-flop FF3 through the winding *a*, through an isolating diode 47, and over a common lead 48 to the output terminal *c* of the flip-flop FF2, for purposes to be described. Thus, a positive pulse from the flip-flop FF3 will cause current to flow through the winding *a* of the switch core SC1, driving it to saturation in one direction, if the flip-flop FF2 is in its logical 0 state, with its output terminal *c* at ground potential, but if the flip-flop FF2 is in its logical 1 state, the diode 47 will be blocked and will prevent the flow of current through the winding *a*. The winding *c* of the switch core SC1 is connected to a delay unit D2A, comprising a conventional diode 49 and a capacitor 50. The output of the delay unit D1A is connected to one terminal of the input winding *a* of a switch core SC2 in the second storage unit S2A. The other terminal of the winding *a* on the core SC2 is connected through a blocking diode 51 to the common lead 48, which is in turn connected to the output terminal *c* of the flip-flop FF2.

The output of the delay unit D1A is also connected to one terminal of the winding *b* of the switch core SC1. The other terminal of the winding *b* is connected through a blocking diode 52 to a common lead 53. As shown, the common lead 53 is connected to the output terminal *d* of the flip-flop FF2.

With these connections, it will be apparent that a voltage induced in the winding *c* will not cause an output pulse from the delay unit D1A unless it is positive-going, and that a positive-going pulse which does produce an output will be effective to cause current to flow through the winding *a* of the switch core SC2 if the flip-flop FF2 is in its logical 0 state, but not with it in its logical 1 state, and will cause a current to flow through the winding *b* of the core SC1 in the logical 1 state of the flip-flop FF2 but not in the logical 0 state. The utility of these connections will be made to appear.

The windings *d* of each of the switch cores SC1, SC2 and SC3 are connected in series to the output terminals of the amplifier 40 in FIG. 3*b*. Similar connections are made to the output terminals of amplifiers 41, 42 and 43, from the switch cores in the remaining storage units. When the associated OR gate produces a positive pulse, each amplifier produces a negative pulse, which serves to shift each switch core to the saturated state opposite to that which a positive pulse applied to its winding *a* would produce.

Referring now to FIG. 3*b*, the print cycle may be initiated in essentially the same way as described in connection with FIGS. 1*a* and 1*b*. Thus, the closing of the switch 13 applies a pulse from the battery 14 through the capacitor C1 to set the flip-flop FF1 to apply a positive voltage to the lead 15. The control of the AND gates A1 and A2 and the flip-flop FF2 may be the same as that described for the first embodiment. Also, the AND gates A3, A4, A5 and A6, may be connected in the same way to the output of the counter 26. As before, the delay unit D1 applies pulses to the counter 26 a delayed time after the AND gates A3–A6 sample the scan code stored in the counter, and at the end of the scan an "end-of-print" signal is applied by the counter 26 to reset the flip-flop FF1. In this embodiment, the delay of the delay line D1 is matched to the delays of the units

such as D1A, D2A, etc., in the storage units, for reasons which will appear.

The hammer drive control unit **27** is sequentially operated by the start print signal from the flip-flop FF**1** and the end-of-print signal from the counter **26**, to control the charging of the capacitors in the print hammer drivers DR1, DR2, etc., in the manner described above in connection with FIGS. 1*a* and 1*b* and 2, and this description will not be repeated.

The operation of this embodiment of my invention will now be described, assuming that the 9th, 1st, and 13th characters are to be printed in the first, second and last columns of a line to be printed. The character for the last column must be entered first, so that the characters will be presented in the order 13, 1, and 9. Assume that the switch **13** is open, so that the flip-flops FF**1** and FF**2** will be in their logical 0 states. Likewise, flip-flops FF**3**, FF**4**, FF**5** and FF**6** may be assumed to be in their logical 0 states.

Loading of the first character, assumed to be the 13th character in the print wheel series having a character code 1100, is accomplished by applying a positive pulse to the input terminals *a*, *b* and *m* in FIG. 3*a*. Flip-flops FF**3** and FF**4** will thus be set to their logical 1 states, in which their output terminals *d* are at ground potential. The pulse applied to input terminal *m* actuates the OR gates OR**4**, OR**5**, OR**6** and OR**7** to produce output pulses. These pulses are inverted by amplifiers **40**, **41**, **42**, and **43** respectively. This action produces a set of negative output pulses; confining attention to the amplifier **40**, its output pulse is applied to each of the windings *d* of the switch cores SC**1**, SC**2** and SC*n*. In the initial state, each of these cores will be saturated, in a manner which will appear, in the same sense directed by the applied pulse, such that no output pulses will be produced. Accordingly, this pulse, and the similar pulses produced by amplifiers **41**, **42**, and **43**, will have no effect on the system.

Referring to FIG. 3*a*, a delayed time after application of the load character pulse to input terminal *m*, the delay line D**2** will produce an output to reset the flip-flop FF**3** and FF**4** to their logical 0 states. Since the flip-flops FF**5** and FF**6** are already in their logical 0 states, they will not be affected. However, the flip-flops FF**3** and FF**4** will produce positive-going output pulses. The pulse from the flip-flop FF**3** will cause current to flow through the winding *a* of the switch core SC**1**, since the flip-flop FF**2** is in its logical 0 state. The switch core SC**1** will then be driven to saturation, causing a pulse to be induced across the windings *b* and *c*. However, the polarity of this pulse is such that it will be blocked by the diode **49**, so that the delay unit D1A will produce no output. The pulse produced across the winding *b* will be of the wrong polarity to affect the logic unit LU**1**; in any event, no sample pulses are applied to the input terminals *c* of the logic units during the loading of characters. In a similar manner the pulse from the flip-flop FF**4** will be stored in the unit S1B.

Next, the first character, having the character code 0000, is stored by applying a pulse only to the input terminal *m*. This pulse will be gated through the OR gates OR**4**, OR**5**, OR**6** and OR**7** to cause amplifiers **40**, **41**, **42**, and **43** to produce outputs in the manner described above. Confining attention to amplifier **40**, since there is no storage in the switch cores SC**2** through SC*n*, the current through the windings *d* on the switch cores in these units will be ineffective. However, the current through the winding *d* of the switch core SC**1** will drive the core to saturation in the opposite sense to that produced by the previous pulse through the winding *a*. This action will cause a pulse to be induced across the winding *c* of the proper polarity to appear, a delayed time later, across the capacitor **50**. Since the diode **52** is blocked at this time, this delayed pulse will not cause current to flow through the winding *b* of the switch core SC**1**. However, it will cause current to flow through the winding *a* of the switch core SC**2**, causing it to be driven to saturation and storing a logical one in the same manner as the switch core S1A was previously set. The remaining storage units operate in essentially the same way, such that the information previously stored in the storage units S1A, S1B, S1C and S1D is transferred to the storage units S2A, S2B, S2C and S2D.

The delayed pulse appearing across the capacitor **50** will be applied to input terminal *a* of the logic unit LU**1**. However, since no sample pulses are applied to input terminal *e* of the logic units during the loading cycle, the presence or absence of pulses at input terminal *a* during this time has no effect on the operation of the system.

A delayed time later, the delay line D**2** will apply a pulse to the flip-flops FF**3**, FF**4**, FF**5** and FF**6**. Since all of these flip-flops are already in their logical 0 states, no outputs will be produced. The next operations would be the loading of the characters between the last column and the second column. Since it is believed that this operation will be apparent from the description given, it will be assumed that only three columns are provided, such that the S*n* storages become the S**3** storages.

Next, the character for the first column to be printed, here assumed to be the ninth character having the character code 1000, is entered by applying a positive pulse to input terminals *a* and *m* in FIG. 3*a*. The flip-flop FF**3** will be set to its logical 1 state. At the same time, the pulse applied to input terminal *m* will be gated to the amplifiers **40**–**43** to energize the windings *d* of the switch cores. Assuming that there are no storages intervening between S2A and S*n*A, this pulse will cause the logical 1 stored in switch core SC**2** to be transferred to the switch core SC*n* after the delay provided by the delay unit D2A. Since there is a logical 0 stored in the switch core SC**1**, switch core SC**2** will not be reset, but will remain in its logical 0 state. Next, the delay unit D**2** will produce an output, causing the flip-flop FF**3** to revert to its logical 0 state which will store logical 1 in the switch core SC**1**. The remaining bits of the character code are entered in their respective storages in the same manner as described for the first bits of each code group. Thus, at this time the character code for the ninth character is stored as a logical 1 in unit S1A and logical 0's in units S1B, S1C and S1D. The first character, having the character code 0000, is stored by logical 0's in each of the storage units S2A, S2B, S2C and S2D. The 13th character, having the character code 1100, is stored by logical 1's in the unit S*n*A, and S*n*B, and by logical 0's in the unit S*n*C and S*n*D.

Next, the print cycle is begun by closing the switch **13**. The flip-flop FF**1** will now be set to its logical 1 state to enable the AND gate A**1**, so that the next index pulse from the generator **3** will produce an output from the AND gate A**1** to set the flip-flop FF**2** to its logical 1 state.

With the flip-flop FF**2** in its logical 1 state, its output terminal *c* is at a positive potential, enabling the AND gate A**2** and blocking all of the *a* windings of the switch cores by cutting off their associated blocking diodes, such as the diode **47** associated with winding *a* of the switch core SC**1**. Terminal *d* of the flip-flop FF**1** is at ground potential, such that current may flow through the windings *b* of the switch cores.

At the first character pulse from the generator **4**, the AND gate A**2** will transmit a pulse to the delay line D**1** and to the AND gates A**3**, A**4**, A**5** and A**6**. The output of the counter **26**, which is 1111 at this time, is then applied through AND gates A3–A6 and OR gates OR**4**–OR**7** to amplifiers **40**–**43**, respectively. Current is then supplied from the amplifiers **40**–**43** to all of the windings *d* of the switch cores. Each switch core in the logical 1 state will then be switched from saturation in one sense to saturation in the opposite sense. In the example given,

the switch cores in the storage units S1A, S*n*A and S*n*B will be thus affected. Considering the switch core SC1 in storage unit S1A, an output pulse will be induced across the winding *c* which will appear, a delayed time later, across the capacitor **50**. Current will then flow through the winding *b* of the switch core SC1, since the diode **52** is not blocked at this time, and this current will again switch the core SC1 to its logical 1 state. In this manner, each time a logical 1 in a switch core is erased by an applied scan code, it is rewritten in the storage by the output of the associated delay unit. The output of the delay unit D1A will also be applied to the winding *a* of the switch core SC2 in the storage unit S2A, but with the diode **51** blocked, no current will flow through the winding *a* and the switch core SC2 will remain in its logical 0 state.

The output pulse from the delay unit D1A will be applied to input terminal *a* of the logic unit LU1, and similar pulses will be applied to input terminals *a* and *b* of the logic unit LU*n* from the delay units in the storage units S*n*A and S*n*B. No pulses will be applied to the input terminals *a*, *b*, *c* and *d* of the logic unit LU2, since none of the switch cores in the storage units S2A–S2D changes state.

At the same time, the pulse which was applied to the delay line D1 by the AND gate A2 will appear at the output of the delay line. This pulse is applied to the input terminals *e* of the logic units LU1, LU2 and LU*n*, and to input terminal *a* of the counter **26**.

It will be recalled from the description of the logic unit LU1 in FIG. 2 that a print command pulse at terminal *f* will be produced if and only if an input pulse is applied to terminal *e* and no input pulse is applied to terminals *a*, *b*, *c* and *d*. Thus, the logic units LU1 and LU*n* will not produce print command pulses, and logic unit LU2 will apply a print command pulse to terminal *a* of the print hammer driver DR2 to print the first character in the second column.

The pulse applied to input terminal *a* of the counter **26** will shift the counter to its next state, in which the output on terminals *b*, *c*, *d* and *e* will be 1110. At the next character pulse, this output will be gated to the amplifiers **40–43**, causing the amplifiers **40**, **41** and **42** to produce output pulses. The operation of the storage units will be the same as for the first scan code, in the example given, and only the logic unit LU2 will produce an output. However, this output will have no effect because the print hammer driver DR2 can operate only once during a print cycle.

The scanning will continue, in a manner which will be apparent from the above description, until the ninth scan code, 0111, is gated from the counter **26** to the amplifiers **40**, **41**, **42** and **43**. Only the amplifiers **41**, **42** and **43** will produce output pulses. Thus, the switch core SC1 will not change state, and none of the switch cores in the units S1B, S1C and S1D will change state because they are already in the logical 0 state. Therefore, when the delayed character pulse is applied to input terminal *e* of the logic unit LU1, a print command pulse will be applied to the print hammer driver DR1 to print the ninth character in the first column.

As before, the logic unit LU2 will produce an output pulse, which will be ineffective because the print hammer driver DR2 is discharged. The logic unit LU*n* will not produce an output, because a pulse will be applied to its input terminal *a* by the storage unit S*n*A, and to its input terminal *b* by the storage unit S*n*B.

At the thirteenth scan code, 0011, output pulses will be provided by the amplifiers **42** and **43**, which will be ineffective because no logical 1's are stored in the storage units supplied by these amplifiers. No outputs will be provided by the amplifiers **40** and **41**, so that no inputs will be applied to the terminals *a*, *b*, *c* and *d* of the logic units LU1, LU2, and LU*n*. Thus, the delayed sample pulse from the delay line D1 will cause print command pulses to be emitted from all of the logic units. The print hammer drivers DR1 and DR2 are already discharged, but the print hammer driver DR*n* will operate to print the thirteenth character in the third column, which is the *n*th column in the example here considered.

Subsequent scan codes will have no effect on the print hammer driver, since they are all discharged. After the sixteenth character pulse, the output of the delay line D1 will set the counter to its first state, corresponding to the scan code 1111, in which state it will remain until the next print cycle. When the counter shifts to its first state, an output pulse is applied from its terminal *f* to terminal *a* of the hammer drive control unit **27**, to cause it to charge the capacitor in the print hammer driver.

When it is desired to move the paper to the next line position, the switch **13** is opened. Thus, the apparatus is restored to its initial condition, except that the data for the line just printed is still stored in the storage units.

The next index pulse from the generator **4** will switch the flip-flop FF2 back to its logical 0 state to cut off the AND gate A2. With the flip-flop FF2 in its logical 0 state, the windings *a* of the switch cores such as SC1 will be unblocked, and the windings *b* will be blocked. When the first load character pulse is applied to the input terminal *m* in FIG. 3*a*, simultaneously with the character code for the last column in the next line to be printed, it will be gated to the amplifier **40–43** to pulse the windings *d* of all of the switch cores, restoring those which were in a logical 1 state to the logical 0 state. In this manner, the data for the line last printed is erased just before the first character code for the next line is stored.

While I have described my invention in terms of the specific details of two illustrative embodiments, various changes and modifications will be apparent to those skilled in the art upon reading my description, and such can obviously be made without departing from the scope of my invention.

Having thus described my invention, what I claim is:

1. A code comparator, comprising storage means for storing a first digital code sequence having a predetermined number of bits, code generating means for generating a series of digital scan code sequences of said predetermined number of bits and ordered with each sequence containing *m* logical 0's preceded by all of the sequences having logical 1's in the same positional relationship as the 1's in said digital scan code sequence and at least one additional logical 1, gate means controlled by said storage means for producing an output pulse for each logical 1 in a scan code corresponding to a logical 1 in the stored code, a source of sample pulses synchronized with said scan code sequences, and means for producing an output pulse for each sample pulse for which said gate means produces no output pulse.

2. An ordered sequence comparator, comprising a set of AND gates each having first and second input terminals and an output terminal and producing an output voltage on said output terminal when and only when input voltages are applied to both input terminals, means for applying a code sequence of input voltages to said first terminals of said gates, means for generating a series of scan code sequences of voltages such that each sequence containing *m* logical 0's is preceded by all of the sequences having logical 1's in the same positional relationship as the 1's in said digital scan code sequence and at least one additional logical 1, means for sequentially applying said series of scan code sequences to the second input terminals of the gates, and circuit means controlled by said gates for producing an output pulse for each scan code sequence for which no gate produces an output pulse.

3. A code comparator, comprising storage means for storing a first digital code sequence having a predetermined number of bits, code generating means for generating a series of digital scan code sequences of said pre-

determined number of bits in an order in which each sequence having *m* 0's is preceded by all of the sequences having logical 1's in the same positional relationship as the 1's in said digital scan code sequence and at least one more logical 1 which complement the logical 1's in a possible stored code sequence, gate means controlled by said storage means and said code generating means for producing a pulse for each logical 1 in a scan code corresponding to a logical 1 in the stored code, a source of sample pulses synchronized with said scan code sequences, means controlled by said source and said gate means for producing an output pulse for each sample pulse for which said gate means produces no pulse, and means responsive to the first of said output pulses for producing a command pulse.

4. In combination with a high speed printer of the type in which printing means for each column in a line to be printed is successively conditioned to print one after another of a predetermined series of characters in a fixed time sequence, means for generating an index pulse between the conditioning of the printing means for the last and the first character in said sequence, means for generating a character pulse corresponding to the conditioning of the printing means for each character in said sequence, storage means for storing a character in digital code form for each printing means, means for generating a binary sequence of digital code sequences having a bit for each bit of the character codes in which each sequence having *m* logical 0's is preceded by all sequences having logical 1's in the same positional relationship as the 1's in said digital scan code sequence and at least one more logical 1, time delay means responsive to said index pulse for applying said character pulses to said counter to step it from one sequence to the next a predetermined time after the printing means is conditioned to print each character, a code comparing means for each printing means controlled by said storage means and said counter for producing a pulse for each counter sequence in which the logical 1's are complemented in the associated character code, and means controlled by each code comparing means for actuating the associated printing means to print the character for which it is then conditioned in response to the first pulse produced by the comparing means.

5. In combination with a high speed printer of the type in which printing means for each column in a line to be printed is successively conditioned to print one after another of a predetermined series of characters in a fixed time sequence, means for generating an index pulse between the conditioning of the printing means for the last and the first character in said sequence, means for generating a character pulse corresponding to the conditioning of the printing means for each character in said sequence, storage means for storing a character in digital code form for each printing means, a binary counter for generating a descending binary sequence of digital code sequences having a bit for each bit of the character codes, time delay means responsive to said index pulse for applying said character pulses to said counter to step it from one sequence to the next a predetermined time after the printing means is conditioned to print each character, a code comparing means for each printing means controlled by said storage means and said counter for producing a pulse for each counter sequence in which the logical 1's are complemented in the associated character code, and means controlled by each code comparing means for actuating the associated printing means to print the character for which it is then conditioned in response to the first pulse produced by the comparing means.

References Cited by the Examiner

UNITED STATES PATENTS

2,692,551	10/54	Potter	101—93
3,054,090	9/62	Link et al.	235—177
3,064,561	11/62	Mauduit	101—93
3,072,047	1/63	Maudsley et al.	101—93
3,103,577	9/63	Willard	340—146.2 X

MALCOLM A. MORRISON, *Primary Examiner.*

ROBERT C. BAILEY, *Examiner.*