

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number: **83109787.8**

(51) Int. Cl.³: **G 09 G 1/16**

(22) Date of filing: **29.09.83**

(30) Priority: **01.10.82 US 432323**

(43) Date of publication of application:
18.04.84 Bulletin 84/16

(84) Designated Contracting States:
CH DE FR GB IT LI SE

(71) Applicant: **High Technology Solutions, Inc.**
Box 4738
Poughkeepsie New York(US)

(72) Inventor: **Horne, Gary D.**
RD 1, Box 312A
Highland, New York(US)

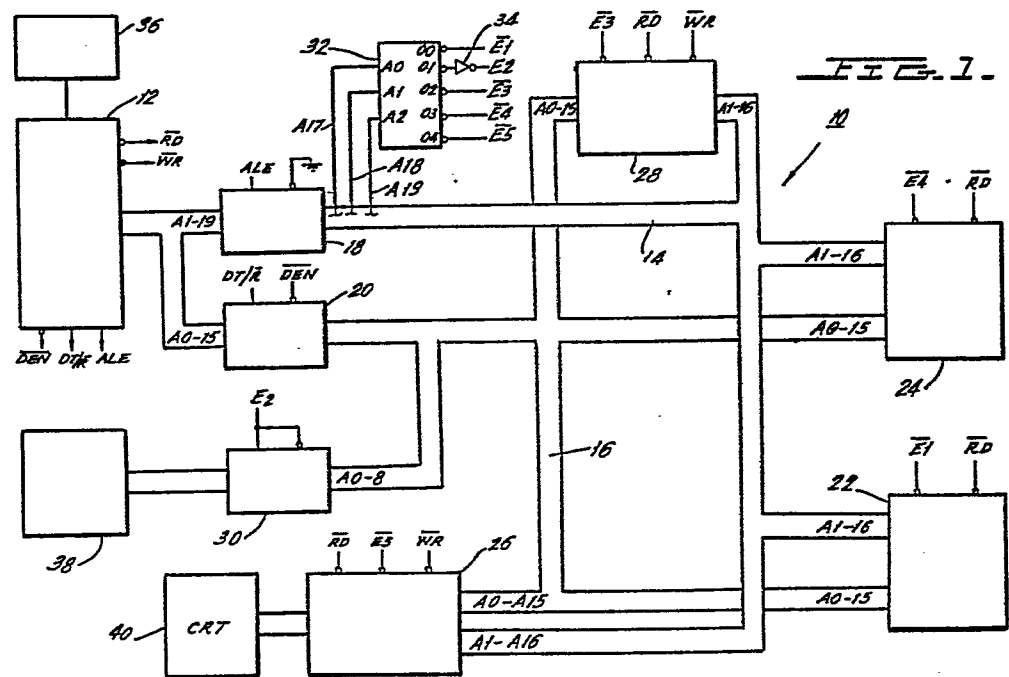
(72) Inventor: **Gottschalk, Martin G.**
Box 220 South Street
Highland, New York(US)

(72) Inventor: **Oehm, Rainer A.**
RD1, Box 300
Wallkill, New York(US)

(74) Representative: **Feldkamp, Rainer, Dipl.-Ing. et al,**
Post Office Box 920
D-8000 München 33(DE)

(54) **Font display and text editing system.**

(57) A font display and text editing system 10 is disclosed. The system includes a display medium 40 on which alphabetical characters may be displayed. A memory 24 stores digital information describing the shape of each alphabetical character of a plurality of sets of alphabetical characters. Each set of alphabetical characters defines a respective font. A human responsive input device 38 permits the user of the system to elect from the sets of alphabetical characters an alphabetical character to be displayed at a specific location on the display medium 40. The input device 38 generates a signal indicative of the alphabetical character selected by the user. A circuit 12-22 and 26-36 responsive to the signal generated by the input device causes the selected alphabetical character to be displayed at the location on the display medium.



- 1 -

Font Display and Text Editing SystemBACKGROUND OF THE INVENTION

The present invention is directed towards an electronic text display system such as are
5 commonly used for word processing, text composing, and the like. The present invention is particularly useful in connection with electronic text composing systems wherein text information including character information defining the text
10 to be presented, font information defining the style of font in which selected portions of the text are to be presented, and composition information such as column spacing, and the like, are all entered into the system and displayed on a
15 display device such as a CRT. The text information is ultimately transferred to an electronic photocomposer which forms photographic negatives which can be used to make printing plates in what is known as the cold-type process. The photographic
20 negatives contain the characters defining the text to be presented in the desired font style and with the desired composition in accordance with the text information which had been entered into the

- 2 -

electronic text composing system.

In prior art text composing systems, such as the Printext Composing System sold by IBM, the set of
5 characters or symbols which can be represented on the display medium is limited to one font style. This is a highly limiting feature since the user cannot obtain an accurate representation of the actual font style which will be ultimately
10 produced on the printed page. While special symbols may be displayed on the display medium to indicate the fact that associated characters will ultimately be presented in a specific font style, the actual characters displayed on the display
15 medium will be formed in a single font style. This requires that the user of the system use his imagination to determine what the final printed page will actually look like. Often a certain composition scheme which appears to be
20 aesthetically pleasing in the single style font displayed on the display medium turns out to be quite unsatisfactory when transferred to the printed page with the actual font style.

25 In an effort to overcome this drawback, at least one composing system manufactured by Compugraphics utilizes a preview screen to provide a detailed and accurate image of different font styles and weights as they will ultimately appear on the
30 printed page. In this system, however, text data is initially, edited and composed on a standard CRT display capable of illustrating only a single font style. Once the user has completed his editing and composition of the page of text, he

- 3 -

can display an accurate image of the composed data on a separate preview screen for review before electronic phototypesetting takes place. The information displayed on the preview screen cannot, however, be edited or recomposed on the preview screen. Accordingly, if the user does not like the composition of the page he must return to the standard CRT display and must recompose the page of data using the single character set.

Accordingly, he cannot be sure that the final composition will be satisfactory until he completes his editing and again transfers the information to the preview screen.

15

BRIEF DESCRIPTION OF THE PRESENT INVENTION

The present invention overcomes the foregoing deficiencies of the prior art by providing a display screen which provides a detailed and accurate image of different fonts and weights and which permits the user of the system to interactively edit and compose the information displayed on the screen. This result is achieved by storing characters for a plurality of font styles in digital form with the shape of each character being described by a unique character set of digital words. The system may be provided with as many text editing and/or composing capabilities as is desired.

In the preferred embodiment, the display medium forms the desired character from a plurality of dots or pixels which are arranged at predetermined

- 4 -

locations in a character cell which may be of constant or variable size. The character cell defines a space on the display medium in which the character may be presented. The character cell
5 is preferably divided into a grid of pixel locations, each of which may contain a single pixel. By placing pixels in only selected pixel locations of the grid, the display medium can produce a character having substantially any form
10 desired.

In the preferred embodiment, the shape of each character is defined by a unique character set comprising a plurality of data words. These data
15 words contain information regarding the locations of the pixels within a character cell which are required to produce the desired character shape. Since the number of characters and font styles which may be reproduced in this manner is limited
20 only by the size and number of pixel elements in a character cell and the size of the font memory holding the data words, presentation of the characters in this pixel matrix form provides for great flexibility in the system.

25 The text editing and composing capability of the system are made possible primarily through the use of a bit mapped RAM which contains storage locations which correspond on a one-to-one basis
30 to the pixel locations on the display medium. In the preferred embodiment, the system is run by a microprocessor which constantly monitors keyboard instructions generated by a user controlled keyboard. These instructions provide information

- 5 -

regarding the particular character and font style which the user wishes to display on the display medium. The keyboard also provides the microprocessor with information which enables the microprocessor to determine the location on the display medium where the character identified by the character code is to be placed. Whenever a new keyboard instruction is generated by the keyboard, the microprocessor removes the character set associated with the character and font identified by the character code from the font memory and stores the data words of that character set in memory locations of the bit mapped RAM which correspond to the location on the display medium where the character is to be displayed. Since the display medium reproduces the pixel information stored in the bit mapped RAM on its own screen, the user can place desired characters from any selected font at any location on the display medium by causing the microprocessor to place the appropriate character sets in the appropriate storage locations of the bit mapped RAM in response to appropriate entries into the keyboard.

By defining the character shape in the form of a unique character set which identifies the pixel location within a character cell required to reproduce that character shape and by placing that information in any desired location of the bit mapped RAM, the present invention makes it possible to incorporate substantially any text editing and composing capability which is used on standard text editing and electronic composing systems. Since these text editing and composition

- 6 -

capabilities are per se known and do not themselves define the inventive features of the present invention, the following detailed description of the invention is directed primarily
5 to the font display features of the system. Relatively few text editing features have been described. It should be understood, however, that any presently known or future developed text editing and/or composing capabilities can be
10 incorporated into the system without departing from the spirit or scope of the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

15 For the purpose of illustrating the invention, there is shown in the drawings an embodiment which is presently preferred, it being understood, however, that the invention is not limited to the
20 precise arrangements and instrumentalities shown.

Figure 1 is a schematic diagram of the hardware of the font display and text editing system of the present invention.

25 Figure 2 is a schematic representation of the CRT of Figure 1.

Figure 3A is a schematic representation
30 illustrating the manner in which a character may be formed by a plurality of pixels located in a character cell.

Figure 3B illustrates the binary words which may

- 7 -

be used to define the character
illustrated in Figure 3A.

5 Figure 4 is a schematic representation of the
display RAM of Figure 1.

Figures 5A, 5B and 5C are flow diagrams
illustrating the main program stored in
the program ROM of Figure 1.

10

Figures 6 and 7 are flow diagrams illustrating
subroutines of the program stored in the
program ROM of Figure 1.

15

DETAILED DESCRIPTION OF THE INVENTION

Referring now to the drawings wherein like
numerals indicate like elements, there is shown in
20 Figure 1 a font display and text editing system
constructed in accordance with the principles of
the present invention and designated generally as
10.

25 The heart of font display and text editing system
10 is a microprocessor 24 which may be an 8086
microprocessor manufactured by Intel Corporation.
A complete description of the structure and
operation of this microprocessor, as well as
30 various applications thereof, is described in
Intel's "iPX 86, 88 User's Manual" dated August
21, 1981. The disclosure of this manual is
incorporated herein by reference.

- 8 -

Throughout the following description, reference will be made to signals which are either active low or active high. An active low signal will be indicated by the presence of a line over the
5 signal (e. g., $\overline{\text{DEN}}$). An active low signal will be referred to as being set or generated when it is at the binary "0" level and reset when it is at the binary "1" level. An active high signal will be referred to as being set or generated when it
10 is at the binary "1" level and being reset when it is at the binary "0" level.

In addition to active low and active high signals, various elements of system 10 have active high and
15 active low inputs and outputs. An active low input or output will be indicated by the presence of a small circle at the input or output of the element. For example, each of the outputs of the 3 to 8 decoder 44 are active low outputs. An active
20 low input will be activated by the presence of a binary "0" on its input. An active low output will place a binary "0" on its output when it is activated. Any input or output which is not indicated to be active low is active high.

25
Microprocessor 12 communicates with the remaining elements of system 10 by writing address information onto address bus 14 and by both writing information onto and reading information
30 off of data bus 16. Microprocessor 12 has a common set of input/output ports A0-A19 which are connected to both address bus 14 and data bus 16 through address latch 18 and transceiver 20, respectively. Whenever microprocessor 12 wishes to

- 9 -

place address information on address bus 14, it generates a binary signal corresponding to the desired address on its output ports A1-A19 (for reasons which will be described below, output port
5 A0 is not used for address purposes) and generates the address latch enable signal ALE which is applied to the strobe input STB of address latch 18. This causes the 19 bit address signal generated by microprocessor 12 to be placed on
10 address bus 14. Since the output enable input $\overline{OE}$ of address latch 18 is grounded, the 19 bit address applied to the input of address latch 18 will remain on address bus 14 until a new address is strobed into latch 18. One suitable address
15 latch is manufactured by Intel Corporation under the product designation 8282 Octal latch. While a single latch 18 is shown, it will be apparent to those skilled in the art that three latches must be used in parallel to latch all 19 outputs of
20 microprocessor 12.

The 16 least significant bits of the address signal are contained on address lines A1-A16 of address bus 14 and are used to address the memory
25 elements 22-28 of system 10. The three most significant bits of the address signal are contained on address lines A17-A19 and are applied to a one of eight decoder 32 which is used to generate chip enable signals which enable only one
30 of the memory elements 22-28 at any given time. One suitable decoder is manufactured by Intel Corporation under the product designation 8205 one of eight decoder. Decoder 32 receives the three lines A17-A19 on its address inputs A0-A2,

- 10 -

respectively, and causes that one of its eight outputs 00-07 (only outputs 00-04 are used in system 10) to be set. Thus, if the binary signal 000 is applied to the inputs of decoder 32, its
5 output 00 will be set (will be placed at the binary "0" level) while the remaining outputs will be reset (will be at the binary "1" level). Similarly, when the binary address signal 001 is applied to the input of decoder 32, its output 01
10 will be set and the remaining outputs will be reset. In this manner, decoder 32 can generate chip enable signals $\overline{E1}$, $E2$ (the 01 output of decoder 32 is inverted by an inverter 34), $\overline{E3}$, $\overline{E4}$ and $\overline{E5}$.

15

Once the appropriate address has been placed on address bus 14, microprocessor 12 can either write data onto data bus 16 or read data on the data bus 16 into its internal memories. This is
20 accomplished through the use of a transceiver 20 which may be a 8286 Octal Bus Transceiver manufactured by Intel Corporation.

In the system 10 described herein, all information
25 is transmitted as either an 11 or 16 bit word. For this reason, only output lines A0-A15 of microprocessor 12 are applied to transceiver 20. Transceiver 20 applies the 16 bits of data contained on the output ports A0-A15 of
30 microprocessor 12 onto data bus 16 whenever the data enable signal $\overline{DEN}$ is applied to its output enable input $\overline{OE}$ and the data transmission signal $DT/\overline{R}$ is at the binary "1" level. When the data enable signal $\overline{DEN}$ is generated but the data

- 11 -

transmit signal $DT/\overline{R}$ is at the binary "0" level, data contained on bus 16 will be applied to ports A0-A15 of microprocessor 12 and will thereby be read into the internal memory of microprocessor 12. Since the 8286 transceiver is an octal transceiver, two transceivers must be connected in parallel to handle all 16 data bits.

Microprocessor 12 controls the operation of font display and text editing system 10 by following a software program stored in program ROM 22. The software program, which will be described below with reference to the flow diagrams of Figures 5-7, is stored in program ROM 22 in machine code as a plurality of 16 bit words. Microprocessor 12 will sequence through the various steps of its program by periodically requesting new program instructions from program ROM 22 at time intervals determined by clock pulses generated by system clock 36. Each time microprocessor 12 needs a new program instruction, it applies that address signal to lines A1-A16 of address bus 14 which identifies the storage location of the desired program instruction, causes decoder 32 to generate the chip enable signal $\overline{E1}$ and generates the read signal $\overline{RD}$. As a result, a 16 bit word containing the desired program instruction(s) will appear on data bus 16. Microprocessor 12 then reads this instruction into its internal memory via transceiver 20.

While any available memory can be used, one suitable program ROM 22 is an 8Kx8 UV erasable PROM sold under the product designation 2764 PROM

- 12 -

by Intel Corporation. Since each PROM can store only 8 bits of information, two PROMs are connected in parallel so that a single address generated by microprocessor 12 is applied to the address inputs of both PROMs and the 8 bit outputs of each PROM are combined to form a single 16 bit word which is applied to data bus 16.

Following the program instructions contained in program ROM 22, microprocessor 12 will cause system 10 to display the shape of specific font characters which are identified by a character code generated by an input device (preferably an electronic keyboard) 38 on a display device (preferably a CRT) 40 with sufficient resolution to permit the user of the system to view an accurate representation of what will appear on the final printed page. In order to attain satisfactory results, it is preferable that the CRT 40 have a resolution of between 800 to 1,100 lines, each divided into between 800 and 1,100 pixel (picture element) locations. Each pixel should be small, on the order of 0.01 inches in diameter, so that the individual characters which are formed by a combination of pixel dots will appear to be smooth and continuous.

In the embodiment disclosed herein, the CRT 40 is divided into 1,024 lines each containing 1,024 pixel locations. This division of the CRT is illustrated graphically in Figure 2 wherein each box 42 represents a single pixel location. It will be understood by those skilled in the art that the grid lines shown in Figure 2 will not actually

- 13 -

appear on the CRT 40 but are rather shown for purposes of explanation. The grid lines merely define areas on the CRT 40 which represent the pixel locations 42. As the electron beam scans the face of the CRT 40, it is modulated in a manner which causes it to excite certain pixel locations 42 but not others. Those pixel locations 42 excited by the electron beam will fluoresce (producing a pixel 44) so as to form the character desired.

In the embodiment disclosed herein, CRT 40 is divided into 64 rows by 64 columns of character cells 46, each of which is 16 pixels wide and 16 pixels deep. A single character can be formed in each character cell 46 so that the CRT 40 can display up to $64 \times 64 = 4,096$ characters. This represents a single page of text.

While a character cell 46, having constant dimensions of 16 x 16 pixels, is disclosed, it should be understood that the invention is not so limited. Thus, character cells of other sizes may also be used. Additionally, it is not necessary that the size of the character cell remain constant, i. e., one character may be stored in a cell 20 x 16 pixels while a second character may be stored in a cell 28 x 18 pixels. Alternatively, the character cell size can remain constant, but characters can be stored in less than and/or more than one cell. While such variations complicate the design of the system, such design modifications are well within the skill of those of ordinary skill in the art.

- 14 -

A character may be displayed in any given character cell 46 by energizing selected pixel locations 42 in that cell. The particular pixel locations 42 which must be energized to form a
5 desired character are defined by a unique set of 16 binary words (hereinafter data words), each 16 bits in length. Each set of 16 data words describes the shape of the character to be displayed and will be referred to hereinafter as
10 character set.

The preferred relationship between the individual words of the character set and the individual pixel locations 42 of a character cell 46 may best
15 be understood with reference to Figures 3A and 3B. Figure 3A illustrates a single character cell 46 containing the letter "S". Figure 3B shows the 16 data words of the character set defining the letter "S". As shown in Figure 3A, the letter "S"
20 is formed by a plurality of pixels 44, each of which is located in a respective pixel location 42. Since pixel row 0 of the character cell 46 contains no pixels 44, data word 1 (Figure 3B) of the character set is represented by the binary
25 number: 0000000000000000. Since pixel row 1 contains pixels 44 at the four pixel locations 42 corresponding to pixel columns 6, 7, 8 and 9, data word 2 is stored as: 000000111100000. This sequence continues through to data word 16 such
30 that the 16 data words of the character set contain all the information required to produce a single character in a single character cell 46. Using this technique, any shape character may be described by a unique character set of 16 data

- 15 -

words each 16 bits in length.

Since the shape of a character to be displayed on CRT 40 may be defined by a 16 word character set,
5 the shape of the characters displayed is limited only by the size of the font ROM memory 24 in which the character sets are stored and the resolution of a 16 x 16 pixel character cell. This provides tremendous flexibility which makes it
10 possible to store a large number of font styles and call up any character of any font style onto the CRT 40 by merely entering appropriate commands in the input device 38. Once the user of system 10 has entered the desired characters, he may then
15 rearrange the position of the characters on the CRT 40 utilizing any text editing capabilities available. As a result, the user is presented with an accurate representation of what will appear on the final printed page.

20

By way of simple example, the user may call up the characters:

SHEM in Hebrew is שֶׁם.

25

In this example, characters from two fonts (a bold Roman font and Hebrew font) have been called up from font ROM 24 onto the CRT 40. After entering this text data, the user may decide it is
30 desirable to write the word SHEM in a second style of font so as to offset the name from the rest of the sentence. In a manner described in detail below, the user will then cause system 10 to replace the bold Roman characters of the word

- 16 -

SHEM with characters from a different style font, for example, Roman script. This is done by writing the script letters *SHEM* over the bold letters SHEM. As a result, the following words will appear
5 on CRT 12:

SHEM in Hebrew is *שׁוּׁן*.

In the following description of font display and
10 text editing system 10, it will be assumed that three font styles ROMAN 1 (a bold Roman type), ROMAN 2 (a script Roman type) and HEBREW are stored in font ROM 24. It should be recognized, however, that a greater number of font styles (as
15 well as different font styles) may be stored.

In the presently preferred embodiment, each font style includes 128 characters comprising the upper and lower case letters of the alphabet,
20 punctuation marks, a blank space and any other characters which are to be displayed on the CRT 40. The font style ROMAN 1 will contain the Roman characters necessary for the English language in a first style, for example, bold. The font ROMAN 2
25 will also contain the Roman characters for the English language, but in a second style, for example, script. The font HEBREW will contain the various Hebrew letters in a desired style.

30 Each character of the three fonts is assigned a unique character code which identifies the address locations of the 16 word character set defining the character in font ROM 24. Since each font contains 128 characters, $128 \times 3 = 384$ character

- 17 -

codes are required. These codes may be expressed as a 9 bit binary word. For example, the lower case letter "a" of the character font ROMAN 1 can be accorded the character code 0 (binary 000000000), the lower case letter "b" of the character font ROMAN 1 can be accorded the character code 1 (binary 000000001), etc. Similarly, the lower case letter "a" of the character font ROMAN 2 can be accorded the character code 128 (binary 010000000), the lower case letter "b" of character font ROMAN 2 can be accorded the character code 129 (binary 010000001), etc. In a similar manner, the lower case letter "aleph" of the character font HEBREW can be accorded the character code 256 (binary 100000000), the lower case "bet" of the character font HEBREW can be accorded the character code 257 (binary 100000001), etc.

In accordance with the foregoing, the 16 words of each character set are preferably stored in ROM memory 24 in the order determined by the character code identifying that character. Thus, the 16 word character set defining the lower case letter "a" of the character font ROMAN 1 is stored at address locations 0-15 of memory 20 while the 16 words of the character set defining the lower case letter "b" of the character font ROMAN 1 is stored at address locations 16-31 of memory 20. All 128 character sets of character font ROMAN 1 are in the first $128 \times 16 = 2,048$ address locations (numbered 0-2,047) of font ROM 24. The 128 character sets of character font ROMAN 2 are stored at address locations 2,048-4,095 of font ROM 24.

- 18 -

Particularly, the 16 word character set describing the lower case "a" of character ROMAN 2 font is stored at address locations 2,048-2,053, while the 16 word character set describing the lower case "b" of character ROMAN 2 font is stored at address locations 2,054-2,069, etc. Finally, the 128 character sets defining the font HEBREW are stored at address locations 4,096-6,143 of font ROM 24. Particularly, the character set describing the lower case "aleph" of the font HEBREW is stored at address locations 4,096-4,111; the character set describing the lower case "bet" is stored at address locations 4,112-4,127, etc.

15 The 9 bit character codes identifying the address locations of the 16 word character sets in font ROM 24 are generated by an input device 38 which may take any known form. In the preferred embodiment, input device 38 is an electronic keyboard and will be described as such. It should be recognized, however, that any other input device (for example, one utilizing menu selection technique or one responsive to voice actuation) could also be used.

25 The keyboard 38 preferably takes the form described in copending application Serial No. (M-9760), filed October 1, 1982, and entitled FONT DISPLAY AND TEXT EDITING SYSTEM WITH CHARACTER OVERLAY FEATURE. This application is assignet to the assignee of the present invention. This keyboard includes both character and command keys. The character keys may be used to identify the characters to be displayed on CRT 40 while the

- 19 -

command keys will identify both the font style in which the character is to be displayed as well as standard command functions such as cursor left, cursor right, carriage return and backspace, etc.

- 5 Keyboard 38 responds to the depression of the character and command keys by generating a keyboard instruction representative of either a character code identifying the character in font RAM 24 to be displayed on CRT 40 or a command code
- 10 identifying the command instruction to be carried out. The keyboard instruction is preferably an 11 bit word which includes a 2 bit format block followed by a 9 bit data block. The format block identifies the data block of the keyboard
- 15 instruction as either a command code or a character code. When the 9 bit data block is a command code, it identifies the specific command instruction to be followed by microprocessor 12. When the 9 bit data block is a character code, it
- 20 identifies the specific character in font ROM memory 22 to be displayed on CRT 40.

- As described in copending application Serial No. (M-9760), the user first depresses one or more
- 25 command keys indicating the character font to be used and then depresses appropriate character keys to cause the desired character of the selected font to appear on CRT 40. Whenever the user wishes to change the font style of subsequent characters
- 30 to be displayed on CRT 40, he depresses one or more command keys identifying the new font to be used and then depresses appropriate character keys to enter the desired characters on the screen.

- 20 -

Microprocessor 12 constantly monitors the output of keyboard 38 (by periodically causing decoder 32 to generate the chip enable signal E2 so as to strobe keyboard latch 30) to determine if a new
5 keyboard instruction has been generated. Keyboard latch 30 receives the 11 bit keyboard instruction generated by keyboard 38 and places it on data lines A0-A10 of data bus 16 whenever it is strobed by enable signal EL. Latch 30 may be formed from a
10 pair of 8282 Octal latches connected in parallel. When the data block of the keyboard instruction is a character code, microprocessor 12 determines where in font ROM 24 the 16 word character set identified by that character code is located and
15 causes the 16 words of the character set to be placed in display RAM 26. Display RAM 26 then causes this character to be generated in the appropriate character cell 46 of CRT 40.

20 As noted above, each data word of each character code stored in ROM memory 24 is 16 bits in length. A font ROM 24 constructed of commercially available devices such as those described above with references to program ROM 22 can store these
25 data words at sequential locations. Whenever microprocessor 12 wishes to read a data word of the selected character set from font ROM 24, it places the appropriate address on lines A1-A16 of address bus 14, causes decoder 32 to generate the
30 chip enable signal $\overline{E4}$ and simultaneously generates the read signal $\overline{RD}$. This will cause the 16 bit data word to appear on bus 16 which can then be read into an internal memory of microprocessor 12. This data word is then transferred to the

- 21 -

appropriate storage location in display RAM 26 by placing the appropriate address signal on lines A1-A16 of address bus 14 and placing the data received from font ROM 24 on lines A0-A15 of data bus 16. At the same time that the address and data information is placed on the address and data buses 14, 16, microprocessor 12 will simultaneously cause decoder 32 to generate the chip enable signal $\overline{E5}$ and also generate the write signal $\overline{WR}$. This causes display RAM 26 to read the 16 bit data word on bus 16 into the address storage location identified by the address on bus 14. At certain times, it is necessary for microprocessor 12 to read specific data words out of display RAM 26. This is accomplished by placing an appropriate address on address bus 14, causing decoder 32 to generate the chip enable signal $\overline{E5}$ and simultaneously generating the read signal $\overline{RD}$.

As shown in Figure 4, the display RAM 26 is broken up into 1,024 x 1,024 pixel locations 44' which correspond on a one-to-one basis to pixel locations 44 on the CRT 40. One commercially available unit which incorporates the bit mapped display RAM, the CRT display and the necessary drive circuitry to cause the pixel information stored in the display RAM to be reproduced on the display is a model GMDM-1000 bit mapped high resolution CRT display manufactured by Image Automation Inc. The 16 bit data words read from font ROM 24 are written into display RAM 26 16 bits at a time. As such, 16 consecutive pixel locations 44' define a single address location of display RAM 26. Consecutive address locations are

- 22 -

located adjacent one another in a pixel row. Thus, address location 000 is the first storage location in pixel row 0, address location 001 is the second storage location in pixel row 0, etc. Since the
5 RAM memory 26 is 1,024 pixels wide, and since each data word is 16 pixels in length, each pixel row of display RAM 26 contains 64 data words at address locations 000-063. The 64th data location is located at the leftmost end of pixel row 1 with
10 64 data words being stored at address locations 63-127 of row 1. In a similar manner, 64 data words will be stored in each of the 1,024 pixel rows of display RAM 26.

15 The memory space of display RAM 26 is logically divided into character cells 46' which correspond on a one-to-one basis to character cells 46 of CRT 40. Thus, the character cell 46' located in the upper left-hand corner of display RAM 26
20 corresponds to the character cell 46 located in the upper left-hand corner of CRT 40. In accordance with this protocol, the character cell 46' located in the upper left-hand corner of display RAM 26 will contain the 16 word character
25 set which defines the character to be displayed in the upper left-hand character cell 46 of CRT 40. The 16 data words of the character set are stored at storage locations 0, 64, 128, 192 ... 1,024 of display RAM 26.

30 Display RAM 26 will automatically apply appropriate biasing signals (e. g., vertical sync, horizontal sync and data stream) to the CRT 40 so as to cause the CRT 40 to display the character information

- 23 -

stored in display RAM 26. Thus, as information is written into display RAM 26, it is, for practical purposes, simultaneously displaced on CRT 40.

5 In order to identify the particular character cells 46, 46' in both CRT 40 and display RAM 26, both CRT 40 and display RAM 26 are logically broken up into 64 cell columns and 64 cell rows. Referring to Figures 2 and 3, the cell columns are
10 numbered 0-63 as are the cell rows. As such, each character cell 46, 46' has a unique set of coordinates. For example, the letter "S" illustrated in Figure 2 is displayed in the character cell 46 located at cell row 0, cell
15 column 0; the letter "H" is displayed in the character cell 46 located at cell row 0, cell column 1, etc.

The character cell 46 in which the next character
20 identified by the character code generated by keyboard 38 is to be placed will be referred to as the "active" character cell. Microprocessor 12 keeps track of the location of the active character cell by storing cell row and column
25 pointers CR, CC, respectively, in scratch pad RAM 28. The microprocessor 12 identifies the location of the active cell to the user of system 10 by generating a cursor 48 in the active cell. In the preferred embodiment, cursor 48 takes the form of
30 a line of pixels 44 located in the lowermost line of the active character cell 46. Each time keyboard 38 generates a new character code identifying a character to be placed in the active character cell, microprocessor 12 moves the cursor

- 24 -

48 one character cell 46 to the right. When the
cursor is located in the last character cell 46 in
a given row, microprocessor 12 moves the cursor to
the leftmost character cell 46 of the next cell
5 row.

In the embodiment disclosed herein, the location
of the active cell, and therefore the position of
cursor 48, can also be moved to the left or to the
10 right in response to cursor left or cursor right
command signals, respectively. If desired, cursor
up and cursor down command signal could also be
provided as well as any other cursor movements
common to text editing and photocomposing
15 apparatus. When the position of is changed in
response to cursor left or cursor right commands,
cursor 48 is moved without causing the character
located in the character cells 46 traversed by
cursor 48 to be removed from CRT 40. In contrast,
20 when the cursor 48 is moved to the left or to the
right in response to space or backspace commands,
the characters stored in the character cells 46
traversed by cursor 48 will be erased by
microprocessor 12. The position of the cursor 48
25 can also be changed in response to a carriage
return command signal generated by keyboard 38. In
this case, microprocessor 12 causes the cursor to
be moved to the leftmost character cell 46 in the
next succeeding cell row. Again, movement of the
30 cursor 48 into this new character cell 46 will not
cause the character stored in the cell 46, if any,
to be erased.

As noted above, microprocessor 12 continually

- 25 -

strokes keyboard latch 30 to determine if a new keyboard instruction has been generated by keyboard 38. Microprocessor 12 will enter new character information in display RAM 26 and/or
5 move the location of the cursor 48 in response to these keyboard instructions. Microprocessor 12 also stores each data word corresponding to a character code generated by keyboard 38 in text buffer and scratch pad RAM 28 at a memory location
10 corresponding to the character cell 46' in which the character identified by that character code is stored. In this manner, text buffer and scratch pad RAM 28 contains character code information corresponding to all of the characters stored in
15 display RAM 26. While any appropriate memory can be used for text buffer and scratch pad RAM 28, one suitable memory is an 8,192 x 8-bit integrated RAM which is sold by Intel Corporation under the product designation 2186 RAM. Since
20 each 2186 RAM stores 8 bit words, and since microprocessor 12 places either 9 or 16 bit words into RAM 28, two 2186 RAMs must be connected in parallel. Both address inputs of the 2186 RAMs will receive address lines A1-A16 of address bus
25 14 while the data output of one of the RAMs will be connected to the data lines A0-A7 and the data outputs of the remaining RAM will be connected to the lines A8-A15 of data bus 16.

30 The information stored in RAM 28 can be used to refresh the memory in display RAM 26 whenever necessary. Additionally, once an entire page of information has been stored in display RAM 26, it must be cleared to enter a new page of

- 26 -

information. At this time, the character code stored in text buffer and scratch pad RAM 28 may be transferred to a larger, more permanent mass memory (not shown) such as a floppy disk or hard
5 disk. In this manner, character code information for a plurality of pages may be stored in the mass memory. This information may be recalled at any time and may be also used to transfer character code information to a phototypesetter which
10 creates photographic negatives of a printing plate from the character code information. The phototypesetter will contain font information corresponding to that stored in font ROM 24 so that the characters produced by the phototype-
15 setter take substantially the same shape as those displayed on CRT 40. While the manner in which information is transferred from text buffer and scratch pad RAM 28 to the mass memory is not described herein, such information transfer
20 procedures are well known to those of ordinary skill in the art. Exemplary mass storage media and message for transferring information from a temporary memory to such media are described in the SYSTEMS DATA CATALOG, dated January 1982, and
25 published by Intel Corporation. The disclosure of this catalogue is incorporated herein by reference.

As noted above, the least significant bit of the
30 address generated by microprocessor 12 (which bit is located on output port A0) is not placed on address bus 14. As a result, the address actually received by memories 22-28 is equal to the address generated by microprocessor 12 divided by two. The

- 27 -

reason for this somewhat unusual arrangement is particular to the structure and operation of the 8086 microprocessor.

5 As described in some detail in the "iAPX 86, 88
User's Manual", the 8086 microprocessor can access
either 8 or 16 bits of memory at a time. Whenever
the 8086 microprocessor wishes to access a 16 bit
word in memory in a single bus cycle, it must
10 generate an even number address (i. e., 2, 4,
6, ...) on its output ports A-A19. Whenever it
generates an odd number address, the 8086
microprocessor must access the external memories
one 8 bit byte at a time in two consecutive bus
15 cycles. Since such 8 bit byte addressing is not
required by the remaining elements of system 10,
and since the use of 8 bit byte addresses
complicates the design of the system 10, it is
preferred that the microprocessor generate only
20 even numbered addresses.

While it is preferable for microprocessor 12 to
generate even number addresses, it would be
wasteful not to use the odd address locations in
25 the memories 22-28. This problem is simply solved
by not connecting address line A0 (which contains
the least significant bit of the address
generated by microprocessor 12) to the address bus
14. The effect of the foregoing is that
30 microprocessor 12 will generate even addresses
only but the memory elements 22-28 of the system
10 will receive both odd and even addresses. Thus,
the addresses 2, 4, 6, 8, etc. generated by
microprocessor 12 will be applied to address bus

- 28 -

14 as addresses 1, 2, 3, 4, etc.

The operation of font display and text editing system 10 will now be described with reference to
5 Figures 5-7 which show the program stored in program ROM 22 in flow chart form. The main program is illustrated in Figures 5A, B and C. Two subroutines are illustrated in Figures 6 and 7.

10 The main program starts at instruction block 100 which instructs microprocessor 12 to clear both display RAM 26 and text buffer and scratch pad RAM 28. At the same time, the character codes previously stored in text buffer 28 may be
15 transferred to a larger, more permanent mass memory for later retrieval and ultimately for transfer to a phototypesetting machine. Once the RAMs 26 and 28 have been cleared, microprocessor 12 proceeds to instruction block 102 and sets the
20 cell row pointer CR and cell column pointer CC to zero. These pointers define the character cell 46 located at the upper left-hand corner of CRT 40 as the active character cell.

25 Microprocessor 12 then proceeds to instruction block 104 which tells it to go to cursor subroutine 200 and return. Subroutine 200 causes a cursor 48 to be placed at the bottom of the active character cell 46 identified by pointers CR and
30 CC.

Referring to Figure 6, instruction block 202 causes microprocessor 12 to calculate and store the following value in its internal register B:

- 29 -

$$\text{REG B} = (\text{CR} \times 64 \times 16 + 64 \times 15 + \text{CC})2 \quad \text{Eq. 1}$$

Since there are 64 x 16 storage locations in each cell row of display RAM 26, the terms of equation 5 1 which are located in brackets define the address of the last data word of the character cell 46' of display RAM 26 which corresponds to the active character cell 46 of CRT 40. This address is multiplied by two since the address generated by 10 microprocessor 12 must be twice the address which appears on address bus 14 (it should be remembered that the least significant bit of the address generated by microprocessor 12 is not applied to address bus 14 since the output port A0 of 15 microprocessor 12 is not connected to address latch 18).

Upon completion of the foregoing calculation, microprocessor 12 proceeds to instruction block 20 204 and reads the data word stored in display RAM 26 at the display RAM address DR ADD = REG B/2, inverts the word and writes the inverted word back into the display RAM 26 at REG B/2. The result of the foregoing is that a cursor line 48 is placed 25 at the bottom of the upper left-hand character cell 46 of CRT 40. At this point, microprocessor 12 returns to the main program.

Referring again to Figure 5A, microprocessor 12 30 proceeds to decision block 106 which requires that it determine if a keyboard instruction is being generated by keyboard 38. Microprocessor 12 makes this determination by periodically strobing keyboard latch 30. Once microprocessor 12 has

- 30 -

determined that a data word has been generated by keyboard 38, it proceeds to decision block 108 and determines if the 9 bit data block of the keyboard instruction is a command code (this is done by
5 examining the two bit format block of the keyboard instruction). If it is, microprocessor 12 proceeds to instruction block 128 which is illustrated in Figure 5B. In such a case, microprocessor 12 moves the cursor in a manner determined by the command
10 code following the various program steps illustrated in Figures 5B and 5C. This action will be described below.

If the keyboard instruction is not a command code,
15 it must be a character code identifying a character to be displayed in the active character cell 46. When the keyboard instruction is not a command code, microprocessor 12 proceeds to instruction block 110 and sets its internal
20 register A equal to the character code set forth in the data block of the keyboard instruction. This code identifies the position in font ROM 24 of the character set describing the letter identified by the character code. Microprocessor
25 12 then proceeds to instruction block 112 and stores the character code in text buffer and scratch pad RAM 28 at the following text buffer address:

$$30 \qquad \qquad \qquad \text{TB ADD} = \text{CR} \times 64 + \text{CC} \qquad \qquad \qquad \text{Eq. 2}$$

Since there are 64 cell rows in each cell column, this calculation causes the character code identified by keyboard 38 to be stored in text

- 31 -

buffer and scratch pad RAM 28 at an address location corresponding to the particular character cell 46 in which the character identified by the character code is displayed on CRT 40.

5

Microprocessor 12 now proceeds to instruction block 14 which causes it to go to subroutine 300 and then return to the main program. Display character subroutine 300 is illustrated in Figure 7 and causes the character associated with the character code identified by keyboard 38 to be placed in the active character cell 46' of display RAM 26. This, in turn, causes the character to be displayed in the active character cell 46 of CRT 40.

15

Referring to Figure 7, microprocessor 12 first proceeds to instruction block 302 which causes microprocessor 12 to set its internal register B with the following number:

20

$$\text{REG B} = \text{REG A} \times 16 \times 2$$

Eq. 3

This calculation results in a number being stored in register B which corresponds to the address location in font ROM 24 where the first data word of the character set identified by the character code identified by keyboard 38 is located. Again, it should be remembered that the multiplicand 2 is used in equation 3 to ensure that the address generated by the microprocessor 12 is twice address received by font ROM 24.

25

30

Microprocessor 12 then proceeds to instruction

- 32 -

block 304 and sets its internal register C with the following number:

$$\text{REG C} = \text{CR} \times 64 \times 16 \times 2 + \text{CC} \times 2 \quad \text{Eq. 4}$$

5

Since there are 64 x 16 address locations in each cell row of display RAM 26, equation 4 identifies the address in display RAM 26 of the first data word of the active character cell 46'. Again, the address generated by microprocessor 12 is double the actual address signal applied to display RAM 26 since the least significant bit of the address generated by microprocessor 12 is not applied to address bus 14.

15

Microprocessor 12 now proceeds to instruction block 306 which causes microprocessor 12 to set the incremental pointer IP = 0. This number can be stored in an appropriate storage location of text buffer and scratch pad RAM 28.

20

Proceeding to instruction block 308, microprocessor 12 then reads the word stored in font ROM 24 at the following font ROM address and stores the word in its internal register D:

25

$$\text{FR ADD} = \text{REG B}/2 + \text{IP} \quad \text{Eq. 5}$$

Since the incremental pointer IP is zero, microprocessor 12 reads the first word of the character set which corresponds to the character identified by the keyboard instruction generated by keyboard 38 from font ROM 24 into register D. Microprocessor 12 then proceeds to instruction

30

- 33 -

block 310 and writes the word stored in register D into the display RAM 26 at the following display RAM address:

$$\begin{array}{lll} 5 & DR\ ADD = REG\ C/2 + IP \times 64 & Eq. 6 \end{array}$$

Since the incremental pointer IP is set at zero, microprocessor 12 will write the data word stored in register D into the display RAM 26 at the display RAM address corresponding to the first address of the active character cell 46'. Proceeding to instruction block 312, microprocessor 12 increases the incremental pointer by one and then proceeds to decision block 314. If the incremental pointer is less than 16, microprocessor 12 returns to instruction block 308 and reads the data word located in the next address location of font ROM 24 (since incremental pointer IP now is equal to 1) into register D. This data word is then read into the second storage location of the active character cell 46' and the incremental pointer is again increased by one. This process repeats itself 16 times with the result that the 16 data words of the character set corresponding to the character code generated by keyboard 38 are placed in the 16 storage locations of the active character cell 46' of display RAM 26. Simultaneously, display RAM 26 causes this character to appear in the active character cell 46 of CRT 40. Once microprocessor 12 has stepped through instruction blocks 308-312 16 times, the incremental pointer will be equal to 16 and microprocessor 12 will return to the main program.

- 34 -

Referring again to Figure 5A, microprocessor 12 proceeds to instruction block 116 and increases the cell column pointer CC by one. Proceeding to instruction block 118, microprocessor 12
5 determines if the cell column pointer is equal to 64. If it is less than 64, microprocessor 12 proceeds directly to decision block 122. If the cell column pointer is equal to 64, this indicates that the cursor has moved off the
10 right-hand edge of CRT 40 and must be reset at the leftmost character cell 46' of the next cell row. To this end, microprocessor 12 proceeds to instruction block 120 wherein it sets the cell column pointer at zero and increases by the cell
15 row pointer by one.

Microprocessor 12 then proceeds to decision block 122 and determines if the cell row is equal to 64. If it is, this indicates that an attempt has been
20 made to drop the cursor below the bottom edge of CRT 40. Since this is an invalid condition, microprocessor 124 causes the generation of a tone (this may be done in any known manner) which alerts the user of system 10 to the invalid
25 condition. Microprocessor 12 then proceeds to instruction block 125 which resets the cursor row and cursor column points at 63 and writes a cursor 48 in that cell (see block 126). The program then returns to decision block 106 wherein the
30 microprocessor waits for another keyboard instruction to be generated by keyboard 38.

If the cell row is less than 64, microprocessor 12 proceeds to instruction block 126 which directs it

- 35 -

to go to subroutine 200 and return. As a result, a cursor will appear at the bottom of the active character cell 46 identified by the cursor row and cursor column pointers. At this point,
5 microprocessor 12 returns to decision block 106 and waits for an additional keyboard instruction to be generated by keyboard 38.

The manner in which microprocessor 12 responds to
10 a command code generated by keyboard 38 will now be described with reference to Figures 5B and 5C. After microprocessor 12 has determined that the keyboard instruction generated by keyboard 38 is a command code (see decision block 108), it proceeds
15 to instruction block 128 which causes it to go to cursor subroutine 200 and return. This causes the cursor 48 to be removed from the active character cell 46 identified by the cell row and cell column pointers.

20

Proceeding to decision block 130, microprocessor 12 determines if the keyboard instruction is a cursor right command code. If it is, microprocessor 12 proceeds to instruction block
25 132 and increases the cell column pointer by one. Proceeding to decision block 134, microprocessor 12 determines if the cell column pointer is equal to 64. If it does equal 64, the cursor 48 cannot be moved further to the right in the present cell
30 row. Rather, it must be moved to the leftmost character cell 46 of the following cell row. To this end, microprocessor 12 sets the cell column pointer at zero and increases the cell row pointer by one as shown in instruction block 136. If the

- 36 -

cell column pointer was less than 64, or if it was equal to 64 and has been reset in accordance with instruction block 136, microprocessor 12 proceeds to decision block 138 and determines if the cell
5 row pointer is equal to 64. If it is, this indicates that an attempt has been made to move the cursor 48 off of the bottom right-hand corner of CRT 40. Since this is an invalid condition, microprocessor 12 causes the generation of a tone
10 (see instruction block 140), sets the cell column pointer at 63 and reduces the cell row pointer by one (see instruction block 142). This has the effect of moving the cursor 48 to the last cell column in the last cell row of CRT 40 once
15 microprocessor 12 advances to instruction block 144. If the cell row pointer does not equal 64 (see decision block 138), microprocessor 12 proceeds to instruction block 144 and places a cursor 48 on the bottom of the active character
20 cell 46.

Returning to decision block 130, when microprocessor 12 determines that the keyboard instruction is not a cursor right command code, it
25 proceeds to decision block 146 and determines if it is a cursor left command code. If it is, it proceeds to instruction block 148 and reduces the cell column pointer by one. Microprocessor 12 then determines if the cell column pointer is less than
30 zero (see decision block 150). If it is not, microprocessor 12 proceeds to instruction block 152 which causes a cursor 48 to be placed in the bottom of the active character cell 46 defined by the modified cell column pointer. At this point,

- 37 -

the program returns to decision block 106 and the microprocessor 12 waits for the next data word generated by keyboard 38.

- 5 Returning to decision block 150, if the cell column pointer is less than zero, this indicates that an attempt has been made to move the cursor 48 off the left-hand side of the CRT 40. Accordingly, the cursor 48 must be moved up one
- 10 cell row and must be moved to the rightmost column. To this end, microprocessor 12 resets the cell column pointer at 63 and decreases the cell row pointer by one (see instruction block 154). Proceeding to decision block 156, microprocessor
- 15 12 determines if the cell row pointer is less than zero. If it is, this indicates that an attempt has been made to move the cursor to a point above the top cell row of the CRT 40. Since this is an invalid condition, microprocessor 12 generates a
- 20 tone (see instruction block 158), resets the cell column pointer to zero and increases the cell row pointer by one. This has the effect of placing the cursor at the bottom of the upper left-hand character cell 46 of CRT 40 once the program
- 25 proceeds to instruction block 162. At this point, the program will return to decision block 106 and microprocessor 12 waits for the next keyboard instruction to be generated by keyboard 38.
- 30 Returning to decision block 156, if the cell row pointer is not less than zero, microprocessor 12 proceeds to instruction block 162 and places a cursor 48 at the bottom of the character cell 64 identified by the cell column and cell row

- 38 -

pointers. Thereafter, the program returns to decision block 106 and microprocessor 12 awaits the next keyboard instruction generated by keyboard 38.

5

Returning to decision block 146 (see Figure 5B), if microprocessor 12 determined that the keyboard instruction is not a cursor left command signal, it proceeds to decision block 164 (see Figure 5C).

10

In accordance with decision block 164, microprocessor 12 determines if the keyboard instruction is a carriage return command code. If it is, microprocessor 12 proceeds to decision block 166 and determines if the cell row pointer is equal to 63. If the cell row pointer is equal to 63, the carriage return command code is attempting to place the cursor 48 below the bottom edge of CRT 40. Since this is an invalid condition,

15

microprocessor 12 causes the generation of a tone (see block 168) to alert the user of the invalid condition. Since the cursor 48 had been removed in instruction block 128, it must now be replaced. To this end, instruction block 172 requires that microprocessor 12 go to cursor subroutine 200 and return. Microprocessor 12 then returns to decision block 106 where it awaits the next keyboard instruction generated by keyboard 38.

20

25

If the cell row pointer is less than 63, microprocessor 12 proceeds to instruction block 170 and sets the cell column pointer at zero and increases the cell row pointer by one.

30

Microprocessor 12 then proceeds to instruction block 172 with the result that a cursor 48 is

- 39 -

placed in the first character cell 46 in the next cell row. The program is then returned to decision block 106 wherein microprocessor 12 again awaits the next keyboard instruction from keyboard 38.

5

Returning to decision block 164, if microprocessor 12 determines that the keyboard instruction is not a carriage return command code (it has already determined that it is not a cursor left or a
10 cursor right command code), it must be a backspace command code since there are only four commands in the system disclosed herein.

Proceeding to decision block 174, microprocessor
15 12 determines if the cell row and cell column pointers are both equal to zero. If they are, the backspace command is an invalid command. In such a case, microprocessor 12 generates a tone (see instruction block 176) and causes a cursor 48 to
20 be placed in the bottom of the character cell 46 located in the upper left-hand corner of CRT 40 (see instruction block 178). The program then returns to decision block 106 and the microprocessor again awaits the next keyboard
25 instruction generated by keyboard 38.

Returning to decision block 174, if the cell row and cell character pointers are not both equal to zero, microprocessor 12 proceeds to decision block
30 180 and determines if the cell column pointer is equal to zero. If it is not, microprocessor 12 reduces the cell column by one (see instruction block 182) and proceeds to instruction block 186. If the cell column pointer does equal zero, the

- 40 -

cell column pointer is reset to 63 and the cell
row pointer is decreased by one. See instruction
block 184. This has the effect of moving the
pointer to the rightmost edge of the next higher
5 row.

Since the backspace command should erase any
character stored in the newly active character
cell 46, microprocessor 12 proceeds to instruction
10 block 186 and sets its internal register A for the
character code for a blank. Proceeding to
instruction block 188, microprocessor 12 jumps to
character subroutine 300 and writes the blank in
the newly active character cell 46. Microprocessor
15 12 then proceeds to instruction block 190 where it
goes to cursor subroutine 200 with the result that
a cursor 48 is placed in the bottom of the active
character cell 46 identified by the updated cell
column and cell row pointers. The program then
20 returns to decision block 106 and microprocessor
12 awaits generation of an additional keyboard
instruction by keyboard 38.

As used in the following claims, the term
25 "alphabetical characters" shall be interpreted as
including alphanumeric characters as well as
idiographic characters.

The present invention may be embodied in other
30 specific forms without departing from the spirit
or essential attributes thereof and, accordingly,
reference should be made to the appended claims,
rather than to the foregoing specification as
indicating the scope of the invention.

- 41 -

What is claimed is:

1. A font display and text editing system,
comprising:

- a display medium;

5

- a memory storing digital information
describing the shape of each alphabetical
character of a plurality of sets of
alphabetical characters, each said set of
10 alphabetical characters defining a respective
font;

- a human responsive input device which
permits the user of said system to select
15 from said sets of alphabetical characters an
alphabetical character to be displayed at a
specific location on said display medium,
said input device generating a signal
indicative of the alphabetical character
20 selected by said user; and

- a circuit responsive to said signal for
displaying said selected alphabetical
character at said location in said display
25 medium.

2. The system of claim 1, wherein characters from
different said sets of alphabetical characters
can be displayed on said display medium
30 simultaneously.

3. The system of claim 1, wherein said input

- 42 -

device and said circuit cooperate to permit the position of said characters displayed on said display device to be changed.

- 5 4. The system of claim 1, wherein said circuit reacts to the generation of two successive signals generated by said input device, said two successive signals respectively identifying first and second characters which
10 form part of different said sets of alphanumeric characters, by simultaneously displaying said first character at a first location on said display medium and said second character at a second location on said
15 display medium.
- 20 5. The system of claim 1, wherein each character is stored in said memory as a unique set of binary numbers which describe the shape of that character.
- 25 6. The system of claim 5, wherein the shape of each character is defined by an array of n rows and m columns of binary numbers, each number indicating whether or not a pixel is to appear at a corresponding pixel location on said display medium, n and m being positive integers.
- 30 7. The system of claim 6, wherein said array is stored as n binary words, each word including m bits of information.
8. The system of claim 7, in which each word of a

- 43 -

given character is stored in sequential locations in said memory.

- 5 9. The system of claim 8, wherein each character is assigned a unique character code which identifies the storage location of said memory at which the first word of that character is stored.
- 10 10. The system of claim 9, wherein said signal generated by said input device identifies the character code of the character selected by said user.
- 15 11. The system of claim 10, wherein said circuit responds to said signal by sequentially reading each word of the character identified by said character code out of said memory and displaying the corresponding character on said display medium.
- 20 12. The system of claim 1, wherein said display device is able to display one page of text at a time and wherein said system further includes a second memory for storing each of the signals generated by said input device and corresponding to said one page of text displayed on said display device.
- 25 13. The system of claim 12, wherein said circuit erases the page of information displayed on said display device responsive to an appropriate control signal generated by said input device, and wherein said system further
- 30

- 44 -

includes means for transferring all of said signals stored in said second memory into a mass memory when said circuit erases said page of information displayed on said display device.

- 5
14. The system of claim 1, wherein said circuit includes:
- 10 - a bit mapped RAM which contains an array of g by p storage location and wherein said display device is divided into g by p pixel locations, said storage locations corresponding to said pixel locations on a one-to-one basis, g and p being positive integers much greater than 1; and
 - 15 - means for displaying, on said display device, the information stored in said bit mapped RAM.
- 20 15. The system of claim 14, wherein $800 \leq g \leq 1100$ and $800 \leq p \leq 1100$.
- 25 16. The system of claim 15, wherein said display device is a cathode ray tube divided into 800 to 1100 lines of information, each line containing 800 to 1100 pixel locations such that each pixel location corresponds to a corresponding one of said storage locations on a one-to-one basis.
- 30 17. The system of claim 15, wherein each character is stored in said memory as a unique set of binary numbers which describe the shape of that character.

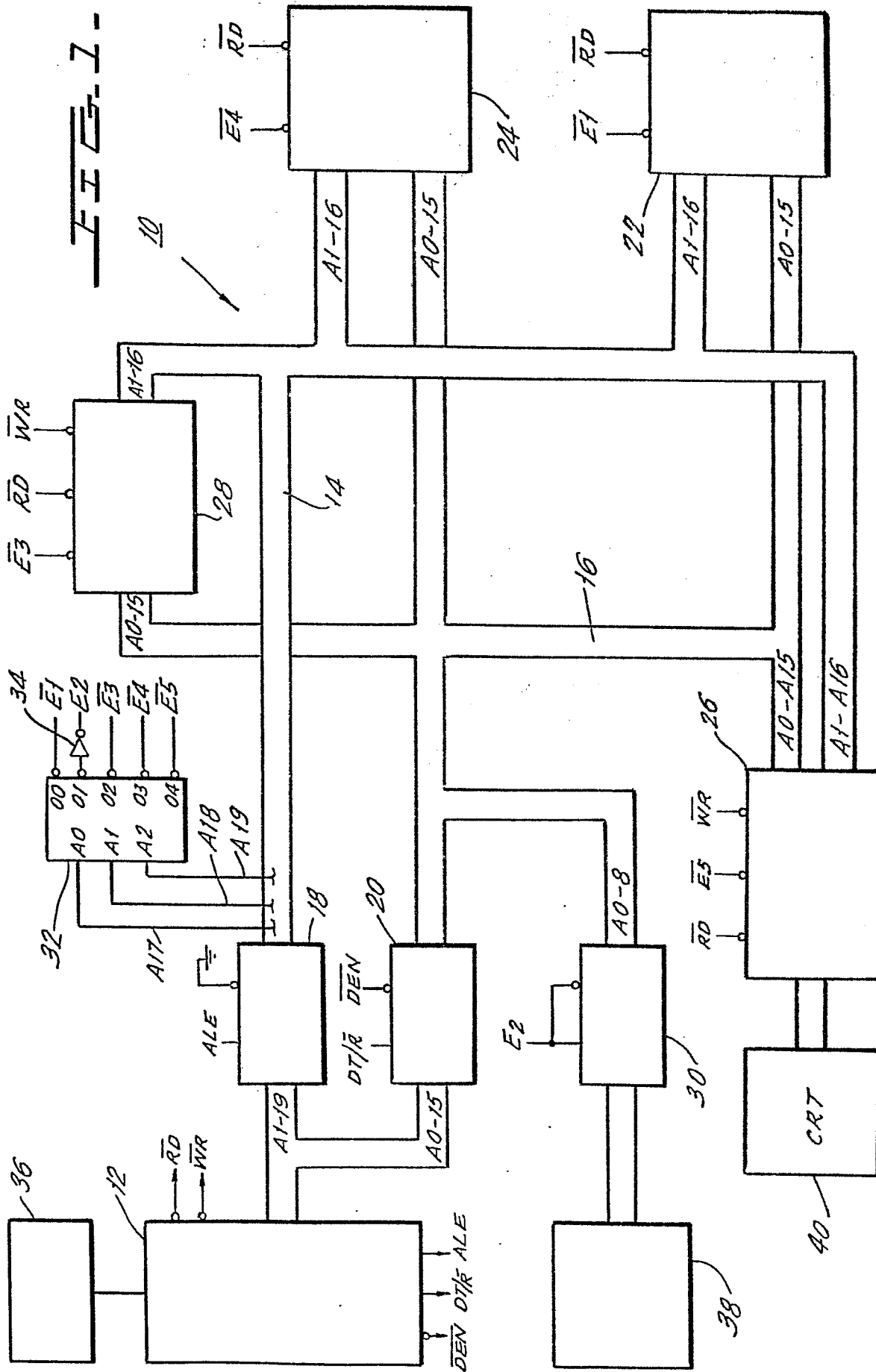
- 45 -

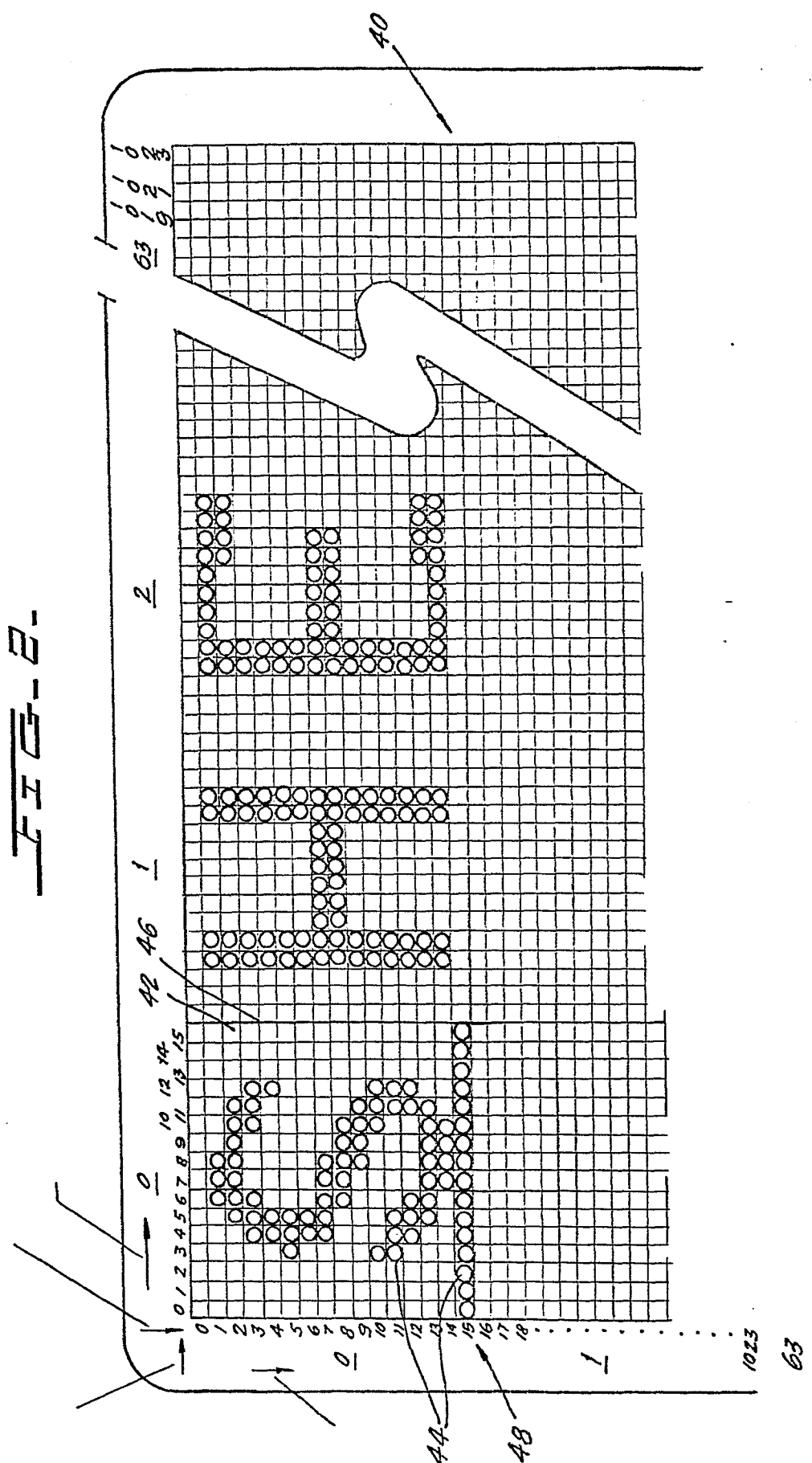
18. The system of claim 17, wherein the shape of each character is defined by an array of n rows and m columns of binary numbers, each binary number indicating whether or not a pixel is to appear at one or more corresponding pixel locations on said display medium, n and m being positive integers much less than p and g , respectively.
19. The system of claim 18, wherein said array is stored as n binary words, each word including m bits of information.
20. The system of claim 19, wherein each word of a given character is stored in sequential locations in said memory.
21. The system of claim 20, wherein each character is assigned a unique character code which identifies the storage location of said memory at which the first word of that character is stored.
22. The system of claim 21, wherein said signal generated by said input device identified the character code of the character selected by said user.
23. The system of claim 22, wherein said circuit responds to said signal by sequentially reading each word of the character identified by said character code out of said memory and storing corresponding bits of information in corresponding storage locations of said bit

- 46 -

mapped RAM.

24. The system of claim 14, wherein characters
from different said sets of alphabetical
5 characters can be displayed on said display
medium simultaneously.
25. The system of claim 14, wherein said input
device and said circuit cooperate to permit
10 the user of said system to change the position
of said characters displayed on said display.
26. The system of claim 14, wherein said circuit
reacts to the generation of two successive
15 signals generated by said input device, said
two successive signals respectively identifying
first and second characters which form part of
different said sets of alphanumeric characters
by simultaneously displaying said first
20 character at a first location on said display
medium and said second character at a second
location on said display medium.

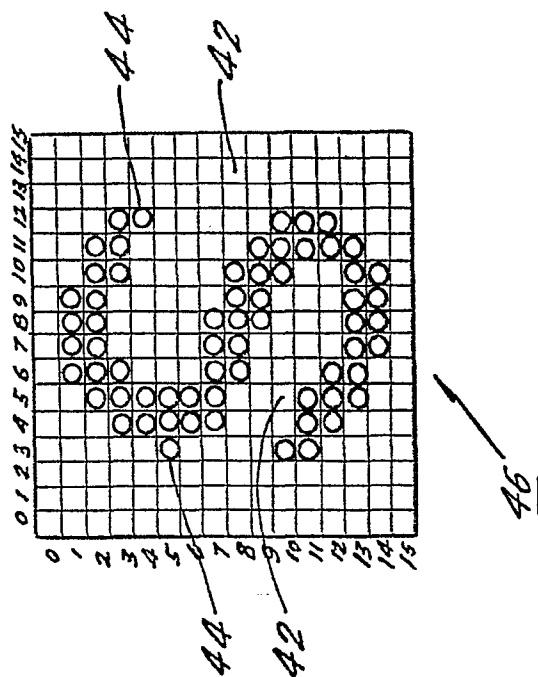




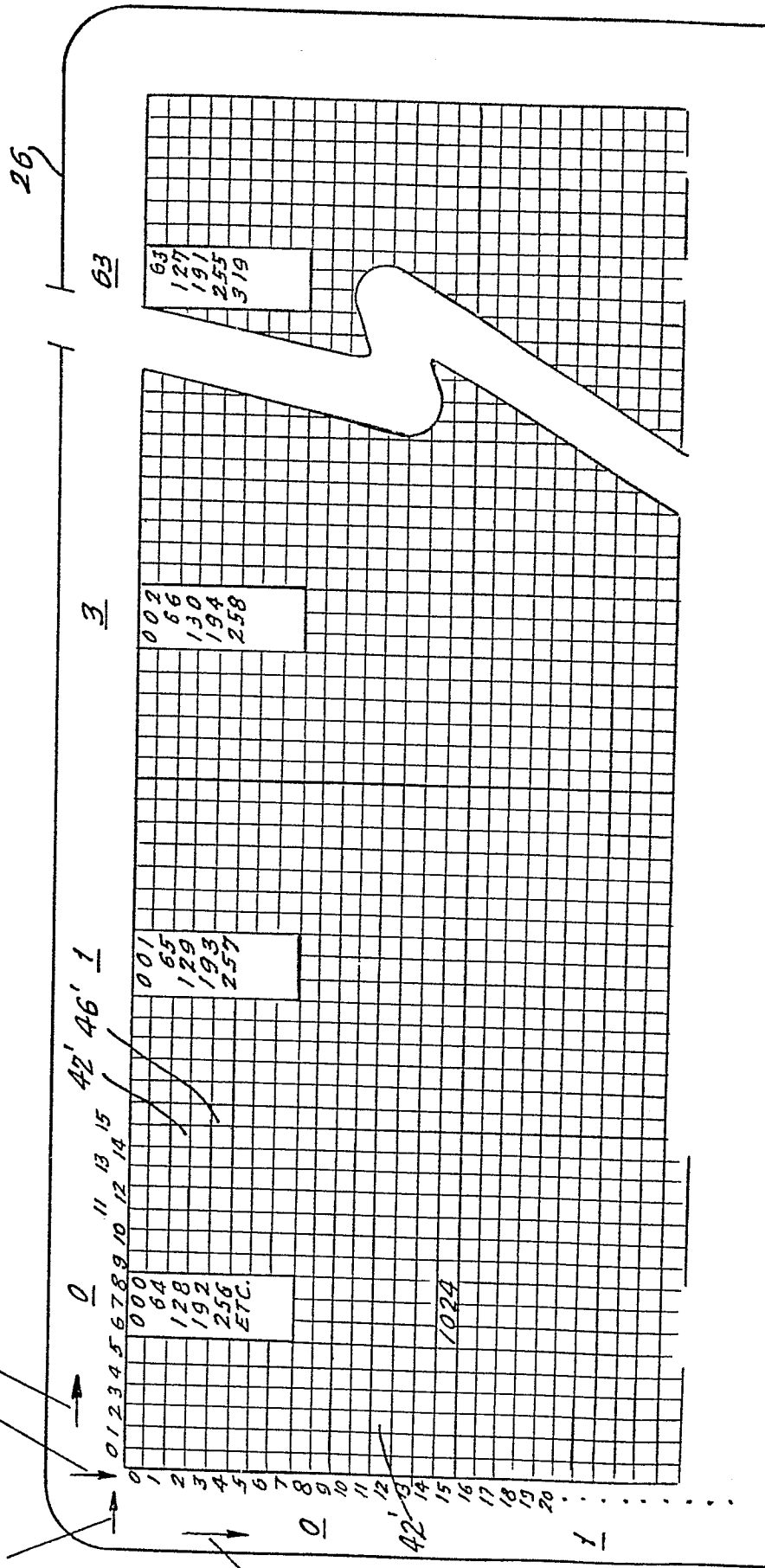
FF.3B.

1: 00000000000000000000
 2: 00000001110000000000
 3: 00000011111100000000
 4: 00000110001110000000
 5: 00000100000001000000
 6: 00001100000000000000
 7: 00000100000000000000
 8: 00000111100000000000
 9: 00000001111000000000
 10: 00000000011100000000
 11: 00001000000011000000
 12: 00001100000011000000
 13: 00001100000110000000
 14: 00000111111000000000
 15: 00000000111100000000
 16: 00000000000000000000

FF.3A.



FF-4-



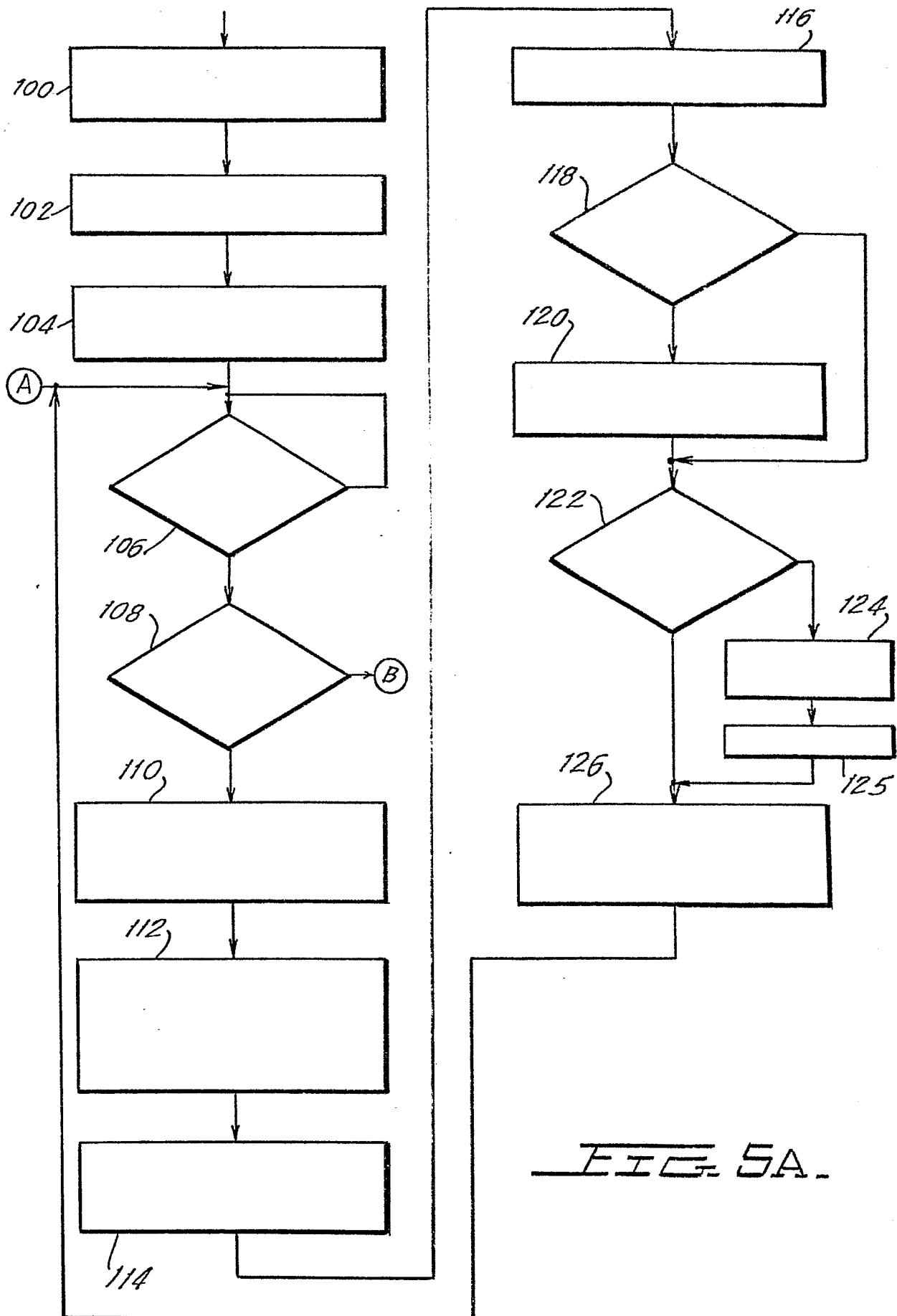
FIG. 5A.

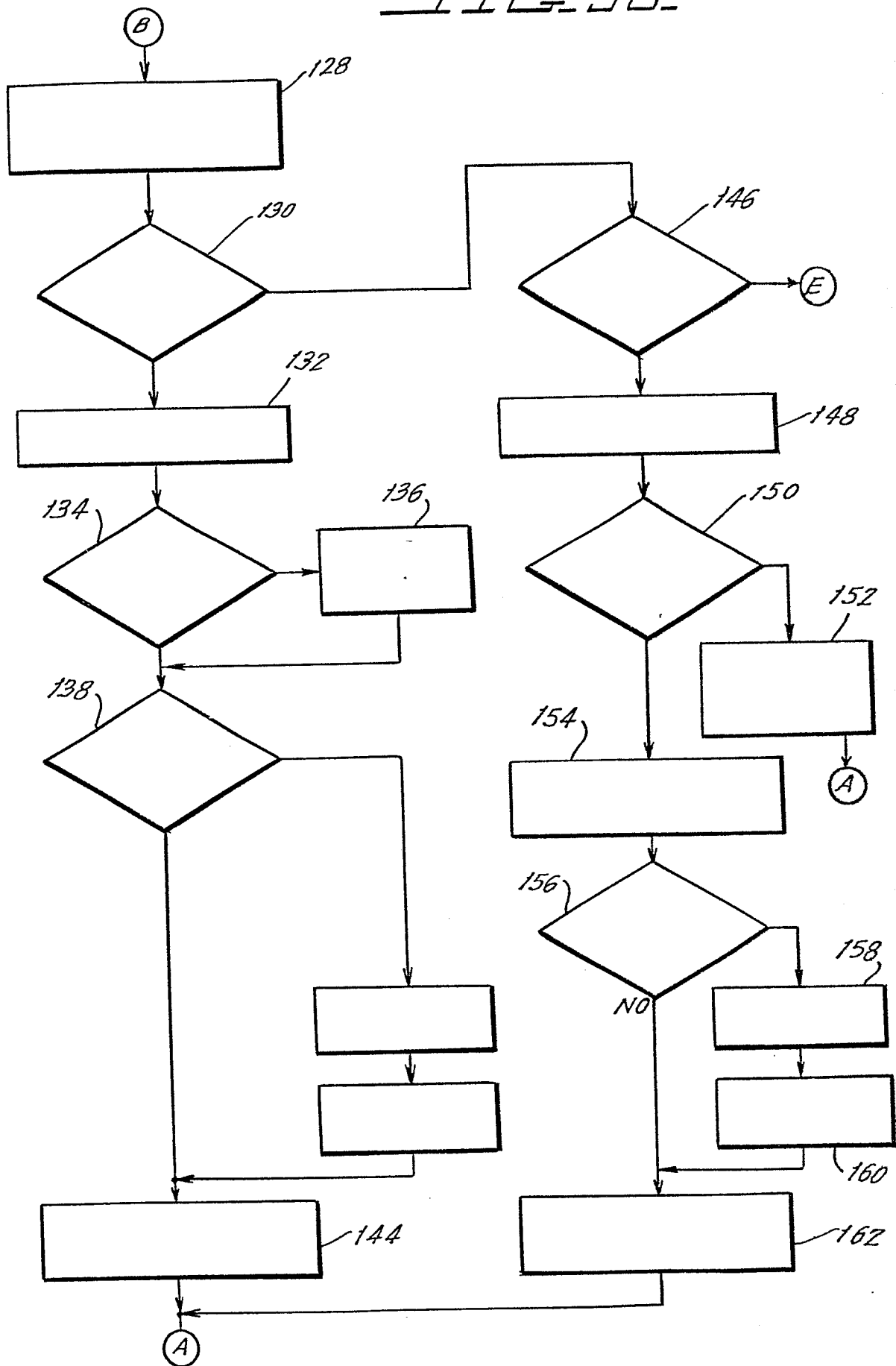
FIG. 5B.

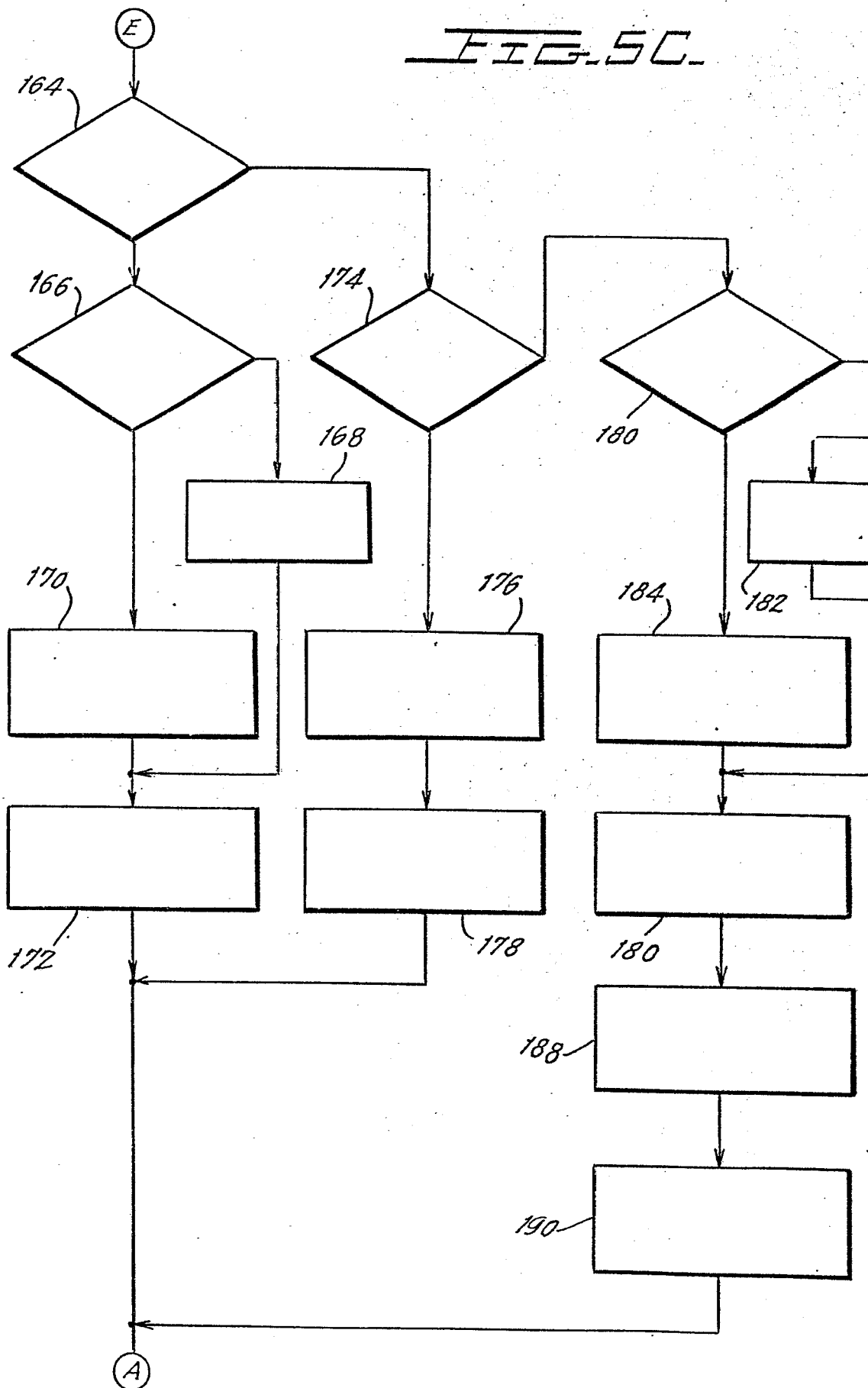
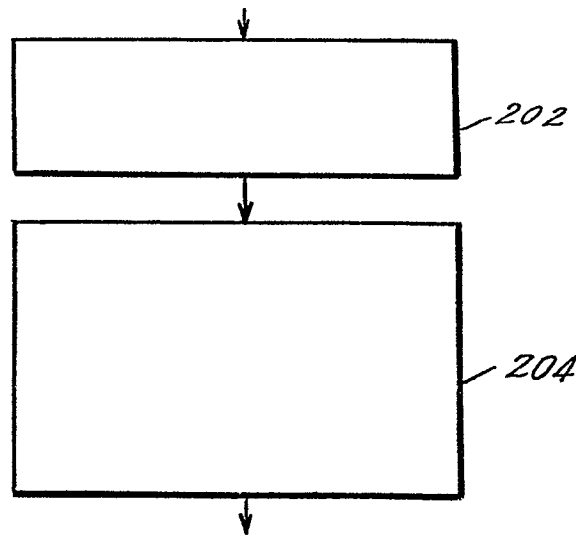
FIG. 5C

FIG. 6FIG. 7