960470S-5
IPOS
9.)

REPLACED SHEETS SG
☑ Description : paged ...1.... to paged ...14.....
☑ Claim(s) : paged ...11... to paged ....2/....
☑ Drawing(s) : paged ...1/8... to paged ...8/8....
☑ Abstract : paged ....1/1... to paged .....
DATE OF RECEIPT .....

BIT MAP DATA GENERATION APPARATUS

The present invention relates to a bit map data generation apparatus for generating bit map data from an outline font, and more particularly to a technique of a half-dot protect process. The apparatus is particularly suitable for a serial printer using a wire dot type recording head.

In an outline font, data indicative of a character or a figure are stored in the form of its contour line. Since the amount of the data is extremely smaller than that of bit map data, an outline font is widely used in a printer which requires a high print quality.

Such an outline font should be once converted into bit map data during a printing process. Especially, in a wire dot type recording head, every other dot of dots adjoining in the row direction is printed in order to attain a high-speed operation. Therefore, it is necessary to perform a half-dot protect process in which every other bit in the row direction is thinned out.

In such a process, generally, every other dot is mechanically thinned out by a logical circuit. This arises a problem in the case wherein an even number of dots are continued in the row direction. Namely, among dots constituting a character or pattern, dots in the contour portion are thinned out as shown in Figs. 9(A) and 9(B), with the result that the contour of the printed character or pattern is unclear.

This Sheet Has Been Replaced

U:\BENUTZER\WI\5865EP2C.TXT

The invention has been conducted in view of such a problem. An object of the invention is to provide a novel bit map data generation apparatus which, irrespective of a half-dot protect process, can surely preserve dots constituting a contour portion, thereby preventing the print quality from being degraded. This object is solved by the apparatus of independent claim 1. Further advantageous features, aspects and details of the invention are evident from the depending claims, the description and the drawings. The claims are intended to be understood as a first non-limiting approach of defining the invention in general terms.

In a preferred embodiment the bit map data generation apparatus of the invention comprises: means for converting data from contour data storing means into bit map data; first reading means for reading data preceding a current address by one dot from said converting means or print data which has been processed immediately before; second reading means for reading data at the current address from said converting means; third reading means for reading data succeeding the current address by one dot from said converting means; and fourth reading means for reading data succeeding the current address by two dots from said converting means; first data inversion means for outputting data which is the inversion of said data preceding the current address by one dot; second data inversion means for outputting data which is the inversion

This Sheet Has Been Replaced

of said data succeeding the current address by two dots; first logical product operation means for performing a logical product operation between data from said first data inversion means and data from said second reading means; second logical product operation means for performing a logical product operation between data from said third reading means and data from said second data inversion means; third data inversion means for outputting data which is the inversion of said data from said third reading means; fourth data inversion means for outputting data which is the inversion of data from said second logical product operation means; third logical product operation means for performing a logical product operation between data from said first logical product operation means and data from said fourth inversion means; fourth logical product operation means for performing a logical product operation between data from said second reading means and data from said third data inversion means; and logical sum operation means for performing a logical sum operation between data from said third and fourth logical product operation means to produce print data to be output onto a print head and fed back onto said first reading means.

The first to fourth reading means obtain data of four bits continued in the printing direction, at a current address, an address preceding the current address by one bit, and addresses succeeding the current address by one bit and two bits. Among bits constituting a contour portion, two

This Sheet Has Been Replaced

consecutive data "1" which precede data "1" at the forward end of a series of an even number of data "1" in the printing direction are converted into two data "0". In this way, a half-dot protect process is performed, so that dot data constituting at least right and left portions of the contour are prevented from disappearing.

Fig. 1 is a block diagram showing an embodiment of the invention by functions to be effected by a microcomputer;

Fig. 2 is a block diagram showing the embodiment of the invention;

Figs. 3(A) and 3(B) are diagrams respectively showing bit map data before the conversion performed by the apparatus, and bit map data after the conversion;

Fig. 4 is a flowchart showing the operation of the apparatus;

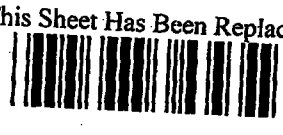
Fig. 5 is a diagram showing operation processes I and II in the apparatus;

Fig. 6 is a diagram showing operation processes III and IV in the apparatus;

Fig. 7 is a diagram showing operation processes V and VI in the apparatus;

Fig. 8 is a diagram showing operation processes VII and VIII in the apparatus; and

This Sheet Has Been Replaced



Figs. 9(A) and 9(B) are diagrams respectively showing bit map data before performing a half-dot protect process, and bit map data after performing the half-dot protect process in a conventional method.

The invention will be described in detail by illustrating an embodiment shown in the figures.

Fig. 2 shows an embodiment of the invention. In Fig. 2, reference numeral 1 denotes a contour data storing device. The contour data storing device 1 stores contour data which represent the contour line of a character or a pattern, correspondingly to a character code. The contour data consist of a combination of X coordinate values and Y coordinate values of reference points, i.e., starting points and end points of a plurality of lines which represent the number of data and a pattern, and line type data that specify the type of line joining the starting point and the end point with each other, e.g., a straight line or a parabola.

Reference numeral 3 denotes a microcomputer which constitutes a bit map data generation apparatus. The microcomputer 3 is so programmed as to output bit map data in accordance with a character code and attribute data in an input buffer 2 to a printing head 5 through a printing buffer 4.

Fig. 1 is a block diagram showing functions to be performed by the above-mentioned microcomputer 3. In Fig. 1,

This Sheet Has Been Replaced

reference numeral 10 denotes contour data reading means which accesses the contour data storing device 1 based on the character code in the input buffer 2, and outputs contour data. Reference numeral 11 denotes bit data conversion means. In this embodiment, a contour line is represented by bits. The bit data conversion means 11 is so constructed that logical operations are sequentially performed for the bits in the main scanning direction, so that data "1" are generated in the area surrounded by the contour line.

Reference numeral 20 denotes half-dot protect process means which is a feature of the invention. The half-dot protect process means 20 is so programmed as to realize the following means: left-side data reading means 21 for reading bit data on the left side of a current address by one bit based on the bit data produced by the bit conversion means 11, or print data which has been processed immediately before; current data reading means 22 for reading data at the current address; right-side by one-dot data reading means 23 for reading data on the right side of the current address by one bit; right-side by two-dot data reading means 24 for reading data on the right side of the current address by two bits; data inversion means 25 and 26 for respectively inverting data from the left-side data reading means 21 and the right-side by two-dot data reading means 24; logical product operation means 27 for performing a logical product operation between data from the data inversion means 25 and data from the current data reading

This Sheet Has Been Replaced



means 22; logical product operation means 28 for performing a logical product operation between data from the right-side by one-dot data reading means 23 and data from the data inversion means 26; data inversion means 29 for inverting data from the right-side by one-dot data reading means 23; data inversion means 30 for inverting data from the logical product operation means 28; logical product operation means 31 for performing a logical product operation between data from the data inversion means 30 and data from the logical product operation means 27; logical product operation means 32 for performing a logical product operation between data from the current data reading means 22 and data from the data inversion means 29; and logical sum operation means 33 for performing a logical sum operation between data from the logical product operation means 31 and data from the logical product operation means 32 to output print data onto a print head and the left-side data reading means 21. As a result, truth values shown in Table 1 below are given.

This Sheet Has Been Replaced



Table 1

	Left-side data	Data of current address	Right-side by one-dot data	Right-side by tow-dot data	Judgment
1	●	●	●	●	0
2	●	●	●	0	0
3	●	●	0	●	●
4	●	●	0	0	0
5	●	0	●	●	0
6	●	0	●	0	0
7	●	0	0	●	0
8	●	0	0	0	0
9	0	●	●	●	●
10	0	●	●	0	0
11	0	●	0	●	●
12	0	●	0	0	●
13	0	0	●	●	0
14	0	0	●	0	0
15	0	0	0	●	0
16	0	0	0	0	0

The operation result at No. 3 should properly be 0, but is ●. However, since the process is sequentially performed from the left, the operation result becomes 0 when the process proceeds to No. 10. Therefore, in fact, the operation result shown at No. 3 cannot be obtained.

This Sheet Has Been Replaced

Hereinafter, the operation of the device having the above-mentioned configuration will be described with reference to a flowchart shown in Fig. 4.

When a printing instruction is output, the contour data reading means 10 reads a character code stored in the input buffer 2 and magnification data associated with the data, and then reads out contour data corresponding to the character code, from the contour data storing device 1. When the outline font corresponding to the character code stored in the input buffer 2 is converted into bit map data in this way, data in the column direction at the address on the left side of the current address by one dot are read (step A). These data are inverted and then temporarily stored (step B). Logical product operations are performed between the data at the current address and the inverted data which are temporarily stored, and the operation result is temporarily stored as result 1 (step C).

Thereafter, data in the column direction at the address on the right side of the current address by two dots are read (step D). These data are inverted and then temporarily stored (step E). Thereafter, data in the column direction at the address on the right side of the current address by one dot are read (step F). Logical product operations are performed between the data and the inverted data obtained in step E, and the operation result is temporarily stored as result 2 (step G).

This Sheet Has Been Replaced



Logical product operations are performed between data obtained by inverting the data of result 2 which are temporarily stored and the data of result 1 obtained in step C, and the operation result is temporarily stored as result 3 (step H).

Then, data in the column direction at the address on the right side of the current address by one dot are read (step I). These data are inverted to obtain the inversions thereof (step J). Logical product operations are performed between the inverted data and data in the column direction at the current address, and the operation result is temporarily stored as result 4 (step K).

Logical sum operations are performed between the data of result 3 obtained in step H and the data of result 4 obtained in step K, and the result is output as printing data for the current address (step L).

When the printing bit data for the current address are output, the address from which data is to be read is shifted by one dot in the printing direction (in this embodiment in the right direction) (step M). The foregoing steps beginning with step A are repeated, whereby the half-dot protect process is executed on the bit map data produced by the bit conversion means 11. In the above-described process, when data to be currently processed is on or after the second column, data preceding the current address by one bit is directed to print data which has been converted immediately before.

This Sheet Has Been Replaced

Figs. 5 to 8 show the above-mentioned operation process steps. The figures illustrate an exemplary case where an outline font to be printed is converted by the bit data conversion means 11 into bit map data having a size of 4 bits by 8 bits as shown in Fig. 3(A). In Figs. 3(A), 3(B) and 5 to 8, each of the Roman numerals I to VIII indicates a transition of the current address.

The bit data constituting the column I in Fig. 3(A) are all "0", so that data of "1" are output for all the four bits as a result of the operation process shown in I of Fig. 5 (I in of Fig. 3(B)).

Next, the current address is shifted to the right by one bit, so that the column II in Fig. 3(A) is to be processed. Data of "1" are output for all the four bits as a result of the operation process shown in II of Fig. 5, while reading print data which has been produced by the operation process shown in I of Fig. 5. This means that the bit data are converted into the data preserving the left-side contour (II in Fig. 3(B)).

The current address is further shifted to the right by one bit, so that the column III is to be processed. The bit data for all the four bits are converted into data of "0" as a result of the operation process shown in III of Fig. 6. Accordingly, data which have been subjected to the half-dot protect process are produced (III in Fig. 3(B)).

When the column IV is to be processed, the operation process shown in IV of Fig. 6 is performed. According to this

This Sheet Has Been Replaced



process, a data set in which the second bit from the top is "0" and the other bits are "1" is output, while reading all the print data which have been converted into "0" by the calculation made immediately before. As a result, the bit in the first row (the uppermost line in the figure) forming the contour is output to be "1" which is the same as the original data, thereby preserving the contour portion. The dot in the second row from the top in the figure and adjacent to the dot of the contour portion is converted into data of "0" which is the same as in the column III.

Then, the object to be processed is shifted to the column V. Similarly, print data including data "0" in the second row which has been converted by the calculation made immediately before is read. As a result of the operation process shown in V of Fig. 7, the bit in the second row which constitutes the contour portion is output as data "1". The bit preceding (in this embodiment, on the left side) by one bit with respect to the bit which constitutes the contour portion in the next column VI is converted into data "0".

Thereafter, the processes shown in VI of Fig. 7 and in VII and VIII of Fig. 8 are performed. If there are a series of an odd number of data "1" which continues from one end (in this embodiment, the left end), the bit data is converted into data "0" every other column so as to thin out alternate bits. If there are a series of an even number of "1" data, a bit map pattern in which two bits preceding the other end (in this

This Sheet Has Been Replaced



embodiment, the right end) are converted into "0", and the bit at the other end portion is converted into "1" data is output.

As a result, even in a portion where an even number of dots are continued, a half-dot protect process can be executed without causing dots in the contour portion to disappear. In other words, an edge is constituted by four dots including data succeeding the current address by one dot and two dots, and data preceding the current address by one dot, so that it is possible to prevent dots constituting the counter portion, from being thinned out in performing the half-dot protect process.

In the embodiment described above, the printing is performed in the direction from the left to the right. It will be appreciated to those skilled in the art that, in a case where the printing is performed in the direction from the right to the left, the sampling of bit map data for an operation process is started at the right side, whereby the same effects are achieved.

As described above, in this invention, the first to fourth reading means obtain data of four columns in total: data at the current address; data preceding it by one bit or print data which has been processed immediated before; and data succeeding it by one bit and two bits. Then, the half-dot protect process is performed in such a manner that, in a series of an even number of "1" data and among the bits constituting the contour portion, the two "1" data preceding "1" data of the forward end in the printing direction are converted into "0"

This Sheet Has Been Replaced



data. Therefore, the half-dot protect process can be performed without causing the dots which constitute the left and right contour lines to disappear, whereby the print quality can be prevented from being degraded.

This Sheet Has Been Replaced



CLAIMS

1. An apparatus for generating bit map data, comprising:

means (1) for storing contour data;

means (11) for converting data from said contour data storing means (1) into bit map data; and

first reading means (21) for reading one of data preceding a current address by one dot from said converting means (11) and print data which has been processed immediately before;

second reading means (22) for reading data at the current address from said converting means (11);

third reading means (23) for reading data succeeding the current address by one dot from said converting means (11);

fourth reading means (24) for reading data succeeding the current address by two dots from said converting means (11);

first data inversion means (25) for inverting data from said first reading means (21);

second data inversion means (26) for inverting data from said fourth reading means (24);

first logical product operation means (27) for performing a logical product operation between data from said first data inversion means (25) and data from said second reading means (22);

second logical product operation means (28) for performing a logical product operation between data from said third reading means (23) and data from said second data inversion means (26);

This Sheet Has Been Replaced

This Sheet Has Been Replaced

* C L M S *

third data inversion means (29) for inverting data from said third reading means (23);

fourth data inversion means (30) for inverting data from said second logical product operation means (28);

third logical product operation means (31) for performing a logical product operation between data from said first logical product operation means (27) and data from said fourth inversion means (30);

fourth logical product operation means (32) for performing a logical product operation between data from said second reading means (22) and data from said third data inversion means (29); and

logical sum operation means (33) for performing a logical sum operation between data from said third (31) and fourth (32) logical product operation means to produce print data to be output to a print head (5) and said first reading means (21).

2. An apparatus as claimed in claim 1, further comprising:

input buffer means (2) for storing a character code;

contour data reading means (10) for accessing said contour data storing means (1) based on the character code from said input buffer means (2) to output contour data to said converting means (11).

3. An apparatus as claimed in claim 1 or 2, wherein when data to be processed currently is initial data, then said first reading means (21) reads data preceding the current address by one dot from said converting means (11), and when data to be processed currently is not initial data, then said first reading means (21) reads the print data output from said logical sum operation means (33).