

(12) UK Patent Application (19) GB (11) 2 139 447A

(43) Application published 7 Nov 1984

(21) Application No 8402620

(22) Date of filing 1 Feb 1984

(30) Priority data

(31) 58/053401 (32) 28 Mar 1983 (33) JP

(71) Applicant
Dainippon Screen Mfg Co Ltd (Japan),
1—1 Kitatenjin-Cho 4 Chome, Horikawdouri-
Teranouchiagaru, Kamigyo-Ku, Kyoto-shi, Japan

(72) Inventor
Mitsuhiko Yamada

(74) Agent and/or address for service
Withers & Rogers, 4 Dyer's Buildings, London, EC1N
2JT

(51) INT CL³
H04N 1/41

(52) Domestic classification
H4F EM S1 S30B S30K

(56) Documents cited
GB 1572121

(58) Field of search
H4F

(54) Methods and Systems for Condensing Binary Image Data

(57) Line drawing is scanned and pixels arranged in $m \times n$ arrays on blocks (B). A block may be of three types; (a) All vacant (e.g. B₁₁), (b) All solid (eg B₃₄), (c) Hybrid (eg B₃₃). Data is transmitted as a digital word expressing I+II or I+III, where I (R) gives the type of block, II (T) in the case of a type (a) or (b) gives a run length ($L \times M + N$) of a procession of the same type, III (T) in the case of type (c) gives the pixel pattern within the block.

FIG.3

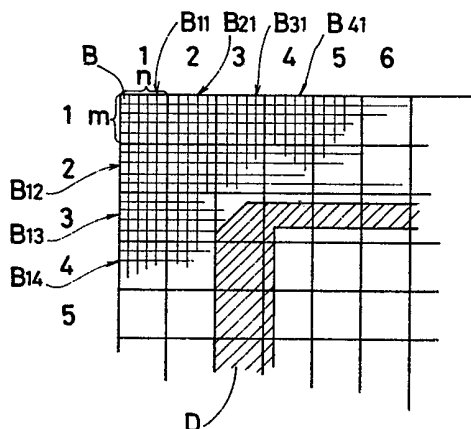
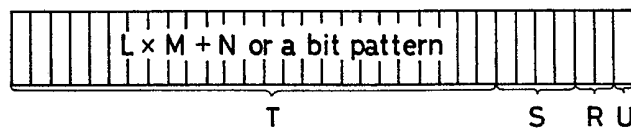


FIG.5



GB 2 139 447 A

FIG.1

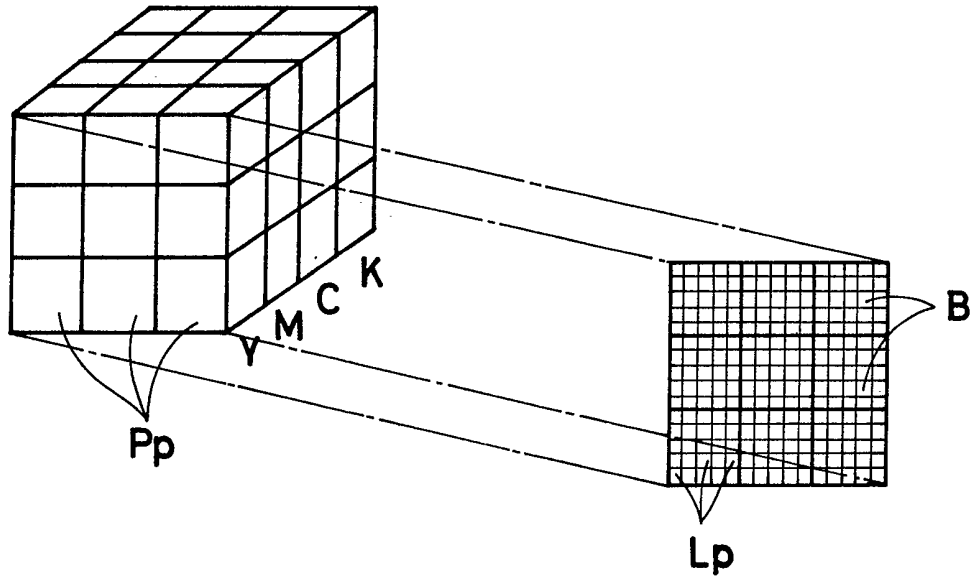


FIG.2

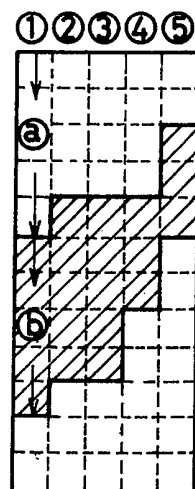


FIG.3

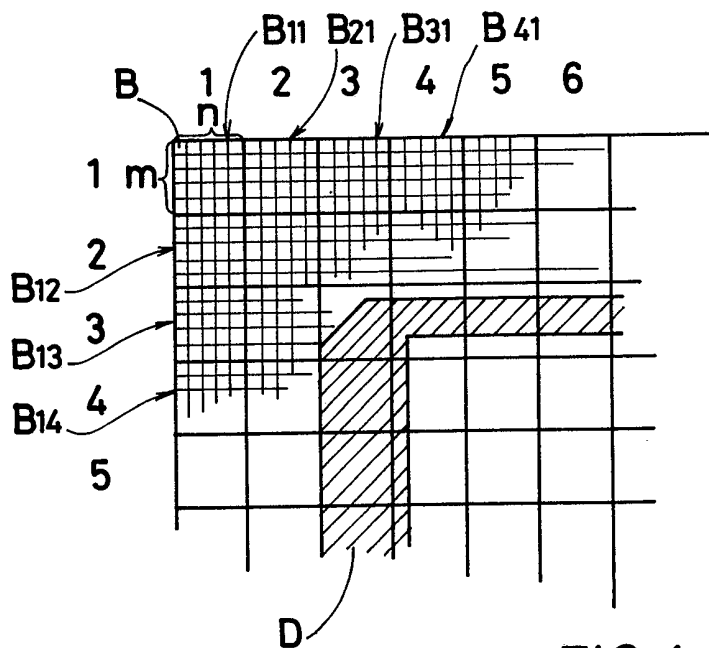


FIG.4

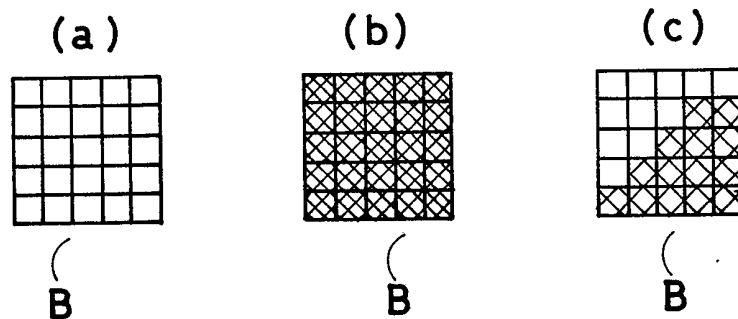
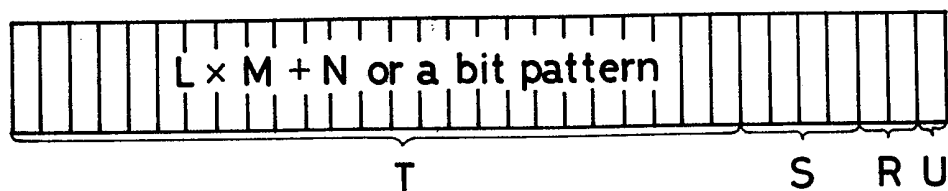


FIG.5



3/10

FIG. 6

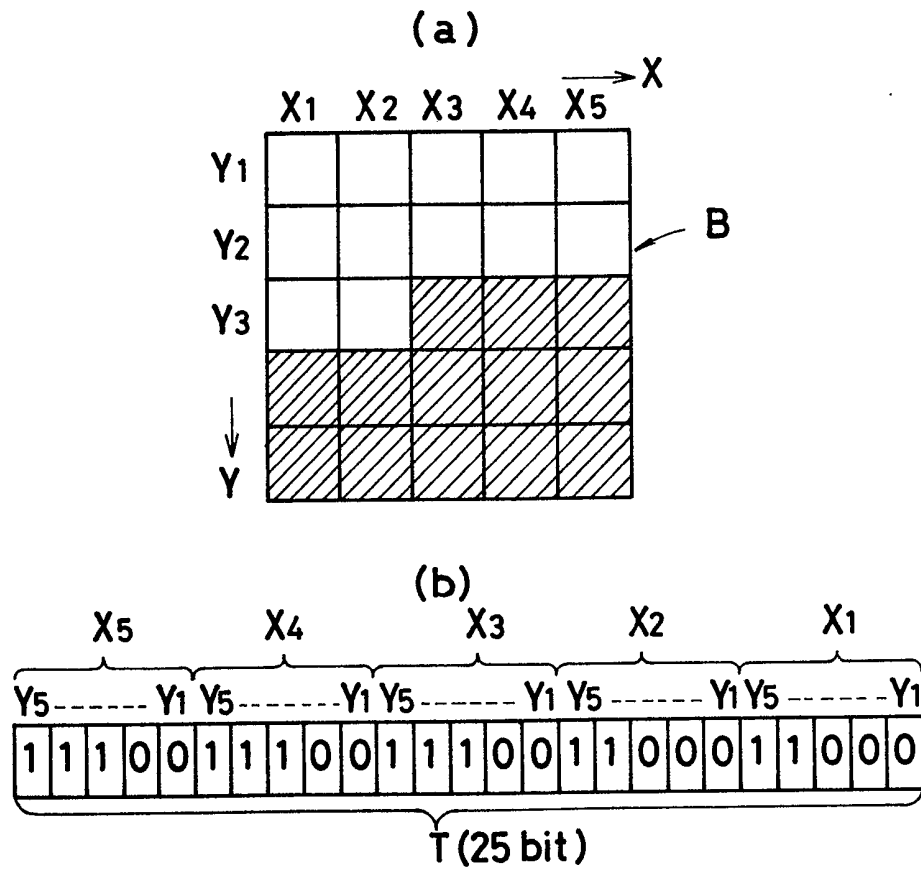


FIG. 7

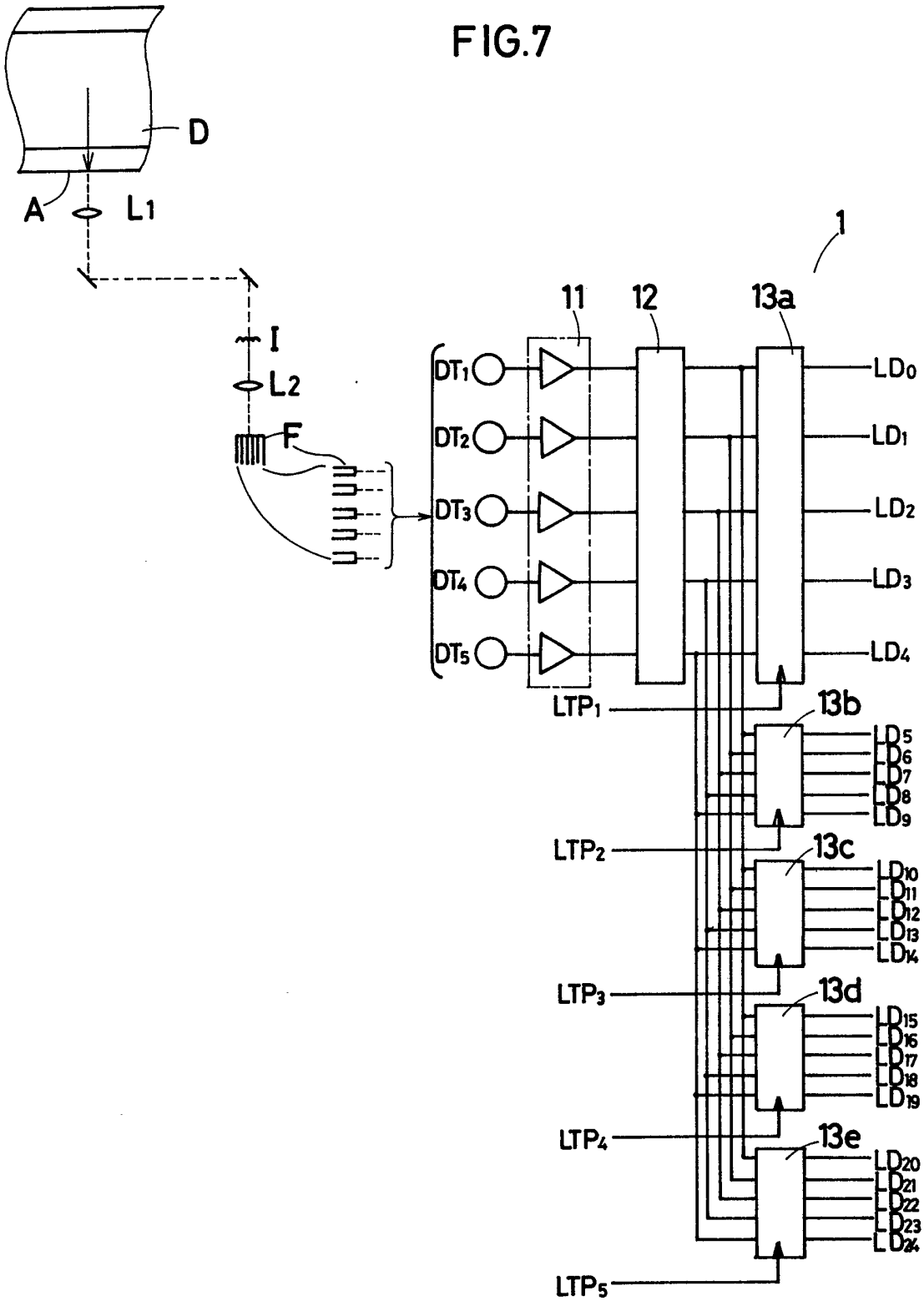


FIG.8

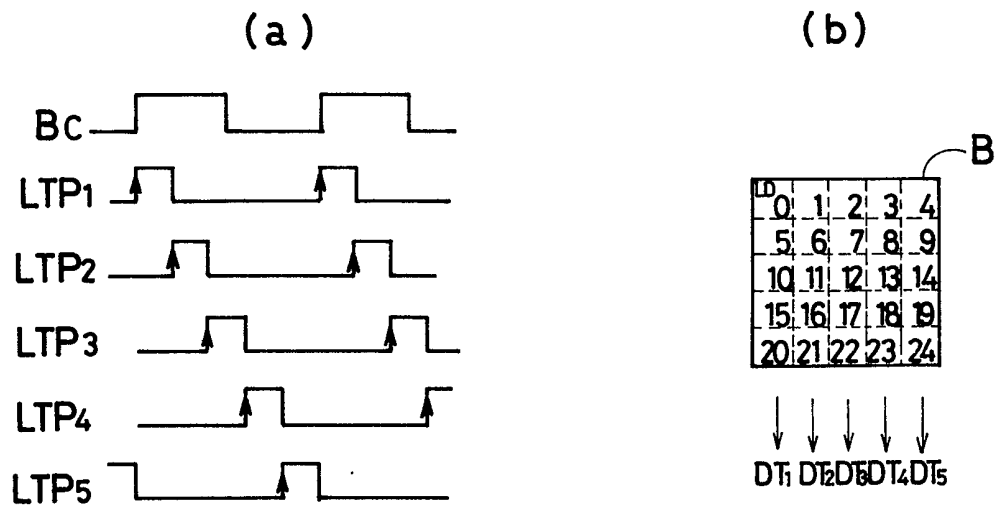


FIG.10

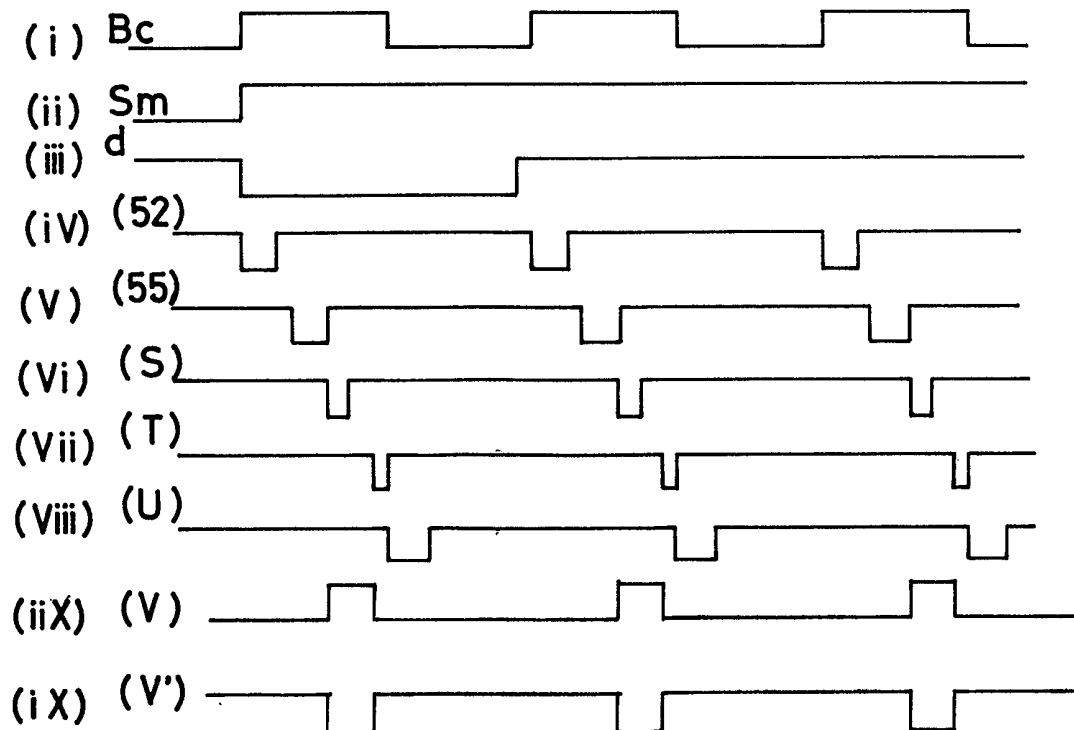
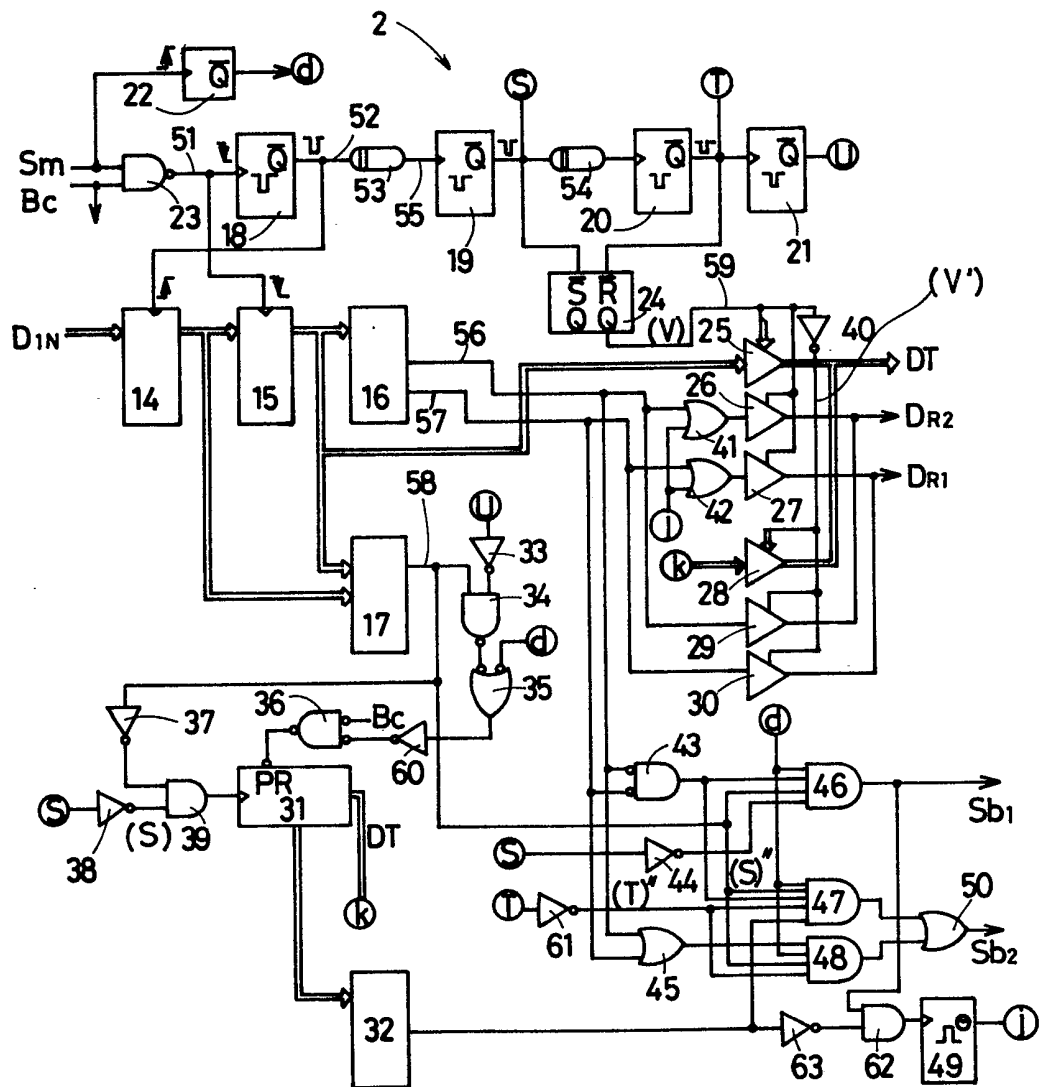


FIG. 9



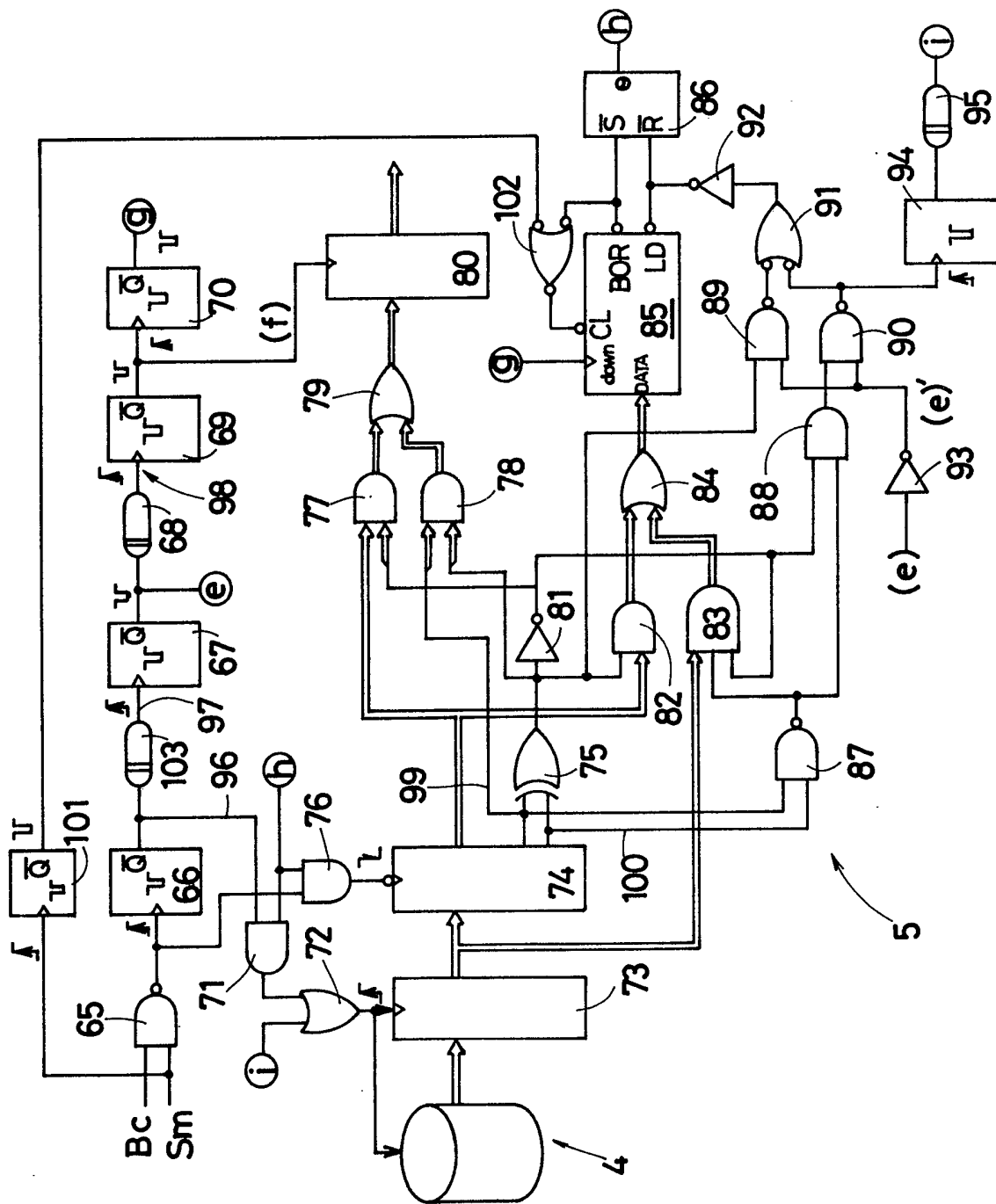


FIG. 11

FIG.12

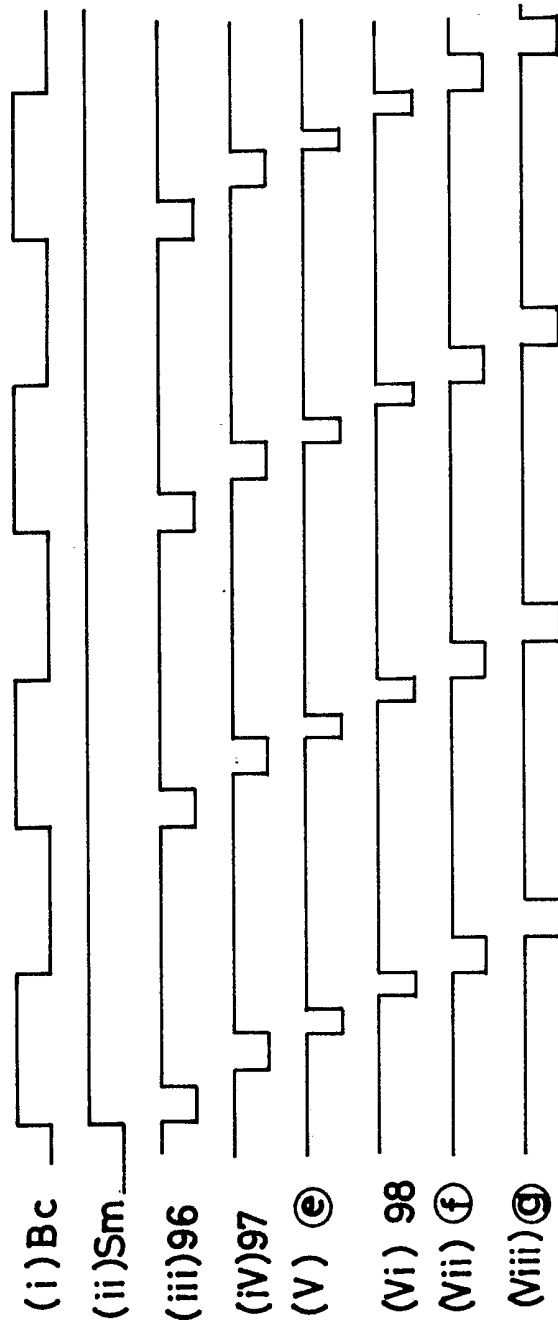


FIG.13

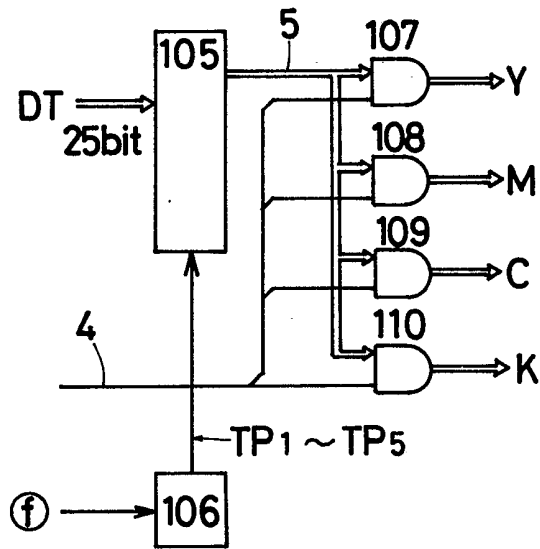


FIG.14

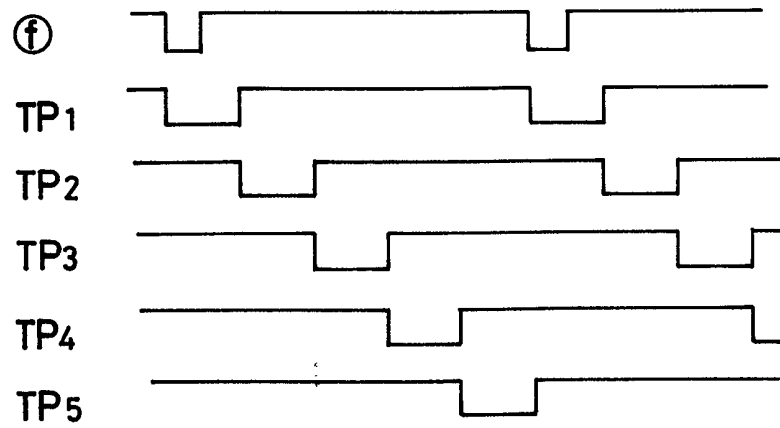
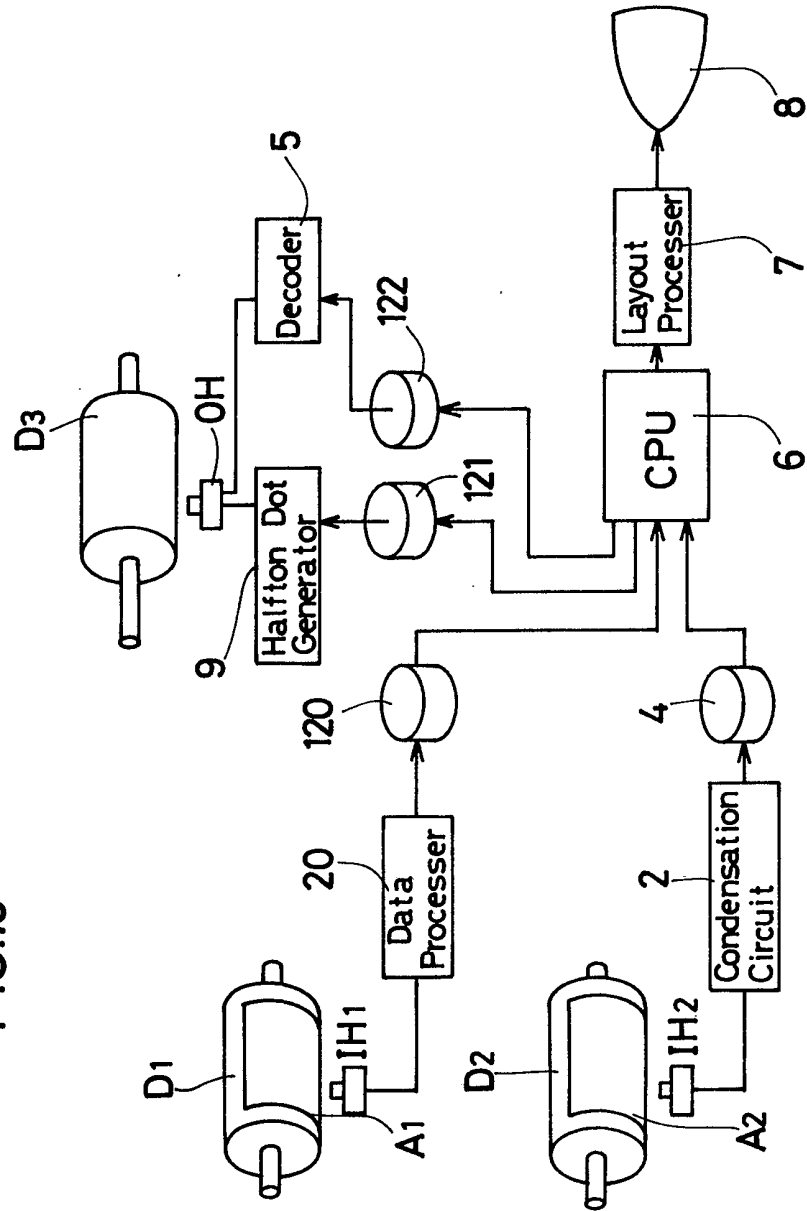


FIG.15



SPECIFICATION

Methods and Systems for Condensing Binary Image Data

This invention relates to methods and systems for condensing binary data of such as a literal or a line drawing image, specifically to such a method and system in which binary data of such as a literal or a line drawing image are condensed by an input scanning or a computation means of an image reproducing system such as a facsimile or a scanner.

Conventionally, as a method or processing binary image data (line-drawing or literal data expressed by two logic levels, in which logic "1" represents a "solid" pixel, and logic "0" represents a "vacant" pixel) of such as a literal or a line-drawing image both covered by "line-drawing image" hereinafter) used in a colour scanner, United States Application Serial Nos. 471,869 or 507,719 discloses the following method.

As shown in Figure 1, binary data corresponding to a line-drawing pixel block B composed of multiple line-drawing pixels Lp (covering an area being a fraction of that of a pictorial pixel) are obtained in an input stage. In the case of Figure 1, one line drawing pixel block B composed of 25 line-drawing pixels, namely 5×5 line-drawing pixels arranged in a matrix in the main and the sub-scanning directions corresponds to one pictorial pixel. Then, the thus obtained line drawing (binary) data are expressed in a bit signal used for expressing pictorial color separation images Y (Yellow), M (Magenta), C (Cyan) and K (Black). So when data of each color separation image are expressed in bit signal, the line drawing image data are to be expressed in an 8×4=32 bit signal. In the abovementioned method, the line-drawing pixel block B (in Fig. 1, it consists of 25 pixels) can be in one of the following three conditions. It can be in (a) a condition in which all pixels are transparent or white, namely containing no line-drawing images (called "vacant" hereinafter), (b) a condition in which all pixels are opaque or black, namely containing line-drawing images (called "solid" hereinafter) or (c) a condition in which some of the pixels contain line-drawing images (called "hybrid" hereinafter). In this, data of one line-drawing pixel block must be expressed for example in 32 bits, which consequently brings about a demerit the final stored data mass swells.

On the other hand, ruling image data are obtained by a computation as in the following way. That is, at first scale of a ruling image pixel is computed, which must correspond to a fraction of a pictorial pixel in the main and the sub-scanning directions. Then according to a desired arrangement, ruling image data are made and put into a run-length form to be stored in a line-drawing data file. As several scanning lines for the line-drawing image corresponds to one scanning line for the pictorial image, the line-drawing data must be output in synchronization with the

pictorial data. More precisely, as shown in Fig. 2, line-drawing data of the "vacant" part (a) and the "solid" part (b) of Line 1 are stored in a run-length form into the line-drawing data file. The same operation is performed for lines 2 to 5

respectively and the line-drawing data of Lines 1 to 5 are output in parallel simultaneously to be used for driving a recording beam, wherein of course the data are reconverted into their original forms.

In this case, although the ruling image data are condensed by being put into a run-length form, this method still contains a defect that several same processing and output units are necessary as the ruling image data must be output in the abovementioned parallel way.

Incidentally, literal data can also be obtained by using pre-digitized letter font data, but the abovementioned defect is not resolved as the data are output also in aforesaid parallel way.

A prime object of this invention is to resolve the abovementioned problem.

According to the invention we propose a method for condensing binary image data in reproducing images, comprising:

obtaining data of a line-drawing pixel block composed of m×n line-drawing pixels by an input scanning means or by a computation means; and generating a "word for expressing (I) and (II) or (I) and (III) of the following, (I) data for judging whether the pixels of the line-drawing pixel block are in a "vacant" (as herein defined) or "solid" (as herein defined) "hybrid" (as herein defined) condition, (II) run-length data of the number in one sequence of the line-drawing pixel blocks which are in the "vacant" or "solid" condition; and (III) m×n bit pattern data of a line-drawing pixel block in the "hybrid" condition.

According to another aspect of the invention, we propose a system for condensing binary image data in reproducing image, comprising:

means for obtaining data of a line-drawing pixel block composed of m×n line-drawing pixels by an input scanning means or by a computation means; and

means for generating a "word" for expressing (I) and (II) or (I) and (III) of the following, (I) data for judging whether the pixels of said line-drawing pixel block are in a "vacant" (as herein defined) or "solid" (as herein defined) or "hybrid" (as herein defined) condition; (II) run-length data of the number in one sequence of the line-drawing pixel blocks which are in the "vacant" or "solid" condition; and (III) m×n bit pattern data of a line-drawing pixel block in the "hybrid" condition.

At first binary data are obtained from a line-drawing pixel block composed of m×n (where m and n are integers) line-drawing pixels arranged in matrix corresponding to a pictorial pixel. Then each line-drawing block data are distributed into three types of "vacant", "solid" or "hybrid", and are processed as follows respectively. When "vacant" line-drawing blocks are in a sequence, the sequence number of the "vacant" blocks are counted and condensed to be expressed in a

"word" or bit signal (for example 32 bit signal). When "solid" line-drawing blocks are in a sequence, the same procedure as above is performed. When a "hybrid" line-drawing block exists singly, bit pattern data of the block are expressed in a "word". When "hybrid" line-drawing blocks are in a sequence, bit pattern data of the blocks are condensed to be expressed in a "word" and the sequence number of the blocks are expressed in the next "word". Namely, this type of line-drawing pixel blocks can be condensed. In addition, said four types can be distinguished by attaching a special code to each "word", and colour factors can also be managed by the same procedure.

An embodiment according to this invention is described by way of example, with reference to the drawings in which:—

Figure 1 shows a relation of between a line-drawing pixel and a pictorial pixel;

Figure 2 shows a line-drawing image illustrating condensing binary image data of a conventional system;

Figure 3 shows a line-drawing image on an original picture illustration this invention;

Figure 4 shows three possible types of line-drawing image pattern of a line-drawing pixel block;

Figure 5 shows a conceptual chart of a "word" (a bit signal);

Figures 6(a), (b) show a relation of between a line-drawing image pattern and bit pattern data on a "word";

Figure 7 shows an input device for this invention;

Figure 8(a), (b) show a timing chart and a data form for the input device shown in Figure 7;

Fig. 9 shows a data condensation circuit;

Fig. 10 shows a timing chart for the condensation circuit shown in Fig. 9;

Fig. 11 shows a decoder;

Fig. 12 shows a timing chart for the decoder shown in Fig. 11;

Fig. 13 shows an embodiment of a converter provided behind the decoder shown in Fig. 11;

Fig. 14 shows a timing chart for the converter shown in Fig. 13;

Fig. 15 shows a layout color scanner system to which the method of this invention can be applied;

As shown in Fig. 3, by scanning a line drawn image D, binary data of the line-drawn image are obtained, wherein line-drawing pixel block B composed of $m \times n$ line-drawing pixels corresponds to one pictorial pixel. In this, data of every line-drawing pixel block are distributed to each corresponding type (a) "vacant", (b) "solid" or (c) "hybrid" as shown in Fig. 4, and are processed in corresponding one of the following four ways. When line-drawing blocks of the "vacant" type (a) are in a sequence as from Block B_{11} of the first scanning line to Block B_{32} of the third scanning line, the sequence number of such blocks for example Blocks B_{11} to B_{32} are expressed in a "word" in run-length form. When line-

drawing pixel blocks of (b) "solid" type are in a sequence as in Block B_{34} and succeeding blocks, the sequence number of such blocks are expressed in the same way as the previous case.

When a line-drawing pixel block of "hybrid" type (c) exists independently as in Block B_{33} , B_{34} or B_{53} , data of such a block are expressed in a bit pattern mode (a mode in which an image pattern is expressed in a "word" (bit signal)). When line-drawing pixel blocks of "hybrid" type (c) are in a sequence as in blocks B_{44} , $B_{45} \dots$, data of such blocks are expressed in two "words", in which one "word" is used for expressing the line-drawn image pattern of one of the blocks and the other "word" is used for expressing the sequence number of the blocks.

Fig. 5 shows the composition of a "word" (in this case, it consists of 32 bits). The first bit U is used for distributing the data contained in the "word" to a pictorial data group ($U=0$) or to a line-drawing data group ($U=1$). The second two bits R is used for judging whether the data contained in the "word" are (i) "vacant" (for example $R=01$), (ii) "solid" (for example $R=10$), (iii) "hybrid in isolation" (for example $R=11$) or (iv) "hybrid in a sequence" (for example $R=00$). The next four bits S is used for expressing colour factor when line-drawing images are coloured.

The last 25 bits T is used for expressing data of line-drawing pixel blocks in respective cases.

When the line-drawing pixel blocks B are in a condition of (i) or (ii), the sequence number of the blocks (including a case in which one block exists independently) is allotted to the last 25 bits T. The sequence number can be expressed as M (the number of the blocks in the main scanning direction) $\times L$ (the number of the blocks in the sub-scanning direction which is a variable) $+ N$ (the number of the blocks situated in the last main scanning line of the sequence). In this, the procedure of putting bit pattern data into run-length data can be renewed every several main scanning lines. When the line-drawing pixel block is in condition (iii) as shown in Fig. 6(a), each bit of the 25 bits T is used for expressing corresponding line-drawing pixel by taking a logic level of "1" or "0" as shown in Fig. 6(b). When the line-drawing pixel blocks are in condition (iv), a "word" is used for expressing line-drawing image pattern of a block, and another "word" is used for expressing the sequence number of such blocks in run-length form.

The following explanation is based on a case the method of this invention is applied to an image reproducing system such as a color scanner.

Fig. 7 shows an input scanner device for line-drawing pixels. A laser beam transmitted through (refracted at) an original picture A placed on an original picture drum D is brought to pickup lenses L_1 and L_2 . The beam from the lens L_2 is introduced to one edge of an optical fiber F in certain projection size. The optical fiber F distributes fractionated line-drawing image data to photo-diodes DT_1 , DT_2 , DT_3 , DT_4 and DT_5 , which

diodes convert optical data into electronic analog data. Then each analog data is amplified in an amplifier 11 and converted into binary data in a comparator 12 and input to latches 13a, 13b, 13c, 13d and 13e.

These latches 13a to 13e hold the binary data and output them according to timing pulses LTP_1 to LTP_5 as shown in Fig. 8(a) in which the pulses having a duration corresponding to one line-drawing pixel and are arranged to have phase difference between each other as in Fig. 8. Namely, line-drawing image data LD_0 to LD_{24} corresponding to aforesaid one line-drawing pixel block are output on command of the pulses LTP_1 to LTP_5 in every five line-drawing pixel data which are the constituents of a line-drawing pixel block in the sub-scanning direction. Finally, line-drawing pixel data as shown in Fig. 8(b) are output to a decoder (mentioned later).

In Fig. 8(a), a block clock B_c is a clock pulse which corresponds to one line-drawing pixel block, which block pulse B_c is also used for sampling a pictorial pixel in a colour scanner.

Thus obtained line-drawing image data are condensed in a condensation circuit 2. In Fig. 9, broad lines indicate 25 bit data lines, which means elements on the way of the broad lines represents one of the 25 same elements. 25 bit data of a line-drawing pixel block output from the latches 13a to 13e (shown in Fig. 7) are input to latches 14 (comprising 25 latch units) on command of the block clock B_c simultaneously.

In a timing chart shown in Fig. 10, when a control signal S_m becomes "1", the block clock B_c gets a sign to pass through a NAND-gate 23 in order to carry out a condensation procedure. Then according to the fall time of the block clock B_c (51), data of a line-drawing pixel block held on the latches 14 are brought to latches 15, and then to a decoder 16.

The block clock B_c (51) is also input to a one shot multi-vibrator 18 (called an "OSMV" hereinafter). The OSMV 18 outputs one pulse (52) in the fall time of the block clock B_c . This pulse (52) is input via a delay unit 53 to another OSMV 19. When said one pulse (52) is output from the OSMV 18, data of the next line-drawing pixel block are input to the latches 14. A control signal (S) is output from the OSMV 19, and is input via a delay unit 54 to an OSMV 20. The OSMV 20 outputs a control signal (T), which is input to an OSMV 21. The OSMV 21 outputs a control signal (U) as shown in Fig. 10 (vi), (vii), (viii). Said control pulse (S) and (T) are input to the set terminal and the reset terminal of a flip-flop circuit 24 respectively. The flip-flop circuit 24 outputs a pulse (v) which rises synchronizing with the control pulse (S) and falls synchronizing with the control pulse (T). And the pulse (V) is inverted in an inverter 40 to be a pulse (V)' which rises synchronizing with the control pulse (T) as shown in Fig. 10 (ix), (x). These pulses (S), (T), (U), (V), and (V)' are used for controlling the whole condensation circuit 2 as mentioned later.

Meanwhile, the data of the line-drawing pixel

block held on the latches 15 are input to the decoder 16 as well as to a coincidence circuit 17. The decoder 16 judges whether data of each line-drawing pixel block are in a type of "vacant", "solid" or "hybrid" and express them in a combination of logic signals of "0" and "1" (56) (57).

Assuming that the data of a line-drawing pixel block held on the latches 15 are (X) and the data of a line-drawing pixel block held on the latches 14 are (Y), they are processed in the following way.

In the case (X) are "vacant in isolation" and (Y) are other than "vacant", the logic signals (56) and (57) become "0" and "1" respectively. These logic signals (56) and (57) are to be output as judge data "0, 1" from tri-state buffers 29 and 30 synchronizing with the pulse (V)'. While, a logic signal (58) from the coincidence circuit 17 becomes "1" meaning "disagreement". Said logic signal (58) is inverted in an inverter 37 to be "0" and is used for closing an AND-gate 39.

Therefore, an inverted pulse (S)' (a pulse which is obtained by inverting the control pulse (S) in an inverter 38) isn't input to a counter 31 (as mentioned later, the count number of the counter 31 represents the number of run-length).

Meanwhile, an inverted pulse (U)' (a pulse which is obtained by inverting the control pulse (U) of the previous line-drawing block is input via an inverter 33, a NAND-gate 34, a NOR-gate 35, an inverter 60 and NAND-gate 36 to the counter 31 to set its preset number at "1". Then data D_T having the value "1" (the sequence number of the line-drawing block of "vacant") is output from tri-state buffers 28 synchronizing with the inverted pulse (V)' (with the judge data "0, 1"). At the same time, the decoder 16 outputs the logic signal (56) and (57) via an OR-gate 45 to open an AND-gate 48. The AND-gate 48 outputs an inverted pulse (T)'' being inverted in an inverter 61 as a strobing signal Sb_2 synchronizing with the pulse (V)'. This strobing signal is used for controlling timings of data reciprocation of between the condensation circuit etc. and a disc memory etc. provided behind.

In the case (X) are "vacant in a sequence" and (Y) are other than "vacant", the logic signals (56) and (57) become "0" and "1" respectively. Consequently as in the previous case, these signals (56) and (57) are to be output as the judge data "0, 1" from the tri-state buffers 29 and 30 respectively synchronizing with the pulse (V)'. While the logic signal (58) from the coincidence circuit 17 becomes "0" meaning "agreement". Therefore, the output signal from the inverter 37 becomes "1", which makes the inverted pulse (S)' pass through the AND-gate 39, and the counter 31 counts the number of the input inverted pulses (S)'. Data of the count number of the counter 31 (in other words, the sequence number of the blocks of "vacant") are output as data D_T from the tri-state buffers 28 synchronizing with the inverted pulse (V)' (with the judge data "0, 1").

The logic signal (58) from the coincidence circuit

17 is changed from "0" to "1" when "vacant" block data are held on the latches 15 and block data other than "vacant" are held on the latches 14. Then the count number of the counter 31 is preset to "1" by the inverted pulse (U)'. At the same time, the signals (56) and (57) output from the decoder 16 via the OR-gate 45 opens the AND-gate 48 which makes the inverted pulse (T)' pass through the gate to be output as the strobing signal Sb_2 synchronizing with the pulse (V)'.

In the case (X) are "hybrid in isolation" and (Y) are other than "hybrid", both of the signals (56) and (57) output from the decoder 16 become "0". These signals (56) and (57) are to be output as judge data "1, 1" from respective tri-state buffers 26 and 27 synchronizing with the pulse (V) as follows. That is, at first a decoder 32 judges whether the count number of the counter 31 is "1" or not. In this case, as the count number is "1", the decoder 32 outputs a signal of logic "0". The signal of logic "0" is inverted in an inverter 63 to be a signal of logic "1" and is input via an AND-gate 62 to an OSMV 49. The OSMV 49 outputs a pulse (j) (in this case, its logic is "1"), which is output via AND-gates 41 and 42 from the tri-state buffers 26 and 27 synchronizing with the pulse (V). This output signal contains the judge data "1, 1". While the logic signal (58) from the coincidence circuit 17 becomes "1" meaning "disagreement". The logic signal (58) is inverted in an inverter 37 consequently closing the AND-gate 39. So the signal (S)' isn't input to the counter 31. Meanwhile, bit pattern data are output from the latches 15 via tri-state buffers 25 as the data D_T synchronizing with the pulse (V) (with the judge data "1, 1"). While the logic signals (56) and (57) open a NAND-gate 43, of which output signal of logic "1" is input to an AND-gate 46. As the signal of logic "1" from the coincidence circuit 17 is input to the other terminal of the AND-gate 46 beforehand, the AND-gate 46 is opened to pass the inverted pulse (S)' being inverted in an inverter 44 through the gate as a strobing signal Sb_1 .

In the case (X) are "hybrid in a sequence" and (Y) are other than "hybrid", both of the logic signals (56) and (57) output from the decoder 16 become "0" being the same as the previous case. In this, the output logic signal of the decoder 32 becomes "1" to make the OSMV 49 produce a signal of logic "0", which is input to the OR-gates 41 and 42. Then both OR-gates 41 and 42 output the signal of logic "0" to respective tri-state buffers 26 and 27, which output judge data "0, 0" synchronizing with the pulse (V). While bit pattern data are output from the latches 15 via the tri-state buffers 25 as the data D_T synchronizing with the pulse (V). In this case the strobing signal Sb_1 is output from the AND-gate 46 synchronizing with the pulse (V). On the other hand, the logic signal (58) of the coincidence circuit 17 becomes "0" and the counter 31 counts the inverted pulses (S)' until the line-drawing blocks of a sequence ends. Then the count number of the counter 31 is output from

the tri-state buffers 28 as the data D_T synchronizing with the pulse (V)'. While the judge data "0, 0" is output from the tri-state buffers 29 and 30. In this case, as the output of the decoder 32 becomes "1" and the output of the NAND-gate 43 becomes "1", when the output of the coincidence circuit 17 becomes "1" meaning "disagreement", the strobing signal Sb_2 is output from the AND-gate 47. After that, the count number of the counter 31 is preset to "1" on command of the inverted pulse (U)'. That is, only when the "hybrid" blocks are in a sequence, a "word" is used for expressing data of the line-drawing block, and the next "word" is used for expressing the sequence number of the blocks.

In the case (X) are "solid" and (Y) are other than "solid", the logic signals (56) and (57) output from the decoder 16 become "1" and "0" respectively. While the output logic signal of the coincidence circuit 17 becomes "1" meaning "disagreement". Therefore, the counter 31 doesn't count the inverted pulses (S)'. And the count number of the counter 31 is preset to "1" on command of the inverted pulse (U)'. So the logic signal of "1" is output from the tri-state buffers 28 as the data D_T synchronizing with the pulse (V)'. At the same time, judge data "1, 0" are output from the tri-state buffers 29 and 30, and an inverted signal (T)' is output via the AND-gate 48 from an OR-gate 50.

In the case (X) are "solid" in a sequence and (Y) are other than "solid", the logic signals (56) and (57) from the decoder 16 become "1" and "0" respectively. In this case, data of the sequence number of line-drawing pixel blocks are output from the tri-state buffers 28, the judge data "0, 1" are output from the tri-state buffers 29 and 30 synchronizing with the pulse (V)'. At the same time, the output signals (56) and (57) from the decoder 16 opens AND-gate 48 which outputs the strobing signal Sb_2 synchronizing with the pulse (V)'.

It should be pointed out that colour codes (see Fig. 5) can be processed as the data D_T or the judge data D_{R2} and D_{R1} .

In addition, the above explanation includes condensing data of line-drawing pixel blocks of "hybrid in a sequence", but if such blocks are limited, they can also be expressed in a conventional way.

Thus condensed line-drawing data by the operation of the condensation circuit 2 are stored into a disc memory 4. After undergoing color designation by an input from an external device like a control panel at need, the line-drawing data are output from the memory 4 to a recording head by the operation of a decoder 5 as shown in Fig. 11.

A disc memory 4 shown in Fig. 11 comprises a buffer memory, an address counter etc., which reciprocates data of one line-drawing pixel block (one "word") with other devices on command of said strobing signals Sb_1 , Sb_2 or of the block clock Bc. Broad lines shows 32 bit data lines, in which the 32 bit data are composed of 25 bit data D_T , 2

bit judge data DR_1 and DR_2 for distinguishing line-drawing block types, 4 bit color data and 1 bit judge data for distinguishing pictorial and line-drawing pixels. Therefore, of course every

5 elements on the way of the broad lines represents one of 32 same elements, the following mainly mentions about the 25 bit data D_T and 2 bit judge data DR_1 and DR_2 for distinguishing line-drawing block types.

10 When a data control logic signal S_m becomes "1" to begin decoding as shown in Fig. 12, the block clock B_c passes through a NAND-gate 65. While data of a line-drawing pixel block held on latches 73 are brought to latches 74

15 synchronizing with the fall time of the block clock B_c . The block clock B_c is input to an OSMV 66, which outputs a pulse (96) synchronizing with the rise-time of the block clock B_c , and the pulse (96) is input via an AND-gate 71 and an OR-gate 72 to the latches 73. The latches 73 hold data of next line-drawing pixel block on command of the pulse (96).

On the other hand, an OSMV 67 outputs a pulse (e) synchronizing with the rise time of the pulse (97) being input via a delay unit 103 from the OSMV 66. The pulse (e) is input to a delay unit 68 which outputs a pulse (98). A OSMV 69 outputs a pulse (f) synchronizing with the rise time of the pulse (98) being input from the delay unit 68. An OSMV 70 outputs a pulse (g) synchronizing with the pulse (f) being input from the OSMV 69. These pulses (e), (f) and (g) controls the timing of the decoder 5 as mentioned later.

35 In a case data of a line-drawing pixel block of "vacant" or "solid" are held on the latches 74, the judge data "0, 1" or "1, 0" are output from the latches 74 to an Exclusive OR-gate 75, and the E-OR-gate 75 outputs a signal of logic "1" to open AND-gates 78. When data of a line-drawing pixel block of "vacant" are on the latches 74, as a signal (99) of logic "0" is input to the AND-gates 78. Therefore the AND-gates 78 output 25 bit data where every bit is logic "0" via OR-gates 79 to latches 80. When line-drawing block data of "solid" are held on the latches 74, as a signal (99) of logic "1" is input to the AND-gates 78. Therefore the AND-gates 78 output 25 bit data where every bit is of logic "0" via the OR-gates

50 79 to the latches 80.

In this, the data of a line-drawing pixel block are input to the latches 80 synchronizing with the fall time of said pulse (f). The data of a line-drawing pixel block can be output synchronizing with any strobing signal. For example the rise time of the pulse (f) can provide a preferable output timing of the data to subsequent devices.

Meanwhile a signal of logic "1" output from the E-OR-gate 75 opens an AND-gates 82 to input data of the sequence number of the line-drawing blocks held on the latches 74 in a run-length form via the AND-gate 82 and an AND-gates 84 to a counter 85. While the signal of logic "1" from the E-OR-gate 75 also opens an AND-gate 89 which makes an inverted pulse (e)'

obtained by inverting the pulse (e) in an inverter 93 pass the gate through to be input via an OR-gate 91 and an inverter 92 to the load terminal of the counter 85 as a load signal. The load signal designates the counter 85 to take in the data of the sequence number of the line-drawing blocks being input from the AND-gates 82.

Meanwhile the load signal is input to the reset terminal of a flip-flop circuit 86 to reset its output to be logic "0". While a count-down pulse (g) is input to the counter 85. Therefore, if the pre-set number of the counter 85 is "1", it is made to be "0" when only one pulse (g) is input to the counter 85. When the pre-set number of the counter 85 becomes "0", the counter 85 outputs a borrow signal to the set terminal of the flip-flop circuit 86 which outputs a signal (h) of logic "1". When the signal (h) is input to the latches 74 and 73, the latches 74 takes in the data of a line-drawing pixel block held on the latches 73, and data of the next line-drawing pixel block are input to the latches 73.

In the case data of a line-drawing pixel blocks of "vacant" or "solid" are in a sequence, the latches 80 outputs 25 bit data in which each bit has a logic of "0" or "1" synchronizing with said pulse (f) until the preset number (the sequence number) of the counter 85 becomes "0". During in the above condition, the latches 74 doesn't take in data of the next line-drawing pixel block because the signal (h) output from the flip-flop circuit 86 is "0".

In the case data of a line-drawing pixel block are "hybrid in isolation", in other words when the judge data have value, "1, 1", the E-OR-gate 75 outputs a signal of logic "0" which opens AND-gates 77. Then the 25 bit line-drawing pattern data held on the latches 74 are input via the OR-gates 79 to the latches 80 in the same way as the previous case. At the same time, the judge data "1, 1" ((99), (100)) are output from the latches 74 to a NAND-gate 87. Unlike the next case, AND-gates 83 and 88 are not opened, but only the bit pattern data are output from the latches

110 In the case data of line-drawing blocks are "hybrid in a sequence", in other words when the judge data have value "0, 0", the E-OR-gate 75 outputs a signal of logic "0" which opens the AND-gates 77. Then the 25 bit pattern data held on the latches 74 are input via the OR-gate 79 to the latches 80 like in the previous case. At the same time, the NAND-gate 87 opens AND-gates 83 and 88. Then sequence number data of the line-drawing pixel blocks held on the latches 73 are input via the AND-gates 83 and 84 to the counter 85. While the inverted pulse (e)' is also input to the counter 85 via an AND-gate 90 and the AND-gate 91 and the inverter 92 as the load signal. On command of the load signal, the sequence number of the line-drawing blocks are set up on the counter 85 and begins to be counted down by the pulse (g). In the course of time, the bit pattern data held on the latches 80 are output synchronizing with the pulse (f) certain times corresponding to the sequence number. The

inverted pulse (e)' is also input to the flip-flop circuit 86, which outputs a signal (h) of logic "0" for preventing the data of the next line-drawing pixel block from being input to the latches 73 and 74. As mentioned before, the flip-flop circuit 86

outputs the signal of logic "1" on command of the borrow signal from the counter 85 to take in the data of the next line-drawing pixel block to the latches 73 and 74.

An OSMV 94 outputs a pulse signal synchronizing with the inverted pulse (e)' via the AND-gate 90. The pulse signal is output from a delay unit 95 as a signal (i) to the AND-gate 72, which gives a command to input the data of the next line-drawing pixel block from the disc memory 4 to the latches 73.

Thus obtained data of the line-drawing pixel blocks in the condition of "vacant", "solid" or "hybrid" are converted into for example 5x5 matrix data before being input to the recording head. Fig. 13 shows an example of a converter for performing the abovementioned conversion process. At first, on command of timing pulses TP₁ to TP₅ as shown in Fig. 14 obtained by using the pulse (f), a converter 105 composed of 25 AND-gates converts 25 bits serial data of a line-drawing pixel block into 5x5 matrix data (which correspond to one pictorial pixel). Then the 5x5 matrix data of the line-drawing pixel block are distributed to corresponding data group of separation colors Y, M, C and K according to the color codes included in the data and are output from corresponding AND-gates 107, 108, 109 and 110.

Fig. 15 shows a layout scanner system to which the method of this invention is applied. An original picture of pictorial components on a drum D₁ and an original picture of line-drawing components on a drum D₂ are scanned individually. That is, image data obtained by scanning the original picture of pictorial components A₁ with use of an input head IH₁ as colour component signals B, G and R are converted into colour separation image signals Y, M, C and K and undergo colour correction, gradation correction, under colour removal, sharpness emphasis etc. as required in a scanner data processor 20. The processed data from the data processor 20 are stored into a disc memory 120. On the other hand, image data obtained by scanning the original picture of line-drawing components A₂ with use of an input head IH₂ having an input unit as shown in Fig. 7 are condensed in the condensation circuit 2 to be binary image data and stored into the disc memory 4. The thus obtained data are input to a layout processor 7 under control of a CPU 6 to be displayed as a corresponding image on a monitor 8. Based on the displayed image on the screen of the monitor 8, an operator performs layout work, such as by a method disclosed in United States Patent Application Serial No. 517,982.

Pictorial image data and line-drawing image data which experienced a layout process are once stored in disc memories 121 and 122

respectively. Then the pictorial image data are compared to the dot patterns of a halftone dot generator 9 as disclosed in United States Patent No. 3,725,574 or in United States Patent application Serial No. 365,890 in order to be halftone image data for driving a recording head OH. While the line-drawing image data are decoded in the decoder 5 to be output to the recording head OH. This kind of recording method for the pictorial and the line-drawing data is disclosed in United States Patent application Serial Nos. 471,869 or 507,719.

Data of a line-drawing pixel block are expressed in a run-length form or in a bit pattern form appropriately case by case, which enables an image reproducing system to attain a higher condensation ratio consequently permitting a memory of lower capacity. Therefore, the cost of the image reproducing system is minimised.

CLAIMS

1. A method for condensing binary image data in reproducing images, comprising:

obtaining data of a line-drawing pixel block composed of m x n line-drawing pixels by an input scanning means or by a computation means; and generating a "word for expressing (I) and (II) or (I) and (III) of the following, (I) data for judging whether the pixels of the line-drawing pixel block are in a "vacant" (as herein defined) or "solid" (as herein defined) "hybrid" (as herein defined) condition, (II) run-length data of the number in one sequence of the line-drawing pixel blocks which are in the "vacant" or "solid" condition; and (III) m x n bit pattern data of a line-drawing pixel block in the "hybrid" condition.

2. A method as claimed in claim 1 in which, when line-drawing pixel blocks are in the "hybrid" condition in a sequence, the word is used for expressing the judge data and the m x n bit pattern data, and the next word is used for expressing the run-length data corresponding to the number of the line-drawing pixel blocks in the sequence.

3. A method as claimed in claim 1 in which colour data are given to the "word".

4. A method as claimed in claim 1 in which, when line-drawing image data are processed with pictorial image data, the line-drawing pixel block corresponds to one pictorial pixel.

5. A method as claimed in claim 1 in which a classification data for between a line-drawing pixel block and a pictorial pixel block are given to the "word".

6. A system for condensing binary image data in reproducing image, comprising:

means for obtaining data of a line-drawing pixel block composed of m x n line-drawing pixels by an input scanning means or by a computation means; and

means for generating a "word" for expressing (I) and (II) or (I) and (III) of the following, (I) data for judging whether the pixels of said line-drawing pixel block are in a "vacant" (as herein defined) or "solid" (as herein defined) or "hybrid" (as herein defined) condition; (II) run-length data of the

- number in one sequence of the line-drawing pixel blocks which are in the "vacant" or "solid" condition; and (iii) $m \times n$ bit pattern data of a line-drawing pixel block in the "hybrid" condition.
- 5 7. A system as claimed in claim 6, in which said "word" generating means comprises:
- (a) means for outputting said judge data for indicating a "vacant" or "solid" or "hybrid" condition;
- 10 (b) means for outputting said run-length data of the number in one sequence of line-drawing pixel blocks which are "vacant" or "solid" when the line-drawing pixel blocks are in a sequence; and
- 15 (c) means for outputting the $m \times n$ bit pattern data of a line-drawing pixel block which are "hybrid".
8. A system as claimed in Claim (7) in which the bit pattern data output means outputs the bit
- 20 pattern data of line-drawing pixel blocks of which are "hybrid" in a sequence for expressing them in the "word", and the run-length data output means outputs the sequence number of the line-drawing pixel block of which are "hybrid" for
- 25 expressing them in the next word.
9. A system as claimed in Claim (7) in which the judge data output means comprises:
- (a) a decoder for decoding the data of the line-drawing pixel block;
- 30 (b) two tri-state buffers which output the decoded data of the line-drawing pixel block which are "vacant" or "solid" from the decoder synchronizing with an output timing of the run-length data; and
- 35 (c) two other tri-state buffers which output the decoded data of the line-drawing pixel blocks which are "hybrid" from the decoder synchronizing with an output timing of the bit pattern data.
- 40 10. A system as claimed in Claim (7) in which the run-length data output means comprises:
- (a) a counter which counts the sequence number of the line-drawing pixel blocks which are "vacant" or "solid"; and
- 45 (b) a tri-state buffer which outputs data of the count number synchronizing with an output timing of the judge data.
11. A system as claimed in Claim (10) comprising a counter for counting the sequence number of the line-drawing pixel "hybrid" blocks.
- 50 12. A system as claimed in Claim (7) in which the bit pattern data output means comprises a tri-state buffer which outputs the bit pattern data of the line-drawing blocks synchronizing with the output timing of the judge data.
- 55 13. A system as claimed in claim 7, in which the "word" generator comprises a strobing signal generator which effect the judge data, and the run-length data or the bit pattern data of the line-drawing pixel block.
- 60 14. A method for condensing binary image data substantially as herein described with reference to any of the figures of the drawings.
15. A system for condensing binary image data
- 65 substantially as herein described with reference to any of the figures of the drawings.