OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(12) (19) (CA) **Brevet-Patent**

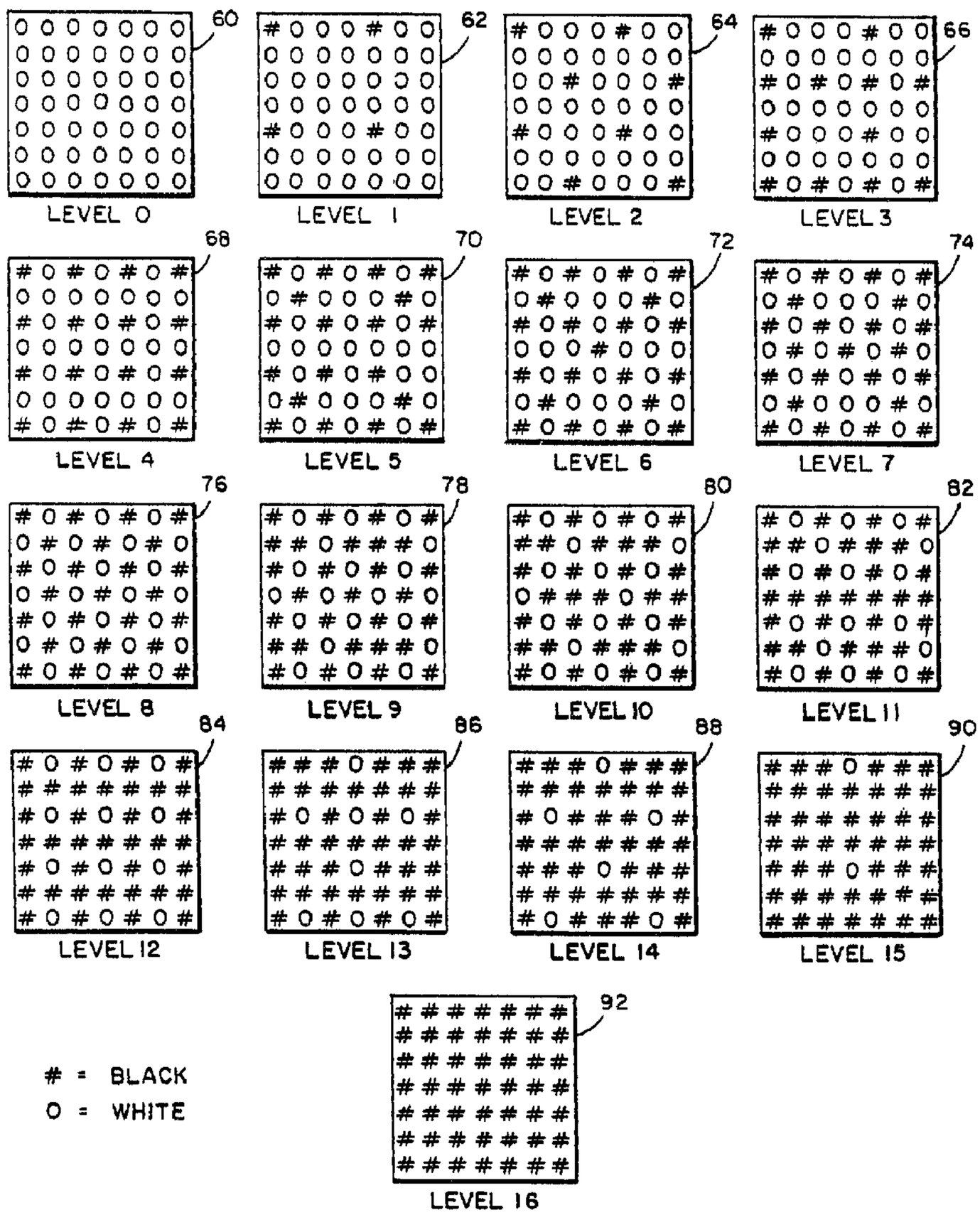
(11) (21) (C) **2,100,064**
(86) 1991/10/01
(87) 1992/07/11
(45) 2001/02/27

(72) Knowlton, Kenneth C., US
(73) KODAK LIMITED, GB

(51) Int.Cl.[5] H04N 1/40, G06F 15/68
(30) 1991/01/10 (639,465) US

(54) **DISPOSITIF ET METHODE DE SUPPRESSION DES SIGNAUX SUPERPOSES AUX SIGNAUX D'IMAGERIE**

(54) **IMAGE UNDITHERING APPARATUS AND METHOD**



(57) Un procédé permet à une représentation d'une image, produite par juxtaposition de pixels noirs et blancs, d'être convertie en une représentation en tons continus. Selon ce procédé de conversion d'un affichage par juxtaposition de pixels, les zones dont la

(57) A method enables a dithered representation of an image to be converted to a continuous tone representation of the image. This undithering method compares the regions of dithered representation with sections of dither patterns to determine an appropriate

OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(11) (21) (C) **2,100,064**
(86) 1991/10/01
(87) 1992/07/11
(45) 2001/02/27

représentation est obtenue par une juxtaposition de points, sont comparées avec des sections de configurations à juxtaposition de points afin de déterminer une représentation appropriée en tons continus qui aurait pu produire l'image obtenue par juxtaposition de pixels. Ce procédé peut être répété sur un plan spatial, pixel par pixel, jusqu'à ce que la représentation initiale tout entière de l'image soit convertie en une représentation en tons continus. Le procédé peut aussi être modifié en ce qui concerne la technique produisant des élément linéaires. En outre, le procédé peut être utilisé pour effectuer des modifications de la grandeur de l'image, y compris des miniaturisations et des agrandissements de l'image originale. Selon cette application, des parties produites par juxtaposition de points de l'image originale sont à nouveau traitées par juxtaposition de points de façon à produire une image de sortie d'une grandeur différente. De même, des parties de l'image originale traitées selon la technique d'éléments linéaires sont traitées sous forme de valeurs d'échelle de gris et à nouveau traitées par juxtaposition de points pour produire des représentations d'éléments linéaires d'une grandeur différente.

continuous tone representation from which the dithered image could have resulted. When the continuous tone representation is found the region is converted into a continuous tone representation. This procedure may be spatially iterated on a pixel by pixel basis until all of the initial representation of the image is converted into a continuous tone representation. The procedure may also be modified to account for line art. Furthermore, the procedure may be applied to produce adjustments in the size of the picture including miniaturizations and enlargements of the original image. In accordance with this application, dithered portions of the original image are redithered in a manner to produce an alternative size output image. Similarly line art portions of the original image are treated as grey scale values and redithered to produce the alternative size representations of the line art.

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :
H04N 1/40, G06F 15/68

A2

2100064

(11) International Publication Number: WO 92/12594

(43) International Publication Date: 23 July 1992 (23.07.92)

(21) International Application Number: PCT/US91/07212

(22) International Filing Date: 1 October 1991 (01.10.91)

(30) Priority data:
639,465 10 January 1991 (10.01.91) US

(71) Applicant: WANG LABORATORIES, INC. [US/US]; One Industrial Avenue, Lowell, MA 01851 (US).

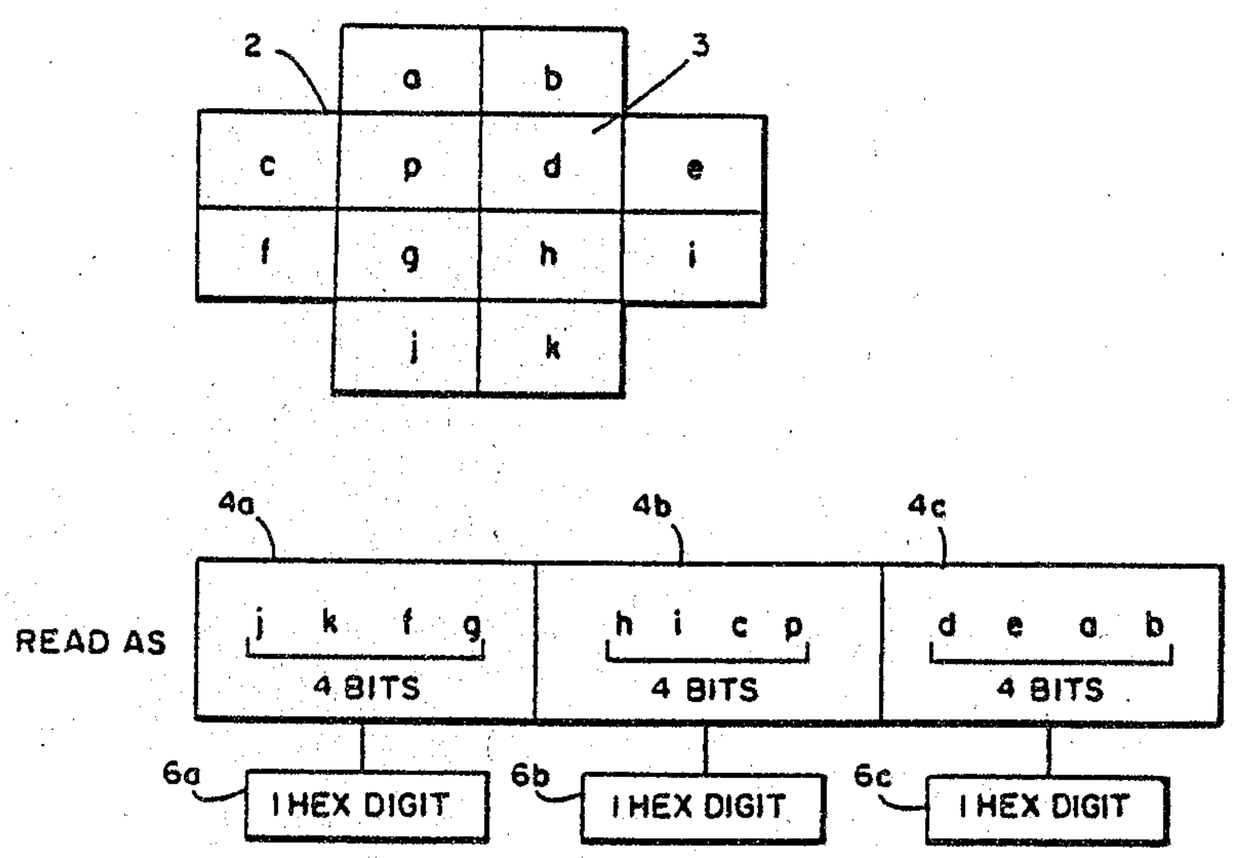
(72) Inventor: KNOWLTON, Kenneth, C. ; 51 Pond View Drive, Merrimack, NH 03045 (US).

(74) Agents: SHANAHAN, Michael, H. et al.; Wang Laboratories, Inc., One Industrial Avenue, Lowell, MA 01851 (US).

(81) Designated States: AT (European patent), AU, BE (European patent), CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: IMAGE UNDITHERING APPARATUS AND METHOD



(57) Abstract

A method enables a dithered representation of an image to be converted to a continuous tone representation of the image. This undithering method compares the regions of dithered representation with sections of dither patterns to determine an appropriate continuous tone representation from which the dithered image could have resulted. When the continuous tone representation is found the region is converted into a continuous tone representation. This procedure may be spatially iterated on a pixel by pixel basis until all of the initial representation of the image is converted into a continuous tone representation. The procedure may also be modified to account for line art. Furthermore, the procedure may be applied to produce adjustments in the size of the picture including miniaturizations and enlargements of the original image. In accordance with this application, dithered portions of the original image are redithered in a manner to produce an alternative size output image. Similarly line art portions of the original image are treated as grey scale values and redithered to produce the alternative size representations of the line art.

IMAGE UNDITHERING APPARATUS AND METHOD

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

Background of the Invention

In discretely quantized continuous tone pictures, each pixel assumes a tone selected from a specified scale of tones. Typically, each tone is assigned a discrete value that uniquely identifies the tone. The tone of each pixel may, thus, be characterized by its corresponding discrete value. This discrete value is known as the pixel value. The use of pixel values may be illustrated by looking at a sample scale such as the grey scale which ranges from white to black. In a 17 level grey scale, the possible pixel values may run from 0 indicating pure white to 16 indicating pure black. The values 1 through 15 indicate successive degrees of darker grey pixels.

Often times it is difficult for display devices to display such continuous tone pixels, for many devices are limited to displaying only white and black pixels. Such devices, therefore, use spatial dithering or digital half-toning to provide the illusion of continuous tone pictures (hereinafter this process shall be referred to as dithering). Dithering uses only black and white pixels to provide the illusion of a continuous tone picture. Specifically, it intermixes the black and white pixels in a predetermined manner to simulate various tones of grey. Two well known dithering approaches are clustered-dot ordered dithering and dispersed-dot ordered dithering (sometimes referred to as scattered dot dithering).

Both of these dithering strategies are classified as ordered dither

algorithms. They rely on a deterministic periodic threshold array for converting a continuous tone image into a dithered image. These types of dithering algorithms are referred to as "ordered" rather than "random" algorithms because the thresholds in the array are organized and applied in an ordered sequence. In dispersed dot dithering, the resulting output is comprised of dispersed dots, whereas in clustered dot dithering the resulting output is comprised of clusters of dots. More information on dithering can be found in Limb, J.O., "Design of dither waveforms for quanitized visual signals," Bell Sys. Tech. J., Sep. 1969, pp. 2555-2582; Bayer, B.E., "An optimum method for rendition of continuous tone pictures," Proc. IEEE Conf. Commun., (1973), pp. (26-11)-(26-15); and Knowlton, K. and Harmon, L., "Computer produced grey scales," Computer Graphics and Image Proc., vol. 1 (1972), pp. 1-20.

An example of a dispersed dot threshold array is:

1	13	4	16
9	5	12	8
3	15	2	14
11	7	10	6

In applying such a threshold array to a continuous tone image, the array or grid of threshold numbers is overlayed repeatedly on the pixels of the quanitized continuous tone image in both the horizontal and vertical directions. Each pixel in the image is then compared with the corresponding grid threshold value that overlays it. If the grey scale pixel value equals or exceeds the corresponding grid threshold value, the resulting output dithered pixel is a black pixel. In contrast, if the grey scale value of the pixel is less than the corresponding grid threshold value, the resulting output dithered pixel is a white pixel.

As an illustration of how the above threshold array is used, assume that the array is placed over a group of pixels in a grey scale picture. For the pixel that underlies the threshold value in the upper left hand corner of the grid, the pixel

must have a value of 1 or more in order to produce a dithered black pixel. Otherwise, a dithered white pixel is produced. Similarly, the next pixel to the right in the grey scale picture must have a value of 13 or greater in order to produce a dithered black pixel. The grid is applied to consecutive groups of pixels in the grey scale picture, i.e., across rows and columns of the pixels, until all of the rows and columns of the image have been successfully dithered.

The dither threshold array shown above is one of many different possible threshold arrays and is purely illustrative. The grid could equally as well have a lowest level of 0 rather than 1. Furthermore, grids of different sizes may be used. For example, 8x8 grids are often used.

Summary of the Invention

The present invention converts a dithered representation of an image to a continuous tone representation of the image by comparing regions of the dithered representation of the image with predetermined patterns of pixels. The predetermined patterns of pixels identify possible continuous tone representations from which the regions of the dithered representation could have originated. An example of the type of patterns that are used for such a comparision is a dispersed dot dither pattern. The comparision yields at least one corresponding continuous tone representation from which the dither pattern could have originated. In accordance with the method, once the corresponding continuous tone representation is selected, a corresponding continuous tone representation is assigned to the respective region of the dithered representation. Preferably the assignment proceeds on a pixel by pixel basis. This method may be readily performed by a data processing system.

The method need not be limited purely to dithered representations; rather it may be modified so as to operate for pictures that are comprised of both dithered regions and line art regions. Line art refers to black letters, lines or shapes on a white background or white letters, lines or shapes on a black background. In view of the modification, the method proceeds as previously described except that

2100064

when line art is encountered, the line art is not converted into a continuous tone representation. Instead, the line art is left alone so that the resulting image duplicates the input image for the regions that are deemed as line art.

One useful approach for comparing subject regions to the predetermined pixel patterns is to define a window of pixels in which a windowed sample of pixels from a source dithered image lie. This windowed sample of pixels is then compared as described above to yield a corresponding continuous tone representation of the source pixels in the window. The window may then be shifted to obtain a new windowed sample of pixels. Preferably, the window is shifted from one end to an opposite end across each of a series of lines of the image, such as in a left to right fashion until all pixels have been examined. By adopting this approach, the window may be moved across the entire source (i.e. dithered) representation of the image to grab successive windowed samples and, thus, convert the entire input image into an appropriate continuous tone output. Preferably the window is substantially t- shaped. Such a window works particularly well for a 17 level grey scale continuous tone representation.

In accordance with one embodiment, the determination of whether a portion of an image is dithered is performed by encoding pixels in the windowed sample of the image as a string of bits. Once they are encoded, the string of bits can be compared with small spatial pixel patterns held in memory as encoded bits to determine if the pixels of the image match the pixels held in memory. If such a match does occur, it is presumed that a portion of the image is dithered and the portion is designated as such.

The present invention also embodies a simpler method of determining whether a region of a picture is dithered. In particular, the region is examined to determine whether all the white pixels in the region are disconnected (i.e., it has no orthogonal white neighbors). If the pixels are disconnected, each white pixel has only black pixels as orthogonally adjacent neighbors. Similarly the same approach is adopted for each black pixel in the region. If either the white pixels or the black pixels are all disconnected, the region is deemed to be dithered.

Otherwise, the region is deemed to be not dithered.

In accordance with the present invention, an additional method generates a pixel in a reduced size version of an original image. The additional method compares a source region of the image to dither patterns such as previously described to designate if the source region is dithered. The source region is comprised of an inner subregion and a surrounding outer subregion. If the source region is designated as dithered, the inner subregion of the source region is redithered to generate a single pixel value, wherein the inner subregion is viewed as a single pixel having a grey scale value assigned to the source region. This redithering results in a reduced size version of the source region. On the other hand, if the source region is designated as not dithered, the inner subregion is compared with the outer subregion, and a pixel having a value based on the number of black pixels in the outer subregion relative to the number of black pixels in the inner subregion is generated. Preferably, the inner subregion is comprised of 4 pixels and the outer region is comprised of 8 pixels that surround the inner subregion (for a 17 level grey scale). Such a source region results in a miniaturization that is linearly half as large as the original image. The method can be used iteratively to produce reduced sized versions of the original image which are reduced in size linearly four times, eight times, etc.

The above described methods may be implemented in a data processing system comprising a digital processor and a memory. The memory may comprise a first table of grey scale values indexed by given pixel patterns. Alternatively, an associative memory may be used. The first table or associative memory preferably holds grey scale values for both dithered patterns and line art. The memory also holds a second table of dithered pixel patterns associated with predetermined grey scale values. A procedure executed by the processor accesses the first table to provide grey scale values for the designated dithered regions and then accesses the second table to obtain bits to be used for redithering.

If a reduction in the size of the original image is to be performed, the processor also redithers adjacent regions of the image according to the grey scale

2100064

values in a manner such that a reduced size image (such as a shrunken icon image) is generated. This process does not explicitly branch for the line art or dithering case but rather operates as if uniformly redithering the image; the tables, however, are constructed so that for the line art case, the pixels are re-rendered in accordance with optimal treatment for line art.

The present invention is not limited to production of shrunken replicas of the original image. The shrunken replicas represent a specific instance of a broader method encompassed within the present invention. Specifically, the present invention includes a more general method of producing a replica of a source picture such that the replica is a different size than the source picture. The different size replica may be an enlargment or, alternatively, a reduction of the source picture.

In accordance with this approach, for each pixel in the replica, the source pixels which contribute to the pixel in the replica are identified. Next, a decision is made, by weighted voting, as to whether the destination pixel is to be considered as having come from line art or dithering. In accordance with that decision, for each line art destination pixel, a value of "black" or "white" is determined according to a weighted average of contributing source pixels. For each grey scale destination pixel, a grey scale value is determined for each contributing pixel in the source picture. Based on the respective contributions from pixels in the source picture and grey scale values of the contributing source pixels, a cummulative grey scale value is determined. This cumulative grey scale value is then associated with each grey scale pixel in the replica for redithering purposes. Utilizing the cumulative grey scale values, the source picture is dithered to produce the different size destination replica of the source picture.

As mentioned above, the replica may be smaller, larger or have a different aspect ratio than the source picture.

Brief Description of the Drawings

Figure 1 illustrates a sample window pattern and how the pixels are read

from the sample window.

Figure 2 depicts an example case of the sample window pattern of Figure 1 and how the pattern may be referenced as binary digits or as hexadecimal digits.

Figure 3 illustrates the dither patterns of a 7x7 region for each level of a 17 level grey scale.

Figure 4 depicts the 16 window positions that the sample window of Figure 1 may assume within a 7x7 region.

Figure 5 lists the hexadecimal patterns for the dither patterns associated with each of the 17 grey levels.

Figure 6 is a consolidation of the digit patterns depicted in Figure 5 such that there are only even numbered levels.

Figure 7 depicts the digit patterns of Figure 6 with redundancies removed.

Figures 8a, 8b and 8c illustrate sample window positions and the corresponding output produced for the sample window positions.

Figure 9 is a picture comprised of line art and dithered regions.

Figures 10a, 10b and 10c show shrunken versions of the Figure 9 picture.

Figure 11 depicts three example window patterns and how they are interpreted as output.

Figure 12 illustrates a windowed sampled within a set of words fetched by the system.

Figures 13a, 13b, 13c, 13d and 13e depict how logical operators are used to compose a windowed sample as an index to the look-up table.

Figures 14a, 14b and 14c illustrate sample window positions for a dithered region of an image.

Figures 15a and 15b illustrate the sample window positions and corresponding output produced for those sample window positions of Figures 14b and 14c.

Figure 15c illustrates portions of a dither patterns stored in an "ansbyt" lookup table.

Figures 16a and 16b illustrate the basic components of the data processing

2100064

system used to implement the undithering and shrinking procedures.

Figure 17 illustrates how the bits from bytes b0-b7 are used to form the output byte.

Figure 18 is a flowchart of the major steps of an image reduction method embodying the present invention.

Figure 19 illustrates a generalized approach to re-sizing.

Figures 20a and 20b illustrate an optional compression method employable in embodiments of the present invention.

Detailed Description of the Preferred Embodiment

The preferred embodiment of the present invention provides a means for approximate undithering of an image having dithered regions, that is it converts a source picture with dithered regions into a destination continuous tone (grey scale) picture. As such, it is an approximate reversal of the dithering process. It achieves the conversion on a pixel by pixel basis in an iterative fashion and suffers from statistically few errors. The source picture need not be a completely dithered picture; rather, it may be a picture wherein only portions are dithered. The present invention examines a portion of the source picture that may contain dithering to find an exact match with a portion of a uniformly grey dither result. If a match is found, it is presumed that the portion of the source picture is the product of dithering. Given this presumption, a selected pixel within the portion of the source picture currently being examined is converted into an appropriate destination grey scale pixel having an approximately correct grey scale value (i.e. the pixel is undithered). The approximate grey scale value is determined during the matching process by deciding which grey scale value could have produced the resulting portion of the source picture. If, on the other hand, there is no exact match, the corresponding pixel in the destination picture is given the current black or white state of the source pixel. All of the pixels are examined in such a manner until the entire source picture has been examined and converted.

The conversion process is performed by examining successive samples of the

source picture. Each successive sample of the source picture is taken in a pattern that utilizes a window which defines the pixels currently being examined. The pixels within the window will be referred to hereinafter as a windowed sample. A suitable window 2 defining a windowed sample 3 in a dithered source picture is shown in Figure 1. It should be appreciated that different window configurations may be utilized especially for grey scales having other than 17 levels.

The sample window configuration 2 shown in Figure 1 works particularly well for a 17 level grey scale ranging from 0 denoting white to 16 denoting black. The use of this window configuration 2 will be illustrated for a picture comprised of black line art on a white background and dithered regions that have been dispersed dot dithered according to the 4x4 repeating threshold array given above in the Background section. The undithering works best with application to dispersed dot dithering because a smaller neighborhood of pixels suffices to identify a locality as dithering than required with other dithering techniques.

The windowed sample 3 in the window 2 shown in Figure 1 is comprised of twelve pixels organized into a shape resembling a Greek cross. Each of the pixels in the windowed sample 3 is represented as a bit having a value of zero if it is white and a value of one if it is black. As can be seen in Figure 1, eleven of these twelve pixels are denoted by letters ranging from "a" to "k". The remaining pixel is denoted by the letter "P". The pixel "P" identifies the source pixel in the sample that is converted into a destination continuous tone representation if a match is found (as discussed below). Hence, the entire pattern of pixels in the windowed sample 3 is examined for determining whether a match with a dither product exists, but only a single pixel in the windowed sample 3 is converted at a time. To simplify referencing of the bits in the windowed sample 3, the bits are referenced in a predefined order as shown in Figure 1. In particular, the bits are referenced in four-bit chunks expressed in binary form as groups 4a, 4b and 4c. The first group 4a includes pixels "j", "k", "f", and "g" (in that order), whereas the second group 4b is comprised of pixels "h", "i", "c", and "P" (in that order). The third and final group 4c is comprised of pixels "d", "e", "a" and "b" (in that order). Each of

these groups 4a, 4b and 4c may be more compactly referenced as a single hexadecimal digit 6a, 6b, 6c, respectively. As a result, the entire contents of the windowed sample may be referenced as three hexadecimal digits 6a, 6b and 6c. This particular referencing scheme is purely a matter of programming or notational preference. It is not a necessary element of the described method.

An example of a windowed sample along with the corresponding bit pattern encoding the windowed sample is shown in Figure 2. The windowed sample 3 is comprised of a series of data bits having values of either zero (indicating a white pixel) or one (indicating a black pixel). The windowed sample 3 shown in Figure 2 is referenced as three groups of four bits each where the first group 4a equals "1010"; the second group 4b equals "1001"; and the third group 4c equals "1101". When the three groups 4a, 4b and 4c of four bits are referenced in hexadecimal notation, they are expressed as "A", "9" and "D", respectively, or when combined as "A9D".

Given the above description, it must be considered how the window 2 is applied to the source picture. In particular, it must be considered in what sequence and where the sample window 2 is placed on the source picture. This issue is of particular importance because the relative position of the window on the dither pattern must be taken into account, for different sample window positions on a given dither pattern yield different pixel patterns. As an illustration, suppose that the sample window 2 is positioned within a 7 pixel by 7 pixel dithered region of a source picture. Since this 7x7 region is dithered, the region is made of a specific periodic pattern of pixels. The pattern of pixels is dictated by the grey scale values of an original region from which the dither pattern was produced. Figure 3 depicts the 17 pixel patterns 60-92 produced when the 7x7 pixel region is dithered for locally uniform areas of the 17 levels of grey scale. Each dither pattern 60-92, shown in Figure 3, is associated with a particular grey scale level value. The #'s indicate black pixels and the 0's indicate white pixels in Figure 3. The patterns shown in Figure 3 are those that result from applying the threshold array discussed in the Background section to the 7x7

region. As can be seen in Figure 3, the level zero dither pattern is entirely white, and the level 16 pattern is entirely black. Moreover, the level 8 pattern is comprised of exactly half white pixels and half black pixels intermixed equally in a checkerboard pattern.

For such a 7x7 region, the window 2 can fit into the region in 16 the different positions 94-124 that are illustrated in Figure 4. As such, for any 7x7 region, the sample window 2 may capture as many as 16 distinct patterns for certain grey levels of dithering, one pattern for each window position in the 7x7 region. A larger sample, such as an 8x8 region is not used because the additional window patterns obtained with such a larger sample would be redundant with those obtained in the 7x7 region.

Figure 5 shows, in table form, the different hexadecimal digit patterns that the respective sample window positions may yield for each of the different grey scale levels (indicated on the left-hand side of the table). Redundancy within given levels has been removed. It will be noted, for example, that the grey level having all zeroes which represents solid white yields only one window pattern (i.e. "000"), and the grey level having all ones which represents solid black yields only one window pattern (i.e. "FFF"), whereas the fine checkerboard of grey level 8 yields only two patterns "A95" and "56A" which differ in spatial phase. On the other hand, certain grey scale levels such as level 11 have sixteen different patterns (i.e. one for each window position). Some levels have fewer than sixteen different samples because the resultant dither pattern exhibits regularity.

It is apparent that there is a great deal of interpretative ambiguity between levels (i.e. different grey scale levels may be dithered such that the window patterns taken from the dither result of the different grey scale levels are the same), because the pixel that makes the difference between the given level and the next level is not caught by the window position. For instance, both a level 2 dither result and a level 3 dither result can generate a sample pattern encoded in hexadecimal digits as "201". One means of dealing with the ambiguity is to consolidate the dither patterns into half as many levels so that adjacent levels

2100064

that would otherwise exhibit ambiguity are merged into a single level. One possible consolidation is shown in Figure 6. In accordance with this grouping, the sample patterns for the odd numbered levels are merged into adjacent even numbered levels. For example, the first four sample patterns of level 3 are merged into interpretative level 4. The remaining 12 sample patterns of level 3, however, are merged into level 2. This merger provides a convenient way of resolving ambiguity while still producing satisfactory results. The interpretation chart of Figure 6 may be further consolidated into that of Figure 7 wherein each pattern appears only once.

The sixteen different sample window positions 94-124 (Figure 4) within the 7x7 region are sufficient to obtain the various distinct samples that can be obtained by the window when applied to dither results. As such, the permutations shown in Figure 7 represent all the possible sample patterns that may be obtained for dither results, provided that no more than one transition between levels traverses the windowed area. It is equally apparent that the sample window need not be applied strictly to 7x7 regions of a picture, but rather may be applied progressively from left to right across successive lines in a top to bottom fashion.

An example of the positioning of the window 2 is shown in Figure 8a where the window 2 is in window position 18. The source pixel P 10 (surrounded by a dotted box in Figure 8a) is the pixel that is converted to a destination grey scale pixel. The windowed sample 3 shown in Figure 8a may be expressed as "280" in hexadecimal notation according to the previously discussed scheme of Figure 1. This value "280" is compared with the dithered pattern results. From Figure 5, it is apparent that the pixel pattern is associated with a level 3 grey scale, however because of the consolidation, the pixel pattern is interpreted as originating from a level 2 grey scale. In particular, an examination of the table in Figure 7 reveals that the "280" hexadecimal pattern is to be interpreted as a level 2 grey scale. Accordingly, to convert the source pixel 10 to a destination grey scale representation, a grey scale value of 2 is placed in the destination position 12

(Figure 8c) corresponding with the source pixel 10.

Since the first source pixel 10 has been converted, the window 2 is next moved over to the right by a single bit to window position 19 shown in Figure 8b. As a result of the shifting of the sample window 2 to position 19, the pixel to be converted in the windowed sample 3 is now pixel 14 (surrounded by a dotted box in Figure 8b). The new value for the windowed sample 3 is "140" in hexadecimal notation. The windowed sample 3 is interpreted as originating from a level 2 grey scale such as shown in Figure 7. Accordingly, a grey scale value of 2 is placed in the destination position 20 that corresponds with the source pixel 14. In general, this process is repeated until all of the pixels in a source picture have been examined and converted, if necessary, (as mentioned previously, if the comparison of the sample pattern indicates no match with a dither result, the original pixel value becomes the value of the corresponding output pixel). If the entire source picture is dithered, an alternative approach may be adopted. Specifically, the percentage of black pixels to white pixels in the windowed sample 3 may be used to determine the grey scale value.

The major method described above for undithering performs quite well with minimal errors. The errors to which it is prone include erroneously interpreting line art as dithering. This variety of error occurs seldom because lines and text seldom result in speckled pixel arrays such as those that originate from dithering. Typically, these errors are noticeable only upon very meticulous scrutiny of the destination picture.

Another category of error that may occur is interpretation of dithering as line art. Such an error occurs with reasonable frequency. This type of error arises where the source grey scale picture exhibits spatially rapid change within the windowed samples, such as in the case of pixel patterns spanning more than one transistion between levels. (In general, a single grey scale level transition causes a sample to look like either a piece of the higher level or a piece of the lower level because at most one of the differentiating pixels is caught in the sample). The resulting pixel pattern in the windowed sample does not exhibit a

stored dither pattern. Thus, the pixel pattern is treated as line art rather than dithering. The result of the error is to apply the line art method such as local contrast enhancement which may have the visual effect of edge sharpening. This effect is typically not deleterious.

Another error that may occur is properly recognizing line art as line art but producing black pixels in the destination picture where white pixels were desired. This variety of error does not occur frequently and, in any event, is a failure of the line art method employed (discussed later) as opposed to the undithering method.

A final error that may occur is to properly recognize dithering as dithering but to render the wrong grey scale values for pixels in the destination picture. This error occurs because of the consolidation and occurs quite often. Nevertheless, this error is not problematic because the resulting grey scale output is commonly only off by a single level.

A much simpler method for designating whether a source region is dithered is also embodied within the present invention. In accordance with this simpler method, samples are taken as described above with the sample window 2 (Figure 1). However, instead of comparing the spatial pixel pattern of the windowed sample with dither patterns, the connectivity of the white or black pixels is examined. In particular, if either none of the white pixels, or none of the black pixels have like colored orthogonal neighbors, the pixel pattern is designated as a dither product of the grey level indicated by the percentage of black pixels in the windowed sample. This approach works only for dispersed dot dithering.

The undithering method may be readily implemented by a data processing system such as that shown in Figures 16a and 16b. The data processing system includes a processor 210 for executing procedures to perform the described method. The data processing system may also include an associative memory 212 such as shown in Figure 16a to match sample patterns with dither result patterns. A table 214 (Figure 16b) may, instead, be used to store the dither result patterns of Figures 5-7. In the table 214, the hexadecimal encoding of the

windowed sample is used as an index value to look up a grey scale value.

The undithering method has uses in various picture processing operations. For such operations, better results can be achieved by working from earlier generation images or approximations of such earlier generation images. Undithering provides the capability to accurately approximate such earlier generation images. Examples of such picture processing operations include: conversion to cluster-dot dither for hard copy output, conversion to other than the original number of grey scale levels, conversion to different pixel aspect ratios, and operations such as edge detection, histogram leveling and contrast enhancement.

The above method and data processing system may also be utilized to produce alternative sized images of pictures or documents referred to hereinafter as stamps. This process of producing stamps is known as stamp making. For more information on stamps and stamp making see copending Patent Application No. 07/245,419, entitled "Document Manipulation in a Data Processing System". Only a few modifications need to be implemented relative to the above described method to enable the production of stamps. Figure 9 illustrates a picture comprised of both line art and dithered images. By adopting the stamp-generating approach of the present invention, a half-sized stamp such as that shown in Figure 10a, a quarter-sized stamp such as that shown in Figure 10b and an eighth-sized stamp such as that shown in Figure 10c may be generated by performing successive iterations of the modified algorithm. In addition, many other sizes of stamps may be produced. The preferred embodiment operates by generating a two times reduction of the original image at each step. By applying this process iteratively, stamps of one half size, one fourth size, one eighth size and so on, are generated. Like the previously described method, this approach will be illustrated with respect to a 4x4 threshold array for dithering and with the same sample window configuration 2 as shown in Figure 1.

The stamp producing method takes the four inner pixels of the windowed

sample 3 (denoted as "P", "d", "g" and "h" in Figure 1, respectively) and produces a corresponding single output pixel. The number of pixels in the sample window is limited to minimize computational requirements and decrease the size of the look-up table. The inner region of the sample region is a 2x2 region, whereas the output is a 1x1 region; hence, it is apparent that the output is reduced in size by a linear factor of 2.

The stamp producing algorithm first determines whether the pixel pattern within the window is likely to have arisen from a dither of a grey scale or from line art. If the windowed sample is likely to have arisen from a dither of a grey scale, an appropriate grey scale value that could have produced the dither product of the sample window is found using an approach like that described above. Once the grey scale value is found, the inner four pixels are redithered to produce a single output pixel. The details of how the redithering occurs will be given below. If it is determined that the windowed sample contains a pixel pattern that is likely to be line art, the four inner pixels "P", "d", "g" and "h" are compared with the eight surrounding neighbor pixels "a", "b", "c", "e", "f", "i", "j", and "k". Based upon this comparison, the system generates an appropriate output.

The general approach adopted for line art comprises counting the number of black pixels within the inner region of the windowed sample and comparing that number of black pixels with the number of black pixels in the outer neighboring region of eight pixels. If the number of black pixels in the outer neighboring region does not exceed twice the number of black pixels in the inner region, the output destination pixel is black. The rationale for assigning the output pixel a black value is that the inner region is not lighter than its neighborhood. This case includes the instance wherein twice the number of inner region black pixels equals the number of black pixels in the outer neighboring region. In general, it is desirable to favor the minority or foreground color in the case of ties. Thus, in the present case, black is favored because it is assumed that the coloring scheme is black on a white background. On the other hand, if a black background is

used, the ties should favor white pixels and the rule is altered accordingly. On the other hand, if the number of black pixels in the neighboring region exceeds twice the number of black pixels in the inner region, the output pixel is white. There is one exception to the application of this general rule. If the inner region is entirely white, the output region is automatically white. The effect of this procedure, for line art, is to accentuate the difference between the source 2x2 set and its immediate neighborhood and also to display the enhanced difference as a single black or white pixel.

Figure 11 provides three illustrations of the line art strategy in operation. In particular, for windowed sample 30, there is one black pixel in the inner region and one black pixel in the outer region. As such, since two times the number of black pixels in the inner region equals two and there is a single black pixel in the outer (neighboring) region, the output pixel is black pursuant the previously described rule. In the windowed sample 32, on the other hand, there are seven black pixels in the outside region and only one black pixel in the inner region. There are more than twice the number of black pixels in the outside region as there are in the inner region. The output pixel is, therefore, a white pixel. Lastly, as shown in windowed sample 34, there are all white pixels in the outside region and all white pixels in the inner region. Instead of comparing the number of pixels in the inner and outer regions, the exception to the rule is implemented so that the output is a white pixel.

When the windowed sample is found to be the result of dithering, the steps involved are somewhat more complex than in the line art case. As mentioned above, the basic approach is to determine the grey scale value for the dither pattern of the windowed sample and then, to redither to produce only a single pixel output for the 4 pixels in the inner region. The output pixel must be chosen from the appropriate X and Y phases of the resulting pixel's location because the redithering pattern is phase locked to the destination grid.

Two procedures, "shnkinit" and "shrink2X", which are encoded as software, process lines of a picture to produce the dithered and line art output in the form

of a one half-sized stamp. How the redithering is performed in this context will be described with reference to these two procedures. Source code listings of the procedures encoded in the computer language C are included in the attached Appendix.

The system embodying these two procedures operates by relying on two tables. The first table of concern is the look-up table denoted as "lut". This table is much like the table suggested for the previously described method (Figures 5-7). For each possible windowed sample that is the product of dithering, an appropriate grey scale value is stored in the table. The three hexadecimal digits that characterize a windowed sample (6a, 6b and 6c in Figure 1) are used as an index to look up the proper grey scale value in the table "lut". The table also provides the appropriate line art characterization given that the line art output can be characterized as a grey scale 0 for a white pixel or a grey scale 16 for a black pixel.

The table "lut" is initialized in the procedure "shnkinit". The table is comprised of 4,096 entries wherein a separate entry is provided for all of the 2^{12} ($2^{12} = 4096$) possible bit patterns that the windowed sample may assume. The "shnkinit" procedure begins initially by placing values of 0 and 16 corresponding to the line art results for each of the 4096 patterns in each slot of the table. To determine the line art results, "shnkinit" counts the number of bits that are black within the inner region of the windowed sample as well as the number of bits in the outer neighboring region of the windowed sample and compares them. If the number of black bits in the outer region does not exceed twice the number of black bits in the inner region, a value of 16 is placed in the table entry to indicate a black pixel. Likewise, if the outer region has more than twice the number of black pixels, a value of zero indicating a white pixel is placed in the table entry. Furthermore, if all of the inner region pixels are white, a value of zero is placed in the table entry regardless of the number of black pixels in the outer region. It is, thus, apparent that this code implements the rule previously described for line art. The rule is applied for all 4,096 entries even those that may be the product of

dithering.

Having filled the table with the proper line art characterization for each sample pattern, the system proceeds to over-write the grey scale values in the table for the sample patterns which, if found, are likely to be results of dithering. This is done by use of the "ini" procedure. As is evident by the code in the Appendix, the "ini" procedure accepts a single parameter which is an index that is used in conjunction with the variable "newentry". The parameter "index" is comprised of the hexadecimal digits that encode the sample pattern. The procedure "ini" assigns a grey scale value equal to "newentry" to the table entry at the index, but it also assigns 16 minus "newentry" as the grey scale value for the table entry at 4,095 minus the index. "ini" is instituted in this way to shorten the program listing by exploiting the symmetry between the two halves of the table. Moreover, "newentry" only assumes even grey scale values such as in Figure 7 to resolve ambiguity of interpretation.

The "lut" table is used to determine the proper grey scale value for the sample window pattern being examined. If the window pattern is likely to be line art, the output is a "grey scale" value of 0 or 16 which redithers into a white pixel and a black pixel, respectively, regardless of the X and Y destination phase. On the other hand, if the sample window pattern is likely to be the product of dithering, the grey scale value (i.e. the condensed grey scale table value of Figure 7) is used to redither and produce a dithered output.

To fully understand how the "shrink2X" procedure produces the output, it is necessary to first understand the second table exploited by this process. The second table is a 4x17 array denoted as "ansbyt". "ansbyt" is perhaps best viewed as being comprised of 4 separate tables, one for each vertical phase (i.e. row) of the 4x4 dither results. Each table holds a series of bits of the dither patterns that result when the threshold array previously described is applied to a 4x8 region of pixels having a grey scale value equal to the index of the table entry. Each entry in the array is comprised of 2 hexadecimal digits (i.e. 4 binary bits per hexadecimal digit) and, thus, is the equivalent of one byte (8 bits). The entries

are stored according to the grey scale values and the corresponding row of dither results (referred to as the Y phase of the output). Thus, the first set of 17output byte patterns is used for output lines 0, 4, 8, 12,...; the second set is used for output lines 1, 5, 9, 13,...; etc.

The first entry in the first table provides the first row that would result if the threshold array were applied to a 4x8 region of zero level pixels. Similarly, the second entry in the first table is the first row of the dither result for a 4x8 region of level 1 pixels. Moreover, the second entry in the second table is the second row of the dither result for a 4x8 region of level 1 pixels. All of the entries corresponding to even level grey scale values are zero so that "ansbyt" is consistent with the condensed approach of Figure 7 which resolves interpretative ambiguity.

Given this understanding of the "ansbyt" table, the following discussion will focus on the "shrink2X" procedure. As mentioned above, the "shrink2X" procedure serves a vital role in the generation of dithered bits in the shrunken stamp. Several parameters are passed to this procedure. First, four line pointers are passed as parameters. Each of these pointers points to a separate line in the source picture. In particular, the line pointers "topp" and "bott" point to the top line and the bottom line, respectively, of the pair of lines being compressed by "shrink2X". The pointer "abov" points to the line above the top line, and the pointer "next" points to the next line after the bottom line. The procedure "shrink2X" is also passed, as a parameter, a line pointer "dest" to the destination output line buffer. The final two parameters to this procedure, n and levmod4, indicate the number of characters in the destination line and the y destination phase, respectively.

The movement of the window and the processing of bits within the sample window must account for the ordering of the binary data of the picture. There are two common ways of ordering binary picture data. The first method packs the leftmost pixel of a set of eight pixels of the picture into the low order bit position of a byte. The second method, in contrast, packs the leftmost bit of the set of eight

pixels into the high order bit postition of a byte. Both of these methods order the lines from top to bottom and pack successive groups of eight pixels from left to right in successive bytes. Which method is appropriate is dictated in large part by the hardware being used. For present purposes, it is assumed that the second method is employed.

The "shrink2X" procedure operates to convert the source picture into the destination stamp by examining four lines at a time. These lines are those designated by "abov", "topp", "bott" and "next". Specifically, this procedure retains two successive bytes from each of the lines pointed to by these pointers. The current byte pair taken from the "abov" line is held in the variable "wa". Similarly, the current byte pairs taken from the lines pointed to by the pointers "topp" and "bott" are held in the variables "wt" and "wb", respectively. Lastly, the current byte pair taken from the line pointed to by "next" is held in the variable "wn".

As the code in the Appendix reveals, the "shrink2X" procedure processes this data from four respective lines to determine bytes from which the output bits will be selected. These bytes are denoted as b0 through b7 in the code listing in the Appendix. One bit is selected from each of these bytes (i.e., b0 through b7). Which bit is selected is dependent upon the X-phase of the output byte . The composition produces one output byte. This process continues along the four lines and along each successive collection of lines until all of the output bytes have been produced.

To get a better feel for how "shrink2X" operates, it is helpful to look at the code contained within the Appendix. As the code indicates, the variables "wa", "wt", "wb" and "wn" initially are assigned to the first two bytes in the lines pointed to by "abov", "topp", "bott" and "next", respectively. These words of data from the four lines are processed within the "for loop" that has i as an index. This "for loop" continues processing data until i=n. The variable "n" is a parameter passed into the "shrink2X" procedure indicating the number of output bytes to be produced. Since each iteration through the loop produces one output byte, i has a

value equal to n when n output bytes have been produced.

Within the loop, the bytes b0, b1 and b2 are fetched initially. From these bytes, the three left-most bits of the output byte will be selected. After these bytes have been fetched, the variables "wa", "wt", "wb" and "wn" are shifted so as to get the next four input bytes. They are shifted to the left by 8 bits or one byte; hence, the left-most byte of these variables is shifted out and a new byte is shifted into the righthand byte of the word. Once this shifting is completed, the bytes from which the next four output bytes will be selected are fetched. In particular, the bytes b3, b4, b5 and b6 are fetched.

Having selected these bytes, the system performs another shift that shifts the variables left along the lines by a byte. When this second shift is completed, the final byte, b7, is fetched. The right-most output bit is selected from this byte. It should be noted that in both of the shifts, if the end of line is reached for the current source lines, the pointers are each incremented by two lines so that new lines of the source picture are processed. If the end of the line is not currently reached, then standard shifting occurs as described above.

Having selected all the bytes from which the bits will be selected, the system composes an output byte wherein it chooses one bit from each of the selected bytes. The selection is realized by ANDing each of the bytes with a mask that selects a particular bit position.

Each of the fetched bytes, b0 through b7, is comprised of a dither result. The appropriate pixel within this dither result is selected by the composition step. In order to understand how the dither result byte is selected, it is necessary to look into the assignment statements that assign values to b0 through b7. The general pattern of these assignment statments is for that the fetched byte to equal the value pointed to by the pointer "p" plus a look-up table value. The pointer "p" points to one of the four tables within the "ansbyt" table. The look-up table "lut" is accessed to produce a grey scale value corresponding to the windowed sample pattern indicated by the index, thus "p" plus the look-up table value designates a particular entry within one of the four "ansbyt" tables. The

logical statement that serves as the index for the look-up table refers to a particular windowed sample. How these logical statments compose a windowed sample is perhaps best illustrated by an example.

The statement that fetches a value for the byte b4 is a good illustration of how the logical statement that serves as the index to the look-up table composes a windowed sample pattern. For illustrative purposes, suppose that the words from the four lines have been gathered such as shown in Figure 12. In this instance, the system seeks to determine the grey scale value for the windowed sample 3 having bits in the positions such as set out in Figure 1. As the code indicates, the variable "wa" is shifted right by six bits and ANDed with the mask "am". To understand the effect of this shifting and ANDing, it is useful to refer to Figure 13a. The figure indicates the initial value of "wa". When "wa" is shifted to the right by six bits, the bits that "wa" contributes to the windowed sample 3 are also shifted right six bits so that they are in the right-most positions of the word. This shifted version of "wa" is then ANDed with the mask "am" that selects only these two right-most bits. As a result, the bit- shifted and masked version of "wa" is comprised of all zeroes except for the bits "a" and "b" which are located in the right-most positions as indicated in Figure 13a.

The statement in the code that fetches a value for b4 also indicates that the shifted and masked version of "wa" is ORed with the shifted and masked version of "wt". "wt" initially has the value indicated in Figure 13b. It is shifted right by three bits to shift the respective position of the bits that lie within the windowed sample 3. The bit shifted version is than masked with the mask "tm". The mask selects only the bits within the windowed sample 3 and does not select any of the remaining bits. As a result, the shifted and masked version of "wt" is comprised of all zeros other than the bits that lie within the windowed sample 3.

The remaining shifted and masked words selected from the four lines are ORed with those previously described to obtain the complete windowed sample. The shifting and masking of "wb" are depicted in Figure 13c and the shifting and masking of "wn" are depicted in Figure 13d. The result of ORing all of these

shifted and masked words is depicted in Figure 13e. The effect is that each of the bits from the windowed sample is composed into a single word of two bytes in length. The order in which the bits are positioned within this word corresponds with the order that bits are read out of the windowed sample as outlined in Figure 1. Hence, it is apparent that the shifting, masking and ORing enables the selection of a windowed sample that is used as an index to the look-up table. The look-up table uses a windowed sample as an index to look up a grey scale value associated with that windowed sample. This grey scale value is then used along with the pointer "p" to select a dither product appropriate for the grey scale value retrieved by the look-up table. It is from this dither product that the output bit is selected, depending upon the X phase.

To illustrate the movement of the window that defines the windowed sample, a sample dither pattern for a 4x11 region of grey scale level two pixels is shown in Figure 14a. Note that it is presumed that "0" neighbors are positioned all around the outside of the picture including above the top row and to the left of the left column. "shrink2X" initially positions the sample window (noted in phantom form) in position 40 as shown in Figure 14b. It then moves the sample window over by 2 bits into sample position 42 shown in Figure 14c. These two sample window positions 40 and 42 result in the windowed samples 41 and 43 shown in Figure 15a and 15b, respectively. Given these sample patterns, the "shrink2X" procedure accesses the "lut" look-up table to determine the grey scale value for these samples. The first windowed sample 41 has a hexadecimal value of "010". It is interpreted as having a grey scale value of 2.

Already knowing the grey scale value, the procedure looks inside the second table "ansbyt" to fetch a row of a dither pattern that is appropriate for that grey scale value of 2 and a given Y phase. For illustrative purposes, it is assumed that the Y phase is on the first row of the dither pattern. Each of the 4 consecutive tables in "ansbyt" holds a consecutive row of the dither pattern shown in Figure 15c. Thus, the entry in the first table for an indexing grey scale value of two, holds a row of the dither pattern having a hexadecimal value of "88" (the first

part 38a of the dither pattern defining one digit of the hexadecimal value, and the second part 38b of the dither pattern defining the second digit of the hexadecimal value). The second and forth tables each hold a respective row of dither pattern having a hexadecimal value of "00" for a grey scale index of two. And the third table holds a row of dither pattern having a hexidecimal value of "22" for a grey scale index of two.

In the current example the "shrink2X" procedure would assign b0 a value equal to the first row of the dither pattern. Moreover, the windowed sample 43 ("804" in hexadecimal) is deemed to have grey scale value of two. Thus, b1 would also be assigned the first row of the level 4 dither pattern shown in Figure 15c (given the above assumed Y phase of the output). As mentioned above, this process continues until eight such bytes of dither patterns have been fetched. These bytes are labelled as b0-b7.

Once the "shrink2X" procedure has gathered all of the 8 bytes, b0 through b7, it composes the output byte 241 (Figure 17). In particular, the bit 243' (i.e. the highest order bit) is selected to have the value of bit 243 from b0, the second bit 245' is selected to have the value of bit 245 from b1, the third bit 247' is selected to have the value of bit 247 from b2, and so on until the output byte 241 is entirely composed. In the example case, the output for the sample window 41 shown in Figure 15a, assuming a Y phase equal to 1, is the bit at position 45 of the dither pattern in Figure 15c. This pixel is a black pixel. Similarly, the output for the second windowed sample pattern 43 (Figure 15b) is the bit at position 47 (Figure 15c) which is a white pixel. This composition of the output byte is listed in the Appendix at the statement that assigns a value to the byte to which "dest" points. The "++" indicates that the destination pointer is then incremented.

The stamp making process thus may be summarized in a number of steps. These steps are set out in a flowchart in Figure 18. First, the variables and tables are initialized (step 259). Then a sample of 12 pixels is obtained (step 260). The grey scale value associated with the sample pattern is found by looking up the grey scale value in the look up table (step 262). Line art is therefore treated as a

grey scale value (i.e. either 0 or 16). Next, an 8 bit wide strip of the dither pattern associated with the grey scale value of the sample is selected (step 264). The 8 bit wide strip is chosen based on the Y phase of the ouput. The output pixel is selected from this 8 bit wide strip based on the X phase as determined by X mod 8 (step 266). The system checks to see if it is finished (step 267). If not, it repeats the steps 260-267 for a next windowed sample.

The stamp making method has been described with reference to a software implementation. Hardware implementations are, nonetheless, possible. One straightforward hardware approach is to hold four input lines streaming by a set of gates. The gates pick off successive patterns of bits in the sample window. The gates, then determine from the selected sample an appropriate output destination line on a pixel by pixel basis.

The stamp-making approach described above for generating miniaturized stamps may be generalized to produce enlargments as well as reductions. This generalized approach may be used to "re-size" the source image into a different size destination image having potentially different X and Y dimensions. In accordance with the generated approach, a source grid and a destination grid are defined. The two grids are then superimposed so that all of the source pixels are used and so that each part of a source pixel goes into one and only one destination pixel.

Figure 19 provides an illustration of the generalized approach. For picture reduction, each destination pixel such as 232 encompasses a total of more than one source pixel 230 and thus, must necessarily have contributions from more than one source pixel 230. As can be seen in Figure 19, the destination grid 222 for this case is superimposed over the source grid 220.

For each picture enlargement, each destination pixel 234 comes from a total of less than one source pixel 230. As Figure 19 illustrates, the destination grid 224 superimposes the source grid 220 such that each destination pixel 234 covers less than a single source pixel 230, but a destination pixel may nevertheless receive contributions from more than one source pixel. Lastly for a change in

aspect ratio, the destination grid 226 is such that a destination pixel 236 may receive contributions from portions of different source pixels 230.

In resizing pictures, the decision is first made for each destination pixel as to whether it is to represent the pixel as line art or as a grey scale pixel. This is done by weighted voting of the contributing source pixels. The outcome of this vote then focuses all contributing source pixels to offer this type of contribution - line art or grey scale - to a weighted average result. Finally, if the destination pixel is considered to represent line art, the final weighted average of partial 16's and partial 0's is forced to 0 or 16 depending on whether it exceeds some threshold such as 8.

While the invention has been particularly shown and described with reference to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made without departing from the spirit and scope of the invention as defined in appended claims. For example the shrinking of a picture into stamps need not follow the particular software described with reference to the preferred embodiment. Many alternative software approaches will be readily recognizable by those skilled in the art. Moreover, dithering arrays having sizes other than 4x4 may be used. A primary example of an alternative size is a 8x8 array. The 8x8, however, can be viewed as being comprised of four 4x4 grids. These grids will realize the same small dither samples as are produced by a single 4x4 grid.

Another alternative embodiment concerns the sample window. In particular, other sample window configurations are possible. It should be noted though that if such alternative window configurations span more than 1 dither array pattern in any direction, then, non-recognition of dithering errors are likely to occur. The errors of this type are likely to occur because in order to have a match with a dither pattern exact conformance with dithering over a larger area is necessary but unlikely at transitions of even a single grey level. Furthermore, the size of "lut" has to be changed to accomodate the different window size.

An additional alternative concerns the choice of dithering algorithm. The

preferred embodiment has been described with reference to dispersed dot dithering. Other varieties of dithering algorithms may be used. Nevertheless, dispersed dot appears to be the optimal kind of dithering because the patterns it produces are very unlikely to arise in letter fonts, lines or other black and white art work.

An additional alternative concerns compression of image data for advantages in data handling. Typical image compression schemes use one- or two-dimensional coherence as the basis for establishing an expectation of what's coming next and therefore, on the average, the compressed image usually consists of fewer bytes than explicit pixel-by-pixel expanded image. With dithered images or images containing dithered parts, there is a lot of switching between black and white (1's and 0's) and therefore a special compression/expansion scheme is needed -- one which takes advantage of the likelihood that what's happening at a point is the same thing that happened, not at adjacent pixels, but rather what happened a dither-pattern width to the left and/or a dither-pattern height above. It should also serve well for line art, and should be simple and fast, both for encoding and decoding.

Such an encoding scheme is here presented for the case of a 4x4 dither pattern. It could obviously be altered to suit other sizes/shapes of dither patterns. To begin, lines are taken 4 at a time, from top to bottom of the total image. Each four line-high horizontal stripe is broken into large chunks 208 (Figure 20a), four bytes (32 pixels) wide. Each of these 4 lines high by 32 pixels wide chunks 208 is here called a '4x32', and in turn is broken into four smaller 4x8 chunks as shown in Figure 20a.

Each 4x32 is encoded first of all with a one-byte header consisting of 4 dibits, one for each 4x8 pixel chunk (i.e. 4-lines high by one-byte-wide):

0 0 the 4x8 is solid background;

0 1 the 4x8 exactly matches the neighboring 4x8 above; default background is assumed "above" the top of the image;

1 0 the 4x8 exactly matches the 4x8 neighbor to the left;

1 1 other (the 4x8 is decomposed).

Note that the "background" may be started as either 0 or 1. It can be arranged to switch automatically if or when the solid non-bakcground count of 4x8's in the recent past exceeds the solid background count (encoder and decoder must, of course, follow the same switching rules). And identity with left neighbor byte or left neighbor 4x8 can't be used on the left edge, not because of low probability wrap-around spatial coherence, but because, on decoding, the decoded left context isn't even there yet.

The header of each 4x32 is followed by respective decompositions (if any), each of which contains (1) a subheader of four dibits, one for each byte:

0 0 the byte is solid background (as described above)

0 1 the byte is identical to the one above it

1 0 the byte is identical to the one to its left (with the exception noted above)

1 1 other: the byte is as follows (i.e. an "explicit" byte)

and (2) the explicit bytes (if any).

Figure 20b is illustrative of the foregoing. As an example, Figure 20b shows four bytes (1) thru (4) from the compressed file top to bottom. The first byte (1) indicates two dibits each having a value "00". Thus, two 4x8's (four lines high by 8 pixels wide) have solid background, i.e. either 0 or 1 as explained above. One dibit with the value "11" indicates one decomposed 4x8. And the last dibit "01" of byte (1) indicates one 4x8 like that directly above it. Byte (2) provides one dibit value "00". This dibit indicates that one 4x8 has a solid background. The following 2 dibits 501, 503 each of value "11" indicate two 4x8's of explicit image data. And the last dibit "10" of byte (2) indicates one 4x8 like the left neighbor byte. Accordingly, byte (3) holds the exact bit pattern of the first "explicit" byte

2100064

501 in byte (2). And byte (4) indicates the exact bit pattern of the other explicit byte 503 in byte (2).

It is understood that other similiar compression/decompression schemes may be employed by the present invention.

The foregoing alternative embodiments are intended to be encompassed within the present invention.

CLAIMS

1. A method of converting a dithered representation of an image to a discretely quantized continuous tone representation of the image, the dithered representation including a plurality of binary data bits, each binary data bit representing a state of a pixel of the image, said method comprising the steps of:

a) providing a viewing window having a defined size and shape;

b) using the viewing window, establishing windowed regions of the dithered representation of the image, each windowed region defining an ordered pattern of pixels in the image;

c) for each windowed region, comparing the ordered pattern of pixels defined by the windowed region with a plurality of predetermined ordered patterns of pixels that identify a plurality of possible continuous tone representations from which the pattern in the windowed region could have originated by dithering, said comparing of ordered patterns of pixels being performed on a pixel by pixel basis such that, for each windowed region, the comparing step provides at least one corresponding continuous tone representation; and

SUBSTITUTE SHEET
12.02.93

d) assigning corresponding continuous tone representation for respective windowed regions of the dithered representation of the image to create a discretely quantized continuous tone representation of the image, wherein, for each windowed region, the assigned continuous tone representation is selected from the corresponding continuous tone representations provided by the comparing step.

2. A method as recited in Claim 1 wherein the comparing and assigning steps are performed by the data processing system.

3. A method as recited in Claim 1 wherein the predetermined patters of pixels are dispersed dot patterns.

4. A method as recited in Claim 1 wherein the step of assigning continuous tone representations proceeds a pixel at a time within a region.

SUBSTITUTE SHEET

5. A method as recited in Claim 1 wherein the original image comprises dithered regions and line art, and the ordered pattern of pixels defined by the windowed regions is first compared with a plurality of predetermined dither patterns of pixels to designate regions of the original image as dithered, and said continuous tone representations are assigned only for said designated dithered regions.

6. A method as recited in Claim 5 wherein the predetermined dither patterns of pixels are those which result from dispersed dot dithering.

7. A method as recited in Claim 5 wherein the step of comparing includes providing a table of predetermined dither patterns, including at least one pattern for line art such that said step of assigning a continuous representation of the image produces desired line art corresponding to line art in the initial representation of the image.

8. A method as recited in Claim 5 wherein the process applies to a given pixel, further including, prior to the step of comparing, determining the position of the given pixel within said window.

9. A method for designating whether a portion of an image is dithered, comprising the steps of:

a) encoding pixels in the portion of the image as a string of bytes;

b) comparing the string of bytes with pixel patterns held in memory as encoded bytes, to determine if the pixels in the portion of the image match any pixel patterns held in memory; and

c) where the pixels in the portion of the image match any pixel pattern held in memory, designating that the portion of the image is dithered.

10. A method as recited in claim 9 wherein the steps are performed by a data processing system.

11. A method as recited in claim 9 wherein less than all pixel patterns possible for a given dithering scheme are held in memory.

12. A method of determining whether a region of a picture is dithered, comprising the steps of:

a) examining the region to determine whether all white pixels in the region are disconnected such that each white pixel has only black pixels as immediately adjacent neighbors;

b) examining the region to determine whether all black pixels in the region are disconnected such that each black pixel has only white pixels as immediately adjacent neighbors;

c) where one of all white pixels are disconnected and all black pixels are disconnected, determining that the region is dithered;

d) otherwise, determining that the region is not dithered.

13. In a data processing system, a method of generating a miniaturization of an image, comprising the steps of:

a) iteratively examining original regions of the image to designate if the original regions are dithered;

b) for each original region of the image designated as dithered, generating an output region that is a shrunken re-dithered representation of the original region of the image; and

c) for each remaining original region of the image generating an output region that is a shrunken non-dithered representation of the original region of the image.

2100064

1 / 17

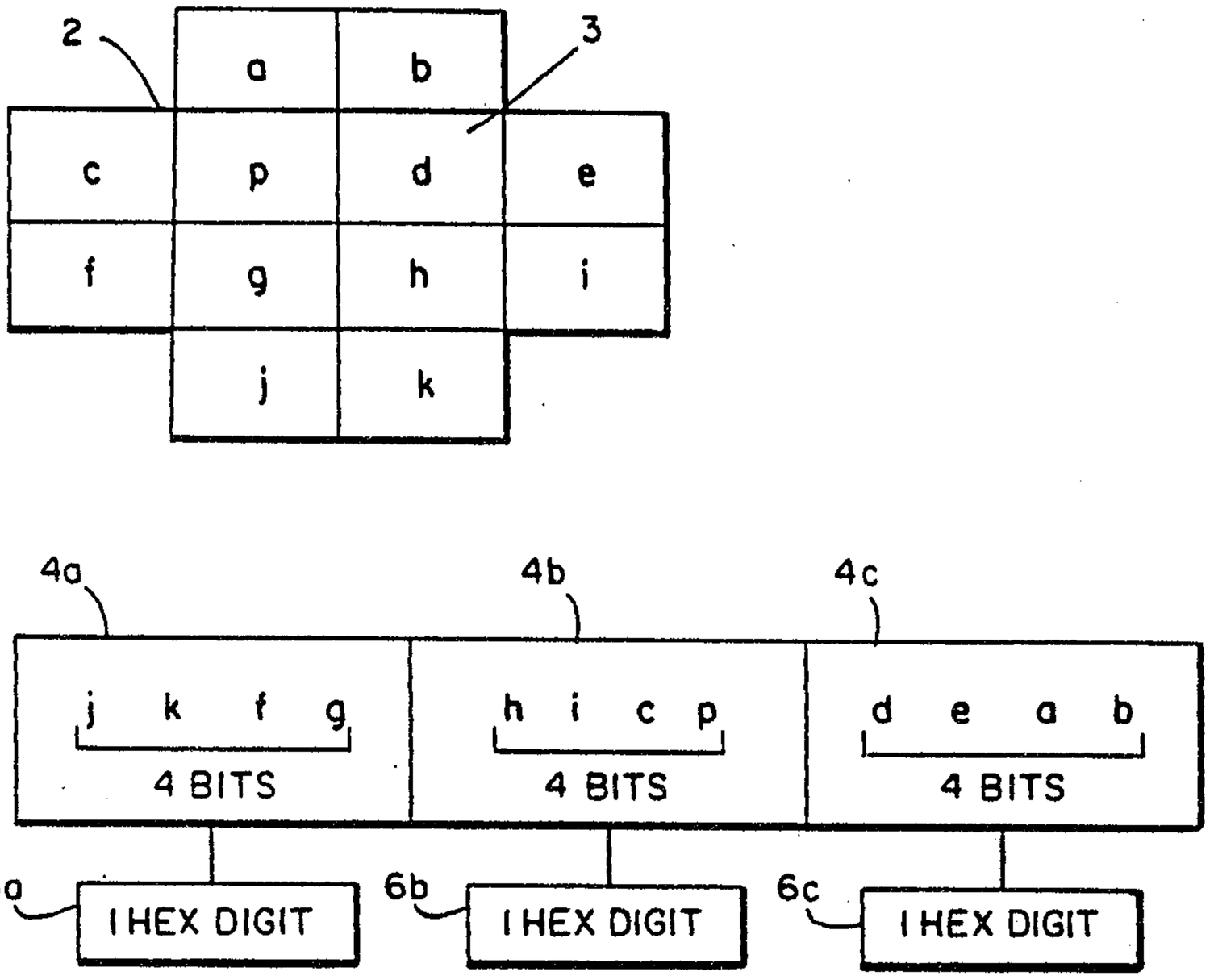


FIG. 1

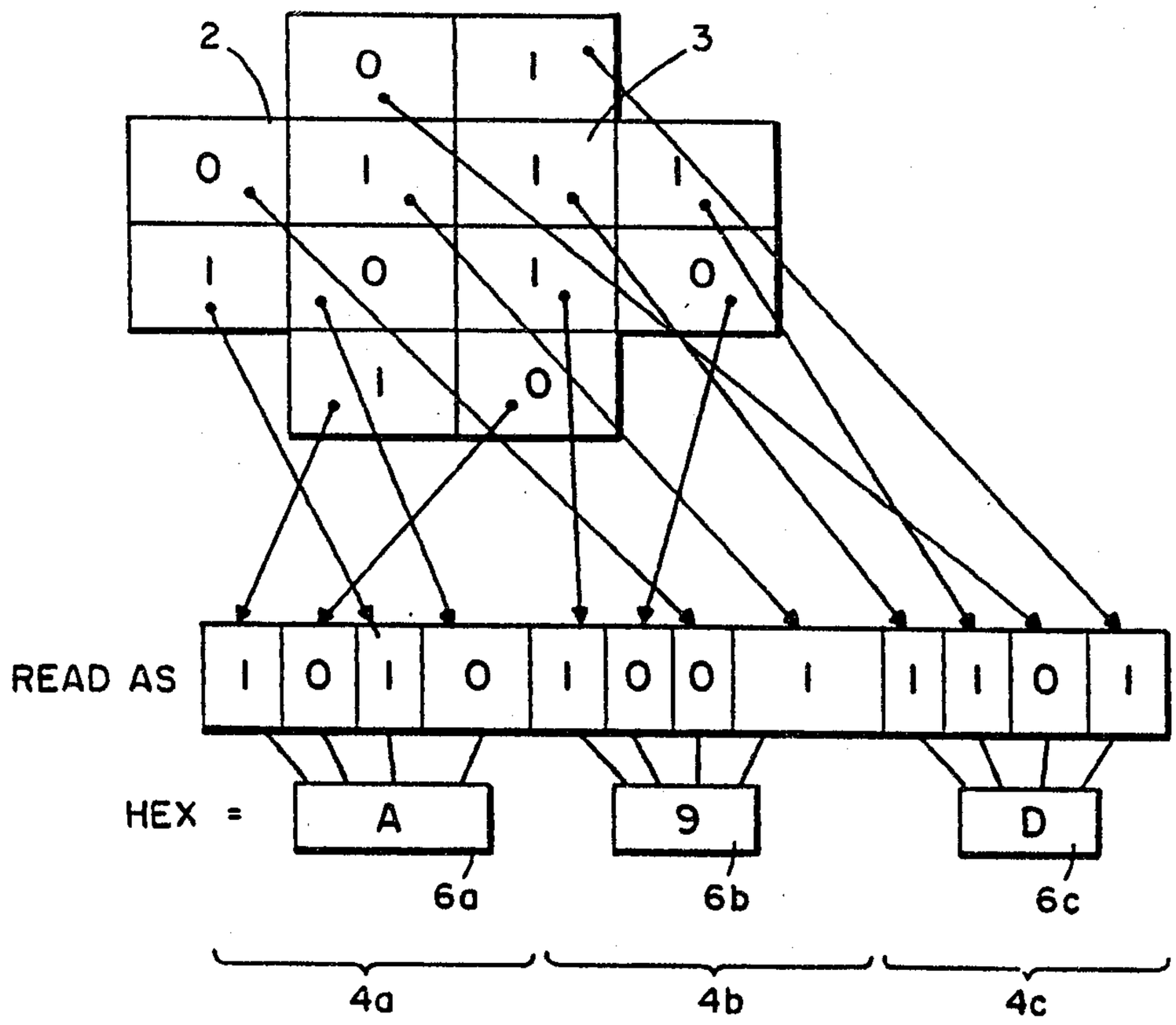


FIG. 2

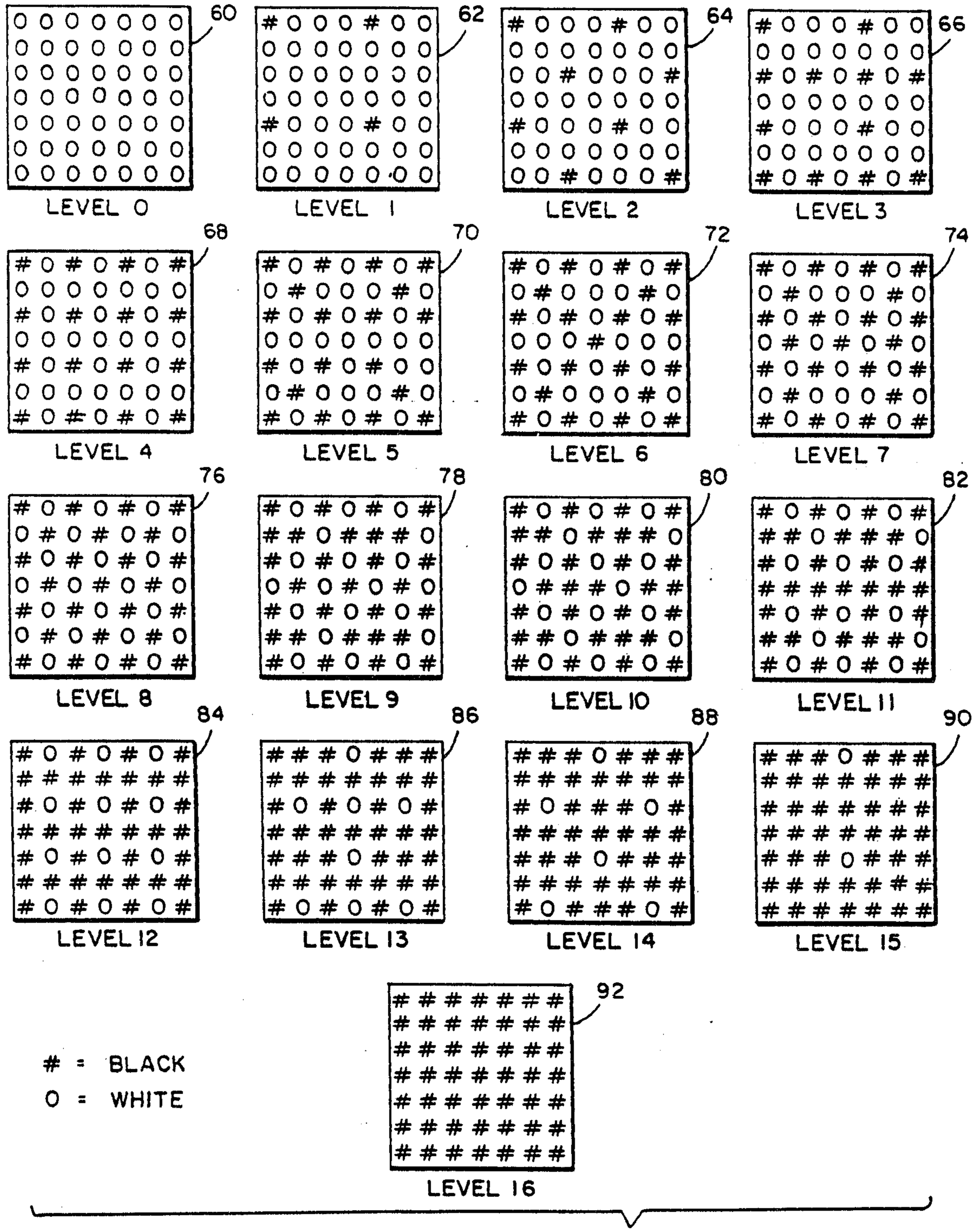

60
LEVEL 0
62
LEVEL 1
64
LEVEL 2
66
LEVEL 3
68
LEVEL 4
70
LEVEL 5
72
LEVEL 6
74
LEVEL 7
76
LEVEL 8
78
LEVEL 9
80
LEVEL 10
82
LEVEL 11
84
LEVEL 12
86
LEVEL 13
88
LEVEL 14
90
LEVEL 15
92
LEVEL 16
# = BLACK
O = WHITE

FIG. 3

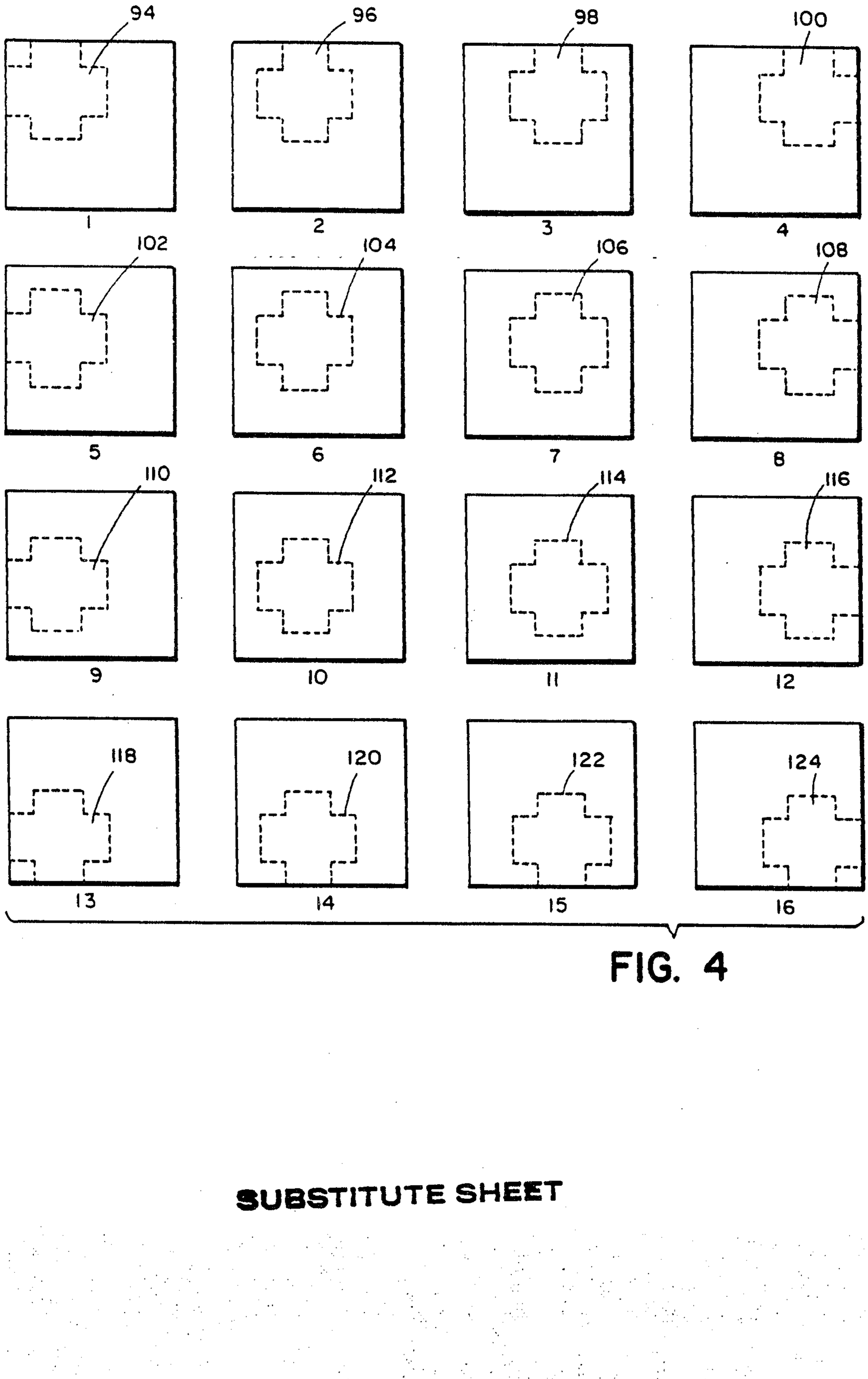

94
96
98
100
1
2
3
4
102
104
106
108
5
6
7
8
110
112
114
116
9
10
11
12
118
120
122
124
13
14
15
16

FIG. 4

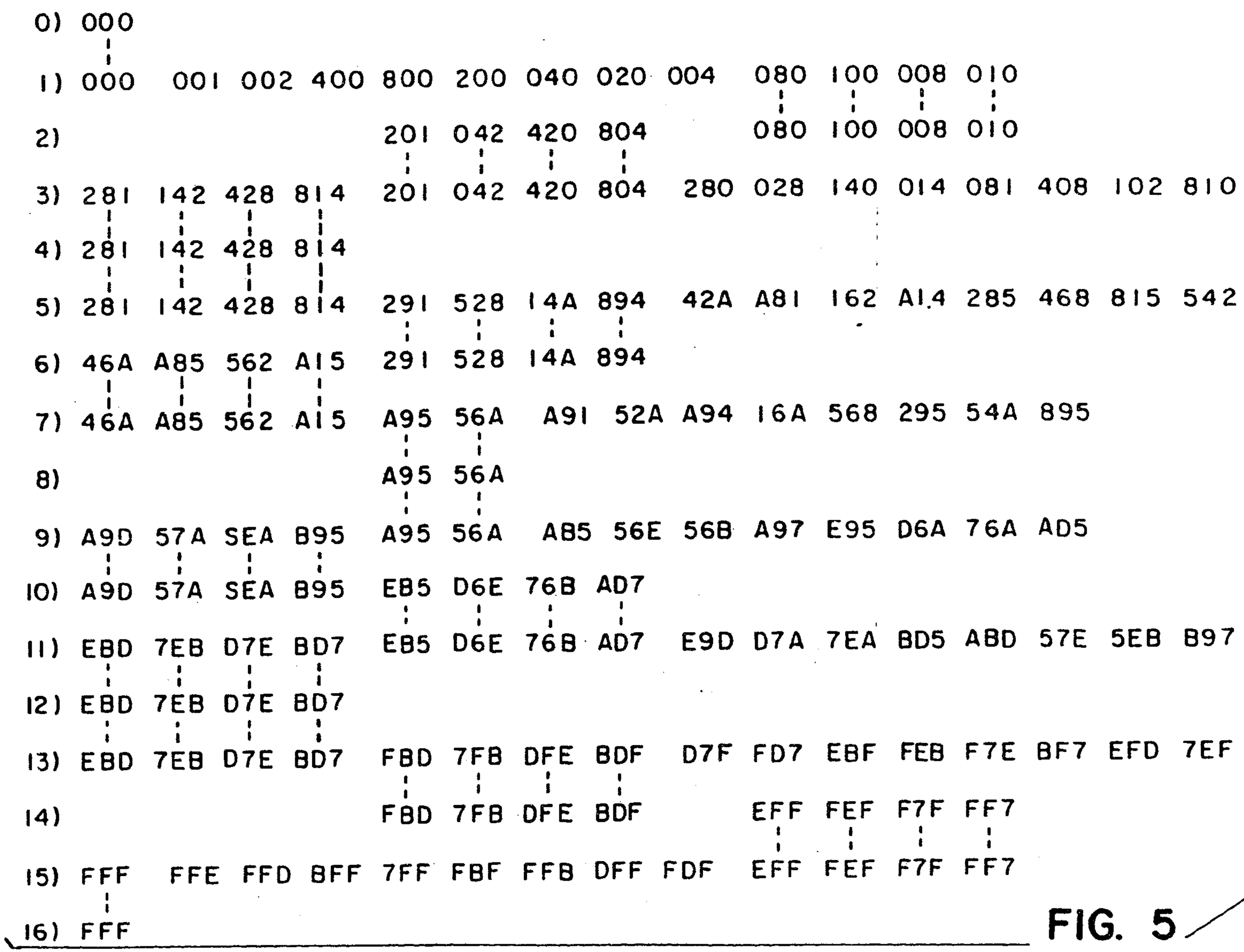

0) 000
1) 000 001 002 400 800 200 040 020 004 080 100 008 010
2) 201 042 420 804 080 100 008 010
3) 281 142 428 814 201 042 420 804 280 028 140 014 081 408 102 810
4) 281 142 428 814
5) 281 142 428 814 291 528 14A 894 42A A81 162 A14 285 468 815 542
6) 46A A85 562 A15 291 528 14A 894
7) 46A A85 562 A15 A95 56A A91 52A A94 16A 568 295 54A 895
8) A95 56A
9) A9D 57A SEA B95 A95 56A AB5 56E 56B A97 E95 D6A 76A AD5
10) A9D 57A SEA B95 EB5 D6E 76B AD7
11) EBD 7EB D7E BD7 EB5 D6E 76B AD7 E9D D7A 7EA BD5 ABD 57E 5EB B97
12) EBD 7EB D7E BD7
13) EBD 7EB D7E BD7 FBD 7FB DFE BDF D7F FD7 EBF FEB F7E BF7 EFD 7EF
14) FBD 7FB DFE BDF EFF FEF F7F FF7
15) FFF FFE FFD BFF 7FF FBF FFB DFF FDF EFF FEF F7F FF7
16) FFF

FIG. 5

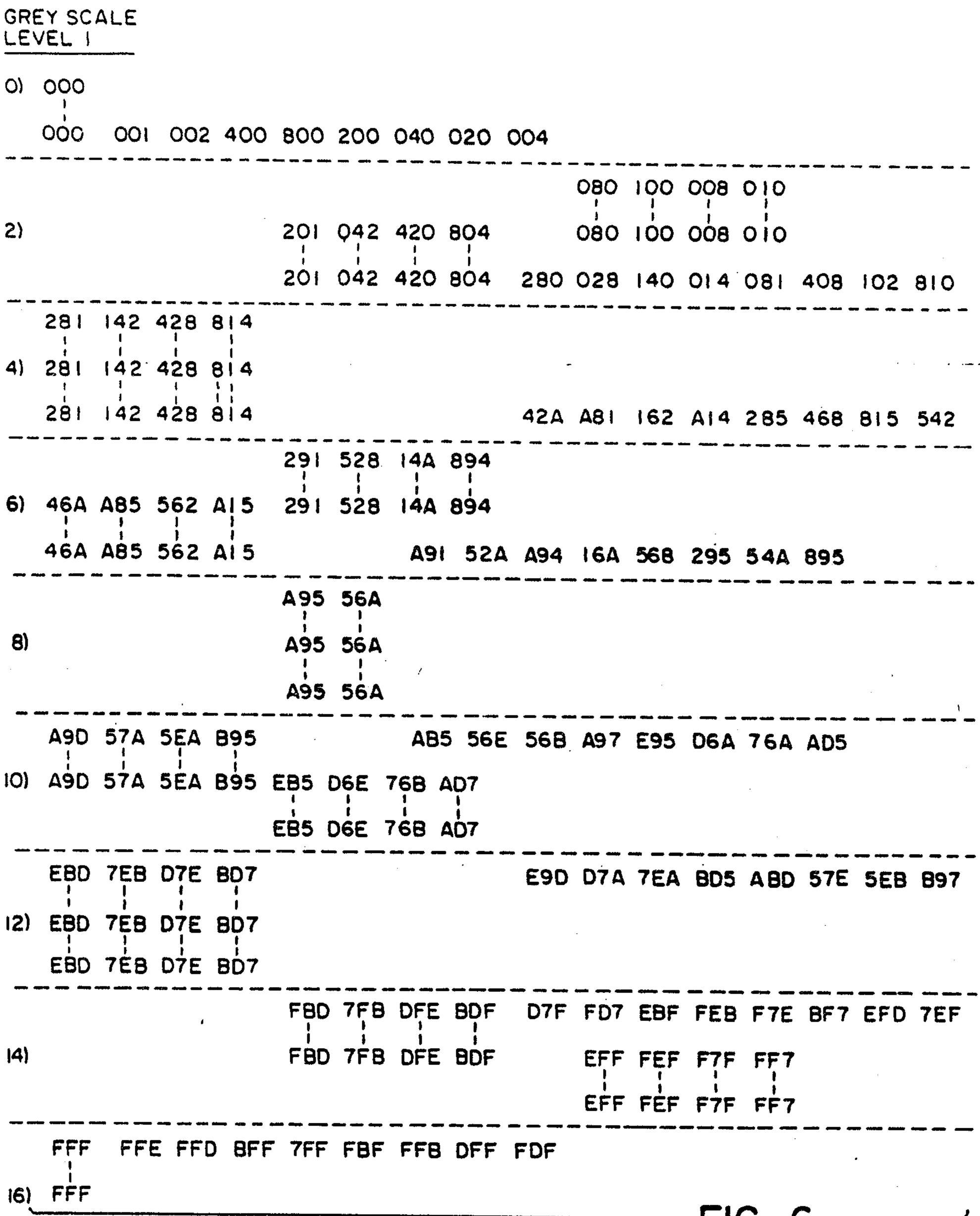


FIG. 6

SUBSTITUTE SHEET

GREY SCALE LEVEL	
01	000 001 002 400 800 200 040 020 004
2)	201 042 420 804 080 100 008 010 280 028 140 014 081 408 102 810
4)	281 142 428 814 42A A81 162 A14 285 468 815 542
6)	46A A85 562 A15 291 528 14A 894 A91 52A A94 16A 568 295 54A 895
8)	A95 56A
10	A9D 57A 5EA B95 EB5 D6E 768 AD7 AB5 56E 56B A97 E95 D6A 76A AD5
12)	EBD 7EB D7E BD7 E9D D7A 7EA BD5 ABD 57E 5EB B97
14)	FBD 7FB DFE BDF EFF FEF F7F FF7 D7F FD7 EBF FEB F7E BF7 EFD 7EF
16)	FFF FFE FFD BFF 7FF FBF FFB DFF FDF

FIG. 7

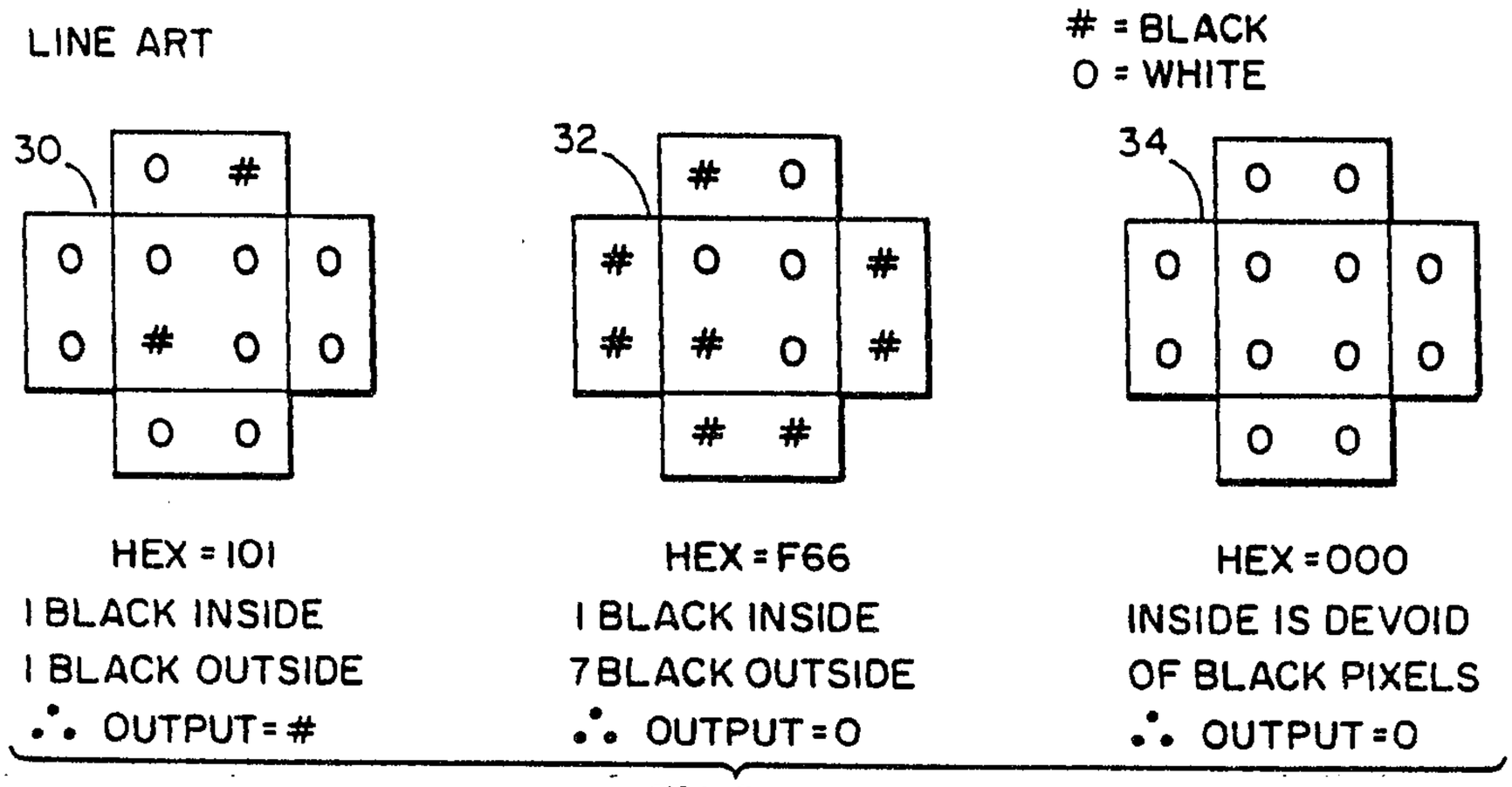


FIG. 11

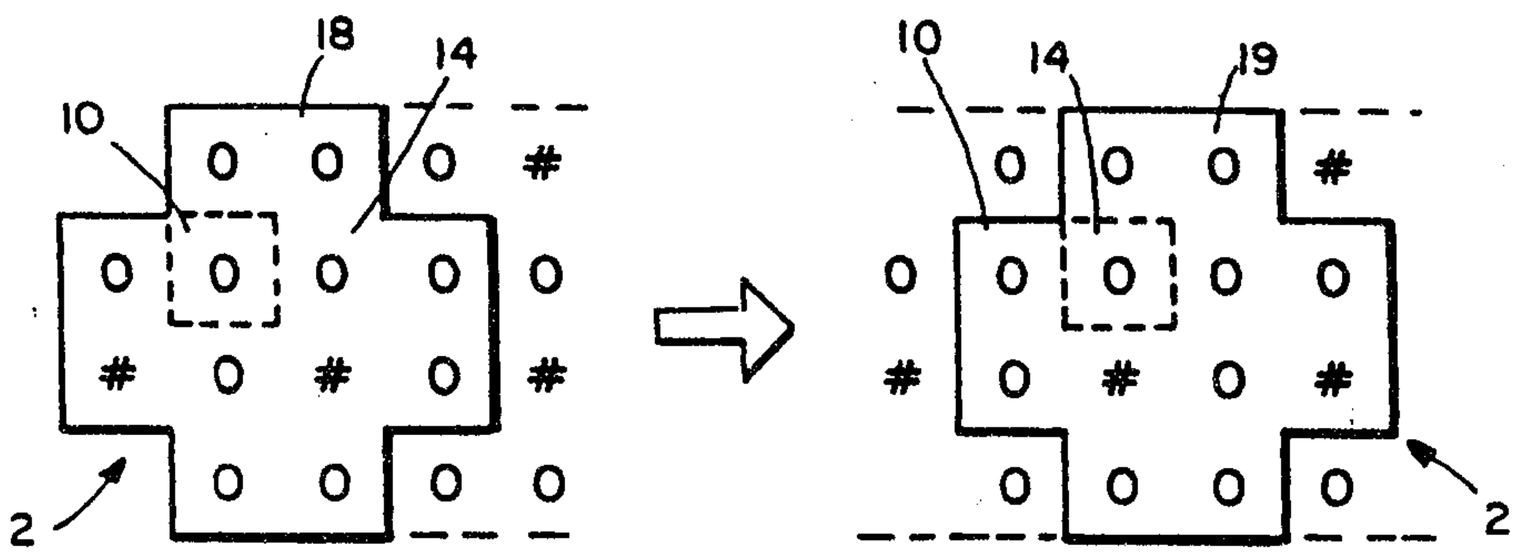


FIG. 8a

FIG. 8b

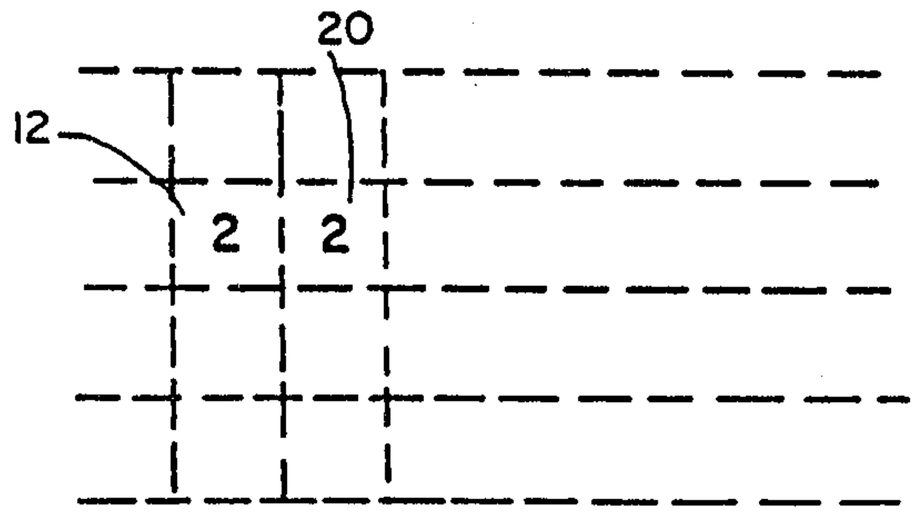


FIG. 8c

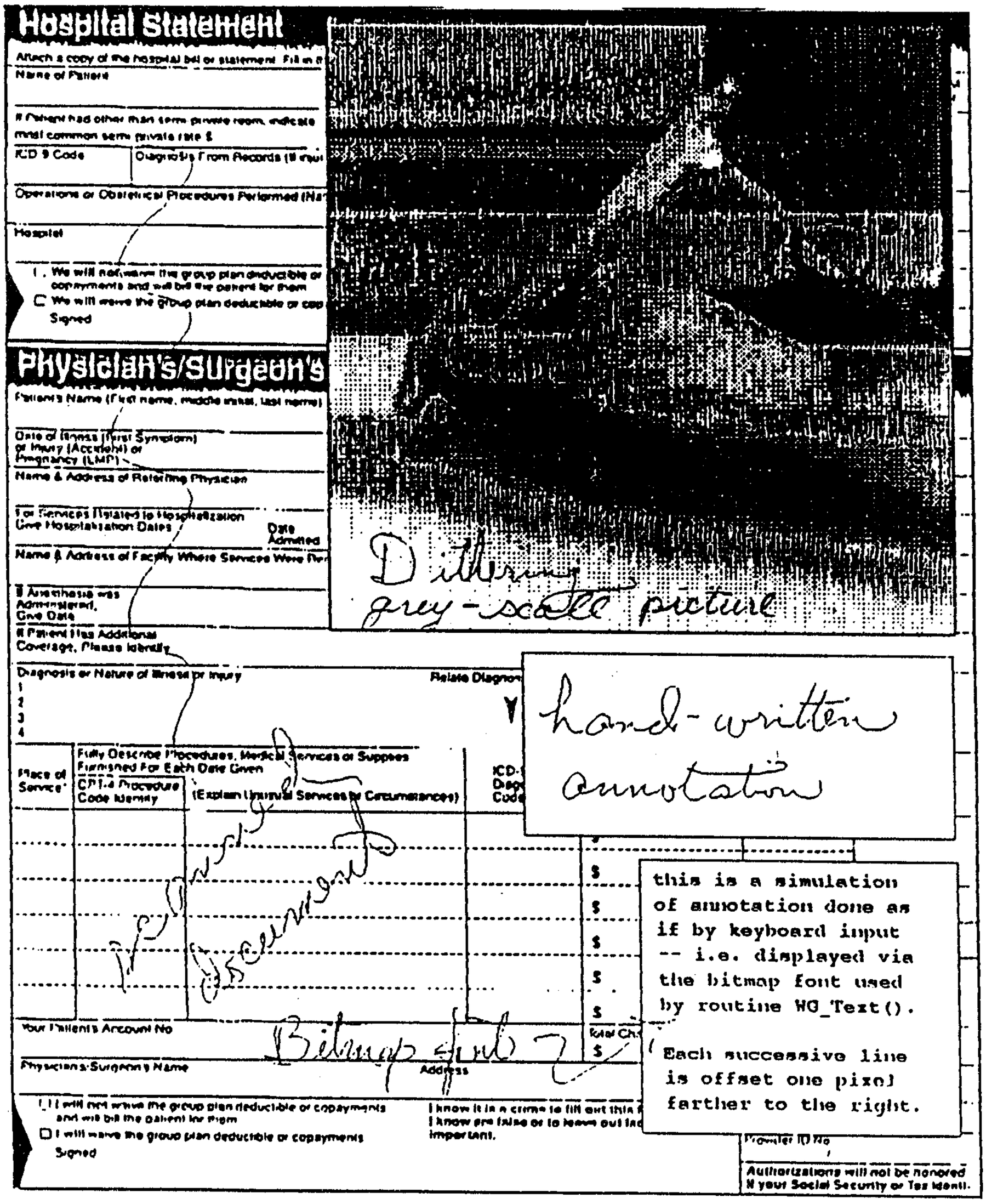

Hospital Statement
Attach a copy of the hospital bill or statement. Fill in t
Name of Patient
If Patient had other than semi-private room, indicate
most common semi-private rate $
ICD 9 Code
Diagnosis From Records (If requi
Operations or Obstetrical Procedures Performed (Na
Hospital
1. We will not waive the group plan deductible or
copayments and will bill the patient for them
C We will waive the group plan deductible or cop
Signed
Physician's/Surgeon's
Patient's Name (First name, middle initial, last name)
Date of Illness (First Symptom)
or Injury (Accident) or
Pregnancy (LMP)
Name & Address of Referring Physician
For Services Related to Hospitalization
Give Hospitalization Dates
Date
Admitted
Name & Address of Facility Where Services Were Pe
If Anesthesia was
Administered,
Give Date
If Patient Has Additional
Coverage, Please Identify
Diagnosis or Nature of Illness or Injury
1
2
3
4
Relate Diagnos
Place of
Service
Fully Describe Procedures, Medical Services or Supplies
Furnished For Each Date Given
CPT-4 Procedure
Code Number
(Explain Unusual Services or Circumstances)
ICD-
Diag
Code
$
$
$
$
$
Your Patient's Account No
Physician's/Surgeon's Name
Address
I will not waive the group plan deductible or copayments
and will bill the patient for them
I will waive the group plan deductible or copayments
Signed
I know it is a crime to fill out this
I know are false or to leave out
important.
Provider ID No
Authorizations will not be honored
If your Social Security or Tax Identi.
Digitized grey-scale picture
hand-written annotation
this is a simulation of annotation done as if by keyboard input -- i.e. displayed via the bitmap font used by routine WG_Text().
Each successive line is offset one pixel farther to the right.
Bitmap font ?

FIG. 9

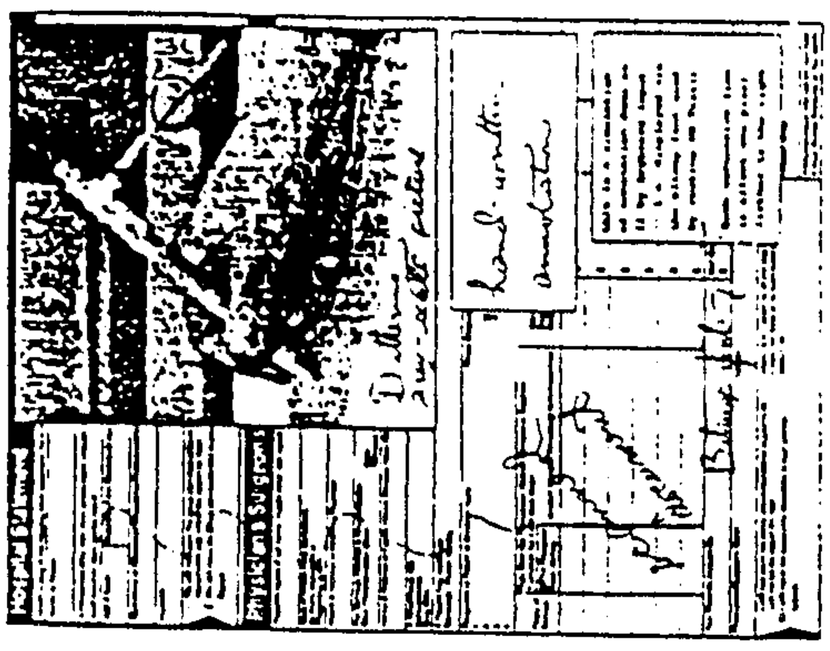


FIG. 10b

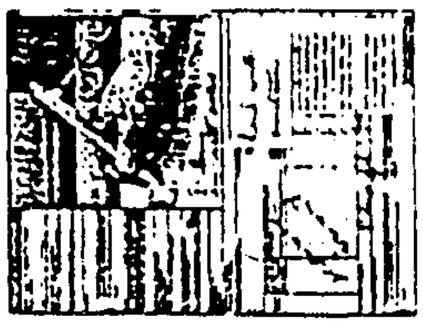


FIG. 10c

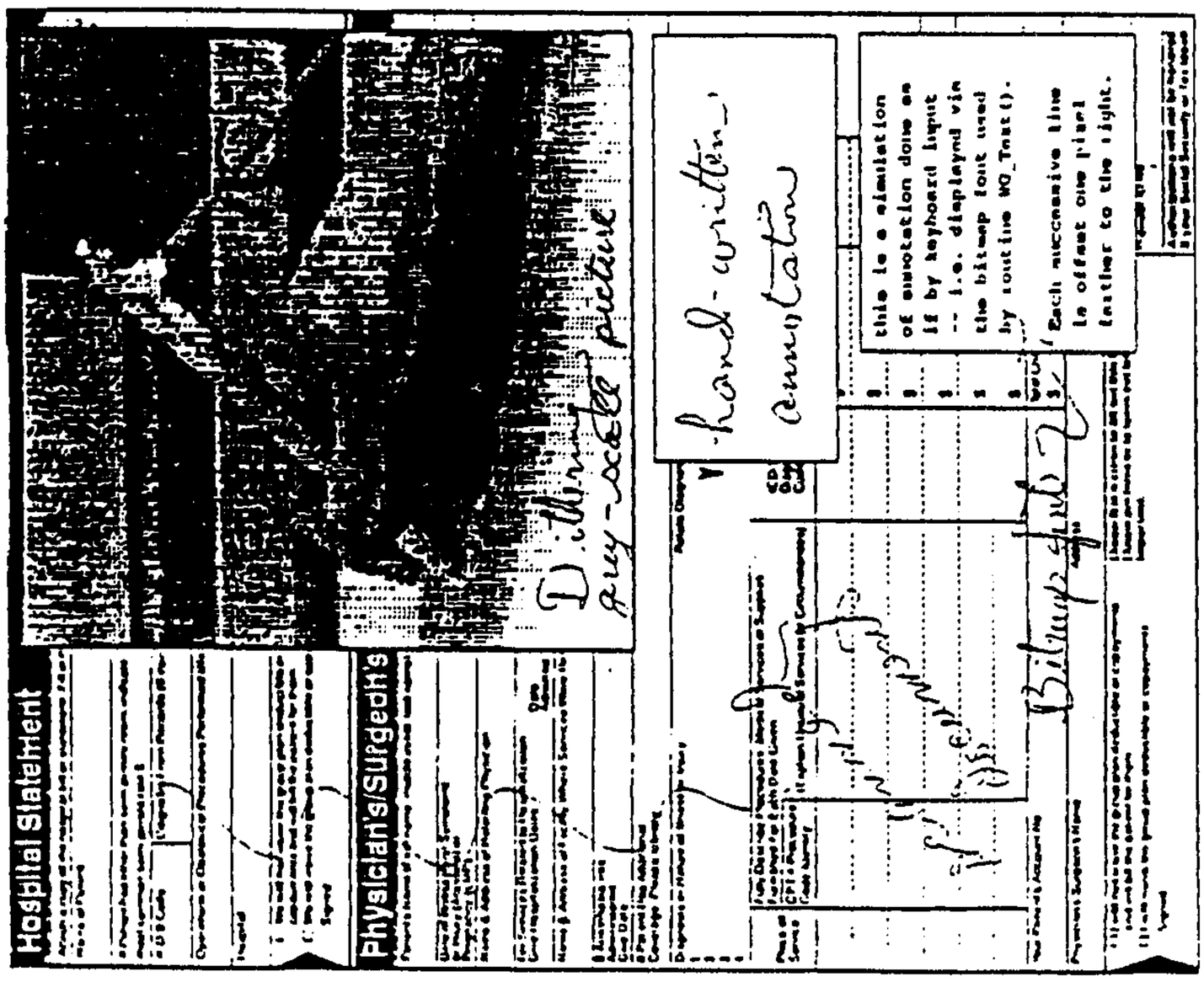


FIG. 10a

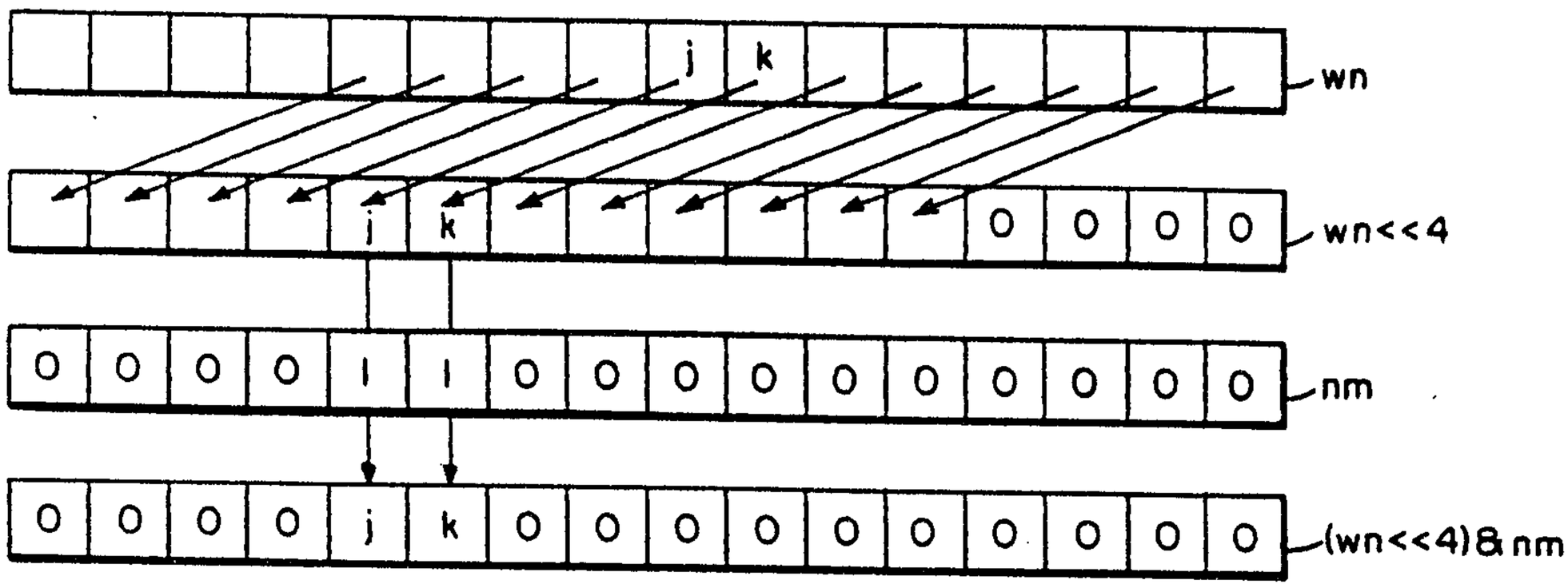


FIG. 13d

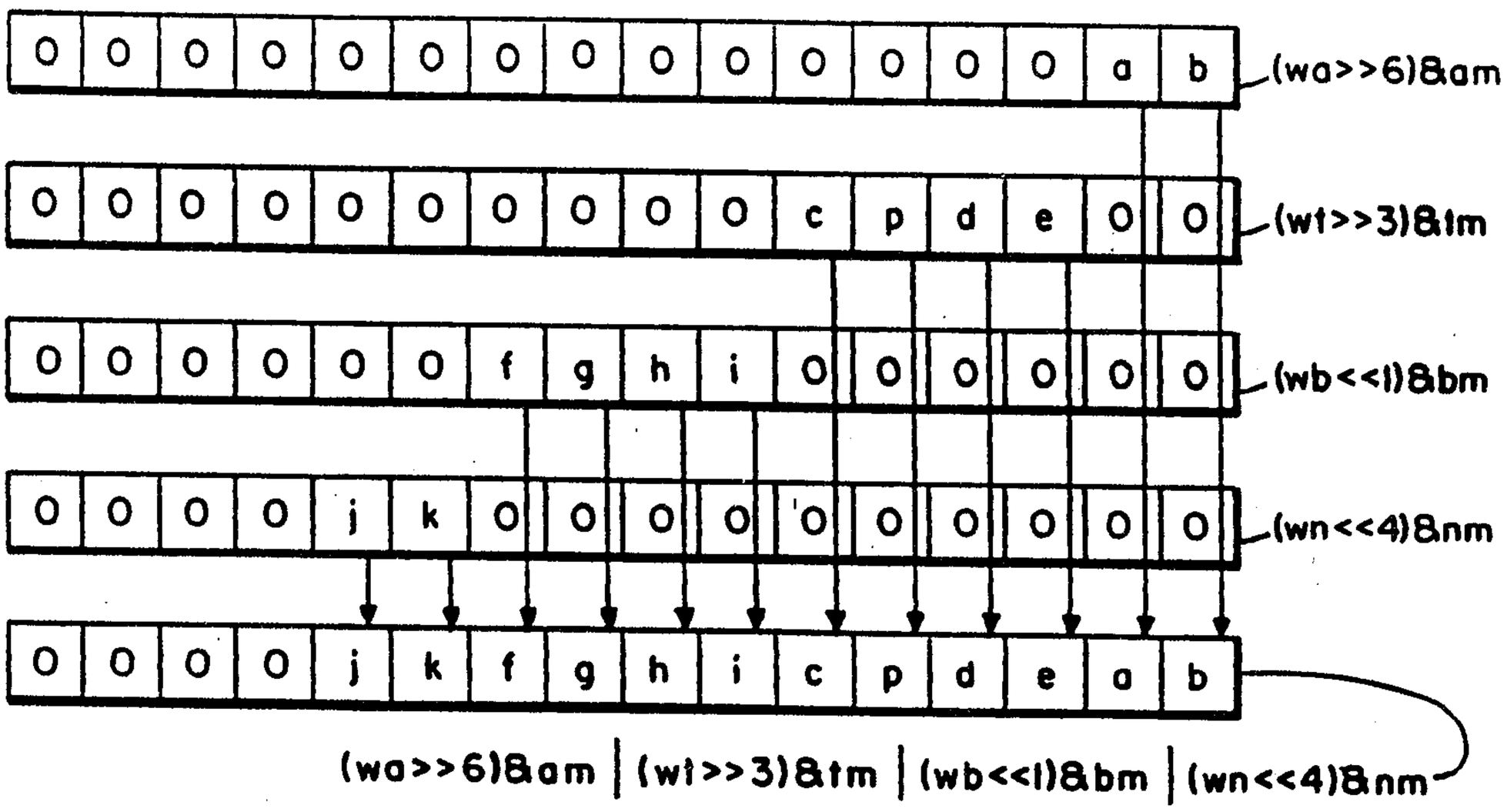


FIG. 13e

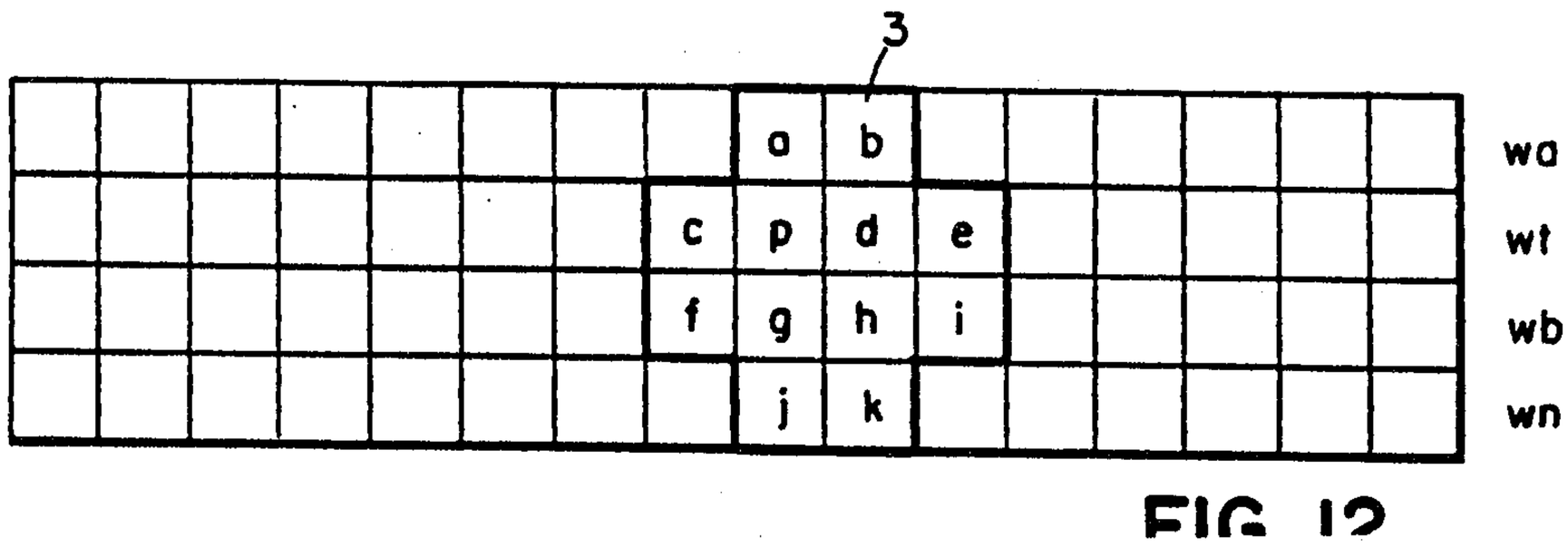


FIG. 12

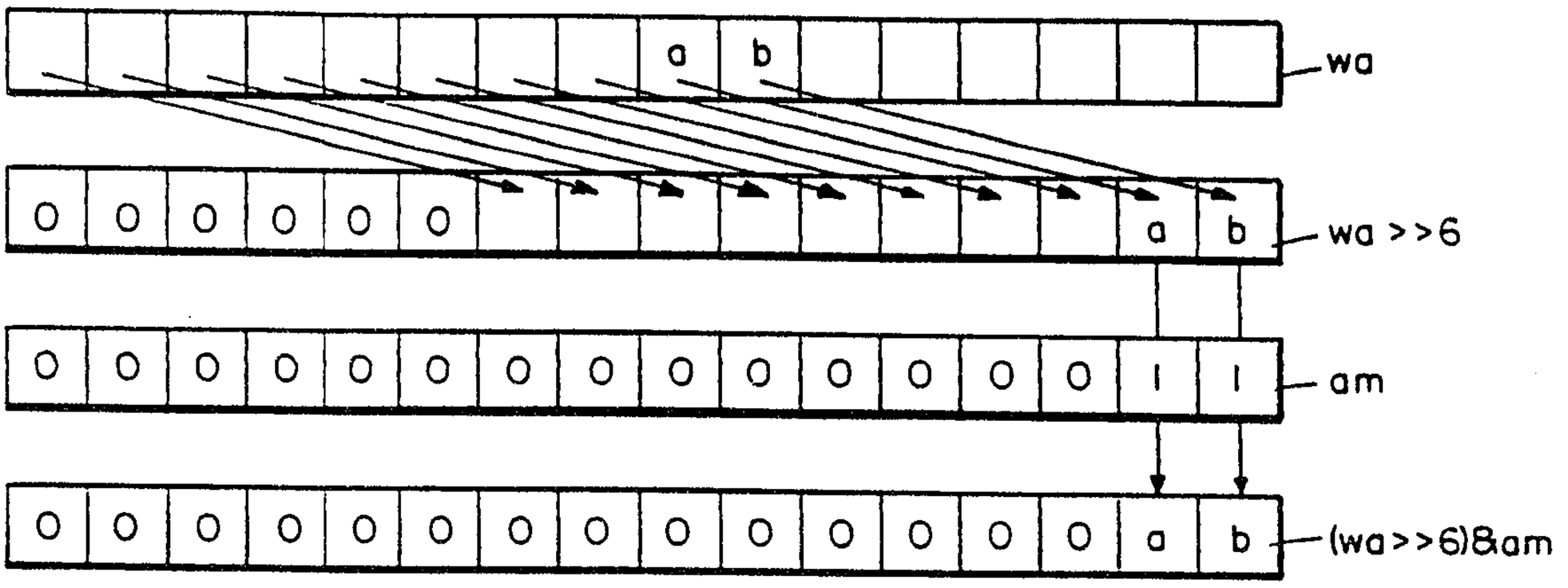


FIG. 13a

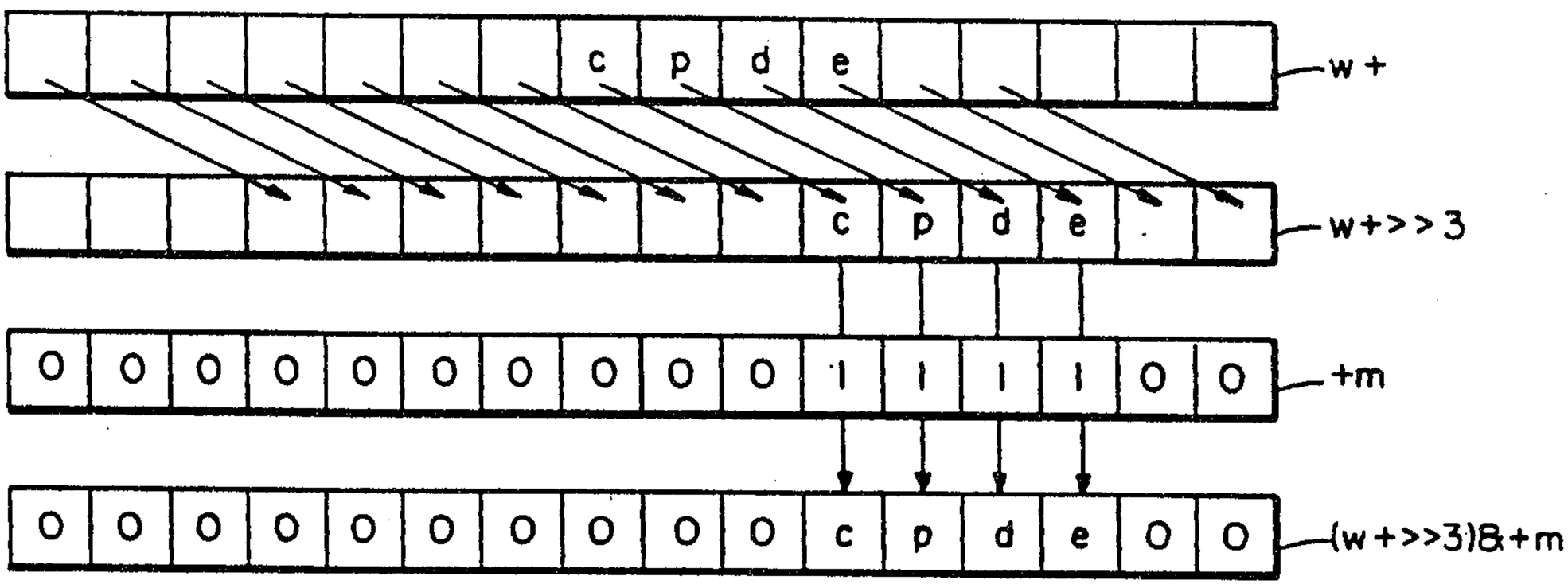


FIG. 13b

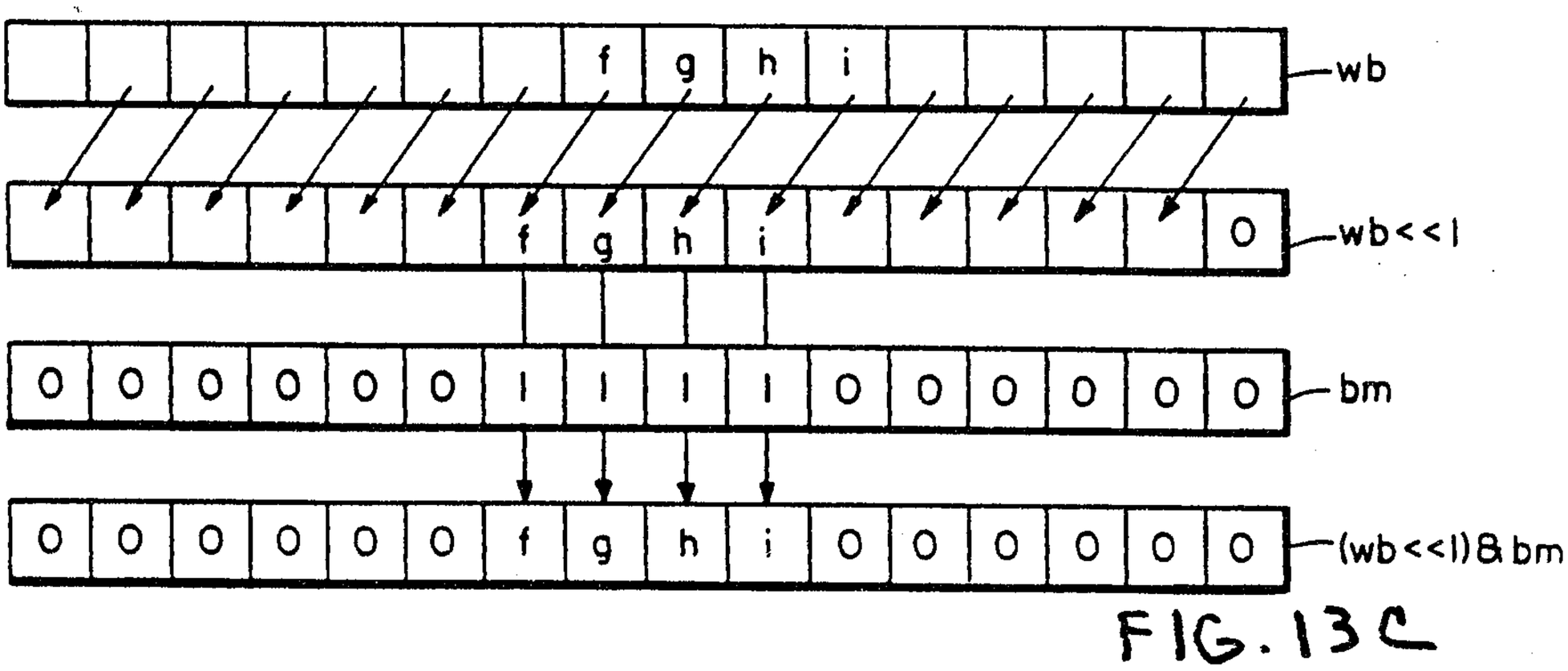


FIG. 13c

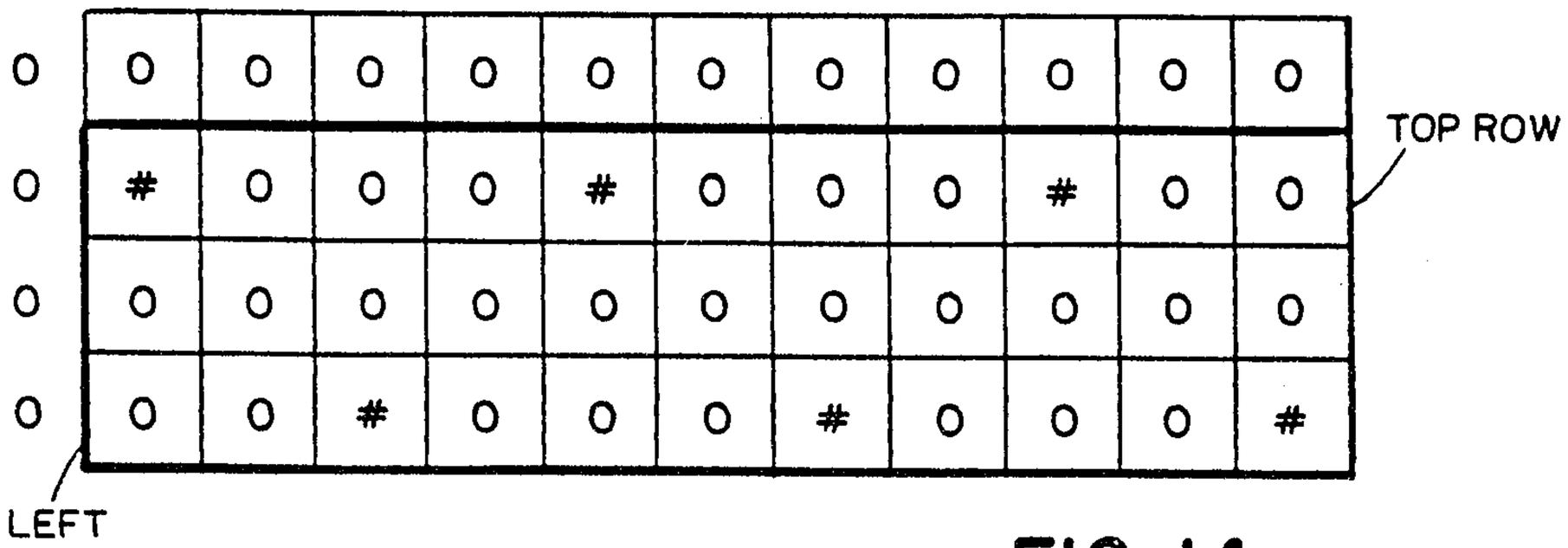


FIG. 14a

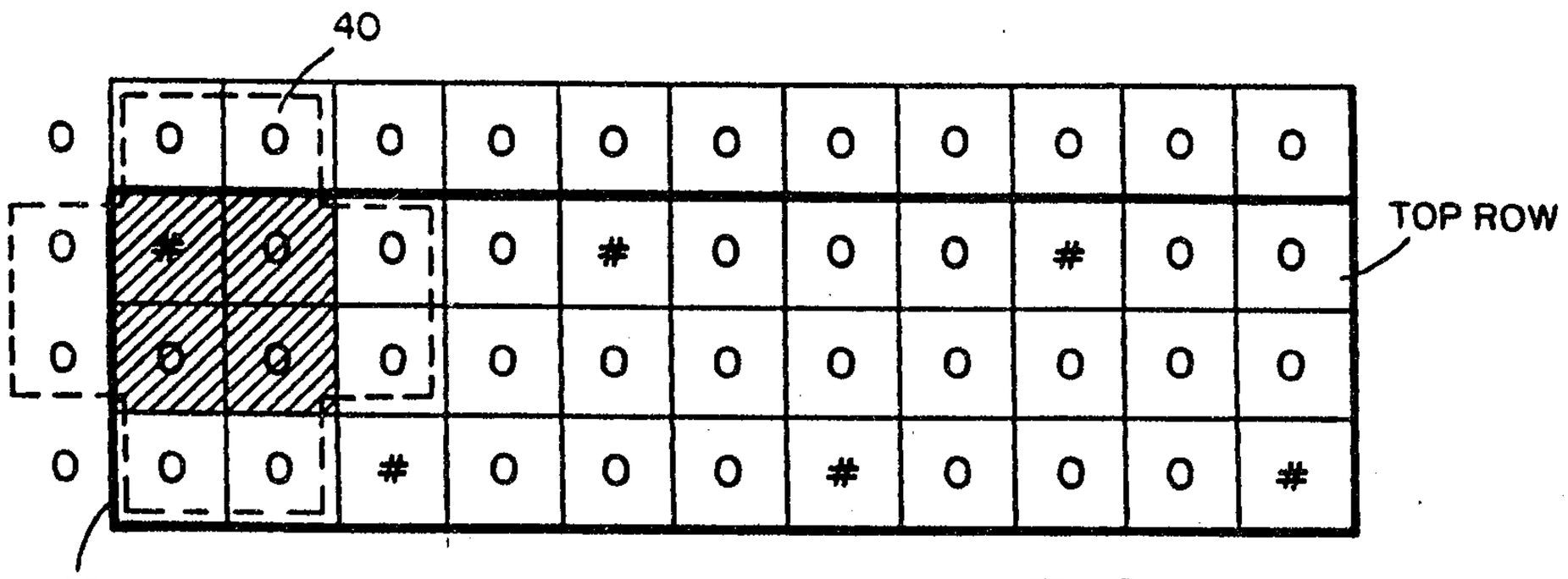


FIG. 14b

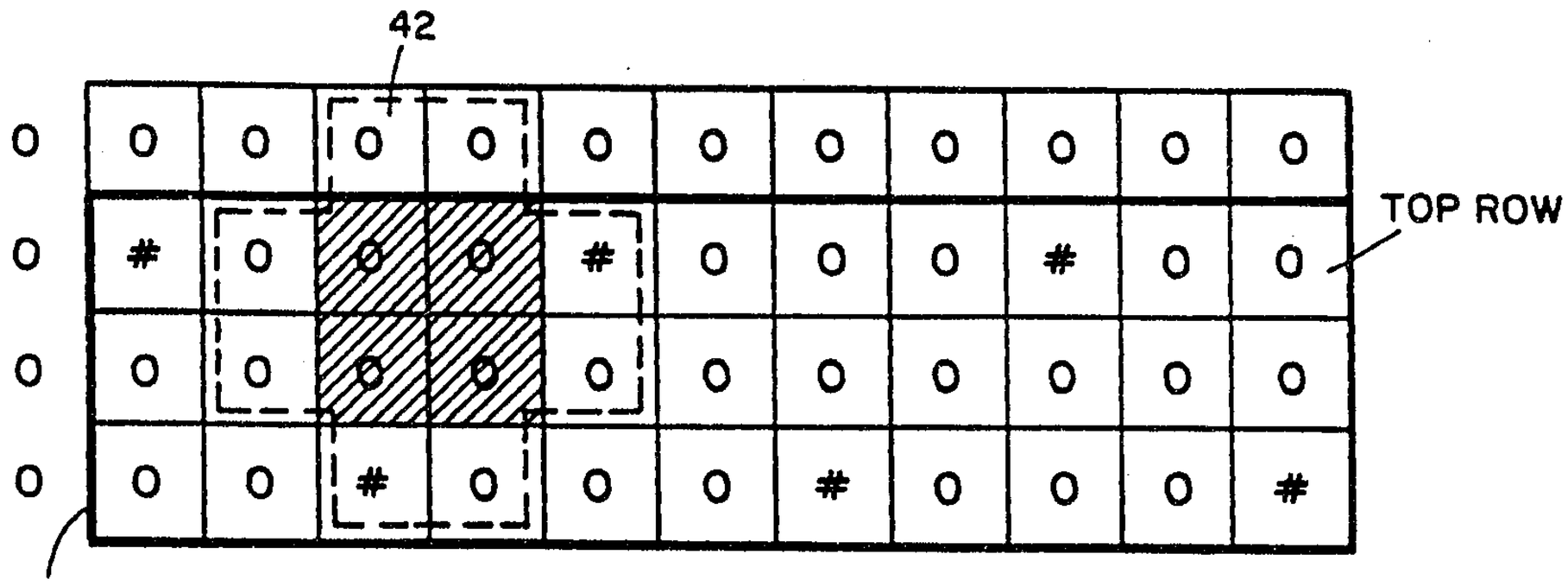


FIG. 14c

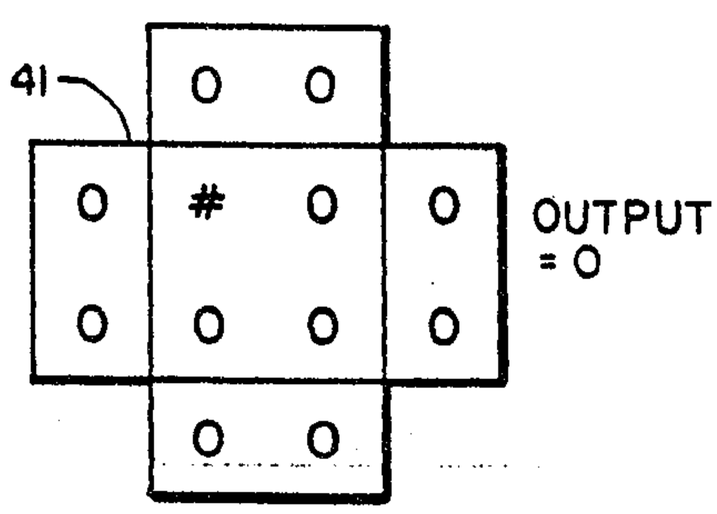


FIG. 15a

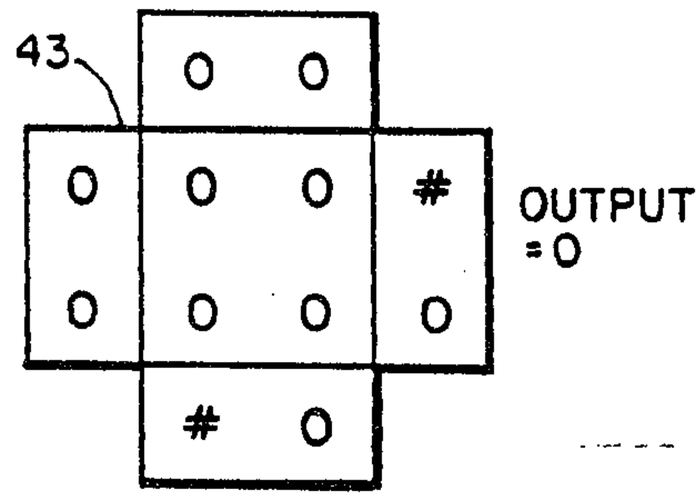


FIG. 15b

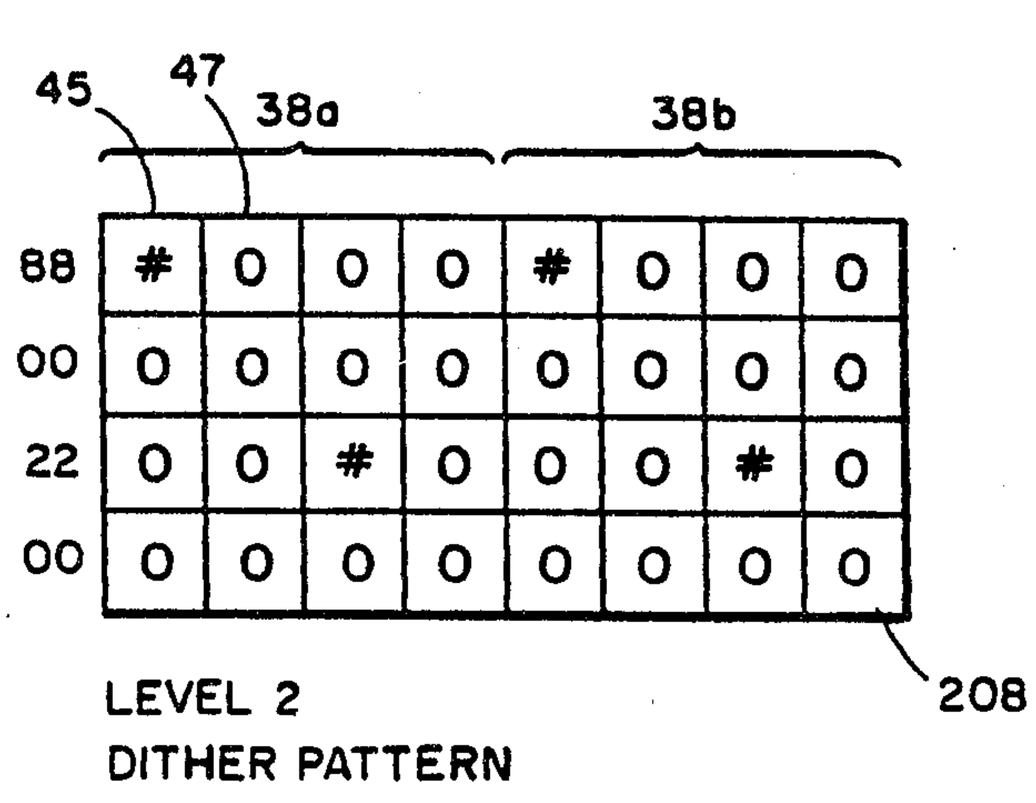


FIG. 15c

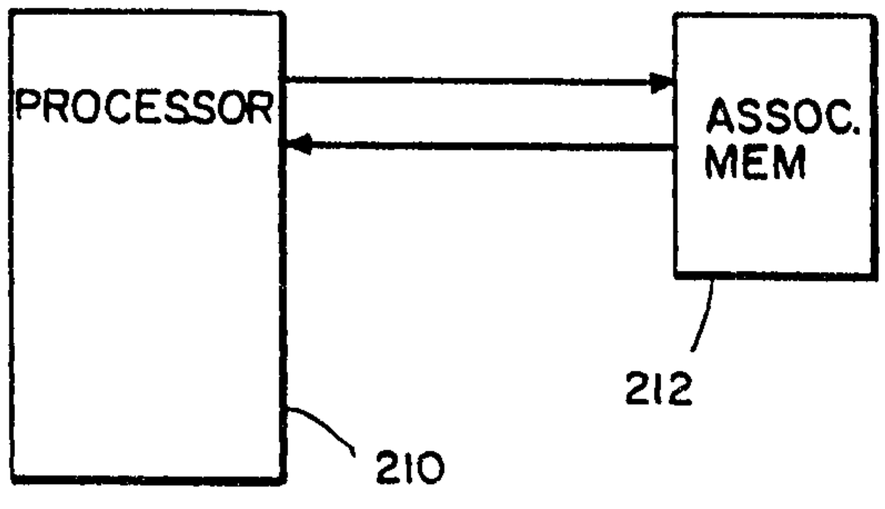


FIG. 16a

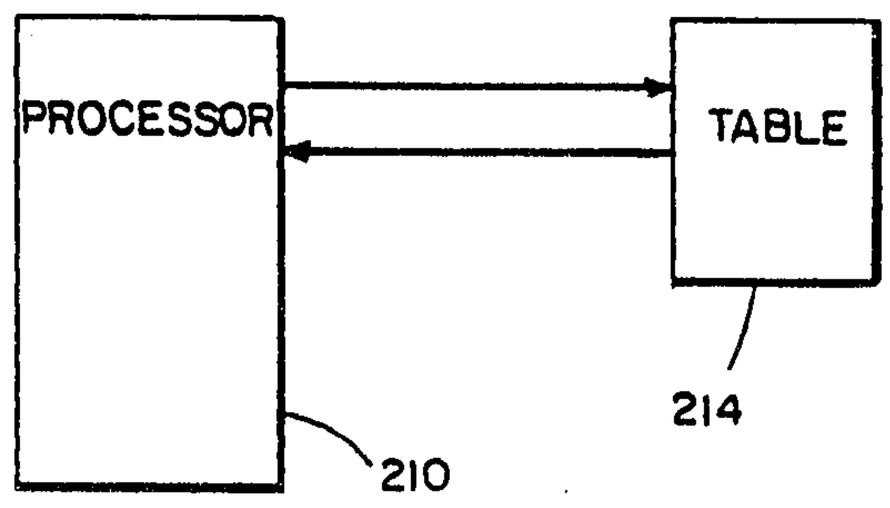


FIG. 16b

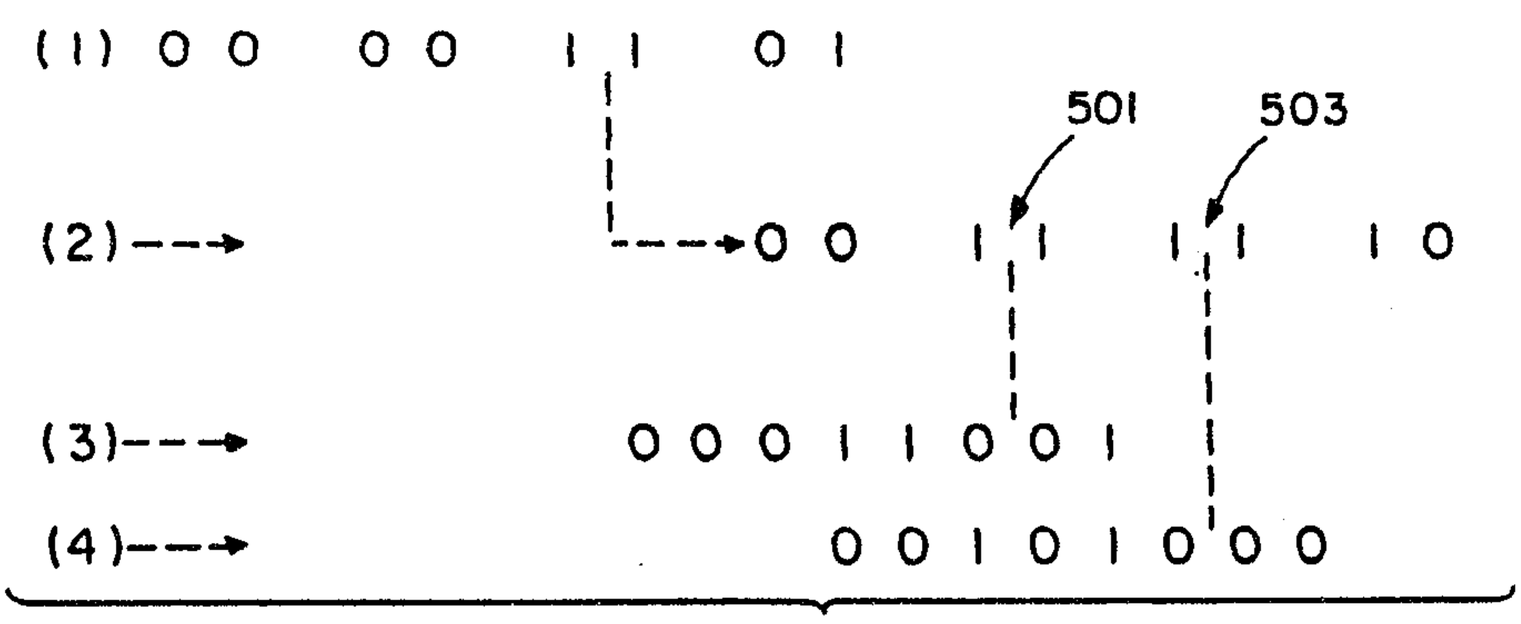


FIG. 20b

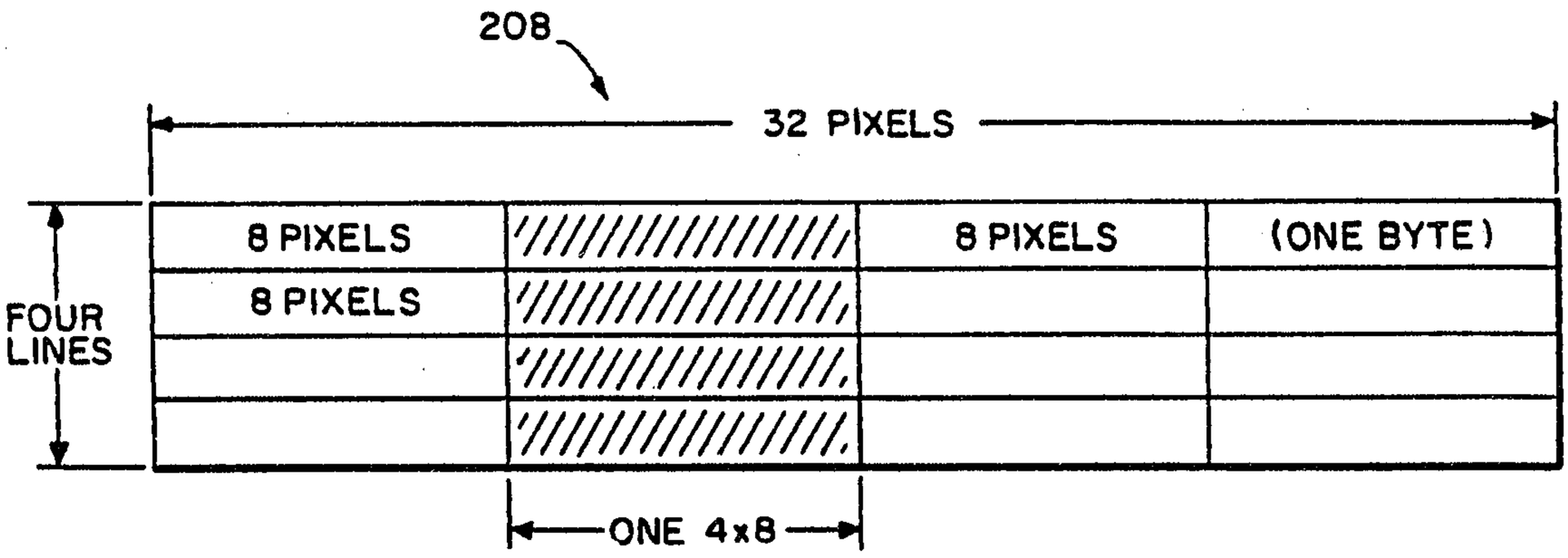


FIG. 20a

2100064

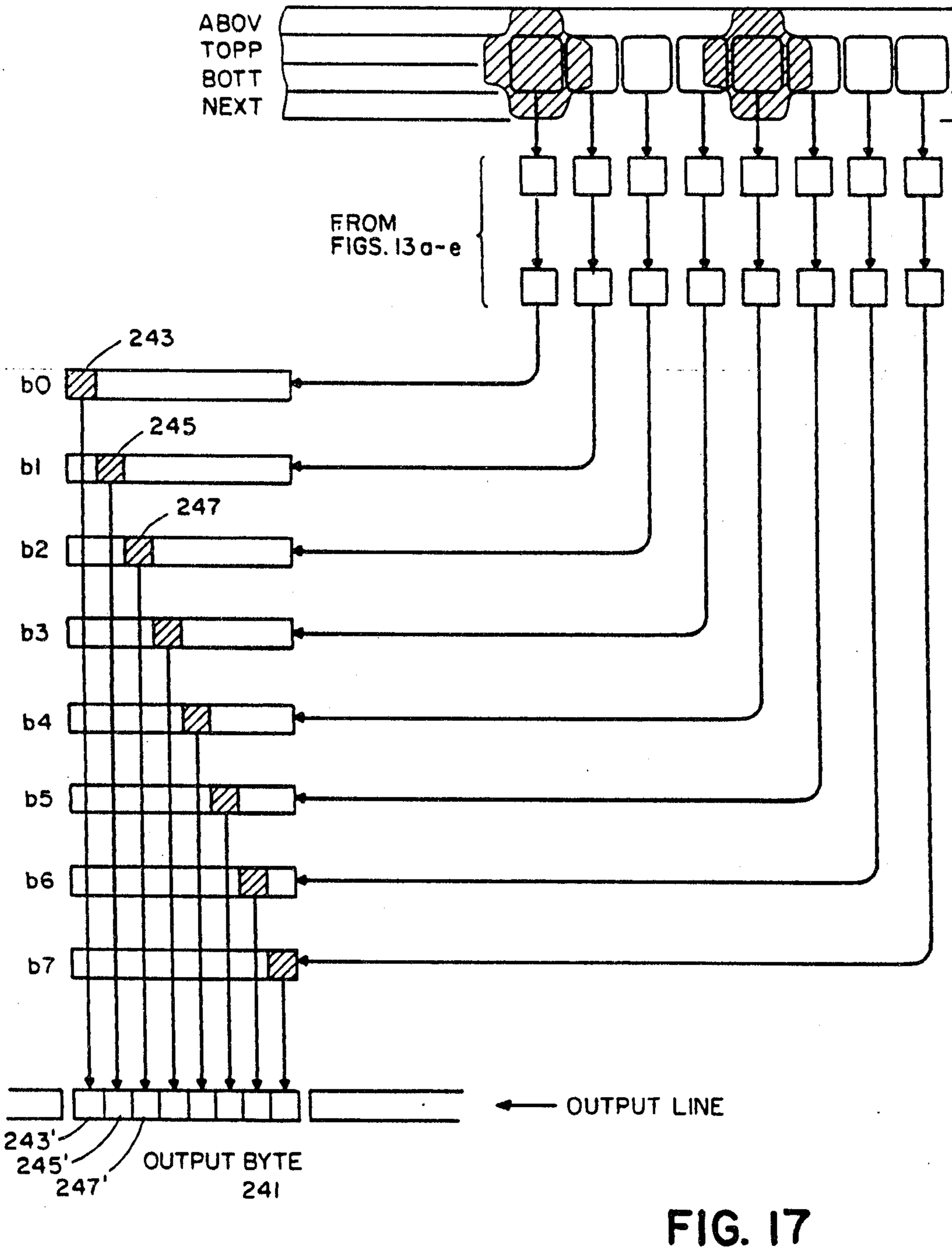


FIG. 17

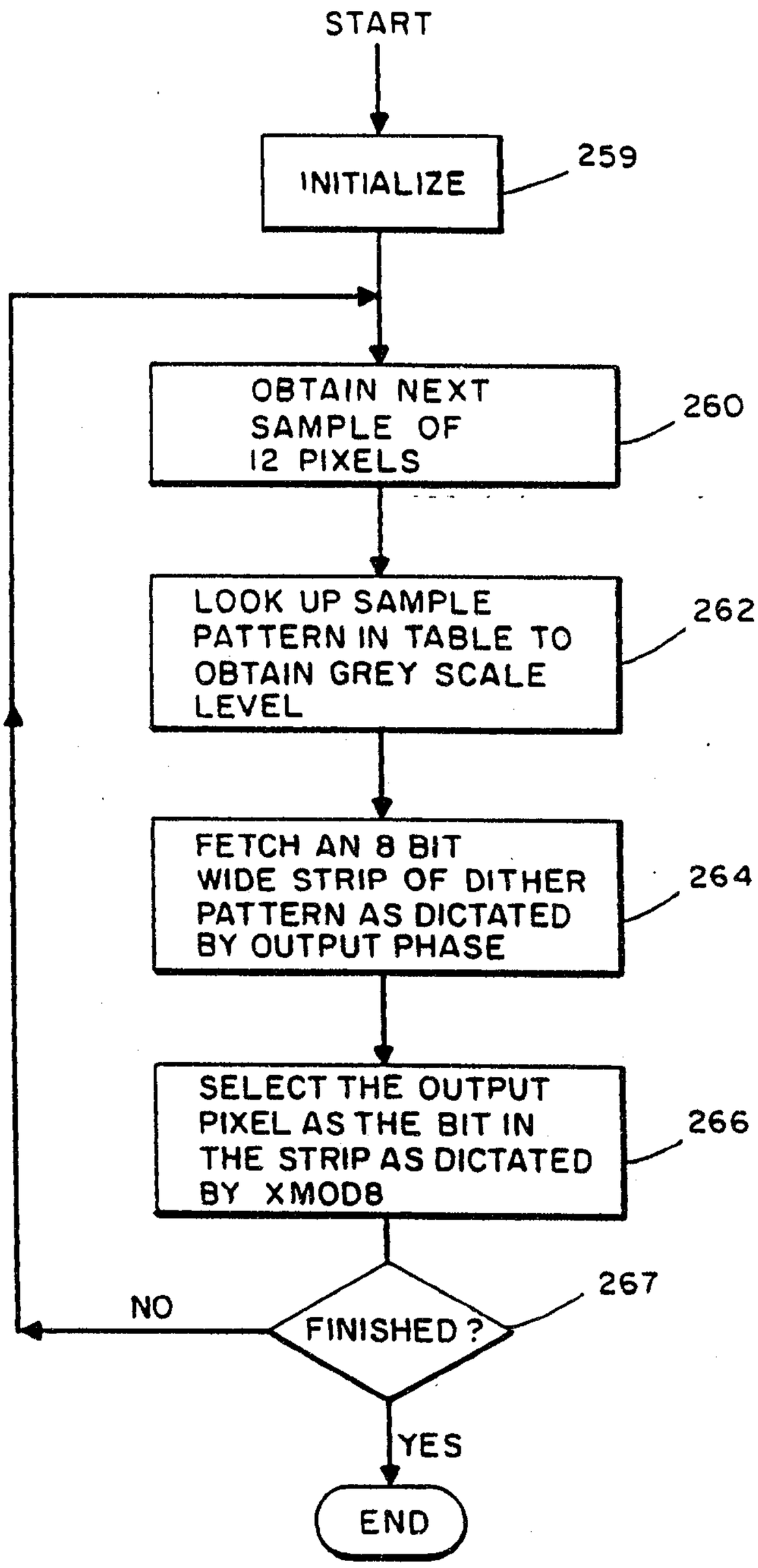

START
INITIALIZE
259
OBTAIN NEXT SAMPLE OF 12 PIXELS
260
LOOK UP SAMPLE PATTERN IN TABLE TO OBTAIN GREY SCALE LEVEL
262
FETCH AN 8 BIT WIDE STRIP OF DITHER PATTERN AS DICTATED BY OUTPUT PHASE
264
SELECT THE OUTPUT PIXEL AS THE BIT IN THE STRIP AS DICTATED BY XMOD8
266
FINISHED ?
267
NO
YES
END

FIG. 18

2100064

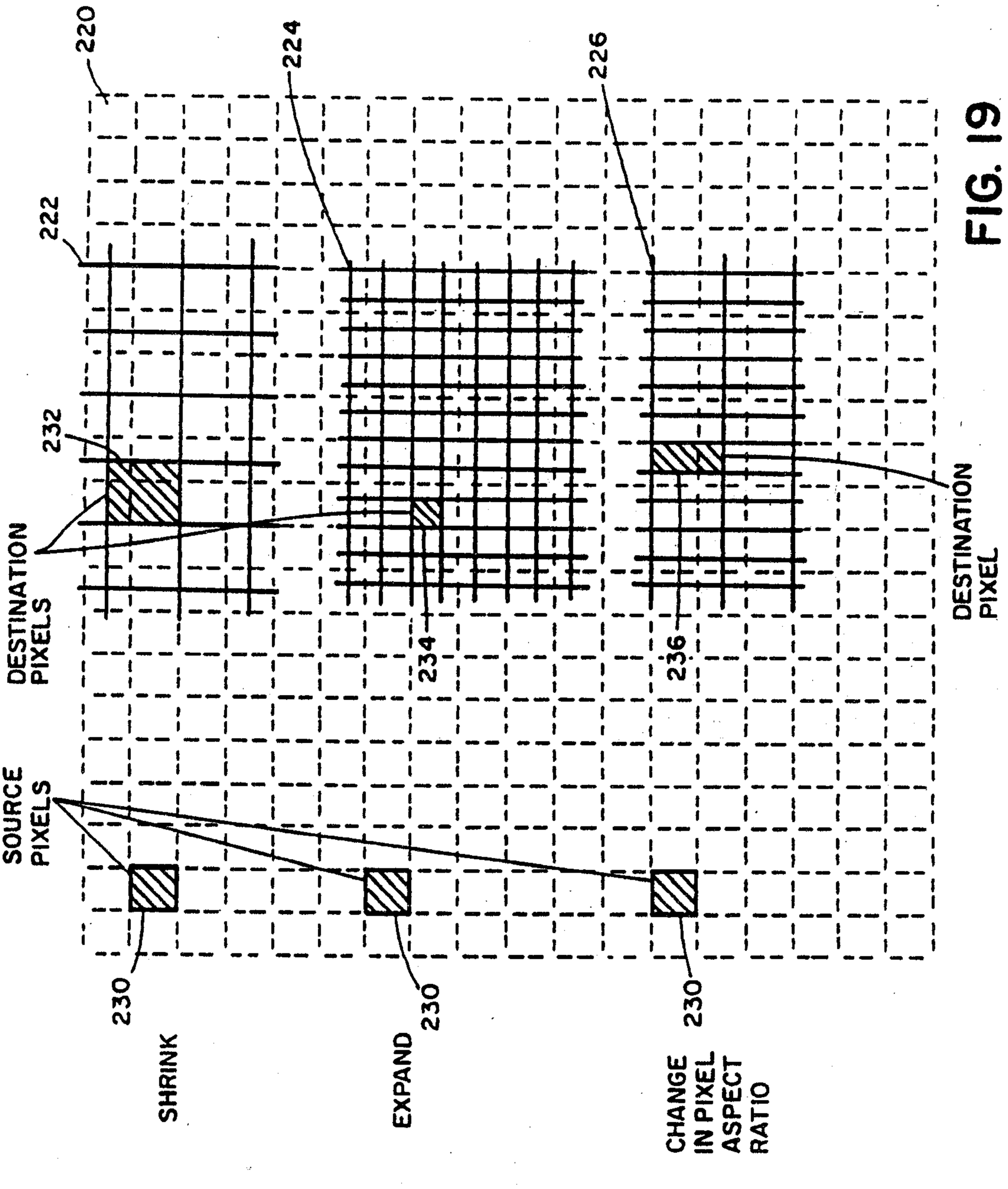
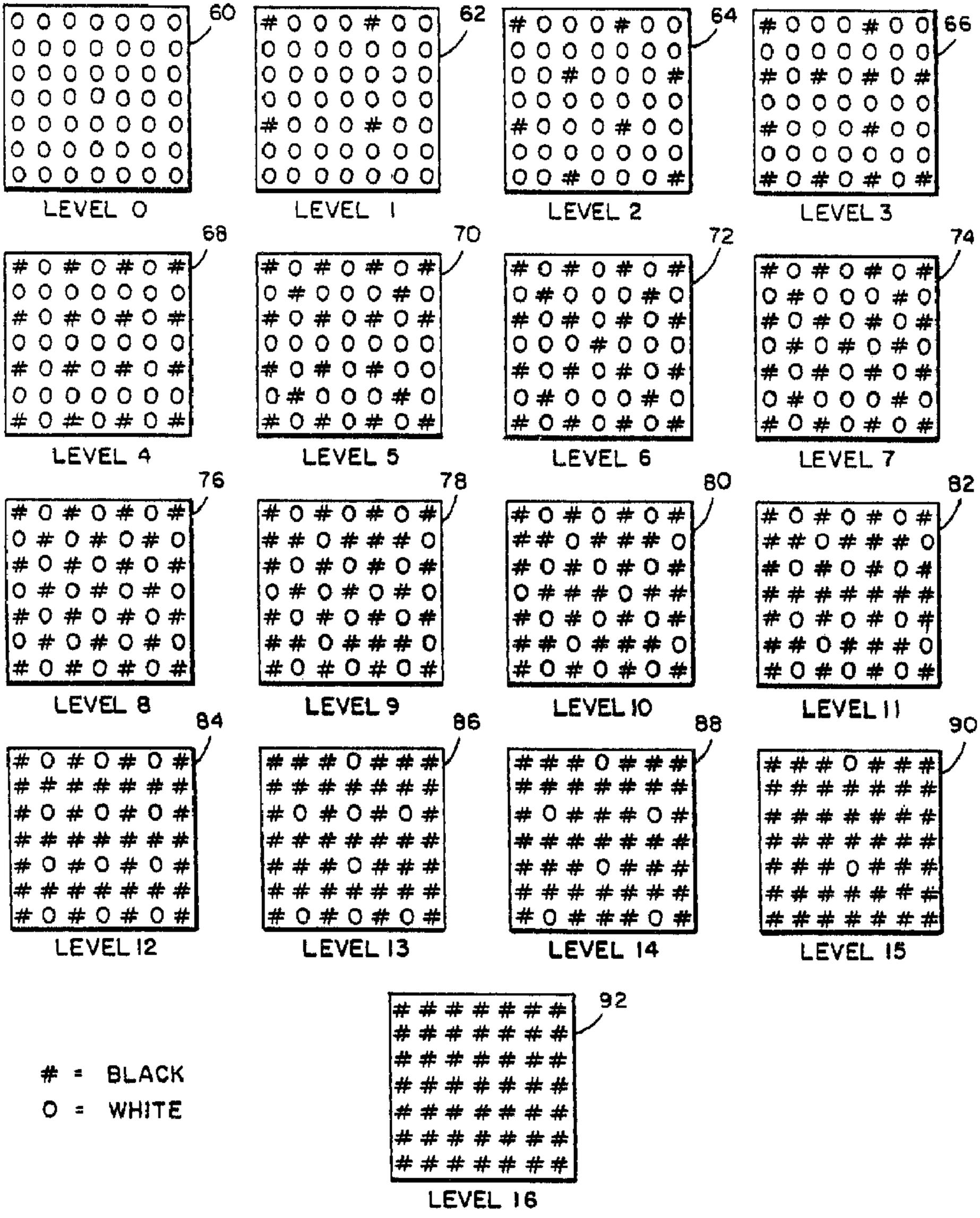


FIG. 19


60
LEVEL 0
62
LEVEL 1
64
LEVEL 2
66
LEVEL 3
68
LEVEL 4
70
LEVEL 5
72
LEVEL 6
74
LEVEL 7
76
LEVEL 8
78
LEVEL 9
80
LEVEL 10
82
LEVEL 11
84
LEVEL 12
86
LEVEL 13
88
LEVEL 14
90
LEVEL 15
92
LEVEL 16
# = BLACK
O = WHITE