OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

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

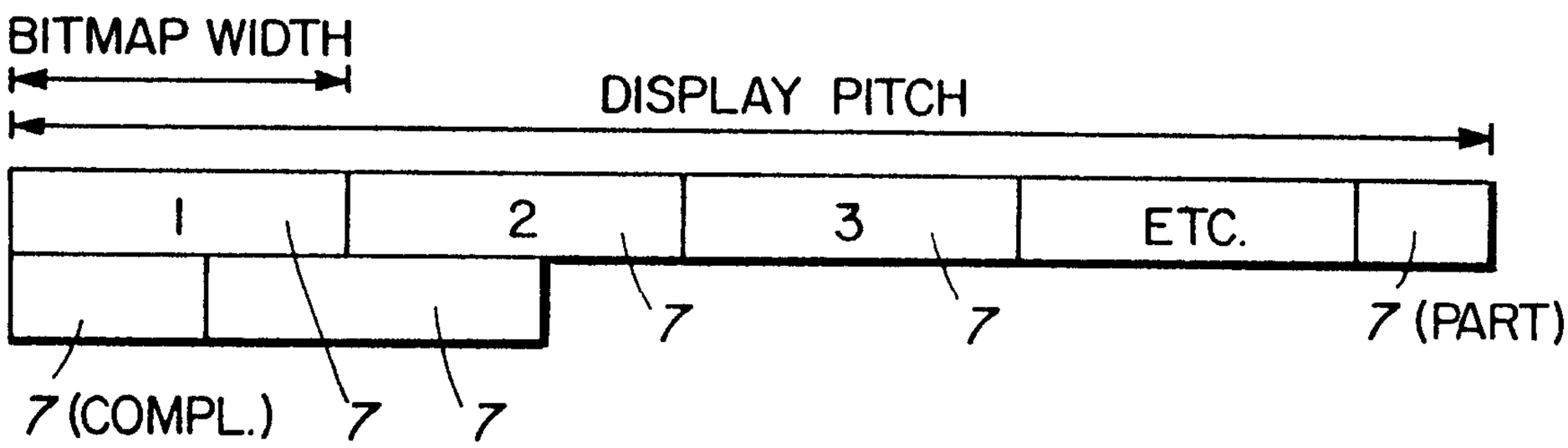
(11) (21) (C) **2,140,960**
(22) 1995/01/24
(43) 1996/03/24
(45) 2000/11/07

(72) Hartog, Adrian, CA
(72) Lum, Sanford S., CA
(72) Grossman, Josh, CA
(73) ATI TECHNOLOGIES INC., CA

(51) Int.Cl.[6] G06T 3/20, G06F 13/20
(30) 1994/09/23 (08/311,171) US

(54) **TYPE DE SOURCE A TRANSFERT DE BLOCS DE BITS LINEAIRE**

(54) **LINEAR BITBLT SOURCE TYPE**




(57) A method of managing display memory in a computer system comprised of defining a first memory portion for storing display pixel data and defining a second memory portion as a cache memory, storing pixel data from each bitmapped object line of an object in linear sequence in the cache memory, and transferring the linearly stored pixel data from the cache memory to the first portion, for subsequent processing or display.

## ABSTRACT

A method of managing display memory in a computer system comprised of defining a first memory portion for storing display pixel data and defining a second memory portion as a cache memory, storing pixel data from each bitmapped object line of an object in linear sequence in the cache memory, and transferring the linearly stored pixel data from the cache memory to the first portion, for subsequent processing or display.

FIELD OF THE INVENTION

This invention relates to computers, and in particular to a method of managing data in a display memory.

BACKGROUND TO THE INVENTION

Widespread use of graphical user interface (GUI) based computer operating systems has created a demand for high performance graphics accelerators. GUIs perform bitblt operations (bit block transfers of image data from one part of display memory to another) frequently. The bitblt operation is a basis for nearly all graphics accelerators.

In a bitblt operation, a source object such as a rectangle is copied to a destination rectangle. Normally the source rectangle is exactly the same size and follows the same direction as the destination rectangle, i.e. each pixel is moved the same number of pixels in the two plane coordinates from the source to the destination, although in the Mach8 and Mach30 graphics accelerators manufactured by ATI Technologies the source object (referred to below as a rectangle for ease of understanding the invention) can be a different shape and each pixel move in a different direction to a destination.

Graphical user interfaces also created a demand for display of higher resolutions, and for deeper pixel depths to display more colors. Memory requirements to store data of course increase for higher resolutions and larger pixel definitions (depth). For example for a 1028x768 pixel display with 8 bits per pixel, 768k bytes memory is required. Indeed, virtually all standard resolutions and pixel depths result in odd size memory requirements e.g. 640x480x16 requires 614,400 bytes, 1280x1024x16 requires 2,621,440 bytes, etc. However, memory size is sold and manufactured in increments of 1

M for high performance graphics accelerators. As a result not all of the memory is used for displaying. In the aforenoted example of the 1028x768x8 display requiring 768k, 256k of memory of a 1 M memory is not displayed.

The "left-over" part of the memory, referred to as off-screen memory, is not wasted, but is typically used by programmers to store font caches or bitmap caches, in order to reduce data traffic over the relatively slow main computer host expansion bus, and thus to improve performance of the display. Figure 1 illustrates division of a memory 1 into a display area 3 and an off-screen memory portion 5. In the display area, data is stored in bitmapped form, and thus the display pitch is shown which corresponds to the pixel width of the display or to a rectangle in the display, while the orthogonal coordinate stores display pixel data in a scanline by scanline format corresponding to the bitmap, as shown in Figure 2.

If only rectangle source type objects are allowed with the same dimensions as the destination rectangle, the usage of off screen memory is very inefficient, since gaps will appear between rectangles of various sizes. In addition, it is possible that a large font character or bitmap may not fit entirely into this offscreen memory area.

SUMMARY OF THE INVENTION

In accordance with the present invention, a bitmap from the displayed area is cached in the off screen area in unravelled form, i.e. in linear form in the memory, and in a bitblt operation of the cached data from the off-screen memory to the displayed area of the memory, the source type is specified to be linear. This provides for much more efficient operation of the memory and simpler specification of the data to be moved than

specification of a wide source rectangle as in the prior art, and has been found to require fewer register write cycles, thus improving performance. With the linear storage of the cached data in the off-screen memory, gaps in usage of storage area can be avoided, and memory addressing is simpler since page reads can be implemented.

In accordance with an embodiment of the invention, A method of managing display memory in a computer system is comprised of defining a first memory portion for storing display pixel data and defining a second memory portion as a cache memory, storing pixel data from each bitmapped object line of an object in linear sequence in the cache memory, and transferring the linearly stored pixel data from the cache memory to the first portion of the memory, for subsequent processing or display.

In accordance with another embodiment, the method described above further includes addressing the cache memory to transfer the pixel data using a linear address specification which defines a linear source type.

BRIEF INTRODUCTION TO THE DRAWINGS

A better understanding of the invention will be obtained by reading the description of the invention below, with reference to the following drawings, in which:

Figure 1 is an illustration of a display memory,

Figure 2 is an illustration of how data is stored in the displayed area of the display memory, and

Figure 3 is an illustration of how cached data is stored linearly in an off-screen memory portion of the display memory.

DETAILED DESCRIPTION OF THE INVENTION

With reference to Figure 3, the bitmap width (linear sequence of pixels) are stored line 7 by line 7 in linear sequence in the cache memory. In this figure, one of the lines at the end of a memory row or column has been split into two parts, 7(part) and 7(completion), the latter in a subsequent memory row or column following the row or column containing 7(part). In this manner the entire memory from an initial location is used without gaps.

The host application thus can store a font or bitmap image in an "unraveled" state in the off-screen cache, whereby the next scan line of an off-screen image immediately follows in the memory, rather than on the next physical scan line. To bitblt from on-screen memory to off-screen memory, the on-screen memory must be specified as to it coordinates, i.e. a first (x) axis start position, a screen width equal to the image display pitch (number of pixels), an orthogonal (y) axis start position and a number of scan lines (which equals the number of pixels in the orthogonal y direction).

In the past, the data stored in the cache memory would be similarly specified, to perform a bitblt from the cache memory to the displayed area of the memory. In accordance with the present invention, however, with the data stored in unravelled linear form in the cache memory, the bitblt source type is specified to be linear. The command need only specify a cache memory start position (a memory offset), and an enable bit if the enable bit does not arrive from another control input. The cache is read from the start position for the length of the bitmap size, the data being transferred to a destination object specified in the displayed area of the memory.

In the illustration of Figure 3, the offset is zero, since the linear storage in the cache is stored beginning at the first address location in that memory. The bitmap source is specified only by a starting point and will fill the destination.

Of course the invention is not restricted to use on a single chip, but can be used with a display memory in conjunction with other memory chips used as a cache.

A person understanding this invention may now conceive of alternative structures and embodiments or variations of the above. All of those which fall within the scope of the claims appended hereto are considered to be part of the present invention.

We claim:

1. A method of managing display memory in a computer system comprising:

a) defining a first memory portion for storing display pixel data and defining a second memory portion as a cache memory,

b) storing pixel data from all bitmapped object lines of an object in unravelled linear sequence in the cache memory, and

c) transferring the linearly stored pixel data from the cache memory to said first portion, for subsequent processing or display.

2. A method as defined in claim 1 including addressing the cache memory to transfer said pixel data using a linear address specification which defines a linear source type.

3. A method as defined in claim 2 in which said linear address specification is defined by a cache register offset.

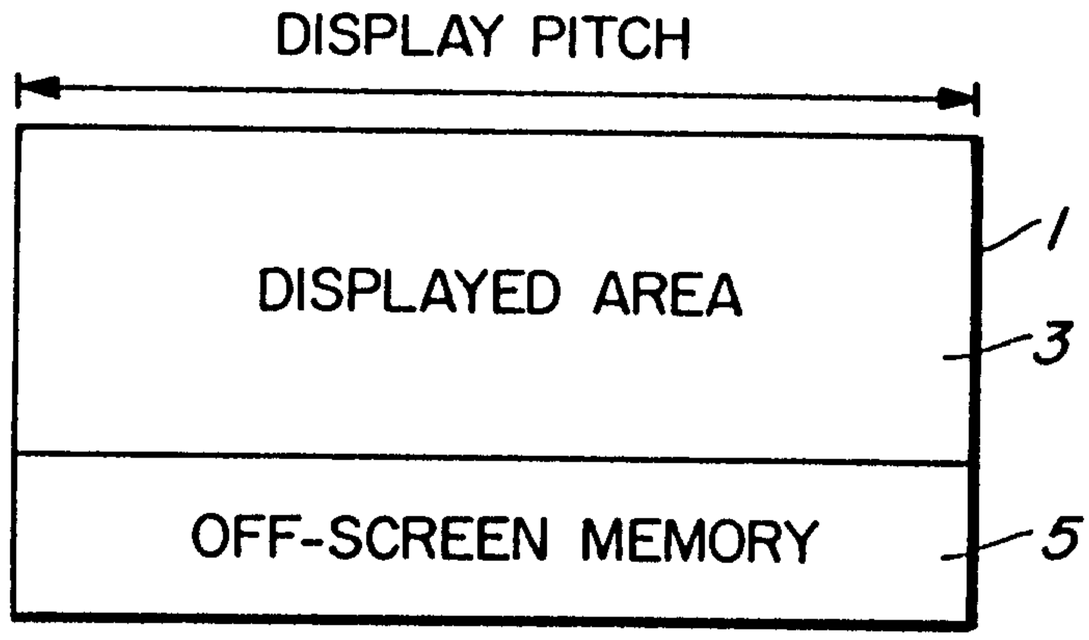



FIG. 1 PRIOR ART

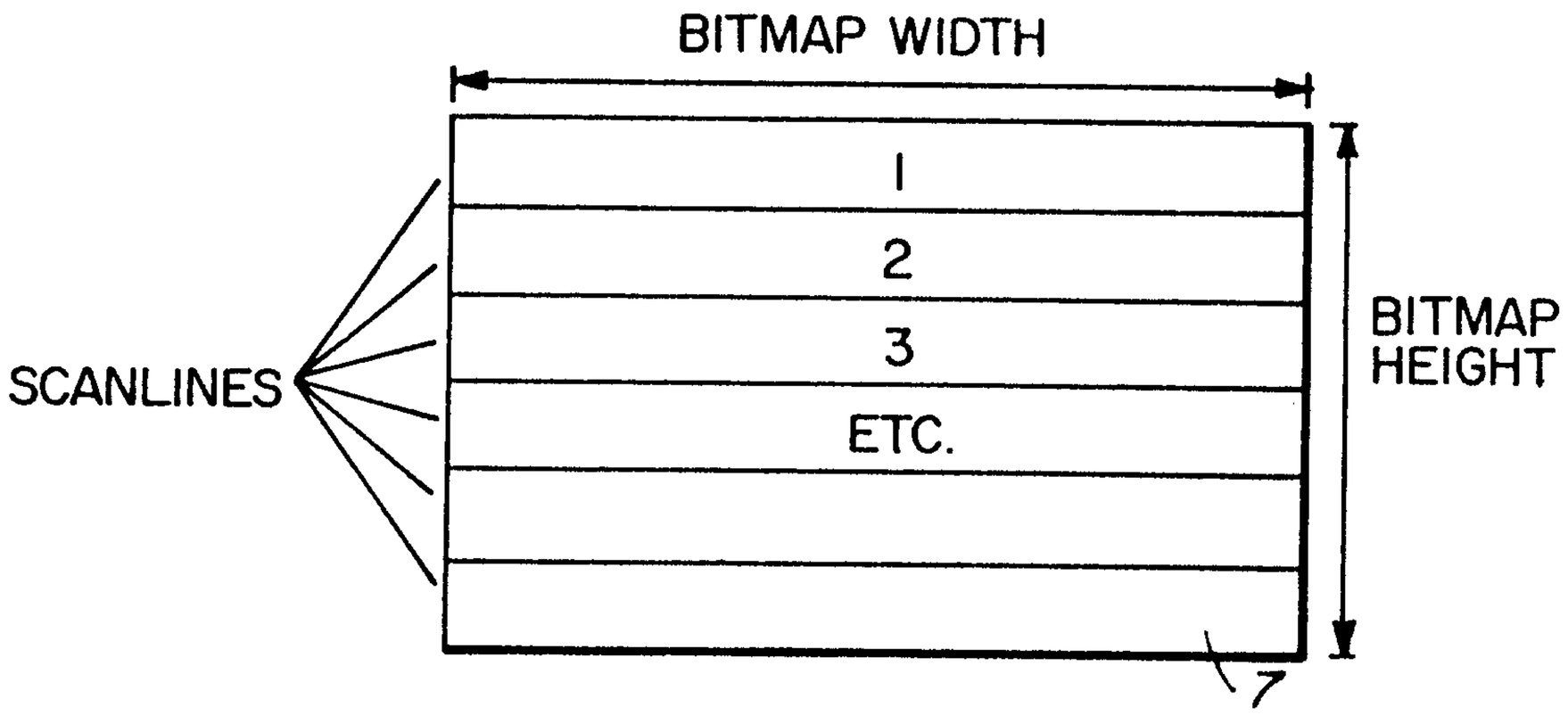



FIG. 2 PRIOR ART

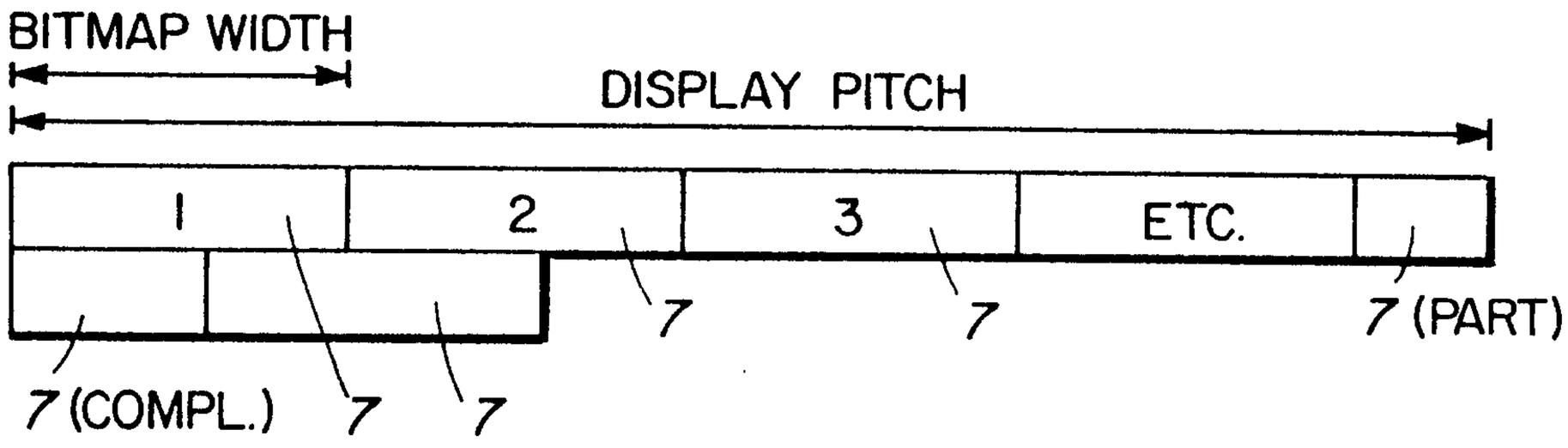
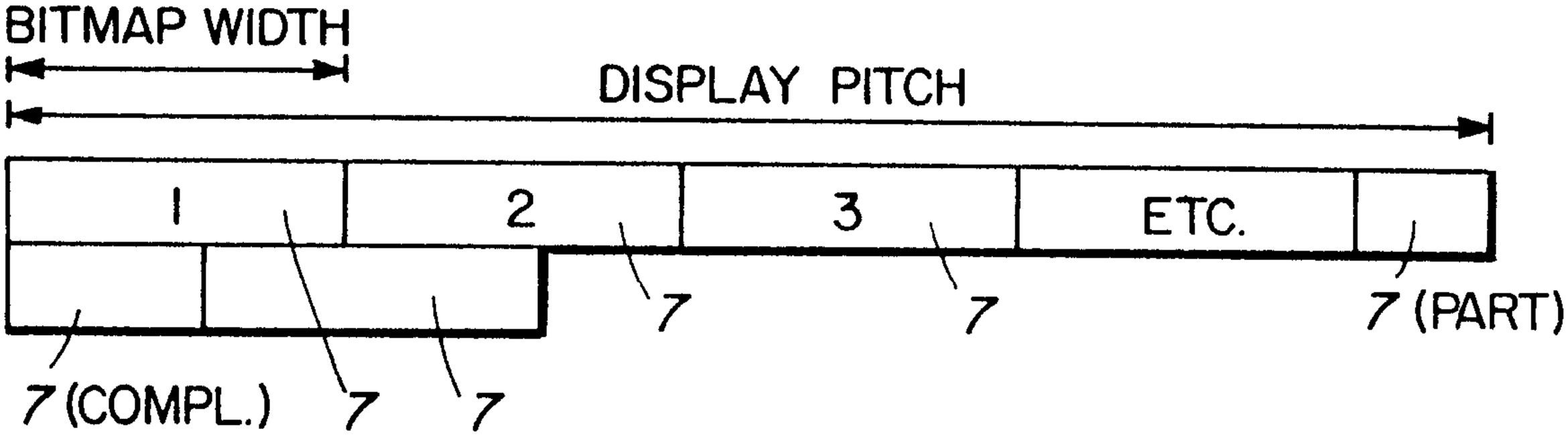



FIG. 3

A


BITMAP WIDTH
DISPLAY PITCH
1
2
3
ETC.
7 (COMPL.)
7
7
7
7
7 (PART)