

EUROPEAN PATENT APPLICATION

Application number: **86200620.2**

Int. Cl.4: **G 09 G 1/16**

Date of filing: **14.04.86**

Priority: **17.04.85 GB 8509853**

Applicant: **PHILIPS ELECTRONIC AND ASSOCIATED INDUSTRIES LIMITED, Arundel Great Court 8 Arundel Street, London WC2R 3DT (GB)**

Designated Contracting States: **GB**

Date of publication of application: **22.10.86**
Bulletin 86/43

Applicant: **N.V. Philips' Gloeilampenfabrieken, Groenewoudseweg 1, NL-5621 BA Eindhoven (NL)**

Designated Contracting States: **DE FR IT SE**

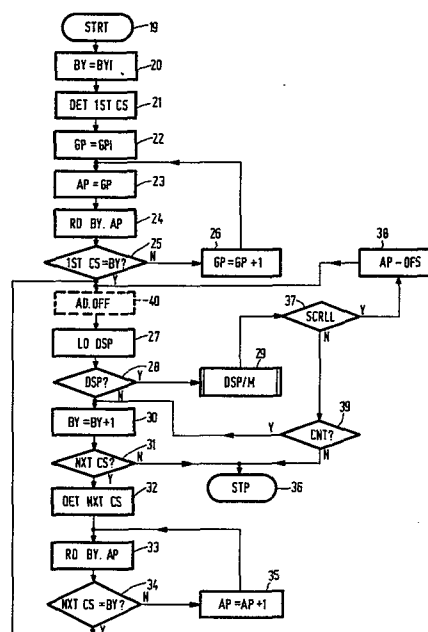
Inventor: **Penna, David Edward, c/o Philips Research Laboratories, Cross Oak Lane Redhill Surrey RH1 5HA (GB)**

Designated Contracting States: **DE FR GB IT SE**

Representative: **Boxall, Robin John et al, Philips Electronic and Associated Ind. Ltd. Patent Department Mullard House Torrington Place, London WC1E 7HD (GB)**

Data display arrangements.

In a data display arrangement which is adapted to operate as electronic information apparatus, a quantity of data stored in a display memory is displayed as an entity on a CRT display device. The stored data is selected from a large background memory under processor control in response to user command signals. The mode of selection involves the progressive comparison of each of a sequence of command signals with key data codes to transfer discrete portions of the data from the background memory to the display memory as determined by each command signal in conjunction with each preceding command signal. Fig. 5 shows a flow chart of the selection mode.



"DATA DISPLAY ARRANGEMENTS"

This invention relates to digitally operable data display arrangements for displaying on the screen of a CRT (cathode ray tube) or other raster scan display device, a quantity of data which is stored in a display memory and is accessed repeatedly for its display in a recurrent cycle of scanning lines. The scanning lines, which can be produced with or without interlaced field scanning, are modulated at a given pixel or dot rate to produce a dot matrix display (e.g. a 320 x 250 resolution dot matrix colour display).

In a data display arrangement of the type set forth above, the stored data can comprise at least one information bit in respect of each of the pixels or dots which represent the data as produced on the screen of the display device by the scanning action. Provided that the information bits for the successive dot rows of the display are read out from the display memory in the same lines of the scanning cycle in each field, the display is a static display.

This static display can be "scrolled" by modifying the read out operation every few fields such that information bits for the dot rows of the display are read out from the display memory in progressively different scanning lines. The display can be cycled/wrapped-around, if dot rows "lost" at the top of the display are re-inserted at the bottom. Alternatively, the "lost" dot rows can be replaced by new dot rows by progressively changing the stored data in the display memory as the scrolling takes place. This produces a continuously up-dated display.

If, every few fields, the information bits for each dot row are read out in the preceding adjacent scanning line rather than the scanning line in which they were currently being read out, then visually smooth scrolling occurs because the display is moved (up) only one scanning line every few fields. Visually smooth scrolling would also occur if the display is moved up only a few scanning lines every few fields. The scrolling can be made hard, i.e. less smooth such that it is visually "jerky", by moving up the display a significant number of scanning lines every few fields.

In a data display arrangement of the type set forth above display scrolling is a convenient way of searching through data, the rate at which the scrolling (smooth or hard) is effected being determined by how frequently the scanning lines for displaying the dot rows are changed. However, in applications where a large amount of data is to be accessed and displayed a portion at a time in order to find a particular item, it can become impracticable to search through the data by scrolling. Although it is apparent that scrolling can be made to take place at any desired rate, the actual maximum rate that can be used in practice is limited by the ability of a user to read the data during scrolling.

It is an object of the present invention to provide a data display arrangement of the above type as an electronic information apparatus for accessing and displaying a portion at a time a large amount of data in a simple and improved manner.

According to the invention such electronic information apparatus comprises a background memory wherein a list of items of information is stored, each item comprising at least two data parts represented by respective coded bytes and each item being stored at a respective memory location; a display memory for storing data representing a sub-list of said items as selected from the background memory for display on a display screen of a display device; a display generator for producing video signals in accordance with the data stored in the display memory for driving the display device; and a data processor and an associated user interface device for controlling the selection and display of sub-lists of selected items each comprising a number of characters; in which apparatus the items of the list are organised in groups, the items of a same group having a same first coded byte and being stored at respective memory locations which form a group; and the data processor is operable:-

- to generate one at a time a plurality of group addresses which identify respectively an initial memory location in each memory location group,
- to generate one at a time subsequent addresses of memory

locations whose group has been identified by the relevant group address,

- to set an address pointer to the memory location whose address is generated last,
- 5 - to set a byte pointer to indicate in turn the byte positions of a memory location to which the address pointer is set, and
- to compare a command signal received from the interface device and representing a character with the coded byte in the byte position indicated by the byte pointer in the memory location
- 10 to which the address pointer is currently set.

The data processor may be further operable:-

- (a) under control of an initialisation signal generated by the user interface device to set the address pointer to the first memory location of the first memory location group and to set the byte pointer to indicate the first byte position,
- 15 (b) under control of a first command signal indicating the first data part of an item to be selected, to read the coded byte from the byte position indicated by the byte pointer and to supply the read code byte to a comparator,
- 20 (c) under control of an 'error' signal, generated as a result of non-correspondence between the read coded byte and the first command signal, to set the address pointer to the group address of a subsequent memory location group, to
- 25 read the coded byte indicated by the byte pointer and to supply the read coded byte to the comparator,
- (d) under control of a 'result' signal, generated as a result of correspondence between the read coded byte and the first command signal, to set the byte pointer to indicate
- 30 the second byte position, and to set the address pointer to a subsequent memory location in the same memory location groups under control of a further command signal
- (e) indicating a subsequent data part of the item to be selected, to read the coded byte from the byte position
- 35 indicated by the byte pointer and to supply the read

coded byte to the comparator;

- 5 (f) under control of an 'error' signal, generated as a result of non-correspondence between the read coded byte and the further command signal, to set the address pointer to a subsequent memory location in the same memory location group, to read the coded byte at the byte position indicated by the byte pointer, and to supply the read byte to the comparator; and
- 10 (g) under control of a 'result' signal, generated, as a result of correspondence between the read coded byte and the further command signal, to set the byte pointer to indicate a subsequent byte position.

The invention thus affords the advantage that it provides a means whereby, without a scrolling technique being used, a selected
15 sub-list of a given size taken from a much larger list of items, can be displayed selectively as an entity, with the selected sub-list being readily progressively altered in response to different combinations of sequentially produced command signals.

The invention was primarily envisaged as a practical means for
20 searching through a dictionary list of words arranged in alphabetical order. Only a relatively few of the words of the list can be displayed at any one time, so that with the list containing several thousand words, which is likely, it becomes impracticable to search through the list using known scrolling techniques to find
25 a particular word. The present invention enables the list to be accessed a portion at a time by the selective use of a sequence of command signals which control the display of successive discrete portions of the list as sub-lists until a portion containing the required word is displayed.

30 In carrying out the invention, particularly for the above-mentioned application thereof to a word list, the interface device is suitably a data entry keyboard device of known type, for instance a conventional ASCII keyboard device. The (ASCII) codes which are produced in response to key operation constitute said
35 command signals. The use of such a keyboard device thus provides a

simple means of producing a sequence of 'select' command signals.

In electronic information apparatus in accordance with the invention the data processor may be further operable to provide also a scrolling function in response to 'scroll' command signals
5 from a second interface device. Although such a scrolling function would in practice normally be used only for local scrolling relative to a displayed sub-list of items, it could in principle be used to scroll through all the whole list of items in the background memory.

10 The second interface device is suitably a graphics tablet the pen of which provides successive indexing 'scroll' command signals in response to vertical movement (up or down) of the pen on the graphics tablet.

Preferably, both for the selection and display of a sub-list
15 of items in response to 'select' command signals and for the selection and display of a modified sub-list by scrolling in response to 'scroll' command signals, the processor is arranged to redefine the entire contents of the display memory for each different display.

20 In order to scroll the display a short distance, the data in the display memory could be re-located or 'lost', as necessary, and new data for display could then be introduced into the display memory at the appropriate locations. However, it is preferred simply to redefine the entire contents of the display memory for
25 each change in the display.

Conveniently, the entire display is redefined every field scan of the display device using a scan synchronisation technique, which involves erasing the data in the display memory progressively for each scan line of the display after the data has been used for the
30 display in that scan line and inserting into the display memory the same or new data, as required, before that data is required for the display in that scan line of the next field. Assuming standard TV scanning of 50 fields per second, plus 4ms field blanking, approximately 30ms is available in which to redefine the display.
35 In effect, the display is redefined every field scan although the

scan synchronisation is effected over two successive field scans, as will be described. This technique gives the advantage that an adequate period is available between successive displays for re-defining the data in the display memory.

5 In order that the invention may be more fully understood, reference will now be made by way of example to the accompanying drawings, of which:-

Figure 1 shows a block diagram of an electronic information apparatus embodying the invention;

10 Figure 2 shows a table for the ASCII data code;

Figures 3 and 4 show memory maps illustrating the principles of the invention;

Figure 5 shows in the form of a flow chart logic operations of the apparatus in the performance of the invention; and

15 Figures 6a to 6e shows diagrammatically a scan synchronisation technique.

Referring to the drawings, the electronic information apparatus shown in Figure 1 comprises a display device 1, a display generator 2, a data processor 3, a background memory 4, a display
20 memory 5 and user interface apparatus 6 and 7. The display device 1 is suitably a colour television monitor which is connected to receive R, G, B, video signals from the display generator 2. These R, G, B, video signals are produced in the display generator 2 by three digital-to-analogue converters 8, 9 and 10, respectively.
25 The display generator 2 also includes a colour look-up table 11 which is suitably a read/write memory and is responsive to dot information received from the display memory 5 over a bus 12 to produce digital signals for driving the converters 8, 9 and 10. A display timer 13 in the display generator 2 provides line and field
30 synchronisation signals LS and FS for the television monitor 1 over a connection 14. The timer 13 also provides over a connection 15 timing signals T for controlling the transfer of dot information from the display memory 5 to the colour look-up table 11.

The display memory 5 is suitably a random-access memory which
35 has a capacity for storing dot information for at least one display

frame. The dot information would comprise one or more information bits per dot to be displayed, depending on the range of colours afforded by the colour look-up table 11. A combined address/data bus 16 interconnects the display generator 2 and the display memory 5 with the data processor 3. The background memory 4, which is also at least partially a random-access memory, is also connected to the address/data bus 16. The background memory 4 may also have a read-only memory part which contains permanent program data for controlling the "house-keeping" operations of the data processor 3. The user interface apparatus comprise a keyboard data entry device 6 and a graphics tablet 7. The data processor 3 can be a commercially available microprocessor (μ p), for instance the Signetics S68000 μ p.

It may be assumed that data is loaded into the random-access part of the background memory 4 from a mass memory, for example, from an optical record carrier (e.g. a compact disc in its role as a CD ROM). This data is assumed to be textual and comprises multi-bit codes which represent text characters. When a text character is to be displayed, the code therefor is read from the RAM memory 4 by the data processor 3 and written into the display memory 5 at an appropriate location as dot information.

Consider now the application of the present invention for, for example, displaying any selected portion of given size of a dictionary list of words which contains words beginning with all the letters of the alphabet. The list can comprise many thousands of words, for example 30,000 words, which is the sort of capacity of a small English language paper dictionary. The words are in alphabetical order in the list. It is assumed each letter of the alphabet is represented by a respective 7-bit code combination of the ASCII code.

A table for the full ASCII code is shown in Figure 2, although for the purposes of the present description only the codes for certain letters of the alphabet will be referred to. In the bit notation used in the table, bit b7 is the high-order, and bit b1 the low-order, bit position. Thus, the 7-bit code for the letter

"R" is b7(1), b6(0), b5(1), b4(0), b3(0), b2(1) and b1(0).

In the background memory 4, each word of the list of words is stored at a separate memory location, each memory location comprising a plurality of byte positions BY1, BY2, ... BY_n of 8 bits b8 to b1 (see Figure 4), one byte position for each of the letter codes which represent the letters of the word. Assuming a list of words containing 30,000 words, as aforesaid, 2¹⁵ (32K) different memory locations are required for this purpose in the background memory 4. In hexadecimal notation these memory locations can be identified by respective address codes in the range 0000 to FFFF.

A memory map for this part of the background memory 4 is illustrated diagrammatically in Figure 3. In this memory map, which is designated by the reference numeral 17, the first memory location is identified the hexadecimal address code 0000 and the last memory location is identified by the hexadecimal address code 7530. The memory locations are divided into a number of different groups GP1 to GP26 of different size. Each group contains memory locations for storing the letters codes for all the words beginning with a respective particular letter, so that the size of a group that is, how many memory locations it contains, depends on the number of words in the word list that begin with the letter concerned. The arrow 18 represents an address pointer which identifies any particular memory location from which read out is to occur.

Consider for example the memory location group GP15 which comprises the memory locations in which are stored all the words in the word list that begin with the letter 'O'. If the hexadecimal address code for the first memory location of this group is, say, 3A98 and there are 500 sequential memory locations in the group, then these memory locations will have the address codes 3A98 to 3C8C, as indicated. Group GP1, which comprises the memory locations in which are stored all the words in the word list that begins with the letter 'A', is shown as having 900 sequential memory locations with address codes 0000 to 0385. Group GP26 has

only 40 sequential memory locations with address codes 7509 to 7530. This latter group comprises the memory locations in which are stored all the words in the word list that begin with the letter 'Z'.

5 The group GP15 for words beginning with the letter 'O' is shown more fully in Figure 4. By way of example, it is assumed that the first 60 memory locations, and the 75th, 76th and 500th memory locations of this group contain, respectively, the letter codes for the following words which begin with the letter 'O'.

10	1. OAF	21. OBJURGATE	41. OBSESS	
	2. OAK	22. OBLATION	42. OBSIDIAN	
	3. OAKUM	23. OBLIGATO	43. OBSOLETE	
	4. OAR	24. OBLIGE	44. OBSTACLE	
	5. OASIS	25. OBLIGATION	45. OBSTETRIC	
	6. OAST	26. OBLIQUE	46. OBSTINATE	
	7. OAT	27. OBLITERATE	47. OBSTREPEROUS	
15	8. OATH	28. OBLIVION	48. OBSTRUCT	
	9. OBDURATE	29. OBLONG	49. OBSTRUENT	
	10. OBEDIENCE	30. OBLOQUY	50. OBTAIN	
	11. OBEISANCE	31. OBNOXIOUS	51. OBTRUDE	
	12. OBELISK	32. OBOE	52. OBTUSE	
	13. OBESE	33. OBOLE	53. OBVERSE	
	14. OBEY	34. OBOVATE	54. OBVIATE	
	15. OBFUSCATE	35. OBSCENE	55. OBVIOUS	
20	16. OBITER	36. OBSCURE	56. OCARINA	
	17. OBITUARY	37. OBSECRATE	57. OCCASION	75. OCTROI
	18. OBJECT	38. OBSEQUIES	58. OCCIDENT	76. OCULAR
	19. OBJECTIVE	39. OBSEQUIOUS	59. OCCLUDE	
	20. OBJURE	40. OBSERVE	60. OCCULT	500. OZONE

In Figure 4, the letter codes for the 1st, 9th, 24th, 32nd, 25 53rd, 76th and 500th words are shown in their respective memory locations 3A98, 3AA0, 3AAF, 3AB9, 3ACD, 3AE3 and 3C8C. Each letter code comprises 7 bits in correspondence with the ASCII table shown in Figure 2. The left-hand (high-order) bit position b8 of each 8-bit byte is unused. The address pointer 18 is shown again 30 in Figure 4. It will be appreciated that, in practice a memory location can comprise more than one actual memory segment which is addressable by a respective memory address, depending on the size of the word and the byte capacity of the memory locations. Each of the different bytes BY1, BY2, ... BYn of a word is indicated by 35 means of a byte printer 45.

The logic operations which are performed by the processor (3-Figure 1) of the electronic information apparatus to produce the display of a selected portion of the stored list of words will now be explained in outline with reference to the flow chart shown in Figure 5. For this explanation, it is convenient to consider the request by a user of the display of that portion of the word list which contains a particular word which is located centrally in the displayed portion. The word chosen for this purpose is 'OBVIOUS' which will be typed on the keyboard device 6 to provide a sequence of command signals which are the respective ASCII codes for the typed letters.

In the flow chart the legends in the various boxes have the following meanings.

- 19 - STRT - when the apparatus is switched-on,
initialisation occurs to set various counters,
flags, indicators, etc., to their starting
values.
- 20 - BY = BY1 - a byte pointer is set to indicate the first
byte position of the memory locations.
- 21 - DET 1st CS - the first command signal produced by the
keyboard device 6 is fetched and stored in a
first register.

(In the present instance, this first command signal will be the ASCII code for the letter 'O': i.e. 1001111).

- 22 - GP = GP1 - a group memory location pointer is set to
indicate the first position of a group code
table wherein there are stored group codes
which identify respectively the groups of
memory locations.

(Each group of memory locations has an individual group code allotted to it. The group codes can be the address codes for the first memory location of the respective groups. Alternatively, the group codes can be the letter codes in the first byte positions of the respective groups. Thus, in the present instance, the group code for the group GP15 may be hexadecimal address code 3A98 or

binary letter code 1001111).

23 - AP = GP - an address pointer is set to indicate the
memory location of the group identified by the
group memory location pointer.

5 24 - RD BY.AP - read out the letter code stored in the first
byte position (BY = BY1) of the memory
location indicated by the address pointer and
store the read letter code in a second
register.

10 (In the present instance, this letter code will be ASCII code
for the letter 'A': i.e. 1000001).

25 - 1stCS = BY? - the contents of the first and second registers
are compared with each other.

15 (In the present instance this comparison will fail because
1001111 $\neq$ 1000001 and an "error" signal will thus be generated).

26 - GP = GP+1 - the group memory location pointer is
incremented to the next position of the group
code table to identify the next group of
memory locations.

20 (Step 23 is then returned to to set the address pointer to the
first memory location of the next group. The letter code in the
first byte position BY1 of the memory location of this next group
(i.e. letter code 'B' = 1000010) is then read and stored in the
second register by step 24. Step 25 is then performed again,
25 followed by step 26; and these steps 23, 24, 25 and 26 are
thereafter repeated until the comparison executed by step 25 is
satisfied, i.e. when GP = 3A98 (or 1001111) and BY = 1001111).

27 - LD DSP - when step 25 detects correspondence between
the compared values CS and BY, the words
30 stored in the first 24 memory locations of the
group concerned, i.e. group GP15 in the
present instance, are read out and the
appropriate dot information for these words is
written into the display memory 5, as
35 mentioned previously.

(Because the address pointer is incremented with the group memory location pointer it is now set to the first memory location of group GP15 to identify it as the starting location for this read out).

- 5 28 - DSP? - this is a logic decision to display or not
display the contents of the display memory 5
on the display device 1.

(This logic decision is for example realised by the operation of a dedicated switch by the user. It is assumed that there are 24
10 words which can be displayed as a sub-list at any time as the selected portion of the word list, hence the read-out from the first 24 sequential memory locations of the group GP15. This sub-list therefore comprises the words 'OAF' to 'OBLIGE' in the foregoing word list, as read from the memory locations with the
15 address codes 3A98 to 3AAF).

- 29 - DSP/M - this is a sub-routine for displaying the
contents of the display memory.
- 30 - BY = BY+1 - the byte pointer is incremented by '1' to
indicate the next byte position BY(i) of the
20 memory locations (of the detected group).
- 31 - NXT CS? - this is a logic decision to determine whether
or not a further command signal has been
generated.
- 32 - DET NXT CS - the next (second) command signal produced by
25 the keyboard device 6 is fetched and stored in
the first register in place of the previous
command signal.

(In the present instance, this second command signal will be the ASCII code for the letter 'B': i.e. 1000010).

- 30 33 - RD BY.AP - the letter code in the next byte position
(BY = BY2) is read and stored in the second
register in place of the letter code for the
letter 'O').

(In this instance, this letter code will be the ASCII code for
35 the letter 'A': i.e. 1000001).

34 - NXT CS = BY? - the contents of the first and second registers are compared with each other.

(In the present instance this comparison will fail because 1000010 = 1000001).

5 35 - AP = AP+1 - the address pointer is incremented by '1' to indicate the next memory location in the current group (GP15).

(In the present instance, the byte position BY2 of this next memory location also contains the letter code for the letter 'A' (for the word OAK) so that when step 34 is repeated the comparison will again fail. When the comparison in step 34 is satisfied, step 27 is returned to and the next sub-list of words is now available for display. In the present example, this occurs with the address pointer 18 set to the memory location containing the word 'OBDURATE' (address code 3AA0) so that the sub-list comprises the words 'OBDURATE' to 'OBOE').

Thereafter, the byte pointer is incremented by '1' (to BY3) and the next command signal (ASCII code for the letter 'V') is compared with the letter codes stored in the third byte positions of successive memory locations. Comparison for step 34 occurs when the memory location with the address code 3ACD is reached (for the word 'OBVERSE') and the sub-list available for display now comprises the words 'OBVERSE' to 'OCULAR' in the foregoing sub-list. The word chosen for display (i.e. 'OBVIOUS') is included in this sub-list so that there may be no further command signals from the keyboard 6, in which event step 31 goes to step 36 - STP which is an instruction to exit the flow chart.

However, since the word 'OBVIOUS' is the third in the sub-list it appears near the top of the display screen of the display device. The displayed sub-list can be scrolled to move the word chosen to a more central position in the displayed sub-list under the control of the graphics tablet 7. The decision to scroll or not is represented in the flow chart by step 37 - SCROLL? When a 'scroll' command signal is detected, step 38 - AP-OFS is entered. This step 38 determines the address pointer offset (up or down by

one address) depending on the requested scroll direction to provide a new memory location starting from which a new sub-list of words is to be read. Step 27 - LD DSP is then returned to, to write the dot information for the new sub-list of words into the display
5 memory. When scrolling is not, or no longer required, step 39 - CNT? is entered. This step 39 provides a decision for continuing with or interrupting the overall routine. If the decision is made to interrupt, then step 36 is entered, which as aforesaid is an instruction to exit the flow chart. If the routine is to be
10 continued with, then step 30 is returned to, to look at the next byte location.

Steps 38, 27 and 28 accomplish a 'hard scroll' by which the new displayed sub-lists are changed (i.e. moved up or down) one word at a time. For smoother scrolling, the display memory would
15 be required to store at least one extra word at each end of the words which form a displayed sub-list, with the scan lines initially receiving display data progressively for one or the other of these extra words, depending on the scrolling direction.

As regards the detection of the address codes which identify
20 the memory location groups GP1 to GP26, all the memory locations could be addressed sequentially and the code in the first byte BY1 read out from each successively and compared with a received 'select' command signal code. Alternatively, the address pointer could be programmed to cause only the first location in each of the
25 memory location to be addressed. This would dispense with the code table.

As a further modification, an address offset step 40 - AD.OFF (as shown in dotted lines in Figure 5) can be provided so that the 24 memory locations from which data is read out for display by
30 step 27 do not run from the actual memory location that contains a letter code which satisfies the comparison step 25. The effect of this, when the offset results in a given preceding memory location being the first of the 24 memory locations read out, is to cause a word which is selected by a sequence of 'select' command signal
35 codes which has a code for each letter of the word, to be

positioned centrally without any scrolling being necessary.

The diagrams 6a to 6f represent the dot information in the display memory (5 - Figure 1) during different instants that the display sub-routine (DSPM - Figure 5) is being performed. In each of these diagrams, the block 34 represents one information bit level of the display memory, the lined rectangle 35 represents the information bits for the sub-list of words to be displayed, and the arrow 36 represents the addressing scan position from which the bit information is being read out for a scan line at any instant. In diagram 6a, the bit information for the entire word sub-list is present in the display memory, and the addressing scan position is about one-third way down the display memory. In diagram 6b, the addressing scan position is about two-thirds way down the display memory and about one-third of the dot information which has already been used for this field scan has been erased. In diagram 6c, the addressing scan position has reached the end of the scan and all of the dot information behind it has been erased. To achieve this, the rate at which the dot information is erased row-by-row is made fast enough to "catch-up" with the line scan, but not before the end of the field scan. In diagram 6d, the addressing scan position is now, in effect, in the vertical blanking period between successive field scans. During this period, new dot information is being stored in the display memory for the next field scan. In diagram 6e, the next field scan has started and the display memory is about half-filled with new dot information. In diagram 6f, the display memory is completely filled with new dot information and the next field scan is nearly completed. Thereafter the cycle represented by diagrams 6a to 6f is repeated to refresh the display memory with new dot information every other field scan. It has been calculated that it takes approximately 25 ms. to both erase old dot information from and write new dot information into the display memory. For a field scan period of 20 ms. and vertical blanking interval of 4 ms, approximately 30 ms. is available for the erase/write operation, by starting this operation when the addressing scan position is about one-third way down the display memory as shown in diagram 6a.

CLAIMS:-

1. An electronic information apparatus comprising: a background memory wherein a list of items of information is stored, each item comprising at least two data parts represented by
5 respective coded bytes and each item being stored at a respective memory location; a display memory for storing data representing a sub-list of said items as selected from the background memory for display on a display screen of a display device; a display
10 generator for producing video signals in accordance with the data stored in the display memory for driving the display device; and a data processor and an associated user interface device for controlling the selection and display of sub-lists of selected
15 items each comprising a number of characters; in which apparatus the items of the list are organised in groups, the items of a same group having a same first coded byte and being stored at respective memory locations which form a group; and the data processor is operable:-

- to generate one at a time a plurality of group addresses which identify respectively an initial memory location in each
20 memory location group,
- to generate one at a time subsequent addresses of memory locations whose group has been identified by the relevant group address,
- to set an address pointer to the memory location whose address
25 is generated last,
- to set a byte pointer to indicate in turn the byte positions of a memory location to which the address pointer is set, and
- to compare a command signal received from the interface device
30 and representing a character with the coded byte in the byte position indicated by the byte pointer in the memory location to which the address pointer is currently set.

2. An electronic information apparatus as claimed in Claim 1, where the data processor is further operable:-

- (a) under control of an initialisation signal generated by
35 the user interface device to set the address pointer to

the first memory location the first memory location group
and to set the byte pointer to indicate the first byte
position,

- 5 (b) under control of a first command signal indicating the
first data part of an item to be selected, to read the
coded byte from the byte position indicated by the byte
pointer and to supply the read code byte to a comparator,
- 10 (c) under control of an 'error' signal, generated as a result
of non-correspondence between the read coded byte and the
first command signal, to set the address pointer to the
first memory location of a subsequent memory location
group, to read the coded byte indicated by the byte
pointer and to supply the read coded byte to the
comparator,
- 15 (d) under control of a 'result' signal, generated as a result
of correspondence between the read coded byte and the
first command signal, to set the byte pointer to indicate
the second byte position, and to set the address pointer
to a subsequent memory location in the same memory
20 location group,
- (e) under control of a further command signal indicating a
subsequent data part of the item to be selected, to read
the coded byte from the byte position indicated by the
byte pointer and to supply the read coded byte to the
25 comparator;
- (f) under control of an 'error' signal, generated as a result
of non-correspondence between the read coded byte and the
further command signal, to set the address pointer to a
subsequent memory location in the same memory location
30 group, to read the coded byte at the byte position
indicated by the byte pointer, and to supply the read
byte to the comparator ;
- 35 (g) under control of a 'result' signal, generated as a result
of correspondence between the read coded byte and the
further command signal, to set the byte pointer to
indicate a subsequent byte position.

3. An electronic information apparatus as claimed in Claim 1 or Claim 2, wherein said data processor is further operable under control of a 'result' signal, to load into the display memory the data for the item stored at the memory location indicated by the address pointer, along with the data for the items stored in a predetermined number of subsequent memory locations.

4. An electronic information apparatus as claimed in any preceding Claim, wherein said interface device comprises an ASCII keyboard device, the ASCII codes which are produced in response to key operation constituting said command signals.

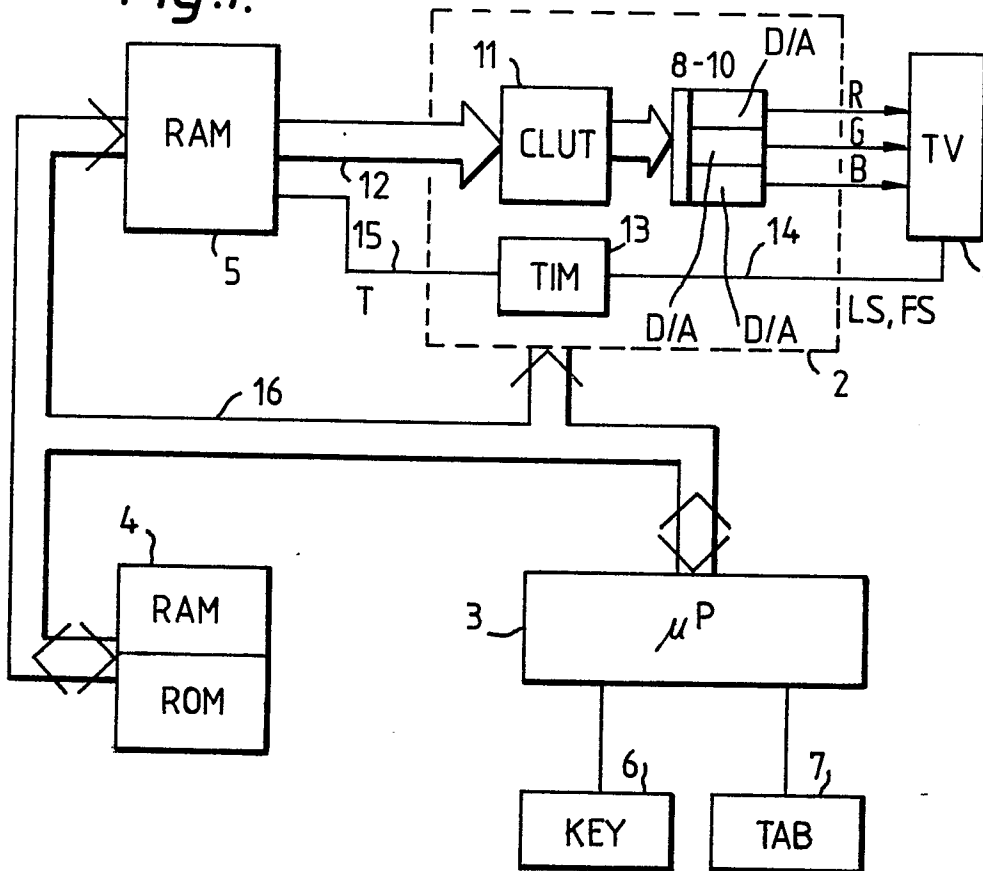
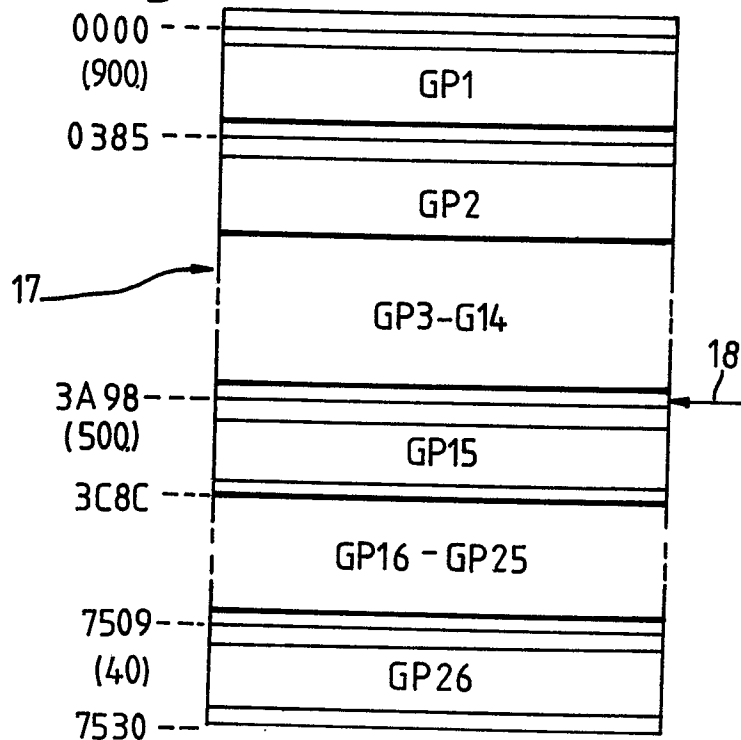
5. An electronic information apparatus as claimed in any preceding Claim, provided with scrolling means connected to said data processor and said display memory, said scrolling means being enabled under control of a 'result' signal.

6. An electronic information apparatus as claimed in Claim 5, wherein said scrolling means comprises a graphics tablet the pen of which provides successive indexing 'scroll' command signals in response to vertical movement (up or down) of the pen on the graphics tablet.

7. An electronic information apparatus as claimed in any preceding Claim, wherein said background memory comprises a program memory in which is stored control information, and a data memory in which said list of items is stored.

8. An electronic information apparatus as claimed in Claim 7, characterised in that data is written into said data memory from a mass memory which is physically separable from the apparatus.

9. An electronic information apparatus as claimed in Claim 8, characterised in that said mass memory comprises an optical record carrier in combination with a suitable read apparatus.

Fig.1.*Fig.3.*

b ₇		0	0	0	0	1	1	1	1				
b ₆		0	0	1	1	0	0	1	1				
b ₅		0	1	0	1	0	1	0	1				
b ₄													
b ₃													
b ₂													
b ₁													
0	0	0	0	NULL	DC ₀	b	0	@	P				
0	0	0	1	SOM	DC ₁	!	1	A	Q				
0	0	1	0	EOA	DC ₂	"	2	B	R				
0	0	1	1	EOM	DC ₃	*	3	C	S				
0	1	0	0	EOT	DC ₄ (STOP)	\$	4	D	T				
0	1	0	1	WRU	ERR	%	5	E	U				
0	1	1	0	RU	SYNC	&	6	F	V				
0	1	1	1	BELL	LEM	(APOS)	7	G	W				
1	0	0	0	FE ₀	S ₀	(	8	H	X				
1	0	0	1	HT SK	S ₁	)	9	I	Y				
1	0	1	0	LF	S ₂	*	:	J	Z				
1	0	1	1	VTAB	S ₃	+	;	K	[				
1	1	0	0	FF	S ₄	(COMMA)	<	L	\				
1	1	0	1	CR	S ₅	-	=	M	]				
1	1	1	0	SO	S ₆	.	>	N	↑				
1	1	1	1	SI	S ₇	/	?	O	←				

Fig. 4.

	b8	b1b8	b1b8	b1b8	b1																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																															
--	----	------	------	------	----	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

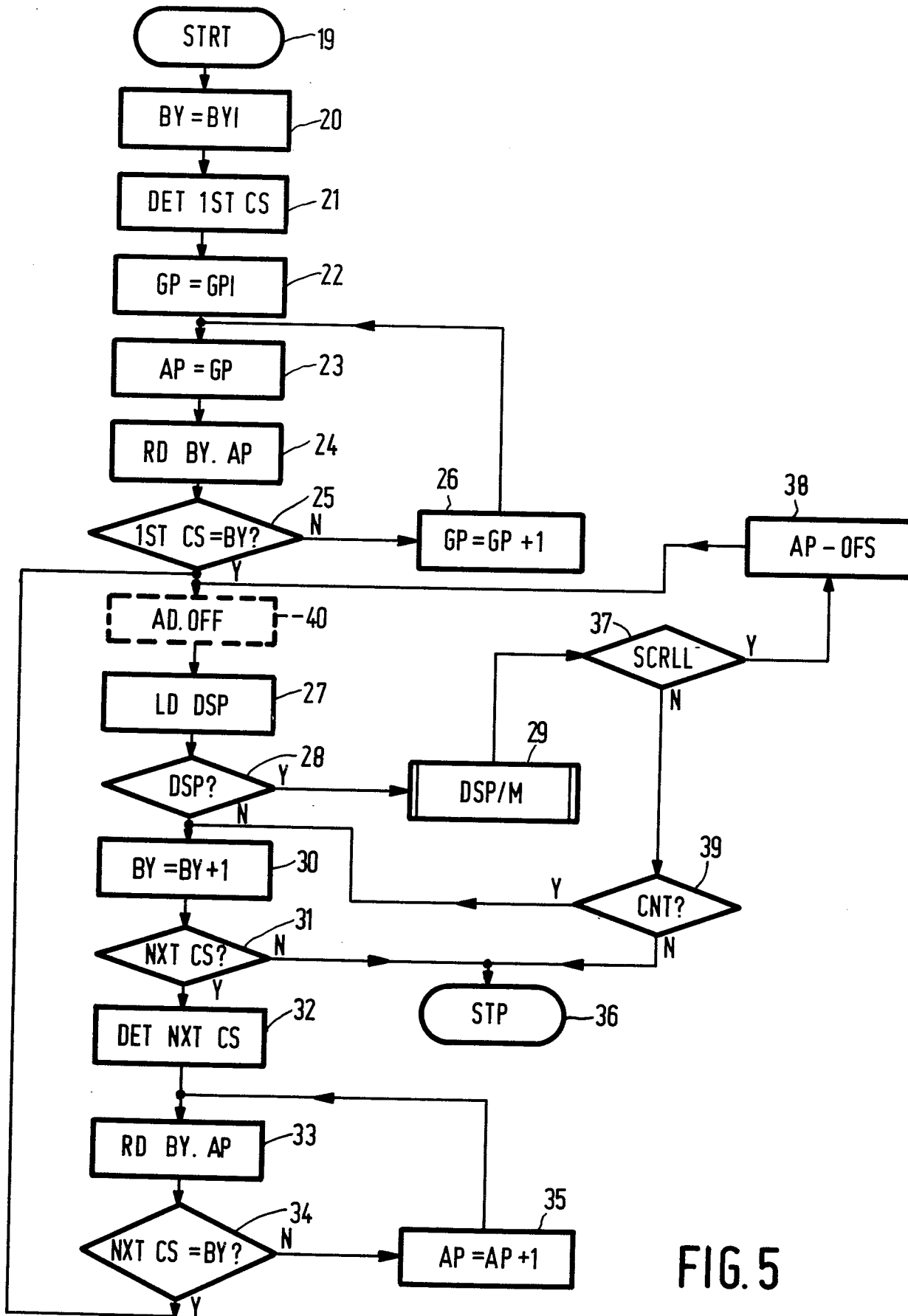


FIG. 5

Fig.6.