RUN LENGTH ENCODING OF VIDEO CHARACTERS

INTRODUCTION

Run length methods are used as a means of character encoding in many present day video character generators. A typical example of such a generator is the Aston 3.

The first part of this handout illustrates the basic principles of run length encoding, and it is shown that this system is generally a more efficient way of defining characters than dot matrix methods. The latter part explains how this method of encoding is employed in the Aston 3.

1.1 Principles of Run Length Encoding

This type of encoding requires each character to be analysed in a slice by slice basis, each slice usually corresponding to one television line in the video display.

Consider the letter 'A' for example, as shown in diagram 1.

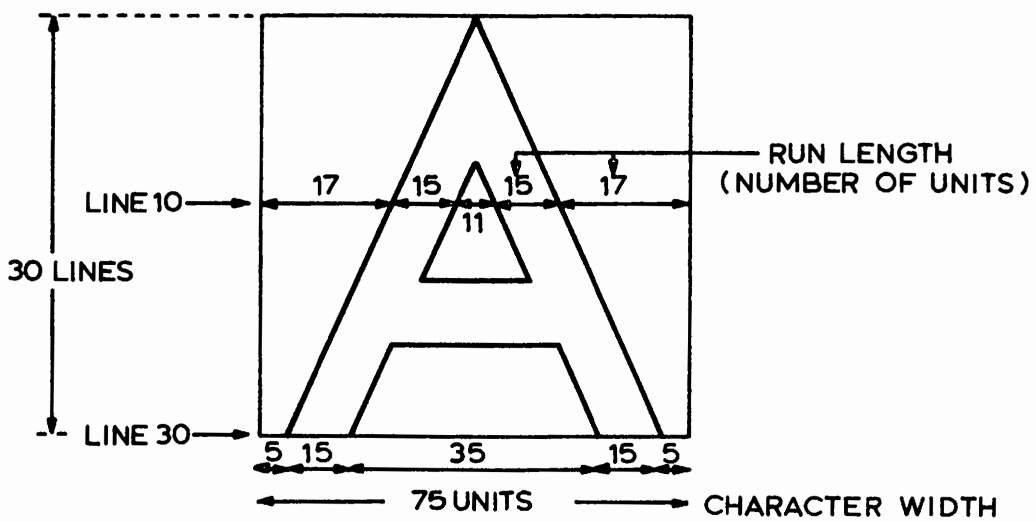


Diagram 1: Slice Analysis of Letter

RUN LENGTH ENCODING OF VIDEO CHARACTERS

The letter is contained within a box which, in this case, is 75 units wide and 30 television lines high. Looking at a horizontal slice through the letter we see that it contains consecutive elements, or 'run lengths' when the character is either present or not, i.e. white or black for a monochrome display respectively. Each run length is described by a number of units. Eg. Line 10 comprises of 17 units black, followed by 15 units white, 11 units black, 15 units white and finally 17 units black.

It is evident that the width of the letter on the display will be dependent on the width of each of the units. The unit width will also determine the horizontal definition of the displayed character - this will be dealt with in greater detail later in the handout.

Line 30 of the letter 'A' has 5 units of black before the start, and after the end of the letter in the horizontal direction. These elements effectively define the horizontal boundaries of the box that the character is contained within. The box width will be different for each character and ensures correct spacing between adjacent letters in the displayed text.

This is termed 'proportional spacing' and is shown in diagram 2.

It gives rise to a more aesthetically spaced text than non-proportional spacing common to dot matrix display systems.

RUN LENGTH ENCODING OF VIDEO CHARACTERS

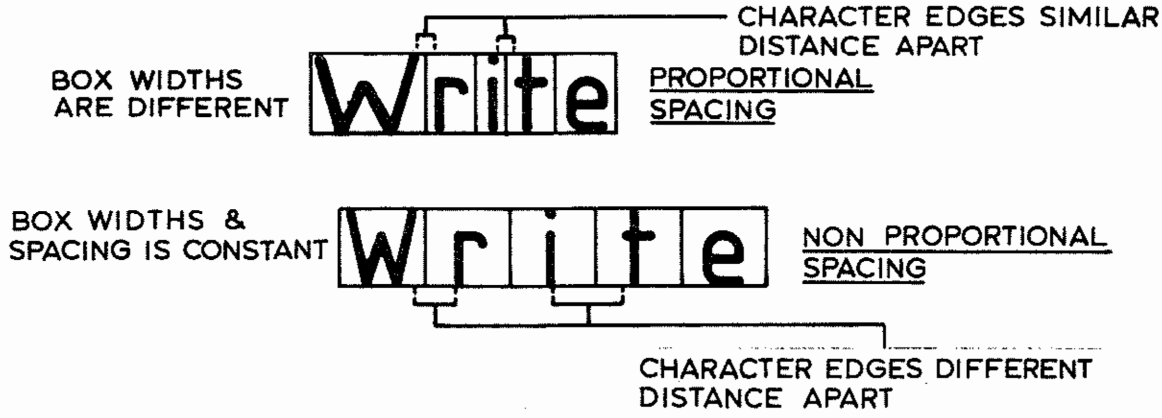
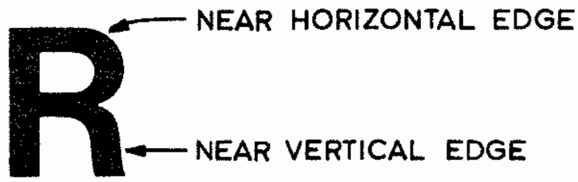


Diagram 2: Proportional and Non-proportional Spacing

As already stated the unit width affects the definition of the displayed character. Let us consider two cases in particular; near-vertical and near-horizontal edges. Such a character containing these would be the letter 'R' as shown below:-



These edges will not be displayed in a continuous manner on the screen, but will contain steps or 'jaggies' as they are called, being due to the line structure of the display and finite unit width.

Diagram 3 shows what these edges might look like on the screen. Edges for a choice of two different unit widths are shown. One unit width being half the other.

It is evident that smaller unit widths give rise to better quality near-vertical edges; but more bytes of data would be required to encode the character 'make up'

RUN LENGTH ENCODING OF VIDEO CHARACTERS

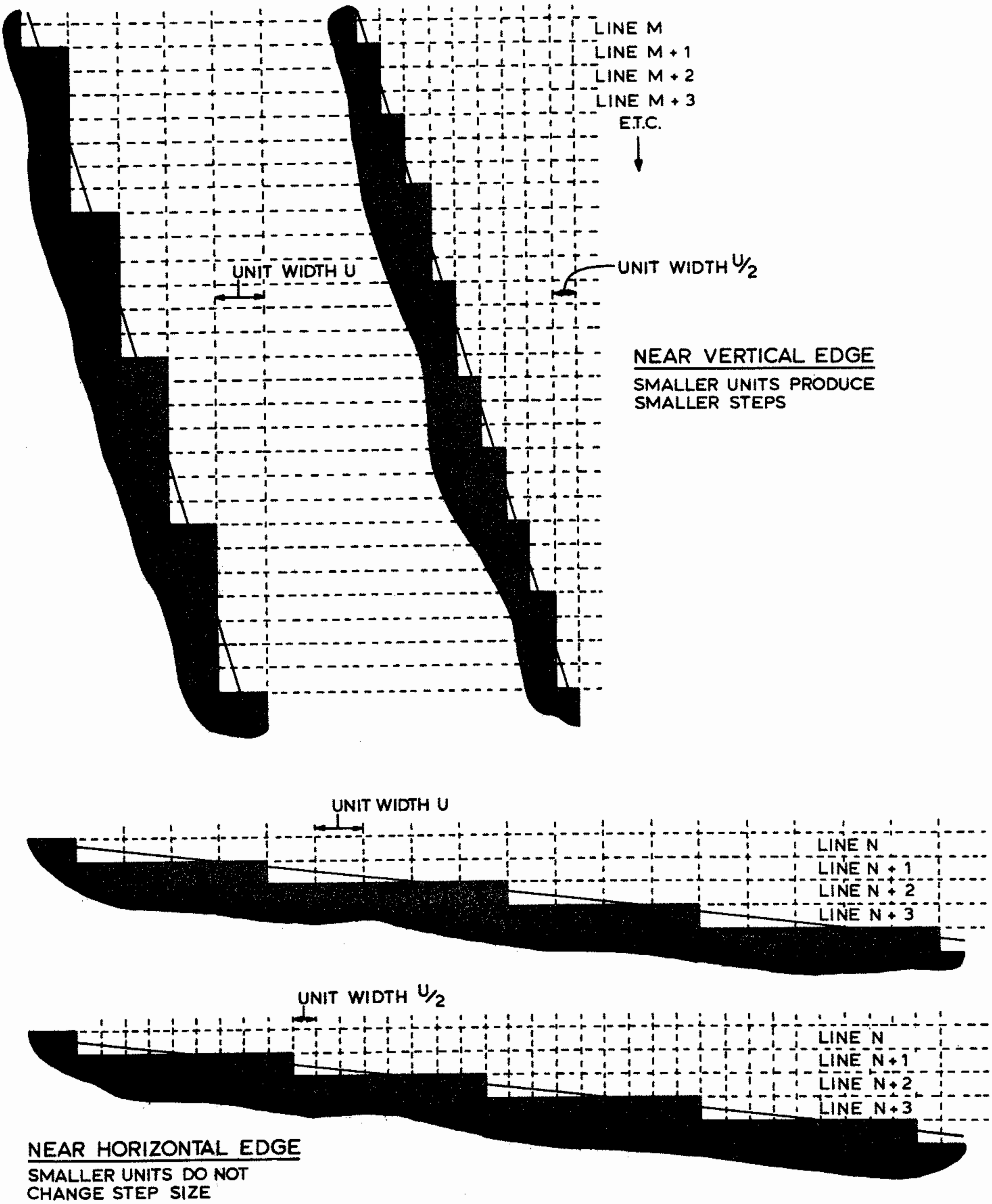


DIAGRAM 3. MAGNIFIED DISPLAY OF NEAR-VERTICAL AND NEAR-HORIZONTAL CHARACTER EDGES FOR TWO DIFFERENT UNIT WIDTHS

in this case. Also, the quality of near-horizontal edges is largely independent of the unit width, being determined mainly by the number of scanning lines in the display.

The 'smoothness' of slopes between these two extreme cases is obviously dependent on both the unit width and the line structure of the screen.

The Aston 3 uses a unit size of 31ns, this being a good compromise between character definition and the size of data required to encode them.

2.1 Encoding of Characters in the Aston 3

Each run length in the character is defined by an 8-bit word, known as 'Element Words'. One bit defines whether the run length is black or white, and another bit indicates if it is a 'Last in Line' element (more about this later!).

The other 6 bits designate the number of 31ns units in the run length. This is summarised in Table 1.

Each element word therefore can describe a maximum length of 64 units (64 x 31ns ≈ 2μs). If a run length greater than this is required two or more element words are used.

Bit No.	Description
1	2's Complement of the length of the element in number of 31ns units.
2	
3	
4	
5	
6	
7	L.I.L i.e. High for a 'last in line' element. Low if not 'last in line'.
8	$\bar{\omega}$ i.e. High for black element. Low for white element.

Table 1: Element Word Format

RUN LENGTH ENCODING OF VIDEO CHARACTERS

The elements must also be defined by a minimum of three units, this being due to speed limitations of the 'Character Processor'. This means that the finest vertical line that can be displayed is 93ns wide, which is approximately equal to the width of the lines of a correctly framed registration chart and is not a serious limitation.

2.2 'Last in Line' Elements

If the width of the character box is known, W, it is not necessary to encode information about the black elements shown by the shaded area in diagram 4. These are simply the difference between W and the sum of the other run lengths on any line. This enables a saving in bytes to describe each character.

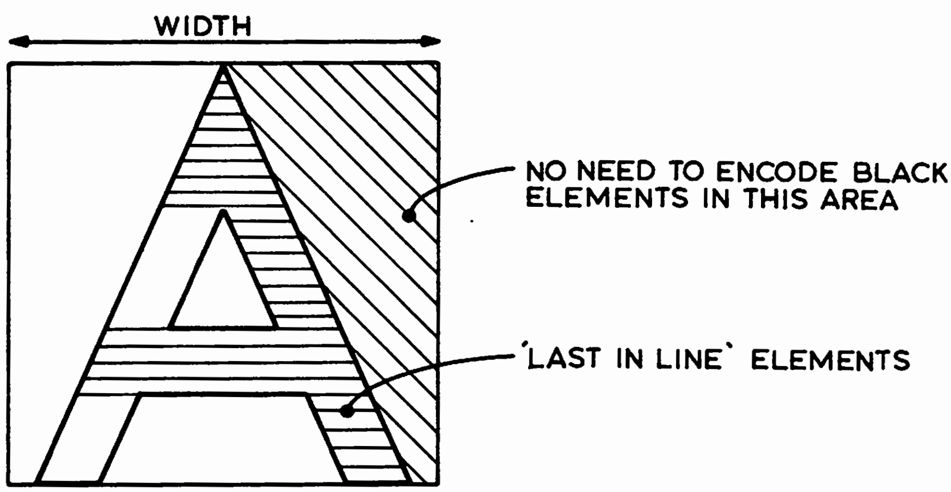


Diagram 4: 'Last in Line' Elements

The last run of white units in any character line are referred to as 'Last in Line' elements. These element words have bit 7 high as shown in table 1. After the end of a last in line element the generator will output black until the start of the next character on that television line.

Note that for every element word defining a run length bits 7 and 8 are never high together, since last in line elements are always white by definition.

For each character, a word describing the number of 31ns units across the character box width is also stored. This is known as a character width word and is 8 bits long.

2.3 'Blank Line Code'

If a particular television line of a character has no white elements in it a 'special' element word is stored instead of run length data. This is 11110000 and is known as a 'Blank Line Code'. It cannot be mistaken for an element word defining a run length since bits 7 and 8 are both high.

Such a letter containing a blank line code would be the letter 'i':-

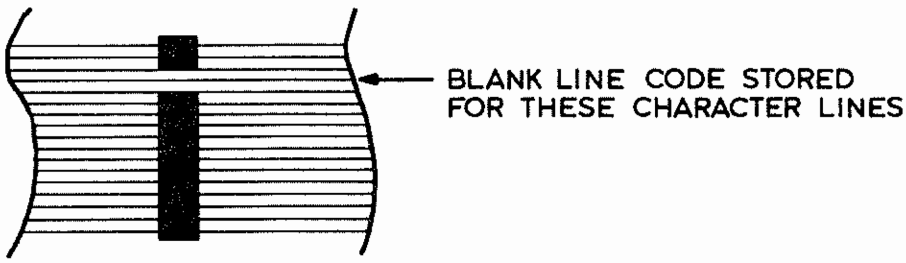


Diagram 5: Blank Line Code

2.4 'Font Height' and 'Start Count'

A whole set of characters is termed a 'Font' and for each font a single 6 bit byte of information is stored which defines the number of television lines required to construct the capital letters. This is called the 'Font Height'. What about lower case letters? Consider diagram 6.

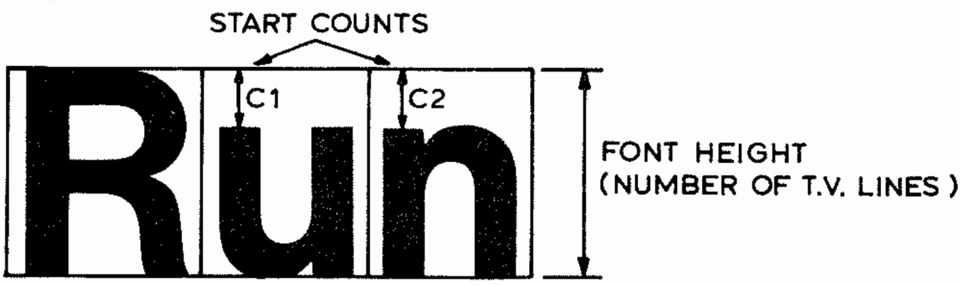


Diagram 6: Font Height and Start Counts

RUN LENGTHS ENCODING OF VIDEO CHARACTERS

In any font the lower case letters will generally be of different heights. Rather than store many blank line codes for the lines from the top of capital letters to the start of each lower case letter, a single word is stored for each character. This is called the 'Start Count' and defines the number of these lines with no character information in them. The 6 bit start count word will generally be different for each lower case character and zero for every capital.

2.5 'End of Character' word

At the bottom of each character another 'special' element word is added after the last white element. This is 11111000 and is known as an 'End of Character' word.

Summary So Far

We have covered the principles involved in run length encoding and it has been shown how the storage of character width, font height and start count words can increase the efficiency of this type of encoding.

Most of the points covered are summarised by diagram 7. This text also includes a space and the parameters required to describe this are a font height and character width word only.

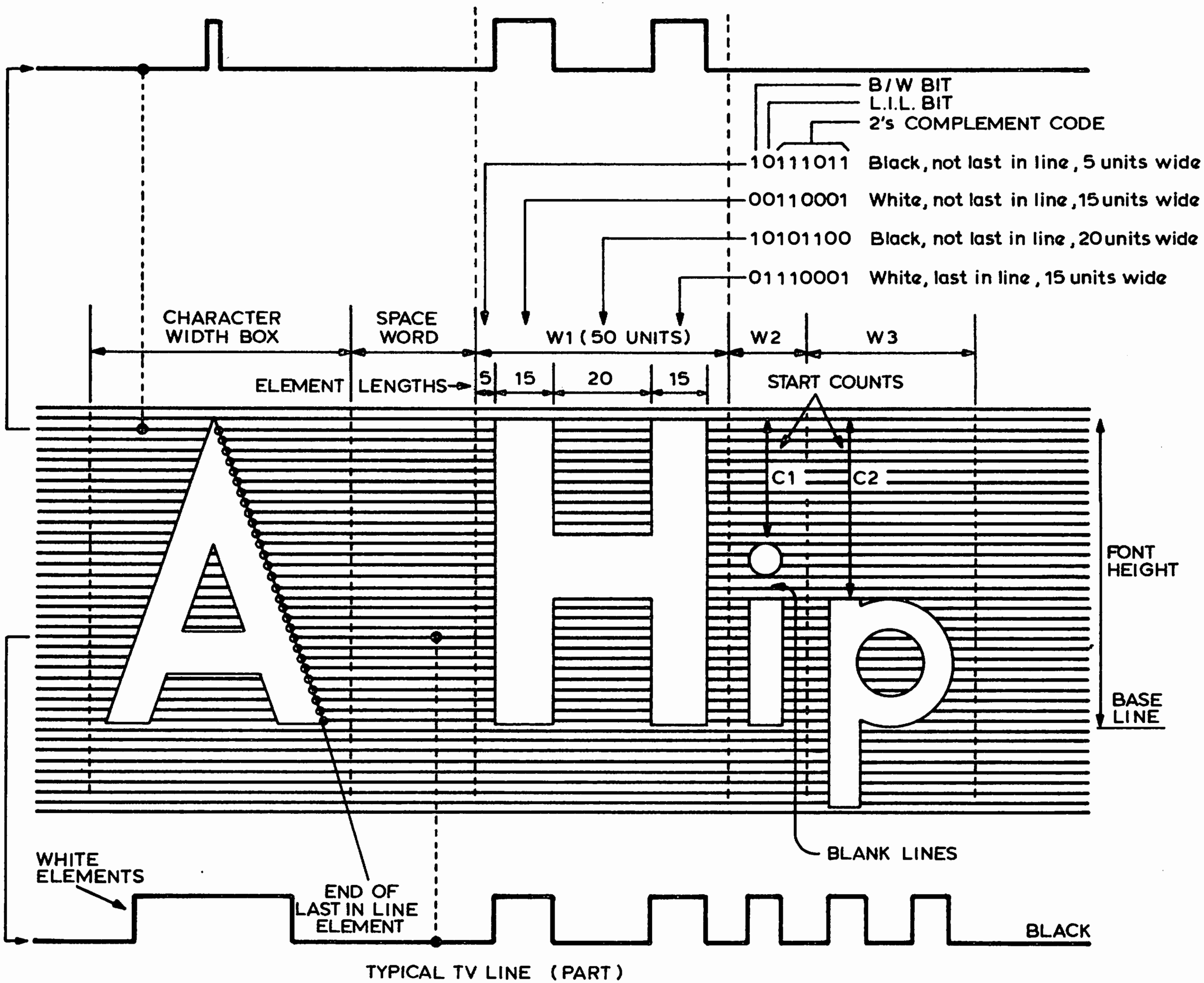


Diagram 7: Summary of Character Construction

RUN LENGTH ENCODING OF VIDEO CHARACTERS

3.1 How Efficient is Run Length Encoding?

Let us do a rough comparison of encoding efficiency between the run length system used by the Aston 3 and a dot matrix system capable of producing the same quality character edges (i.e. one having the same unit size). The letter 'H' used for the comparison is shown in diagram 8.

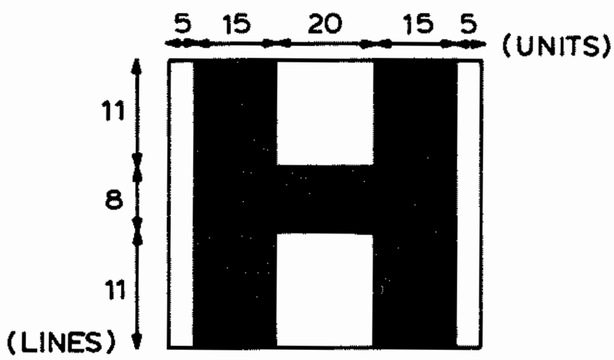


Diagram 8: 'H'

Number of bits of storage for dot matrix system:	Number of bits required for run length system:

For this particular character the run length encoding uses about half the storage of the dot matrix system. Generally speaking, the simpler the character shape, the more efficient the run length method becomes, thus the less memory is required. The use of start counts increases the efficiency even further for lower case letters.

The storage needed for each character in a dot matrix system will generally be the same for a given character box size.

4.1 Character Storage in the Aston 3

The Aston 3 stores the run length data in a memory block called 'Font Memory' which comprises of eight 64K x 1 bit R.A.M.s, i.e. 524288 bits of storage. Different run length information is stored for odd and even fields respectively and the memory is thus split into two halves.

The Aston is designed to hold up to 4 fonts in this memory at any one time (each font comprising of up to 95 characters plus a space), the amount of font memory used up depending on the font styles. These fonts are chosen from a 'library' of different character styles stored on floppy disk and are loaded into font memory using the disk drive on the Aston 3.

4.2 Font Formats on Disc

Font data is stored on the disc as 16 bit words. Two 8 bit element words are thus combined, elements describing odd fields occupying the lower order byte, and those for even fields the higher order byte.

A table showing the font formats on the disc is shown in Table 2.

RUN LENGTH ENCODING OF VIDEO CHARACTERS

Word	Bit															
	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Element	$\overline{W}$	LIL	← Length →						W	LIL	← Length →					
	← Field 1 →								← Field 2 →							
Font header (--C0)	0	0	← Font Height →						1	1	0	0	0	0	0	0
Character header (--E0)	0	← ASCII code →							1	1	1	0	0	0	0	0
Character parameter	0	0	← Start Count (C) →						← Char. width →							
Character parameter, space	0	0	← Font Height →						← Char. width →							
End of font (00D0)	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0

Typical font format

Word No.	Type of word	
1	Font header	
2	Character header (space)	
3	Character parameter (space)	
4	Character header	Repeated for each character
5	Character parameter	
6	Element	
7	Element	
8	Element	
↓	↓	
M	Element	
M + 1	Character header	
M + 2	Character parameter	
M + 3	Element	
↓	↓	
N	Element	
N + 1	End of font	

Table 2: Font Formats on Disc

RUN LENGTH ENCODING OF VIDEO CHARACTERS

5.1 Generation of Character Video

The character 'video' is formed on the 'Counters' board in the Aston 3 using the element words drawn out of font memory. The circuit arrangement is show in a simplified form in diagram 9 and its method of operation can be explained as follows:-

It consists of two 6 bit counters configured as a 'ring' and these 'element counters' are loaded with the two's complement of the run lengths. Only one of the two counters will be running at any time, the other being either reloaded or waiting for a trigger pulse on its 'start' input. Associated with each counter is a latch which holds the status of the run length i.e. black or white. This is obtained from bit 7 of the element word.

While the counter is running the output from its associated AND gate will be high or low depending on whether the run length is white or black respectively.

The 'carry' output from a counter goes high after a time equal to the loaded run length has elapsed.

Eg. A run length of 5 units is required:

Two's complement of 5 is	111011
After 1 clock pulse the count is	111100
After 2 clock pulses the count is	111101
After 3 clock pulses the count is	111110
After 4 clock pulses the count is	111111
After 5 clock pulses the count is	000000 + CARRY

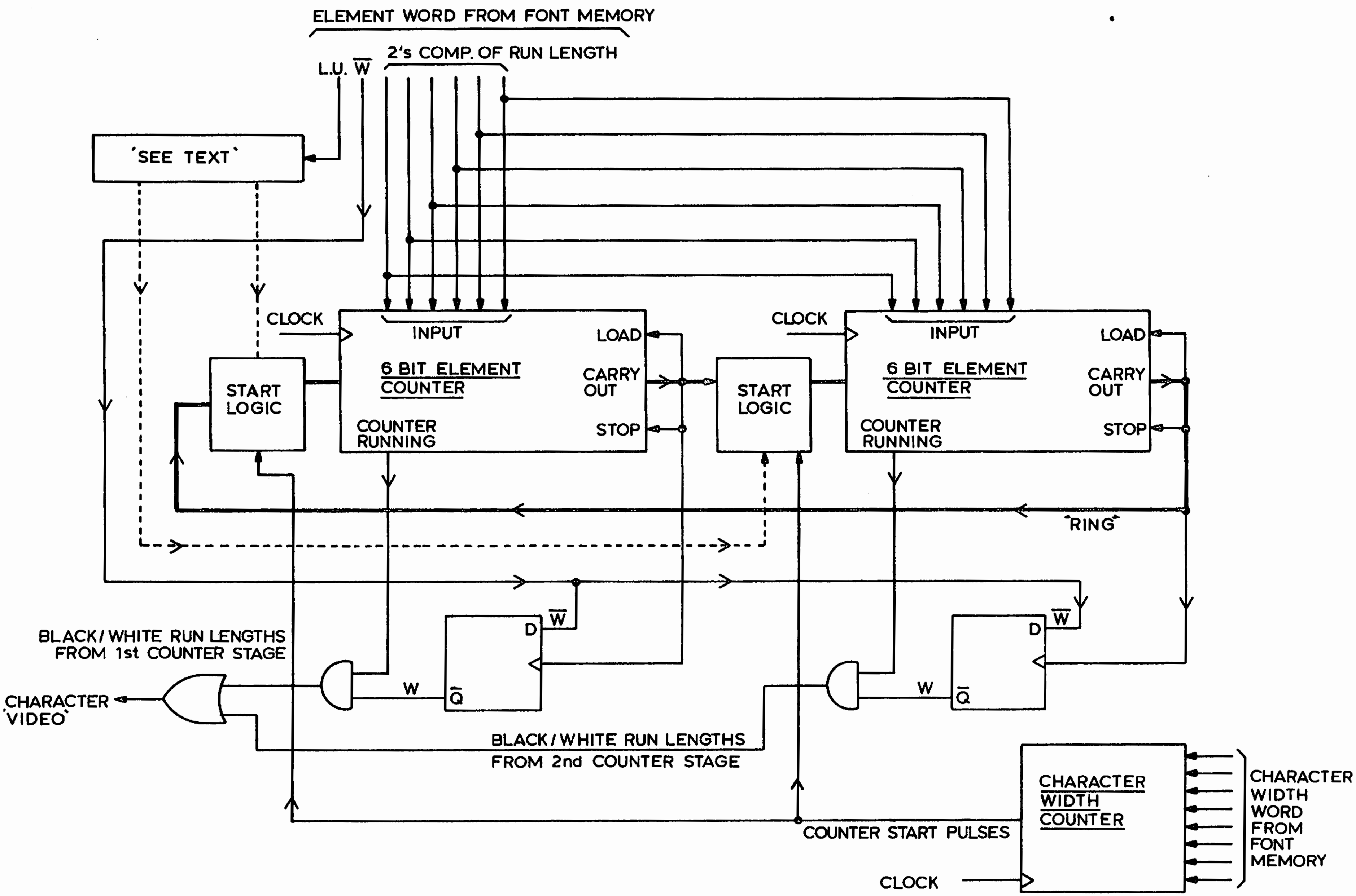


DIAGRAM 9 GENERATION OF CHARACTER VIDEO FROM ELEMENT WORDS

When the 'carry out' occurs the following takes place:-

1. The counter is stopped.
2. The next counter in the ring is started.
3. The counter is loaded with a new run length and its status is clocked into the latch.

Thus the next counter in the ring now produces the character video while the reloading takes place. This process repeats itself and a string of black and white run lengths is produced from the ORed outputs of the counter stages.

Only the basic system has been described and the above process is modified by character width words and the presence of last in line elements:-

5.2 Last in Line Elements

After the output of a last in line element the counter ring is broken by inhibiting the next counter from starting. The element counters will thus lay dormant (producing black at the OR gate output) until the process is restarted by a signal from the 'Character Width Counter'.

5.3 Character Width Counter

This is a separate 8 bit counter and is loaded with the character width words. When clocked the counter produces pulses at the start of each new character in a television line. These pulses are used to restart the element counter ring, triggering the counter containing the first run length of the new character. This is shown in diagram 10.

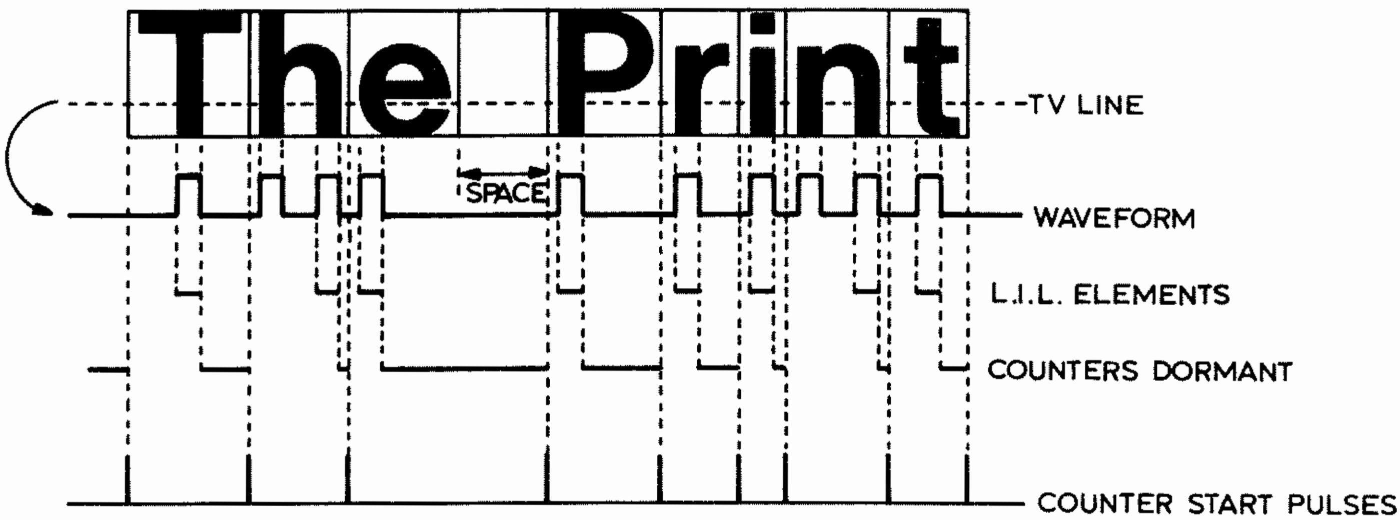


Diagram 10: Run Length Generation For One Television Line

Note that for a space the counters stay dormant until another character width word produces a start pulse.

The Aston actually uses three 6 bit element counters connected in a ring, but the method of operation is as described for two counters. The extra one is necessary because for very short run lengths the loading process takes a significant amount of time, and if two counters only were used the loading process may not have been completed by time the counter was required to start.
The three counter system works on the assumption that two very short run lengths will not occur consecutively.

Summary

Run length encoding is generally a very efficient way of defining characters for electronic generation. These can be produced fairly easily by use of a counter ring system and this is employed in many present day video character generators.

J.S. Calvert/VC/VIDCHARS
4th February 1985