



(which uses an extra byte but is fast). This demonstrates that there's no right way to build a computer, there's only the individual designer's way. It is also a good example of the sorts of things that computer designers must take into consideration. They know that they have a fundamental choice between designing a machine that is slow but inexpensive, or one that is fast but expensive. Similarly, when writing a BASIC program on machines with limited memory (the

unexpanded Vic-20 and ZX81 are good examples), you have to decide how to trade speed of execution against efficiency of memory usage.

Finally, notice that in the Text Area there will be a line-start or line-end marker for each line of the BASIC program. On the BBC Micro each line starts with a byte containing 13 (ASCII for Carriage Return), whereas this ends a Spectrum line. The Commodore BASIC line ends with a zero-byte (ASCII for " ").

Each machine has its own variations in the way it stores a line of BASIC Text. Consider the individual techniques shown for the lines of text given below

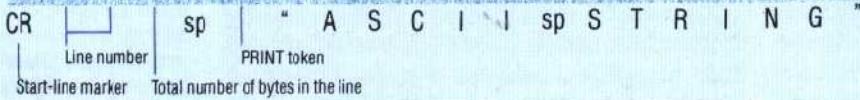
## How BASIC Programs Are Stored

### BBC Micro

```
200 PRINT"ASCII STRING"
300 A=1963.2:B=INT(A):A$="C"
```

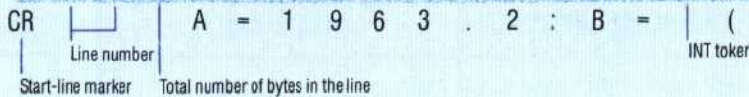
Contents of the memory bytes

13	0	200	20	32	241	34	65	83	67	73	73	32	83	84	82	73	78	71	34
----	---	-----	----	----	-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



In this example, the BASIC keywords are replaced by single-byte tokens. All the other characters are stored as ASCII codes. The start-line marker, line-number and line-length bytes are added by the Operating System

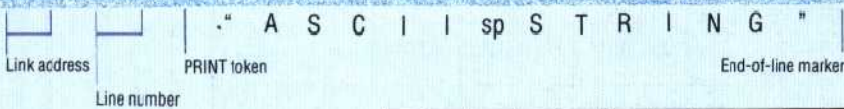
13	1	44	27	65	61	49	57	54	51	46	50	58	66	61	168	40	65	41	58	65	36	61	34	67	34
----	---	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	----	----	----	----	----	----	----	----	----	----



### Commodore 64

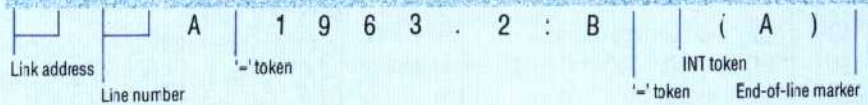
```
200 PRINT "ASCII STRING"
300 A=1963.2:B=INT (A)
```

240	9	200	0	153	34	65	83	67	73	73	32	83	84	82	73	78	71	34	0
-----	---	-----	---	-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	---



The Link address gives the address of the first byte of the next line. Note also that the Link address and Line number bytes are in the form offset byte followed by page byte

4	10	44	1	65	178	49	57	54	51	46	50	58	66	178	181	40	65	41	0
---	----	----	---	----	-----	----	----	----	----	----	----	----	----	-----	-----	----	----	----	---



### Spectrum

```
200 PRINT "ASCII STRING"
300 LET A=1963.2:LET B=INT A
```

0	200	16	0	245	34	65	83	67	73	73	32	83	84	82	73	78	71	34	13
---	-----	----	---	-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----



Notice that the length of the line is expressed in two bytes rather than one, so that program lines longer than 255 characters are possible. Furthermore, note that the numerical constant 1963.2 is stored first in ASCII codes, and then in a special binary format. This improves program execution speed

1	44	22	0	241	65	61	49	57	54	51	46	50	14	139	117	102	102	102	58	241	66	61	186	65	13
---	----	----	---	-----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	----	-----	----	----	-----	----	----

