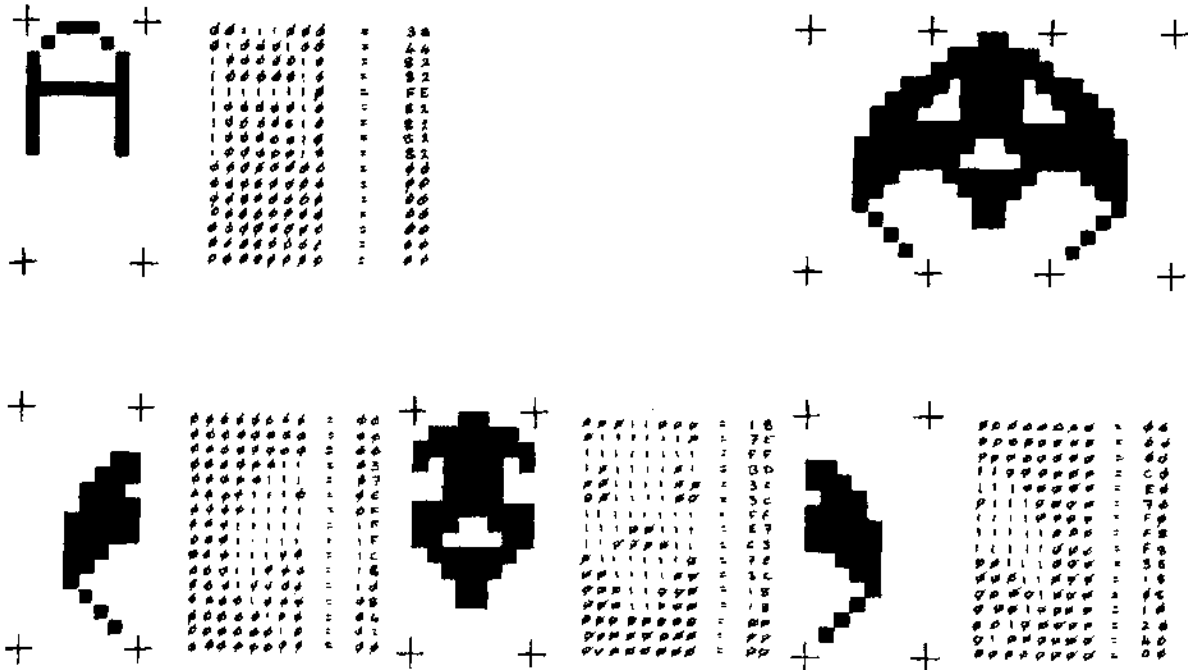```
18 7E FF BD 3C 3C FF E7
C3 7E BC 18 18 0F 00 00

00 00 00 C0 E0 70 F0 F8
F8 38 18 08 10 20 40 00
```

Each is a discrete character, which when displayed together on the
screen form a complete 'Invader'.



So having discovered how a character is built up, how is it
organised in the ROM ? The first thing to realise that a ROM is an
independant device, the first byte in the ROM is address 000H,
regardless of where the ROM may sit in a memory map. The first byte of
the first character starts at address 000H and the character takes up
16 (10H) bytes. The second character therefore starts at address 010H,
the third at 020H, etc., the last character starts at address 7F0H.
Simple really. Now another sneaky trick. Consider the alpha-numeric
character generator, notice that the character numbers are 00, 01, 02,
etc. Well relate that to the addresses in the ROM. Character 00 starts
at address 000H, character 01 starts at address 010H, 02 at 020H etc.
See the pattern ? The character number is the same as the address with
the least significant '0' missing. Not difficult. Now the graphics
characters start at character number 80. So all you have to do to find
your new character in your new graphics ROM is to start counting at 80
instead of 00.

The characters must first be programmed into RAM, and as the
graphics start at 80, a convenient place to start is an X800H memory
boundary, that way you can keep track of the character addresses
without mental arithmetic to add 80 to each character number. You don't
have to program all 128 characters at once, but leave all unprogrammed
bytes at FFH, that way you can 'over program' later. With all the
characters in RAM, make a tape of them, and then blow the EPROM. At
long last it can be said that Nascom owners are going dotty !!!!