Z80 made simple

THE KIDDIES' GUIDE TO Z80 ASSEMBLER PROGRAMMING. D. R. Hunt

Part: The First (and if doesn't go down too well, Part: The Last).

Funny numbers, counting with sixteen fingers, and all that.

We have had many appeals for a beginner's guide to Z80 assembler programming, and as no-one else volunteered, I thought I might have a go. After all, my qualifications for this task are impressive.

1) Three years ago I knew nothing about it.
2) It is arguable if I have learned much in the meantime.
3) I don't mind making myself look an idiot in print (in the eyes of of the enlightened).

All this means is that I'm not too clever at it to baffle the reader, and that I'm new enough at it remember the difficulties I had at first.

So here goes: The first thing to learn is HEXadecimal, the numbering system used when writing machine code. HEX numbers are usually indicated by numbers either being prefixed '#', or suffixed 'H'. Now to use HEX effectively, really means that you should grow three more fingers on each hand, as this is a little difficult for most normal people, an explanation will have to suffice.

Binary and HEXadecimal

Most of us count in units of 1, 10, 100, etc. For historical reasons we need not discuss, this counting in 10s business (known as 'base 10' counting or Decimal) is so much second nature that counting in any other form may well seem ludicrous. However, there are other things on this earth which do use different systems, and computers figure largely among them. Right at the heart of it, the computer uses the Binary system, and all it knows is that a number represented as 'no volts' will be interpreted as a '0', whilst a number represented by 'some volts' will be interpreted as '1'. From this it can be seen that the computer counts in twos (known as 'base 2' counting or Binary). In the same way that we count in 1s, 10s, 100s, etc, the computer counts in 1s, 10s, 100s etc. Unfortunately, although the numbers look the same, they are in fact different. As each unit is the base number raised to next power (mathematically speaking) they actually mean:

We think	Computers think
10 to the power 0 = 1	2 to the power 0 = 1 (= 1 Decimal)
10 to the power 1 = 10	2 to the power 1 = 10 (= 2 Decimal)
10 to the power 2 = 100	2 to the power 2 = 100 (= 4 Decimal)
10 to the power 3 = 1000	2 to the power 3 = 1000 (= 8 Decimal)

etc.

So that the number fifteen (Decimal) is expressed as: