David Robinson – Talk at Science Museum 2009

I joined EMI in October 1955, having just completed my two years' National Service. They took me on as an engineer, on the strength of a degree in mathematics, an interest in anything that could do arithmetic, a knowledge of Ohm's Law, and a copy of the Royal Signals Handbook that I had filched from the army. I had always wanted to work on computers, ever since I first heard about the earliest developments in the USA and at NPL back in the late 1940's while I was still at school. I had a family connection with EMI, so I suppose that's how I got the first interview.

At that time EMI had a contract to develop a computer to perform payroll calculations for British Motor Corporation (as it was in those days), and in order to test their basic concepts and circuitry they started by building a "Pilot" machine, to be followed immediately afterwards by the real thing. The technology was based on the use of valves – thousands of them – the voltages used (+ and – 300) were lethal and the heat dissipated was awesome.

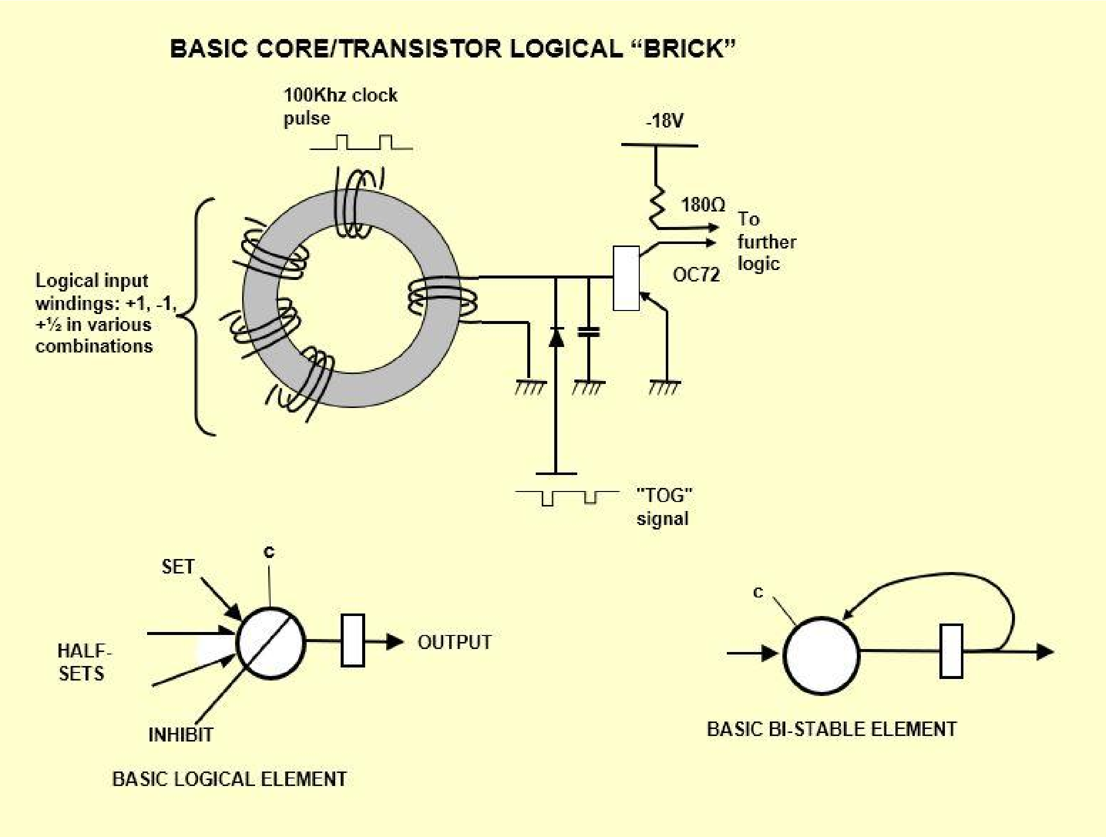
Anyway, they gave me a drawing board and a T-square, set me up in the corner of an office, and told me to design the necessary logic so that the machine could multiply and divide. I have to say that these were great days – you didn't have to *know* anything before you were asked to do it! Nowadays one would need a PhD in Computer science before you are let in through the door.

The BMC payroll machine used quarter-inch tape decks to hold carry-forward information, and the engineer in charge of this part of the project was Godfrey Hounsfield. Godfrey decided from the start that he was going to engineer this section using transistors, which at that time hadn't been around for very long and were viewed with some suspicion by the older and more senior members of the team.

Most people will have heard of Godfrey – Sir Godfrey Hounsfield FRS and Nobel Prizewinner, as he later became as a result of his developing the use of computerised tomography for brain scanning. Many people owe their lives to his work including my own wife. However, when I knew him this was all in the future (his future not mine) and we lovingly just knew him as "H".

Then EMI management decided to proceed with the design of a new computer based entirely on the use of transistors. The logical initial design thinking was done by Bob Froggatt and myself, and Godfrey ("H") became the project leader or chief engineer. We decided that a combination of ferrite ring and transistor offered interesting possibilities as a logical building brick for the new computer, and we decided on a parallel architecture (the BMC payroll machine was serial). Our objective was to keep things small and simple. In fact, to start with and privately, Froggatt and I called our brainchild "OXO" to reflect its small size and compactness. Needless to say, as the design progressed the size got bigger and bigger, and the finished machine, officially called the EMIDEC 1100, occupied a whole room, just as all other computers did in those days.

Let's look at the basic logical building brick.



The ferrite ring, which I shall refer to as a "core" (because it's fewer syllables and time is short), is the same as was used for data storage in machines of this generation. It is about 2mm across (0.08") and can be magnetised in either direction, clockwise or anticlockwise as we look at the picture, and the state of magnetisation remains until it is changed by current through one of the windings. We can talk about the core being "set" or "unset", depending upon which way it is magnetised.

It is an essential feature of this sort of ferrite core that the magnetism is permanent, that is to say the core will remain magnetised, one way or the other, when the current through the windings ceases. Also, to reverse the direction of magnetisation it needs a certain minimum amount of current through the windings. With less than this minimum amount of current, the direction of magnetisation won't change at all.

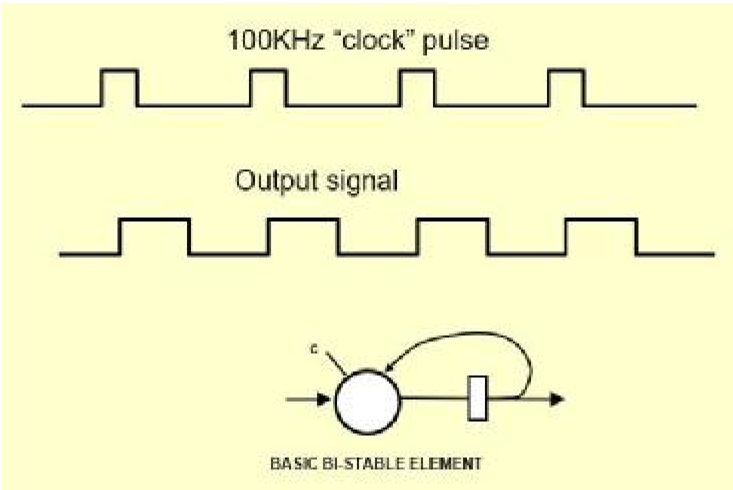
You can see that this can lead to a basic logical element: if half-strength current flows though two windings at the same time the core will be set, but current through just one winding will not set the core (actually we halve the number of turns on the winding, rather than halving the current, but the effect is the same).

Of course you cannot tell whether a given core is set or unset just by looking at it! The only way is to pass current through another winding such as to unset the core. If it was set, then a pulse of current is induced in the output winding. If the core was already unset, then because there is then no change in the direction of magnetisation, no output pulse will be generated. The signal from the output winding is then amplified by a transistor and the output from that goes on to feed further similar logical elements

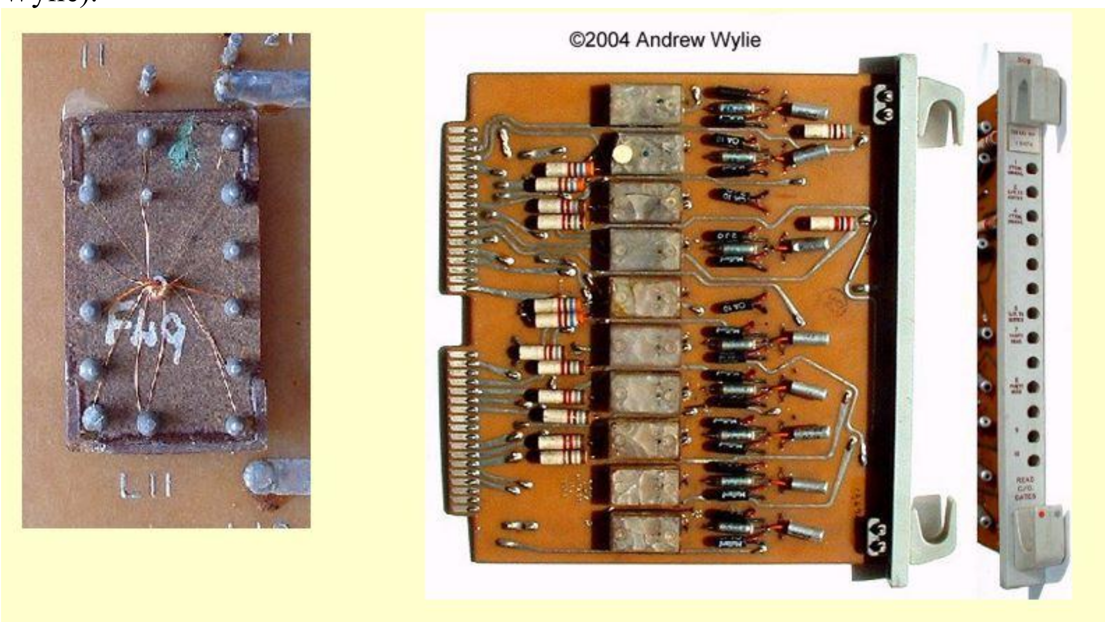
In practice each logical element is driven by a continuous 100KHz "clock" pulse.

My slide also shows an element arranged to act as a two-state device or "Trigger" as we used to call them. If the core is initially set, then with the action of the clock pulse it will continue to re-set itself after every clock pulse. If it was initially unset, then it will remain so.

Next slide shows the same trigger with waveforms - transistor output is lengthened partly by capacitor and partly by "hole storage" - a consequence of over-driving the base of the OC72 transistor.



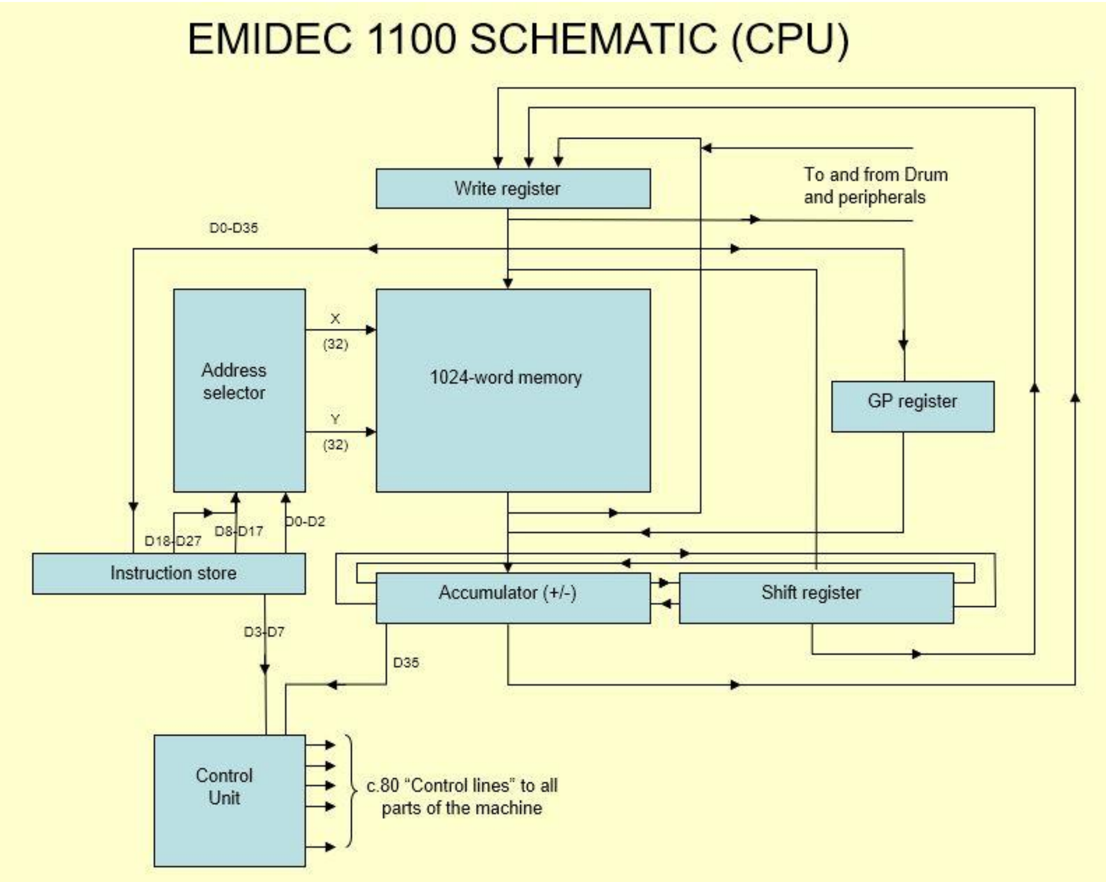
Next slide shows a wound core and a typical logic circuit board (Thanks to Andrew Wylie).



At this point you might be tempted to say "Hey, but this is a pretty slow old logical element, taking a whole clock pulse for the simplest operation: anything wanting complicated logic will take an age". This is of course true, but an awful lot of the logical operations in a computer are not that time-critical - card reader or printer control for example. And after all we were coming from a serial machine background where each word took 36 clock pulses to trundle by!

But there is of course one operation in a parallel machine that is time-critical and that is the question of the propagation of carries in the adder. For this we departed from our standard approach and let each core in the carry chain provide the clock signal for the next. This way we achieved about a carry transfer rate of about 180nsec per stage, so that a complete end-to-end carry propagation could be fitted within a standard 10µsec clock pulse interval.

So having looked at the basic circuitry, let's look at the architecture of the machine itself. This slide just shows the CPU. I have omitted the drum storage and the card reader and printer. To get this picture right I had to pay a visit to the Science Museum library at Swindon, where they still hold the original 1100 manuals. I was amazed by how much I'd forgotten! There was a lot more to the 1100 that I had initially remembered.



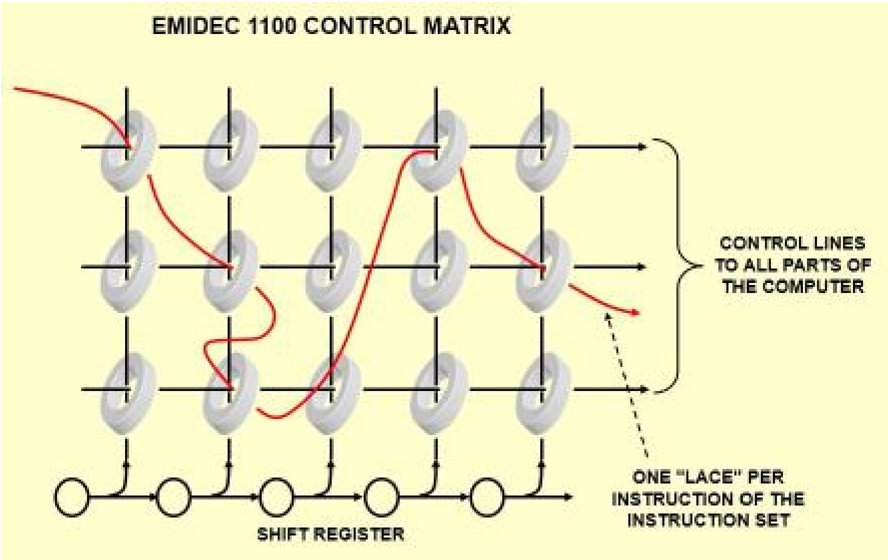
Memory
Arithmetic unit or accumulator(+/-)
Shift register

All items driven by what we called control lines: a single pulse of current down the line made something happen. There were about 80 control lines altogether.

eg: read next instruction
add output from memory to accumulator
shift left
write back to memory

etc
etc

So now let's look at the control unit for the machine. This is responsible for issuing the correct pulses along the right control lines at the right time.



5-bit function - 32 basic instructions - 32 laces

actually 2 additional bits giving up to 128 laces.

7-bit lace address are also control lines

overflow bit also drives one of these two additional bits.

Mechanical structure of matrix