Topic 8

Architectures and Organisation

Consider the past and you shall know the future. (Proverb of Chinese origin)

Contents

Learning Objectives

- **Appreciation that computers do not have to be digital, sequential and electronic**
- **Awareness of the existence of a theoretical discipline of computer science**
- **Familiarity with computer architectures of the past**
- **Appreciation of computer architectures of the present**
- **Awareness of the possibilities for computer architectures of the future**
- **Experience in thinking about the development of computer technology**

8.1 The essence of computing

So far in this unit we have been concerned with the software that runs on computers and how it is produced. We shall now take a look at the hardware but before doing so we should heed the fact that we have been able to ignore the hardware thus far.

It is all too easy to make the mistake of thinking of computers only in terms of the digital, sequential, electronic machines we use today. Whilst the early machines which you have been looking into were much less powerful than modern computers, they do show that the science of computing has a separate existence to the hardware that is used to implement it. Certainly, the hardware of the day imposes practical restrictions on what is possible and which cannot be ignored, but might there be some idealised theoretical device, set free from these hardware limitations, about which issues of computation could be discussed and analysed unfettered by the physical mechanisms available at a given time?

Indeed there is and there is also a growing corpus of theoretical results in computer science which are totally independent of hardware issues. Nor is this a new field, for it has its origins back in 1936 when Alan Turing developed a theoretical device which has since become known as the Universal Turing Machine (Turing 1936). Using this abstract artefact it is possible to investigate the really big questions in computer science such as what can be computed and what cannot – ever.

Information Technology might be a child of digital, sequential, electronic computers but it will outlive them and there are new hardware avenues opening up which we must be ready to embrace.

8.2 Yesterday's computers

You have already investigated or attended presentations on a number of historical computing devices. We shall briefly fill in some of the gaps here.

8.2.1 The mechanical age

Probably the most successful of the mechanical calculators was De Colmar's *Arithmometer* which was built to perform insurance calculations in about 1820. Versions of this device continued to be produced well into the 20th century.

The idea of a difference engine was first presented by Johann Müller in 1784, a German master builder. It is not clear whether Charles Babbage was aware of Müller's design but he was aware of the work of Alfred Deacon, another British designer of difference engines working in the early 19th century, and this may have informed Babbage's work.

Charles Babbage actually designed three difference engines in addition to the Analytical Engine. Most people are unaware of the earliest one, which this author has elsewhere dubbed *Difference Engine № 0* (Taylor 1992), but it is significant in being the only machine that

Babbage actually built. It was completed in 1822 and demonstrated to the Astronomical Society in London.

At the same time as Babbage was working on his designs, a father and son team from Sweden, Georg and Edvard Scheutz, were building machines based on simplified designs of Babbage's Difference Engine N$^{o.}$ 1. The Scheutzes produced three difference engines in total - one prototype (1843) and two others which were sold to the Dudley Observatory in Albany, USA (1853) and the General Register Office in London (1859).

Other developers of difference engines include another Swede Martin Wiberg in 1860 and the American Barnard Grant in 1876 (Swade 1991).

8.2.2 The electro-mechanical age

The 20th century saw the advent of general purpose computing devices. The key ingredient in a truly general purpose computer is the ability to change the course of a series of calculations depending on the data being processed. Babbage's Analytical Engine would have been able to do this but only to a very limited degree.

The importance of switching devices in the development of computing machinery has meant that the history of computers has sometimes been called the history of "switchology".

8.2.2.1 Relays

<!-- IU
A photograph of a relay would be nice here.
-->

Konrad Zuse's Z1-Z4 machines (1934-45) and the Harvard Mark I (due to Howard Aiken in 1944) used electro-magnetic relays as their switches. This continued dependence on mechanical components meant that the machines remained large and unreliable (as well as rather noisy!).

8.2.2.2 Valves

<!-- IU
A photograph of a valve would be nice here.
-->

At Bletchley Park in 1943, Colossus (due to Flowers, Broadhurst and Chandler) became the first computer to use thermionic valves. In 1946 ENIAC (due to Mauchly and Eckert) became the first totally electronic computer. It inspired the design of EDVAC which incorporated the idea of storing the program as data within the computer and organising the central processing unit in a particular way (von Neumann 1945). The stored program computer, or **von Neumann architecture**, was born and led to the development of the Manchester Mark I (due to Kilburn and Williams), EDSAC (due to Wilkes), the UNIVAC 1 (Mauchly and Eckert) and nearly all modern-day computers. For this reason, these machines are called **first generation** computers.

8.2.3 The electronic age

Valves too were unreliable devices and would bring computers crashing to a halt every few minutes but the next development in switchology was just around the corner.

8.2.3.1 Transistors

<!-- IU
A photograph of a transistor would be nice here.
-->

In 1947 Bardeen, Brattain and Shockley of Bell Labs invented the transistor. This device which was smaller, cheaper, faster and more reliable than the valve ushered in the **second generation** of computers.

8.2.3.2 Integrated circuits

<!-- IU
A photograph of an early IC would be nice here.
-->

In 1958 Robert Noyce of Fairchild Semiconductor and Jack Kilby of Texas Instruments independently developed the integrated circuit, a semiconductor chip with a number of transistors packed onto it. The switch had just become even smaller, faster and cheaper. The integrated circuit, or silicon chip, became the **third generation** of computers. Noyce went on to become a co-founder of Intel Corporation.

8.3 Today’s computers

One further development was necessary to bring us to the familiar computer architecture of today. This was the microprocessor.

8.3.1 Microprocessors

<!-- IU
A photograph of the Intel 4004 chip would be nice here. Any chance?
-->

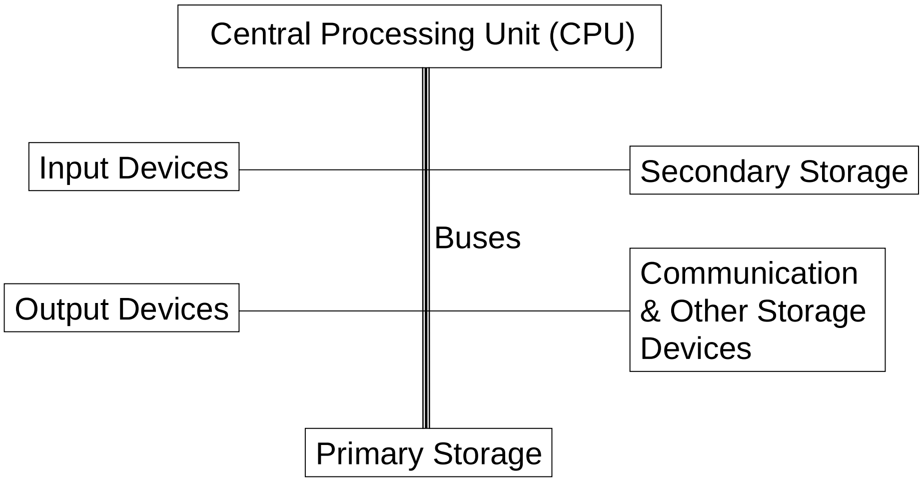
In 1971 the Intel 4004 was announced. It was a complete computer on a single silicon chip. The microprocessor is often referred to as the **fourth generation** of computers but it should be noted that it is considerably more than just another piece of “switchology”.

8.3.1.1 Personal computers

<!-- IU
A photograph of a modern day motherboard would be nice here.
-->

In 1975 the Altair 8800, the first microcomputer, was launched. This was the first of the machines which we now call personal computers or PCs.

A typical PC consists of a Central Processing Unit (CPU) and assorted input, output and storage devices.



The CPU has a von Neumann architecture and consists of a Control Unit, an Arithmetic and Logic Unit (ALU) and various Registers. The Control Unit can load values into the Registers for processing by the ALU and also use the values stored in the Registers to address memory locations.

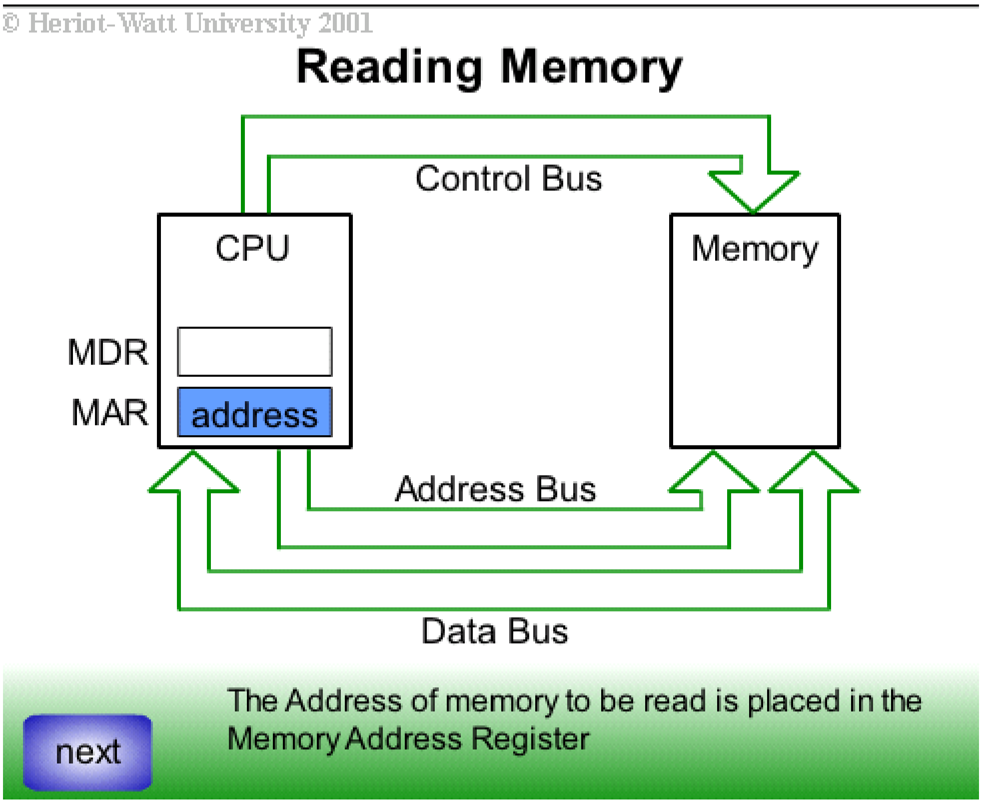
Input devices are used to enter information and include such things as keyboards, mice, joysticks, tablets and touch-sensitive screens. Output devices are monitors, printers, speakers and removable storage media.

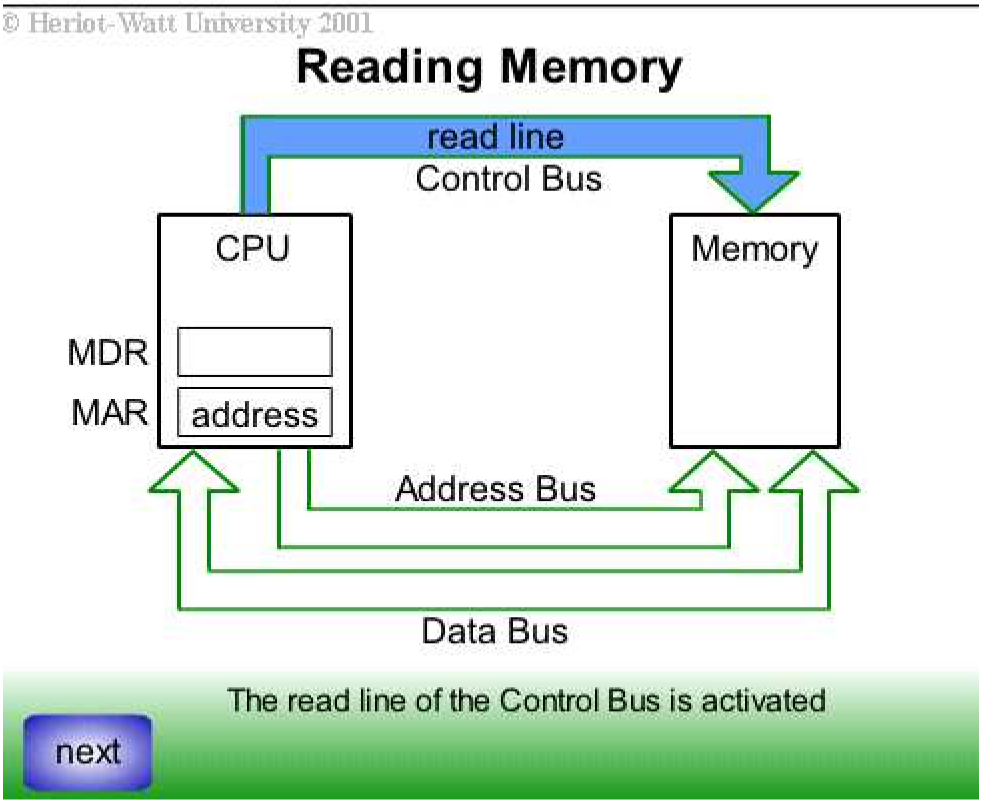
Primary storage is Random Access Memory (RAM) which is volatile and requires a power source to maintain what is held. Secondary storage is non-volatile and does not need power to retain information. Hard disks are the most common form of secondary storage. Other storage devices include external disk drives, DVDs, etc.

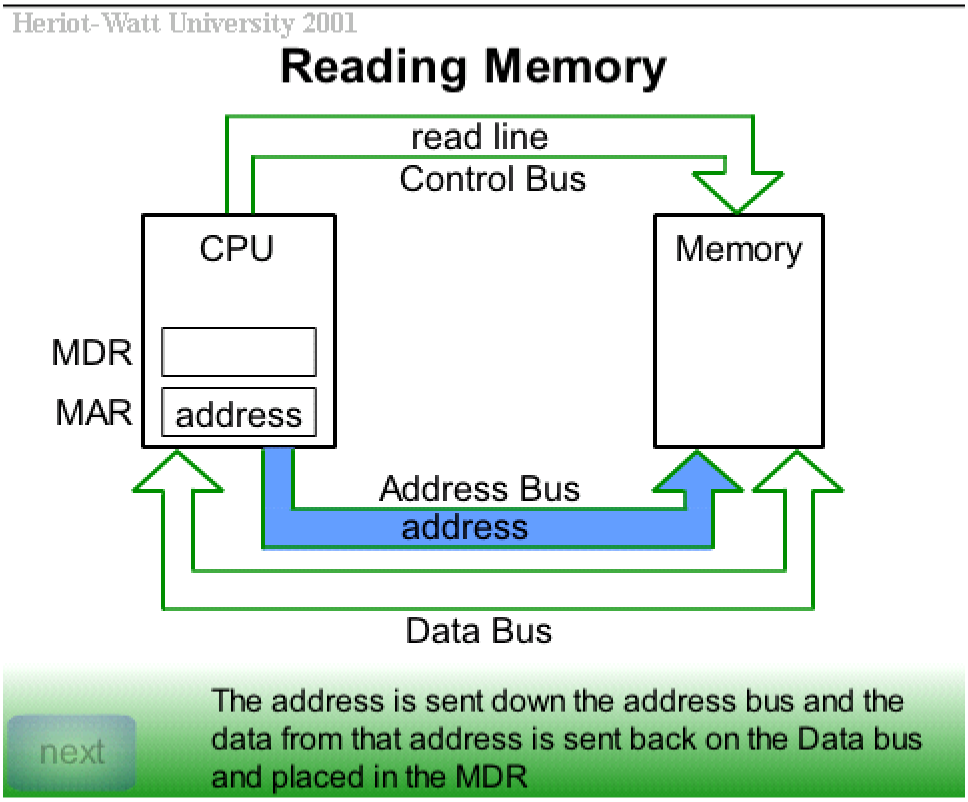
Communication devices cover things like network communication ports and modems. All of these components are linked within the PC by means of buses which can transmit data at very fast rates.

The following ?? SEQUENCE/ANIMATION ?? demonstrates a CPU using two registers (the Memory Address Register and the Memory Data Register) to fetch a data item from a particular address in primary memory for processing.

<!--IU
I have taken the following diagrams from Rick Dewar, another IU author, but I have no idea about the copyright situation - are they IU productions? They look like they might be stills from an animation. An animation in one diagram would be very nice.
-->


© Heriot-Watt University 2001
Reading Memory
Control Bus
CPU
Memory
MDR
MAR
address
Address Bus
Data Bus
next
The Address of memory to be read is placed in the Memory Address Register


© Heriot-Watt University 2001
Reading Memory
read line
Control Bus
CPU
Memory
MDR
MAR
address
Address Bus
Data Bus
next
The read line of the Control Bus is activated



8.3.3 Parallel processors

<!-- IU
A photograph of the Gamma 60 and/or Illiac IV would be nice here. Any chance?
-->

The computers we have been looking at up to now have been sequential. They perform one instruction after another. Parallel processors perform many instructions simultaneously using computer architectures which contain multiple processors. Credit for the first computer which used parallelism should probably go to the Gamma 60 developed by the Compagnie des Machines Bull in 1958. The first machine with a highly parallel architecture was the Illiac IV conceived by Daniel Slotnick in 1965 and built by Burroughs in 1974. Parallel processing can take two main forms; SIMD and MIMD.

8.3.3.1 SIMD

<!-- IU
A set of animations would be nice here. In this one a row of processing elements above a row of data elements with each processing element doing something like adding 1 to its respective data element.
-->

Single Instruction Multiple Data (SIMD) machines apply the same instruction to many items of data simultaneously. Also known as array processors, they are particularly useful when the same operation is required to be applied to many values. For example, multiplying a collection of fractions by 100 to turn them into percentages.

8.3.3.2 MIMD

<!-- IU
A second animation with the same layout as above but each processing element doing a different something to it respective data element, maybe one adding 1, another adding 2, etc.
-->

Multiple Instruction Multiple Data (MIMD) machines also work on multiple data items but permit each one to have a different instruction applied to it. They are much more versatile than SIMD machines and allow computational tasks to be broken down into sub-tasks which, as long as they are independent of each other, can be carried out simultaneously by different processors.

8.3.3.3 Pipelining

<!-- IU
An third animation with the same layout as the second but with the data elements flowing along their row so they get processed by each processing element in turn.
-->

Pipelining maximises the speed at which the same sequence of instructions can be applied to many items of data. This sort of processing is quite common. Each processor is allocated one of the instructions to perform and the data is pipelined from one processor to another in a stream. In a MIMD computer with, say, 5 processors, the first data item is presented to the first processor and processed, no gain over a sequential machine so far. The result of this processing is then passed to the second processor which then performs its operation. Meanwhile, back at the first processor, and at the same time, the second data item is being processed in accordance with the first instruction. By the time the fifth data item has been presented to the first processor all of the processors are busy performing their own operation on a different data item in parallel.

8.4 Tomorrow’s computers

It is likely that the electronic computer will serve us well for some time to come. In 1965 Gordon Moore of Intel predicted that the number of transistors per integrated circuit would double every 18 months. He expected his prediction to hold for no more than a decade. Nearly 40 years on *Moore’s Law*, as it has come to be called, remains valid and chips running at Terahertz speeds have been produced.

However, other possible architectures for computer design are now appearing on the horizon. It is possible that switchology will move on from the electronic age into an opto-electronic, chemical or even biological age. Some of these developments will take us beyond the traditional von Neumann architecture into a world of analogue and parallel computers and some will even leave switchology itself in their wake.

8.4.1 Optical computers

One of the limitations of integrated circuits is that electrical connections cannot cross each other without short-circuiting, even in semiconductors. If the circuits were carrying coherent rays of light instead of electrical impulses this would not be a problem. This is one reason why there has been much research into the development of optical computers.

8.4.2 Nanocomputers

The development of a scanning tunnelling microscope by researchers at IBM has made it possible to see and also manipulate individual atoms. This has opened up the possibility of creating nanocomputers whose switches are on an atomic scale rather than the molecular scale of current semiconductor devices. See Drexler (1992) for coverage of the whole field of nanotechnology.

8.4.3 Biocomputers

Developments in biology and genetics are also providing many new ideas for biocomputing machinery in the future. Biological phenomena such as ligand bonding, enzymes and DNA have the potential to offer much more than just another form of switch. See Kaminuma and Matsumoto (1991) for a more detailed coverage of biocomputers.

8.4.4 Quantum computers

Finally, we should mention quantum computers. Quantum theory suggests that it might be possible to produce computers that work with *qubits*, or quantum bits, as opposed to normal bits (Deutsch 1985). These are bits which are, at one and the same time, both *0 and 1*, not *0 or 1*. They are in a superposition of the two states *0* and *1* and only when they are examined does the superposition collapse and the qubit become either a *0* or a *1*. This offers the possibility of carrying out many computations simultaneously (on all of the combinations of a number of superpositioned qubits for instance) and algorithms have already been developed for factorisation and searching.

8.5 End of topic test

<!-- IU
Please insert a multiple choice test here. The correct answers to each question are underlined.
-->

Q1. Modern day computers are NOT –
- a). Digital
- b). Electronic
- c). Mechanical
- d). Sequential

Q2. The Universal Turing Machine is –
- a). Analogue
- b). Electronic
- c). Optical
- d). Theoretical

Q3. The most successful mechanical calculator was probably –
- a). Arithmometer
- b). Multiplier Wheel
- c). Napier’s Bones
- d). Pascaline

Q4. The transistor was NOT invented by –
a). Bardeen
b). Brattain
c). Kilby
d). Shockley

Q5. In comparing valves to transistors, valves were –
a). Cheaper
b). Faster
c). Less reliable
d). Smaller

Q6. The microprocessor is often regarded as which generation of computer –
a). First
b). Fourth
c). Second
d). Third

Q7. A CPU uses what to store values –
a). ALU
b). Buses
c). Diskettes
d). Registers

Q8. MIMD machines process –
a). Atomic data
b). Multiple data
c). Quantum data
d). Single data

Q9. Biocomputers would NOT use–
a). DNA
b). Enzymes
c). Ligand bonds
d). Schrödinger's cat

Q10. A superposition would be found in a –
a). Nanocomputer
b). Quantum computer
c). SIMD machine
d). Transistor

8.5 Assigned task

1. Think about the future possibilities for computer architectures.

References

Deutsch, D., 1985, ‘Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer’, *Proceedings of the Royal Society of London A*, 400, pp. 97-117.

Drexler, K.E., 1992, *Nanosystems: Molecular Machinery, Manufacturing and Computation*. John Wiley & Son.

Kaminuma, T. & Matsumoto, G. (eds), 1991, *Biocomputers*, Chapman and Hall.

Swade, D., 1991, *Charles Babbage and his Calculating Engines*. Science Musem: London.

Taylor, N.K., 1992, ‘Charles Babbage’s Mini-Computer - Difference Engine N$^{o.}$ 0’, *IMA Bulletin*, **28** (6), pp. 112-114.

Turing, A.M., 1936, ‘On Computable Numbers, with an application to the Entscheidungsproblem’, *Proceedings of the London Mathematical Society*, Series 2, **42**, pp. 230-265.

von Neumann, J., 1945, ‘First Draft of a Report on the EDVAC’, Moore School of Engineering, University of Pennsylvania.