The Ubiquitous Chip or Solid State Electronics

I’m going to start by clarifying a few terms which I shall be using during this talk.

We are all familiar with the word “electronic” but how many know its precise meaning? In particular, what’s the difference between “electrical” and “electronic”?

“Electronic” items use electricity to carry information.

“Electrical” items use electricity as an energy source.

A modern washing machine is both electrical and electronic as it uses electricity to power the motor but it also has an in-built computer to control the various washing cycles. The manufacturers usually describe such a machine as “electronic” as that implies modern and sophisticated.

What is voltage? Voltage is electrical pressure. If the pressure is high enough, electricity can get through anything, the classical example being lightning. Air is normally an insulator but with a high enough voltage applied, it breaks down and becomes a conductor.

What is current? Current is much easier to understand because it’s the amount of electricity flowing past a given point each second.

What is a “signal”?

A signal is a stream of information although the stream can be very short, e.g. “yes” or “no”.

Now what about “digital” and “analogue”?

Digital signals consist of a combination of 1s and 0s, Ons and Offs, Yeses and Nos, Trues and Falses, Heads and Tails, etc. i.e. only two states are possible in a digital signal.

Analogue signals vary continuously. A simple example of an analogue signal is sound which consists of alternating compression and rarefaction of the air in a continuously changing way. Most electronic systems use analogue signals in at least one part of the system.

Why is digital considered superior to analogue?

The answer is noise. Noise is unwanted information.

Examples: background sound obscuring a conversation or music
“snow” on an analogue television broadcast in a poor signal area
radio interference.

If noise is added to an analogue signal, there is no way of distinguishing the noise from the wanted signal, but if noise is added to a digital signal, the underlying signal can usually be recovered and restored to what it should be.

Let’s take an everyday example of digital noise. I have a coin that is badly scratched. Despite the scratches, we can still easily distinguish “heads” from “tails” unless the damage is extremely severe. However, if we wanted to copy the queen’s head from the damaged coin, we’d have to guess at the bits where the damage has obscured part of the queen’s head. i.e. the analogue information has been corrupted.

The worst thing about digital signals is that when the noise becomes bad enough, it completely corrupts the digital signal. I guess some, if not all, of you will on occasion have noticed blockiness in a digital television picture which is far worse than analogue snow. However, even then there are ways to recover the corrupted signal which would be a topic in its own right for another session.

How can electricity be used to carry information? The simplest answer to that is Morse Code which was the first electricity based digital information system.

And now we come to the core of this talk – the silicon chip and how it handles information, and, in particular, digital information. The basic element of a silicon chip is the transistor.

What is a transistor? The word transistor is an amalgamation of two words – transfer and resistor. A transistor is a semiconductor device used to control the flow of electricity in electronic equipment. The first transistor was a concept developed by Julius Edgar Lillienfeld, an Austro-Hungarian Physicist on

22nd October 1925 but the technology didn’t exist to make one. The first working transistor was developed by a team at Bell Labs in 1948 comprising John Bardeen, William Shockley and Walter Brattain, and was made of Germanium, a close cousin of Silicon. All early transistors were made of Germanium but Silicon took over because it can run at higher temperatures. Note that Carbon, Silicon and Germanium are all Group 4 elements in the periodic table. I’ll cover this in a little more depth shortly.

Silicon, as distinct from silicone as used to enhance certain parts of the anatomy, is a hard crystalline element and a close cousin to diamond. (Silicone is a close cousin to rubber).

Here we need to go into the physics of the atom. All atoms consist of a nucleus consisting of positively charged particles called protons held together by uncharged particles called neutrons around which are one or more orbiting electrons. The number of electrons must match the number of protons for the atom to be overall uncharged.

Unlike the solar system, where each planet has its own orbit, in an atom several electrons share each orbit. However the orbits are not 2 dimensional like in the solar system and they are known as shells The atom is most stable when there are 8 electrons in the outermost shell or 2 if there’s only one shell very close to the nucleus – and so not enough room for 8.

Silicon has 4 electrons in its outer shell and so is unstable unless it can borrow or share electrons from adjacent atoms. This gives rise to the tetrahedral crystal structure of silicon where each atom has 4 adjacent atoms equally spaced in 3 dimensions around the central atom and each atom sharing 2 electrons with the central atom, one of its own and one from the central atom. In this way, each atom in the crystal has 8 electrons in its outer shell, 4 of its own (shared with its neighbours) and 4 shared from its immediate neighbours.

As it’s difficult to draw 3 dimensions, we’ll simplify things and use 2 dimensions.

Silicon is known as a semiconductor because it’s neither a good conductor nor a good insulator, but it’s conducting properties can be enhanced by adding impurities, in particular elements with 5 electrons in their outer shells or elements with 3 electrons in their outer shells.

Let’s look at an impurity with 5 electrons in its outer shell. If we try to fit it into the silicon crystal, we have a ‘spare’ electron hanging about. This electron can move through the crystal under the influence of an applied voltage and so the silicon becomes a conductor. Because it has some ‘spare’ electrons and electrons are negatively charged, this impurified silicon is known as n-type.

Let’s look at an impurity with 3 electrons in its outer shell. If we try to fit it into the silicon crystal, we have a ‘hole’ where an electron should be. This hole can move through the crystal under the influence of an applied voltage – actually an electron moves into the hole leaving a different hole behind - and so the silicon becomes a conductor. Because it is lacking in electrons and so has positive holes, this impurified silicon is known as p-type.

If we have a piece of n-type silicon in contact with a piece of p-type and we apply a voltage across the junction:

Current will not flow if we connect the positive voltage to the n-type and the negative battery terminal to the p-type because unlike charges attract so the positive voltage will attract all the negative electrons in the n-type away from the junction and the negative voltage will attract all the positive holes in the p-type away from the junction so we have a region in the middle which contains neither electrons nor holes and so there are no current carriers.

However, if we connect the positive voltage to the p-type and the negative voltage to the n-type, like charges repel so both electrons and holes will be repelled from the edges to the junction where they will cross over and neutralise each other. Meanwhile the voltage supply is providing more electrons to the n-type and more holes (by removing electrons) to the p-type and the current keeps flowing. This is the fundamental characteristic of semiconductors.

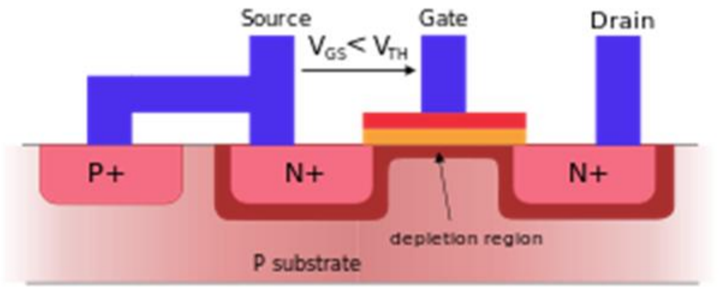
If there is no voltage across the junction, the electrons near the junction will cross over and fill the holes leaving a region with no current carriers called the Depletion Layer.

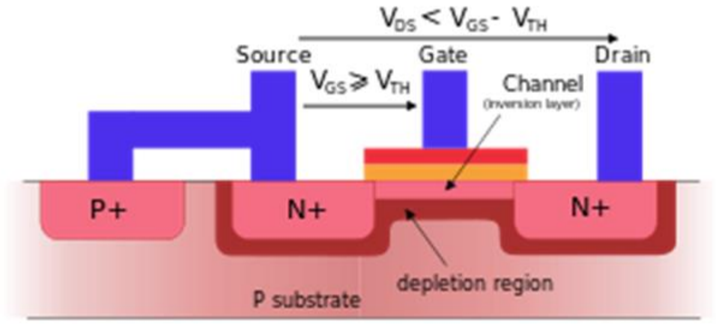
Now let’s look at how we can control the flow of current through a semiconductor:

There is a type of device called a MOSFET which stands for Metal Oxide Silicon Field Effect Transistor – you can see why we use the abbreviation. Its structure is a metal conductor on top of a Silicon Dioxide insulator on top of a piece of p-type silicon as you can see in the diagram.

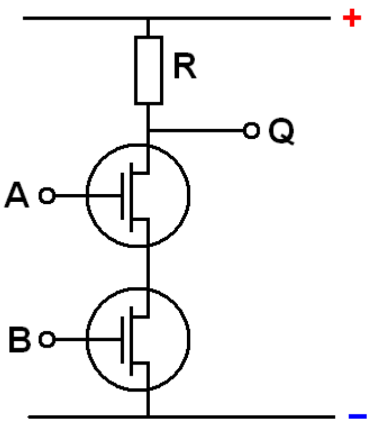
If a voltage is applied across the Source and Drain, no current will flow because of the depletion layers. Applying a positive voltage to the Gate will create a depletion layer under the gate also.

If we now increase the positive voltage on the Gate, not only will all holes be pushed away but some electrons will be attracted so that we effectively get a thin layer of n-type which means that we have n-type all the way from the source to the drain so current can flow from Source to Drain.

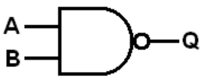


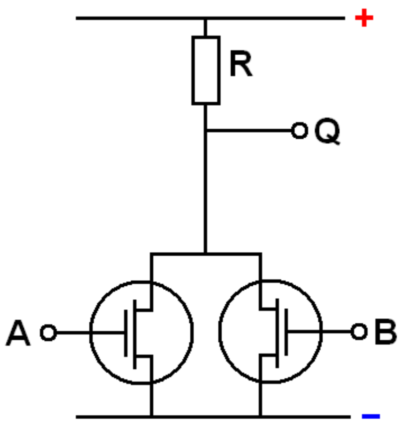


Now let's look at how this device can be used. If we consider a voltage on the gate high enough to create the inversion layer as a "1" and the absence of such a voltage as a "0", we can create various logic circuits:

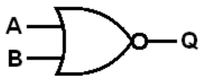


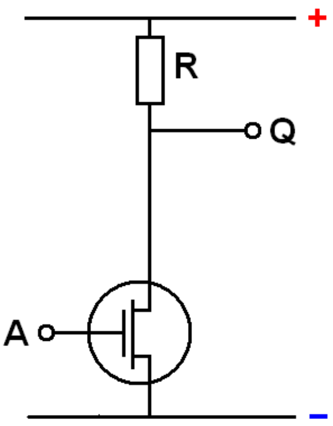
Q = A NAND B



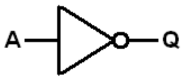


Q = A NOR B





Q = NOT A



These logic circuits are known as logic gates and a computer chip consists of millions of them connected in various ways that allow the computer to perform logical and mathematical operations.

You’ve probably heard that computers work in binary, not decimal, when doing calculations. I’m just going to give a brief introduction to binary to show how any number – and, indeed anything can be represented by a sequence of 1s and 0s.

We’re all familiar with hundreds, tens and units. But why use ten as a base for counting? Simply because we have 10 fingers. Other bases are more useful, 12 being an example since it can be divided by 2, 3, 4 and 6 whereas 10 can only be divided by 2 and 5. Computers use 2 as a base for counting so instead of thousands, hundreds, tens and units we have eights, fours, twos and units. The binary system has the advantage that it only needs two symbols, 0 and 1 but the disadvantage that any number need many more digits in binary than it does in decimal.

For example 99 in decimal is 1100011 in binary, that is

1 x 64 +
1 x 32 +
0 x 16 +
0 x 8 +
0 x 4 +
1 x 2 +
1 x 1.

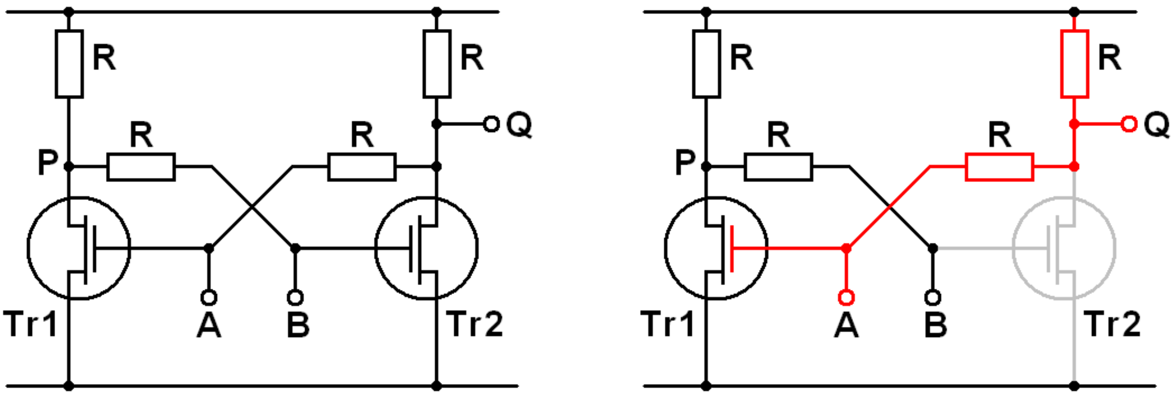
This is cumbersome for humans but dead easy for computers which can only deal with 1s and 0s anyway. I’ll show you a simple addition in binary and how the computer would do the addition:

 5   101
 6+   110+
11   1011

Although I haven’t mentioned it before, there’s a logic circuit called an Exclusive OR which is the same as the OR circuit mentioned earlier except that it excludes the case where both inputs are 1, in which case the output is 0. In other words, if the 2 inputs are different, the output is a 1; if the 2 inputs are the same, the output is a 0. If we look at the sum 5 + 6 we see that the answer apart from the 8s digit is precisely the output of an exclusive OR logic gate. However, we do need an AND gate to detect when both inputs are 1 in which case the AND will generate a carry, but you can see how we’ve got the beginnings of an adding machine just by using two types of logic gate.

One more type of circuit known as a flip-flop can be easily fabricated from MOSFETs and is effectively a memory circuit in that it is stable in two distinct states and so can be used to store a bit of information. A bit, by the way, is short for **b**inary dig**it**.

The circuit is shown here:



If we apply a positive voltage to A, transistor Tr1 will turn on making the voltage on B zero which will ensure that transistor Tr2 is turned off. If we now take away the voltage on A, Tr1 will remain on because the input A is held high by Tr2 being off.

If we now apply a positive voltage to B, this will turn transistor Tr2 on, making input A zero and turning Tr1 off. Removing the voltage from B will have no effect because input B is held high by Tr1 being off.

We've been discussing transistors and circuits but how are they made?

When I was working on chip design in the mid 1970s, chip fabrication was done using photographic processes. A pure silicon wafer – a slice 1.5 inches in diameter, was cut from a long cylindrical crystal of pure silicon which was made from sand which is primarily Silicon Dioxide, also known as Silica.

The silica sand is mixed with carbon and heated in a furnace to over 2000 degrees Celsius. The result is Carbon Dioxide and Silicon. The silicon is then treated with oxygen to reduce calcium and aluminium impurities resulting in 99% pure silicon. This isn't good enough so ...

The silicon is ground to a fine powder and then reacted with hydrogen chloride gas in a fluidised bed reactor at 300 degrees C to give a liquid compound of silicon called trichlorosilane. (This is the silicon version of chloroform). Impurities also react to give their chlorides which are then removed by fractional distillation. The purified trichlorosilane is then vaporised and reacted with hydrogen gas at 1100 degrees C which releases pure silicon which is deposited on the surface of an electrically heated ultra-pure silicon rod to produce a cylindrical silicon ingot which is typically 99.999999% pure.

Although extremely pure, this ingot is not a single crystal but comprises millions of small crystals and is known as polycrystalline silicon. For use in electronic devices, a single crystal is required and this is achieved by melting the silicon in a rotating quartz crucible held just above silicon's melting point of 1414 degrees C. A tiny crystal of silicon is then dipped into the molten silicon and slowly withdrawn while being continuously rotated in the opposite direction to that of the crucible. The crystal acts as a seed causing silicon from the crucible to crystalise around it. This builds up a rod – called a boule – than comprises a single huge silicon crystal.

The diameter of the boule depends on various factors but today's boules are around 12 inches in diameter. I mentioned earlier 1.5 inches but that was over 35 years ago.

The boule is then sliced into thin discs called wafers, typically just under a millimetre thick. The saws are wires coated with silicon carbide. The sharp edges are then smoothed to reduce the chances of chipping during later processes.

Next the wafers are lapped using an abrasive slurry until they are flat to within 2 microns – that's two thousandths of a millimetre. A mixture of nitric, hydrofluoric and acetic acids is then applied to give an even smoother and cleaner surface.

Anisotropic Etching vs. Isotropic Etching

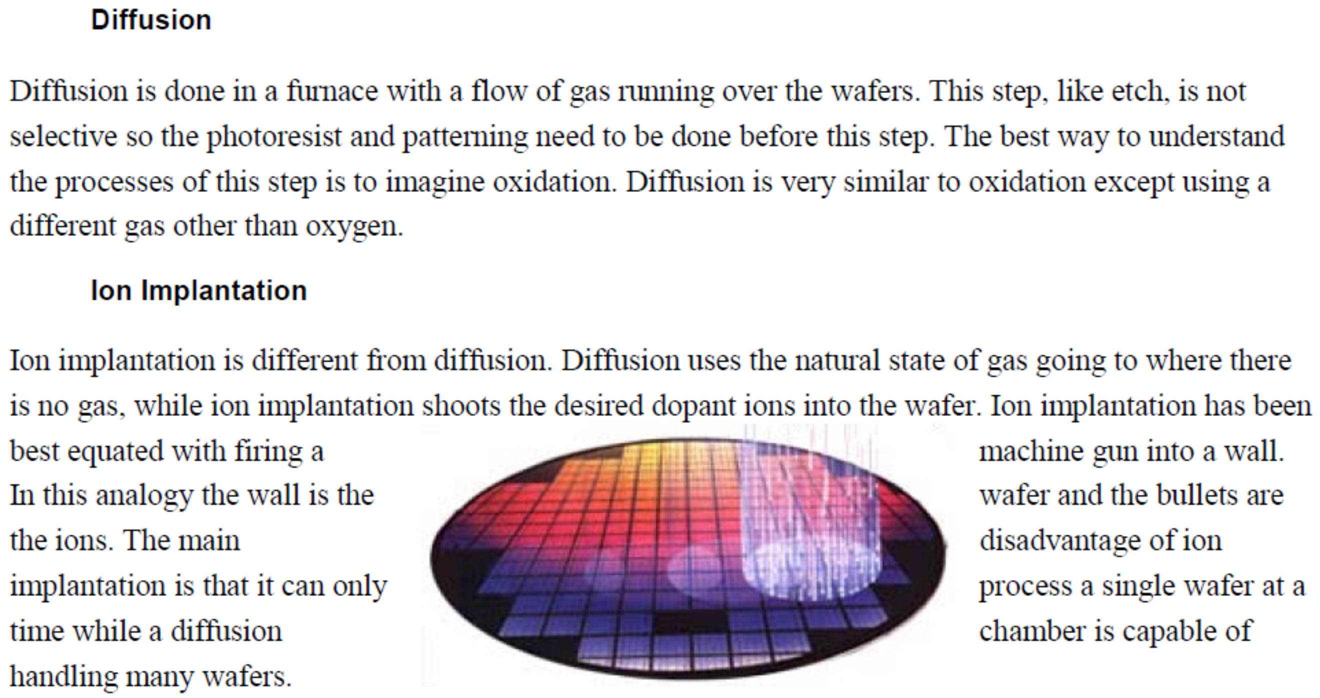
Isotropic etching is a problem that results from chemical etching and some forms of dry etching. The result is that the etchant material will etch to the side (laterally) as well as straight down. This can cause some of the material under the patterned resist to be etched away, resulting in undercutting and poor image accuracy. Anisotropic etching occurs in most forms of dry etching. In this process there is no lateral etching so an exact representation of the pattern is etched onto the wafer. Anisotropic etching is more desirable because there can be problems in maintaining the desired electrical characteristics if there is lateral etching as well as vertical etching.

Implant / Masking Steps: Diffusion & Ion Implant

Electrical characteristics of selected areas on the developing integrated circuit are changed by implanting energized ions (dopants) in the form of specific impurities into areas not protected by resist or other layers. The dopants come to rest below the wafer's surface, creating the positive and negative areas on the wafer which encourage or discourage the flow of electrical current throughout the die. These basic steps are repeated for additional layers of polysilicon, glass, and aluminum. Typical dopants include:

- Boron
- Arsenic
- Phosphorous

These processes can be damaging to the wafer, so a heating process known as annealing is used to reduce any damage to the wafers.



Diffusion

Diffusion is done in a furnace with a flow of gas running over the wafers. This step, like etch, is not selective so the photoresist and patterning need to be done before this step. The best way to understand the processes of this step is to imagine oxidation. Diffusion is very similar to oxidation except using a different gas other than oxygen.

Ion Implantation

Ion implantation is different from diffusion. Diffusion uses the natural state of gas going to where there is no gas, while ion implantation shoots the desired dopant ions into the wafer. Ion implantation has been best equated with firing a machine gun into a wall. In this analogy the wall is the wafer and the bullets are the ions. The main disadvantage of ion implantation is that it can only process a single wafer at a time while a diffusion chamber is capable of handling many wafers.

When I was working in chip design back in 1975, the sizes of the finest details on a chip were of the order of 1 micron = one thousandth of a millimetre. Today’s chips have their finest details less than one twentieth of that, i.e. 40 nanometres or less where a nanometre is one millionth of a millimetre!

In those days I worked at Plessey Research, Caswell, Towcester and we had one of the first computer systems aimed at chip design. Before that, chip designs were done by hand at 500 times actual size and then photographically reduced to make the masks used in the fabrication processes. Just as a matter of interest, the chip design computer had twin 14 inch diameter hard discs, one fixed and one removable, each of which had a storage capacity of 1.25 Megabytes. Compare this phone which has a storage capacity 64000 times as big.

Processor Design

The traditional computer chip or microprocessor has various internal storage areas with dedicated functions called registers. Typical registers are

The Accumulator - where most arithmetic and logic operations are performed

Memory Address Register – which contains the address of a memory location which needs to be accessed

Program Counter – another address register which contains the address of the memory location which contains the next instruction in the program currently running

Stack Pointer - which contains the memory address of the next location on the stack.

The stack is used during subroutine calls

CISC versus RISC

Every processor has an instruction set, effectively a vocabulary, which will include instructions such as “Add the contents of memory address x to the contents of memory address y” which is implemented as a sequence of a few binary numbers, one to tell the processor that it is to add the contents of the two memory locations specified in the following numbers and the rest containing the relevant addresses. This is an example of a simple instruction – addition is very easy to implement in binary. When it comes to operations such as multiplication and division, things get more complicated. I won’t go into more detail at this stage but some processors have a large number of available instructions. Such processors are called CISC.

In the late 60’s a genius, John Cocke, working for IBM came up with the idea of a Reduced Instruction Set Computer or RISC. This seems at first glance to be a crazy idea but on investigation it turns out to be brilliant.

This concept is based on the fact that typically 80% of the computer chip’s time will be spent running 20% of its instruction set, so the physical transistors that constitute the hardware that executes the other 80% of the instructions will be idle most of the time and yet still consuming power, albeit a very small amount of power, but it all adds up. A more important aspect is the space taken up by these infrequently used transistors. It means the chip must be bigger and signals have farther to travel. Although electrical signals travel very fast, in modern computer chips, the time taken to get from A to B on the chip is a significant factor in limiting the maximum speed of operation of the processor. Moreover, the cost of a chip is size dependent and the more complex the chip, the greater the chance of a fatal defect occurring during fabrication. Incidentally, the more complex instructions of the CISC designs can be implemented by a sequence of the simple instructions of the RISC design.

So the advantages of a RISC chip are:

- Faster
- Cheaper
- More reliable
- Lower Power

In 1986, Acorn computers of Cambridge, who designed and built the BBC Micro in the early 80s, had a problem. Other manufacturers were improving their computers by using more powerful chips and Acorn had to decide where it would go. There were two major contenders for Acorn’s next generation at the time, Intel and Motorola. Both had a long history in the microprocessor business and consequently had an interest in making their new processors backward compatible.

Intel chose to make its new processor so that it would run the same instruction set as its older brother as well as an extended instruction set to take advantage of the extra facilities available. This limited the design as well as making it more complex.

Motorola took a different approach. Ultimately, programs get converted to the binary numbers that the computers understand but there are often intermediate stages, one of which is called assembly language. At one time, many programs were written in assembly language as it was the only way of making them fast and compact. An instruction which loaded a number (say 6) into the accumulator might be written LDA #6 but a program called an Assembler would convert this into the binary needed by the processor. Motorola decided to retain backward compatibility at assembler level so that any programs in assembly language could be converted via a different assembler to run on the new chips – a much better idea, but still with limitations as to the design of the new processor.

One aspect of any processor is the interrupt. Imagine you’re holding a party at your home and there’s no doorbell and the music and chatter is so loud that you can’t hear people knocking and you don’t have a telephone. You’ll have to keep going to the door to see if any more guests have arrived, an annoying but more importantly, time consuming task. With a doorbell, as soon as the bell goes, you drop what you’re doing, answer the door, greet the guest, and then return to what you were doing before being interrupted by the doorbell. Processors have the equivalent of a doorbell – a connection which, if its voltage is changed will interrupt what the processor is doing and cause it to service whatever caused the interrupt. This means that the processor must save everything it’s currently doing so that it can retrieve it all and carry on once it’s finished servicing the interrupt. This saving is normally done by saving to memory, usually on the stack, all the processor’s registers.

Now the trouble with more sophisticated processors is that they have more and bigger registers with the consequence that saving them all when an interrupt occurs takes longer. This time is called the Interrupt Latency. The chip which Acorn had used as the heart of the BBC Micro was a simple processor with a very low interrupt latency and Acorn wasn't prepared to compromise on any aspect of the specification of its planned new computer. The interrupt latencies of both the Intel and Motorola chips were significantly longer than that of the BBC Micro chip so Acorn took an astonishing decision – to design its own high performance chip. Here was a small British company with no experience in chip design aiming to beat American giants at their own game. Even more astonishing was the result – a RISC chip that was so fast that it could run a software emulation of the processor in the BBC Micro as fast as the simpler processor could run its hardware. This was the ultimate in backward compatibility because if it could emulate one earlier processor in software, it could emulate others. Moreover, there was no limit or design criterion that had to be observed during the design of the RISC chip which Acorn called the ARM which stood for Acorn RISC Machine. How did Acorn get round the Interrupt Latency problem? A very simple solution – because the chip was so small, they could afford to make it bigger and build in a duplicate set of registers so that when an interrupt occurs, the normal registers are left exactly as they are and the interrupt service program runs in the duplicate set which means nothing has to be saved to memory so the interrupt latency is exceedingly low.

At the time I was working for Acorn in the late 1980s, Acorn released the Archimedes, the first desktop computer based on the ARM chip, and the fastest desktop computer in the world at the time. To give an example of its speed, 2 years after buying an Archimeded I bought my first IBM compatible PC.

The 2 year old Archimedes was running its processor at 8 MHz

The brand new PC was running its Intel processor at 33 MHz, just over 4 times the speed.

I wrote the same program on both computers – repeatedly calculating the trigonometric sine of an angle thousands of times, and did a timing comparison.

On the 8 MHz ARM based computer it took 1.61 seconds

On the 33 MHz Intel based computer it took 1.59 seconds, a mere 2 hundredths of a second faster than the ARM processor running at a quarter the speed.

Unfortunately, Acorn lost out in the desktop manufacturing race, but the company it founded to build on the ARM's success is a huge success story. 80% of mobile devices today are based on an ARM chip, and Intel is still struggling to produce a processor with the performance and low energy consumption of the ARM, and it's only had over 25 years to catch up.

John Foggitt