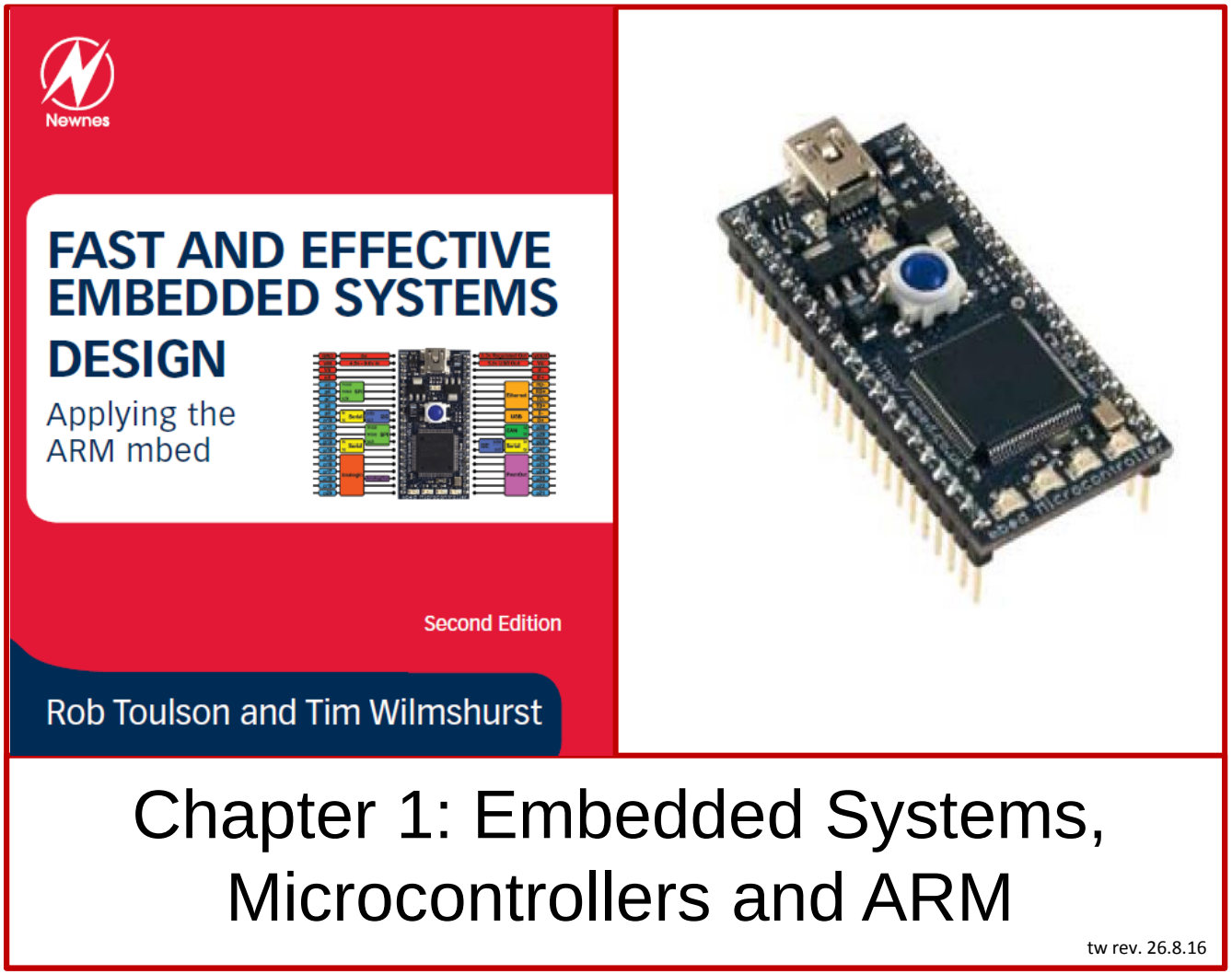



# Chapter 1: Embedded Systems, Microcontrollers and ARM

tw rev. 26.8.16

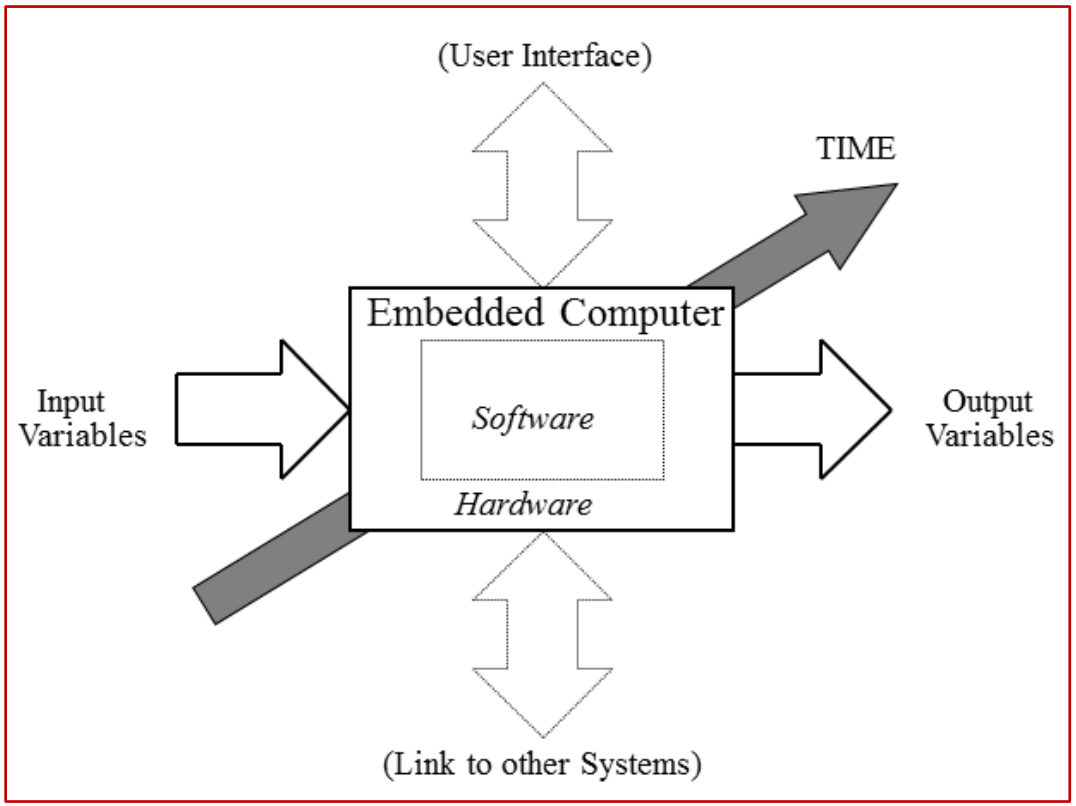
*If you use or reference these slides or the associated textbook, please cite the original authors' work as follows:*

Toulson, R. & Wilmshurst, T. (2016). Fast and Effective Embedded Systems Design - Applying the ARM mbed (2nd edition), Newnes, Oxford, ISBN: 978-0-08-100880-5.

www.embedded-knowhow.co.uk

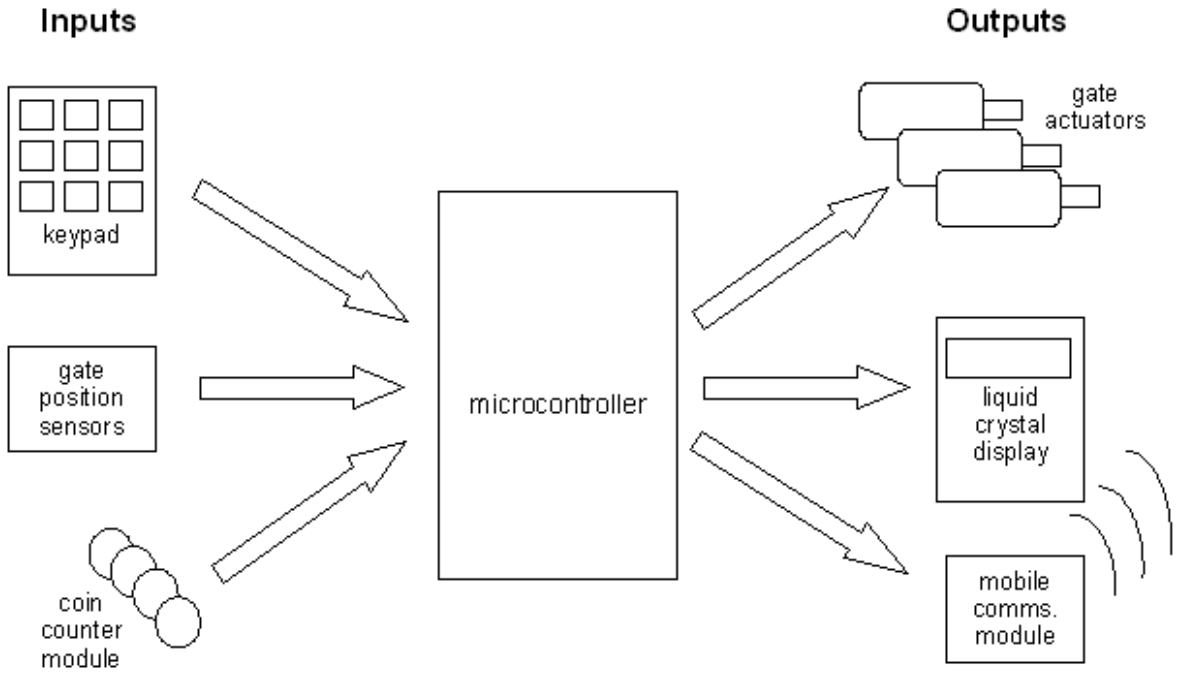
# What is an Embedded System?

An embedded system is a product which is controlled by a computer inside it, but whose overall function is not computing. Examples are near endless, and include washing machines, office equipment, and car engine management.




# An Example Embedded System

A snack vending machine is a good example of an embedded system. At its heart is a single microcontroller. As the diagram shows, this accepts a number of input signals, from the user keypad, the coin counting module, and from the dispensing mechanism itself. It generates output signals dependent on those inputs.




# The Segway Personal Transporter

The Segway personal transporter uses sensitive gyroscopes to ensure that the standing platform always remains horizontal. If weight shifts forward (and the gyroscope shows an imbalance) then the motor in the wheels moves forwards marginally to compensate and stops moving when a horizontal position has been achieved again. The microcontroller, sensors and actuators act together to ensure that the platform stays stable.

# Microprocessors and Microcontrollers

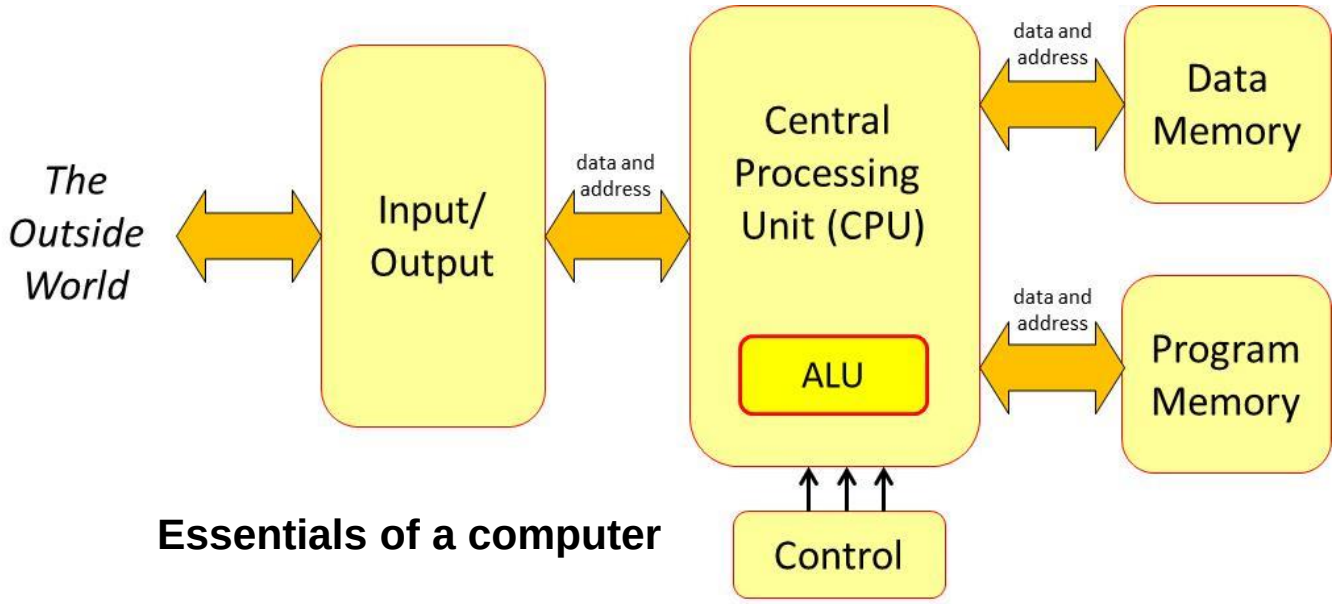
The Figure shows the essential elements of a computer system.

A computer can perform arithmetic or logical calculations. It does this in a digital electronic circuit called the ALU, or *Arithmetic Logic Unit*.

The ALU is placed within a larger circuit, called the *Central Processing Unit* (CPU), which provides some of the supporting features that it needs.

Which arithmetic or logical calculation the ALU does depends on a digital code which is fed to it, called an instruction.

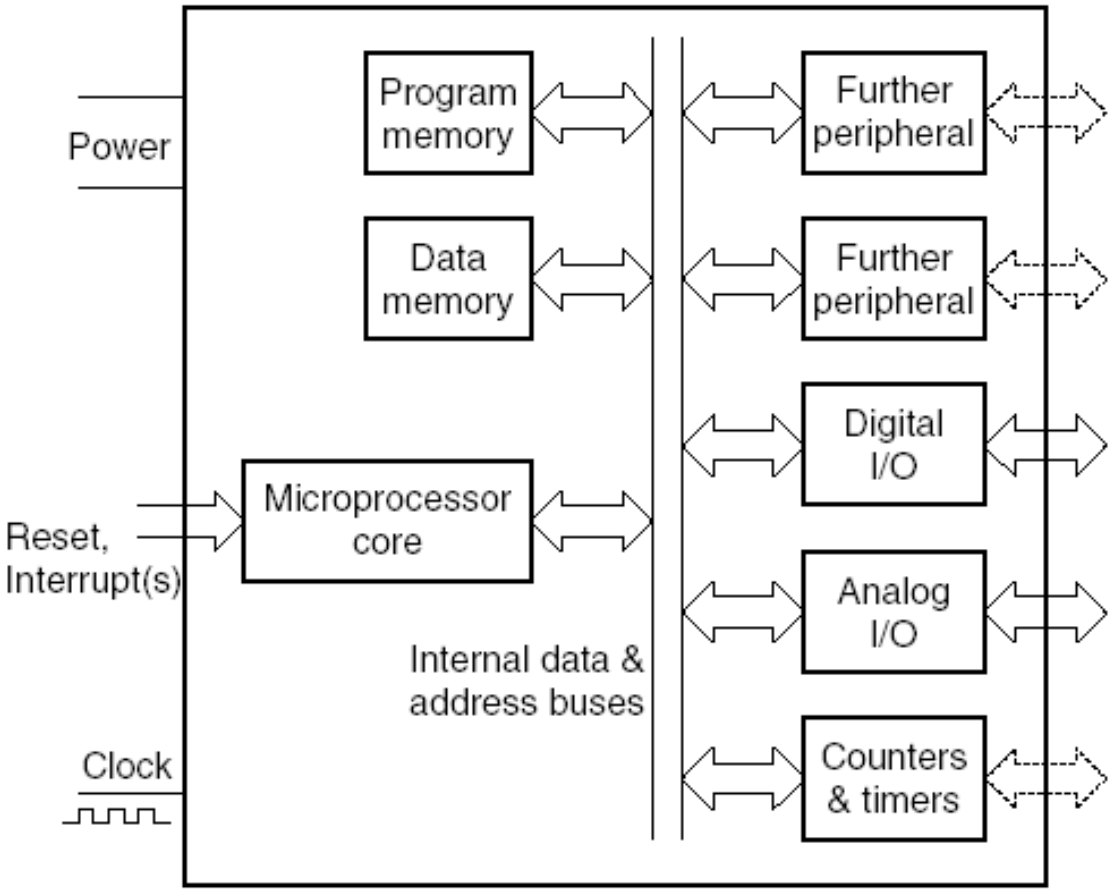
If the ALU is kept busy, by feeding it a sequence of instructions, and also passing it the data it needs to work on, then we have a simple computer.

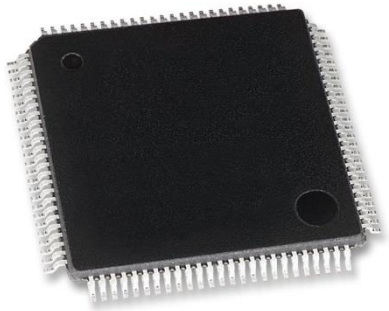



**Essentials of a computer**

# The Microcontroller

A microcontroller takes the essential features of the computer, and adds to these the features that are needed for it to perform its control functions. It's useful to think of it as being made up of three parts: ***core, memory and peripherals***,



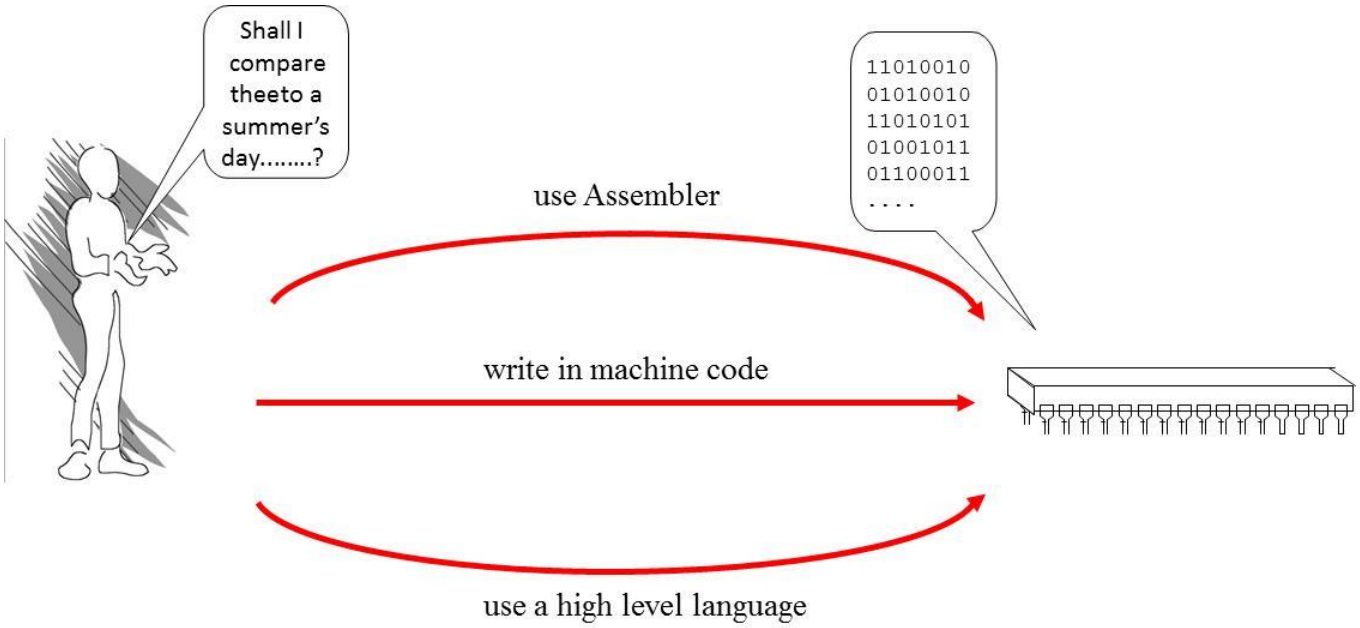



# Development Processes - what's so Special about C/C++?

As programmers, it is our ultimate goal to produce a program which makes the microcontroller do what is needed. That program must be a listing of instructions, in binary, drawn from the instruction set. We sometimes call the program in this raw binary form *machine code.*

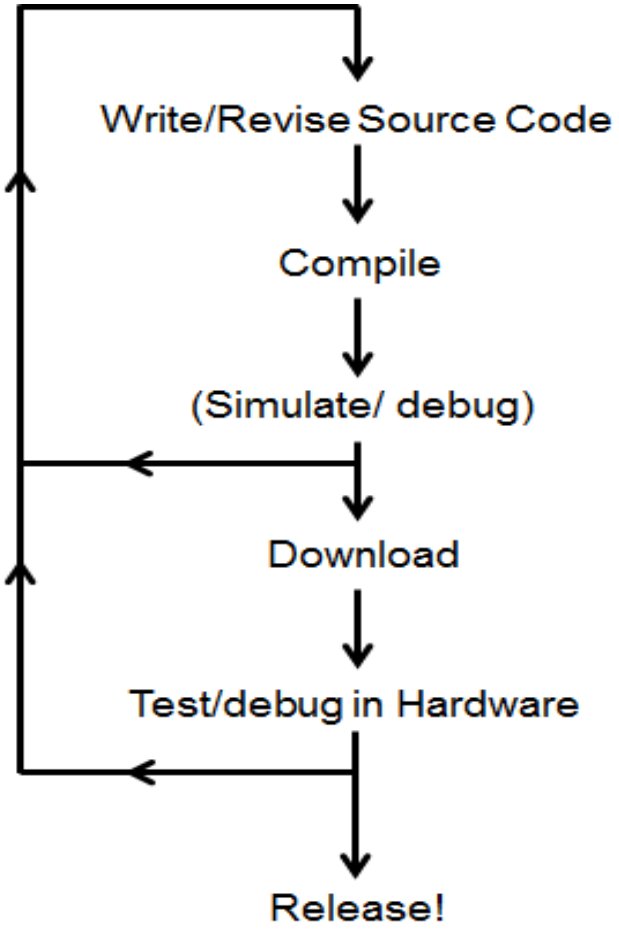
It's tedious, to the point of being near impossible, for us as humans to work with the binary numbers which form the instruction set. A step to programming sanity is to use a *High Level Language* (HLL), which generates the machine code through a *compiler.*

C is a HLL with clear advantages. This is because it is simple, and has features which allow access to the hardware when needed. A step up from C is C++, which is also widely used for more advanced embedded applications.



# The Development Cycle

Program development is based around this simple loop. The source code is written on the host computer, using a text editor. The source code is converted into the binary machine code by a compiler (if writing in C/C++), which is downloaded to the microcontroller itself. Program download to the target system requires temporary connection to the host computer.



# The world of ARM - a Little History

In 1981 the Acorn computer became a popular machine in schools and universities in the UK. It used an 8-bit processor made by MOS Technology.

Around the same time IBM produced its very first PC, based on a more powerful Intel 16-bit microprocessor.

Throughout the 1980s the influence of the IBM PC grew, and its smaller competitors began to fade.

Designers at Acorn realised:

- that they had the capability to design the microprocessor itself and didn't need to buy it in from elsewhere.
- that their future may not lie in selling the completed computer itself.
- that you don't need to manufacture silicon to be a successful designer, what matters is the ideas inside the design. These can be sold as IP, intellectual property.

A group of them formed *Advanced RISC Machines Ltd,* later simply called ARM. They applied the above principles.

# What does RISC mean?

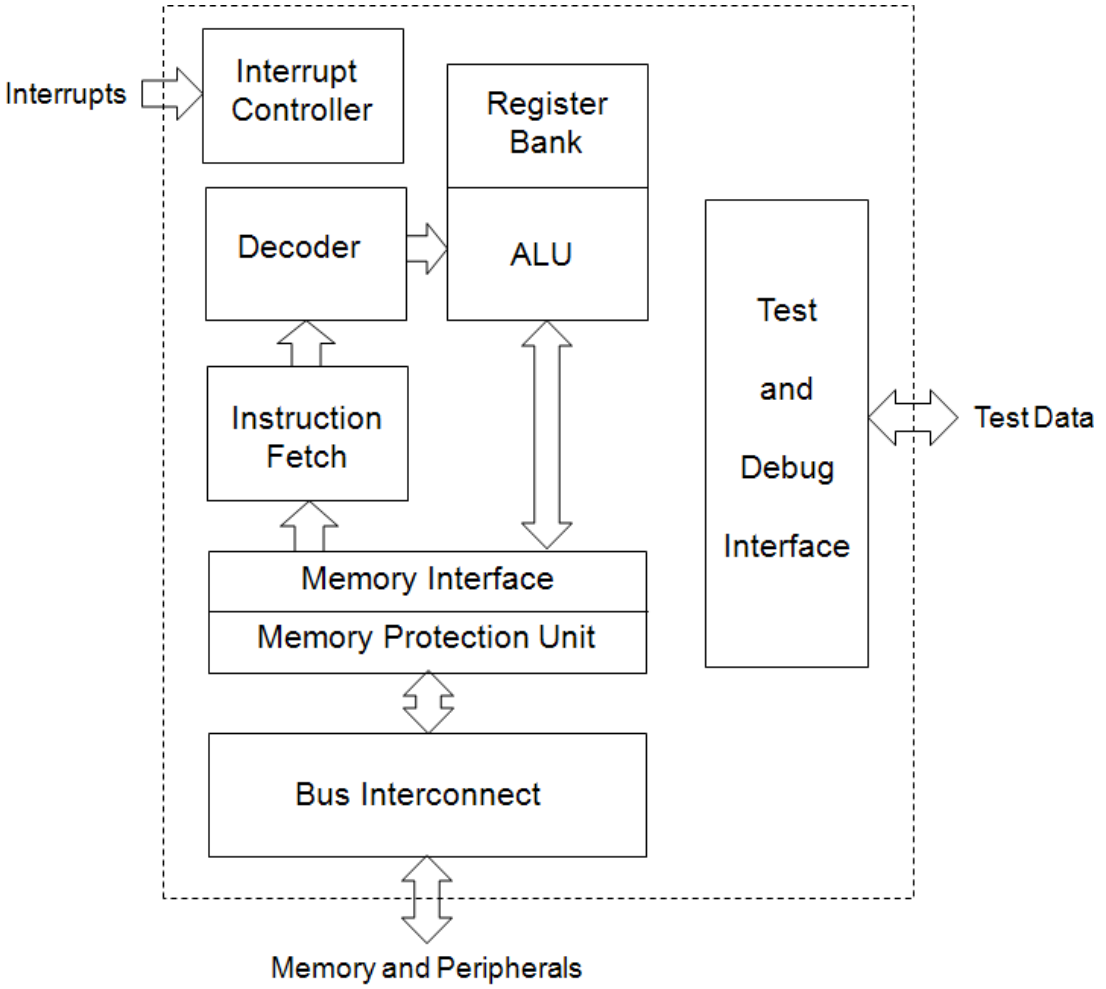
One approach to CPU design is to insist on keeping things simple, and have a limited instruction set. This leads to the RISC approach – the *Reduced Instruction Set Computer*. This differs from the *Complex Instruction Set Computer* (CISC).

The RISC approach looks like a "back to basics" move. A simple RISC CPU can execute code fast, but it may need to execute more instructions to complete a given task, compared to its CISC cousin. Characteristic of the RISC approach include:

- each instruction is contained within a single binary word.
- every instruction normally takes the same amount of time to execute.
- Other useful design features can then be implemented, for example pipelining - an approach by which as one instruction is being executed, the next is already being fetched from memory.

# The world of ARM - The Cortex core

The ARM concept continued to prosper, with a sequence of smart microprocessor designs being sold as IP, to an increasing number of major manufacturers around the world. This led to the Cortex family of microprocessor cores.




**The Cortex M3 core, simplified**

# Chapter Review

- An embedded system contains one or more tiny computers, which control it, and give it a sense of intelligence.
- The embedded computer usually takes the form of a microcontroller, which combines microprocessor core, memory and peripherals.
- Embedded system design combines hardware (electronic, electrical and electro-mechanical) and software (program) design.
- The embedded microcontroller has an instruction set. It is the ultimate goal of the programmer to develop code which is made up of instructions from this instruction set.
- Most programming is done in a high level language, with a compiler being used to convert that program into the binary code, drawn from the instruction set and recognised by the microcontroller.
- ARM has developed a range of effective microprocessor and microcontroller designs, widely applied in embedded systems.