**Module 1 Lesson 2**

**1.5 Designing for Performance**

Desktop applications that require the great power of today’s microprocessor-based systems include

• Image processing

• Speech recognition

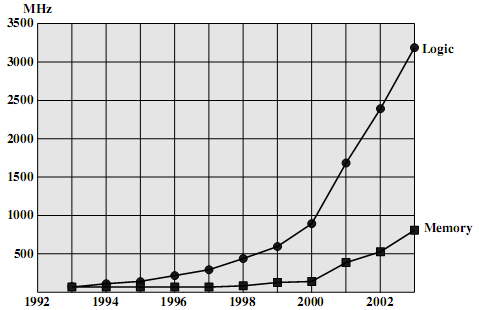
• Videoconferencing

• Multimedia authoring

• Voice and video annotation of files

• Simulation modeling

**1.5.1 Performance Balance**

While processor power has raced ahead at breakneck speed, other critical components of the computer have not kept up. The result is a need to look for performance balance: an adjusting of the organization and architecture to compensate for the mismatch among the capabilities of the various components. 

**Figure 1.11 Logic and Memory Performance Gap**

There are a number of ways that a system architect can attack this problem, all of which are reflected in contemporary computer designs. Consider the following examples:

• Increase the number of bits that are retrieved at one time by making DRAMs “wider” rather than “deeper” and by using wide bus data paths.

• Change the DRAM interface to make it more efficient by including a cache or other buffering scheme on the DRAM chip.

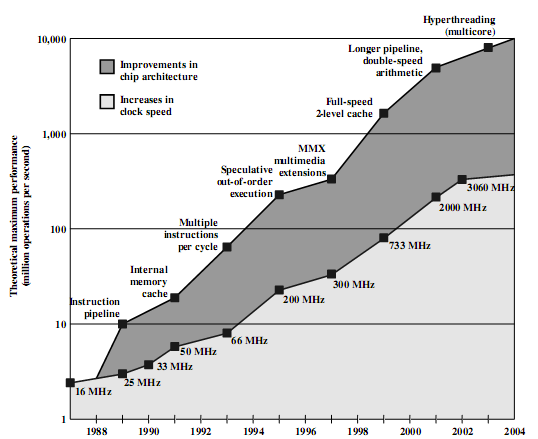
• Reduce the frequency of memory access by incorporating increasingly complex and efficient cache structures between the processor and main memory. This includes the incorporation of one or more caches on the processor chip as well as on an off-chip cache close to the processor chip.

• Increase the interconnect bandwidth between processors and memory by using higher-speed buses and by using a hierarchy of buses to buffer and structure data flow.

**1.5.2 Improvements in Chip Organization and Architecture**

***There are three approaches to achieving increased processor speed:***

* Increase the hardware speed of the processor. This increase is fundamentally due to shrinking the size of the logic gates on the processor chip, so that more gates can be packed together more tightly and to increasing the clock rate. With gates closer together, the propagation time for signals is significantly reduced, enabling a speeding up of the processor. An increase in clock rate means that individual operations are executed more rapidly.
* Increase the size and speed of caches that are interposed between the processor and main memory. In particular, by dedicating a portion of the processor chip itself to the cache, cache access times drop significantly.
* Make changes to the processor organization and architecture that increase the effective speed of instruction execution. Typically, this involves using parallelism in one form or another.



***Figure 1.12 Intel Microprocessor Performance***

Traditionally, the dominant factor in performance gains has been in increases in clock speed due and logic density. Figure 1.12 illustrates this trend for Intel processor chips. However, as clock speed and logic density increase, a number of obstacles become more significant:

* Power: As the density of logic and the clock speed on a chip increase, so does the power density (Watts/cm2).The difficulty of dissipating the heat generated on high-density, high-speed chips is becoming a serious design issue.
* RC delay: The speed at which electrons can flow on a chip between transistors is limited by the resistance and capacitance of the metal wires connecting them; specifically, delay increases as the RC product increases. As components on the chip decrease in size, the wire interconnects become thinner, increasing resistance. Also, the wires are closer together, increasing capacitance.
* Memory latency: Memory speeds lag processor speeds, as previously discussed.

Beginning in the late 1980s, and continuing for about 15 years, two main strategies have been used to increase performance beyond what can be achieved simply increasing levels of Cash memory and enable parallel execution of instructions within the processor.

Both of these approaches are reaching a point of diminishing returns. The internal organization of contemporary processors is exceedingly complex and is able to squeeze a great deal of parallelism out of the instruction stream.

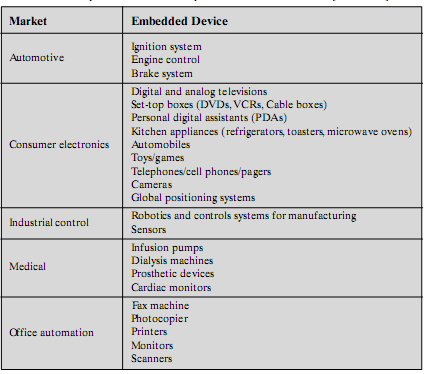
Designers have turned to a fundamentally new approach to improving performance: placing multiple processors on the same chip, with a large shared cache. The use of multiple processors on the same chip, also referred to as multiple cores, or multicore, provides the potential to increase performance without increasing the clock rate.

**1.5.3 Embedded Systems and The ARM**

The ARM architecture refers to a processor architecture that has evolved from RISC (Reduced Instruction Set Computer) design principles and is used in embedded systems.

***Embedded Systems***

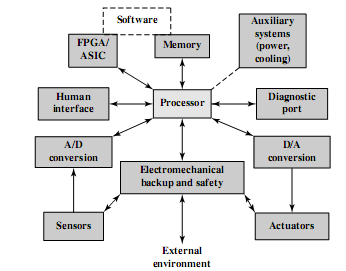
The term embedded system refers to the use of electronics and software within a product, as opposed to a general-purpose computer, such as a laptop or desktop system. The following is a good general definition: **Embedded system**. A combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a dedicated function. In many cases, embedded systems are part of a larger system or product, as in the case of an antilock braking system in a car.

**Table 1.3 Examples of Embedded Systems and Their Markets** 

Embedded systems far outnumber general-purpose computer systems, encompassing a broad range of applications (Table 1.3).These systems have widely varying requirements and constraints, such as the following:

* Small to large systems, implying very different cost constraints, thus different needs for optimization and reuse
* Relaxed to very strict requirements and combinations of different quality requirements, for example, with respect to safety, reliability, real-time, flexibility, and legislation
* Short to long life times
* Different environmental conditions in terms of, for example, radiation, vibrations, and humidity
* Different application characteristics resulting in static versus dynamic loads, slow to fast speed, compute versus interface intensive tasks, and/or combinations thereof
* Different models of computation ranging from discrete-event systems to those involving continuous time dynamics (usually referred to as hybrid systems)

Often, embedded systems are tightly coupled to their environment



***Figure 1.13 Possible Organization of an Embedded System***

In Embedded System, addition to the processor and memory, there are a numbers of elements that differ from the typical desktop or laptop computer:

• There may be a variety of interfaces that enable the system to measure, manipulate, and otherwise interact with the external environment.

• The human interface may be as simple as a flashing light or as complicated as real-time robotic vision.

• The diagnostic port may be used for diagnosing the system that is being controlled—not just for diagnosing the computer.

• Special-purpose field programmable (FPGA), application specific (ASIC), or even non digital hardware may be used to increase performance or safety.

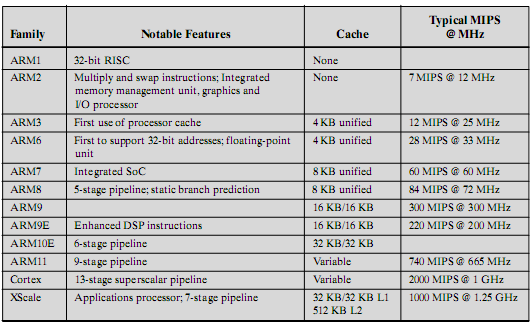
• Software often has a fixed function and is specific to the application.

***ARM Evolution***

ARM (Acorn RISC Machine ) is a family of RISC-based microprocessors and microcontrollers designed by ARM Inc., Cambridge, England. The company doesn’t make processors but instead designs microprocessor and multicore architectures and licenses them to manufacturers. ARM chips are high-speed processors that are known for their small die size and low power requirements. They are widely used in PDAs and other handheld devices, including games and phones as well as a large variety of consumer products.

ARM chips are the processors in Apple’s popular iPod and iPhone devices.ARM is probably the most widely used embedded processor architecture and indeed the most widely used processor architecture of any kind in the world.

**Table 1.4 ARM Evolution**



According to the ARM Web site arm.com, ARM processors are designed to meet the needs of three system categories:

• ***Embedded real-time systems***: Systems for storage, automotive body and power-train, industrial, and networking applications

• ***Application platforms***: Devices running open operating systems including Linux, Palm OS, Symbian OS, and Windows CE in wireless, consumer entertainment and digital imaging applications

• Secure applications: Smart cards, SIM cards, and payment terminals

**1.6 Performance Assessment**

In this section, we will discuss common ways of gauging a system's value in terms of its cost and its performance

An increase in a machine's performance is viewed in one of two (competing) ways:

» Reduced response time to an individual job

» Increase in overall throughput

Which of the following increases throughput, reduces response time, or both?

» Faster clock cycle time

» Multiple processors for separate tasks

» Parallel processing of scientific problems

Recalling that architects design machines to run programs, improved performance is a total system process as embodied by ***Amdahl's Law***:

*The performance improvement to be gained from using some faster mode of execution is limited by the fraction of time the faster mode can be used*

***Clock Speed and Instructions per Second***

Pulse frequency produced by the clock, measured in cycles per second, or Hertz (Hz)

The rate of pulses is known as the **clock rate**, or **clock speed**. One increment, or pulse, of the clock is referred to as a **clock cycle**, or a **clock tick**. The time between pulses is the cycle time.

**1.6.1 Instruction Execution Rate**

A processor is driven by a clock with a constant frequency *f* or, equivalently, a constant cycle time τ , where τ=1/*f* . Define the instruction count, *Ic*, for a program as the number of machine instructions executed for that program until it runs to completion or for some defined time interval. Note that this is the number of instruction executions, not the number of instructions in the object code of the program. An important parameter is the average Cycles Per Instruction CPI for a program. If all instructions required the same number of clock cycles, then CPI would be a constant value for a processor. However, on any give processor, the number of clock cycles required varies for different types of instructions, such as load, store, branch, and so on. Let *CPIi* be the number of cycles required for instruction type *i*. and *Ii* be the number of executed instructions of type *I* for a given program. Then we can calculate an overall *CPI* as follows:

The processor time *T* needed to execute a given program can be expressed as

A common measure of performance for a processor is the rate at which instructions are executed, expressed as millions of instructions per second (MIPS), referred to as the MIPS rate. We can express the MIPS rate in terms of the clock rate and CPI as follows:

Another common performance measure deals only with floating-point instructions. These are common in many scientific and game applications. Floating-point performance is expressed as millions of floating-point operations per second (MFLOPS), defined as follows:

**Benchmark program**

A benchmark suite is a collection of programs, defined in a high-level language, that together attempt to provide a representative test of a computer in a particular application or system programming area.

Lists the following as desirable characteristics of a benchmark program:

1. It is written in a high-level language, making it portable across different machines.
2. It is representative of a particular kind of programming style, such as systems programming, numerical programming, or commercial programming.
3. It can be measured easily.
4. It has wide distribution.

**SPEC Benchmarks** The common need in industry and academic and research communities for generally accepted computer performance measurements has led to the development of standardized benchmark suites. The best known such collection of benchmark suites is defined and maintained by the System Performance Evaluation Corporation (SPEC), an industry consortium. SPEC performance measurements are widely used for comparison and research purposes.

# Assignment 2

2.1 What are the four main components of any general-purpose computer?

2.2 What are the main component of IAS?

2.3 What are the format of numbers and instructure?

2.4 What are the main characteristics of computer family?

2.5 writs some of the methods used in Improvements in Chip Organization and Architecture

2,6 However, as clock speed and logic density increase, a number of obstacles become more significant. Do you agree with it and why? Explain your answer?

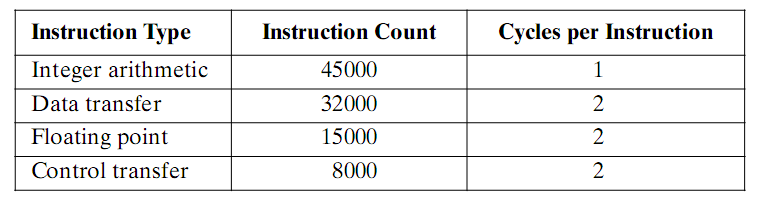
2.7 At the late of 1980, what are the main strategies of increasing computer performance, and where do they lead to?

2.8 clarify these terms in simple words

1. Embedded Systems
2. The ARM architecture

Explain the relation between them? Their application?

2-8A benchmark program is run on a 40 MHz processor. The executed program consists of 100,000 instruction executions, with the following instruction mix and clock cycle count:



Determine the effective CPI, MIPS rate, and execution time for this program

**Quiz2**

1-the improvement of IC technology improve

a)the speed of M.M b)reduce the size of M.M c)the speed of processor d) enlarge the size of processor

2-as the IC industry improve the gap between M.M and Processor

a) Increase b) decrease c) get near to each other

3- Embedded system is ---------as Processor

a)complex b)more simple c) application dependent

4- the measuring unit of computer throughput is

a)Time of Execution b) CPI c) MIPS

Answer (1-c) (2-a) (3-b) (4-c)