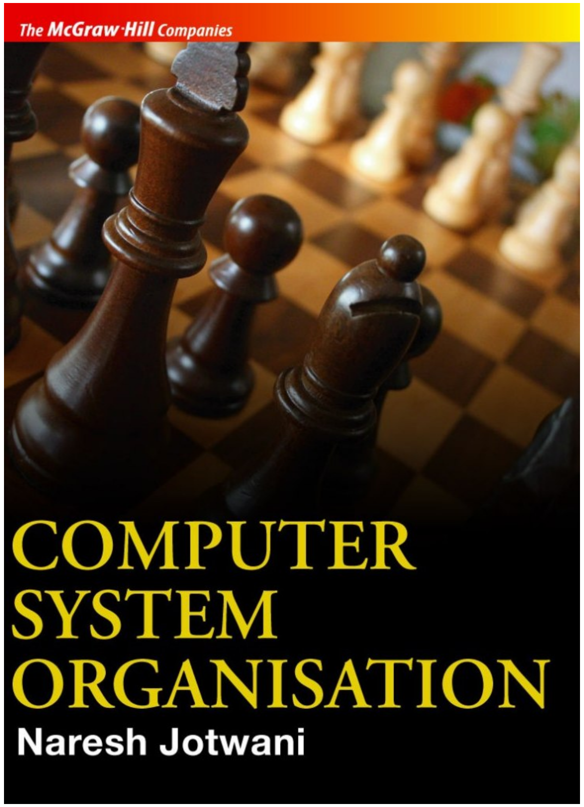
McGraw Hill Education

COMPUTER SYSTEM ORGANISATION

Naresh Jotwani

PowerPoint Slides



CHAPTER 16

CASE STUDIES

Throughput

- In broad terms, *throughput* refers to the capability of a computer system to process data.
- Often, two or more computer systems are compared for performance, and for such purposes the concept of *throughput* needs to be quantified.
- Careful thought is required, since the system is made up of different types of elements – processors, memories, pipelines, data paths, and so on.
- Such quantification may be based on application load. But we may also need to quantify system throughput without reference to a particular application.
- Or we may wish to compare two or more processors for throughput, without reference to the rest of the system, or any particular application.

- In this case, we would need to quantify the throughput of the processors, when all other system elements are made equal or comparable.

- One way to characterize a processor is to determine - based on its design - the theoretical maximum rate at which the processor can execute instructions.
  - Rate can be expressed in units of *million instructions per second* (or *mips*).

- Many compute-intensive applications make heavy use of floating point operations. For these, a more useful measure of performance is *floating point operations per second* (*flops*).
- With prefix *mega* (10^6) or *giga* (10^9), this is written as *megaflops* (*mflops*) or *gigaflops* (*gflops*).

- Compilers play a crucial role in determining the performance of machine language programs.

- Because of frequent pipeline flushes and stalls, the actual processor performance in practice is usually much less than the theoretical maximum for which the processor may have been designed.

- To quantify actual processor performance which can be expected in practice, standard *benchmark programs* are used.

- A widely used set of benchmark programs is developed by System Performance Evaluation Corporation (SPEC).
- It includes benchmark programs for integer & floating point performance, graphics, parallel processing, JAVA applications, mail servers, file servers, and others.

Intel Pentium® family of processors

- In the early 1970s, Intel produced the first ever microprocessor on a single chip, a 4-bit processor which was given Intel part number 4004.
- This processor was soon followed by its 8-bit version 8008, and then the successively improved versions 8080, 8086 and 8088.
- Around 1980, Intel's 8088 processor was selected by IBM for use in its Personal Computer, commonly referred to as IBM PC, or simply PC.
- Other companies – including Apple – had earlier come out with their own path-breaking versions of the personal computer. But IBM PC and its 'clones' soon became the largest selling computers around the world.

- Intel 8088 was a 16-bit processor with 20-bit physical address, i.e. total physical address space of 1 megabyte.
- Logical memory space consisted of four segments – namely, *code*, *data*, *stack* and *extra* segments.
- A 16-bit segment offset in memory address meant that each segment was limited to 64 kilobytes.
    - IBM PC made use of Microsoft operating system DOS, which was distributed and run from 5¼” *floppy disks* with storage capacity of 360 kilobytes each.

- Subsequent processors in Intel’s ‘x86’ processor family were numbered 80286, 80386 and 80486.
    - With advances in VLSI technology, these members of the processor family had larger memory address space, 32 bit word size, higher clock frequencies, on-chip cache and memory management functions, and additional instructions.

- Successive models of the immensely popular PC were built around these processors.

- Therefore maintaining *backward compatibility* of the instruction set with earlier processors of the x86 family has always been a non-negotiable design requirement of any processor of this family.

- It has been Intel's *business strategy* that all software written for earlier versions of the PC must also run with its subsequent versions.

- This means that processors must achieve higher performance with every successive model, while maintaining backward compatibility of instruction set.

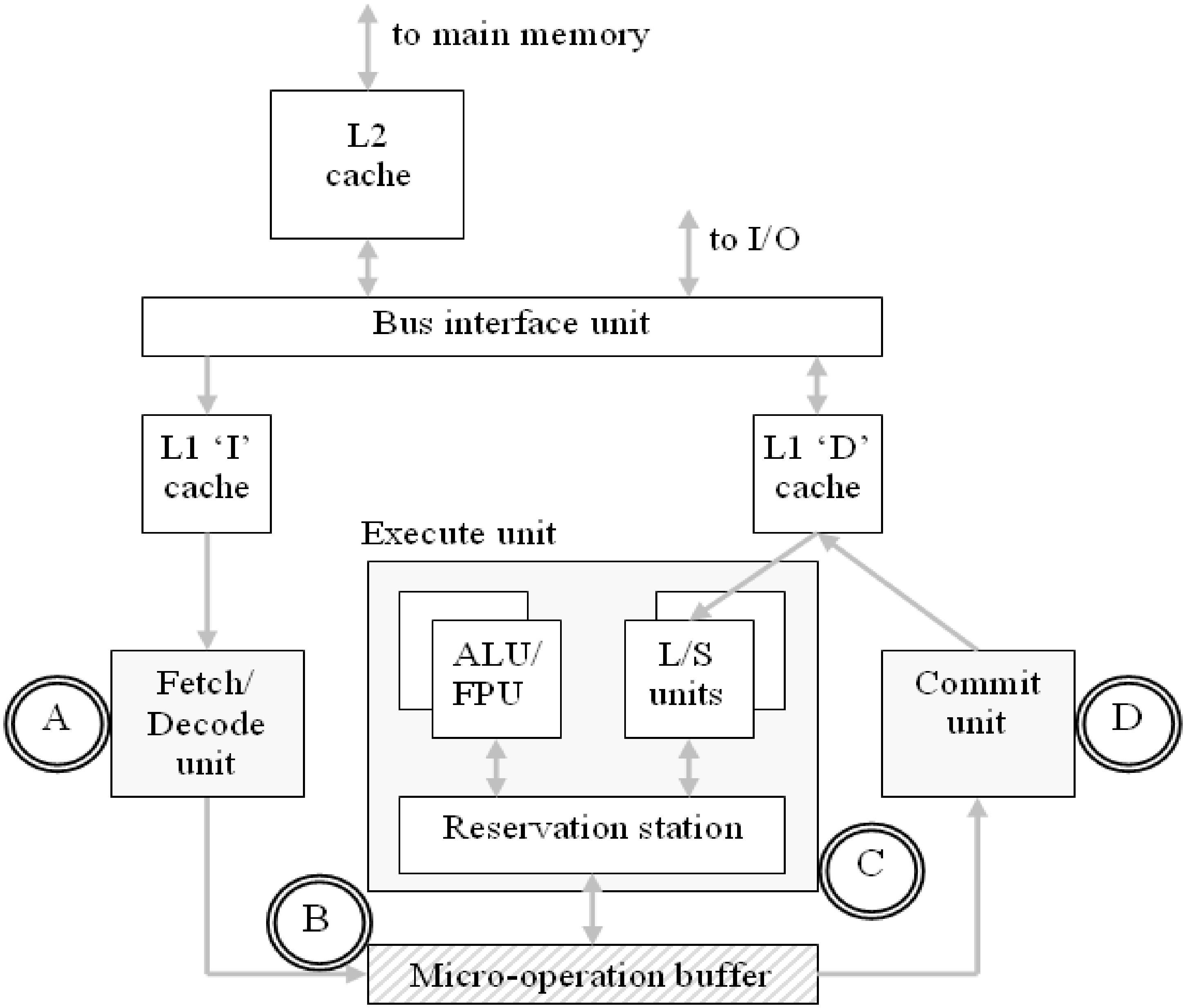
- All other new processor designs had by now benefited from RISC design principles. But, for x86 processors, maintaining backward compatibility was possible only with the CISC approach.

- Designers at Intel pushed the frontiers of VLSI technology to achieve higher processor performance while maintaining backward compatibility.

- The original Pentium and its successors introduced advances over the Intel 80486 processors.

- Even with the inherited CISC instruction set, these processors combined standard RISC design techniques in their internal architecture, such as – a *micro-operation* pipeline, multiple functional units, and out-of-order sequencing.

- Next figure shows the architecture of the Pentium 4 processor.
- The processor has two levels of cache memory – L1 and L2.
  - Faster but smaller L1 cache is divided into 8 kilobytes of instruction cache and 8 kilobytes of data cache
  - Larger L2 cache is a combined instruction and data cache of 256 or 512 kilobytes (depending on the processor model).
- The required instruction or data bytes are accessed from L1 and L2 cache, in that order. If a cache miss occurs in both L1 and L2, access is made from main memory, which may have an additional L3 cache.
- The Fetch/Decode unit ('A' in the figure) is connected to L1 instruction cache. This unit fetches and decodes successive instructions, producing several *micro-operations* for each machine instruction.


to main memory
L2 cache
to I/O
Bus interface unit
L1 'I' cache
L1 'D' cache
Execute unit
ALU/ FPU
L/S units
Fetch/ Decode unit
Commit unit
Reservation station
Micro-operation buffer
A
B
C
D

- Micro-operations are forwarded to the *micro-operation buffer* (marked ‘B’), in which micro-operations produced by multiple machine instructions are buffered.

- For execution by specific functional units – such as the integer ALU or the FPU – micro-operations are forwarded to a *reservation station*.

- An operation is performed in a unit when its operands become available and the functional unit becomes free.

- Execution of micro-operations need not follow the order in which machine instructions are fetched.

- Data load and store operations on memory are carried out by the *load unit* and *store unit*, respectively, which operate as functional units connected to the L1 data cache.

- The reservation station and the functional units together make up the *execute unit* of the processor (marked ‘C’).
- Within this execute unit, hardwired control is provided for the simpler instructions of the processor, whereas complex instructions of the CISC type are provided with microprogram control.
- This is one of the ways in which RISC and CISC approaches are combined in the internal architecture of the Pentium 4 processor.
- When all the micro-operations of an instruction have been performed in the execute unit, the instruction is *committed* (or *retired*) to main memory.

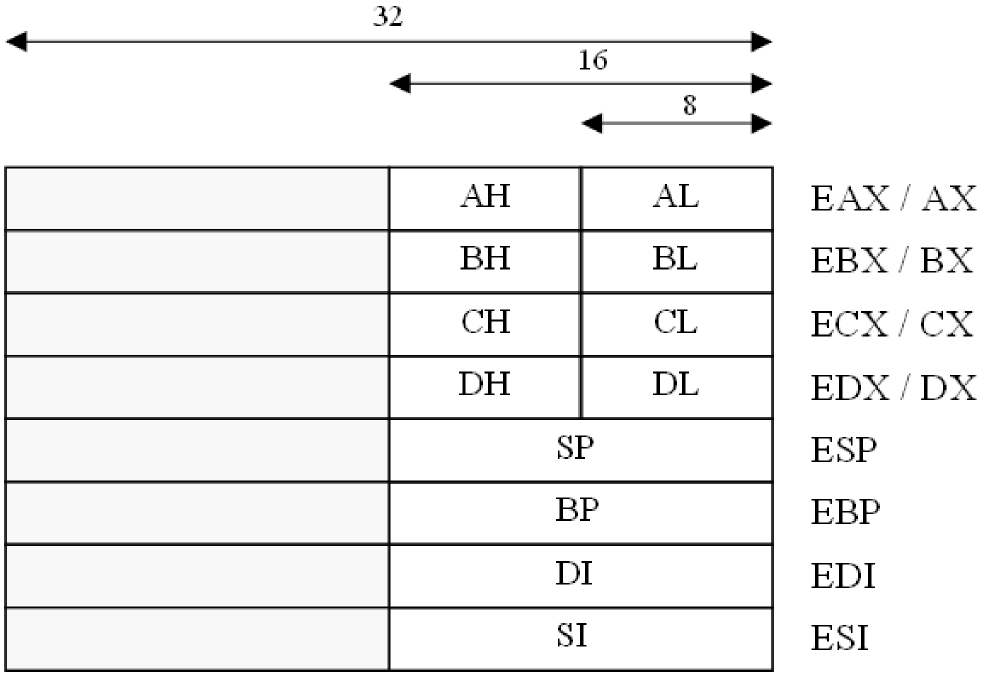
- The *commit unit* (marked ‘D’), ensures that completed machine instructions are committed in the order in which they were fetched.

- *Fetch/decode unit* (‘A’), *execute unit* (‘B’) and *commit unit* (‘D’) operate in parallel, sharing the common *micro-operation buffer* (‘C’).

- Thus these three units form a ‘high-level’ pipeline through which instructions pass.

- But each of these units itself is also implemented as a pipeline, i.e. multiple micro-operations can be in a unit at one time, each in a different stage of processing.

- Branch prediction logic, which is required with the instruction pipeline, is also provided.



	AH	AL	EAX / AX
	BH	BL	EBX / BX
	CH	CL	ECX / CX
	DH	DL	EDX / DX
	SP		ESP
	BP		EBP
	DI		EDI
	SI		ESI

- Set of programmable registers on Pentium processors.
- Starting with Intel 80386 processor, the register size was extended from 16 bits to 32 bits.
  - Portion on the left denotes this extension.
  - Portion on the right shows the 16-bit registers as they existed from the time of the 8088 processor.

- For backward compatibility, register names are chosen to indicate whether an operation is to be performed on 8 bit, 16 bit, or 32 bit register operands.
- Example: ‘AH’ and ‘AL’ denote the higher and lower order bytes, respectively, of the 16 bit ‘AX’ register, and ‘EAX” denotes the 32 bit extension of the ‘AX’ register.
- Registers are not alike in their functionality, and in this sense they are not general purpose registers.
- Memory management functions on the processor provide support for virtual memory using paging and/or segmentation, as well as memory protection for user programs and the operating system. Segments may be shared between running programs.

- *Instruction set architecture* (ISA) of the 32 bit Intel processors is known as IA32. This is the processor as seen by an assembly language programmer or compiler writer, as against the internal processor architecture.
- In theory, the instruction set architecture of a processor can be changed without changing its internal processor architecture.
- However, companies such as AMD do exactly the opposite – they design and build processors with the IA32 instruction set, but develop their own proprietary designs for the internal processor architecture.

Sun UltraSparc T2® Processor

- Concept of *reduced instruction set computing* (RISC) became widely known with the work done by David Patterson at the University of California, Berkeley, and John Hennessey at Stanford University.

- Led to the development of the Sparc processor by Sun Microsystems in the late 1980s.

- Original Sparc processor is a 32 bit RISC processor with load-store architecture, relatively simple addressing modes, and register-to-register arithmetic/logic machine instructions in three-address format. Separate registers are provided for integer and floating point operands.

- Of the 32 integer registers, some play a special role in passing arguments during function calls.

- UltraSparc is the 64 bit enhanced version of Sparc.
- UltraSparc T2 is a *multi-core*, *system on a chip* version, with extensive on-chip support for multithreading, networking, I/O, and so on.

- For increased performance, one option for processor designers is to maximize *instruction issue rate* - by increasing the number of stages in the instruction pipeline. The aim behind this is to drive the processor with a higher clock frequency.

- But the problems of pipeline flushes and stalls do not go away. Also, the total power consumption of the chip increases rapidly with clock frequency, becoming a limiting factor in achieving higher performance.

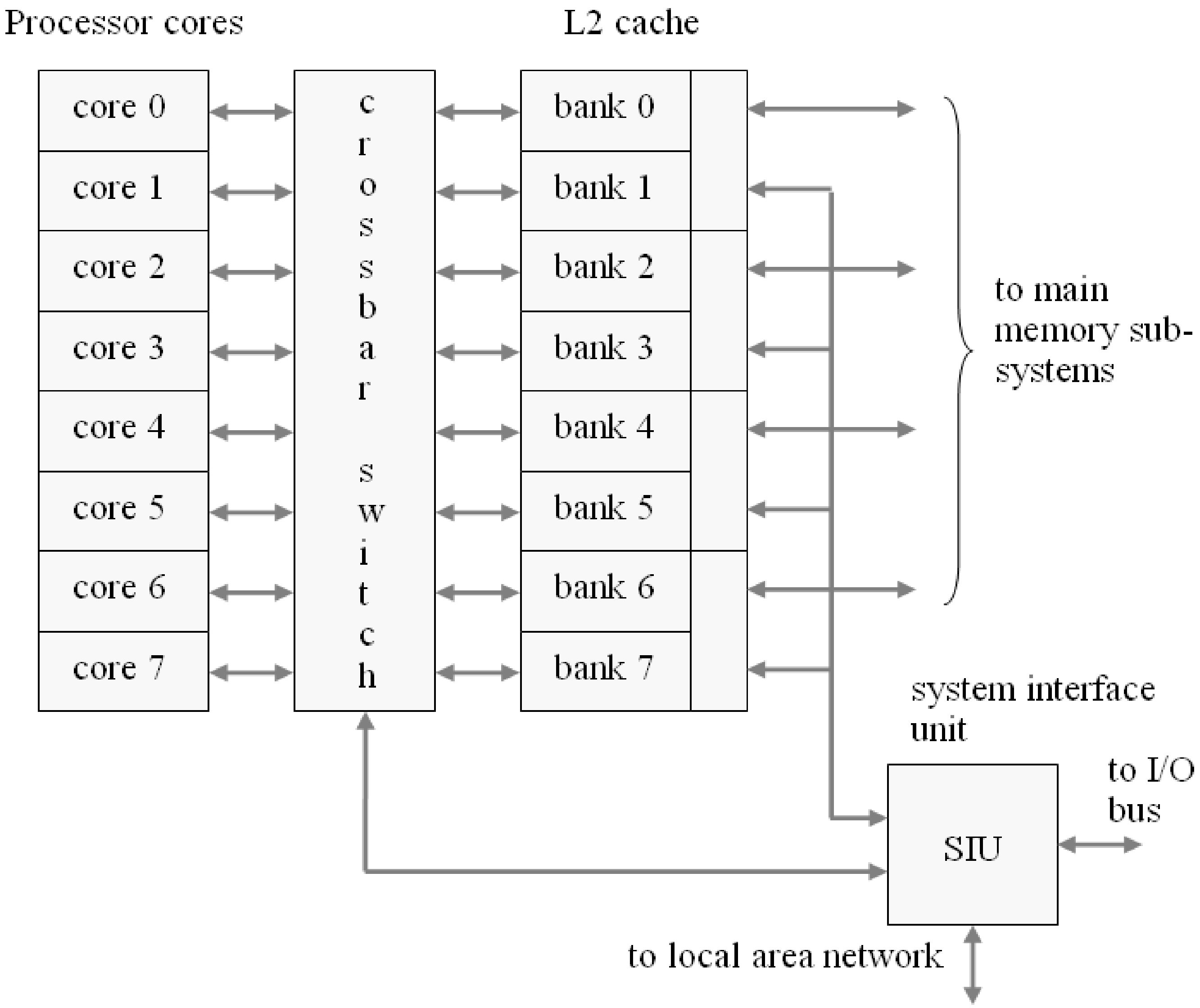
- UltraSparc T2 - and its predecessor UltraSparc T1 - achieve higher processing throughput by adopting a different strategy.

- These multi-core chips are designed for the highly demanding applications with a large degree of thread level parallelism (TLP), but not necessarily much instruction level parallelism (ILP).

- This is *chip multi-threading* (CMT), i.e. the compute time and memory latencies of multiple executing threads are interleaved in time. UltraSparc T2 has *eight* processor cores on the chip, with each supporting *eight-way* fine-grained multi-threading.

- Thus the chip supports sixty four parallel threads. It also contains a crossbar switch, shared L2 cache, and extensive support for I/O and networking – and therefore it is referred to as a *system on a chip* (SoC), rather than 'just a processor' (see figure below).

- Threads run independently of each other, while sharing hardware resources. Therefore the single chip is said to support 64 *virtual systems* on it.



Architecture of UltraSparc T2.

- T2 chip has an area of just under 3.5 cm^2, with about 500 million transistors on it. On average, there is one transistor for every 0.7 square micrometer of chip area.
- The chip operates at 1.4 gigahertz with 1.1 volt supply, and has 1831 pins on its underside for connection to the rest of the computer system.
- Nominal power consumed by the chip is 95 watts.
- Each processing core has its own data paths, register sets for multiple threads, two integer operation units, and an FPU.
- Each core also has hardware provided for cryptography and graphics and – the important point – support for eight-way fine-grained multi-threading.

- Hardware resources − and the per core L1 instruction cache and data cache − are shared by eight threads executing on each core.

- Other hardware resources such as the 4 MB L2 cache are made available to cores through a cross-bar switch.

- For faster access, L2 cache is organized in the form of eight parallel banks. Memory, I/O and network interfaces are also shared amongst the cores.

- Networking capability consists of two network interfaces of 10 gigabits per second.

- Design is targeted towards compute-intensive applications with a high degree of multi-threading. E.g. back-end servers, network devices such as packet routers, switches for local area networks, graphics and imaging applications, and so on.

- In addition to multi-threading support, such applications require very high reliability and availability from the system.
- At chip design stage, this requires intensive verification and simulation – so that all the elements of the *system on a chip* are verified to be logically correct, and are balanced for the aggregate throughput expected.
- For higher reliability, sub-systems and data paths are provided with error detection and correction.
- Complete chip design has been made available on the web to researchers and developers, under an ‘open source’ arrangement. The stated objective behind this decision is to encourage innovations around the world in processor design and applications.

JAVA Virtual Machine

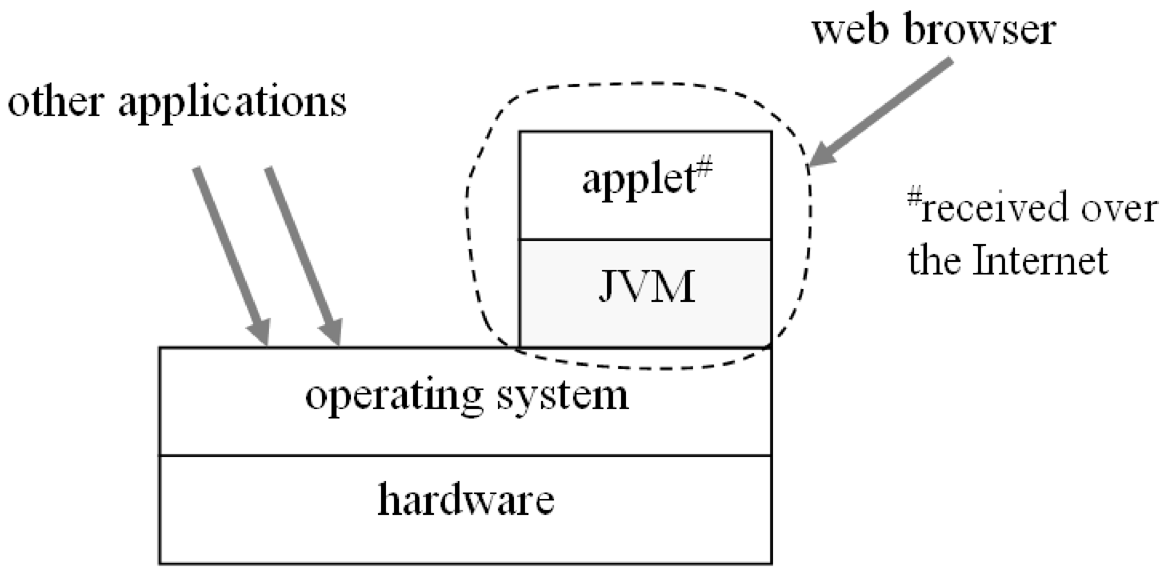
- JAVA is a programming language, developed originally by Sun Microsystems, for writing programs which can run on any computer on the Internet.
- This requires that the program must <u>not</u> be made up of machine instructions of any one particular processor, or of system functions of any one operating system.
- Sun therefore developed <u>*JAVA virtual machine*</u> (JVM) – a processor <u>simulated in software</u>, on which compiled JAVA programs run. JVM is made available as a download on any computer on the Internet.
- Once JVM is available on a computer system, a compiled JAVA program can be executed on the system, regardless of its native hardware and software characteristics .

- Compiled JAVA program is also known as *bytecode*.

- ‘Machine instructions’ in the compiled JAVA program are *interpreted* by JVM – i.e. executed one by one in software. The next figure depicts the relationship between the computer hardware, operating system, JVM, and an applet received over the Internet.

- Typically, JVM is available with the web browser program, which is also responsible for downloading the applet over the Internet.

- The browser, with built-in JVM and the applet running over it, is just one of several applications which may be running on the system at a given time.



- This arrangement ensures that functionality of the applet downloaded over the Internet remains independent of the local hardware and operating system.

- Bytecode is designed to be compact, so as to reduce its download time, and several features of JVM are designed to protect the local system against erroneous or malicious bytecode.

- JVM provides support for *data types* and *classes*. E.g. when an instruction refers to an operand, its *opcode* appears in different variants for integer, ‘long’ integer, single-precision and double-precision operands.

- All JVM opcodes are one byte in length, and most do not have any explicit operands - because JVM instructions work with operands on a runtime *operand stack*. This means JVM has no explicit programmable registers.

- Operands of JVM opcodes are variables which have been brought from memory to runtime stack.

- When an arithmetic or logic operation takes place, its operands are removed from the runtime stack. After the operation, the result is again placed on the stack.

- Operations are type-specific; opcode prefixes are used, such as ‘*i*’ for an integer operation, ‘*c*’ for a character operation etc.

- Floating point data types and operations are compatible with IEEE Floating Point standard.

- Support is provided for *classes*, in terms of which JAVA programs are written.
  - For example, there is a JVM instruction to create a new instance of a class. This is a ‘higher level’ or ‘more abstract’ instruction than the instructions of *NICE* seen earlier.
  - Therefore, when such a JVM instruction is simulated on a hardware processor, it results in the execution of several machine instructions.

- Memory on JVM is not conceived as a linear sequence of bytes.
- Rather it is viewed as a collection of variables of various types, occupying different memory areas.
- Variables and objects of different types are located not by their binary memory addresses, but by *references* which are also type-specific.
- Multiple threads executing on JVM can access shared variables through the use of locks. Each thread has its own (simulated) ‘program counter’.
- A multi-threaded JVM program runs correctly on a processor, regardless of whether the processor has hardware support for multi-threading.

Intel 8051® Microcontroller Family

- *Microcontroller* - a processor with memory and I/O capability, in a compact package for embedded usage. Millions of microcontrollers are embedded in systems and appliances in use all around the world, providing improved functionality to users.

- Intel's 8051 and other 8-bit single-chip microcontrollers of the family – known as the MCS® 51 family – have been in use for well over two decades.

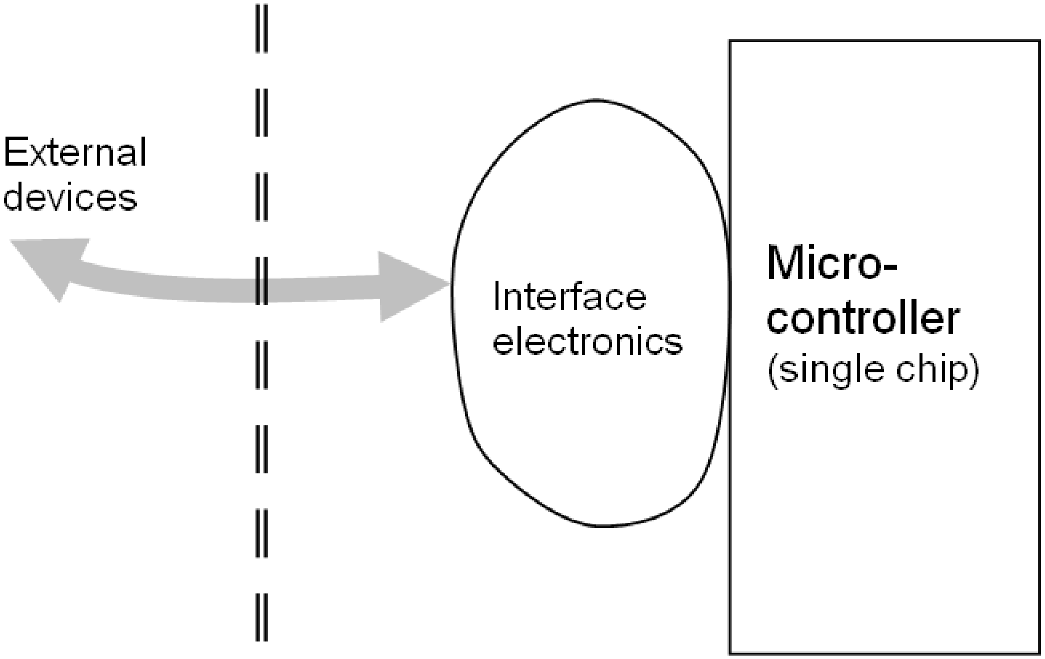
- There are several microcontrollers in the family – such as 8031, 8751 and 87C51 – with the same architecture and instruction set, but differences in on-chip memory and some other features.

- Other manufacturers have produced competing products with additional features, while maintaining pin-level and software compatibility with Intel 8051.

- The original 8051 microcontroller is produced in a 40-pin package. Of these:
  - two pins are used for power supply and ground,
  - two pins are used for connecting an external crystal, required to generate the processor clock signal.
  - other pins provide four bidirectional 8-bit I/O ports, and control signals for externally connected memory and devices.

- But this assignment leaves no pins for the 16 address lines and 8 data lines required for external memory and other devices.
- These external address and data lines are therefore not provided on separate pins, but share the same pins as two of the four 8-bit I/O ports.

- Interrupt lines, serial receive & transmit lines, and a few other control signals are also provided as *alternate functions* of pins of the third I/O port.



- 8751 microcontroller in this family also provides on-chip EPROM for the program. Figure shows the basic design of system based on such a microcontroller.

- In such a system, use of a microcontroller results in lower chip count, and therefore lower cost of production, and also compact and reliable design.

- Appliances are today produced, marketed, and serviced in very large quantities. Lower chip count, cost, and reliability can make the difference between a successful and a failed product.
- 8051 processor architecture has separate memory address space for the program and for data – each allowing for up to 64 kbytes of external memory through the 16 address lines and 8 data lines.
- Program memory – on-chip and external – is *read only*.
- Different members of the microcontroller family have on-chip RAM of either 128 bytes or 256 bytes. Of these, the lower 128 bytes include four banks of byte-wide registers R0 to R7.
  - Several addressing modes of the processor are designed specifically for the effective use of this on-chip RAM.

- Other useful functions – such as two 16-bit timers/counters and a full duplex serial communication port – are provided with the processor in the same package, as well as two external interrupt lines.

- The application program running on a microcontroller system is often organized in the form of multiple *tasks* (quite similar to threads) which are invoked through external or internal interrupts.

- Input and output devices connected to the system – such as measuring instruments or actuators – may require attention from the processor either in response to interrupts, or at regular time intervals, depending on the application.

- For faster access to registers and to cater to the needs of multiple tasks, the 8051 architecture provides four banks of 8-bit registers R0 to R7, within the lower 128 bytes of the on-chip RAM.

- This allows different register banks to be used by the different tasks in the application program.

- When execution switches from one task to another, the processor can be simply switched to use the appropriate register bank, to enable faster access to working variables of the task.

- Instruction set of the 8051 processor family is simple and compact – but it is definitely not orthogonal.

- To enable effective use of on-chip as well as external data and program memory, different addressing modes are provided for accessing data and also for transfers of control within the program.

- Microcontroller applications often require manipulation of individual bits which may reflect the status of various devices controlled by the system.

- For this, the 8051 instruction set provides a number of useful *bit manipulation* instructions. E.g. consider the following instruction:

```
LDB C, bit   //Load addressed bit into C
```

- Here **`bit`** is an eight-bit address pointing to one of 256 bits in the on-chip RAM which are bit-addressable.

- The instruction shown here copies – i.e. loads – the addressed bit into the carry bit C of the processor.

- Other instructions – such as **BB**, for branch on bit – allow transfer of control based on the addressed bit. Each input/output line of the four 8-bit I/O ports can also be read or written individually.
- Parameters of the timers/counters, serial port, etc. are set by the programmer in *configuration registers*.
- Low power versions of the processor – such as 87C51 – provide *power down* and *idle modes* of operation.
- Microcontroller application software can be developed either in assembly language or in C.
- Accordingly, a *cross-assembler* or a *cross-compiler* is used – i.e. one which runs on a PC, but produces a machine language program for the microcontroller, to be downloaded into it over the serial port.

The PC

- Computers were first developed in the 1940s; computer technology has since witnessed incredible advances.

- Around 1980, it became possible to build computers which could be considered ‘personal’.
- Features of such a computer: *low cost*, *compact size*, *ease of use and maintenance*, *choice of applications*, *choice of hardware add-ons*.

- In a knowledge-based society, a PC becomes an inseparable part of the user’s life.

- Many, many millions of PCs are in use around the world, and many millions have also gone out of use since the time the first PC was introduced.

- IBM introduced its PC in 1981, with Intel 8088 processor running at a clock speed of about 5 megahertz.
- The system could be operated with a minimum of 256 kilobytes of main memory.

- Microsoft developed DOS for the IBM PC, which was distributed and ran from 5 ¼” removable magnetic diskettes with a capacity of 360 kb each.

- There was virtually no graphics capability in the original PC, and no graphical user interface.

- Command line interface - still available on present-day PCs - was the only means of system interaction available to the user on the original PC.

- For the PC, IBM followed *open architecture* policy – by making its hardware design freely available.

- The aim was to encourage development of wide range of third party hardware and software for the PC.
- This would provide added value to users, and would also lead to increased demand for the PC.

- This policy worked very well – not only third-party hardware and software, but also third party PC *clones* became readily available in the ensuing years.

- The PC family has since prospered, evolved, and grown to an extent which could not have been predicted by any computer experts in 1981.

- Many key factors have played a role in this process, including − advances in VLSI technology; improved display, battery and packaging technology; lower costs; and almost unlimited range of applications.

- Today the PC is made and sold globally on competitive basis, and the demand amongst users has been growing steadily.
- Worldwide ‘PC market’ has seen thousands of product innovations and innovative products.

- The PC we use today is very different from the first IBM PC of 1981.
- In terms of any of the system elements − processor, memory, secondary storage, display, I/O, networking, or software − the advances have been tremendous.

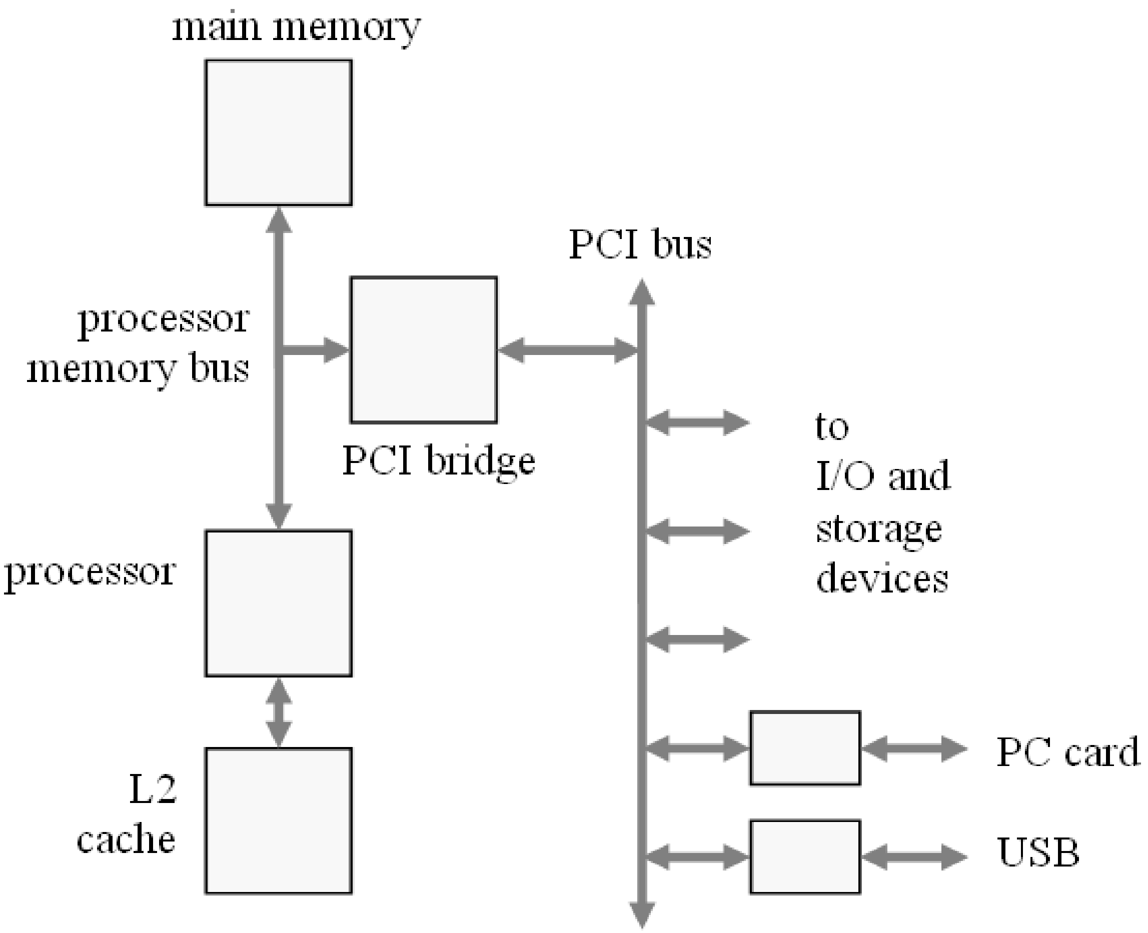
- IBM, and therefore also Intel and Microsoft, maintained full *backward compatibility* in the PC family.

- In theory, any program which ran under DOS operating system on the original PC should run on present-day PC.

- Processor in today's PC is a member of Intel's Pentium family (as seen above), or a compatible processor made by a company such as AMD.

- Memory, magnetic disk, optical disk, display, and other parts may be made by any of a large number of manufacturers around the world.

- PC itself may be assembled in another country, and the software may originate from several different countries.



- Figure shows typical organization of a PC.
- Main I/O bus, the *PCI bus* (*Peripheral Component Interconnect*) *was* developed by Intel in 1993.

- So-called *PCI bridge* is also shown in the figure.
- It ‘bridges’ the differences between the data transfer mechanisms of the two data busses connected to it.

- Thus the *bridge* makes it possible to send and receive data between devices connected to two different busses.
- In the figure, one of these is the high speed processor-memory bus, while the other is the PCI bus.

- Also seen is another high speed bus which connects the processor to the off-chip L2 cache.

- To connect a variety of I/O and storage devices, provision is made for the *PC card* interface, and also one or more *Universal Serial Bus* (USB) ports.

- The former is known as PCMCIA interface.
- It is provided so as to extend the PCI I/O capability for devices which can be plugged in and out by the user.

- USB has also been developed with the same objective – except that it is a serial interface, and therefore is more compact, less expensive, and provides somewhat lower data transfer speeds.

- The original *open architecture* concept of the PC has ensured that hundreds of manufacturers of PCs are in business today, in dozens of different countries around the world.
- But the originator IBM Corporation has since then left the PC business!

SUMMARY

- Throughput
- Intel Pentium® family of processors
- Sun UltraSparc T2® ‘system on a chip’
- JAVA Virtual Machine
- Intel 8051® microcontroller family
- The PC