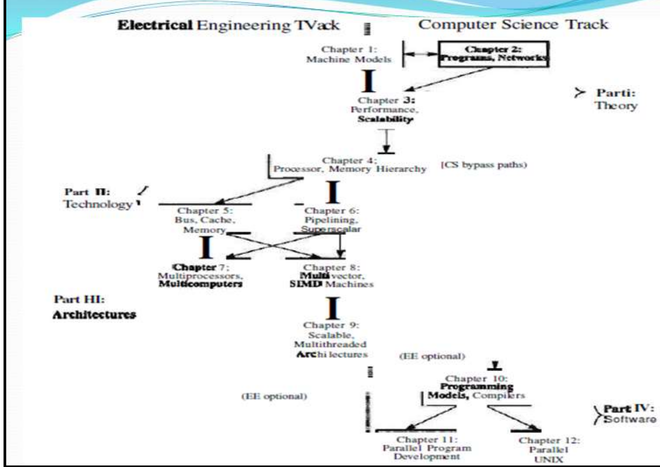
Chapter 1



Generation	Technology and Architecture	Software and Applications	Representative Systems
First (1945–54)	Vacuum tubes and relay memories, CPU driven by PC and accumulator, fixed-point arithmetic.	Machine/assembly languages, single user, no subroutine linkage, programmed I/O using CPU.	ENIAC, Princeton IAS, IBM 701.
Second (1955–64)	Discrete transistors and core memories, floating-point arithmetic, I/O processors, multiplexed memory access.	HLL used with compilers, subroutine libraries, batch processing monitor.	IBM 7090, CDC 1604, Univac LARC.
Third (1965–74)	Integrated circuits (SSI/-MSI), microprogramming, pipelining, cache, and lookahead processors.	Multiprogramming and time-sharing OS, multiuser applications.	IBM 360/370, CDC 6600, TI-ASC, PDP-8.
Fourth (1975–90)	LSI/VLSI and semiconductor memory, multiprocessors, vector supercomputers, multicomputers.	Multiprocessor OS, languages, compilers, and environments for parallel processing.	VAX 9000, Cray X-MP, IBM 3090, BBN TC2000.
Fifth (1991–present)	ULSI/VHSIC processors, memory, and switches, high-density packaging, scalable architectures.	Massively parallel processing, grand challenge applications, heterogeneous processing.	Fujitsu VPP500, Cray/MPP, TMC/CM-5, Intel Paragon.

Elements of a modern computer

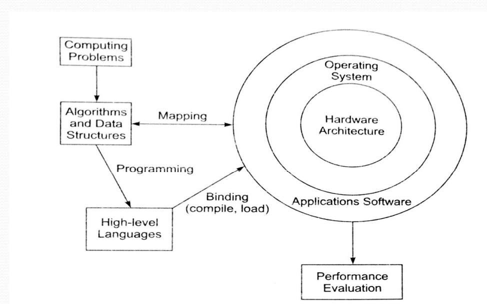


Fig. 1.1 Elements of a modern computer system

Elements of a modern computer

- **Computing problems**: the problems for which computer system should be constructed.
- **Algorithms and data structure**: Special algorithms and data structures are needed to specify the computations and communications involved in computing problems.
- **Hardware resources**: Processors, memory, peripheral devices.

- **Operating system**:
- **System software support**: Programs written in High level language. The source code translated into object code by a compiler.
- Compiler support: 3 compiler approaches:
- 1. Prepocessor: uses a sequential compiler.
- 2. Precompiler: requires some program flow analysis, dependence checking towards parallelism detection.
- 3. parallelizing compiler: demands a fully developed parallelizing compiler which can automatically detect parallelism.

Evolution of Computer Architecture

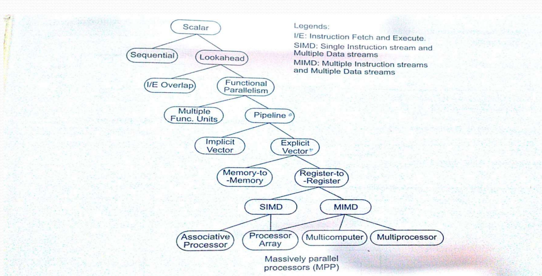
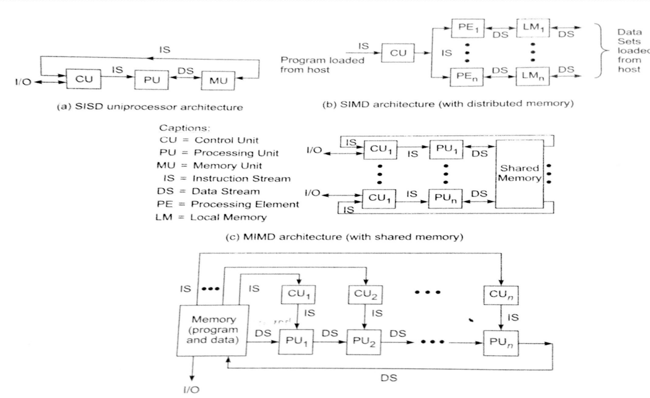


Fig. 1.2 Tree showing architectural evolution from sequential scalar computers to vector processors and parallel computers

FLYNN'S CLASSIFICATION



Parallel/Vector computers

- Execute programs on MIMD mode.
- 2 major classes:
- 1. shared-memory multiprocessors
- 2. message-passing multicomputers.

System attributes to performance

- Turn around Time :-It is the time which includes disk and memory access, input and output activities, compilation time , OS overhead and CPU time.
- Clock rate and CPI: processor is driven by a clock with a constant **cycle time** =**t**. The inverse of cycle time is the **clock rate (f=1/t)**.
- The size of the program is determined by its instruction count. (Ic). Different machine instructions may require different number of clock cycles to execute. Therefore CPI (cycles per instruction) becomes an important parameter.

Performance factors

- Ic= no. of instructions in a given program.
- Thus the CPU time needed to execute the program is by finding the product of three factors:

  T= Ic * CPI * t
- Instruction cycle requires cycle of events: instruction fetch, decode, operand fetch, execution and store results.

- In that cycle, only decode and execution phases are carried out in the CPU. The remaining three operations may require access to the memory.
- Therefore memory cycles is the time needed to complete one memory reference.
- Therefore CPI is divided into 2 components= processor cycles and memory cycles.
- Depending on the instruction type, the complete instruction cycle may involve one to as many as four memory references. (one for instruction fetch, two for operand fetch, one for storing result).

- Therefore T= Ic * (p + m*k) * t
- Ic= instruction count
- P= number of processor cycles.
- M= number of memory references
- K= ratio between memory and processor cycle.
- t= processor cycle time
- T=CPU Time

System Attributes

- The 5 performance factors are influenced by 4 system attributes:
- Instruction-set architecture
- Compiler technology
- CPU implementation and control
- Cache and memory hierarchy

Table 1.2 Performance Factors Versus System Attributes

System Attributes	Performance Factors				
	Instr. Count, I_c	Average Cycles per Instruction, CPI: Processor Cycles per Instruction, p	Memory References per Instruction, m	Memory-Access Latency, k	Processor Cycle Time, τ
Instruction-set Architecture	X	X			
Compiler Technology	X	X	X		
Processor Implementation and Control		X			X
Cache and Memory Hierarchy				X	X

- Instruction-set architecture= Ic, p (processor cycle per instruction)
- Compiler technology= Ic, p, m (memory references per instruction)
- CPU implementation and control= p, t (processor cycle time) total processor time needed.
- Cache and memory hierarchy= affects memory access latency = k, t

MIPS RATE

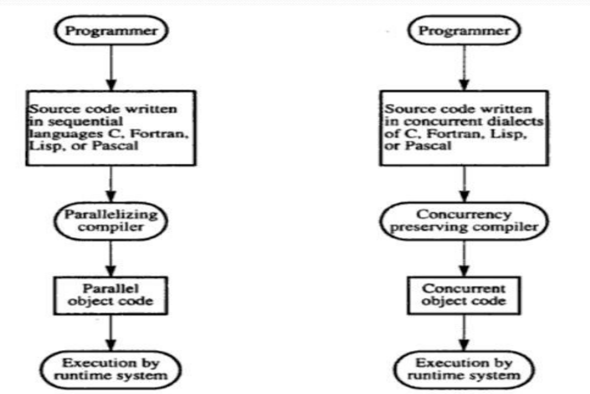
- MIPS (millions instructions per second)
- Throughput rate: how many programs can a system execute per unit time is called system throughput Ws.
- In multiprogrammed, the system throughput is often lower than the cpu throughput Wp.
- Because of additional system overheads caused by the i/o, complier and os.

MIPS

- Let C be the total number of clock cycles needed to execute a program.
- Therefore CPU time T= C*t= C/F
- Furthermore CPI= C/Ic
- T= CPI*Ic*t
- T= Ic*CPI/f
- The processor speed measured in MIPS.

- mips rate = Ic/T*10^6
- = f/CPI*10^6
- = f*Ic/ C*10^6
- F is clock rate/.
- Wp= f/Ic*CPI

Implicit and explicit parallelism



(a) Implicit parallelism (b) Explicit parallelism