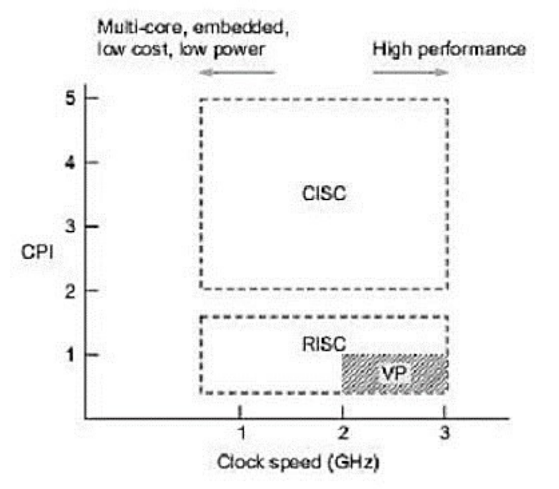
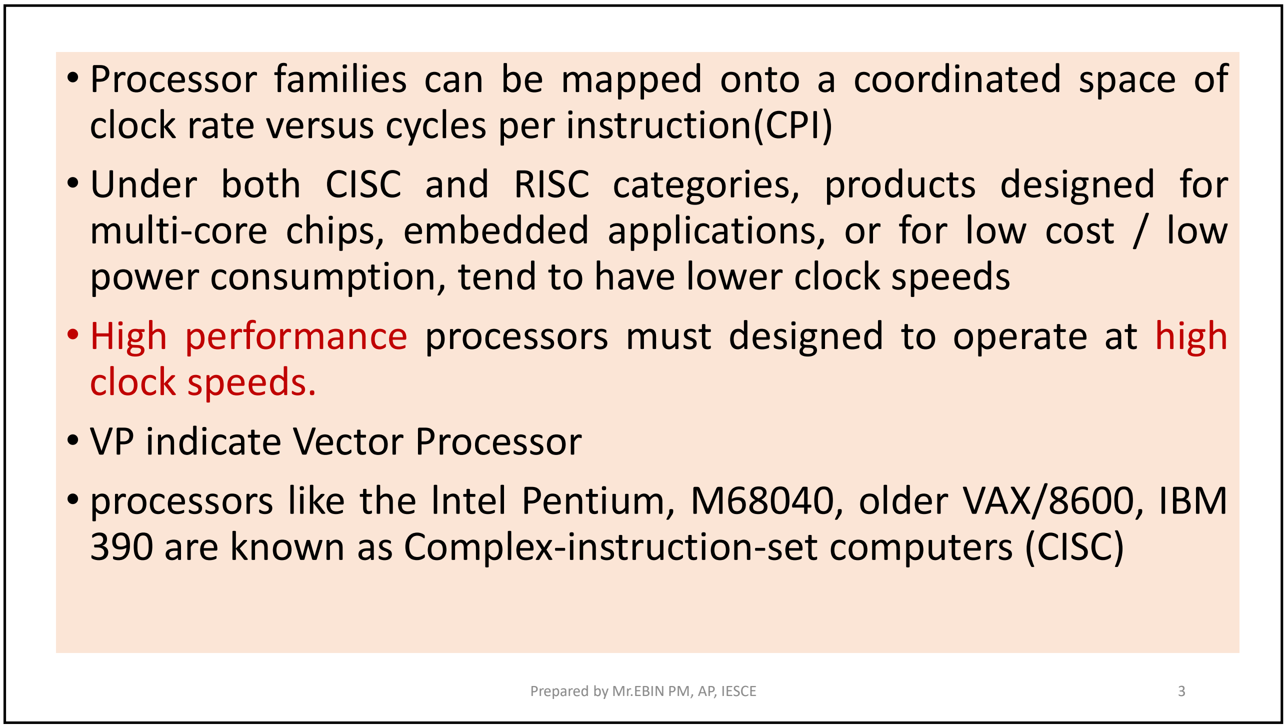
MODULE - 2

PROCESSORS & MEMORY HIERARCHY

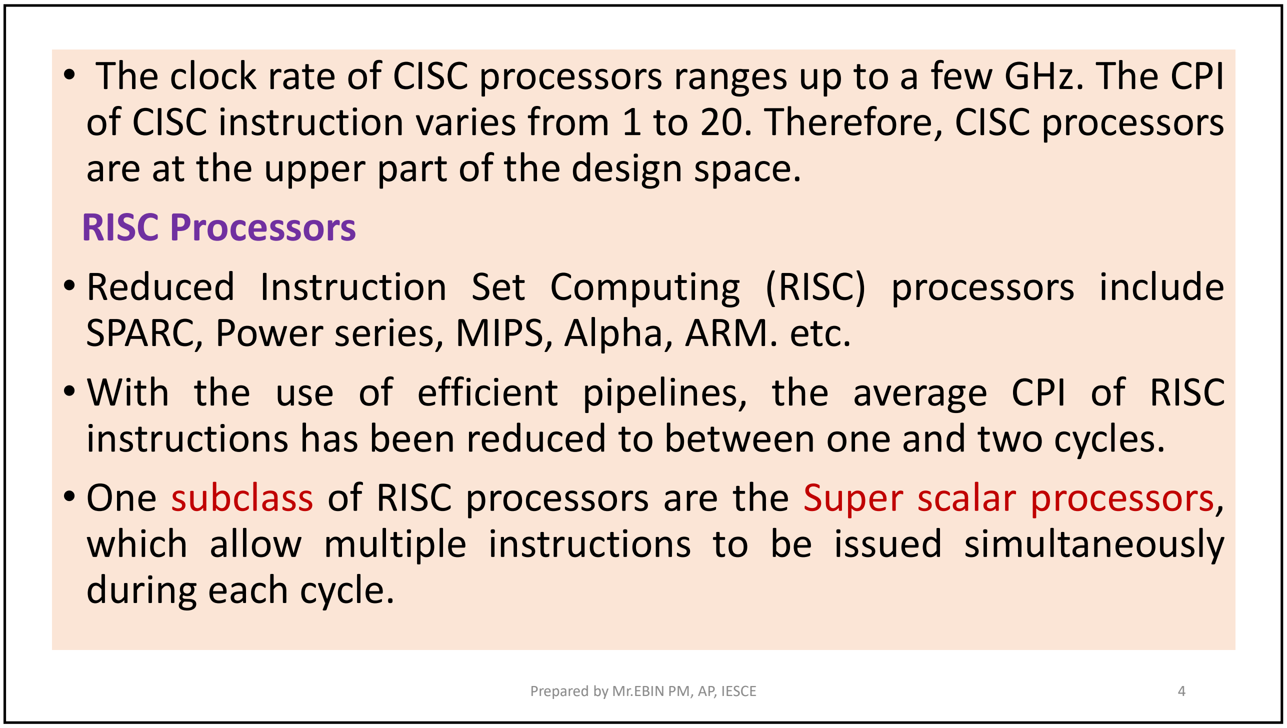
DESIGN SPACE OF PROCESSORS

CPI Vs processor clock speed of major categories of processors





- Processor families can be mapped onto a coordinated space of clock rate versus cycles per instruction(CPI)
- Under both CISC and RISC categories, products designed for multi-core chips, embedded applications, or for low cost / low power consumption, tend to have lower clock speeds
- High performance processors must designed to operate at high clock speeds.
- VP indicate Vector Processor
- processors like the Intel Pentium, M68040, older VAX/8600, IBM 390 are known as Complex-instruction-set computers (CISC)



- The clock rate of CISC processors ranges up to a few GHz. The CPI of CISC instruction varies from 1 to 20. Therefore, CISC processors are at the upper part of the design space.

RISC Processors

- Reduced Instruction Set Computing (RISC) processors include SPARC, Power series, MIPS, Alpha, ARM. etc.
- With the use of efficient pipelines, the average CPI of RISC instructions has been reduced to between one and two cycles.
- One subclass of RISC processors are the Super scalar processors, which allow multiple instructions to be issued simultaneously during each cycle.

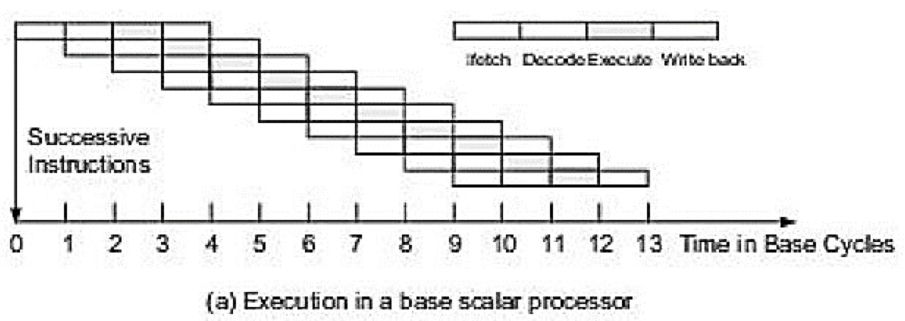
- Thus the effective CPI of a super scalar processor should be lower than that of a scalar RISC processor.

VLIW

- The Very-long instruction word (VLIW) architecture use more functional units than a super scalar processor.
- Thus the CPI of a VLIW processor can be further lowered.
- Eg: Intel‘s i860 RISC processor had VLIW architecture.

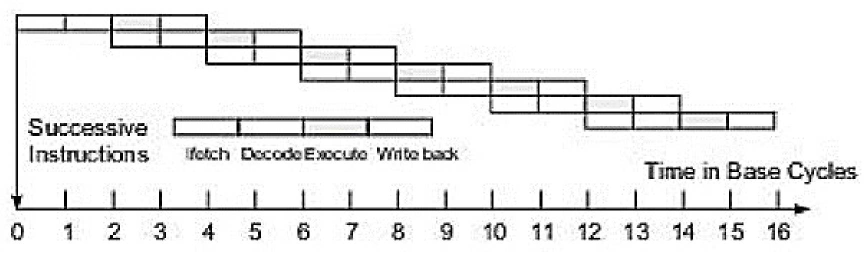
Instruction pipelines

- The execution cycle of a typical instruction includes four phases: fetch, decode, execute, and write-back.
- These instruction phases are often executed by an instruction pipeline



(a) Execution in a base scalar processor

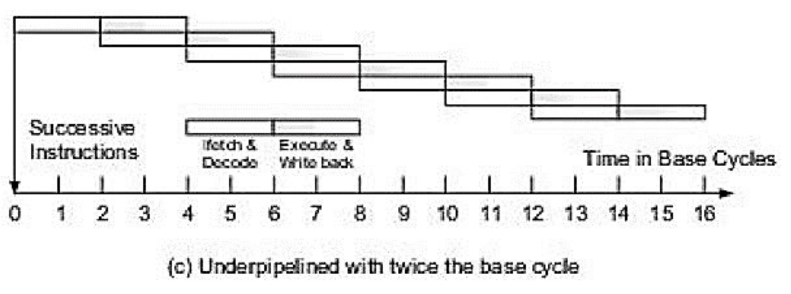
- A base scalar processor is defined as a machine with one instruction issued per cycle, a one-cycle latency for a simple operation, and a one-cycle latency between instruction issues.
- The instruction pipeline can be fully utilized if successive instructions can enter continuously at the rate of one per cycle, as shown in Fig (a)
- The instruction issue latency can be more than one cycle for various reasons.
- If the instruction issue latency is two cycles per instruction, the pipeline can be underutilized, as demonstrated in Fig(b).



(b) Underpipelined with two cycles per instruction issue

- Another underpipelined situation is shown in Fig(c), in which the pipeline cycle time is doubled by combining pipeline stages
- In this case, the fetch and decode phases are combined into one pipeline stage, and execute and write-back are combined into another stage

- This will also result in poor pipeline utilization

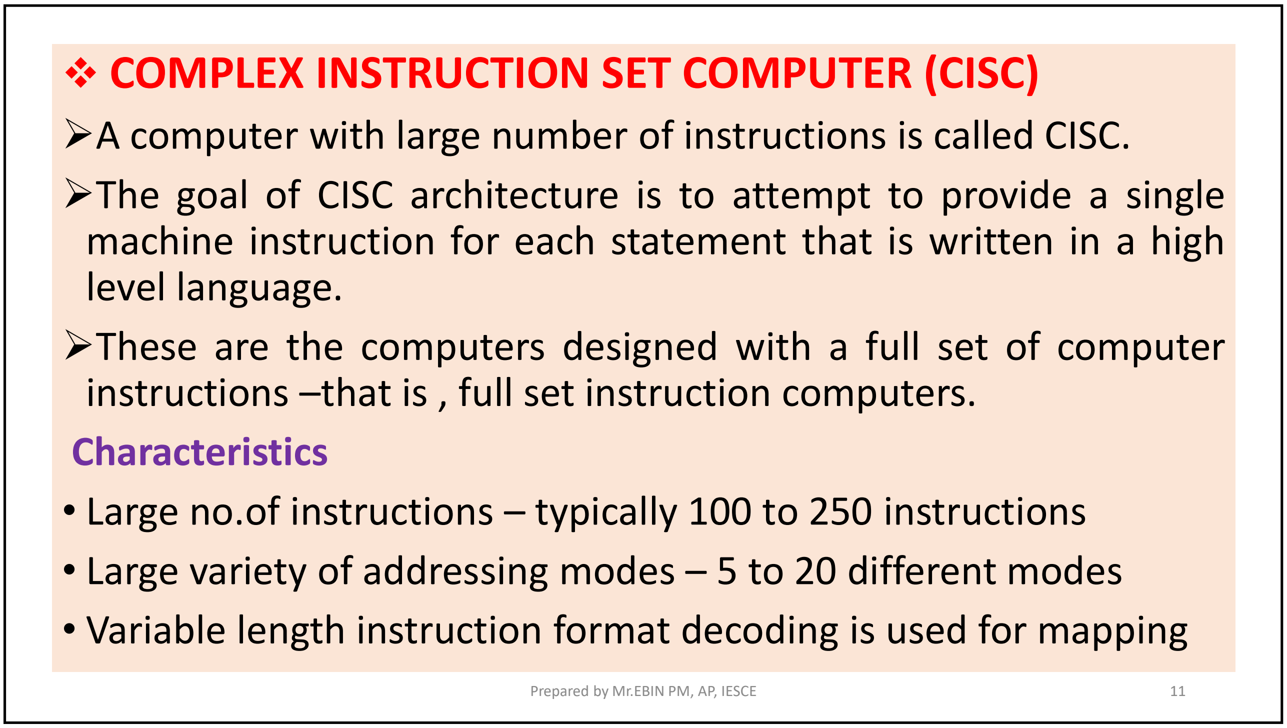


(c) Underpipelined with twice the base cycle

Some basic definitions associated with Instruction Pipeline

- Instruction pipeline cycle —the clock period of the instruction pipeline.
- Instruction issue latency —the time (in cycles) required between the issuing of two adjacent instructions.

- Instruction issue rate —the number of instructions issued per cycle, also called the degree of a superscalar processor.
- Simple operation latency — integer adds, loads, stores, branches, moves, etc. are simple operations. Complex operations requires longer latency such as divides, cache misses etc. These latencies are measured in number of cycles.
- Resource conflicts —This refers to the situation where two or more instructions demand use of the same functional unit at the same time.



❖ COMPLEX INSTRUCTION SET COMPUTER (CISC)

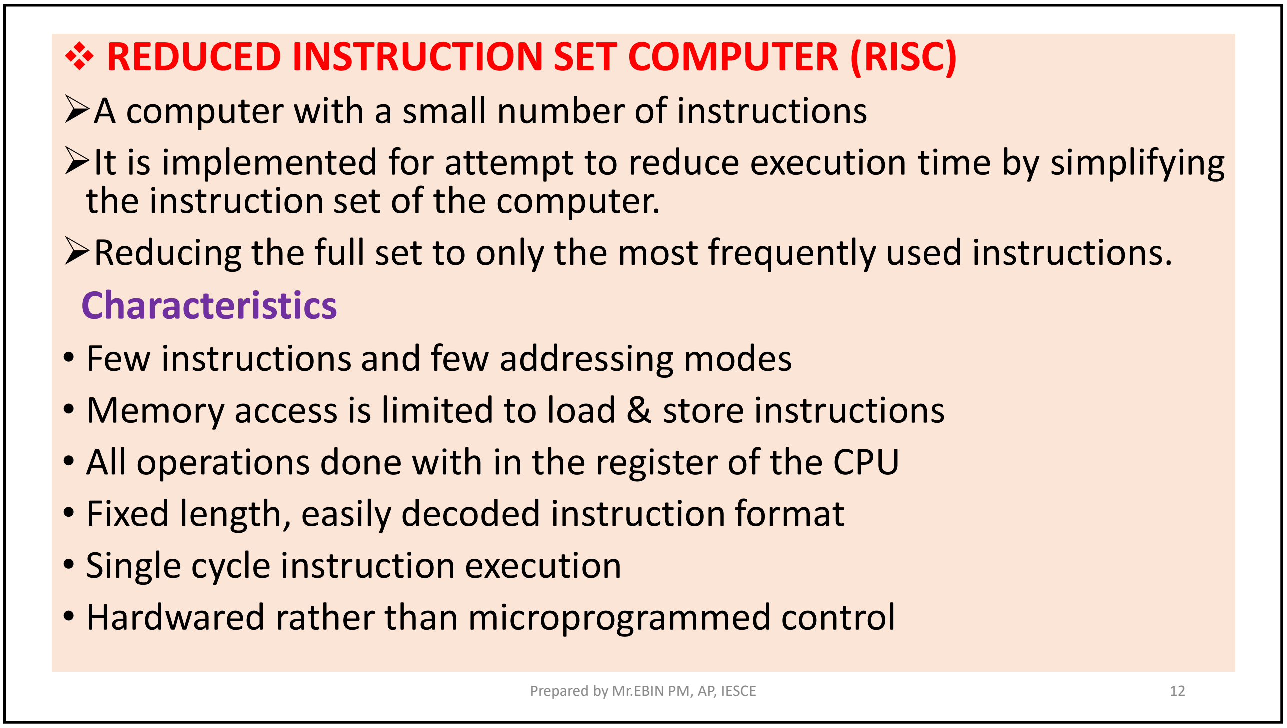
➢A computer with large number of instructions is called CISC.

➢The goal of CISC architecture is to attempt to provide a single machine instruction for each statement that is written in a high level language.

➢These are the computers designed with a full set of computer instructions –that is , full set instruction computers.

Characteristics

- Large no.of instructions – typically 100 to 250 instructions
- Large variety of addressing modes – 5 to 20 different modes
- Variable length instruction format decoding is used for mapping



❖ REDUCED INSTRUCTION SET COMPUTER (RISC)

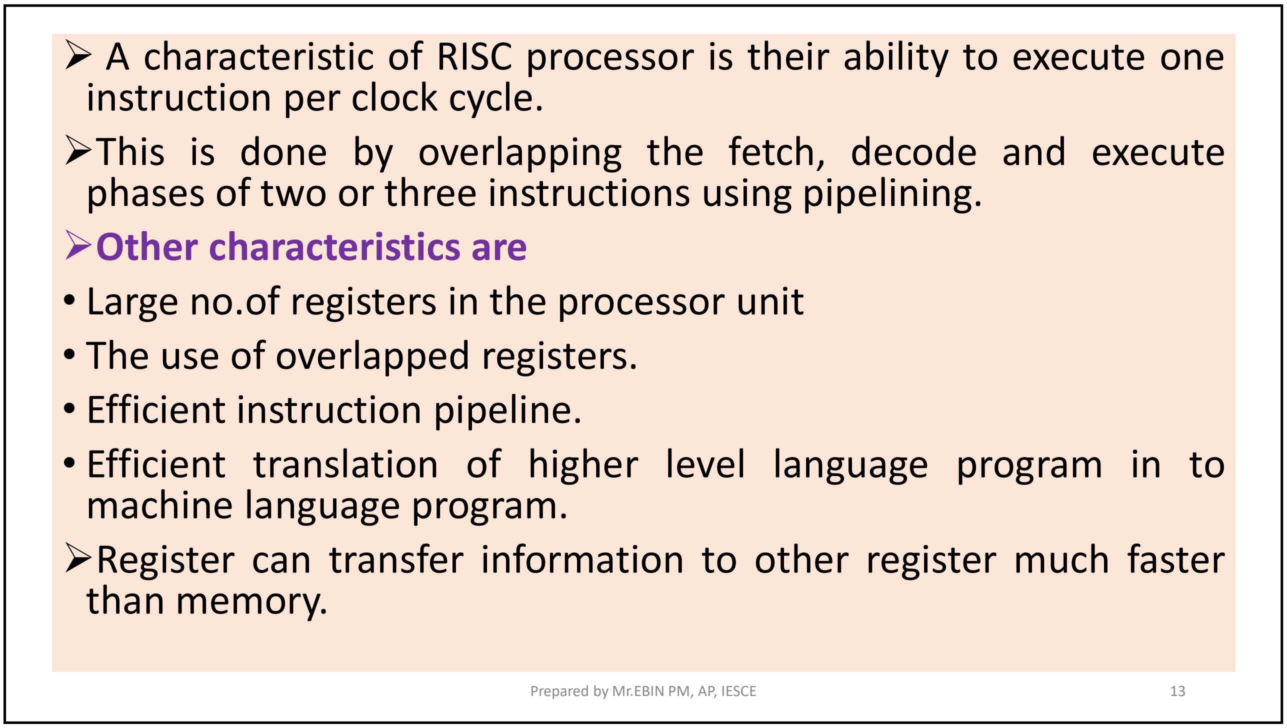
➢A computer with a small number of instructions

➢It is implemented for attempt to reduce execution time by simplifying the instruction set of the computer.

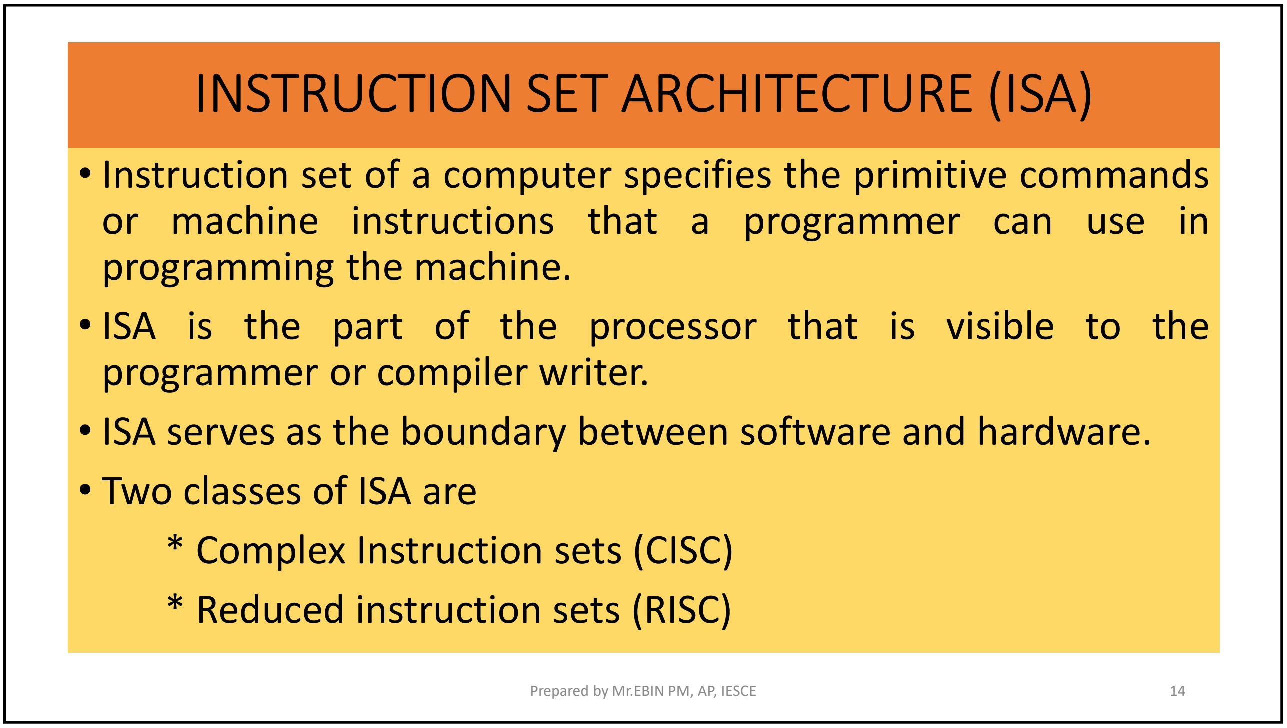
➢Reducing the full set to only the most frequently used instructions.

Characteristics

- Few instructions and few addressing modes
- Memory access is limited to load & store instructions
- All operations done with in the register of the CPU
- Fixed length, easily decoded instruction format
- Single cycle instruction execution
- Hardwared rather than microprogrammed control



- ➢ A characteristic of RISC processor is their ability to execute one instruction per clock cycle.
- ➢ This is done by overlapping the fetch, decode and execute phases of two or three instructions using pipelining.
- ➢ **Other characteristics are**
- Large no.of registers in the processor unit
- The use of overlapped registers.
- Efficient instruction pipeline.
- Efficient translation of higher level language program in to machine language program.
- ➢ Register can transfer information to other register much faster than memory.

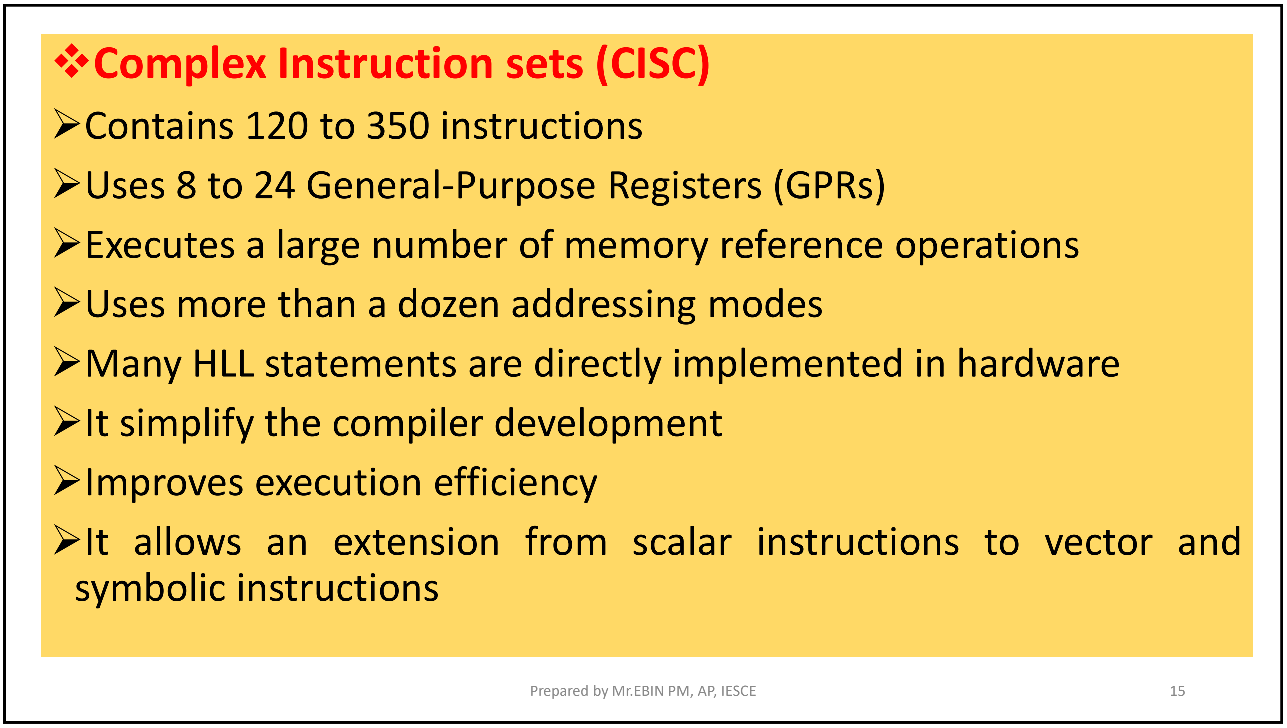


INSTRUCTION SET ARCHITECTURE (ISA)

- Instruction set of a computer specifies the primitive commands or machine instructions that a programmer can use in programming the machine.
- ISA is the part of the processor that is visible to the programmer or compiler writer.
- ISA serves as the boundary between software and hardware.
- Two classes of ISA are

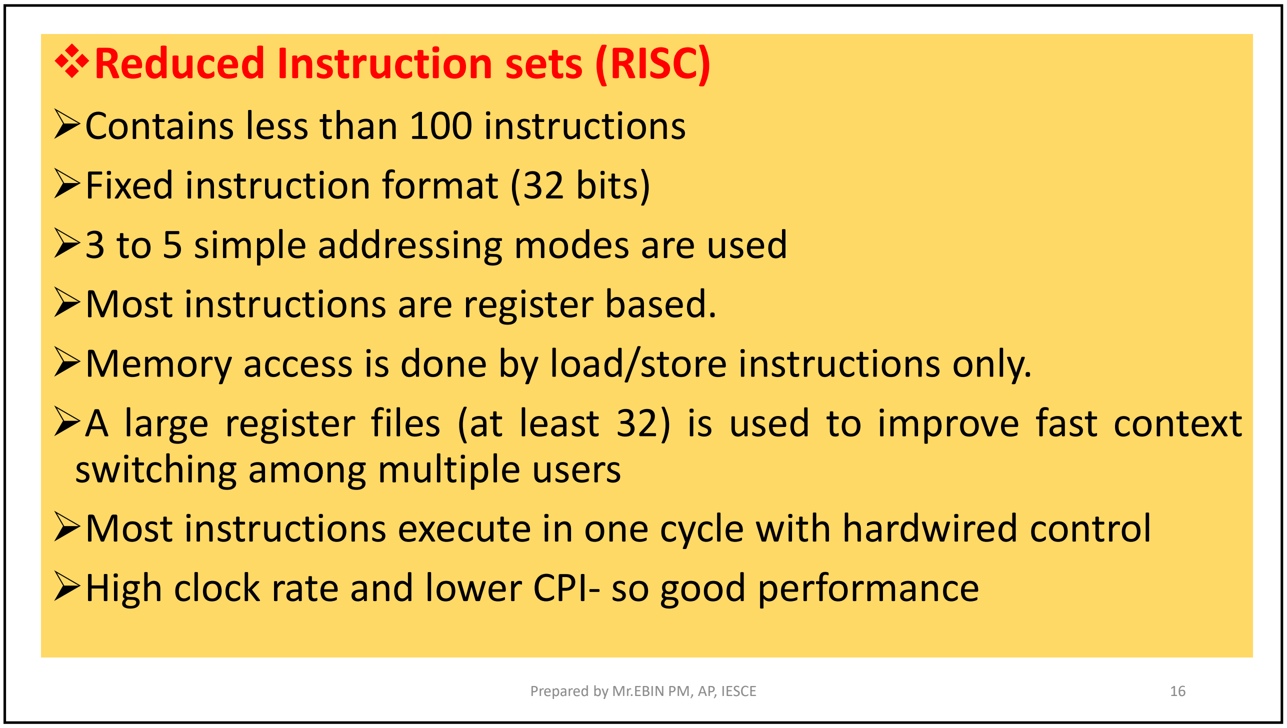
  * Complex Instruction sets (CISC)

  * Reduced instruction sets (RISC)



❖Complex Instruction sets (CISC)

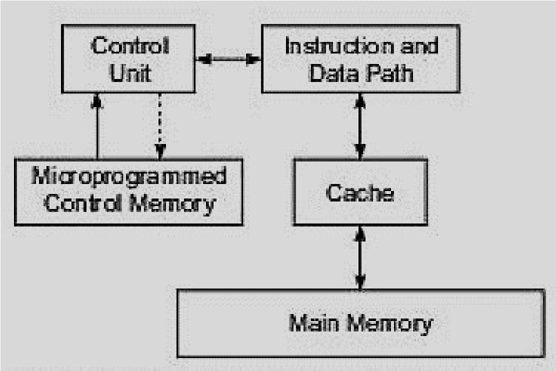
- ➢Contains 120 to 350 instructions
- ➢Uses 8 to 24 General-Purpose Registers (GPRs)
- ➢Executes a large number of memory reference operations
- ➢Uses more than a dozen addressing modes
- ➢Many HLL statements are directly implemented in hardware
- ➢It simplify the compiler development
- ➢Improves execution efficiency
- ➢It allows an extension from scalar instructions to vector and symbolic instructions



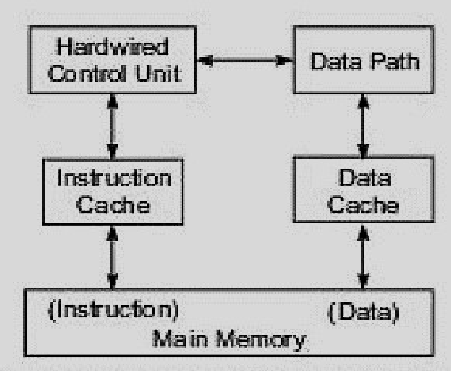
❖Reduced Instruction sets (RISC)

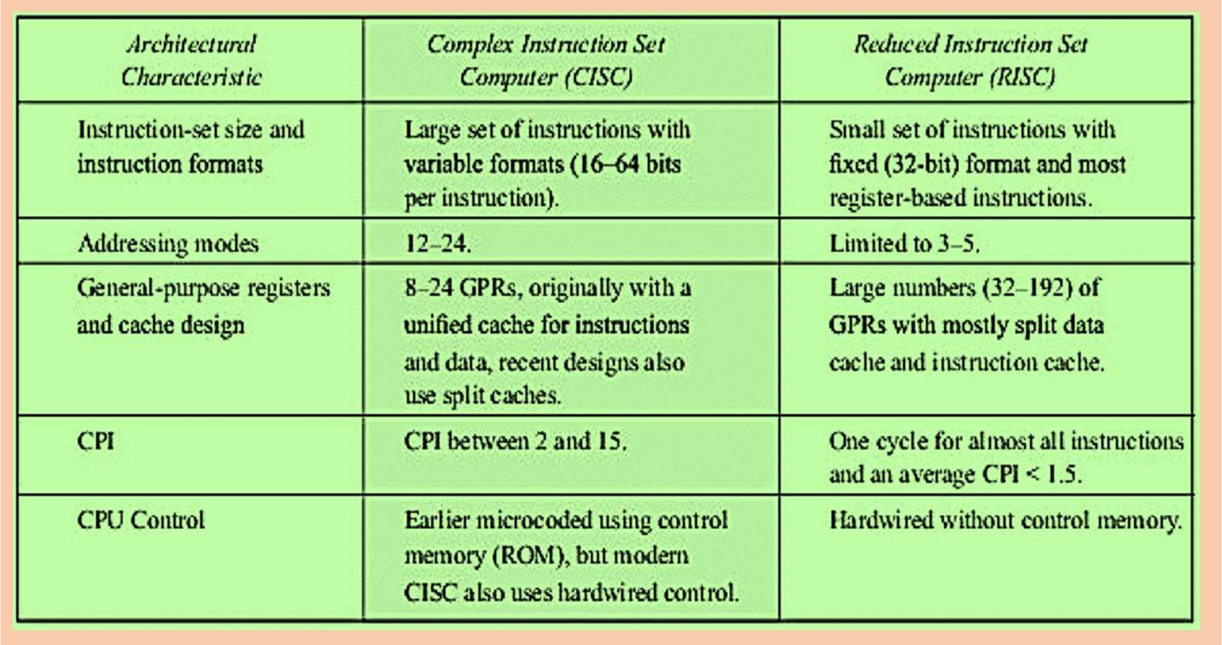
- ➢Contains less than 100 instructions
- ➢Fixed instruction format (32 bits)
- ➢3 to 5 simple addressing modes are used
- ➢Most instructions are register based.
- ➢Memory access is done by load/store instructions only.
- ➢A large register files (at least 32) is used to improve fast context switching among multiple users
- ➢Most instructions execute in one cycle with hardwired control
- ➢High clock rate and lower CPI- so good performance

CISC Processor Architecture

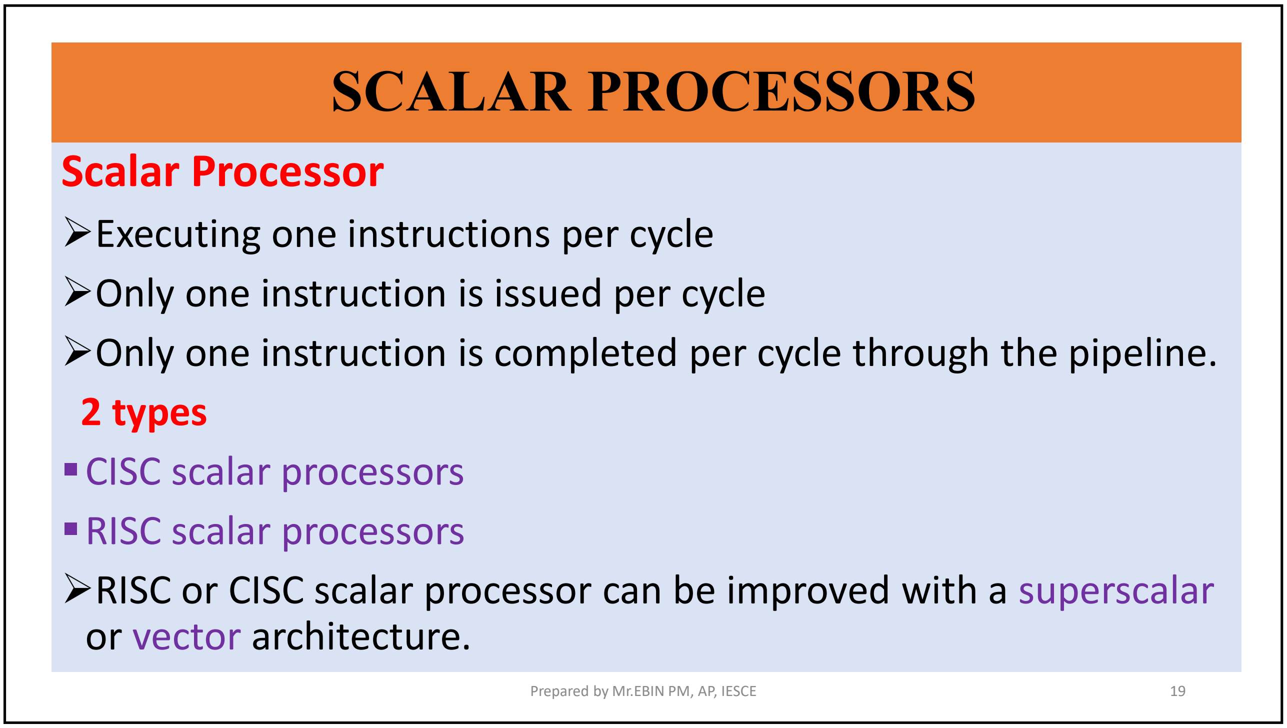


RISC Processor Architecture





Architectural Characteristic	*Complex Instruction Set Computer (CISC)*	*Reduced Instruction Set Computer (RISC)*
Instruction-set size and instruction formats	Large set of instructions with variable formats (16–64 bits per instruction).	Small set of instructions with fixed (32-bit) format and most register-based instructions.
Addressing modes	12–24.	Limited to 3–5.
General-purpose registers and cache design	8–24 GPRs, originally with a unified cache for instructions and data, recent designs also use split caches.	Large numbers (32–192) of GPRs with mostly split data cache and instruction cache.
CPI	CPI between 2 and 15.	One cycle for almost all instructions and an average CPI < 1.5.
CPU Control	Earlier microcoded using control memory (ROM), but modern CISC also uses hardwired control.	Hardwired without control memory.



SCALAR PROCESSORS

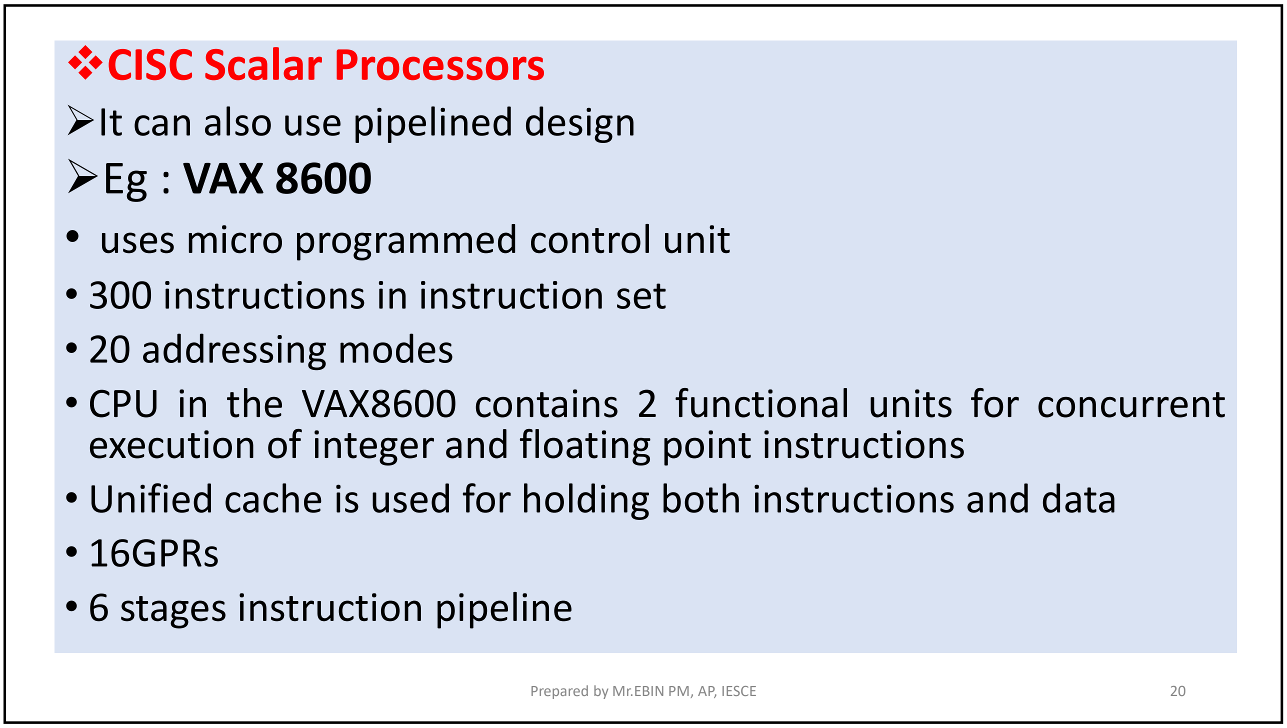
Scalar Processor

- ➢Executing one instructions per cycle
- ➢Only one instruction is issued per cycle
- ➢Only one instruction is completed per cycle through the pipeline.

2 types

- ▪ CISC scalar processors
- ▪ RISC scalar processors

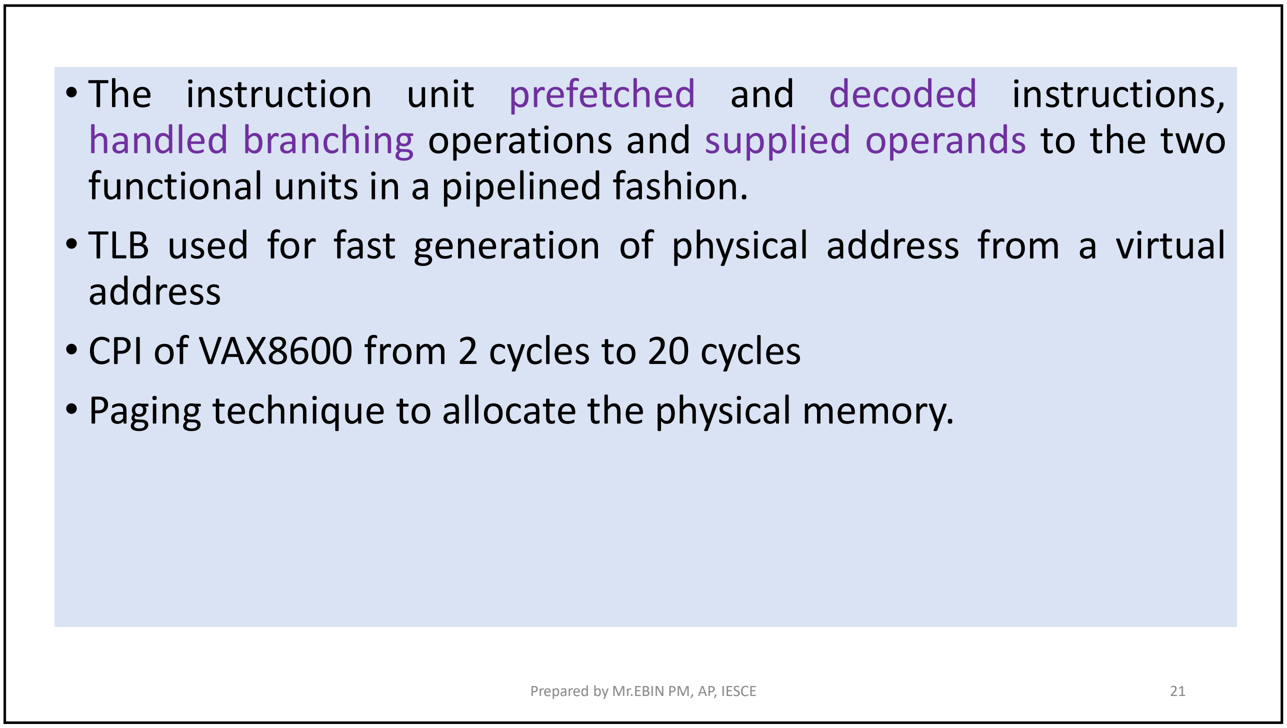
- ➢RISC or CISC scalar processor can be improved with a superscalar or vector architecture.



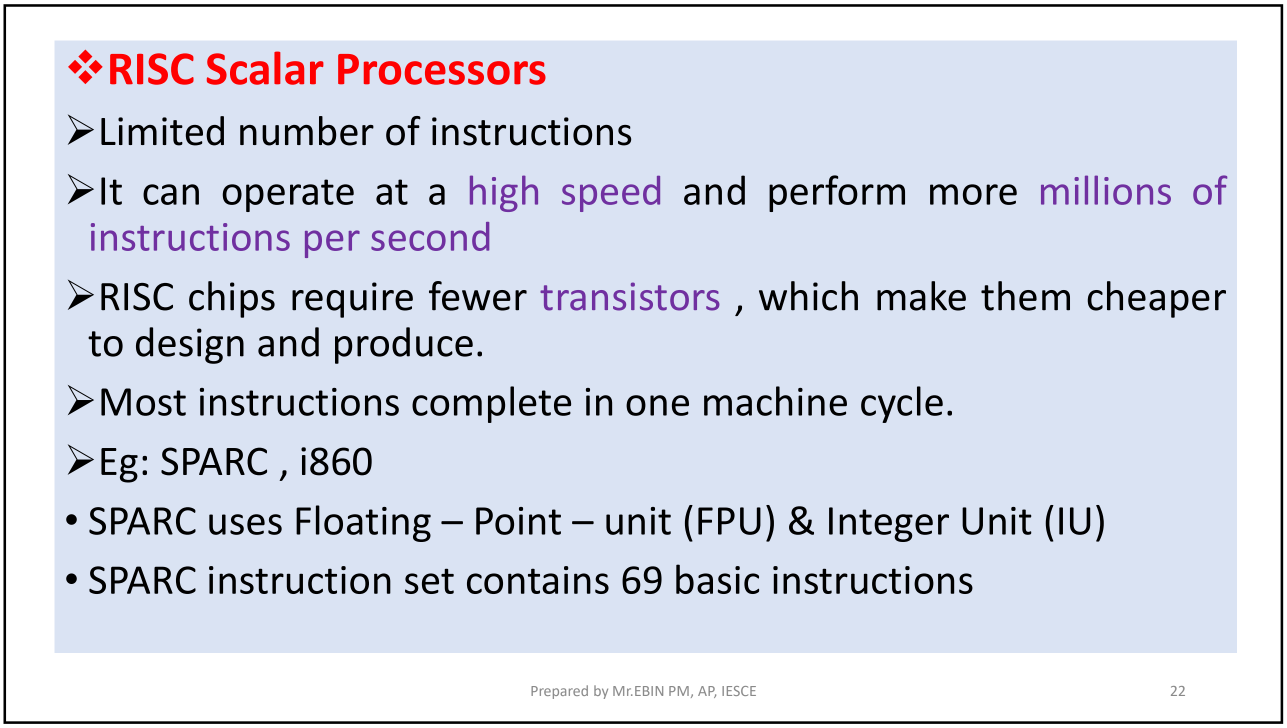
❖CISC Scalar Processors

- ➢It can also use pipelined design
- ➢Eg : **VAX 8600**

- uses micro programmed control unit
- 300 instructions in instruction set
- 20 addressing modes
- CPU in the VAX8600 contains 2 functional units for concurrent execution of integer and floating point instructions
- Unified cache is used for holding both instructions and data
- 16GPRs
- 6 stages instruction pipeline



- The instruction unit prefetched and decoded instructions, handled branching operations and supplied operands to the two functional units in a pipelined fashion.
- TLB used for fast generation of physical address from a virtual address
- CPI of VAX8600 from 2 cycles to 20 cycles
- Paging technique to allocate the physical memory.



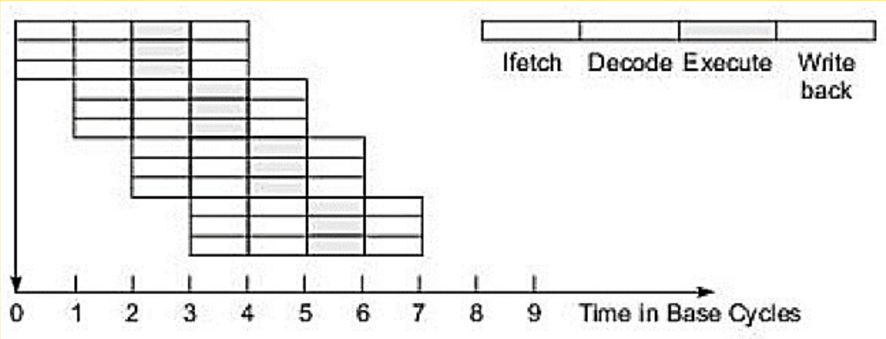
❖RISC Scalar Processors

- ➢Limited number of instructions
- ➢It can operate at a high speed and perform more millions of instructions per second
- ➢RISC chips require fewer transistors , which make them cheaper to design and produce.
- ➢Most instructions complete in one machine cycle.
- ➢Eg: SPARC , i860
- SPARC uses Floating – Point – unit (FPU) & Integer Unit (IU)
- SPARC instruction set contains 69 basic instructions

SUPERSCALAR PROCESSORS

- Multiple instructions are issued per cycle
- Multiple results are generated per cycle
- Exploit more instruction level parallelism in user programs
- Only independent instructions can be executed in parallel without causing a wait state
- Instruction issue degree is limited to 2 to 5
- Super scalar processor of degree ‘m’ can issue ‘m’ instructions per cycle

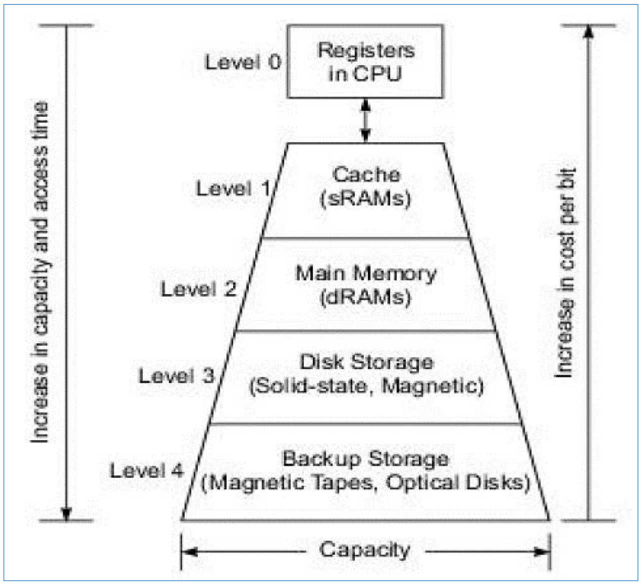
(A superscalar processor of degree m = 3)

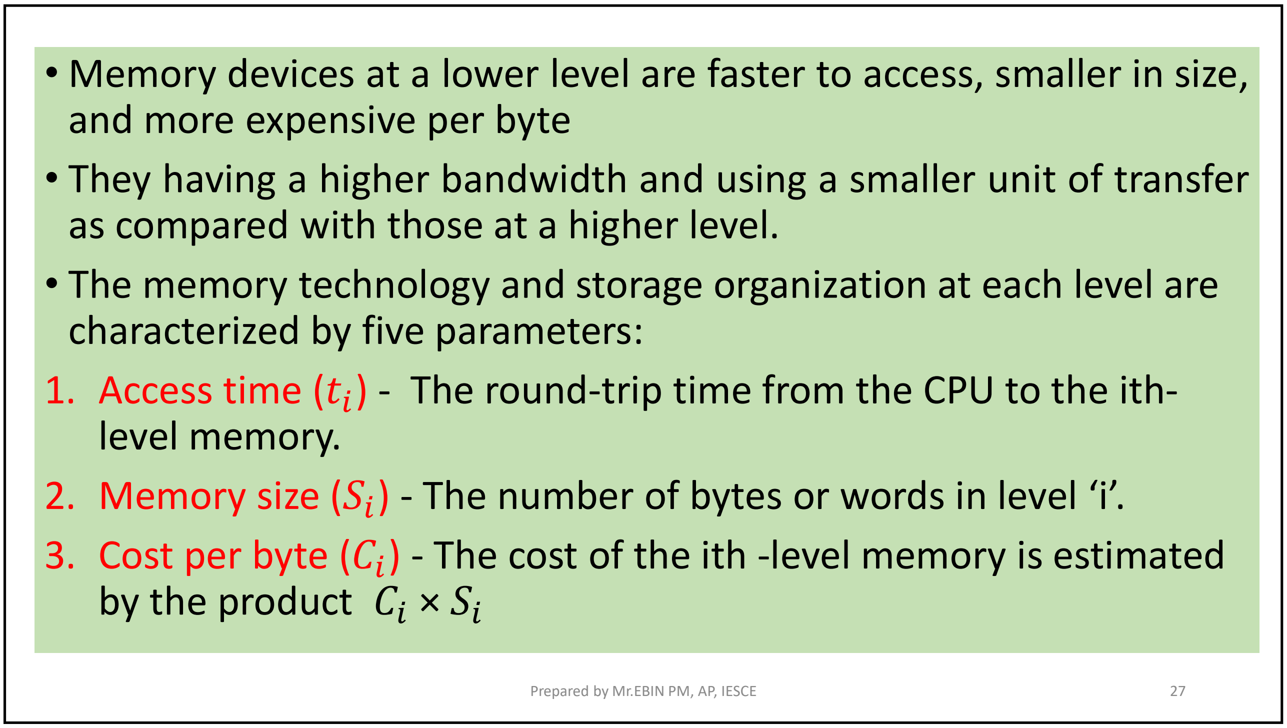


- For getting higher degree of instruction level parallelism, super scalar processors depends an optimizing compiler to exploit parallelism.

- IBM RS/6000 is a super scalar RISC processor
- The maximum number of instructions issued per cycle ranges from 2 to 5
- The register files in the IU and FPU each have 32 registers
- For supporting instruction look-ahead and internal data forwarding, "reservation stations" and "reorder buffers" are used to establish instruction windows
- RS/6000 uses hardwired control unit

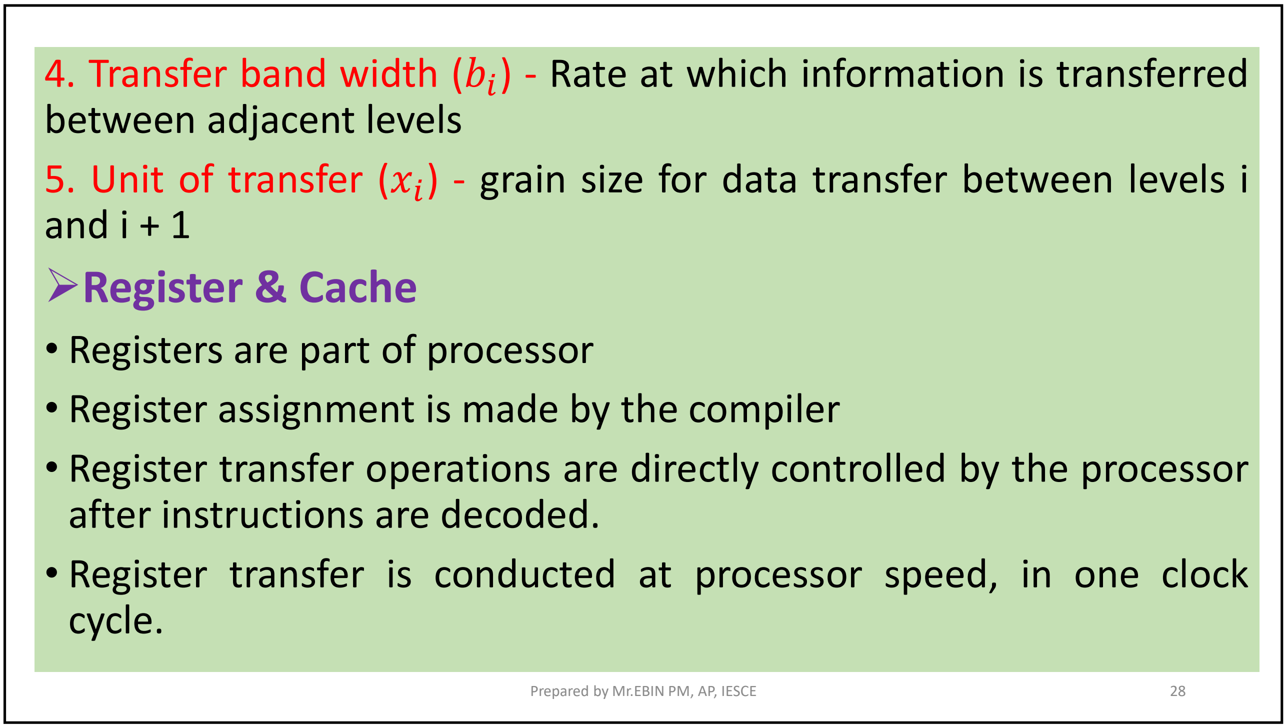
MEMORY HIERARCHY TECHNOLOGY


Level 0
Registers in CPU
Level 1
Cache (sRAMs)
Level 2
Main Memory (dRAMs)
Level 3
Disk Storage (Solid-state, Magnetic)
Level 4
Backup Storage (Magnetic Tapes, Optical Disks)
Increase in capacity and access time
Increase in cost per bit
Capacity



- Memory devices at a lower level are faster to access, smaller in size, and more expensive per byte
- They having a higher bandwidth and using a smaller unit of transfer as compared with those at a higher level.
- The memory technology and storage organization at each level are characterized by five parameters:

1. Access time (t_i) - The round-trip time from the CPU to the ith-level memory.
2. Memory size (S_i) - The number of bytes or words in level ‘i’.
3. Cost per byte (C_i) - The cost of the ith -level memory is estimated by the product $C_i \times S_i$

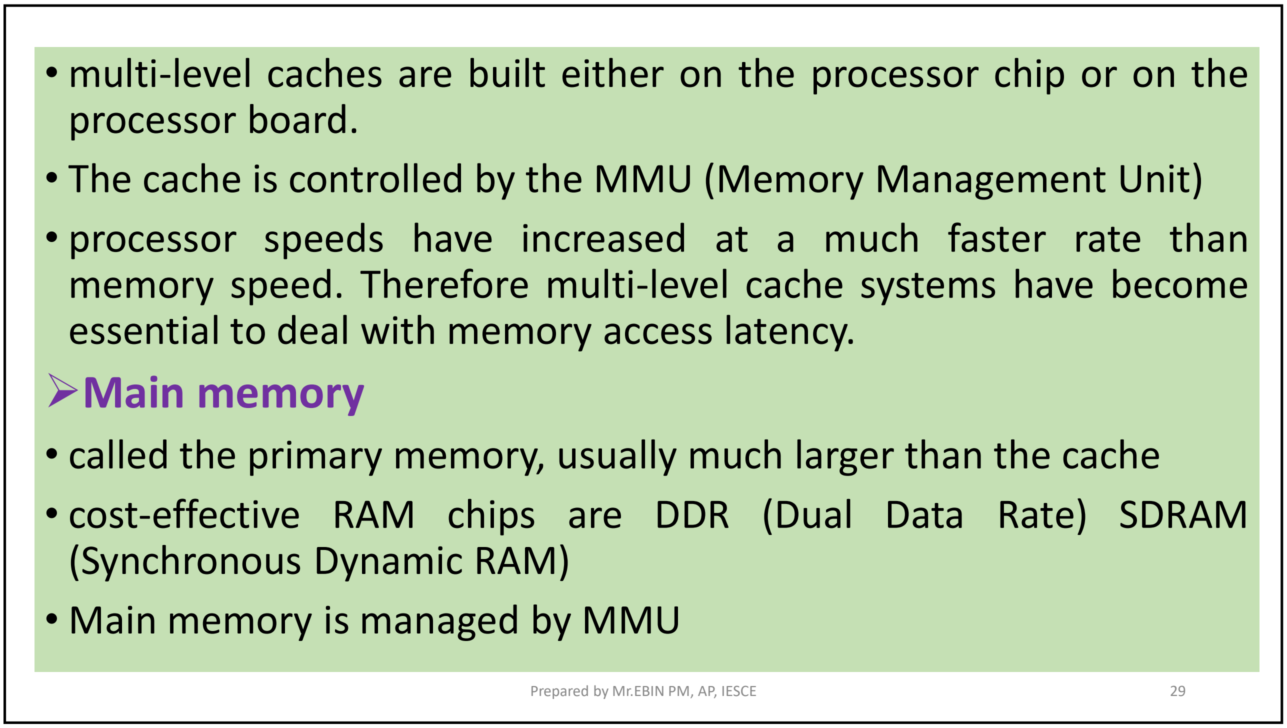


4. Transfer band width (b_i) - Rate at which information is transferred between adjacent levels

5. Unit of transfer (x_i) - grain size for data transfer between levels i and i + 1

➢Register & Cache

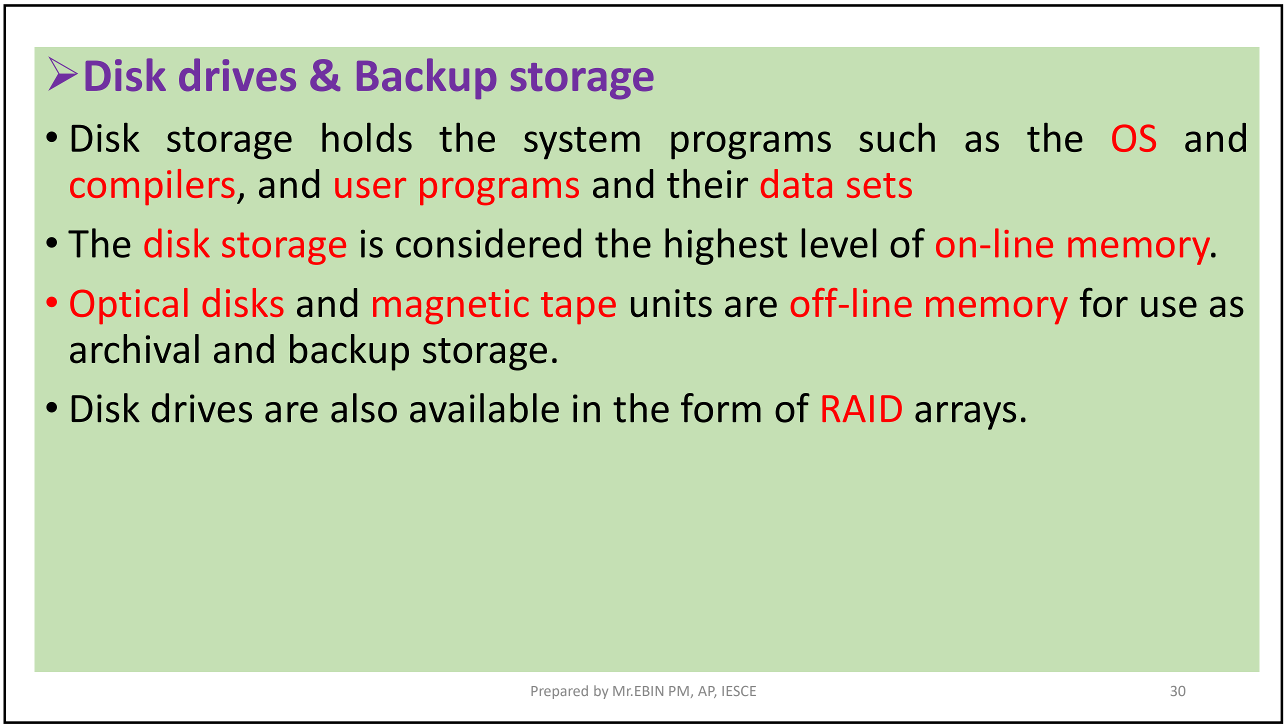
- Registers are part of processor
- Register assignment is made by the compiler
- Register transfer operations are directly controlled by the processor after instructions are decoded.
- Register transfer is conducted at processor speed, in one clock cycle.



- multi-level caches are built either on the processor chip or on the processor board.
- The cache is controlled by the MMU (Memory Management Unit)
- processor speeds have increased at a much faster rate than memory speed. Therefore multi-level cache systems have become essential to deal with memory access latency.

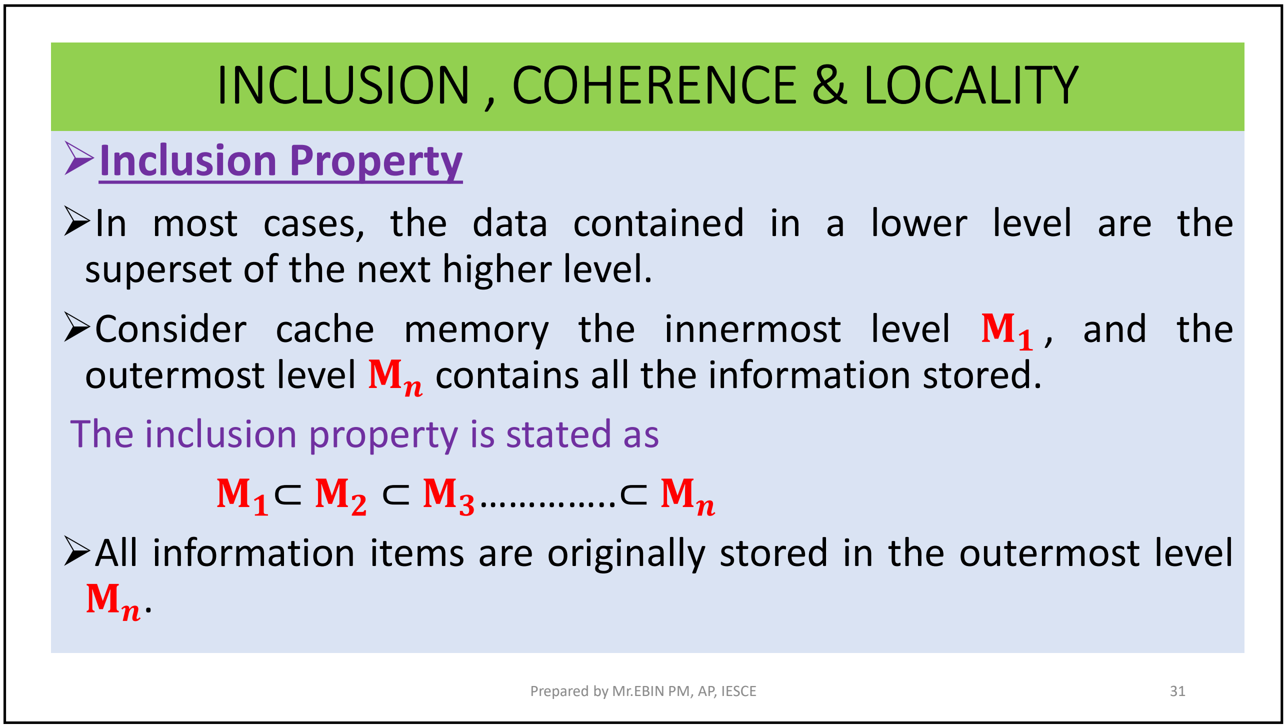
➢Main memory

- called the primary memory, usually much larger than the cache
- cost-effective RAM chips are DDR (Dual Data Rate) SDRAM (Synchronous Dynamic RAM)
- Main memory is managed by MMU



➢Disk drives & Backup storage

- Disk storage holds the system programs such as the OS and compilers, and user programs and their data sets
- The disk storage is considered the highest level of on-line memory.
- Optical disks and magnetic tape units are off-line memory for use as archival and backup storage.
- Disk drives are also available in the form of RAID arrays.



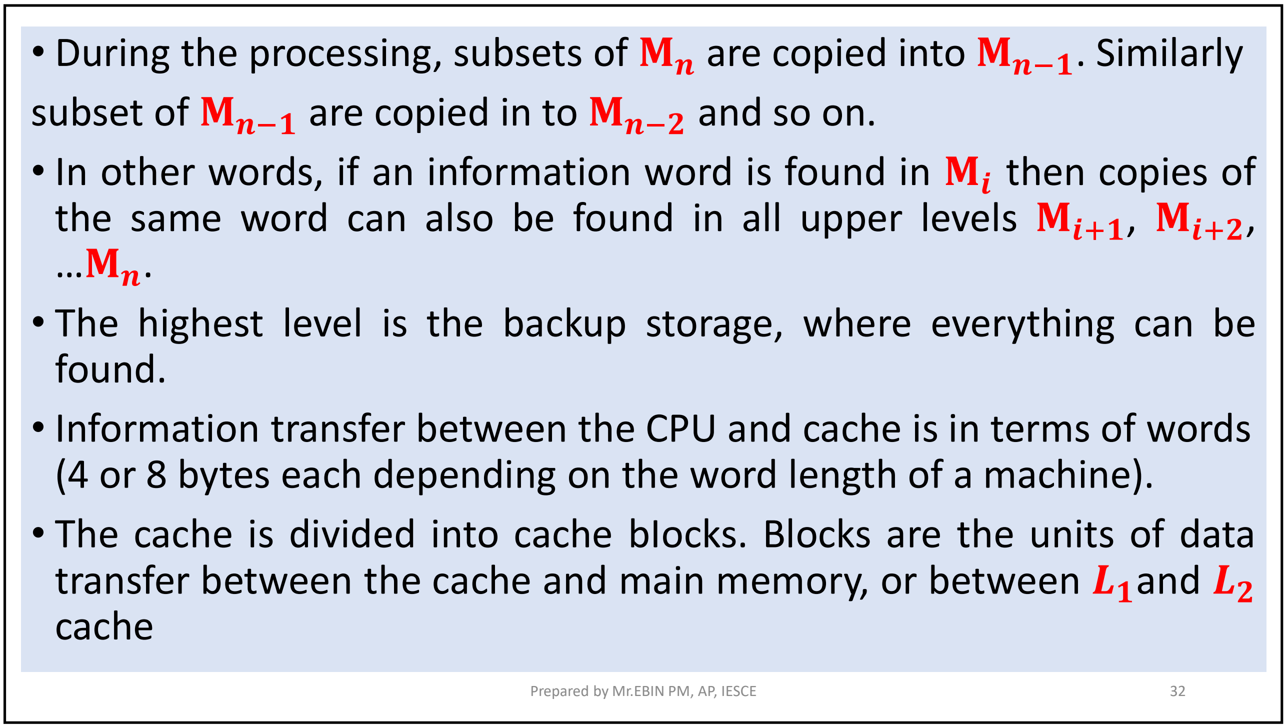
INCLUSION , COHERENCE & LOCALITY

- **Inclusion Property**
- In most cases, the data contained in a lower level are the superset of the next higher level.
- Consider cache memory the innermost level $\mathbf{M}_1$, and the outermost level $\mathbf{M}_n$ contains all the information stored.

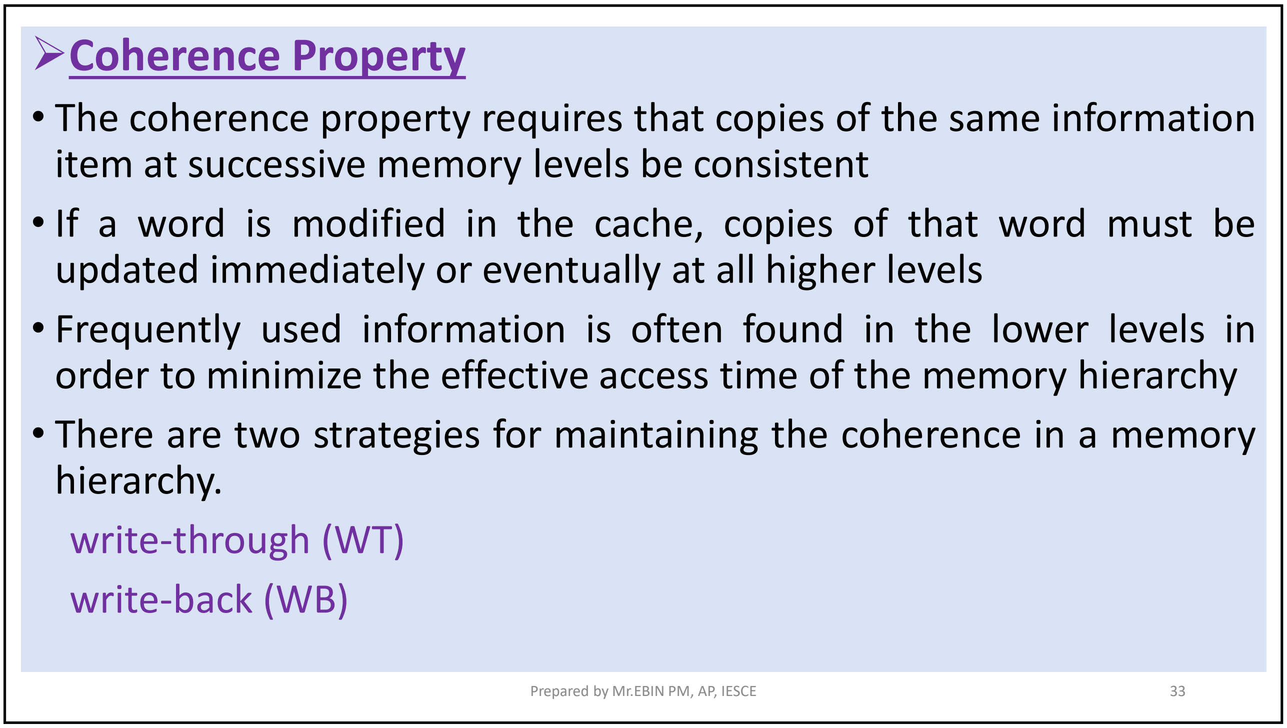
The inclusion property is stated as

$$\mathbf{M}_1 \subset \mathbf{M}_2 \subset \mathbf{M}_3 \ldots\ldots\ldots\ldots \subset \mathbf{M}_n$$

- All information items are originally stored in the outermost level $\mathbf{M}_n$.



- During the processing, subsets of $\mathbf{M}_n$ are copied into $\mathbf{M}_{n-1}$. Similarly subset of $\mathbf{M}_{n-1}$ are copied in to $\mathbf{M}_{n-2}$ and so on.
- In other words, if an information word is found in $\mathbf{M}_i$ then copies of the same word can also be found in all upper levels $\mathbf{M}_{i+1}$, $\mathbf{M}_{i+2}$, …$\mathbf{M}_n$.
- The highest level is the backup storage, where everything can be found.
- Information transfer between the CPU and cache is in terms of words (4 or 8 bytes each depending on the word length of a machine).
- The cache is divided into cache blocks. Blocks are the units of data transfer between the cache and main memory, or between L_1 and L_2 cache

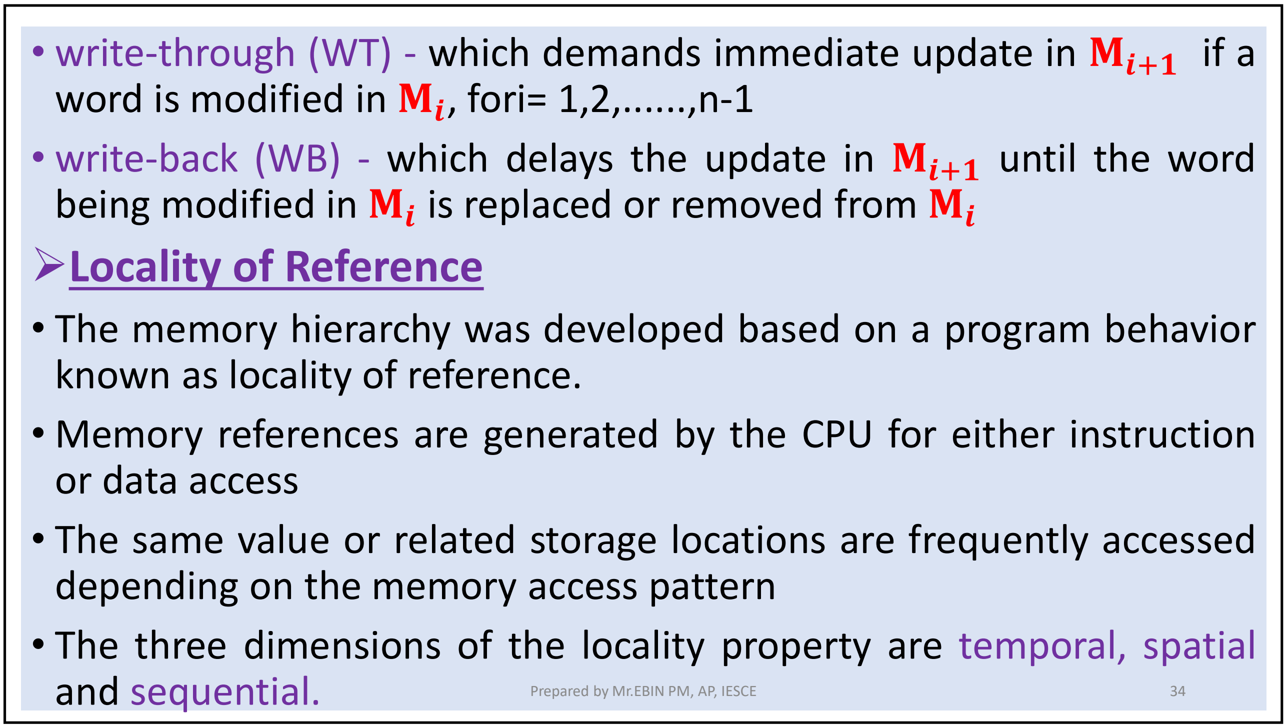


➢Coherence Property

- The coherence property requires that copies of the same information item at successive memory levels be consistent
- If a word is modified in the cache, copies of that word must be updated immediately or eventually at all higher levels
- Frequently used information is often found in the lower levels in order to minimize the effective access time of the memory hierarchy
- There are two strategies for maintaining the coherence in a memory hierarchy.

  write-through (WT)

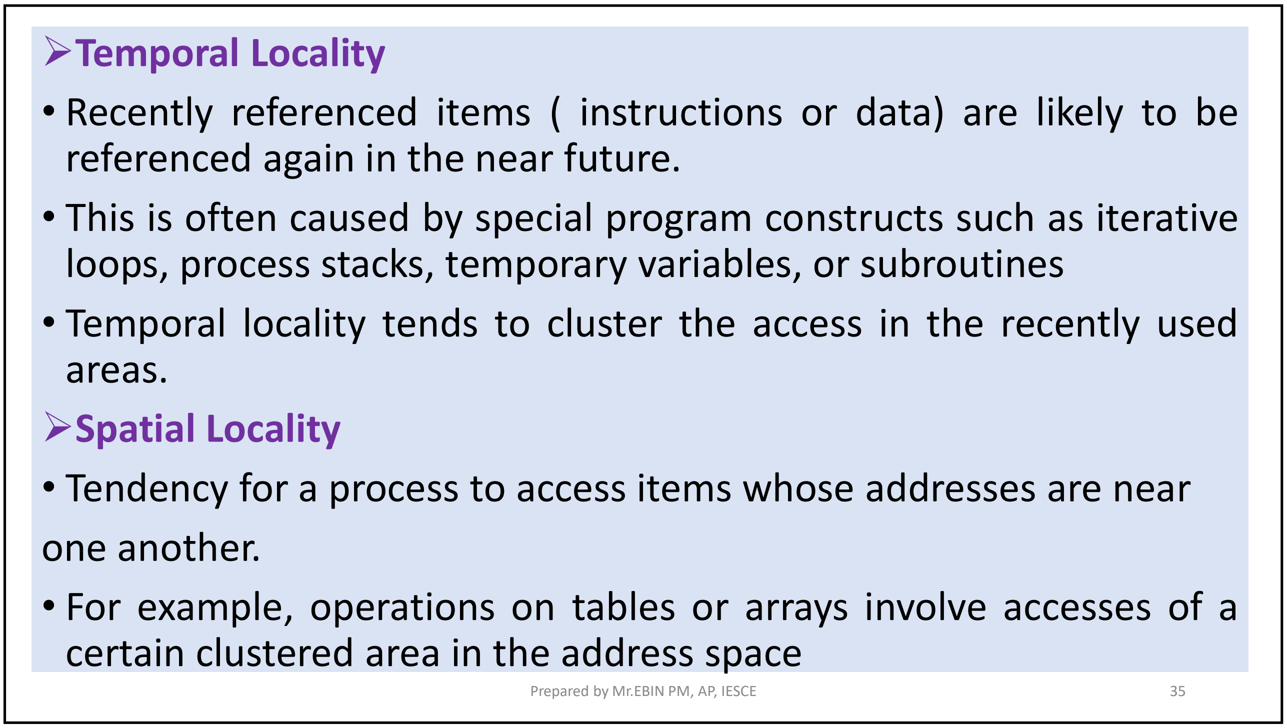
  write-back (WB)



- write-through (WT) - which demands immediate update in $\mathbf{M}_{i+1}$ if a word is modified in $\mathbf{M}_i$, fori= 1,2,......,n-1
- write-back (WB) - which delays the update in $\mathbf{M}_{i+1}$ until the word being modified in $\mathbf{M}_i$ is replaced or removed from $\mathbf{M}_i$

➢Locality of Reference

- The memory hierarchy was developed based on a program behavior known as locality of reference.
- Memory references are generated by the CPU for either instruction or data access
- The same value or related storage locations are frequently accessed depending on the memory access pattern
- The three dimensions of the locality property are temporal, spatial and sequential.



➢Temporal Locality

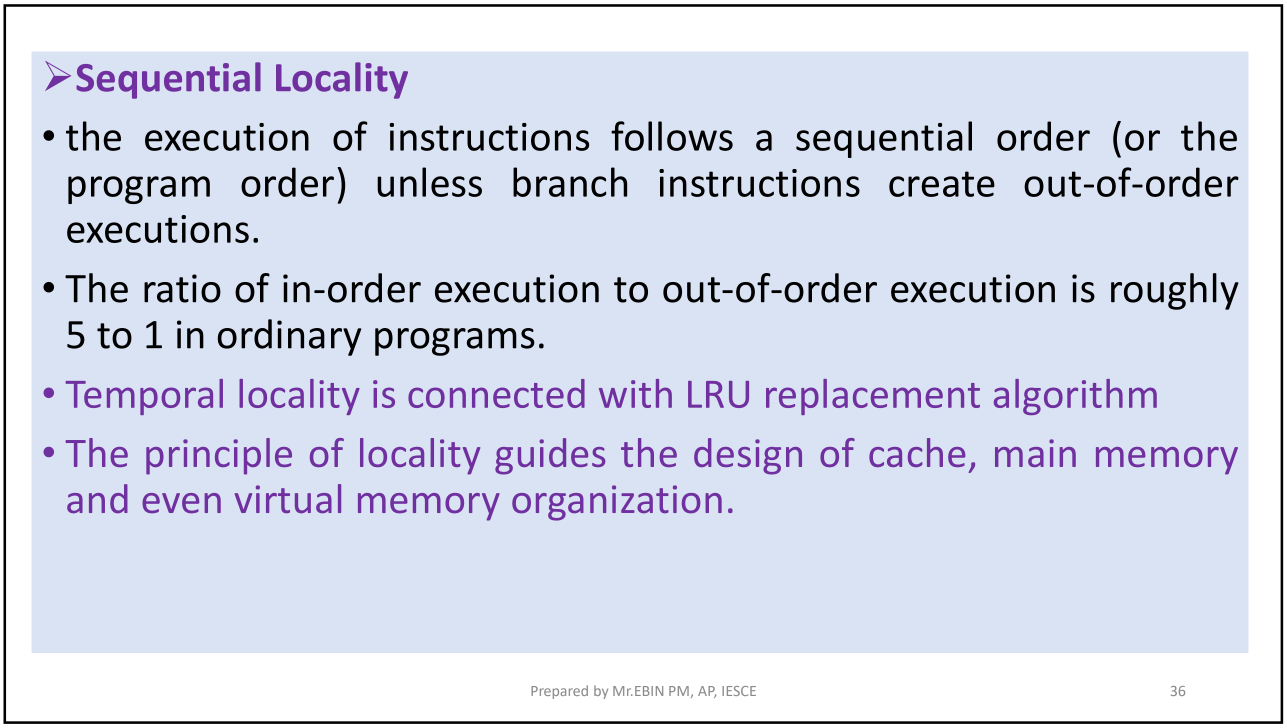
- Recently referenced items (instructions or data) are likely to be referenced again in the near future.
- This is often caused by special program constructs such as iterative loops, process stacks, temporary variables, or subroutines
- Temporal locality tends to cluster the access in the recently used areas.

➢Spatial Locality

- Tendency for a process to access items whose addresses are near

one another.

- For example, operations on tables or arrays involve accesses of a certain clustered area in the address space



➢Sequential Locality

- the execution of instructions follows a sequential order (or the program order) unless branch instructions create out-of-order executions.
- The ratio of in-order execution to out-of-order execution is roughly 5 to 1 in ordinary programs.
- Temporal locality is connected with LRU replacement algorithm
- The principle of locality guides the design of cache, main memory and even virtual memory organization.