BACS1024 INTRODUCTION TO COMPUTER SYSTEMS

Chapter 5: Computer Architecture and Memory

0. Overview

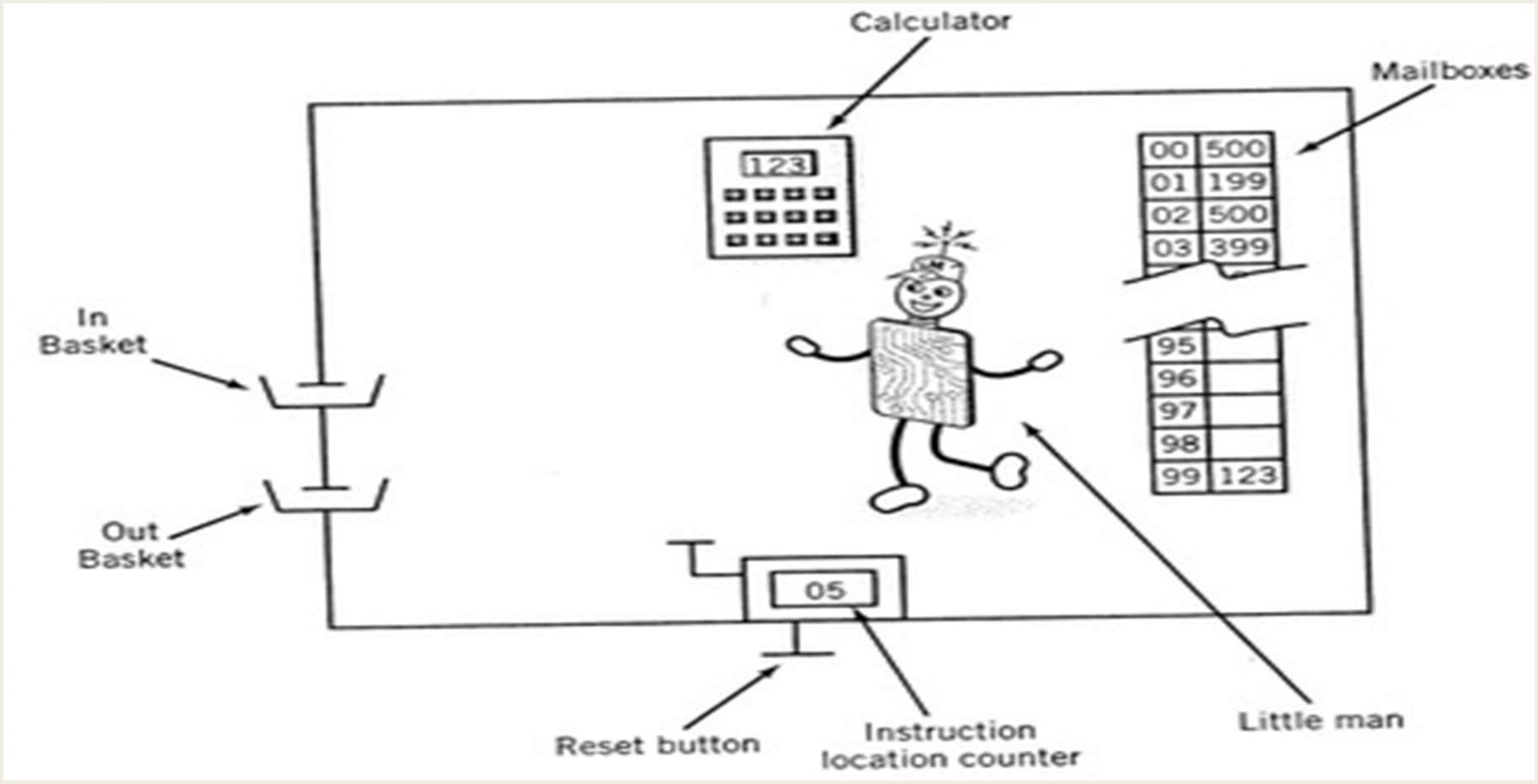
1. The Operation of a Computer

1. The Operation of a Computer

- **Little Man Computer (LMC) Model**
  - ❑ **LMC** model uses a simplified but typical set of instructions (program) in the computer operations.
  - ❑ The execution of these instructions illustrate the working model of the computer.
  - ❑ As programmer's role is to produce the exact sequence of operations to perform a particular tasks correctly, therefore it is essential to study the computer operations in details.

1. The Operation of a Computer

- **Little Man Computer (LMC) Model**



1. The Operation of a Computer

- **Little Man Computer (LMC) Model**
  - ❑ LMC models consists of several key components:

Components	Function	Description
1. Mailboxes	To hold a single slip of paper, upon which is written a three-digit decimal number	Each numbered with an address ranges from 00 to 99 (2 digits only)
2. Calculator	To enter and temporarily hold numbers for computation	The content restricted to 3 digits
3. Instruction location counter	To increment the count	With **RESET** switch, that located outside the mailroom, it facilitate communication between LMC & outside environment

1. The Operation of a Computer

- **Little Man Computer (LMC) Model**
  - ❑ In **LMC** environment, each instruction consists of a single digit, which is the first digit of a three-digit number, to tell the **Little Man** (computer) which operation to perform.

$\boxed{\mathbf{3\quad 25}}$

Instruction / Op code | Mailbox Address

1. The Operation of a Computer

- **Little Man Computer (LMC) Model**

$\boxed{3 \quad 25}$

Instruction / Op code | Mailbox Address

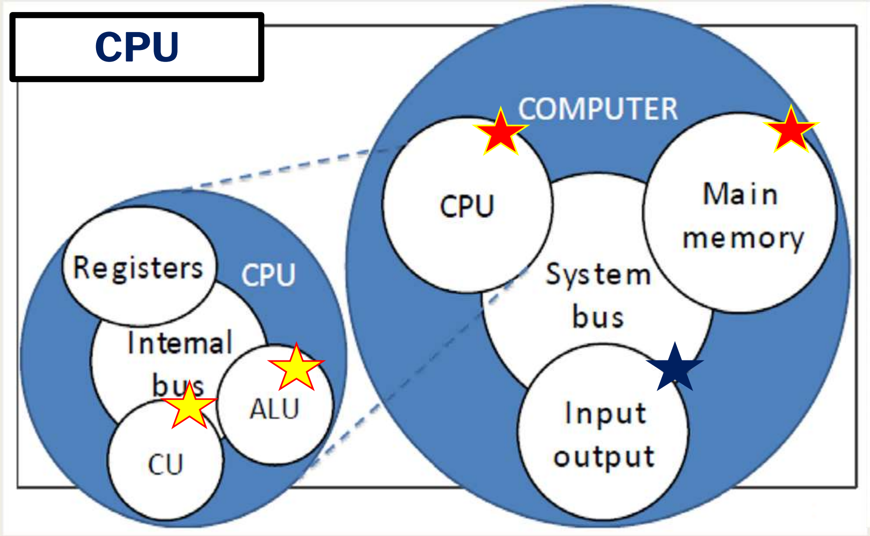
Op code	Instruction	Operation
5	**LOAD** instruction	The Little Man walks to mailbox address specified, reads the three-digit number & punches that number to calculator
3	**STORE** instruction	The Little Man walks to calculator, reads the three-digit number & write that number to mailbox as addressed
1	**ADD** instruction	The Little Man reads the three-digit numbers in mailbox & add it in calculator
2	**SUBTRACT** instruction	The Little Man reads the three-digit numbers in mailbox & subtract it in calculator

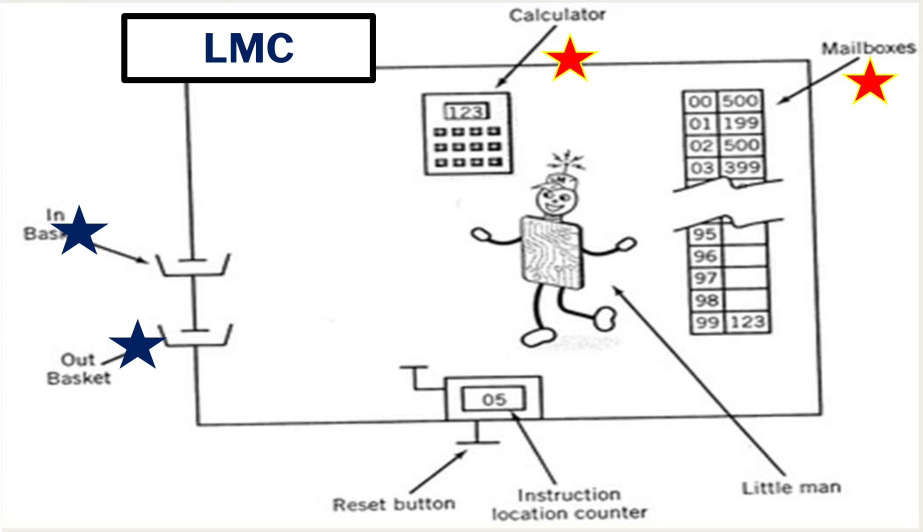
1. The Operation of a Computer

- **Little Man Computer (LMC) Model**
  - The **Little Man** performs an instruction as the machine / instruction cycle
    - ❑ During the *fetch* phase, the **Little Man** find out what instruction he is executed
      - ❖ He walk to location counter & reads its value
      - ❖ Then, go to mailbox as address specified & reads the three-digit number stored, which is instruction to be performed
    - ❑ During the *execute* phase, the **Little Man** perform the work specified in the instruction
      - ❖ Perform **LOAD**, **ADD**, **SUBTRACT** or **STORE** instruction

2. CPU

2. CPU





Function	CPU	LMC Model
Decode	CU	Little Man
Execute	ALU	Calculator
Storage	Memory	Mailbox

2. CPU

- **The concept of Registers**
  - ❑ Register is a single, permanent storage location within CPU used to hold binary value for storage, for manipulation and/or for simple calculations
  - ❑ Registers are not addressed as memory location but instead are manipulated directly by the control unit during execution of instructions

Registers		At	Function
A	Accumulator	ALU	Holds the data that are used for arithmetic operations & result
PC	Program Counter	CU	Holds the address of the current instruction being executed
IR	Instruction Register		Holds the actual instruction being executed currently
MAR	Memory Address Register		Holds the address of a memory location
MDR	Memory Data Register		Holds a data that is being stored to / retrieved from the memory location currently addressed by the memory address register

2. CPU

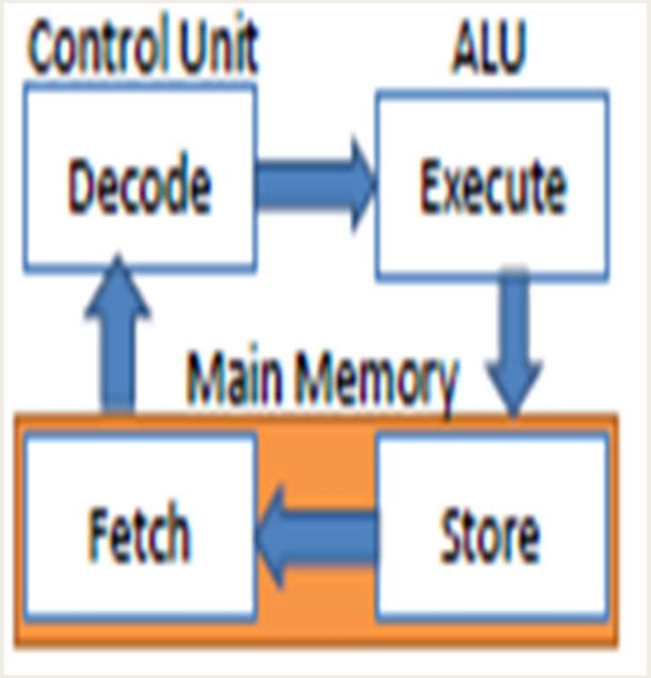
- **The Fetch-Execute Machine / Instruction Cycle**
  - ❑ There are 4 steps performed by CPU for each machine language instruction received.

Fetch phase

- ❖ **Fetch**: Retrieve an instruction from memory
- ❖ **Decode**: Translate the retrieved instruction into computer commands

Execute phase

- ❖ **Execute**: Execute the computer commands
- ❖ **Store**: Send & write the results back to memory



2. CPU

- **The Fetch-Execute Machine / Instruction Cycle**
  - ❑ During the fetch phase
    - ❖ Processor fetches instruction from memory location pointed by **PC**
    - ❖ The instruction loaded into **IR**
  - ❑ During the execute phase
    - ❖ The processor decodes the instruction & perform the required action.
    - ❖ Then, perform action execution of operation
    - ❖ Increment PC upon completion of 1 cycle

2. CPU

- **The Fetch-Execute Machine / Instruction Cycle**
  - ❑ E.g.: The fetch-execute steps for **LOAD** instruction

```
1. PC → MAR               ;MAR  = 30
2. MDR → IR               ;IR   = 540
3. IR[Address] → MAR      ;MAR  = 40
4. MDR → A                ;A    = 660
5. PC + 1 → PC            ;PC   = 31
```

Memory	
Address	Content
30	540
31	141
32	340
:	:
40	660
41	70

2. CPU

- **The Fetch-Execute Machine / Instruction Cycle**
  - ❑ E.g.: The fetch-execute steps for **ADD** instruction

```
1. PC → MAR                ; MAR  = 31
2. MDR → IR                ; IR   = 141
3. IR[Address] → MAR       ; MAR  = 41
4. A + MDR → A             ; A    = 660+70
5. PC + 1 → PC             ; PC   = 32
```

Memory	
Address	Content
30	540
31	141
32	340
:	:
40	660
41	70

2. CPU

- **The Fetch-Execute Machine / Instruction Cycle**
  - ❑ E.g.: The fetch-execute steps for **STORE** instruction

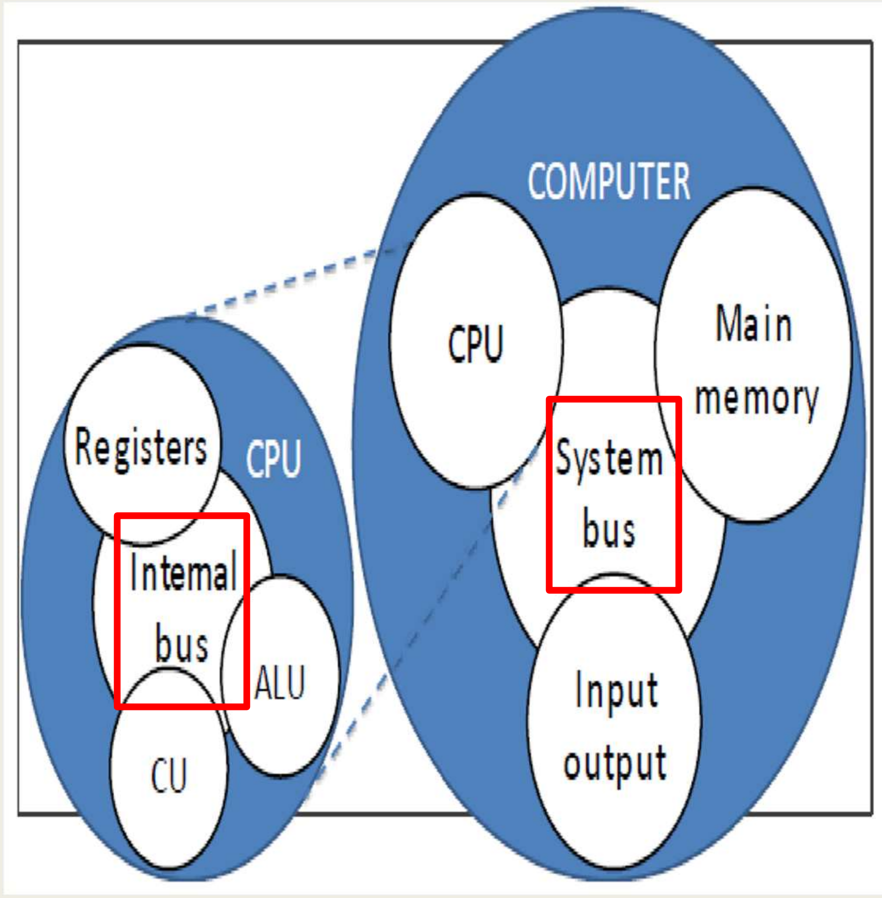
```
1. PC → MAR                ;MAR  = 32
2. MDR → IR                ;IR   = 340
3. IR[Address]→ MAR        ;MAR  = 40
4. A → MDR                 ;MDR  = 730
5. PC + 1 → PC             ;PC   = 33
```

Memory	
Address	Content
30	540
31	141
32	340
:	:
40	**730**
41	70

3. Computer Buses

3. Computer Buses

- **Computer Buses**
  - ❑ Bus is defined as a communication pathway that connecting two or more devices
  - ❑ It is a shared transmission medium
  - ❑ A signal transmitted by one device is broadcast to all other devices attached to the bus
  - ❑ Signal will overlap and become garbled if two devices transmit at the same time
  - ❑ Therefore, only one device can successfully transmit data at a time

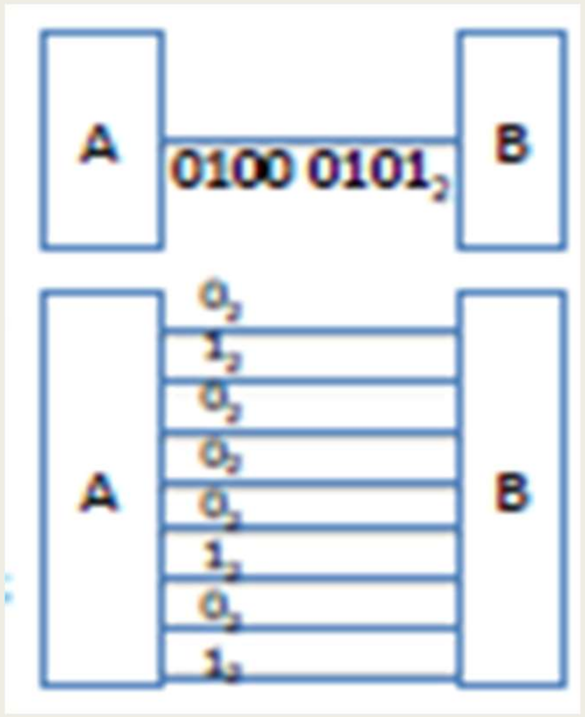


3. Computer Buses

- **Computer Buses – Data Transmission**
  - ❑ A bus consists of multiple communication pathways, known as lines
  - ❑ Each line transmits binary signal.
    - ❖ Serial bus

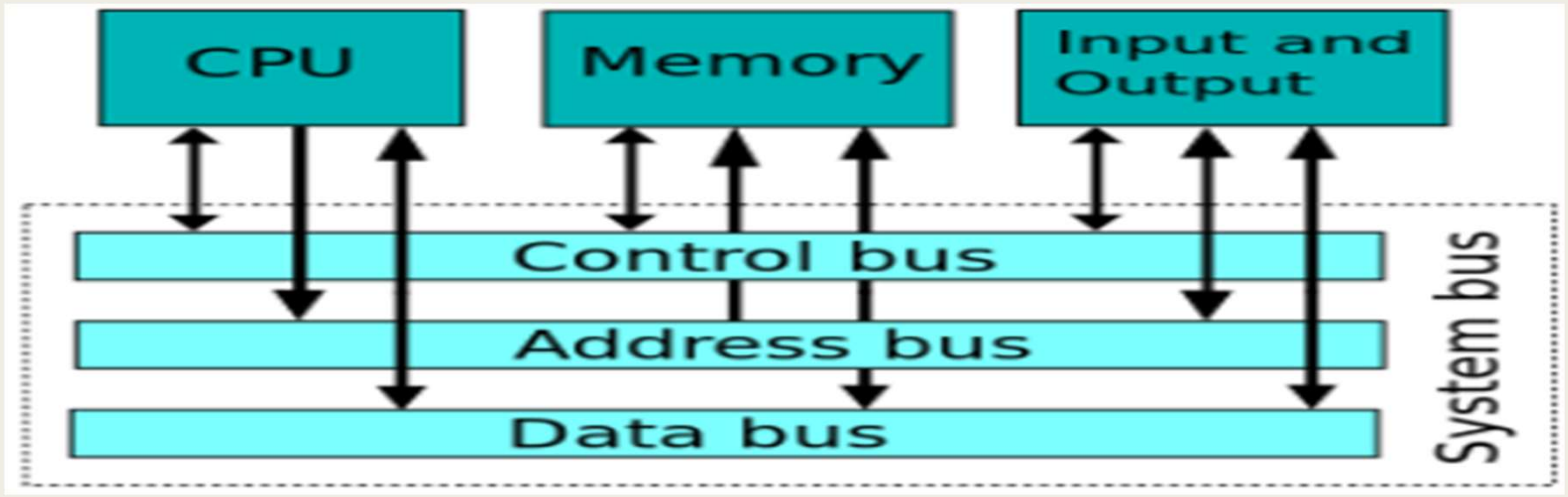
      Transmit a sequence of binary digits across a single line
    - ❖ Parallel bus

      Transmit binary digits simultaneously across a several lines



3. Computer Buses

- **Computer Buses – Types of Buses**
  - ❑ Three types of buses



❖ Data bus

❖ Address bus

❖ Control bus

3. Computer Buses

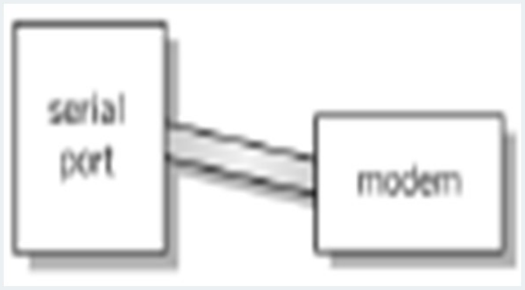
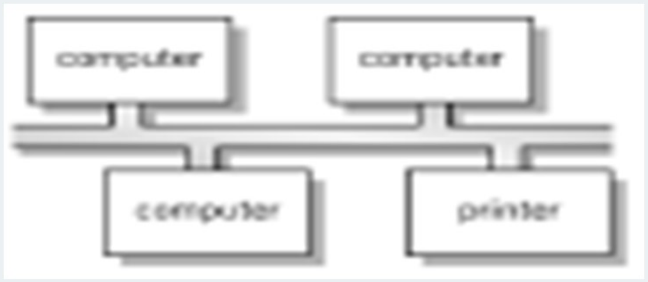
- **Computer Buses – Types of Buses**
  - ❑ **Data bus**
    - ❖ Carries “data” among system models
    - ❖ Consists of 32 / 64 / 128 / more separate line.
    - ❖ No. of line = with of data bus
    - ❖ Each line carry one bit at a time
  - ❑ **Address bus**
    - ❖ Identifies the sources / destination of the data on data bus
    - ❖ Bus width determines the maximum memory capacity of the system
    - ❖ E.g.: ***n*** bits addressing giving 2^n memory locations

3. Computer Buses

- **Computer Buses – Types of Buses**
  - **Control bus**
    - Provides control for the proper synchronization & operation of the bus and the module
    - Transmit both command & timing information among system modules
      - ✓ Command specifies operation to be performed
      - ✓ Timing signals indicate the validity of data & address information

3. Computer Buses

- **Computer Buses – Interconnection Method**

	Point-to-Point Bus	Multipoint Bus
Characteristic	A bus carry signals from a specific source to a specific location	A bus that connect several points together. The signals produced by a source on the bus are ‘broadcast’ to every other point
Example	Computer’s serial port to a printer	Ethernet
Diagram		

3. Computer Buses

- **Computer Buses – The Chipset**
  - ❑ A **chipset** is a component which routes data between the computer buses
  - ❑ There are two major types of chipset used in a typical computer:
    - ❖ North Bridge (memory controller

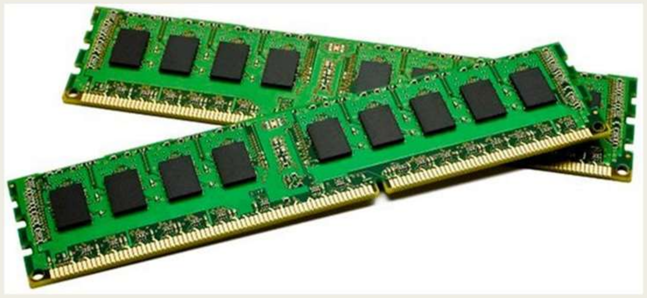
      = control the transfers between the processor & RAM
    - ❖ South Bridge (I/O controller / expansion controller)

      = handles communications between peripheral devices

4. Computer Memory

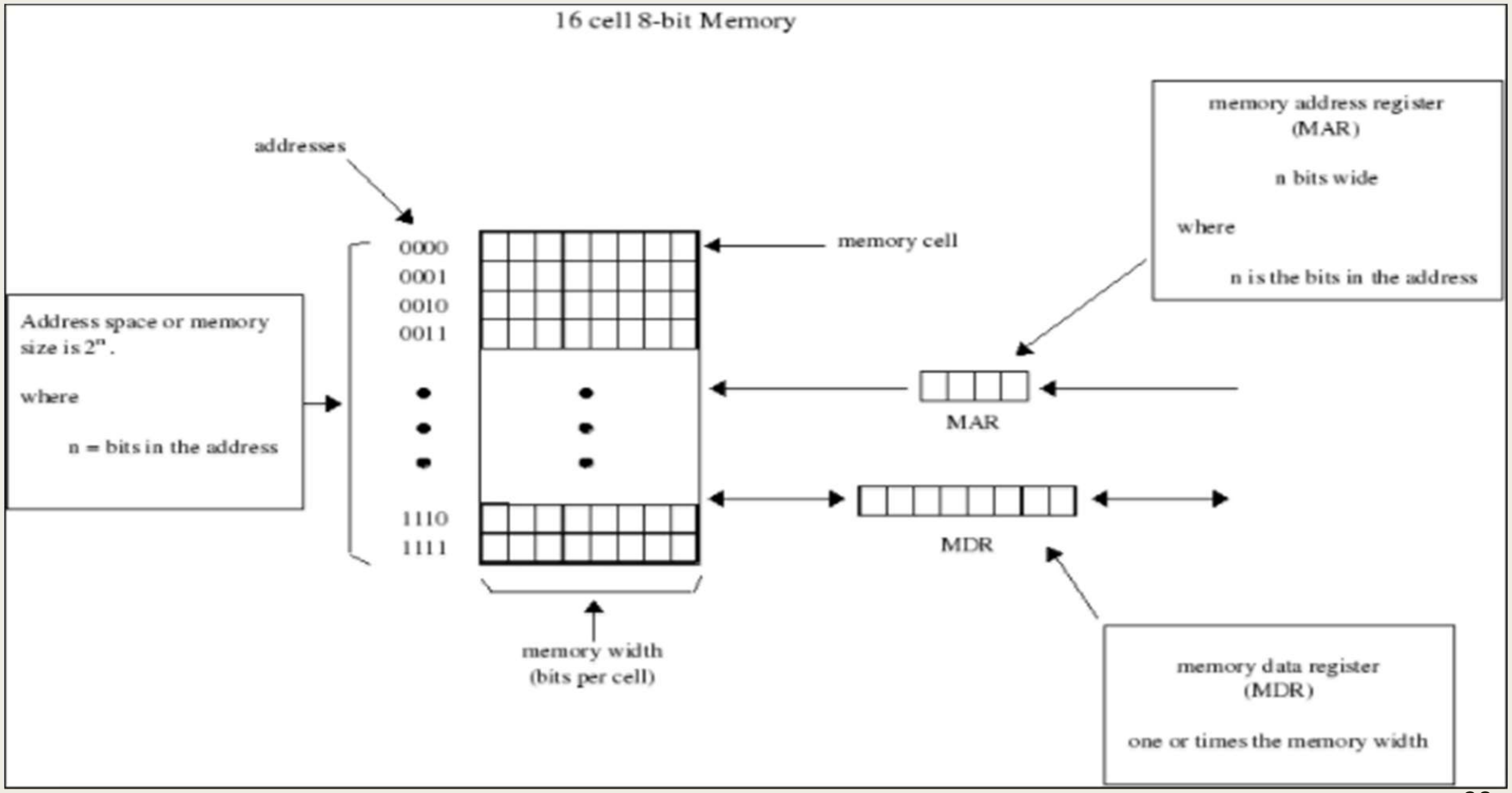
4. Computer Memory

- **Computer Memory**
  - ❑ A.k.a. working storage
  - ❑ Holds programs and data for accessed by CPU
  - ❑ Consists of large number of storage cells, sequentially numbered & individually addressed
  - ❑ The basic size of a cell is 8 buts
  - ❑ Neighboring cells can be grouped to store larger data type



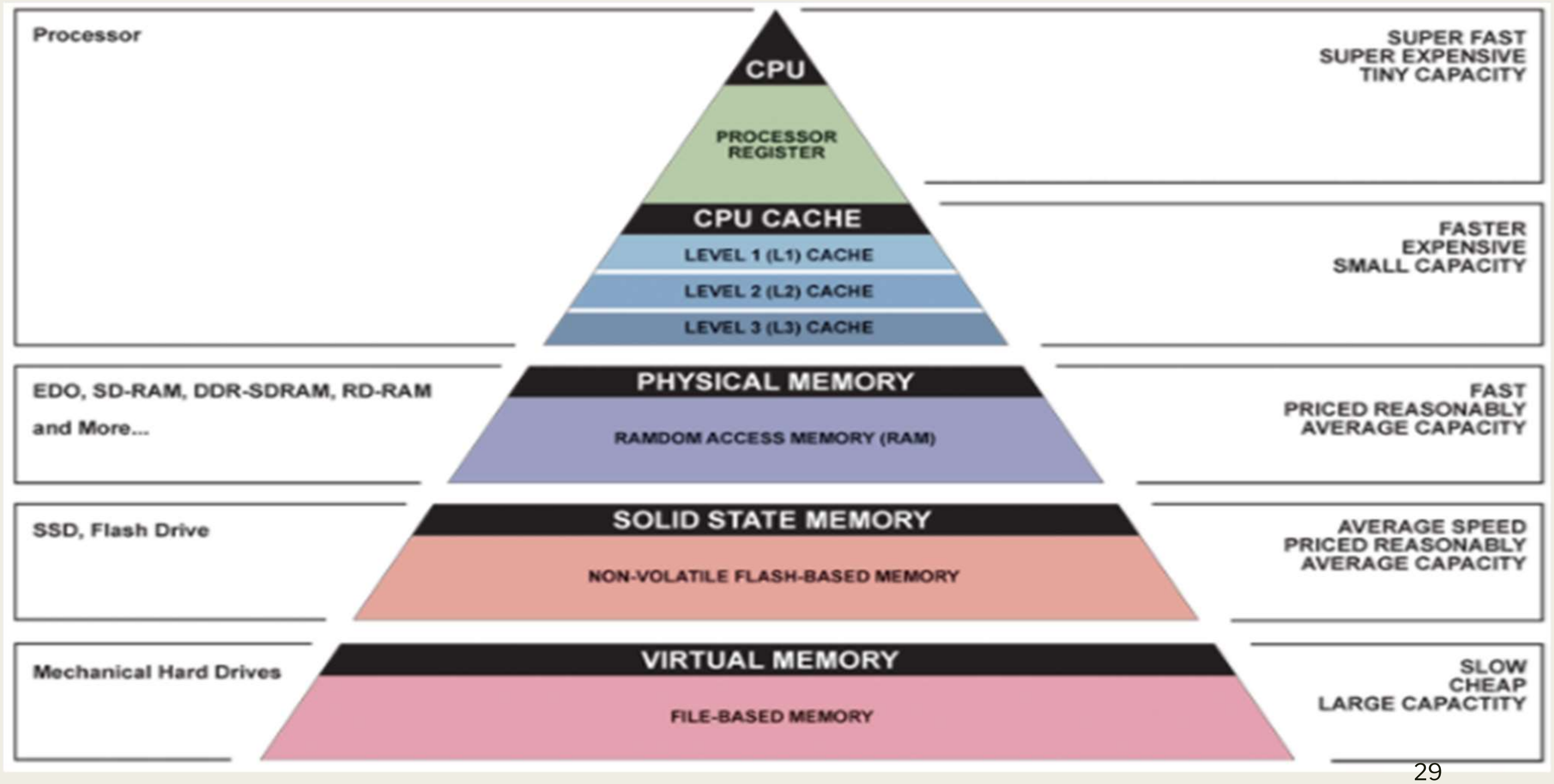
4. Computer Memory

- **Computer Memory**


16 cell 8-bit Memory
addresses
0000
0001
0010
0011
1110
1111
memory cell
memory address register (MAR)
n bits wide
where
n is the bits in the address
Address space or memory size is 2^n.
where
n = bits in the address
MAR
MDR
memory width (bits per cell)
memory data register (MDR)
one or times the memory width

4. Computer Memory

- Computer Memory – Memory Hierarchy



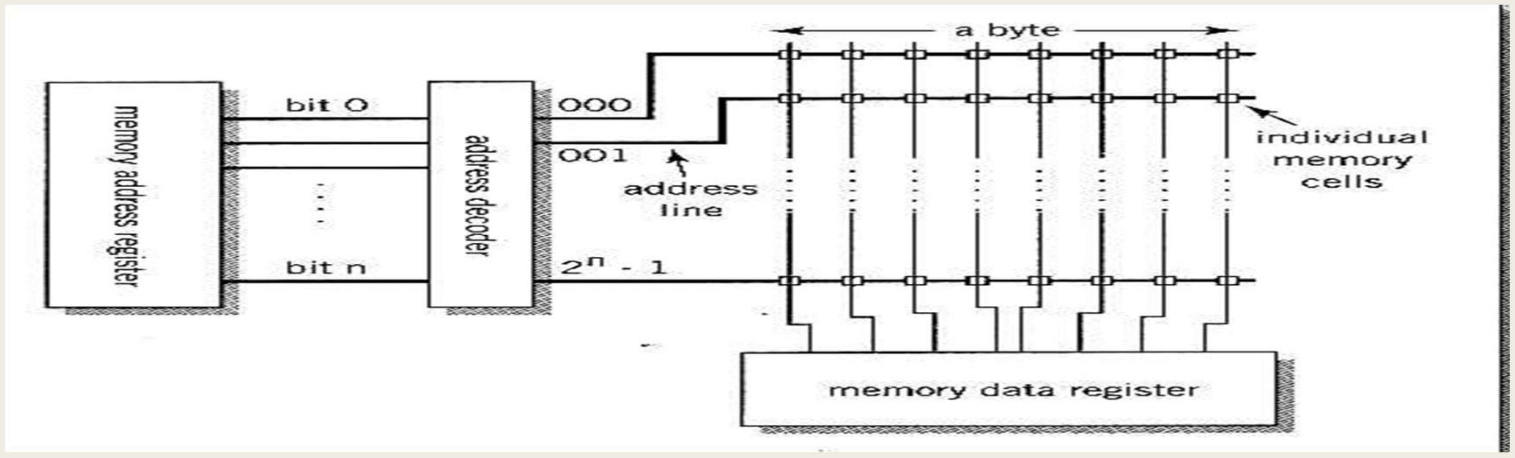
4. Computer Memory

- **Computer Memory – Types of Memory**

Volatile Memory	Non-volatile Memory
Hold data temporarily while the CPU is processing them	Hold data permanently
Requires constant power to maintain the stored information	Does not requires constant power to maintain the stored information
Primary storage	Secondary storage
E.g.: RAM	e.g.: ROM

4. Computer Memory

- **Computer Memory – The Operation of Memory**
  - Instruction and data can be retrieved from memory via an interface between CPU & memory.
  - The two registers used are:
    - **MAR**: Holds the address in memory which is open for data access
    - **MDR** : Holds the content of memory that is currently addressed by MAR



4. Computer Memory

- **Computer Memory – Memory Enhancement**
  - ❑ There are 3 key approaches:
    - ❖ Wider path memory access
    - ❖ Memory interleaving
    - ❖ Cache memory

1. Wider path memory access

- ✓ Several bytes / words can be read/written between CPU & memory with each access
- ✓ It is widely used.
- ✓ However, it increases the complexity in locating memory access.

4. Computer Memory

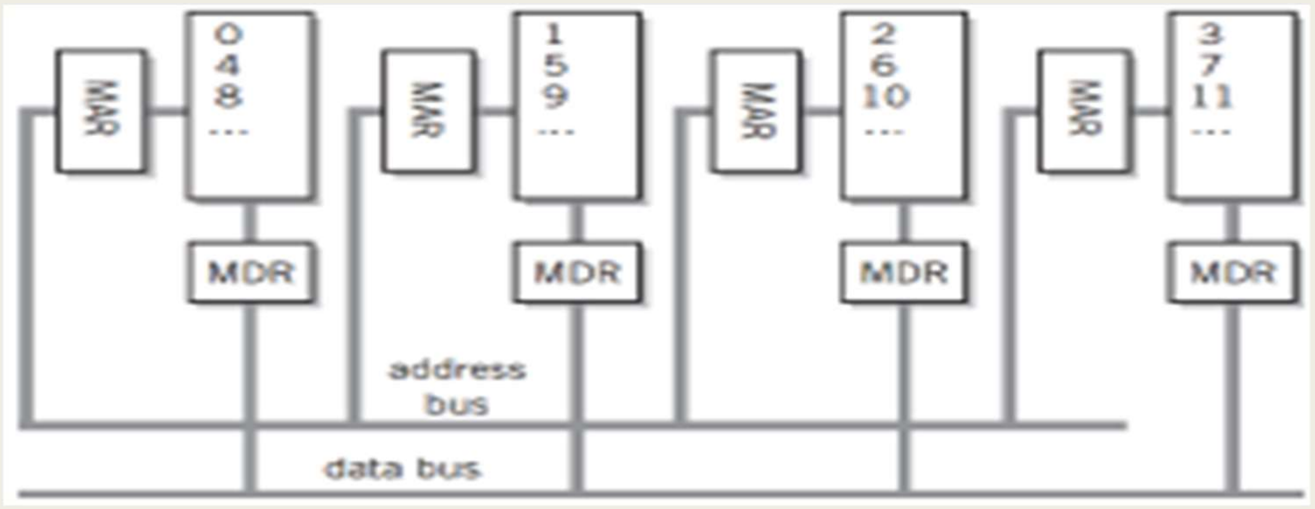
- **Computer Memory – Memory Enhancement**

2. Memory interleaving

✓ Divide memory into parts, for multiple access at the same time.

✓ Each part has its own MAR & MDR.

✓ Each part can be access independently



4. Computer Memory

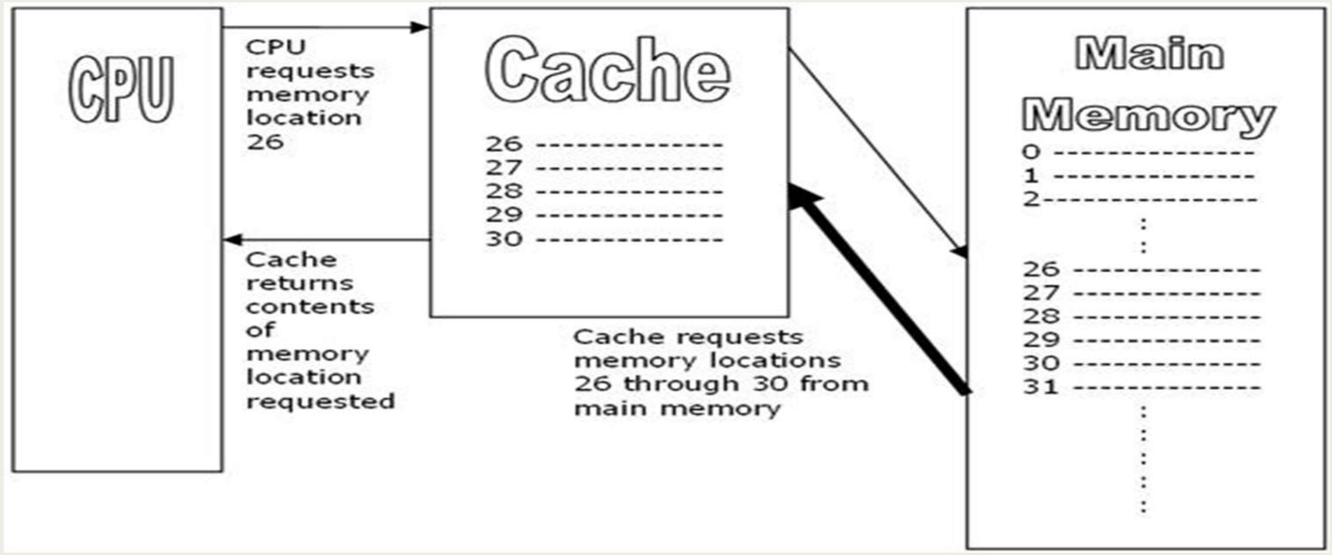
- **Computer Memory – Memory Enhancement**

  3. Cache Memory

  ✓ A small amount of high speed memory between CPU & main memory

  ✓ It is invisible to programmer & cannot be directly addressed

  ✓ Cache memory keeps a reproduction of data of memory


CPU
CPU requests memory location 26
Cache
26
27
28
29
30
Cache returns contents of memory location requested
Cache requests memory locations 26 through 30 from main memory
Main Memory
0
1
2
26
27
28
29
30
31

4. Computer Memory

- **Computer Memory – Memory Enhancement**

3. Cache Memory

✓ Cache memory is organized into blocks

✓ Each block holds a tag which acts as directory to identify the corresponding location of data in memory

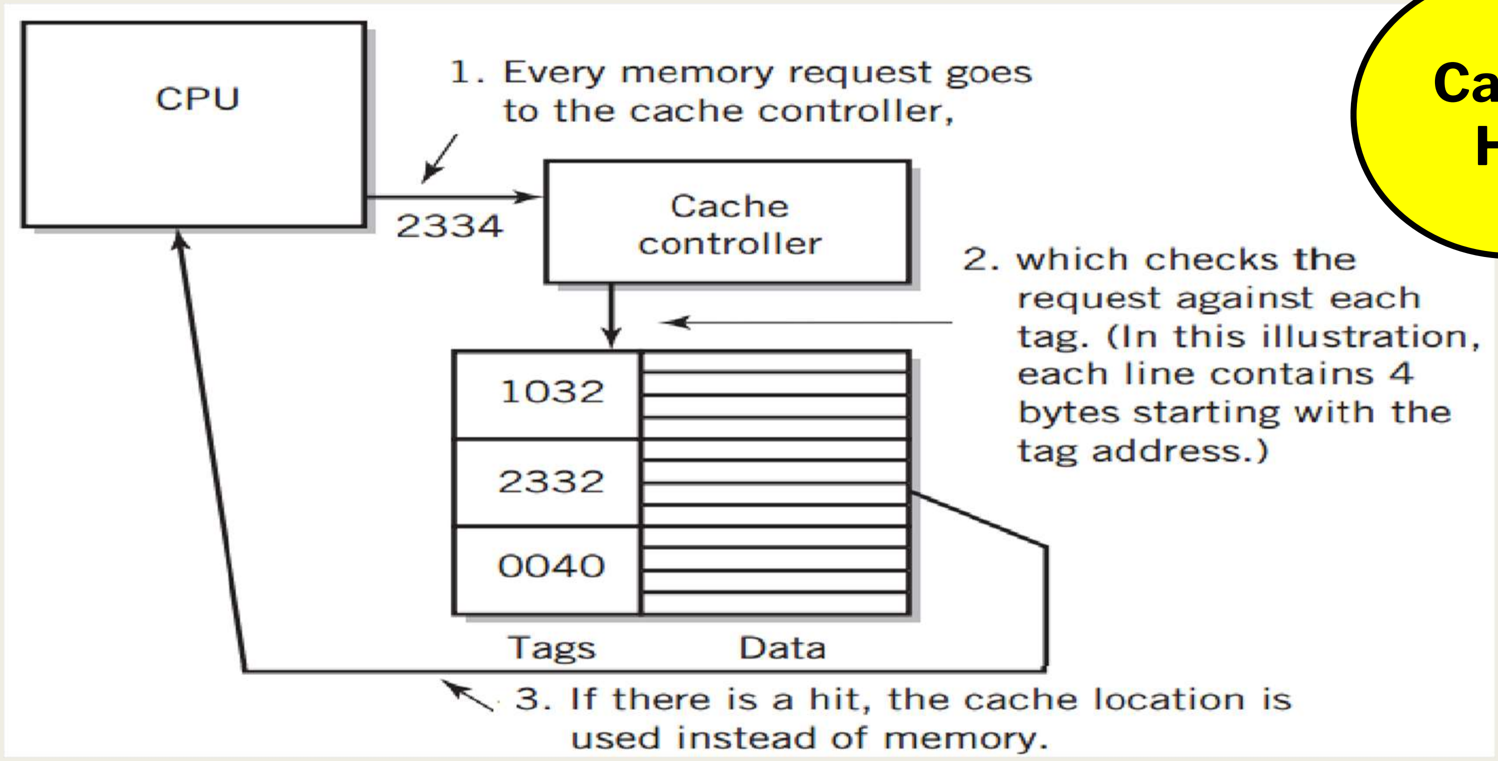
✓ the case operation includes:

- Cache controller: hardware that checks tags
- Cache line: Unit of transfer between storage & cache memory
- Hit ratio: Ration of hits out a total requests

4. Computer Memory

- **Computer Memory – Memory Enhancement**

3. Cache Memory – How cache memory work?

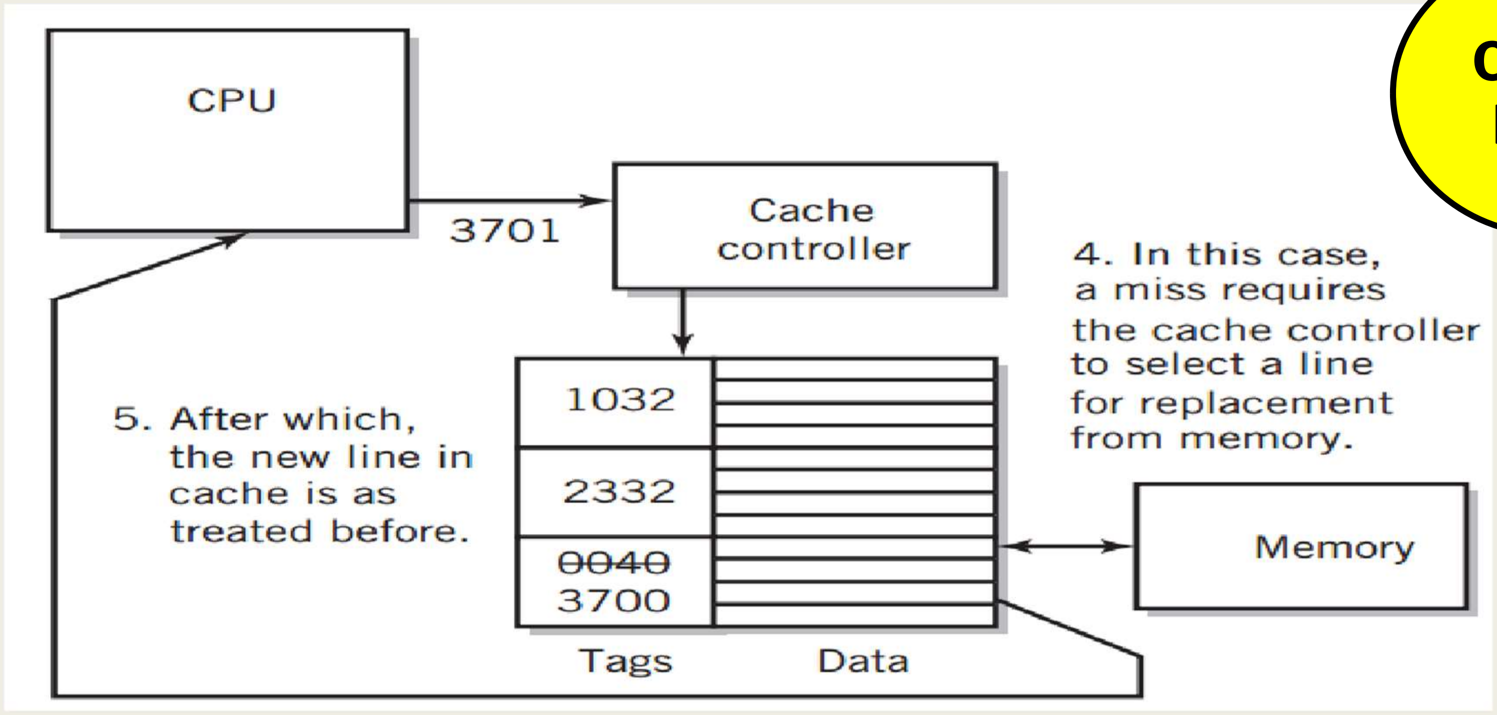

CPU
1. Every memory request goes to the cache controller,
2334
Cache controller
2. which checks the request against each tag. (In this illustration, each line contains 4 bytes starting with the tag address.)
1032
2332
0040
Tags
Data
3. If there is a hit, the cache location is used instead of memory.

Cache Hit

4. Computer Memory

- **Computer Memory – Memory Enhancement**

3. Cache Memory – How cache memory work?


CPU
3701
Cache controller
4. In this case, a miss requires the cache controller to select a line for replacement from memory.
5. After which, the new line in cache is as treated before.
1032
2332
~~0040~~
3700
Tags
Data
Memory

Cache Miss

5. Memory Management

5. Memory Management

- **There are three key models**
  1. Single-user continuous scheme
  2. Static Partition Scheme
  3. Dynamic Partition Scheme

1. Single-user Continuous Scheme

- ❑ Each program is loaded in its entirety into memory and is allocated as much contiguous memory space as needed.
- ❑ If program was too large - it couldn't be executed.
- ❑ Minimal amount of work done by Memory Manager.

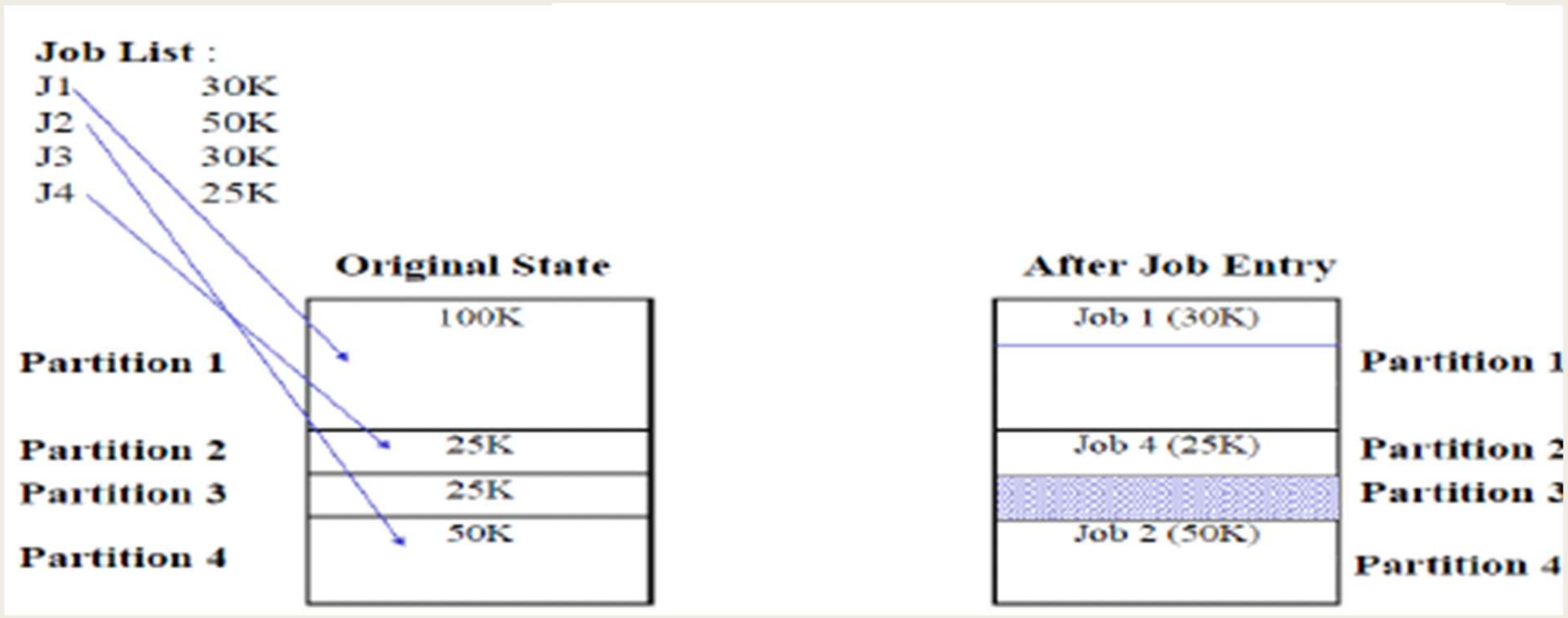
5. Memory Management

2. Static Partition Scheme

- ❑ Allows multiprogramming by using fixed partitions where one partition for each job
- ❑ The size of each partition remains static once the system is in operation. Each partition can only be reconfigured when the computer system is shut down, reconfigured and restarted.
- ❑ The partition sizes are critical. If the partition sizes are too small, larger jobs will be rejected. If partition sizes are too big, memory can be wasted if a job does not occupy the entire partition.
- ❑ Internal fragmentation is a problem. Internal fragmentation occurs when there are unused memory spaces within the partition itself.

5. Memory Management

2. Static Partition Scheme



Job 3 must wait even though **70K** of free space is available in **Partition 1** where **Job 1** only occupies **30K** of the **100K** available. The jobs are allocated space on the basis of “first available partition of required size.”

5. Memory Management

3. Dynamic Partition Scheme

- ❑ Available memory are kept in contiguous blocks and jobs are given only as much memory as they request when loaded.
- ❑ Improves memory use over fixed partitions.
- ❑ Performance deteriorates as new jobs enter the system
- ❑ Fragments of free memory are created between blocks of allocated memory (external fragmentation).
  - ❖ **First-fit**: Allocate the *first* partition that is big enough.
  - ❖ **Best-fit**: Allocate the *smallest* partition that is big enough

5. Memory Management

3. Dynamic Partition Scheme

First Fit	Best Fit	**Worst fit**
Faster to implement	**Slower** to implement because the entire free list table needs to be searched before allocation can be made.	
Algorithm is **simple**.	Algorithm is **complex** because it needs to find smallest block of memory into which the job can fit.	
Memory list organized according to **memory locations**, **low-order**	Memory list organized according to **memory size**, **smallest to largest**.	Memory list organized according to **memory size**, **largest to smallest**.
May not be making **efficient** use of memory space.	Produces the smallest leftover partition. Make **most efficient** use of memory.	Produces the largest leftover partition. Make **worst use** of memory

5. Memory Management

3. Dynamic Partition Scheme

- ❑ Given the info.

Job	Memory requested
J1	10K
J2	20K
J3	30K
J4	15K

- ❑ **First Fit**

Partition	Partition Size	Job	Job Size	Internal Fragmentation
P1	25K	J1	10K	25K – 10K = 15K
P2	20K	J2	20K	20K – 20K = 0K
P3	30K	J3	30K	30K – 30K = 0K
P4	35K	J4	15K	35K – 15K = 20K
P5	15K			
Total available:	125K	Total :	75K	35K

5. Memory Management

3. Dynamic Partition Scheme

- ❑ Given the info.

Job	Memory requested
J1	10K
J2	20K
J3	30K
J4	15K

- ❑ **Best Fit**

Partition	Partition Size	Job	Job Size	Internal Fragmentation
P1	25K	J4	15K	25K – 15K = 10K
P2	20K	J2	20K	20K – 20K = 0K
P3	30K	J3	30K	30K – 30K = 0K
P4	35K			
P5	15K	J1	10K	15K – 10K = 5K
Total available:	125K	Total :		15K

5. Memory Management

3. Dynamic Partition Scheme

- ❑ Given the info.

Job	Memory requested
J1	10K
J2	20K
J3	30K
J4	15K

- ❑ **Worst Fit**

Partition	Partition Size	Job	Job Size	Internal Fragmentation
P1	25K	J4	15K	25K – 15K = 10K
P2	20K			
P3	30K	J2	20K	30K – 20K = 10K
P4	35K	J1	10K	35K – 10K = 25K
P5	15K			
Total available:	125K	Total :	45K	45KK

Note: J3 has to wait

Chapter Review

Chapter Review

1. The Operation of a Computer

- ❑ LMC Model

2. CPU

- ❑ CPU vs LMC
- ❑ Registers
- ❑ Instruction Cycle

3. Computer Buses

- ❑ Data transmission
- ❑ Types of buses
- ❑ Direction of transfer
- ❑ interconnection

4. Cache Memory

- ❑ Memory hierarchy
- ❑ Types of memory
- ❑ Operations of memory
- ❑ Memory enhancement
  - ❖ Wider path memory access
  - ❖ Memory interleaving
  - ❖ Cache memory

5. Memory Management

- ❑ Single user contiguous scheme
- ❑ Static partition scheme
- ❑ Dynamic partition scheme
  - ❖ First fit algorithm
  - ❖ Best fit algorithm
  - ❖ Worst fit algorithm