Computer architecture and Organization

Memory Organization

Memory Organization

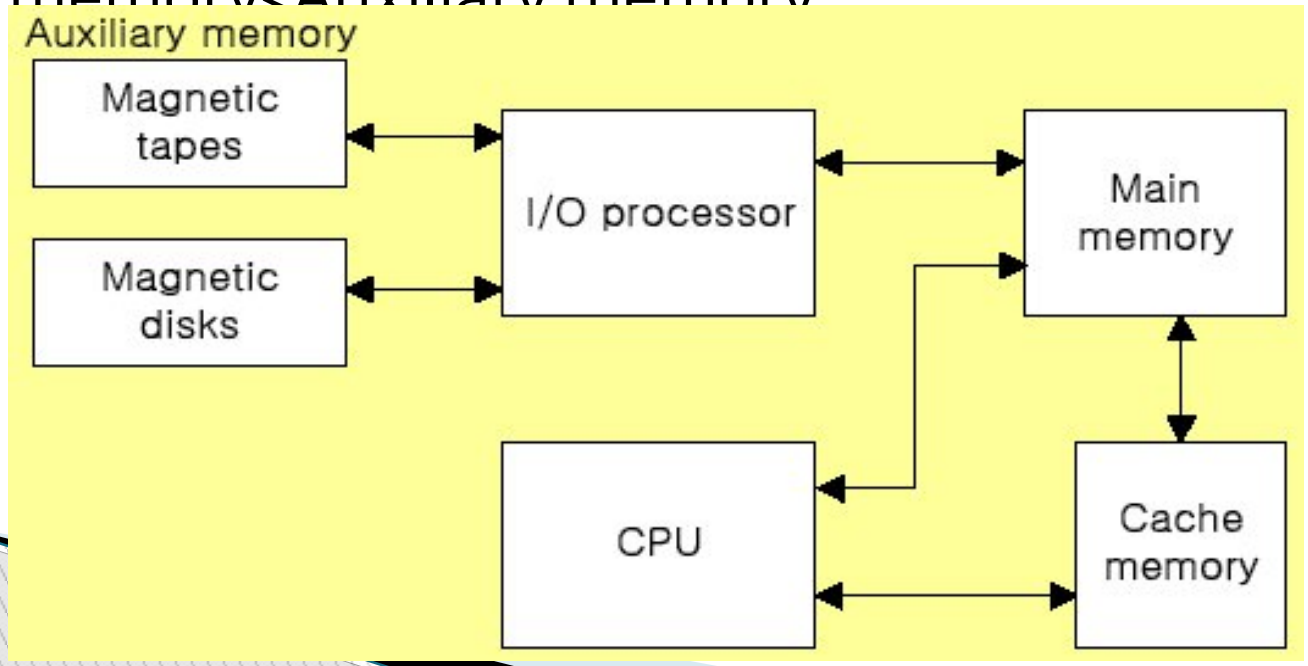
Memory Hierarchy

- Arrangement of different types of memories based on accessibility in computer system
- Main Memory: memory unit that communicates directly with the CPU (RAM)
- Auxiliary Memory: device that provide backup storage (Disk Drives)
- Cache Memory: special very-high-speed memory to increase the processing speed

Memory Organization

Memory Hierarchy

- Order according to speed: Cache>Main Memory>Auxiliary memory
- Order according to size: Cache<Main memory<Auxiliary memory



Memory Organization

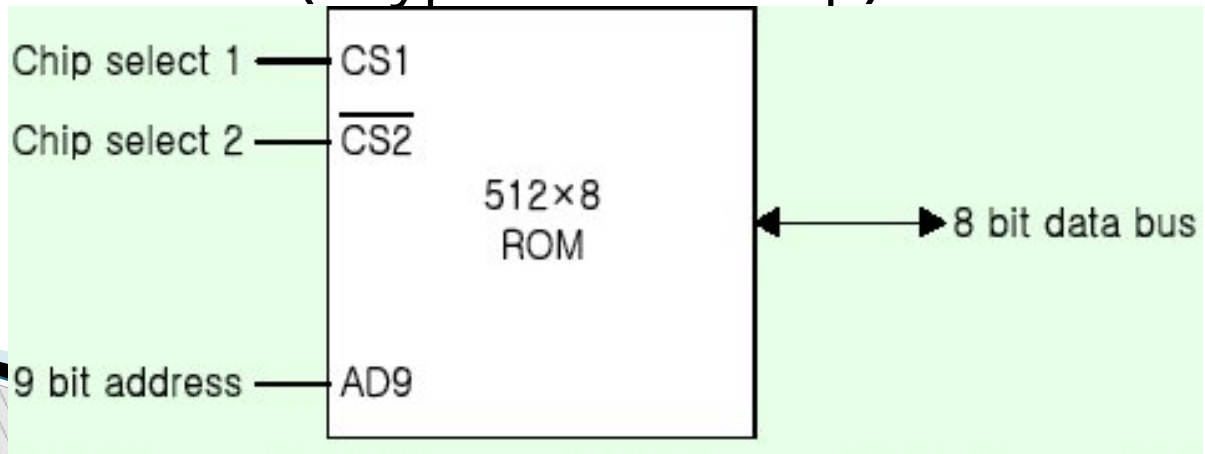
Main Memory

- Main memory is the only one directly accessible to the CPU.
- The CPU continuously reads instructions stored there and executes them as required.
- Any data actively operated on is also stored there in uniform manner.
- RAM (Random access memory) and ROM (Read only memory) are treated as main memory.

Memory Organization

ROM

- Stores a program that helps start up the computer
- ROM is non-volatile (holds programs and data even after power off)
- ROM is also direct access but only read
- The contents of ROM are fixed at the time of manufacture (a typical ROM chip)



Memory Organization

RAM

- Its purpose is to temporarily hold programs and data for processing.
- Described as being volatile (all data are lost when power off)
- It is direct access as it can be both written to or read from in any order

Memory Organization

Types of RAM

- Dynamic RAM (DRAM)
- The RAM which requires the stored information to be periodically re-written, or refreshed, otherwise it would vanish.
- Static RAM (SRAM)
- The RAM which never needs to be refreshed as long as power is applied. (It loses its content if power is removed)

Differences between DRAM and SRAM

DRAM	SRAM
1. Store data on capacitive circuit	1.Store data on Flip/Flop
2. Need to refresh after some time	2.No need to refresh
3. Slower than SRAM	3.Faster than DRAM
4. Less expensive	4.More expensive than DRAM

Memory Organization

Memory address map

- Memory Address Map is a representation of assigned address space for each chip
- Assume that a computer system needs **512** bytes of RAM and **512** bytes of ROM
- The RAM have **128** byte and need seven address lines, where the ROM have **512** bytes and need **9** address lines

Memory Organization

Memory address map

Component	Hexadecimal Address	10	9	8	7	6	5	4	3	2	1
RAM1	0000-007F	0	0	0	*	*	*	*	*	*	*
RAM2	0080-00FF	0	0	1	*	*	*	*	*	*	*
RAM3	0100-017F	0	1	0	*	*	*	*	*	*	*
RAM4	0180-01FF	0	1	1	*	*	*	*	*	*	*
ROM	0200-03FF	1	*	*	*	*	*	*	*	*	*

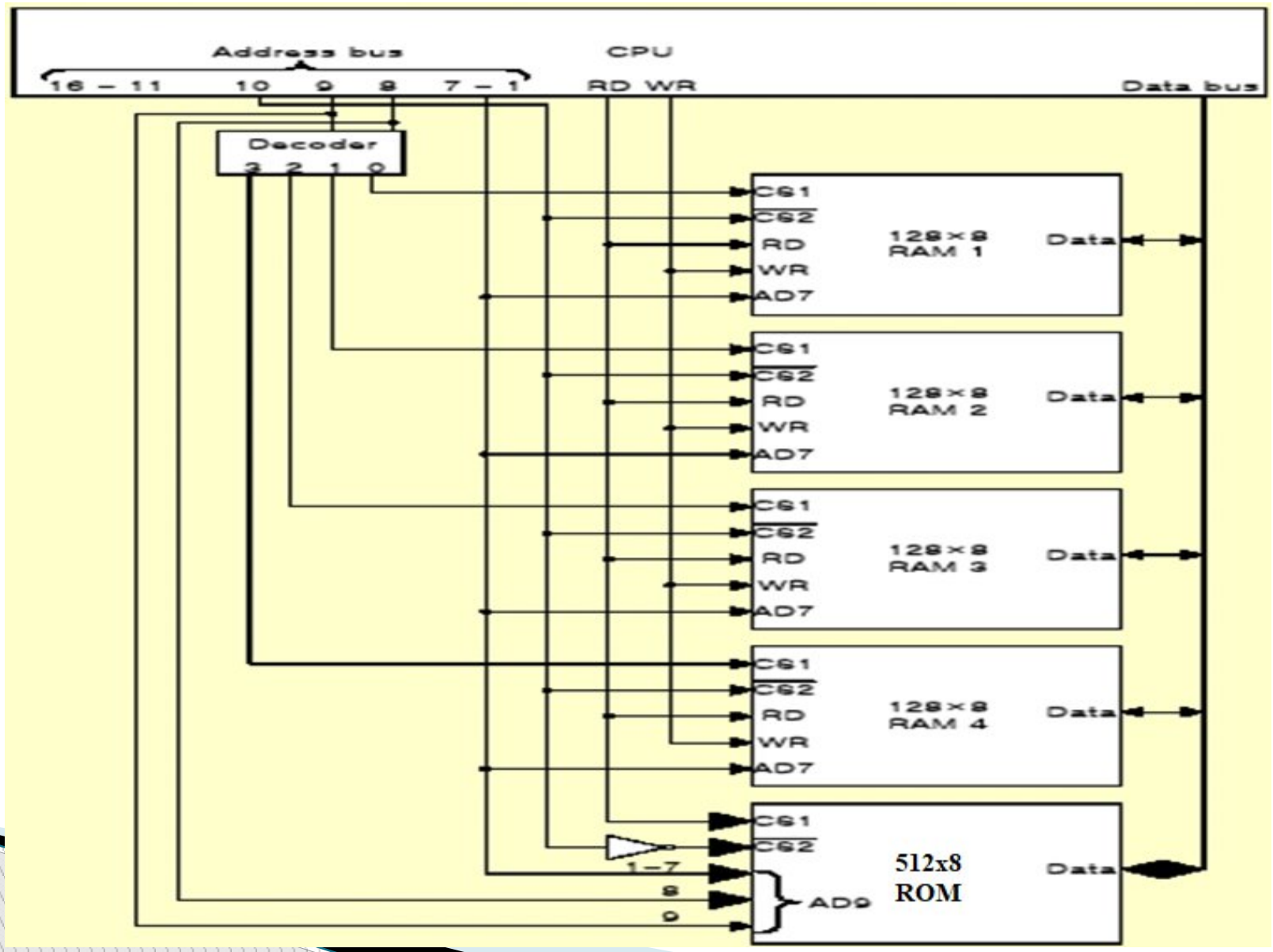
Memory Organization

Memory Connection with CPU

- The hexadecimal address assigns a range of hexadecimal equivalent address for each chip
- Line **8** and **9** represent four distinct binary combination to specify which RAM we chose
- When line **10** is **0,** CPU selects a RAM**.** And when it's **1,** it selects the ROM

Memory Organization

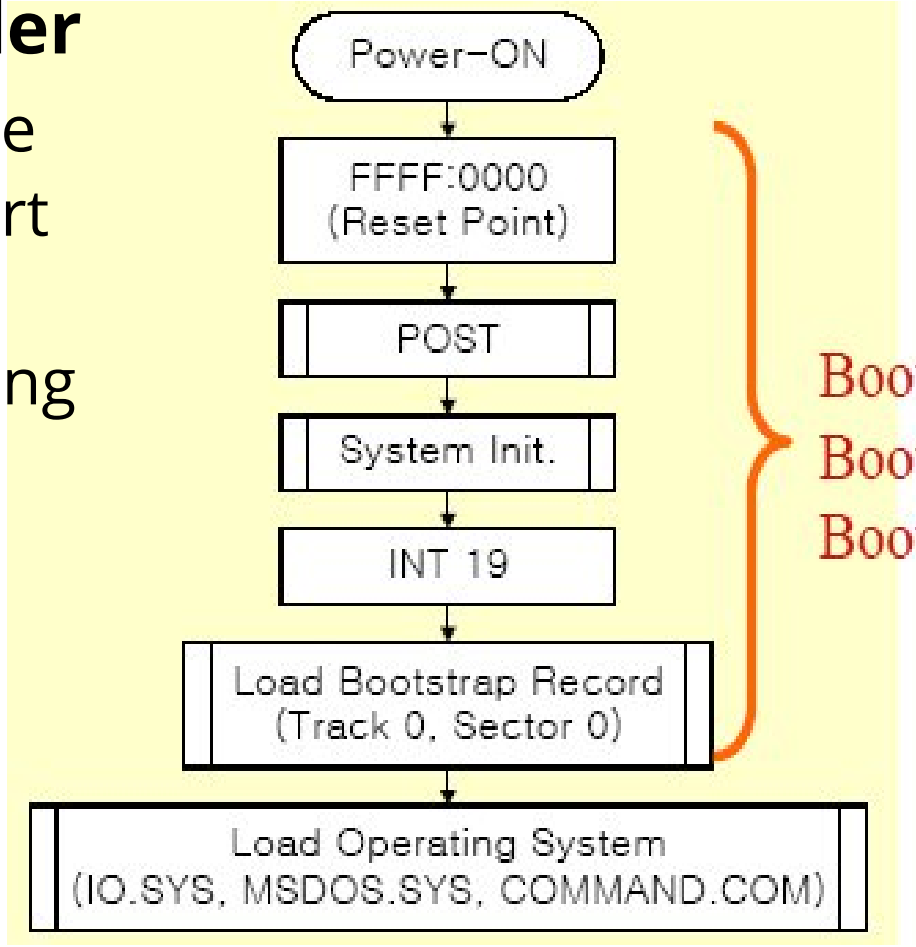
Memory Connection with CPU



Memory Organization

Bootstrap Loader

- A program whose function is to start the computer software operating when power is turned on



Memory Organization

Cache memory

- Small amount of faster memory (SRAM) to speed up computer
- Level 1 cache is built within the CPU (internal)
- Level 2 cache may be on chip or nearby (external)
- Faster for CPU to access than main memory

Memory Organization

Cache performance

- Defined as *hit ratio*
- When the CPU refers to memory and finds the word in cache, it is called *hit*;
- If the word is not found in cache it counts *miss*.
- The hit ratio is defined as:
  hit ratio=*hit*/(*hit*+*miss*).

Memory Organization

Factors of Cache performance

- Cache size
- Mapping method: associative, direct, set-associative
- Replace algorithm: LRU, LFU, FIFO
- Write policy: write-through, write-back

Memory Organization

Cache Replacement Algorithm

- LRU (Least Recently Used)
- LFU (Least Frequently Used)
- FIFO (First-In First-Out)

Writing to Cache

- Write-through: Cache write and main memory
- Write-back: Cache write and set flag to indicate the last writing

First-In First-Out

Consider the page reference string of size 12: 1, 2, 3, 4, 5, 1, 3,1, 6, 3, 2, 3 with frame size 4(i.e. maximum 4 pages in a frame).

1	2	3	4	5	1	3	1	6	3	2	3

1	1	1	1	5	5	5	5	5	5	2	2
	2	2	2	2	1	1	1	1	1	1	1
		3	3	3	3	3	3	6	6	6	6
			4	4	4	4	4	4	3	3	3
M	M	M	M	M	M	H	H	M	M	M	H

First-In First-Out

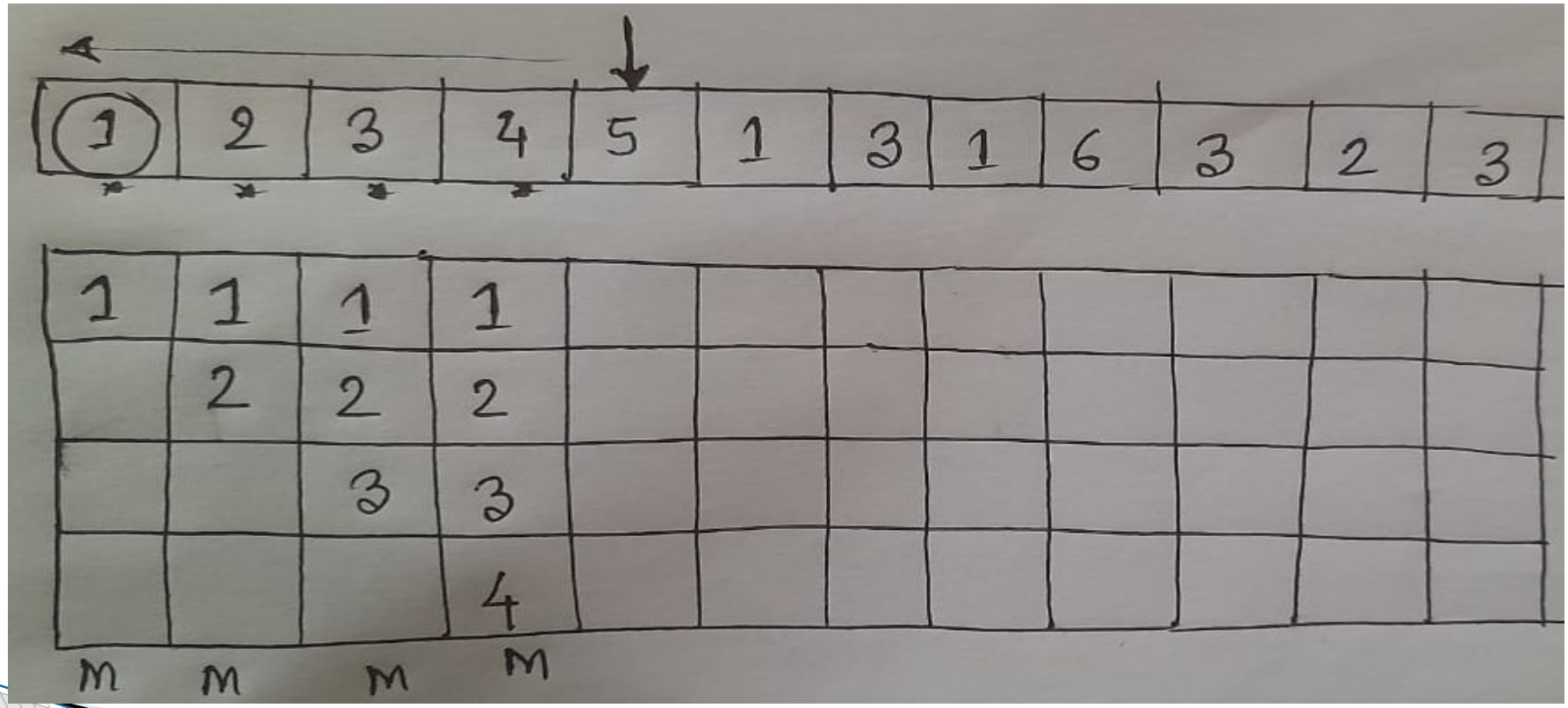
Total Page Fault = 9

Page Fault ratio = total miss/total possible cases

Page Fault ratio = 9/12

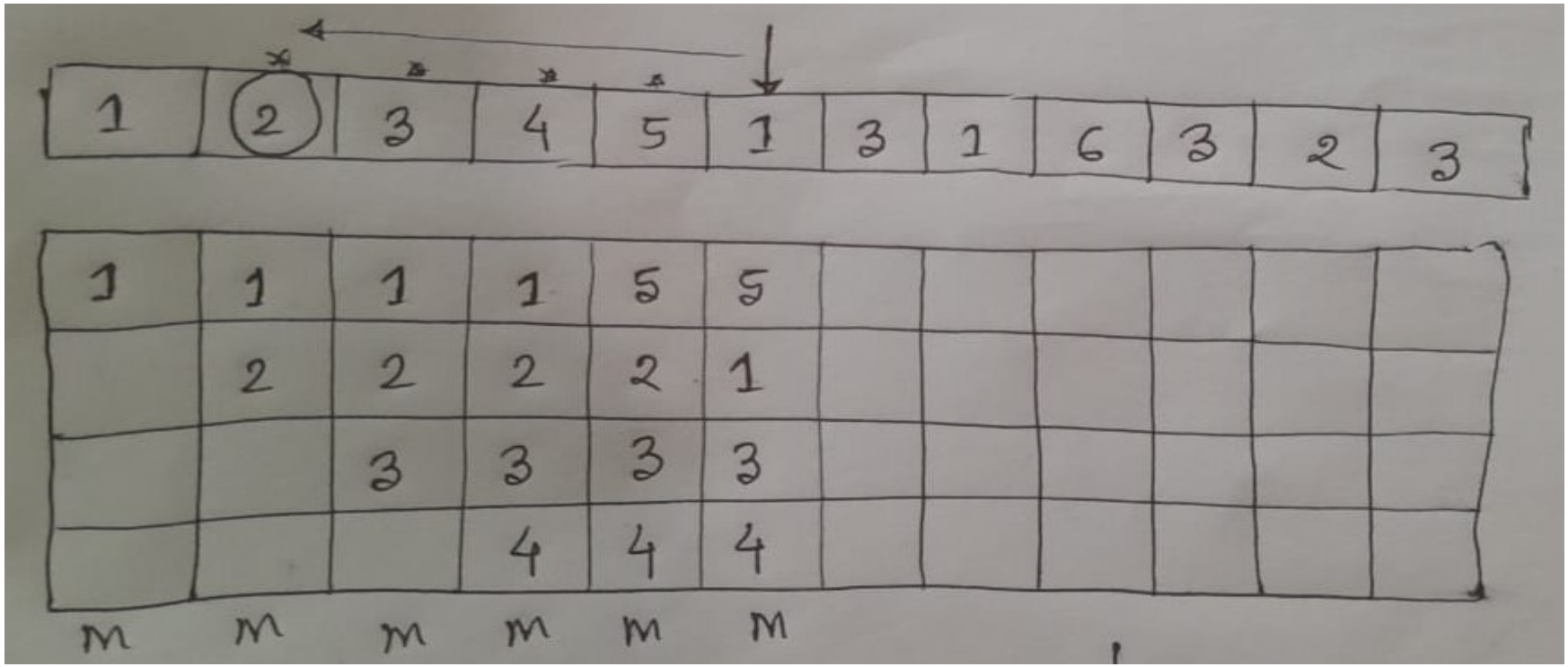
Least Recently Used

Consider the page reference string of size 12: 1, 2, 3, 4, 5, 1, 3,1, 6, 3, 2, 3 with frame size 4(i.e. maximum 4 pages in a frame).



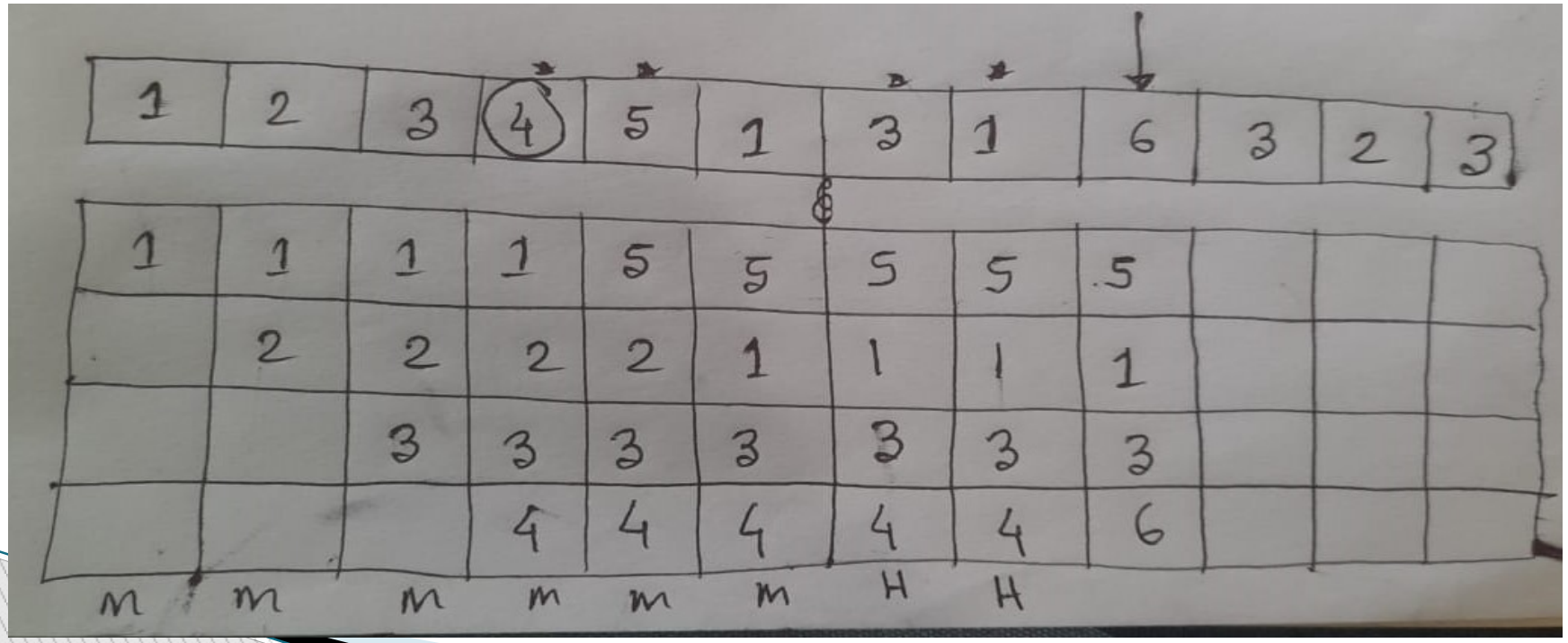
Least Recently Used

Consider the page reference string of size 12: 1, 2, 3, 4, 5, 1, 3,1, 6, 3, 2, 3 with frame size 4(i.e. maximum 4 pages in a frame).



Least Recently Used

Consider the page reference string of size 12: 1, 2, 3, 4, 5, 1, 3,1, 6, 3, 2, 3 with frame size 4(i.e. maximum 4 pages in a frame).



Least Recently Used

1	2	3	4	5	1	3	1	6	3	2	3

1	1	1	1	5	5	5	5	5	5	2	2
	2	2	2	2	1	1	1	1	1	1	1
		3	3	3	3	3	3	3	3	3	3
			4	4	4	4	4	6	6	6	6
M	M	M	M	M	M	H	H	M	H	M	H

Least Recently Used

Total Page Fault = 8

Page Fault ratio = total miss/total possible cases

Page Fault ratio = 8/12

Least Frequently Used

Consider the page reference string of size **10**: **3,5,6,6,1,1,4,2,3,4** with frame size **3**(i.e. maximum **3** pages in a frame).

3	**5**	**6**	**6**	**1**	**1**	**4**	**2**	**3**	**4**
3	3	3	3	1					
	5	5	5	5					
		6	6	6					
M	M	M	H	M					

	3	**5**	**6**	**1**	**4**	**2**
Freq.	1	1	2	0 /1	0	0

Least Frequently Used

Consider the page reference string of size **10**: **3,5,6,6,1,1,4,2,3,4** with frame size **3**(i.e. maximum **3** pages in a frame).

3	**5**	**6**	**6**	**1**	**1**	**4**	**2**	**3**	**4**
3	3	3	3	1	1	1			
	5	5	5	5	5	4			
		6	6	6	6	6			
M	M	M	H	M	H	M			

	3	**5**	**6**	**1**	**4**	**2**
Freq.	**0**	1	2	1/2	0	0

Least Frequently Used

Consider the page reference string of size 10: 3,5,6,6,1,1,4,2,3,4 with frame size 3(i.e. maximum 3 pages in a frame).

3	5	6	6	1	1	4	2	3	4
3	3	3	3	1	1	1	1		
	5	5	5	5	5	4	2		
		6	6	6	6	6	6		
M	M	M	H	M	H	M	M		

	3	5	6	1	4	2
Freq.	0	0	2	2	0/1->0	0->1

Least Frequently Used

Consider the page reference string of size **10**: **3,5,6,6,1,1,4,2,3,4** with frame size **3**(i.e. maximum **3** pages in a frame).

3	**5**	**6**	**6**	**1**	**1**	**4**	**2**	**3**	**4**
3	3	3	3	1	1	1	1	1	
	5	5	5	5	5	4	2	3	
		6	6	6	6	6	6	6	
M	M	M	H	M	H	M	M	M	

	3	**5**	**6**	**1**	**4**	**2**
Freq.	0+1	0	2	2	0	1->0

Least Frequently Used

Consider the page reference string of size **10**: **3,5,6,6,1,1,4,2,3,4** with frame size **3**(i.e. maximum **3** pages in a frame).

3	**5**	**6**	**6**	**1**	**1**	**4**	**2**	**3**	**4**
3	3	3	3	1	1	1	1	1	1
	5	5	5	5	5	4	2	3	4
		6	6	6	6	6	6	6	6
M	M	M	H	M	H	M	M	M	M

	3	**5**	**6**	**1**	**4**	**2**
Freq.	1->0	0	2	2	0+1	0

Least Frequently Used

Total Page Fault = 8

Page Fault ratio = total miss/total possible cases

Page Fault ratio = 8/10

Memory Organization

Secondary Storage

- Secondary storage (auxiliary storage) is not directly accessible by the CPU.
- The computer usually uses its I/O channels to access secondary storage
- Transfers the desired data using intermediate area in primary storage.
- Secondary storage does not lose data when the device is powered off - non-volatile.

Memory Organization

Virtual Memory

- Uses backing storage e.g. hard disk as a temporary location for programs and data where insufficient RAM available
- Swaps programs and data between the hard-disk and RAM as the CPU requires them for processing
- A cheap method of running large or many programs on a computer system
- Cost is speed: the CPU can access RAM in nanoseconds but hard-disk in milliseconds
- Virtual memory is much slower than RAM

Memory Organization

- **Memory Management**
- Collection of hardware and software for managing the various programs residing in memory.
- The basic components of a memory management unit are:
  - A facility for dynamic storage relocation that maps logical memory references into physical memory addresses
  - A provision for sharing common programs stored in memory by different users
  - Protection of information against unauthorized access between users and preventing users from changing operating systems functions.