

Cache Mapping and Input/Output Organisation

Presented by: Mr Akshay Kumar

Associate Professor, SOCIS, IGNOU

For

MCA/BCA /PGDCA Students of IGNOU

Reference Material of IGNOU

Block 2 of MCS-012

Outline of Presentation

- Cache Memory
- I/O Techniques
- I/O Devices – Magnetic Disk and CD

The Memory Hierarchy

- Big Memory is cheap but slow
- Fast Memory expensive, therefore, small
- Increase performance by having “hierarchy”
 - CPU - Registers
 - Internal or Main memory
 - Cache memory
 - Main Memory
 - I/O Interfaces
 - Devices

Mapping

- Mapping of Main Memory to Cache.
 - Where can CPU find a block of Main Memory in Cache?
 - Three basic schemes-
 - Direct Mapping
 - Associative Mapping
 - Set-Associative mapping

Cache Memory

- Main Memory Address
 - Dividing main memory into Blocks
- Cache LINES /Slots
 - Cache Address
- Where a memory Block of Data Can be found in Cache?
 - Determined by TAG no

An Example

- Main Memory Size – 256 Byte
- Byte addressable Memory
- Memory word = 1 Byte
 - 256 Words = 2^8
 - The Main Memory Address is 8 bit long.
- Cache of size – 32 Bytes
- A Cache Line is of 2 bytes
 - The cache has 16 (2^4) lines of 2 bytes each
 - Memory Block size = 2 bytes

Tag	Li#	Word
	0000	Byte 0
		Byte 1
	0001	Byte 0
		Byte 1

	1111	Byte 0
		Byte 1

Bl#	Word
0000 000	0 Byte
	1 Byte
0000 001	0 Byte
	1 Byte
0000 010	0 Byte
	1 Byte
...	...
1111 111	0 Byte
	1 Byte

Modulo function with 4

Decimal	Quotient	Remainder	Binary	Quotient	Remainder
0	0	0	0000	00	00
1	0	1	0001	00	01
2	0	2	0010	00	10
3	0	3	0011	00	11
4	1	0	0100	01	00
5	1	1	0101	01	01
6	1	2	0110	01	10
7	1	3	0111	01	11

Modulo function with 4

Decimal	Quotient	Remainder	Binary	Quotient	Remainder
8	2	0	1000	10	00
9	2	1	1001	10	01
10	2	2	1010	10	10
11	2	3	1011	10	11
12	3	0	1100	11	00
13	3	1	1101	11	01
14	3	2	1110	11	10
15	3	3	1111	11	11

Direct Mapping

- Each block of main memory can be placed ONLY in ONE cache line
- Size of main memory address: 8 bits
 - Least Significant 1 bit identifies unique word (byte in this case) in a memory block of 2 words
 - Most Significant 7 bits specify memory blocks

Memory Block Address			Word
111	1	111	0

Bl#	Word
000	0Byte 0
0000	1Byte 1
000	0Byte 0
0001	1Byte 1
000	0Byte 0
0010	1Byte 1
...	...
111	0 Byte
1111	1Byte 1

Direct Mapping Address Structure

- Cache Address – Cache size 32 Bytes
 - 1 bit word identifier (2 byte) in a cache line
 - 4 bit line number or index (0 to 15)

Memory Block Address		Word
111	1111	0
Tag	Line no	

Tag	Li#	Word
	0000	Byte 0
		Byte 1
	0001	Byte 0
		Byte 1

111	1111	Byte 0
		Byte 1

Bl#	Word
000 0000	Byte 0
	Byte 1
000 0001	Byte 0
	Byte 1
000 0010	Byte 0
	Byte 1
...	...
111 1111	Byte 0
	Byte 1

Cache line/ Index	Main Memory blocks held
0000	0, 16 , 32, 48, 64, 80, 96, 112
0001	1 , 17, 33, 49, 65, 81, 97, 113
.....
1111	15, 31, 47, 63, 79, 95, 111, 127

Memory Block Address		Word
001	0000	1
Tag	Line no	

Example

Tag	Li#	Word
001	0000	Byte 0
		Byte 1
000	0001	Byte 0
		Byte 1

111	1111	Byte 0
		Byte 1

Bl#	Word
000	Byte 0
0000	Byte 1
000	Byte 0
0001	Byte 1
000	Byte 0
0001	Byte 1
001	Byte 0
0000	Byte 1
111	Byte 0
1111	Byte 1

Why Direct Mapping?

- It is very Simple
- It is very Inexpensive
- One major problem
 - If a program accesses two blocks that map to the same line repeatedly, then every access will result in cache miss.

Associative Mapping

- A main memory block can load into any line of cache
- Tag uniquely identifies block of memory
- Cache need to be examined simultaneously, therefore, Cache searching gets expensive on hardware

Memory Block Address	Word
1111111	0
Tag	

Tag	Word
1111111	Byte 0
	Byte 1
0000000	Byte 0
	Byte 1
...	...
0101010	Byte 0
	Byte 1

Bl#	Word
000	Byte 0
0000	Byte 1
000	Byte 0
0001	Byte 1
000	Byte 0
0010	Byte 1
...	...
...	...
111	Byte 0
1111	Byte 1

08 February 2009

Two-way set Associative Mapping

- Divides the Cache into a number of SETs
- If each of these sets contains a fixed number of lines – suppose each set contains TWO LINES then it is called TWO-WAY Set Associative Mapping
- A block of main memory can be mapped to any LINE of only ONE SET.

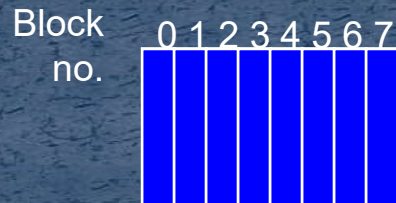
Memory Block Address		Word
1111	111	0
Tag	Set No	

Tag	W	S#	Tag	W	Bl#	Word
	B ₀	000	0000	B ₀	0000	Byte 0
	B ₁			B ₁	000	Byte 1
0000	B ₀	001		B ₀	0000	Byte 0
	B ₁			B ₁	001	Byte 1
	0011	Byte 0
					111	Byte 1
0011	B ₀	111	1111	B ₀
	B ₁			B ₁		
					1111	Byte 0
					111	Byte 1

The Three Mapping Schemes

- Assumption:
 - Main Memory SIZE – 32 Blocks
 - Cache Size – 8 Blocks

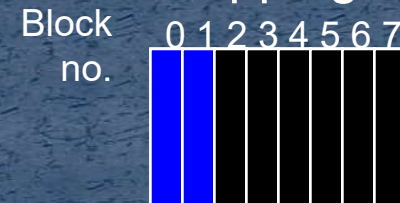
Fully
associative



Direct
mapping



2 way Set
associative
mapping



Block-frame address



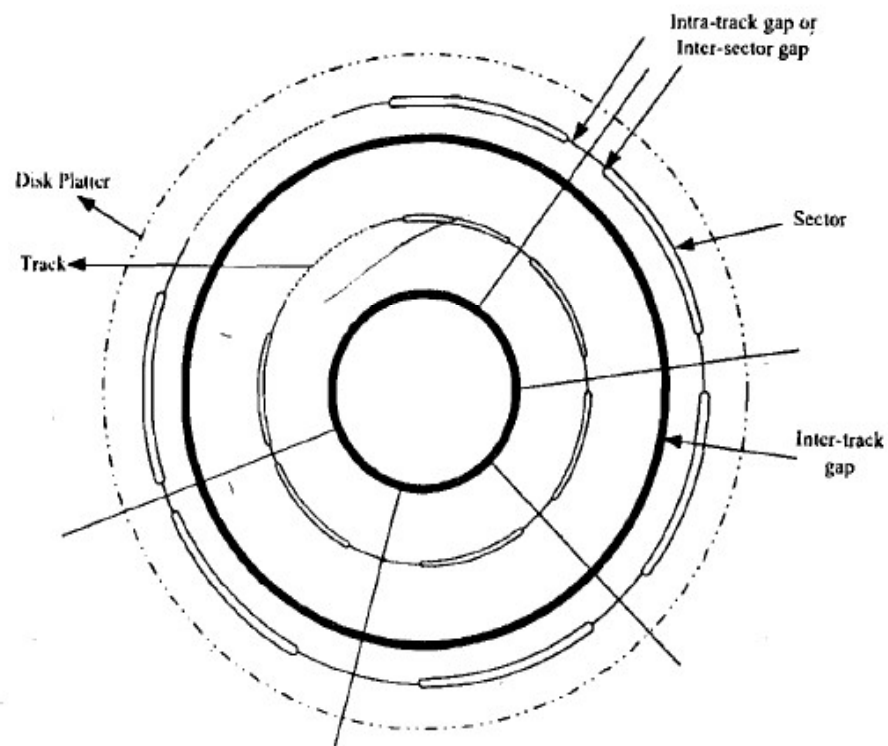
Block no.

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3

Secodary Storage Organisation

Hard Disk Basic Layout

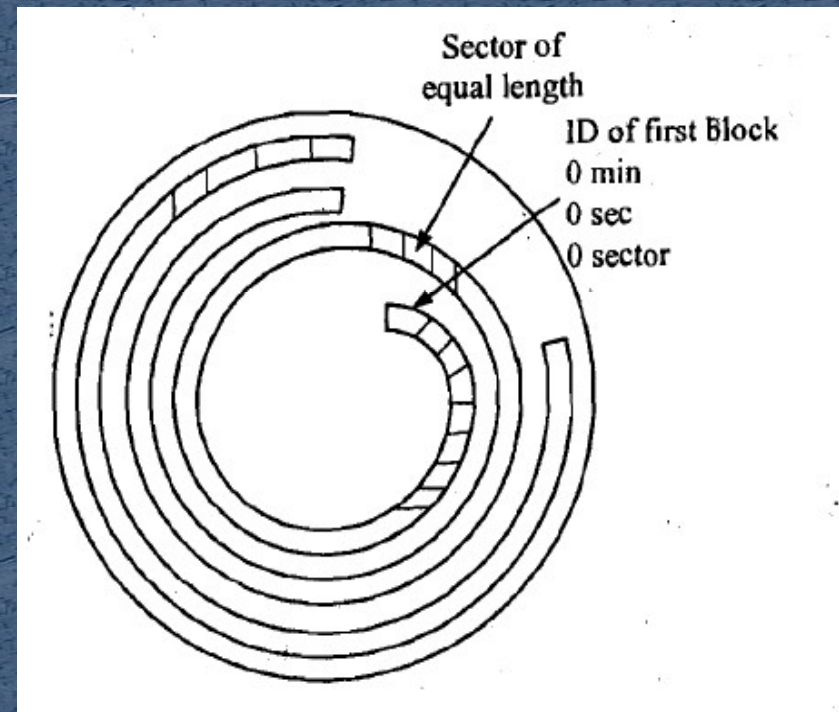
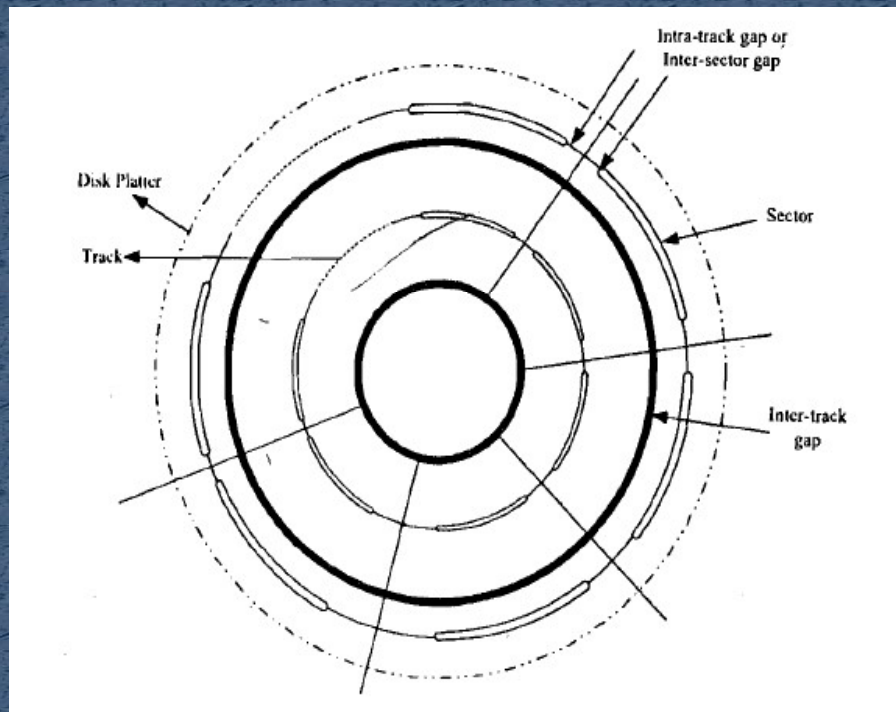


- Disk Access Time
 - Seek Time
 - Latency Time

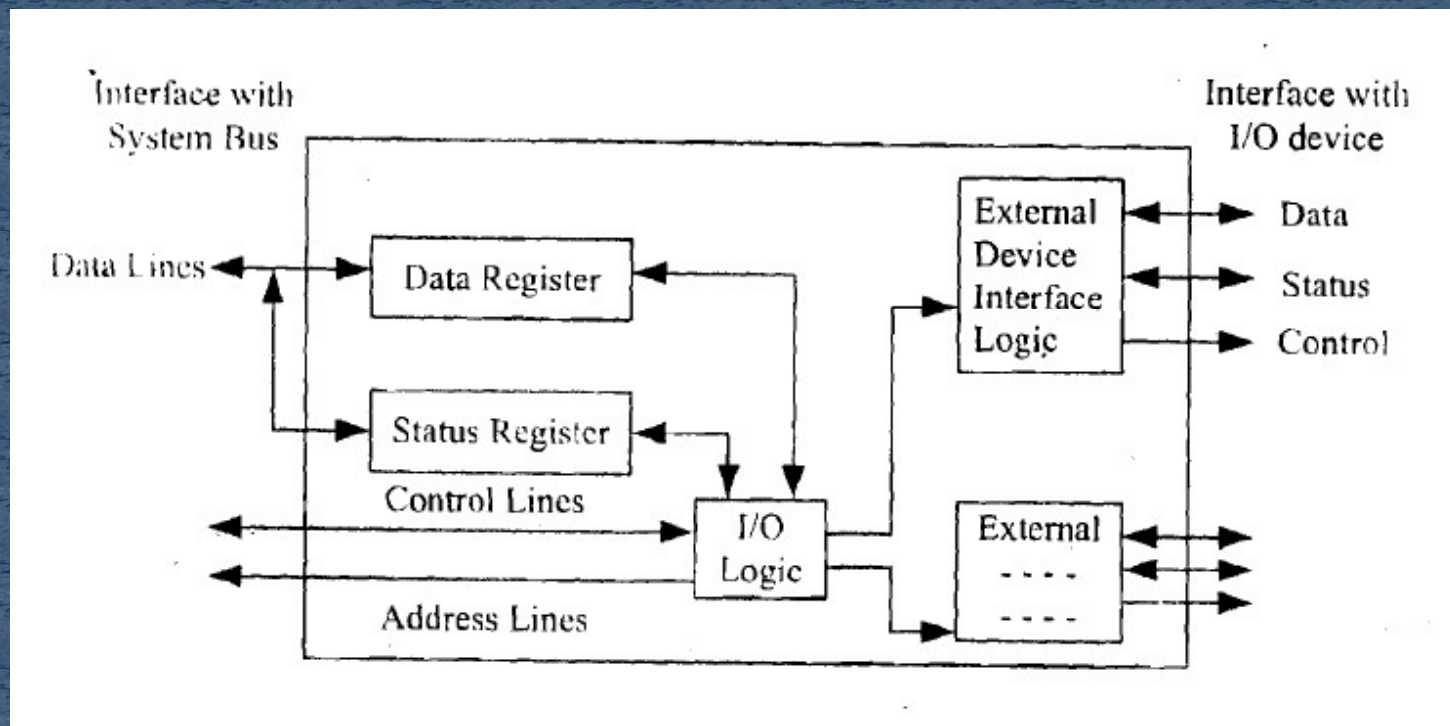
Magnetic Disks

- Direct Access Vs Random Access
- Seek Time is defined in disks
- Latency time calculation
 - Suppose a disk rotates at 10000 rpm
 - Time of one rotation = $1/10000$ minutes
 - = $60/10000$ seconds = $6/1000$ seconds = 6 milli-seconds
 - On an average half of disk rotation, so latency time
 - = $1/2 * 6 = 3$ milli-seconds

Hard Disk vs CD ROM Layout



An Interface for Input/ Output



I/O Techniques

Programmed I/O	Interrupt Driven I/O	DMA
CPU issues Read Command to I/O interface	CPU issues Read Command to I/O interface	CPU issues a Read Command for Block of data to DMA controller
CPU checks the status of I/O Device Repeatedly	CPU is Interrupted by I/O interface once task performed	On Completion DMA controller interrupts and inform CPU
CPU reads Word from I/O Interface and Writes it in memory	CPU reads Word from I/O Interface and Writes it in memory	
CPU checks for Completion	CPU checks for Completion	

Role of I/O Interface

- Control and timing signals with CPU and External Devices'
 - CPU requests data, checks status of device and responds, Gets the data and communicate to CPU
- Communicates with CPU
- Communicates with I/O device
- Data Buffering
- Error Control

Activites to be Pefromed

- Study the Block 2
- Solve questions of CYPs in the Block
- Solve questions given in assignments and previous year question papers
- Discuss with us, if there is any problem.