



Webinar-1

Performance Assessment, Disk Assessment, Cache Memory and CPU-OS Simulator





Performance Assessment

Performance Assessment-Summary

- If you were running a program on two different processors, we would say that the faster is the one that gets the job done first.
- Execution time: The total time required for the computer to complete a task, including disk accesses, memory accesses, I/O activities, operating system overhead, CPU execution

$$Performance_{x} = \frac{1}{Execution time_{x}}$$

This means that for two computers X and Y, if the performance of X is greater than the performance of Y, we have

$$\frac{1}{\text{Execution time}_{x}} > \frac{1}{\text{Execution time}_{y}}$$

$$\frac{1}{\text{Execution time}_{y}} > \frac{1}{\text{Execution time}_{y}}$$

$$\text{Execution time}_{y} > \text{Execution time}_{x}$$

Performance Assessment-Summary

- All computers are governed by a clock that determines when events take place in the hardware.
- These discrete time intervals are called clock cycles.
- The rate of clock pulses is known as the *clock rate*, or clock speed. (e.g., 4 gigahertz, or 4 GHz), which is the inverse of the clock period

CPU Performance and Its Factor

$$\frac{\text{CPU execution time}}{\text{for a program}} = \frac{\text{CPU clock cycles}}{\text{for a program}} \times \text{Clock cycle time}$$

Alternatively, because clock rate and clock cycle time are inverses,

$$\frac{\text{CPU execution time}}{\text{for a program}} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

Performance Assessment-Summary

Instruction Performance

CPU clock cycles = Instructions for a program × Average clock cycles per instruction

 The term clock cycles per instruction, which is the average number of cycles each instruction takes to execute, is often abbreviated as CPI.

CPU time = Instruction count \times CPI \times Clock cycle time

or, since the clock rate is the inverse of clock cycle time:

$$CPU time = \frac{Instruction count \times CPI}{Clock rate}$$

Performance Assessment -



Millions of Instructions per Second (MIPS) Rate

Million Instructions Per Second: The rate at which instructions are executed.

MIPS =
$$\frac{Instruction\ count}{Execution\ time \times 10^6}$$

$$MIPS = \frac{Instruction\ count}{\frac{Instruction\ count \times CPI}{Clock\ rate}} \times 10^{6} = \frac{Clock\ rate}{CPI \times 10^{6}}$$

Problem -1



A benchmark program runs on a system having clock rate of 40MHz. The program consists of 100000 executable instructions with following instruction mix and clock cycle count for each instruction type.

Instruction Type	Instruction Count (IC)	Cycles Per Instruction (CPI)
Integer Arithmetic	45000	1
Data Transfer	32000	2
Floating Point	15000	2
Control Transfer	8000	2

Determine the effective CPI, MIPS and execution time for the program

Solution -1

Given Data:

- Clock speed of the processor = 40MHz
- No. of Instructions the executed program consists of = 100000

$$CPI = \frac{Instruction count \times Cycles per second}{Number of instructions the executed program consists}$$

•
$$CPI = \frac{(45000*1) + (32000*2) + (15000*2) + (8000*2)}{100000} = \frac{155000}{100000} = 1.55$$

To calculate MIPS

MIPS =
$$\frac{Instruction\ count}{Execution\ time \times 10^6}$$

- Execution Time = Instruction Count \times CPI \times Cycle Time = $I_c \times CPI \times (1/f)$
- MIPS = $I_c / [(I_c \times CPI \times (1/f)) \times 10^6]$
- = $f / (CPIx 10^6)$
- = $(40 \times 10^6) / (1.55 \times 10^6)$
- = 25.8

Solution -1



- To calculate Execution Time
- Execution Time = Instruction Count x CPI x Cycle Time
 - = $I_c \times CPI \times (1/f)$
 - $= (100000 \times 1.55) / (40 \times 10^6)$
 - = 0.003875
 - = 3.875 ms

Problem - 2



A Netcom systems developed two computer systems C1 and C2., where C1 has machine instructions for floating point (FP) operations as part of its processor ISA and C2 does NOT have floating point instructions as part of its processor ISA. Since C2 does not have floating point instructions, all floating-point instructions will be implemented in Software level with non-FP instructions. You can assume that both systems are operating at a clock speed of 300 Mhz. We are trying to run the SAME program in both the systems which has the following proportion of commands:

Instruction type	% instructions in Program	Instruction Duration (Number of clock periods CPI)	
		C 1	C2
Addition of FP numbers	16%	6	20
Multiplication of FP numbers	10%	8	32
Division of FP numbers	8%	10	66
Misc. Instructions (non-FP)	66%	3	3

Solution-2

a) Find the MIPS for both C1 and C2.

For C1:

- CPI = 0.16*6 + 0.1*8 + 0.08*10 + 0.66*3 = 4.54
- MIPS = 300 * 10⁶ / (4.54 * 10⁶) = 66.08

For C2:

- CPI = 0.16*20 + 0.1 * 32 + 0.08 * 66 + 0.66 * 3 = 13.66
- MIPS = 300 * 10⁶ / (13.66 * 10⁶) = 21.96



Solution-2 Contd...

- b) Assume that there are 9000 instructions in the program that is getting executed on C1 and C2. What will be the CPU program execution time on each system C1 and C2?
- CPU time for C1 of the program execution

```
= No of instructions * CPI / Clock rate
```

= 9000 * 4.54 / (300*10⁶)

= 0.136 ms

CPU time for C2 of the program execution

= No of instructions * CPI / Clock rate

= 9000 * 13.66/ (300 * 10^6)

 $= 0.41 \, \text{ms}$



Solution-2 Contd...

- c) For the two systems to have the fastest speed and at the same time have equal speed, what would be the possible mixture of the instructions that would be required in the program? WHY?
- For both C1 and C2 should be equally fast,
 - Have a program that does NOT have any floating point instructions as CPI for non-floating point instructions is same between C1 and C2.

Problem -3 (Home work)

 Consider two different machines, with two different instruction sets, both of which have a clock rate of 200 MHz. The following measurements are recorded on the two machines running a given set of benchmark programs:

Instruction Type	Instruction Count (millions)	Cycles per Instruction	
Machine A			
Arithmetic and logic	8	1	
Load and store	4	3	
Branch	2	4	
Others	4	3	
Machine B			
Arithmetic and logic	10	1	
Load and store	8	2	
Branch	2	4	
Others	4	3	

 Determine the effective CPI, MIPS rate, and execution time for each machine.

Amdhal's Law



- Deals with the potential speedup of a program using multiple processors (parallel) compared to a single processor.
- It states that if P is the proportion of a program that can be made parallel and (1-P) is the proportion that cannot be parallelized (remains sequential), then the maximum speed up that can be achieved by using N processors is:

$$SpeedUp(P,N) = \frac{1}{(1-P) + \frac{P}{N}}$$

 As N tends to infinity, the maximum speedup tends to 1/(1-P)

Problem -4 (Amdhal's Law)

- A programmer is given the job to write a program on a computer with processor having speedup factor 3.8 on 4 processors. He makes it 95% parallel and goes home dreaming of a big pay raise. Using Amdahl's law, and assuming the problem size is the same as the serial version, and ignoring communication costs, what is the speedup factor that the programmer will get?
- Solution:
- Given Data: No. of Processors (N) = 4
- Proportion of parallelism = 95%
- Speed up = 1/[(1-P)+P/N]= 1/[(1-0.95)+0.95/4]= 3.47



Problem - 5 (Home Work)

A programmer has parallelized 99% of a program, but there
is no value in increasing the problem size, i.e., the program
will always be run with the same problem size regardless of
the number of processors or cores used. What is the
expected speedup on 20 processors?





Disk Assessment

Disk Access Time

- Average time to access some target sector given by :
 - Taccess = Tavg seek + Tavg rotation + Tavg transfer
- Seek time (Tavg seek)
 - Time to position heads over cylinder containing target sector.
 - Typical Tavg seek is 3-9 ms
- Rotational latency (Tavg rotation)
 - Time waiting for first bit of target sector to pass under r/w head.
 - Tavg rotation = 1/2r, where r is rotation Speed in **revolution per Second**
 - Typical Tavg rotation = 7200 RPMs = 7200/60 RPS
- Transfer time (Tavg transfer)
 - Time to read the bits in the target sector.
 - Tavg transfer = b/rN, where b is the number of bytes to be transferred and N is the average number of bytes on a track

Disk Access Time Example

Given:

- Rotational rate = 7,200 RPM
- Average seek time = 9 ms.
- Avg # sectors/track = 400.
- 512 bytes per sector

Derived:

- Tavg rotation = $1/2r = 1/2 \times (60 \text{ secs}/7200 \text{ RPM})$ = 0.00416 = 4.16 ms.
- Tavg transfer = b/rN= $512 \times 60/7200 \times 1/(400*512)$ = 0.02 ms
- Taccess = 9 ms + 4.16ms + 0.02 ms=13.18ms

Contd...

Important points:

- Access time dominated by seek time and rotational latency.
- First bit in a sector is the most expensive, the rest are free.
- SRAM access time is about 4 ns/doubleword,
 DRAM about 60 ns
 - Disk is about 40,000 times slower than SRAM,
 - 2,500 times slower then DRAM.

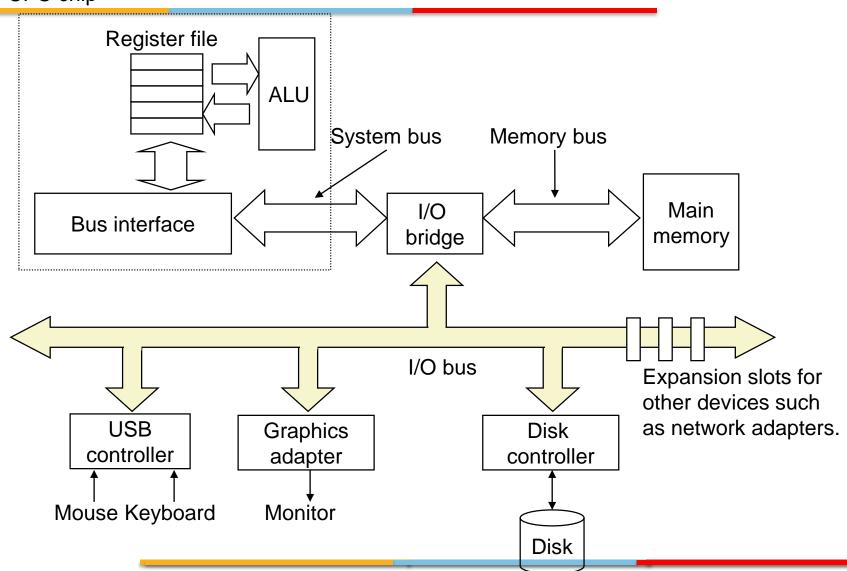
Logical Disk Blocks

- Modern disks present a simpler abstract view of the complex sector geometry:
 - The set of available sectors is modeled as a sequence of b-sized logical blocks (0, 1, 2, ...)
- Mapping between logical blocks and actual (physical) sectors
 - Maintained by hardware/firmware device called disk controller.
 - Converts requests for logical blocks into (surface, track, sector) triples.
- Allows controller to set aside spare cylinders for each zone.
 - Accounts for the difference in "formatted capacity" and "maximum capacity".

lead

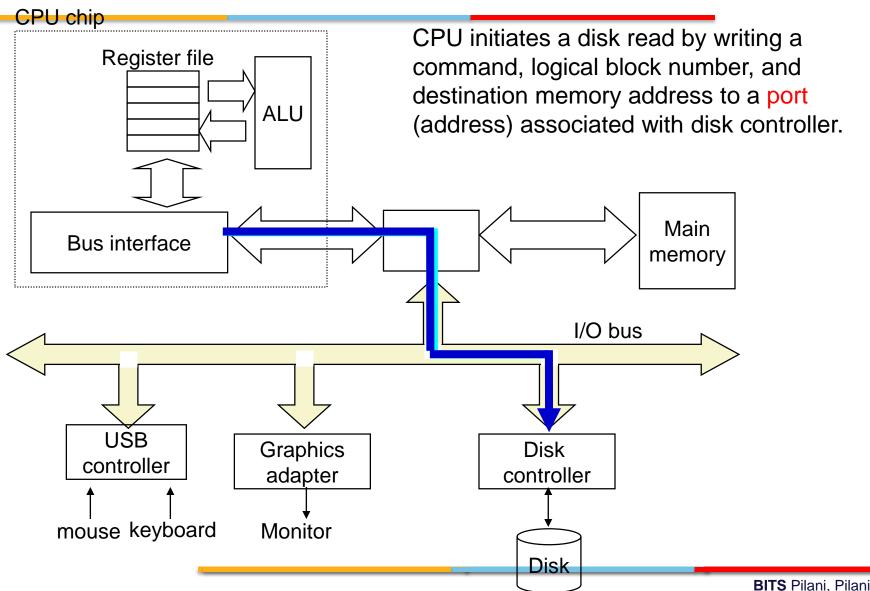
I/O Bus

CPU chip

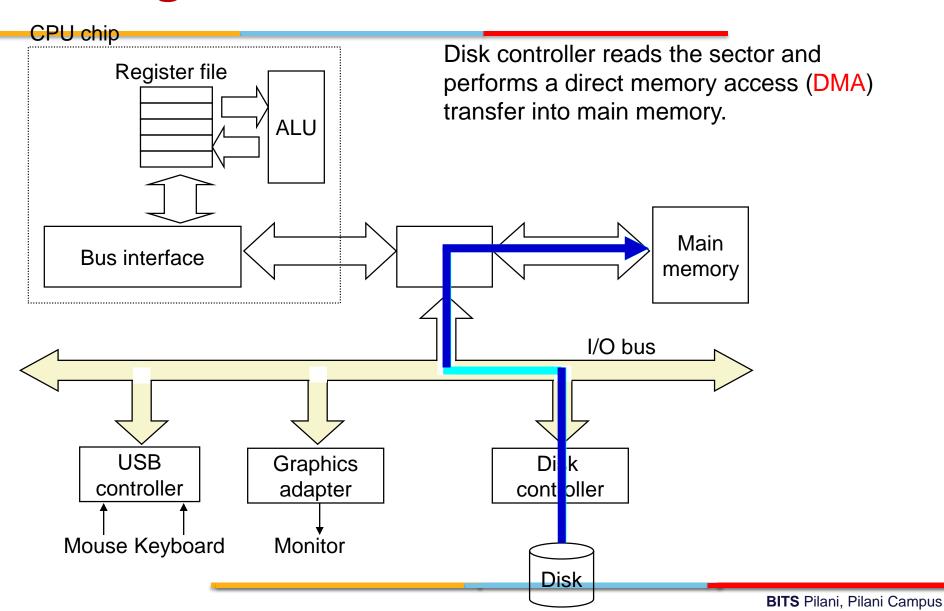


achieve lead innovate

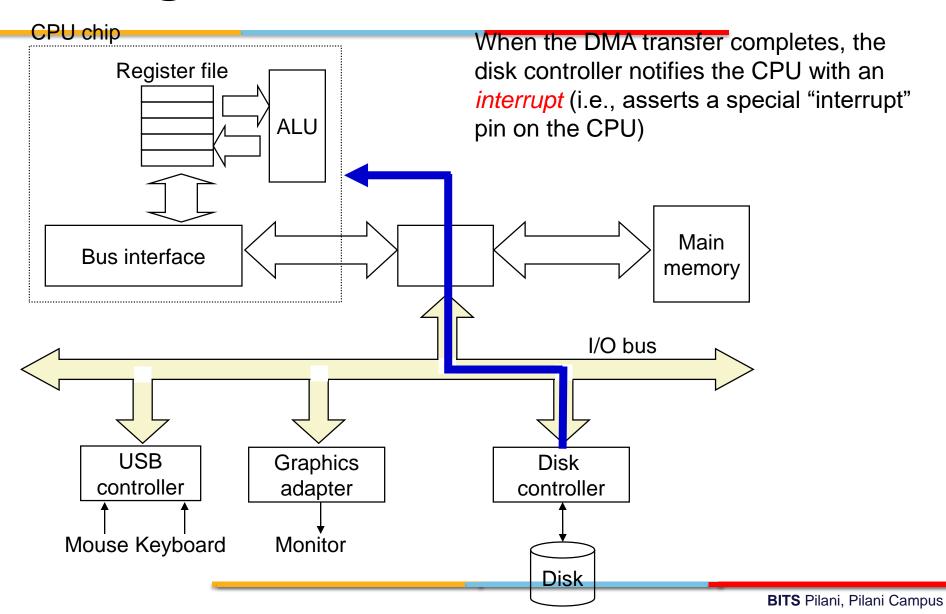
Reading a Disk Sector (1)



Reading a Disk Sector (2)

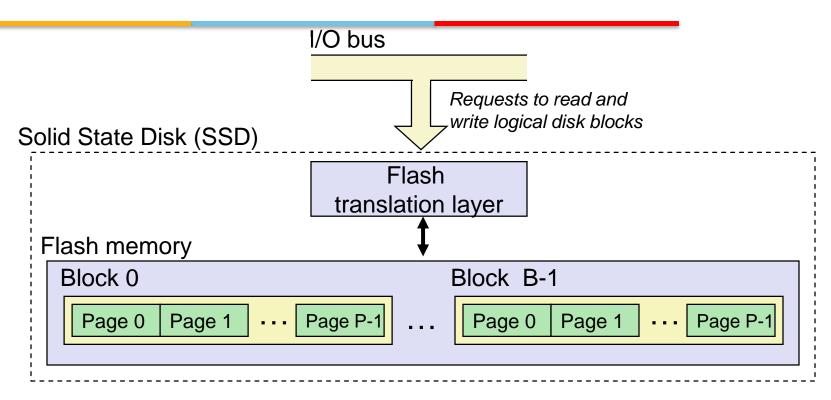


Reading a Disk Sector (3)



Solid State Disks (SSDs)





- Pages: 512KB to 4KB, Blocks: 32 to 128 pages
- Data read/written in units of pages.
- Page can be written only after its block has been erased
- A block wears out after about 100,000 repeated writes.

Sequential read tput	550 MB/s	Sequential write tput	470 MB/s
Random read tput	365 MB/s	Random write tput	303 MB/s
Avg seq read time	50 us	Avg seq write time	60 us

- Sequential access faster than random access
 - Common theme in the memory hierarchy
- Random writes are somewhat slower
 - Erasing a block takes a long time
 - Modifying a block page requires all other pages to be copied to new block
 - In earlier SSDs, the read/write gap was much larger.

SSD Tradeoffs vs Rotating Disks

- Advantages
 - No moving parts □ faster, less power, more rugged
- Disadvantages
 - Have the potential to wear out
 - Mitigated by "wear leveling logic" in flash translation layer
 - E.g. Intel SSD 730 guarantees 128 petabyte (128 \times 10¹⁵ bytes) of writes before they wear out
 - In 2015, about 30 times more expensive per byte
- Applications
 - MP3 players, smart phones, laptops
 - Beginning to appear in desktops and servers

Problem-6



- Consider a disk pack with the following specifications- 16 surfaces, 128 tracks per surface, 256 sectors per track and 512 bytes per sector. Answer the following questions
 - a) What is the capacity of disk pack?
 - b) What is the number of bits required to address the sector?
 - c) If the disk is rotating at 3600 RPM, what is the data transfer rate?
 - d) If the disk system has rotational speed of 3000 RPM, what is the average access time with a seek time of 11.5 msec?

Given Data:

- Number of surfaces = 16
- Number of tracks per surface = 128
- Number of sectors per track = 256
- Number of bytes per sector = 512 bytes
- a) What is the capacity of disk pack?
- Disk Capacity = # of surfaces x # of tracks per surface x # of sectors per track x # of bytes per sector
- Disk Capacity = $16 \times 128 \times 256 \times 512$ bytes. = 2^{28} Bytes = 2^{56} MB

Solution - 6



b) What is the number of bits required to address the sector?

Total number of sectors = Total number of surfaces x Number of tracks per surface x Number of sectors per track

= $16 \times 128 \times 256$ sectors

 $= 2^4 \times 2^7 \times 2^8$ Sectors

= 2¹⁹ sectors

Thus, Number of bits required to address the sector = 19 bits

Solution - 6



c) If the disk is rotating at 3600 RPM, what is the data transfer rate?

Number of rotations in one second = (3600 / 60) rotations/sec = 60 rps

Data transfer rate = Number of heads x Capacity of one track x Number of rotations in one second

- = $16 \times (256 \times 512 \text{ bytes}) \times 60$
- $= 2^4 \times 2^8 \times 2^9 \times 60$ bytes/sec
- = 60×2^{21} bytes/sec
- = 120 MBps

Solution - 6 (Homework)



d) If the disk system has rotational speed of 3000 RPM, what is the average access time with a seek time of 11.5 msec?

- Taccess = Tavgseek + Tavgrotation + Tavgtransfer
- Rotational Rate = 3000 rpm
- Seek Time = = _____
- Avg# sectors/track = 256
- 512 bytes per sector

$$T_{\text{avgrotation}} = \frac{1}{2}r = \frac{1}{2}*(60/3000) ==$$

transferred & N is the average number of bytes on a track)

$$T_{access} = =$$





Cache Memory - Direct Mapping

BITS Pilani

Pilani Campus



Direct Mapped Cache - Summary

- Each block of main memory maps to only one cache line
 - if a block is in cache, it must be in one specific place
 - i = j modulo m
 where i = cache line number
 j = main memory block no.
 m = no.of lines in the cache
- Address is split in three parts:
 - Tag
 - Line
 - Word

Problem - 7: Direct mapping movate

Consider a cache memory of 8192KB with line size of 128B. What is the tag, line and word bits for the main memory address 0xFEEDF00D?

Solution:

- Cache Size = 8192 KB (Given)
- Total No. of Cache Lines = Cache size / Line size

```
= 8192 KB/128B
```

$$= (2^{13}.2^{10}) / 2^7 = 2^{16} = 64$$
K lines

- · Total bits to represent line field
 - = 16 bits
- Line size = block size = $128 B = 2^7$
- Total bits to represent Block offset = 7 bits



Problem- 7: Direct mapping

- Total bits to represent Tag field is:
 - = 32 Line (16) Word (7) = 9 bits
- Tag, Line and Word bits for the given address:
- Given Address: 0xFEEDF00D

F	E	Е	D	F	0	0	D
1 1 1 1	1 1 1 0	1 1 1	0 1 1 0 1	1 1 1 1	0000	0000	1 1 0 1
Tag l	bits (9)		Line bits (16)				offset (7)





Associative Mapping

BITS Pilani

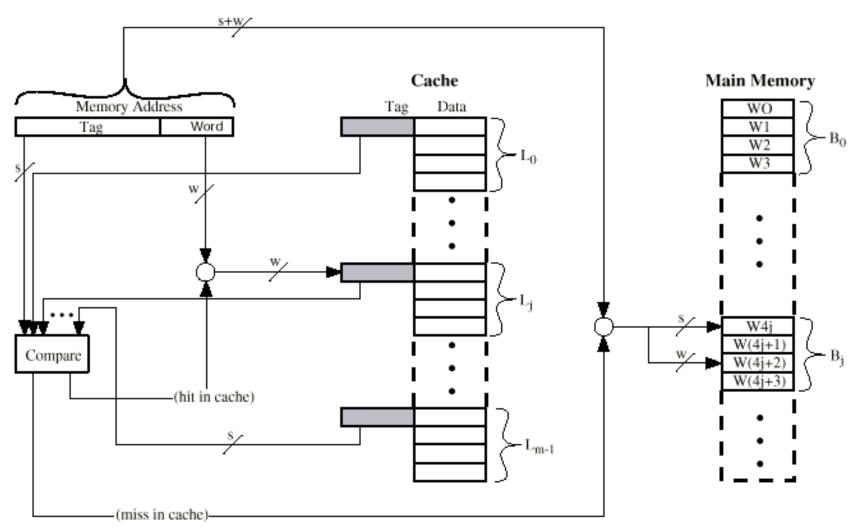
Pilani Campus



Associative Mapping

- A main memory block can load into any line of cache
- Memory address is interpreted as tag and word
- · Tag uniquely identifies block of memory
- Every line's tag is examined for a match
- Cache searching gets expensive

Associative Cache Organization



Associative Mapping Summary

- Address length = (s + w) bits
- Number of addressable units = 2^{s+w} words or bytes
- Block size = line size = 2^w words or bytes
- Number of blocks in main memory = 2^{s+ w}/2^w = 2^s
- Number of lines in cache = undetermined
- Size of tag = s bits

Problem 8: Fully Associative



A system has main memory of size 128Byte with on chip cache 32 Bytes and block size of 8 Bytes, with the system having fully associative cache mapping, find the following:

a) The number of main memory address bits.

Main Memory Size = 128 Bytes = 2^7 Bytes Thus, the size of Physical Address = 7 bits.

b) The number of Tag bits and Word bits.

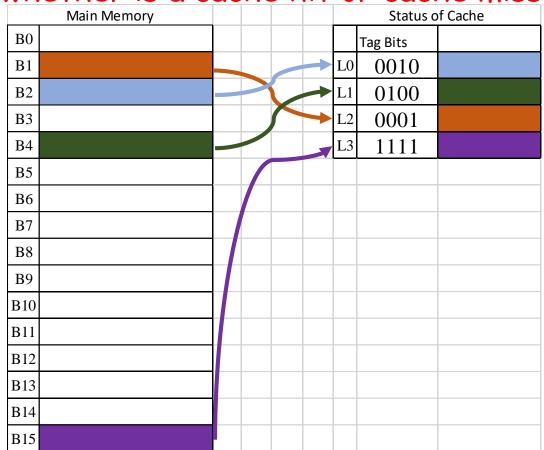
Block size = 8 Bytes = 23 Bytes # of bits for WORD offset field = 3 bits # of Tag bits = 7 - WORD offset = 7-3 = 4 bits

7-Bits					
4-bits	3-bits				
TAG bits	WORD or BLOCK OFFSET				

Fully Associative



c) If the CPU request the addresses 0001100, 0011001 find whether is a cache hit or cache miss for each of the address.



0001100

0001100 - Hit

0011001

0011001-Miss

Problem 9

Given:

- Cache of 128kByte, Cache block of 8 bytes
- 32 MBytes main memory. Find out
- a) Number of bits required to address the memory
- b) Number of blocks in main memory
- c) Number of cache lines
- d) Number of bits required to identify a word (byte) in a block?
- e) Write the Physical Address format with Tag, Word bits.

Problem 9

Given:

- Cache of 128KByte, Cache block of 8 bytes
- 32 MBytes main memory. Find out
- a) Number of bits required to address the memory Main memory size = $32MB = 2^5$. 2^{20} Bytes = 2^{25} Bytes No. of Bits required to address memory = 25 bits
- b) Number of blocks in main memory

Block size = line size = 2^w words or bytes Number of blocks in main memory = $2^{s+w}/2^w = 2^s$ No. of blocks in main memory = $2^{25}/2^3 = 2^{22} = 4M$ blocks

Problem 9

Given:

c) Number of cache lines = 16K Lines

Total number of lines in cache = Cache size / Line size = $128K/8 = 2^{17}/2^3 = 2^{14} = 16K$

d) Number of bits required to identify a word (byte) in a block?

Given that each cache block size = 8 Bytes

Block size = line size = $8 \text{ bytes} = 2^3 \text{ Bytes} = 3 \text{ bits}$

e) Write the Physical Address format with Tag, Word bits.

Tag (22 bits) Word (3 bits)

25 bits

Problem 10 (Home Work)

Cache of 64kByte, Cache block of 4 bytes and 16 M Bytes main memory and associative mapping.

Fill in the blanks:

Number of bits in main memory address = _____

Number of lines in the cache memory = _____

Word bits = _____

Tag bits = _____

Problem 11: Fully Associative move

An 8KB associative cache has a line size of 32 Bytes. Main memory size is 1GB. Find the number of TAG bits and number of comparators required for search.

Given Data:

Cache size =
$$8KB = 2^3.2^{10} = 2^{13}$$
 Bytes
Block Size = 32 Bytes = 2^5 Bytes
Memory size = $1GB = 2^{30}$ Bytes

- # of bits for main memory address = 30 bits
- # of bits for Word offset field = 5 bits
- # of bits for Tag field = 30 5 = 25 Bits

30 bits				
25	5			
Tag	Word			

Problem 11: Fully Associative

- # of cache lines
 = (Cache Size)/(Line Size) = 2¹³/2⁵ = 2⁸ lines = 256 lines
- # of Comparators required = # of Cache lines = 256
- Size of comparator = size of tag bits
 = 25-bit comparator





CPU-OS Simulator

BITS Pilani

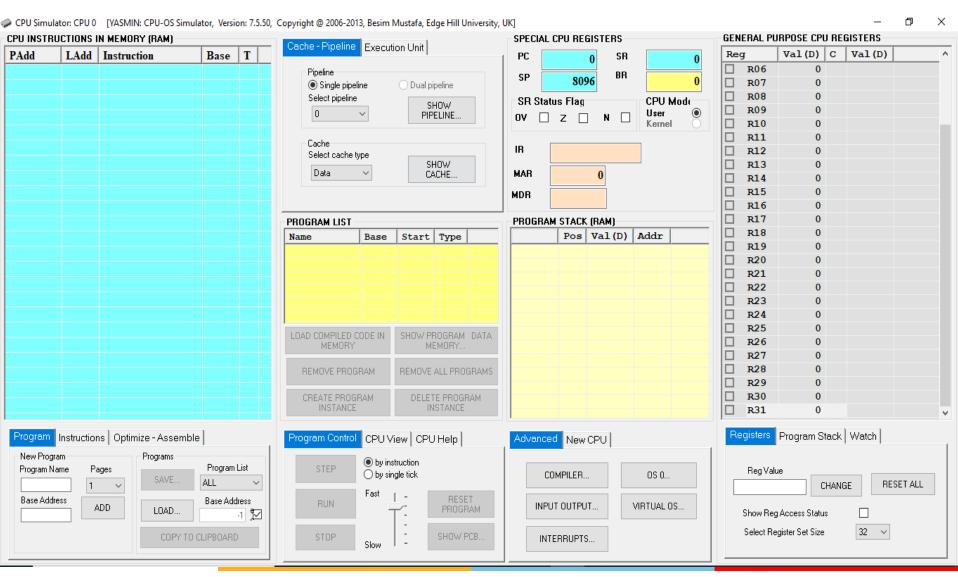
Pilani Campus

CPU OS Simulator

- Components of CPU Simulator
- Execute a program
- Cache updates
- Assignment

CPU-OS Simulator Interface





How to Simulate and Run the program



Simulator Teaching Language (STL)

```
program SelectionSort
VAR a array(10) INTEGER
a(1)=15
a(2)=20
a(3)=8
a(4)=100
a(5)=30
for n = 1 to 5
writeln("num",a(n))
next
for i = 1 to 5
for j = i+1 to 5
p = a(i)
q = a(j)
if p > q then
x = p
a(i) = q
a(i) = x
end if
next
next
writeln("Sorted Array in ascending order")
for n = 1 to 5
writeln("num",a(n))
next
end
```

How to Simulate and Run the program



Simulator Teaching Language (STL)

a) Execute the above program by setting block size to 2, 4, 8, 16 and 32 for cache size = 8, 16 and 32. Record the observation in the following table.

_	arion in the following rabio.							
	Block Size	Cache size	# Hits	# Misses	% Miss Ratio	%Hit Ratio		
	2	8						
	4							
	8							
	2	16						
	4							
	8							

b) Plot a single graph of Cache hit ratio Vs Block size with respect to cache size = 8, 16 and 32. Comment on the graph that is obtained.

Web link for CPU-OS Simulator

Resources

- Link to download CPU-OS Simulator
- https://drive.google.com/drive/folders/12YUK52RQ-JhPOddj6CD_oifW4sTMbsBl?usp=sharing
- Introduction to CPU Simulator
- https://www.youtube.com/watch?v=izcvk0x7kKM&list=PLGz _KyS7cuuWGqFbBITk7Nr4PoVmAWkE0
- Installation on Mac OS
- https://www.youtube.com/watch?v=2vBjXUbvCKs