Webinar-1

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

by - VIVEKANANDA M R

BITS Pilani
Pilani Campus

Performance Assessment

BITS Pilani
Pilani Campus

Performance Assessment-Summary

innovate achieve lead

- 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

$$\text{Performance}_x = \frac{1}{\text{Execution time}_x}$$

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

$$\text{Performance}_x > \text{Performance}_y$$

$$\frac{1}{\text{Execution time}_x} > \frac{1}{\text{Execution time}_y}$$

$$\text{Execution time}_y > \text{Execution time}_x$$

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

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

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

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

Performance Assessment-Summary

Instruction Performance

$$\text{CPU clock cycles} = \text{Instructions for a program} \times \frac{\text{Average clock cycles}}{\text{per instruction}}$$

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

$$\text{CPU time} = \text{Instruction count} \times \text{CPI} \times \text{Clock cycle time}$$

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

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

Performance Assessment –

Millions of Instructions per Second (MIPS) Rate

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

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

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

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

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

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

Solution -1

- To calculate MIPS

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

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

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)	
		C1	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 * 10^6 / (4.54 * 10^6) = 66.08

For C2:

- CPI = 0.16*20 + 0.1 * 32 + 0.08 * 66 + 0.66 * 3 = 13.66
- MIPS = 300 * 10^6 / (13.66 * 10^6) = 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 / MIPS * 10^6

  = 9000*4.54 / (300*10^6)

  = 0.136 ms

- CPU time for C2 of the program execution

  = No of instructions / MIPS * 10^6

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

  = 0.41 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

BITS Pilani
Pilani Campus

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?

Solution-6

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 x 128 x 256 x 512 bytes.

  $= 2^{28}$ Bytes = **256 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 x 128 x 256 sectors

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

= **2^{19} 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 x (256 x 512 Bytes) x 60

= 2^4 x 2^8 x 2^9 x 60 Bytes/sec

= 60 x 2^{21} Bytes/sec

= **120 MBps**

Solution - 6

d) If the disk system has rotational speed of 3600 RPM, what is the average access time with a seek time of 11.5 msec?

- $T_{access} = T_{avgseek} + T_{avgrotation} + T_{avgtransfer}$
- Rotational Rate = 3600 rpm
- Seek Time = 11.5 ms
- Avg# sectors/track = 256
- 512 bytes per sector

$T_{avgrotation} = 1/2r = \frac{1}{2}*(60/3600) =$ 8.33ms

$T_{avgtransfer}$ =b/rN (b - number of bytes to be transferred & N is the average number of bytes on a track)

= 512 *(60/3600)*1/(256*512)=0.065ms

T_{access} =11.5ms+8.33ms+0.065ms

= **19.39ms**

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

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
  = $(2^{13}.2^{10}) / 2^{7} = 2^{16}$ = 64K lines
- Total bits to represent line field
  = 16 bits

- Total bits to represent Tag filed is:
  = (32 – Line – Word) = 9 bits

Problem- 7: Direct mapping

- Tag, Line and Word bits for the given address:
- Given Address : 0xFEEDF00D

F				E				E				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	0	0	0	0	0	0	0	0	1	1	0	1
Tag bits (9)									**Line bits (16)**																**Block offset (7)**							

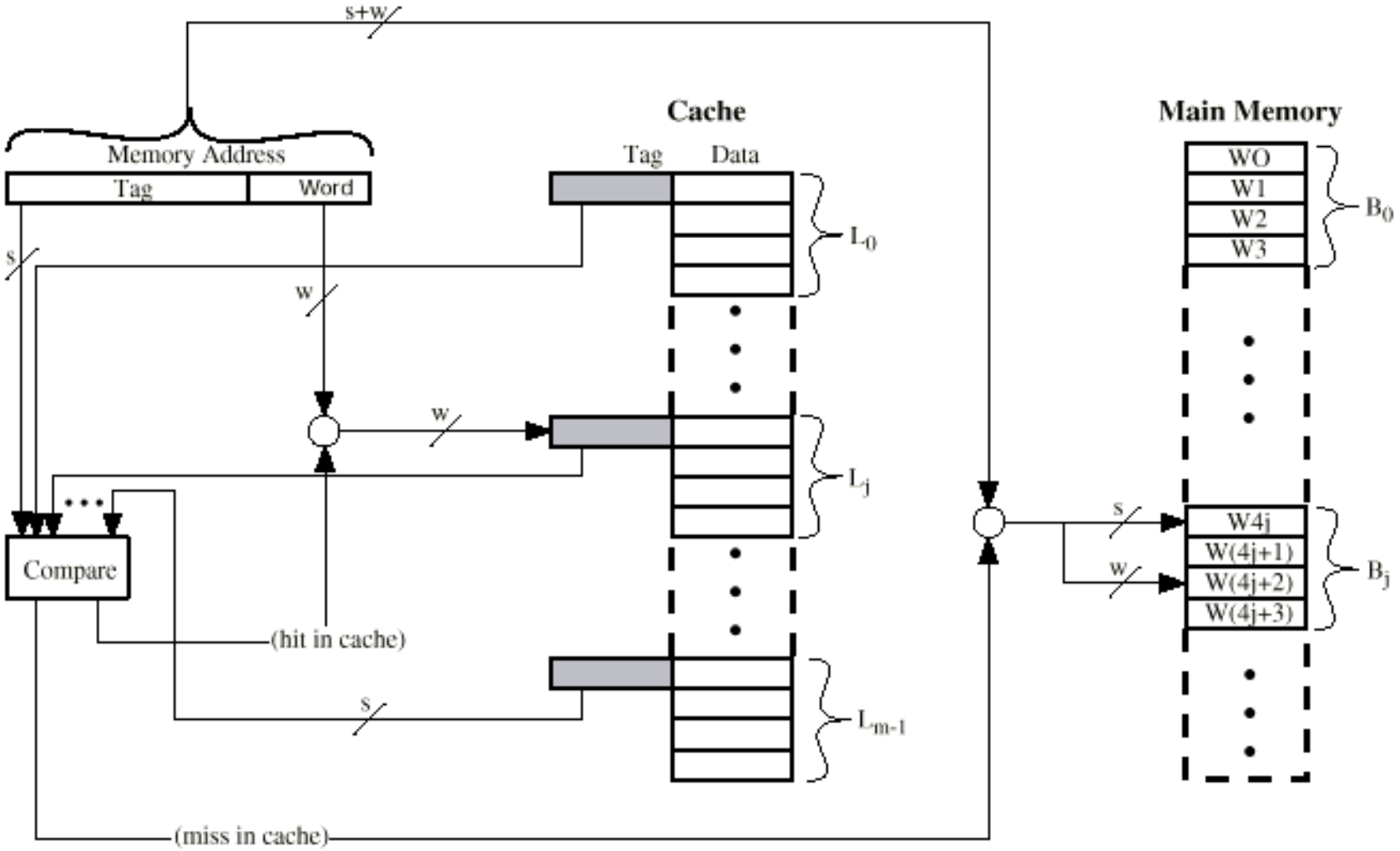
Associative Mapping

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

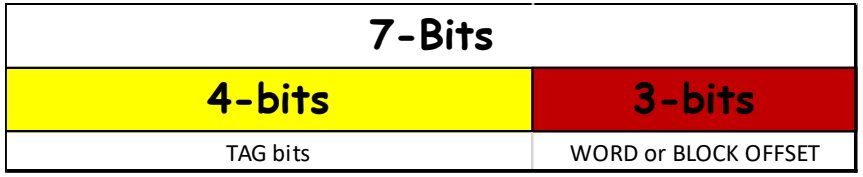
Thus, the size of Physical Address = 7 bits.

b) The number of Tag bits and Word bits.

Block size = 8 Bytes = 2^3 Bytes

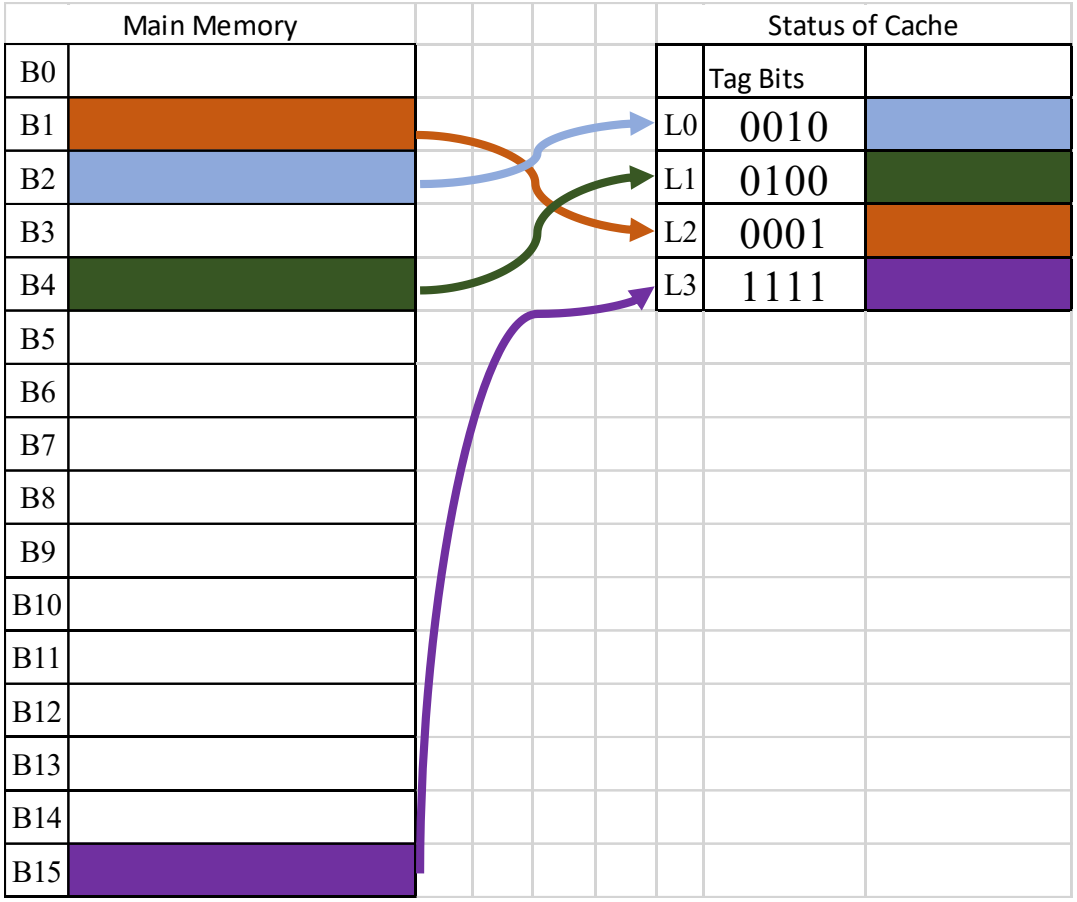
# of bits for WORD offset field = 3 bits

# of Tag bits = 7 – WORD offset = 7-3 = 4 bits



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 bytes $= 2^{3}$ Bytes = **3 bits**

e) Write the Physical Address format with Tag, Word 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

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.

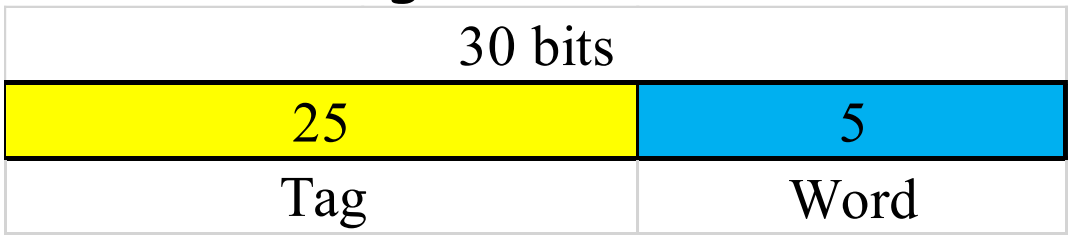
Given Data:

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



Problem 11: Fully Associative

- # of cache lines
  = (Cache Size)/(Line Size) = $2^{13}/2^5 = 2^8$ lines = 256 lines
- # of Comparators required = # of Cache lines = 256
- Size of comparator = size of tag bits
  = 25-bit comparator

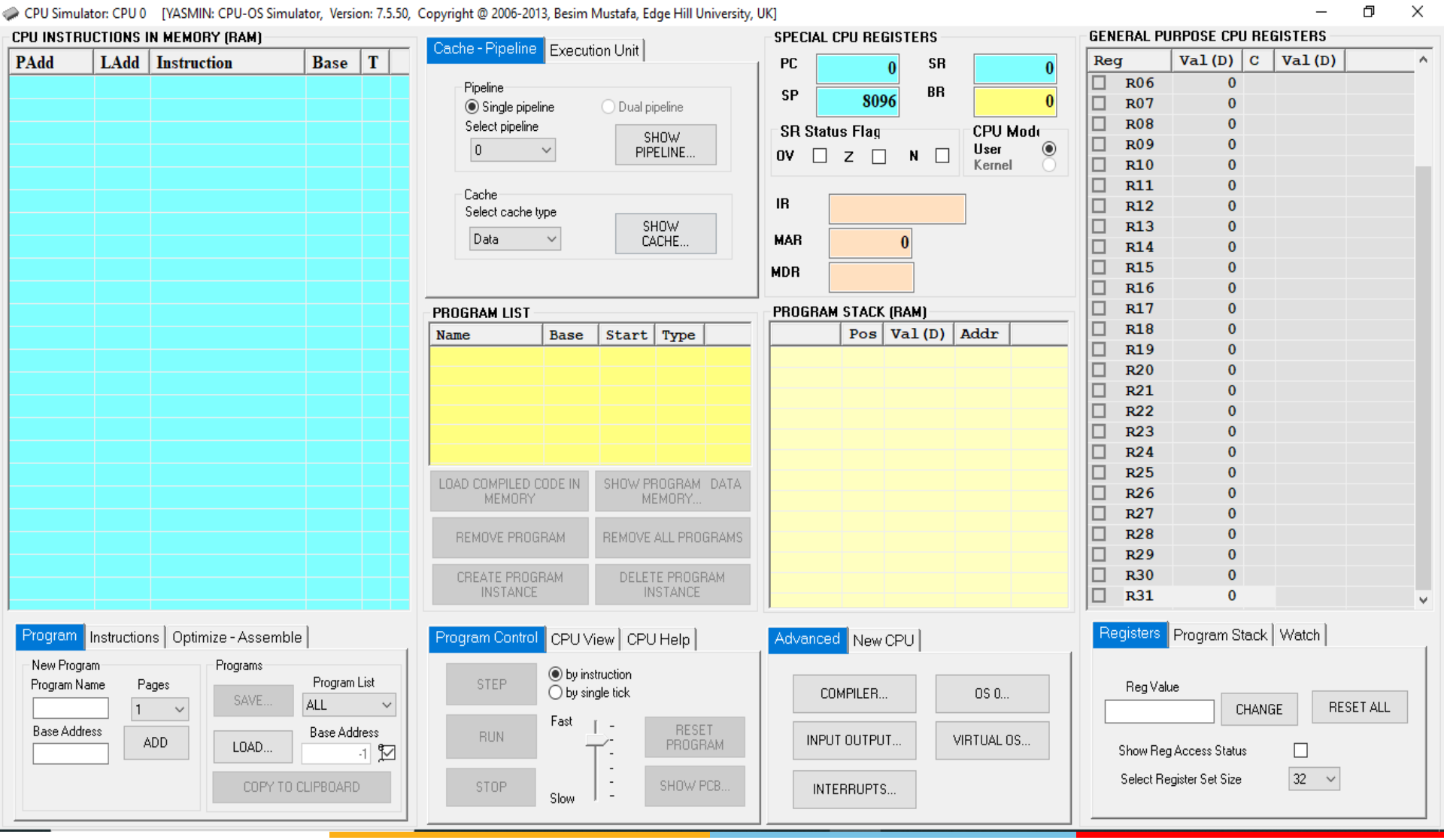
CPU-OS Simulator

BITS Pilani
Pilani Campus

How to write a new Program

- Start CPU simulator and explain how to create a program.
- How to execute a program?
- Where we can check the stored result?

CPU-OS Simulator Interface



STL →

How to Simulate and Run the program

Simulator Teaching Language (STL)

```
Program My_Prgm
i = 0
for n = 0 to 10
i = i+1
next
end
```

How to Simulate and Run the program

innovate achieve lead

Simulator Teaching Language (STL)

```
program LinearSearch
        var a array(50) byte

        for n = 0 to 10
                a(n) = n
        next

        key = 10
        found = 0

        for n = 0 to 20
                temp = a(n)
                if temp = key then
                        found = 1
                        writeln("Key Found",temp)
                        break
                end if
        next
        if found <> 1 then
                writeln("Key Not Found")
        end if
end
```

How to Simulate and Run the program

innovate achieve lead

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.

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-JhP0ddj6CD_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