Webinar-1
Performance Assessment
Error Correction
CPU-OS Simulator

BITS Pilani
Pilani Campus

Performance Assessment

(R1: 1.4, T1: 2.3)
- MIPS Rate, Amdahl's Law

BITS Pilani
Pilani Campus

Performance Assessment

- 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
  - includes 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

$$\text{Performance}_X > \text{Performance}_Y$$

$$\frac{1}{\text{Execution time}_X} > \frac{1}{\text{Execution time}_Y}$$

$$\text{Execution time}_Y > \text{Execution time}_X$$

Example

- If computer A runs a program in 10 seconds and computer B runs the same program in 15 seconds, how much faster is A than B?

$$\frac{\text{Performance}_A}{\text{Performance}_B} = \frac{\text{Execution time}_B}{\text{Execution time}_A} = n$$

- Computer A is therefore 1.5 times faster than B.

Performance Assessment

- 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 (Hertz) which is the inverse of the clock period. For example, 1 GHz processor receives 1 billion pulse / sec

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,

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

Example

- Our favorite program runs in 10 seconds on computer A, which has a 2 GHz clock. We are trying to help a computer designer build a computer, B, which will run this program in 6 seconds. The designer has determined that a substantial increase in the clock rate is possible, but this increase will affect the rest of the CPU design, causing computer B to require 1.2 times as many clock cycles as computer A for this program. What clock rate should we tell the designer to target?

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

Example…

Let's first find the number of clock cycles required for the program on A:

$$\text{CPU time}_A = \frac{\text{CPU clock cycles}_A}{\text{Clock rate}_A}$$

$$10 \text{ seconds} = \frac{\text{CPU clock cycles}_A}{2 \times 10^9 \frac{\text{cycles}}{\text{second}}}$$

$$\text{CPU clock cycles}_A = 10 \text{ seconds} \times 2 \times 10^9 \frac{\text{cycles}}{\text{second}} = 20 \times 10^9 \text{ cycles}$$

CPU time for B can be found using this equation:

$$\text{CPU time}_B = \frac{1.2 \times \text{CPU clock cycles}_A}{\text{Clock rate}_B}$$

$$6 \text{ seconds} = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{\text{Clock rate}_B}$$

$$\text{Clock rate}_B = \frac{1.2 \times 20 \times 10^9 \text{ cycles}}{6 \text{ seconds}} = \frac{0.2 \times 20 \times 10^9 \text{ cycles}}{\text{second}} = \frac{4 \times 10^9 \text{ cycles}}{\text{second}} = 4 \text{ GHz}$$

To run the program in 6 seconds, B must have twice the clock rate of A.

Performance Assessment

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

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

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

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

innovate achieve lead

Million Instructions Per Second (MIPS) is the common measure of performance for a processor is the rate at which instructions are executed, expressed as MIPS or referred to as MIPS rate

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

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

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

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

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

- MIPS = $(40 \times 10^6) / (1.55 \times 10^6)$
- = 25.8

Solution -1

- To calculate Execution Time
- Execution Time = Instruction Count x CPI x Cycle Time

  = I_c x CPI x (1/f)

  = (100000 x 1.55) /(40×10^6)

  = 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

Problem – 2…

a) Find the MIPS for both C1 and C2.

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 ?

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?

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…

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*10^6)

  = 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 ms

Solution-2…

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)

innovate achieve lead

- 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

innovate achieve lead

- 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:

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

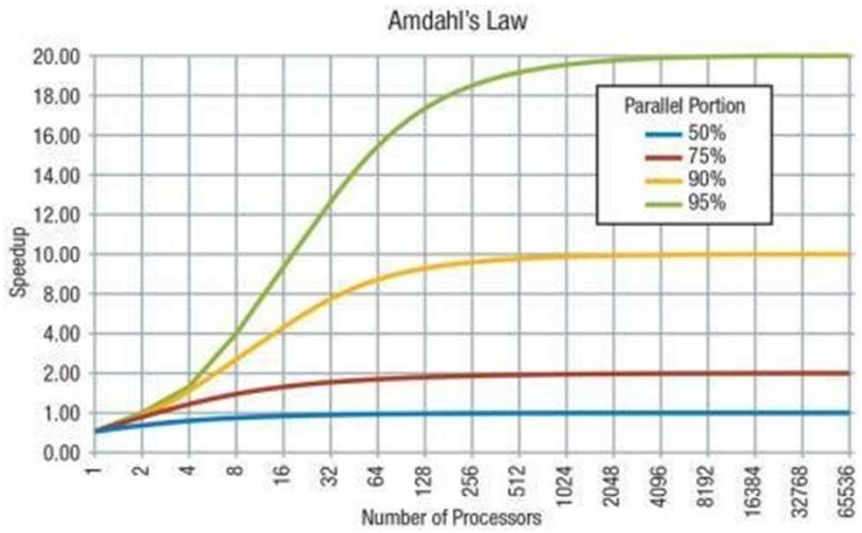
- As N tends to infinity, the maximum speedup tends to 1/(1-P)

Amdhal's Law

$$\text{Speedup} = \frac{\text{Time to execute program on a single processor}}{\text{Time to execute program on } N \text{ parallel processors}}$$

$$= \frac{T(1-P) + TP}{T(1-P) + TP/N}$$

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



- 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)

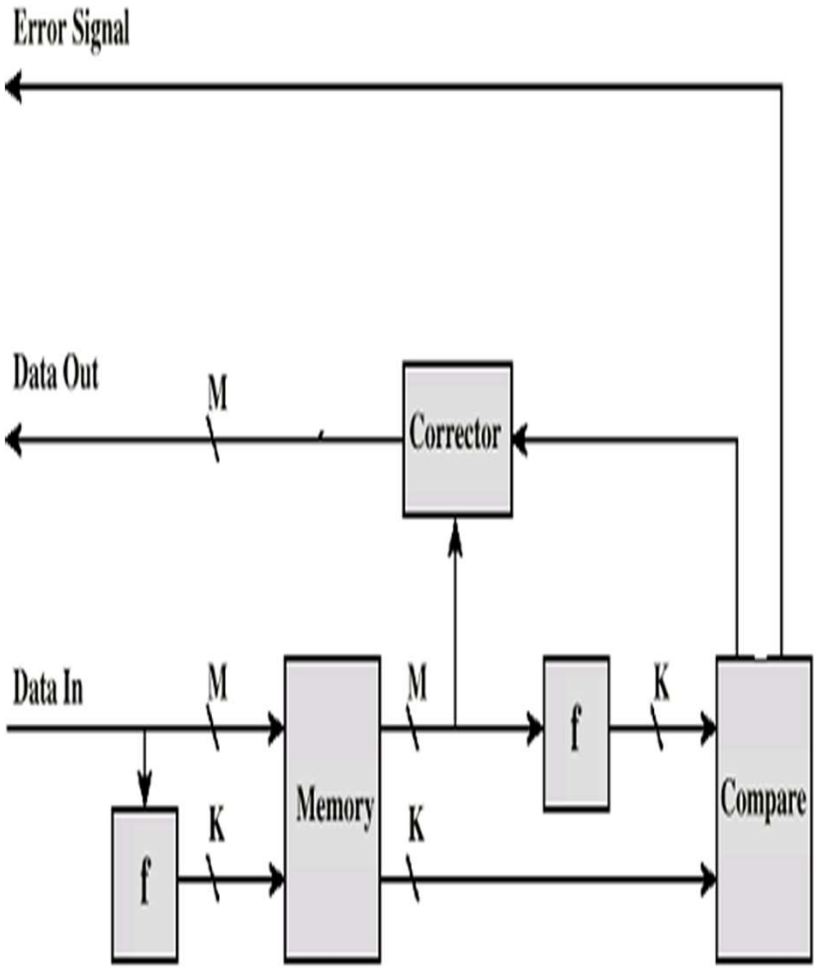
innovate achieve lead

- 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?

Hamming Code

BITS Pilani
Pilani Campus

Error Correcting Code Function



The comparison logic receives as input two K-bit values. A bit-by-bit comparison is done by taking the exclusive-OR of the two inputs. The result is called the *syndrome word*.

The comparison yields one of three results

- No errors are detected.
- An error is detected and it is possible to correct the error
- An error is detected but it is not possible to correct it.

Hamming Code…..

What should be the length of the code K

- length of the syndrome word is K bits,
- Length of Data is "M" bits
- Length of K should satisfy

$$2^k - 1 >= M + K$$

and K has a range between 0 and $2^k - 1$

Problem 1

Let us assume data transmitted is 1011, data corruption in Semiconductor memory has resulted in altering the content of a single memory cell. How do we detect and correct the error using Hamming Code?

Step 1 : Check the length of Check bits required to form Tx word

Data = 1011 (M=4)

$2^k-1 >= M+K$

Here M = 4. at least K value should be 3

Hence there are in total 7 bits for transmission

Problem 1

Step 2: How to calculate Check bits and their position

Now we know there are total 7 bits for transmission
Data bits – D1, D2, D3, D4
Check bits – C1, C2 and C3

Bit Position	7	6	5	4	3	2	1
Position Number	111	110	101	100	011	010	001
Data Bit	D4	D3	D2		D1		
Check Bit				C3		C2	C1

Problem 1…

Step 2: How to calculate Check bits

Bit Position	7	6	5	4	3	2	1
Position Number	111	110	101	100	011	010	001
Data Bit	D4	D3	D2		D1		
Check Bit				C3		C2	C1

- $C1 = D1 \oplus D2 \oplus D4$
- $C2 = D1 \oplus D3 \oplus D4$
- $C3 = D2 \oplus D3 \oplus D4$

Problem 1

Step 2: How to calculate Check bits

				2^2		2^1	2^0
Bit Position	7	6	5	4	3	2	1
Position Number	111	110	101	100	011	010	001
Bits	D4	D3	D2	C3	D1	C2	C1
Data Bits	1	0	1	0	1	0	1

- $C1 = D1 \oplus D2 \oplus D4$
- $C2 = D1 \oplus D3 \oplus D4$
- $C3 = D2 \oplus D3 \oplus D4$

- $C1 = 1 \oplus 1 \oplus 1 = 1$
- $C2 = 1 \oplus 0 \oplus 1 = 0$
- $C3 = 1 \oplus 0 \oplus 1 = 0$

Therefore the date transmitted is 7 bit data : 1010101

Problem 1 – Data bit corrupted

Step 3: Let us assume the corrupted data received is 1111. Calculate Check bits for the received word

				2^2		2^1	2^0
Bit Position	7	6	5	4	3	2	1
Position Number	111	110	101	100	011	010	001
Bits	D4	D3	D2	C3	D1	C2	C1
Data Bits	1	1	1	0	1	0	1

- $C1 = D1 \oplus D2 \oplus D4$
- $C2 = D1 \oplus D3 \oplus D4$
- $C3 = D2 \oplus D3 \oplus D4$

Problem 1 – Data bit corrupted

				2^2		2^1	2^0
Bit Position	7	6	5	4	3	2	1
Position Number	111	110	101	100	011	010	001
Bits	D4	D3	D2	C3	D1	C2	C1
Data Bits	1	0	1	0	1	0	1

- $C1 = D1 \oplus D2 \oplus D4$
- $C2 = D1 \oplus D3 \oplus D4$
- $C3 = D2 \oplus D3 \oplus D4$

- *Calculated Check bit*
- $C1 = 1 \oplus 1 \oplus 1 = 1$
- $C2 = 1 \oplus 1 \oplus 1 = 1$
- $C3 = 1 \oplus 1 \oplus 1 = 1$

Step 4: Compare Check bits to generate Syndrome

Previous checkbits: 001
Calculated check bits : $\oplus$ 111
Syndrome : 110

Toggle the $(110)_2$ or 6^{th} position bit to get the correct data.
Here Corrected data is 1010111. Extracting Data it yields 1011

Problem 1 – Check bit is corrupted

Step 3: Let us assume the data received is same 1011.
Calculate Check bits for the received word

				2^2		2^1	2^0
Bit Position	7	6	5	4	3	2	1
Position Number	111	110	101	100	011	010	001
Bits	D4	D3	D2	C3	D1	C2	C1
Data Bits	1	0	1	0	1	0	0

- $C1 = D1 \oplus D2 \oplus D4$
- $C2 = D1 \oplus D3 \oplus D4$
- $C3 = D2 \oplus D3 \oplus D4$

- $C1 = 1 \oplus 1 \oplus 1 = 1\ (0)$
- $C2 = 1 \oplus 0 \oplus 1 = 0$
- $C3 = 1 \oplus 0 \oplus 1 = 0$

Step 4: Compare Check bits to generate Syndrome

Computed checkbits: 001
Received check bits : $\oplus$ 000
Syndrome : 001

Change in Single bit indicates error in one of the check bits. Only error signal is generated.

Problem 1. Conclusion…

- If the syndrome contains all 0s, no error has been detected.
- If the syndrome contains one and only one bit set to 1, then an error has occurred in one of the 3 check bits. No correction is needed.
- If the syndrome contains more than one bit set to 1, then the numerical value of the syndrome indicates the position of the data bit in error. This data bit is inverted for correction.

Problem 2 - Homework

innovate achieve lead

Consider a 12 bit (Data + Check bit) Code 111101011101

Find out if there is an error. If so which bit is having error?

Layout of Data and Check bits

Bit Position	12	11	10	9	8	7	6	5	4	3	2	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data Bit	D8	D7	D6	D5		D4	D3	D2		D1		
Check Bit					C4				C3		C2	C1

- $C1 = D1 \oplus D2 \oplus D4 \oplus D5 \oplus D7$
- $C2 = D1 \oplus D3 \oplus D4 \oplus D6 \oplus D7$
- $C3 = D2 \oplus D3 \oplus D4 \oplus D8$
- $C4 = D5 \oplus D6 \oplus D7 \oplus D8$

$\oplus$: Bitwise XOR operation

CPU-OS Simulator

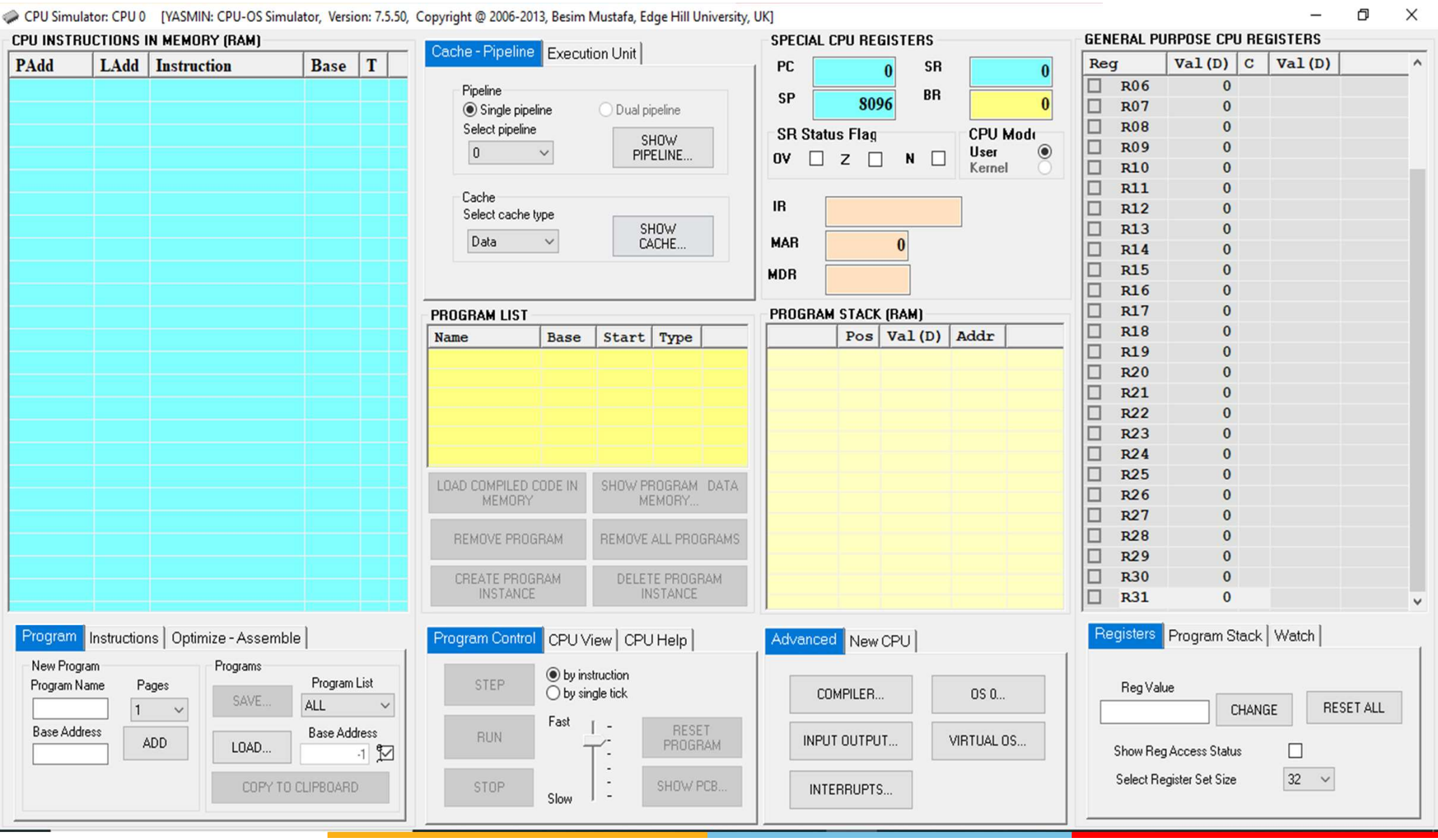
BITS Pilani

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(j) = 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

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.