CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Basics of COA

Lecture No.- 05

By- Vishvadeep Gothi sir

Recap of Previous Lecture

AR DR

Topic CPU Registers

Topic Memory Addressing

Topic Memory Access

Topics to be Covered

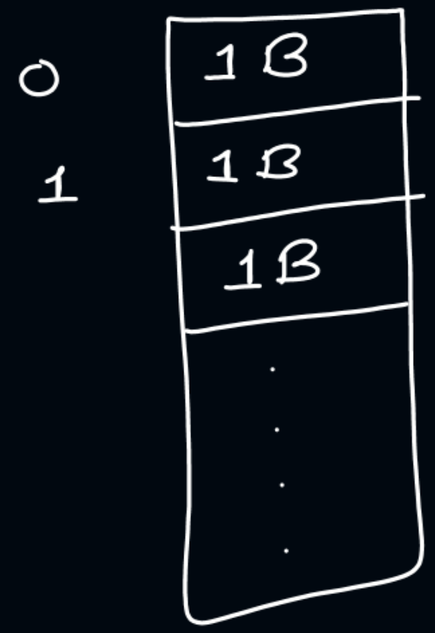
Topic Architecture Type (Based on Size of Input)

Topic Micro Operation

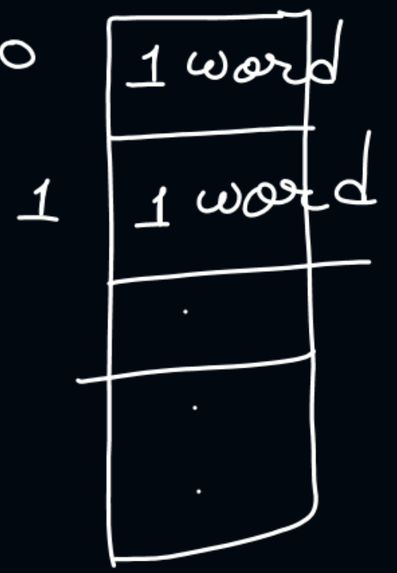
Topic Memory Access

(default)

Byte addressable mem.



word addressable mem.



Topic : Storing Content in Memory

Assume an instruction, size = 4 bytes | starting add. = 500

ex. 1 :-

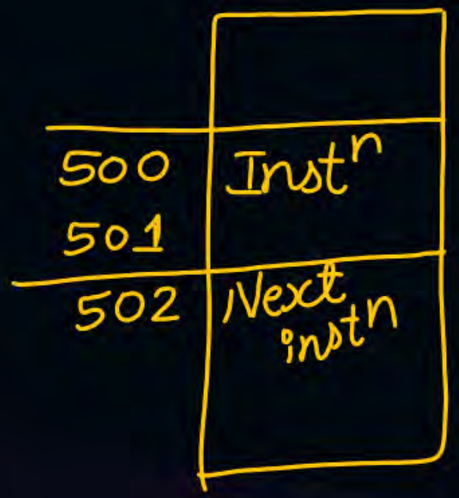
word addressable memory

1 word = 4 Bytes

ex :- 2

word addressable

1word = 2 bytes



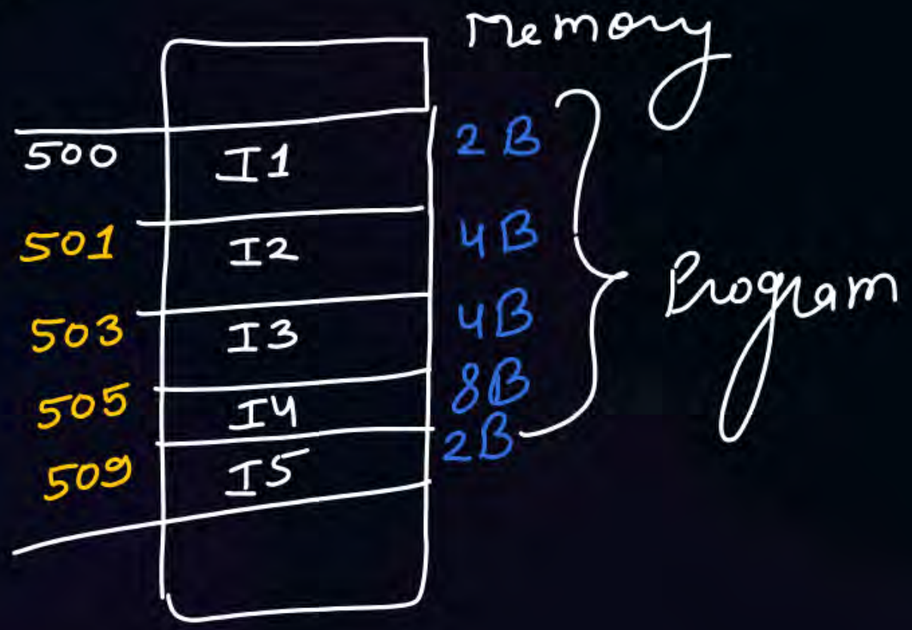
ex :- 3

byte addressable

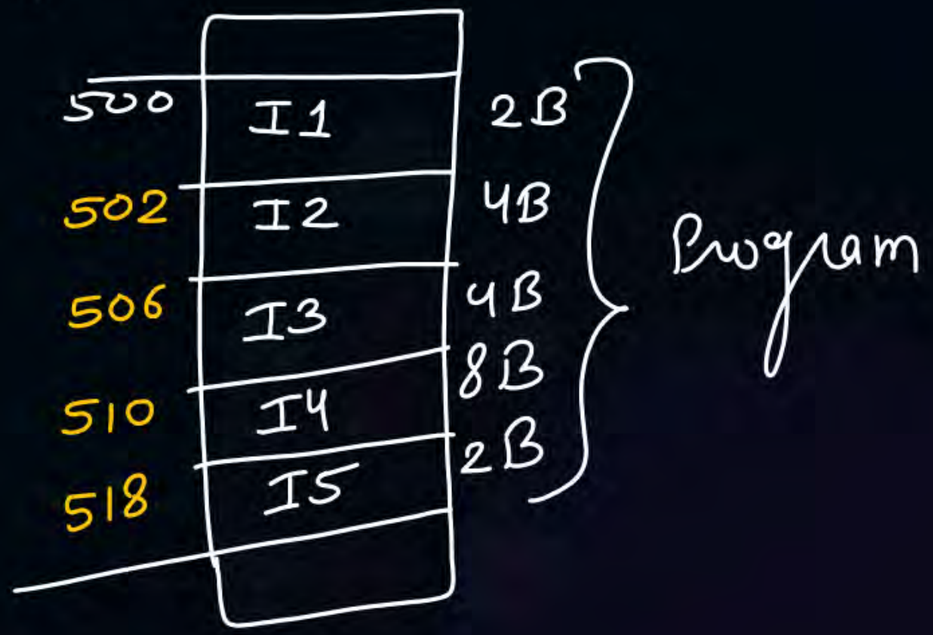
Topic : Storing Content in Memory

1 word = 2 Bytes

Memory ⇒ word addressable



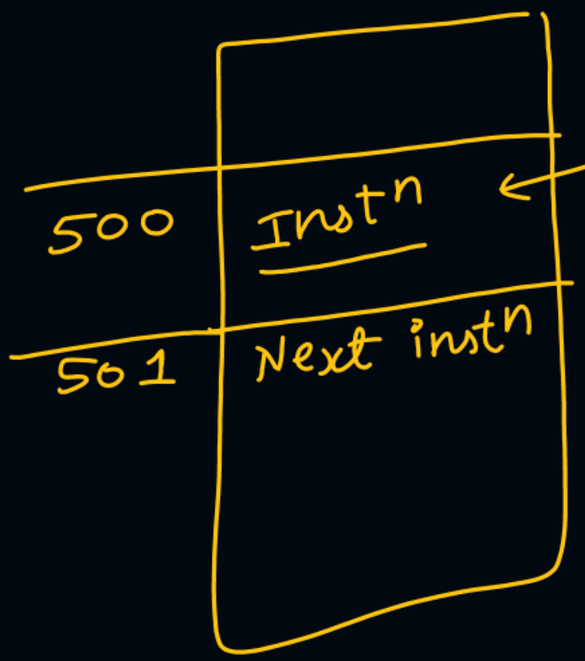
Byte addressable



Instⁿ size = 1 Byte.

memory word addressable.

1 word = 2 B



space wastage of 1 Byte

[NAT]

#Q. A CPU has 4 bytes instructions. A program (Instructions I_1 to I_{200}) starts at address 200 (in decimal). Find the address of following instructions:

1. I_1 ⇒ 200
2. I_5 ⇒ 216
3. I_{120} ⇒ 676

⇓

200 + 4 * 119

= 676

Memory (default ⇒ Byte addressable)

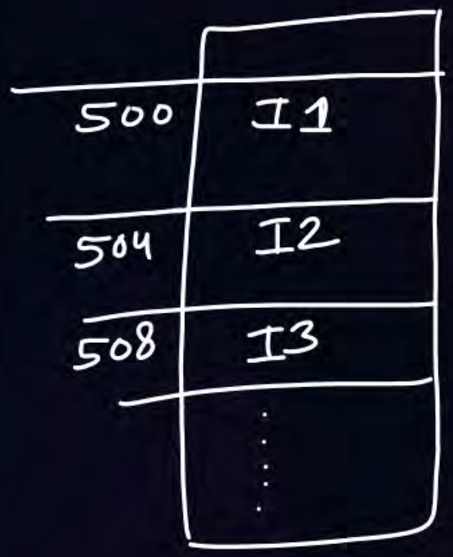
4. I_{193} ⇒ 968

200 + 192 * 4

= 968

[NAT]

#Q. A CPU has 4 bytes instructions. A program (Instructions I_1 to I_{200}) starts at address 500 (in decimal). What should be the PC value when instruction I_6 will be executing in CPU?



when I6 is executing then PC should point Instⁿ I7.

starting add. of I7 = 500 + 6 * 4

= 524 Ans.

[NAT]

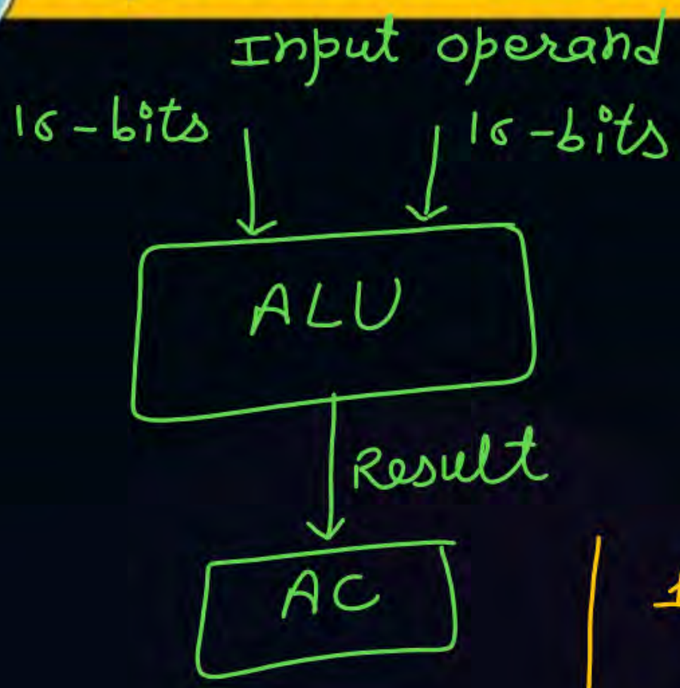
#Q. A CPU has 4 bytes instructions. A program (Instructions I_1 to I_{200}) starts at address 500 (in decimal). What should be the PC value when instruction *i* will be executing in CPU?

Ans:- PC will point to $(i+1)$

add. of i^{th} inst^n $i+1 = 500 + 4 * (i+1-1)$

$= 500 + 4 * i$

Topic : Architecture Type (Based on Size of Input)

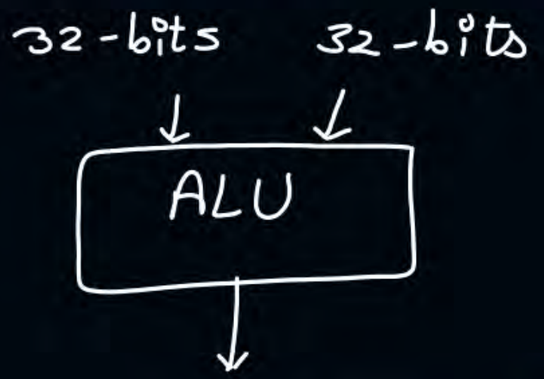


1 word = 16-bits = 2 Bytes

16-bit architecture

each input in ALU can be of max. 16-bits

32-bit architecture :-



1 word = 32 bits = 4 bytes

Topic : Micro Operation

- The operations executed on values stored in registers
- Symbolic Notation to describe the micro-ops: Register Transfer Language (RTL)

⇒ In single step if needed, multiple micro-operations can be performed if all of those are mutually exclusive

↓

not using any common component

Topic : Micro Operation

- Register Transfer: Content of Register R1 copied to Registers R5
  $R5 \leftarrow R1$
- Comma: $R6 \leftarrow R1$, $PC \leftarrow PC+1$ ⇐ both can be performed parallelly
- Memory Transfer:

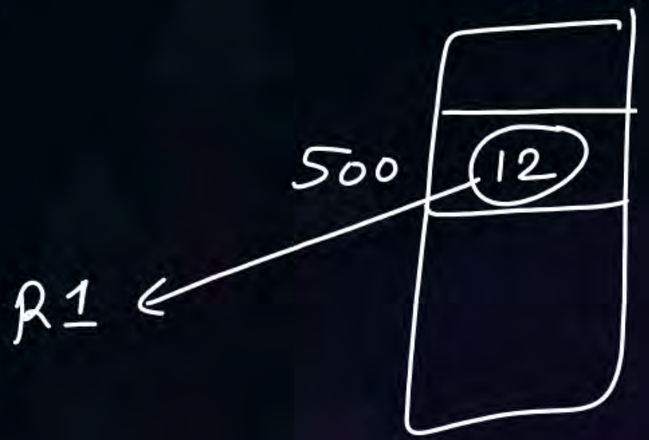
Read :- CPU ← memory from one address

$R1 \leftarrow M[address]$

$R1 \leftarrow M[500]$

$R3 \leftarrow \#500$

$R1 \leftarrow M[R3]$



write :-

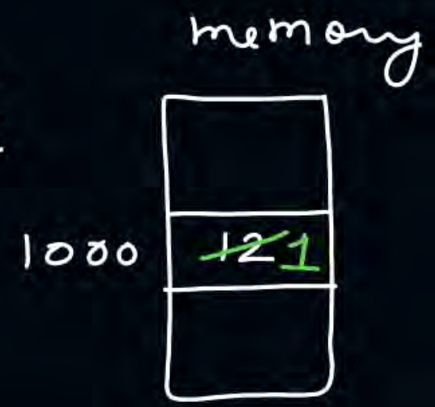
M[address] ← Reg

exc:-

Given

R1 = ~~5~~ ~~13~~ 1

R2 = ~~8~~ 12



R1 ← R1 + R2

R2 ← M[1000]

R1 ← R1 − R2

M[1000] ← R1

[NAT]

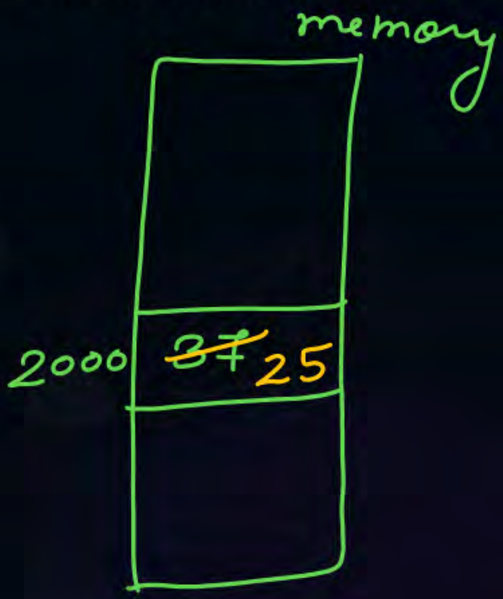
#Q. Consider the following program segment. Here R1 and R2 are the general purpose register. Assume that the content of memory location 2000 is 37. All numbers are in decimal. After the execution of this program the value of memory location 2000 is?

Instructions	Operations
MOV R1, #12	R1 ← #12
MOV R2, (2000)	R2 ← M[2000]
SUB R2, R1	R2 ← R2 – R1
MOV (2000), R2	M[2000] ← R2
HALT	Stop

R1 = 12

R2 = ~~37~~ 25

Ans = 25



[NAT]

#Q. Consider the following program segment. Here R1 and R2 are the general purpose register. Assume that the content of memory location 3000 is 13. All numbers are in decimal. After the execution of this program the value of memory location 3000 is?

Instructions	Operations
MOV R1, #7	R1 ← #7
MOV R2, (3000)	R2 ← M[3000]
ADD R2, R1	R2 ← R2 + R1
ADD R1, R2	R1 ← R1 + R2
MOV (3000), R1	M[3000] ← R1
HALT	Stop

R1 = ~~7~~ 27

R2 = ~~13~~ 20

Ans = 27

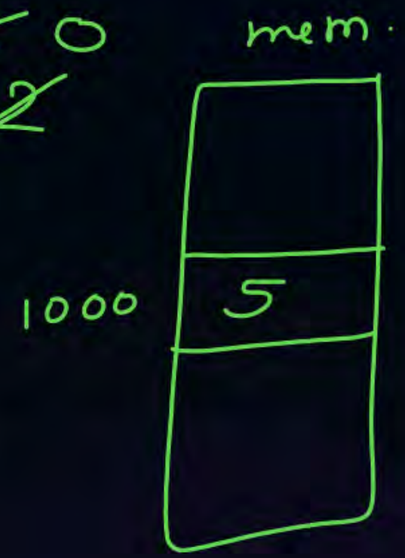
[NAT]

#Q. Consider the following program segment. Here R1 and R2 are the general-purpose register. Assume that the content of memory location 1000 is 5. All the numbers are in decimal.

	Instructions	Operations
	MOV R1, (1000)	R1 ← M[1000]
	MOV R2, #8	R2 ← #8
LOOP:	ADD R2, R1	R2 ← R2 + R1
	DEC R1	R1 ← R1 – 1
	BNZ LOOP	Branch on not zero
	HALT	Stop

R1 = ~~5~~ 4 3 2 1 0

R2 = ~~8~~ ~~13~~ ~~17~~ ~~20~~ ~~22~~ 23



Ans = 23

The value of R2 at the end of program execution is?

[MCQ]

#Q. Consider the following program segment. Here R1, R2 and R3 are the general-purpose registers.

	Instruction	Operation	Instruction Size (no. of words)
	MOV R1, (3000)	R1 ← M[3000]	2
LOOP:	MOV R2, (R3)	R2 ← M[R3]	1
	ADD R2,R1	R2 ← R1 + R2	1
	MOV (R3),R2	M [R3] ← R2	1
	INC R3	R3 ← R3 + 1	1
	DEC R1	R1 ← R1 – 1	1
	BNZ LOOP	Branch on not zero	2
	HALT	Stop	1

Assume that the content of memory location 3000 is 10 and the content of the register R3 is 2000. The content of each of the memory locations from 2000 to 2010 is 100. The program is loaded from the memory location 1000. All the numbers are in decimal.

Assume that the memory is word addressable. The number of memory reference for accessing the data in executing the program completely is

A 10

B 11

C 20

D 21

[MCQ]

#Q. Consider the following program segment. Here R1, R2 and R3 are the general purpose registers.

	Instruction	Operation	Instruction Size (no. of words)
	MOV R1, (3000	R1 ← M[3000]	2
LOOP:	MOV R2, (R3)	R2 ← M[R3]	1
	ADD R2,R1	R2 ← R1 + R2	1
	MOV (R3),R2	M [R3] ← R2	1
	INC R3	R3 ← R3 + 1	1
	DEC R1	R1 ← R1 – 1	1
	BNZ LOOP	Branch on not zero	2
	HALT	Stop	1

Assume that the content of memory location 3000 is 10 and the content of the register R3 is 2000. The content of each of the memory locations from 2000 to 2010 is 100. The program is loaded from the memory location 1000. All the numbers are in decimal.

Assume that the memory is word addressable. After the execution of this program, the content of memory location 2010 is:

A 100

B 101

C 102

D 110

[MCQ]

#Q. Consider the following program segment. Here R1, R2 and R3 are the general-purpose registers.

	Instruction	Operation	Instruction Size (no. of words)
	MOV R1, (3000)	R1 ← M[3000]	2
LOOP:	MOV R2, (R3)	R2 ← M[R3]	1
	ADD R2,R1	R2 ← R1 + R2	1
	MOV (R3),R2	M [R3] ← R2	1
	INC R3	R3 ← R3 + 1	1
	DEC R1	R1 ← R1 – 1	1
	BNZ LOOP	Branch on not zero	2
	HALT	Stop	1

Assume that the content of memory location 3000 is 10 and the content of the register R3 is 2000. The content of each of the memory locations from 2000 to 2010 is 100. The program is loaded from the memory location 1000. All the numbers are in decimal.

Assume that the memory is byte addressable and the word size is 32 bits. If an interrupt occurs during the execution of the instruction "INC R3", what return address will be pushed on the stack?

A 1005

B 1020

C 1024

D 1040

[NAT]

#Q. Consider the following instruction sequence where registers R1, R2 and R3 are general purpose and MEMORY [X] denotes the content at the memory location X.

Instruction	Semantics	Instruction Size (bytes)
MOV R1, (5000)	R1 ← MEMORY[5000]	4
MOV R2, (R3)	R2 ← MEMORY[R3	4
ADD R2, R1	R2 ← R1 + R2	2
MOV (R3), R2	MEMORY[R3] ← R2	4
INC R3	R3 ← R3 + 1	2
DEC R1	R1 ← R1 – 1	2
BNZ 1004	Branch if not zero to the given absolute address	2
HALT	Stop	1

Assume that the content of the memory location 5000 is 10, and the content of the register R3 is 3000. The content of each of the memory locations from 3000 to 3010 is 50. The instruction sequence starts from the memory location 1000. All the numbers are in decimal format. Assume that the memory is byte addressable.

After the execution of the program, the content of memory location 3010 is_____.

2 mins Summary

Topic Architecture Type (Based on Size of Input)

Topic Micro Operation

Topic Memory Access

Happy Learning

THANK - YOU