CS & IT ENGINEERING

COMPUTER ORGANIZATION AND ARCHITECTURE

Instruction & Addressing Modes

Lecture No.- 02

By- Vishvadeep Gothi sir

Recap of Previous Lecture

Topic Instructions

Topic Opcode

Topics to be Covered

Topic Multiple Instructions Support

Topic Variable Length Instructions

[NAT]

Ans = 4

#Q. Consider a computer which supports only 2-address and 1-address instructions. Each instruction is of 6-bits and each address is of 2-bits. If there are 3 2-address instructions supported by the system then maximum number of 1-address instructions supported by system is?

2-add.

6-bits

opcode	a.1	a.2
2	2	2

max opcodes = $2^2 = 4$ (00, 01, 10, 11)

used " = 3 (assume: 00, 01, 10)

unused = 1 (11)

1-add.

6-bits

opcode	a.1
4	2

$\underbrace{\quad}_{2}\underbrace{\quad}_{2}$

1100
1101
1110
1111

max 1-add. instrns = 4

if any inst^{ns} comes to CPU for execution

$\boxed{10\,11\,10}$

↓

should not be used as 2 - add. inst^{ns}

⇓

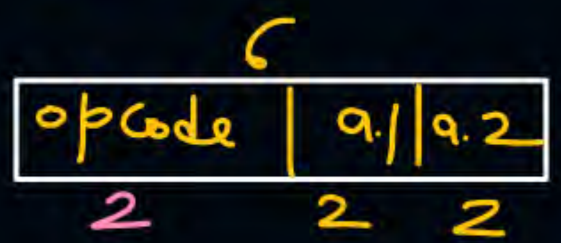
then only treated as 1 - add. inst^{n}

[NAT]

#Q. Consider a computer which supports only 2-address and 1-address instructions. Each instruction is of 6-bits and each address is of 2-bits. If there are 3 2-address instructions supported by the system then maximum number of 1-address instructions supported by system is?

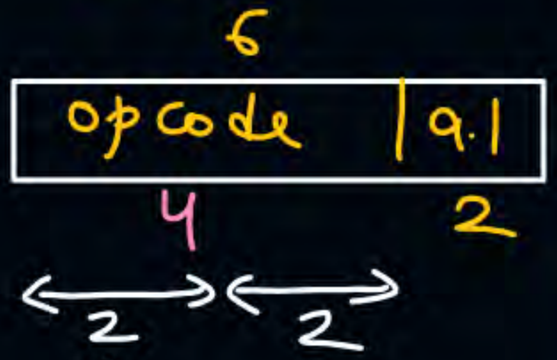
In above instruction what is the range of number of 1-address instructions supported? 1 to 4

2-add.



$$\text{max} = 2^2 = 4$$
$$\text{used} = 3$$
$$\text{unused} = 1$$

1-add.



$$1 * 2^2 = \underline{\underline{4}}$$

Ans.

2-add. inst^{ns}	unused opcodes	max 1-add. inst^{ns}
4	0	$0 * 2^2 = 0$
3	1	$1 * 2^2 = 4$
2	2	$2 * 2^2 = 8$
1	3	$3 * 2^2 = 12$
0	4	$4 * 2^2 = 16$

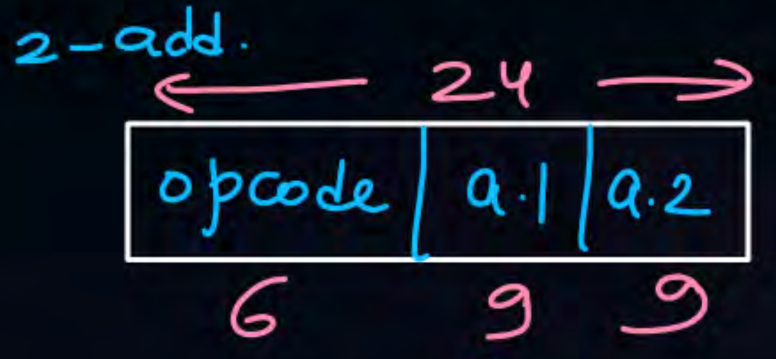
→ only 2-add. type inst^n supported

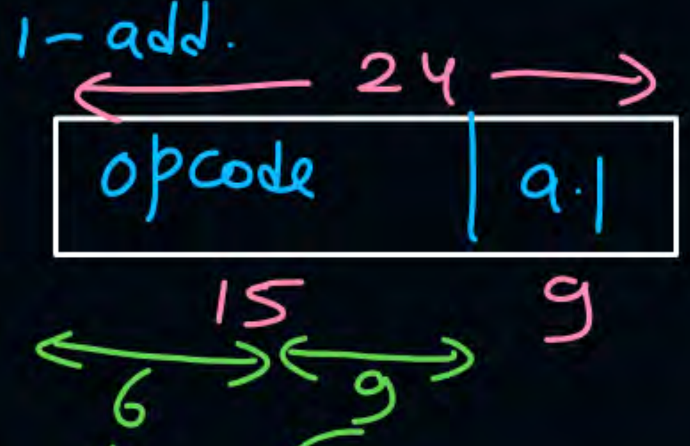
→ only 1-add. type inst^n supported

[NAT]

Ans = 2560

#Q. Consider a system with 24-bit instructions and 9-bit addresses. If there are 59 2-address instructions then maximum how many 1-address instructions can be formulated in the system?





$$\text{max} = 2^6 = 64$$

$$\text{used} = 59$$

$$\text{unused} = 5$$

$$5 * 2^9 = 2560$$

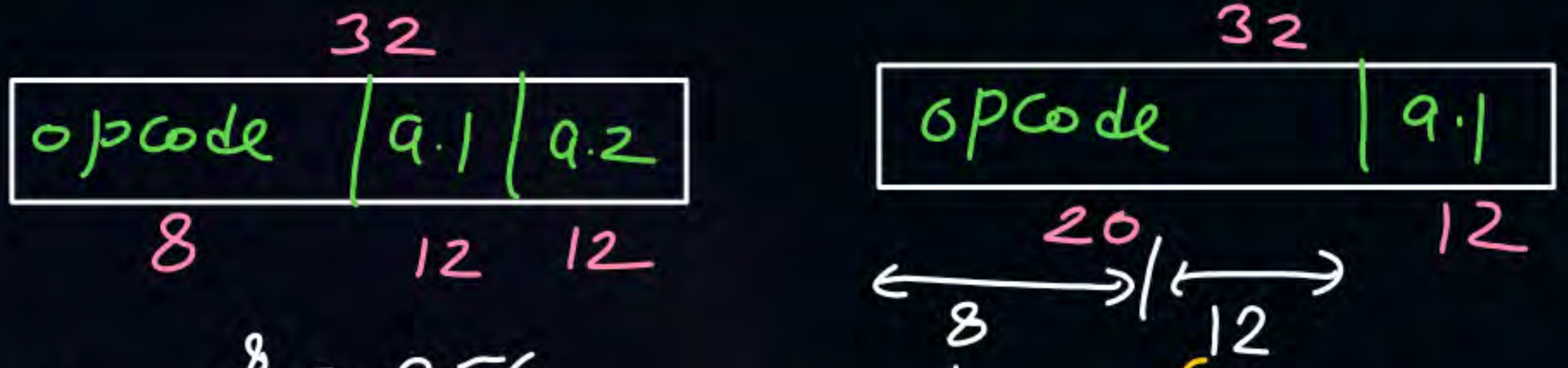
[NAT]

Ans = $\underline{\underline{8192}}$

#Q. Consider a system with 32-bit instructions and 12-bit addresses. If there are 254 2-address instructions then maximum how many 1-address instructions can be formulated in the system?

2-add.

1-add.



$$\text{max} = 2^8 = 256$$
$$\text{used} = 254$$
$$\text{unused} = 2$$

$$2 * 2^{12} = 2^{13} = 8192$$

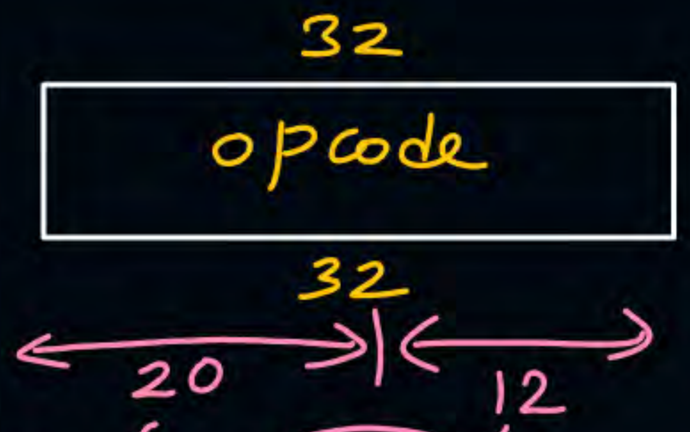
[NAT]

#Q. Consider a system with 32-bit instructions and 12-bit addresses. If there are 254 2-address instructions and 8150 1-address instructions then maximum how many 0-address instructions can be formulated?

max 1 add. inst^{ns} = 8192

used = 8150

unused = 42



max = $42 * 2^{12}$ = $\underline{\underline{172032}}$ Ans.

[NAT]

$$\text{Ans} = (64 - x) * 2^{8}$$

#Q. Consider a system which supports 3-address and 2-address instructions both. It has 30-bit instructions with 8-bit addresses. If there are ‘x’ 3-address instructions then maximum how many 2-address instructions can be formulated?

3-add.

30

opcode	a.1	a.2	a.3
6	8	8	8

2-add.

30

opcode	a.1	a.2
14	8	8

$$14 = \underbrace{\quad}_{6} \mid \underbrace{\quad}_{8}$$

$$\begin{aligned} \text{max} &= 2^{6} = 64 \\ \text{used} &= x \\ \hline \text{unused} &= 64 - x \end{aligned}$$

$$(64 - x) * 2^{8}$$

Ques) In prev. Quest^n if max 2-add inst^{ns} are 2048, then $x =$ <u>56</u> ?

Sol^n

$$(64 - x)\, 2^8 = 2048$$

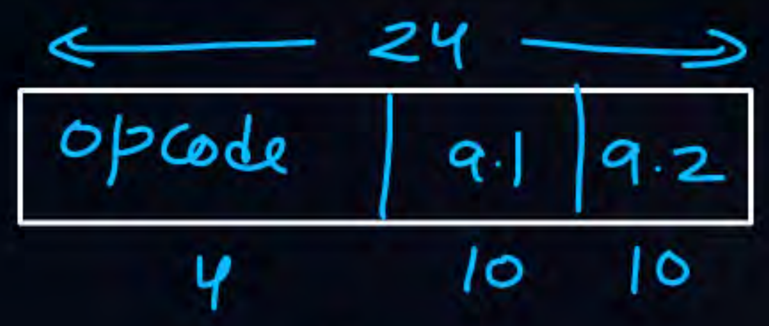
$$64 - x = 8$$

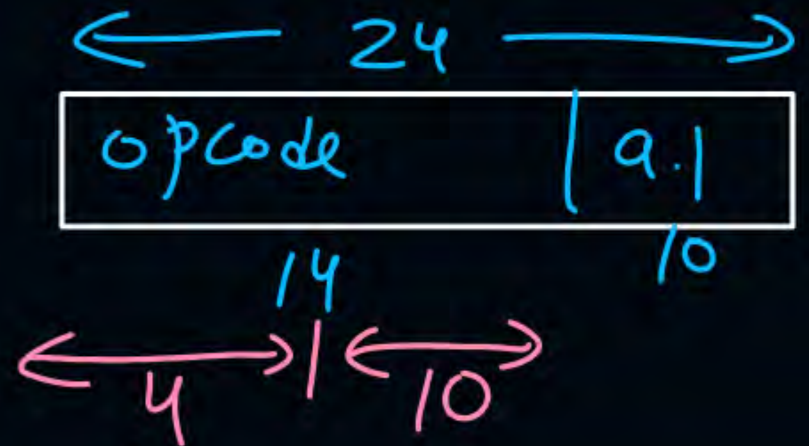
$$\boxed{x = 56}$$

[NAT]

Ans = 12

#Q. Consider a system which supports 2-address and 1-address instructions both. It has 24-bit instructions with 10-bit addresses. If there are 4096 1-address instructions then maximum how many 2-address instructions can be formulated?





$$\begin{aligned} max &= 2^4 = 16 \\ used &= x \\ \hline unused &= 16 - x \end{aligned}$$

$$(16-x) * 2^{10} = 4096$$

$$16 - x = 4$$

$$\boxed{x = 12}$$

[NAT]

→ Reg = 6 bits

#Q. Consider a system with 16-bits instructions and 64 CPU registers. The System supported 2 types of instructions: Type-A and Type-B.

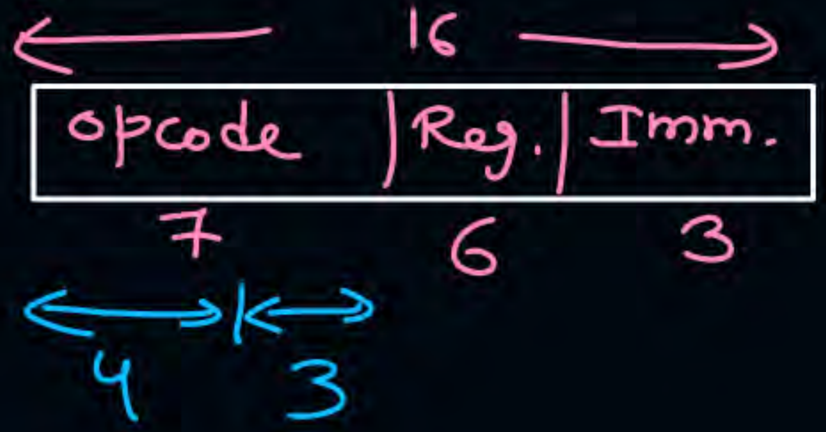
Type-A instructions have an opcode, one register operand and one immediate operand of 3-bits

Type-B instructions have an opcode, and 2 register operands.

If there are 10 Type-B instructions supported by the system then maximum how many Type-A Instructions supported by the system?

Ans = 48

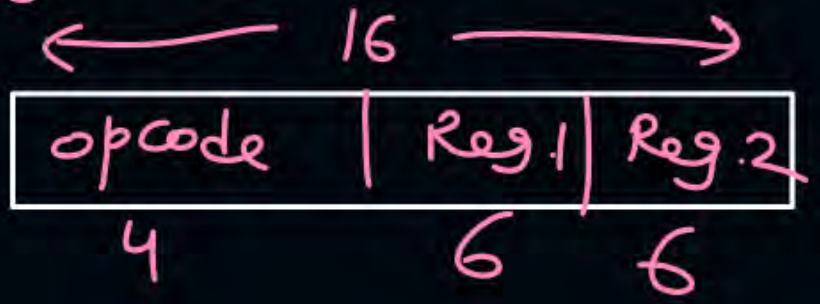
Type - A



↓

$$max = 6 * 2^3 = \underline{\underline{48}}$$ Ans.

Type - B



⇓

$$max = 2^4 = 16$$

$$used = 10$$

$$unused = 6$$

[NAT] GATE-2020

$\boxed{\text{Ans} = 14}$

64 → Reg = 6 bits

#Q. A processor has 64 registers and uses 16-bit instruction format. It has two types of instructions: I-type and R-type. Each I-type instruction contains an opcode, a register name, and a 4-bit immediate value. Each R-type instruction contains an opcode and two register names. If there are 8 distinct I-type opcodes, then the maximum number of distinct R-type opcodes is _____?

I-type

⟵ 16 ⟶

opcode	Reg.	I.V.
6	6	4

6: ⟷ 4 ⟶ 1 ⟷ 2

↓

$(16-x)*2^2 = 8$

$16 - x = 2$

$\boxed{x = 14}$

R-type

⟵ 16 ⟶

opcode	Reg.1	Reg.2
4	6	6

$\text{max} = 2^4 = 16$

$\text{used} = x$

$\text{unused} = 16 - x$

[NAT] GATE-2018

H. W.

#Q. A processor has 16 integer registers (R0, R1, … , R15) and 64 floating point registers (F0, F1, … , F63). It uses a 2-byte instruction format. There are four categories of instructions: Type-1, Type-2, Type-3, and Type 4. Type-1 category consists of four instructions, each with 3 integer register operands (3Rs). Type-2 category consists of eight instructions, each with 2 floating point register operands (2Fs). Type-3 category consists of fourteen instructions, each with one integer register operand and one floating point register operand (1R+1F). Type-4 category consists of N instructions, each with a floating-point register operand (1F).

The maximum value of N is _____ ?

Instruction length

fixed

⇓

opcode variable length

(expanding opcode)

variable

⇓

fixed length opcode

[NAT]

#Q. Consider there are 3 types of instructions in system:

1. Register Operand instructions: One opcode and 2 registers
2. Memory Operand instructions: One opcode, 1 register and 1 memory address
3. Immediate Operand Instructions: One opcode, 1 register and 1 immediate operand

Number of registers = 64 ⇒ Reg. Number = 6 bits

Number of bits in immediate operand = 10-bits

Memory size = 512Mbytes (byte addressable) ⇒ mem. add. = 29 bits

Total Instructions:

1. Reg Operand type: 10 → Total = 26 ⇒ opcode = 5 bits

2. Memory Operand type : 12

3. immediate Operand type : 4

Maximum and Minimum instruction length are? $\boxed{40 \,,\, 17 \text{ bits}}$ Ans.

Reg. operand

opcode	Reg.1	Reg 2
5	6	6

17 bits

Imm. operand

opcode	Reg.	Imm. operand
5	6	10

21 bits

mem. operand

OpCode	Reg.1	mem. add.
5	6	29

40 bits

[NAT]

#Q. In a simplified computer the instructions are:

Reg- mem. based arch.

Reg. or mem. Reg.

ALU

OP R_i , R_j	- Performs R_i Op R_j and stores the result in R_j
OP m, R_i	- Performs val Op R_i and stores the result in R_i val denotes the content of memory location m
MOV m, R_i	- Moves the content of memory location m to register R_i
MOV R_i , m	- Moves the content of register R_i to memory location m

The computer has only two registers and OP is either ADD or SUB. Consider the following basic block:

R1 $t1 = a + b$

$t2 = c + d$

$t3 = e - t2$

R2 $t4 = t1 - t3$

[NAT]

#Q. Assume that all operands are initially in memory. The final value of the computation should be in memory. What is the minimum number of MOV instructions in the code generated for this basic block?

```
✓mov  b, R1       R1 ← b
 ADD  a, R1       R1 ← a + R1
✓mov  d, R2       R2 ← d
 ADD  C, R2       R2 ← C + R2
 SUB  e, R2       R2 ← e − R2
 SUB  R1, R2      R2 ← R1 − R2
✓mov  R2, x       x ← R2
```

Ans = 3

[NAT]

Ans = 5

#Q. In a simplified computer the instructions are:

OP R_i , R_j	- Performs R_i Op R_j and stores the result in R_i
OP R_i, m	- Performs R_i Op val and stores the result in R_i val denotes the content of memory location m
MOV m , R_i	- Moves the content of memory location m to register R_i
MOV R_i , m	- Moves the content of register R_i to memory location m

The computer has only two registers and OP is either ADD or SUB. Consider the following basic block:

R1 t1 = a + b

t2 = c + d

R2 t3 = e – t2

t4 = t1 – t3

[NAT]

#Q. Assume that all operands are initially in memory. The final value of the computation should be in memory. What is the minimum number of MOV instructions in the code generated for this basic block?

```
✓ R1 ← a
  R1 ← R1 + b
✓ R2 ← c
  R2 ← R2 + d
✓ t ← R2
✓ R2 ← e
  R2 ← R2 − t
  R1 ← R1 − R2
✓ x ← R1
```

t ⇒ mem. location

Topic : Register Spill

If sufficient no. of registers are not available in CPU, and for a program an intermediate operand is moved out of register to memory for temporary basis.

[NAT] Ans = 2

Reg. ↓ ALU ↓ mem, or Reg.

#Q. Consider a register-memory architecture system (2-address instructions). For this system the following intermediate code is going to be converted in machine code. Minimum how many registers are required in system so that the code can run without register spill?

Consider first operand is always the register operand and it's the destination for operation too.

~~R1~~ t1 = X + Y

R1 t2 = t1 - Z

R2 t3 = t1 + t2

t4 = M + t3

Assume X, Y, Z and M are memory operands

$R1 \leftarrow X$

$R1 \leftarrow R1 + Y$

$R2 \leftarrow R1$

$R1 \leftarrow R1 - Z$

$R2 \leftarrow R2 + R1$

$R1 \leftarrow M$

$R1 \leftarrow R1 + R2$

[NAT]

Ans = 3

#Q. Consider a register-based architecture system (2-address instructions). For this system the following intermediate code is going to be converted in machine code. Minimum how many registers are required in system so that the code can run without register spill?

Consider first operand is always the register operand and it's the destination for operation too.

t1 = X + Y

R1 t2 = t1 - Z

R2 t3 = t1 + t2

t4 = M + t3

Assume X, Y, Z and M are memory operands

R1 ← X

R2 ← Y

R1 ← R1 + R2

R2 ← R1

R3 ← Z

R1 ← R1 − R3

R2 ← R2 + R1

R1 ← M

R1 ← R1 + R2

2 mins Summary

Topic Multiple Instructions Support

Topic Variable Length Instructions

sat -
sun
} ⇒ 11am - 1 PM

Happy Learning

THANK - YOU