Ques 4 — gate - 2016

Consider a processor with 64 registers and an instruction set of size 12. Each instruction has five distinct field, namely, opcode, two source register identifiers, one destination register ∧ must be stored in memory in a byte aligned fashion. If a program has 100 instruction, the amount of memory in (bytes) consumed by the program text is?

and a twelve bit immediate value each instruction

Sol: No of register = 64 → $\log_2 64$ = 6-bit (size)
Set of instruction size → 12 → $\log_2 12$ = 4bit
means number of instruction supported by the architecture
hence opcode size = 4bit

- each instruction has five distinct field

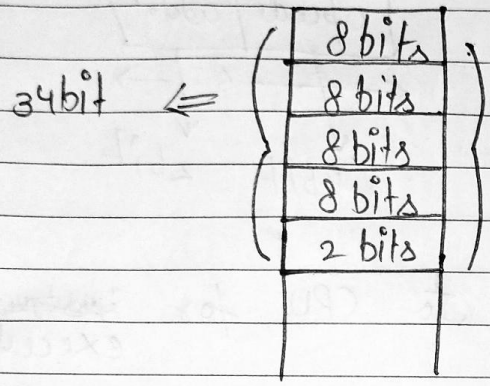
opcode	Reg1	Reg2	Reg3	immediet instruction
4bit	6bit	6bit	6bit	12 bit

- instruction length = 34 bit

- Each instruction must be stored ~~in~~ in memory in a "byte" aligned fashion

1 byte = 8 bit

if a program has 100 instruction, the amount of memory consumed by program text.



5 byte for 1 instruction
100 instruction = 5 × 100 = 500 bytes
Ans

---

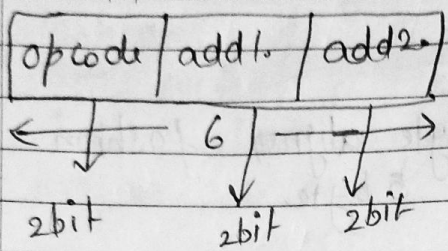
* Multiple type Instruction Lecture - 8

- Instruction length
  - → fixed length instruction => variable length opcode
  - → variable length instruction => fixed length opcode

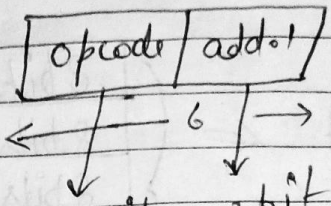
Ques: Consider a system which supports 2 add. and 1 add. both. The system has 6-bit instructions and 2 bit add. if there are three 2-add. instruction in the system, then maximum and minimum how many 1-add. instruction the system can support?

Sol: 2 add. instruction

format ⇒



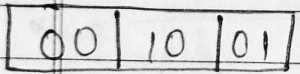
1-add. instruction



- An instruction comes to CPU for execution

011001

as 2 add. instruction



opcode Add.1 Add2.

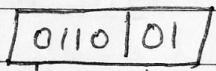
max opcodes ⇒ $2^2 = 4$
(00, 01, 10, 11)

used opcode = 3
assume: (00, 01, 11)

unused 1 opcode = (11)

min = 1 Ans

ans 1 add instruction



opcode add1.

max opcode = 2^4 = 16

1100
1101
1110
1111

max = 4 Ans

MY CHOICE

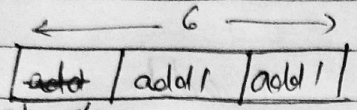
Date:

Page No.:

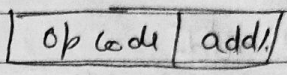
* Method to solve these type of questions

- Make format first
- Start from min bit of opcode

2 add.



1 add 1



max opcode = $2^3 = 4$

used " = 3

unused " = 1

divide this Opcode into 2 half, 1st half is == bit taken by previous instruction

2 remaning 2

1 × $2^2 = 4$

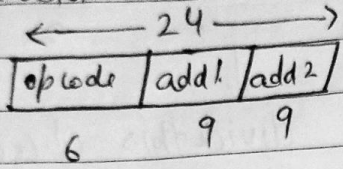
2 add. instru	unused opcode	1 add inst	
4	0	$0 * 2^2 = 0$	only 2 add. instruction supported
3	1	$1 * 2^2 = 4$	
2	2	$2 * 2^2 = 8$	
1	3	$3 \times 2^2 = 12$	
0	4	$4 * 2^2 = 16$	only 1 add. instruction supported

Combinations of opcode we can make

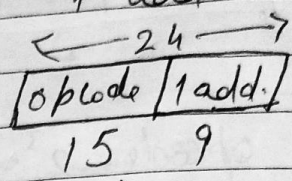
Qu 2 Consider a system with 24 bit instruction and 9 bit addresses if there are 60, 2 addresses instruction, then maximum how many instructions can be ~~found~~ formulated in the system

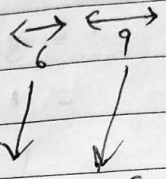
Sol

2 add.



1 add.





max opcodes
$2^6 = 64$

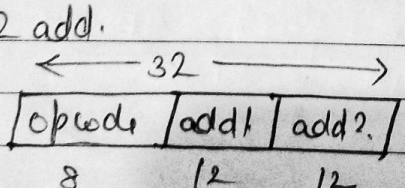
used opcodes = 60
unused = 4

$4 * 2^9 = 2048$ Ans

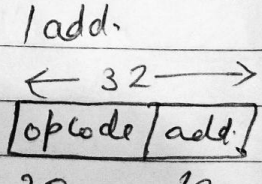
Qu 3 Consider a system with 32-bit instruction and 12-bit addresses. if ~~254~~ there are 254, 2 address instruction and 8000, 1-address instruction then minimum how many 0 address instruction can be formulated?

Sol

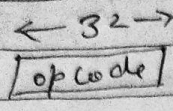
2 add.



1 add.



0 - add.



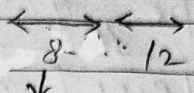
2 add.

max opcode = 2^8 = 256

used " = 254

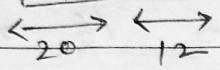
un used " = 2

1 add.



$2 * 2^{12} = 8192$

0 add. 32



192×2^{12} =

maximum 1 add. instruction = 8192

Used instruction = 8000

unused " = 192

maximum 0 add. instruction = 786, 432

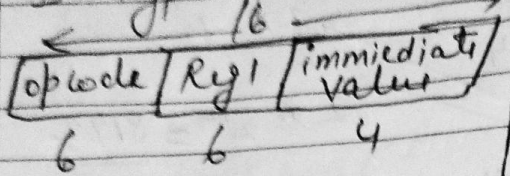
Qu 4 Gate 2020

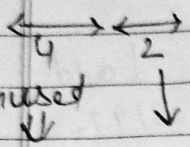
A processor has 64 registers and uses 16 bit instruction format. It has two types of instructions: I types and R types. Each I-type instruction contains an opcode, a register name, and 4-bit immediate value. Each R-type instruction contains an opcode and two register names. If there are 8 distinct I-type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

64 register → $\log_2 64$ → 6 bit size for reg

16 bit instruction

I type instruction





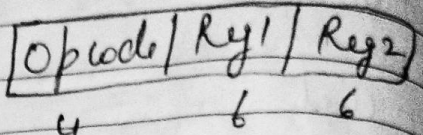
$(16-x) * 2^2 = 8$

$(16-x) \times 4 = 8$

$16 - x = \frac{8}{4} 2$

$14 = x$ Ans

R-type instruction



max opcodes = $2^4 = 16$

used " = x

unused " = $16 - x$

* How instructions are generated and executed? Lecture 9

- Compliler generates instructions based on CPU.

⇒ means only those type of instructions it will generate, what are are supported by CPU.