CS & IT ENGINEERING

Computer Organization Architecture

Instructions & Addressing Modes

DPP 01 Discussion Notes

By- Devvrat Tyagi sir

[NAT]

Ans (2)

#Q. Consider a system which support only 2 address instructions only, and supports word addressable memory. The memory has total capacity of 2MB with word size of 4 Bytes. The system supports 350 distinct instructions. How many memory locations are required to store each instruction in the memory?

Capacity of memory = 2 MB
$= 2^{21}$ B
$= 2^{19}$ 2^2 B

1 word = 4 B
= 32 bits

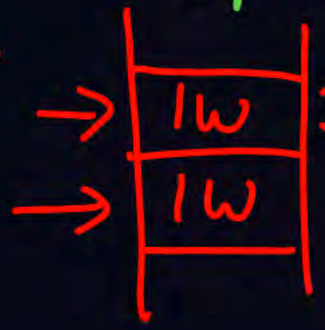
# of Inst^{n} = 350

Inst^{n} Size = 9 + 19 + 19
= 47 bits.

Opcode	Add1	Add2
9 bits	19 bits	19 bits

$= 2^{19}$ Words.

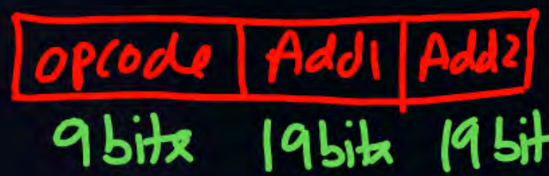
opcode = 9 bits = 2^9 = 512



(2)

[NAT]

#Q. Consider a system which support only 2 address instructions only, and supports word addressable memory. The memory has total capacity of 2MB with word size of 4 Bytes. The system supports 350 distinct instructions. If a program has 500 instructions, which is stored in the memory then the amount of memory required to store the entire program is 4000 bytes?



1 word = 4B

to store one instruction = 2 words.
= 2x4B
= 8B

total memory req = 500 x 8B
= 4000 B

[MCQ]

Ans [C]

#Q. The word addressable memory of a computer has 256K words of 32-bit each. The computer has an instruction format with four fields; an operation code field, a mode field to specify one of 8 addressing modes, a register address field to specify one of the 64 processor registers and a memory address field.

The bits for each field required in instruction format if the instruction is stored exactly in one word in memory?

A. Opcode: 5, Addressing mode: 3, Register: 6, Memory address: 20

B. Opcode: 3, Addressing mode: 3, Register: 6, Memory address: 20

C. Opcode: 5, Addressing mode: 3, Register: 6, Memory address: 18

D. Opcode: 3, Addressing mode: 3, Register: 6, Memory address: 18

# of words = 256 K

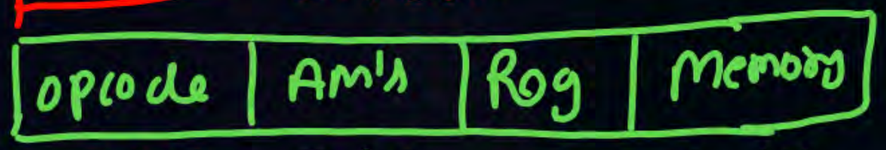
$= 2^{18}$

# of Am's = 8

bits in Am's field = 3

18 bits to represent memory loc.

Memory Add = 18 bits



# of Reg = 64

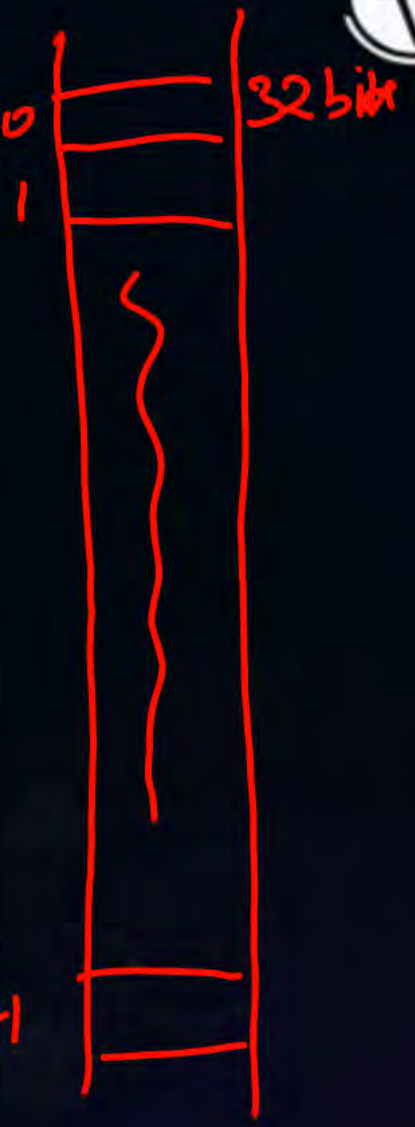
Reg. field = $\log_2 64$

$= \log_2 2^6$

$= 6$ bits

Each word size = 32 bits

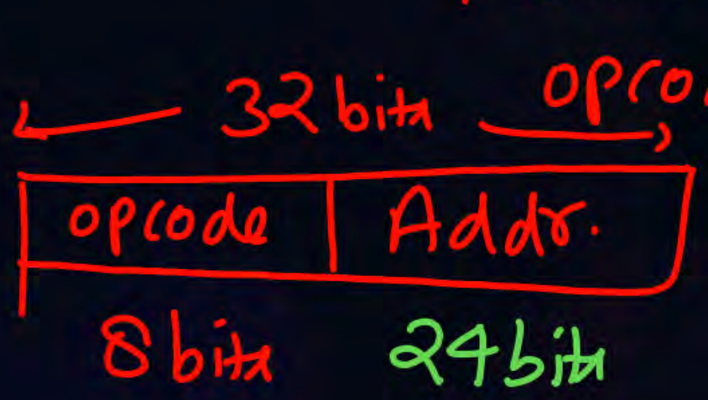
$32 - (27) = 5$



[NAT]

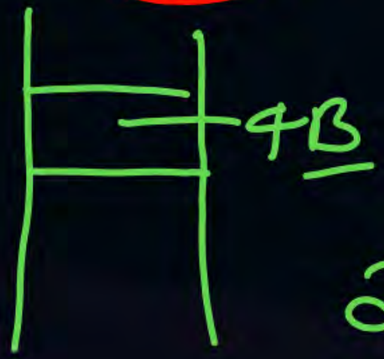
#Q. A digital computer has a memory unit with 32-bits per word. The instruction set consists of 240 different operations. All the instructions have an operation code part (opcode) and an address part (allowed for only 1 address). Each instruction is stored in one word of memory. The maximum allowable size of memory (word addressable) is 64 Mbytes?

1 word = 32 bits



opcode = $\log_2(240)$

6 bits = 64
7 bits = 128
8 bits = 256



$2^{24} \times 4B$

$2^{26} B$

$2^{26} B$

$2^6 \, 2^{20} B$

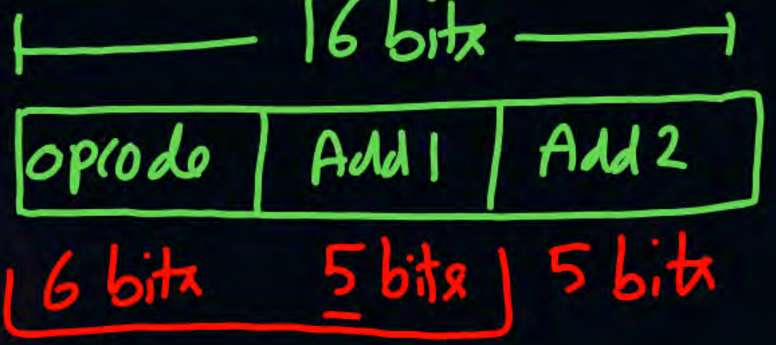
2^6 MB

64 MB

[NAT]

1024

#Q. Consider a system which supports 2-address and 1-address instructions. The system uses 16 bits instructions and 5-bits addresses. If there are total 32 2-address instructions then maximum how many 1-address instructions can be formulated?

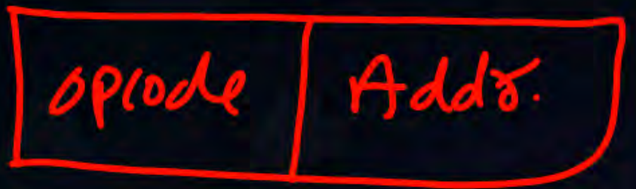


#of 2-Addr. Inst^{n} = 32

$(2^6 - 32) = 32 \times 2^5$

$= 2^5 \times 2^5$

$= 2^{10} = 1024$ 1-Addr. Inst^{n}

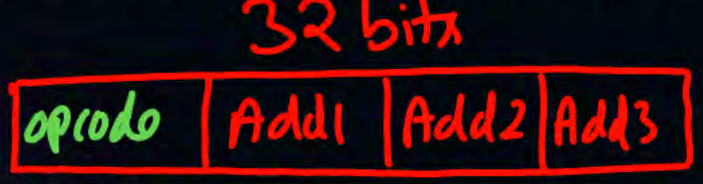


[NAT]

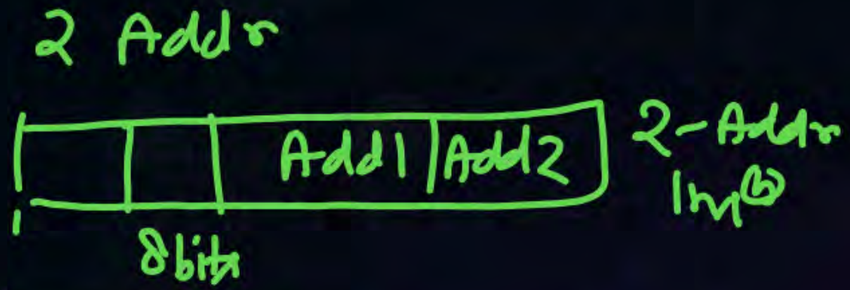
$(2\times2^8 - x)2^8 = 2^{10}$

$512 - x = 4 \Rightarrow x = 512 - 4 = 508$

#Q. 508 Consider a system which supports 3-address, 2-address and 1 -address instructions. It has 32-bit instructions with 8-bits addresses. If there are 254 3-address instructions and 1024 1-address instructions, then maximum how many 2-address instructions can be formulated?

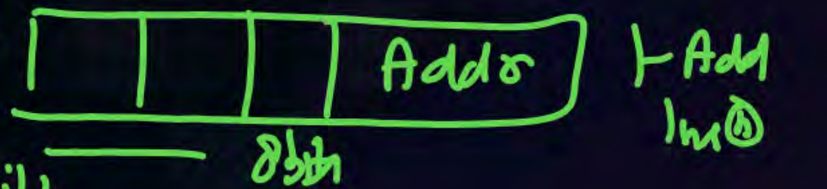


$(2^8 - 254) = 256 - 254 = 2$



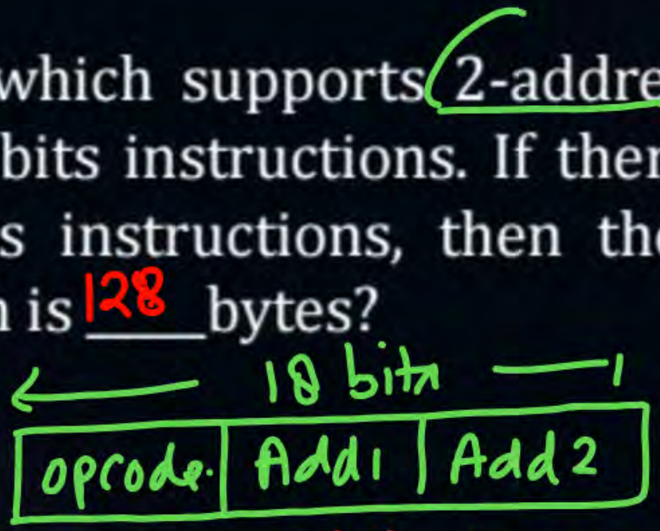
$(2\times2^8 - x)\times2^8$

1-Addr Inst possible



[NAT]

#Q. Consider a system which supports 2-address and 1-address instructions. The system has 18 bits instructions. If there are 7 2-address instructions and 1152 1-address instructions, then the maximum size of memory supported by system is 128 bytes?



2^7

Memory Size $= 2^7 \times B$

$= 128\ B$

$2^3 = 8 - 7 = 1$

4 bits

16

$(2^4 - 7) \times 2^7$ 1 Addr. Inst^{n}

$(16 - 7) \times 2^7$

$9 \times 128 = 1152$

[MCQ]

#Q. Consider a system which supports 2-address, 1-address and 0-address instructions. The system has 'i' bits instructions and 'a' bits addresses. If there are 'x' 2-address instructions and 'y' 1-address instructions then which of the following is correct for maximum number of 0-address instructions supported by system?

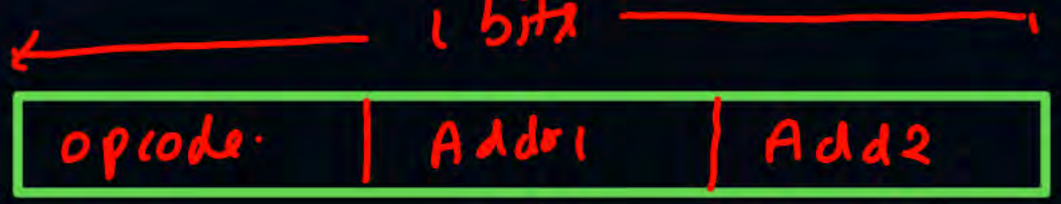
A $2^i - 2^a x - y$

B $2^i - 2^{2a} x - y$

C $2^i - 2^{2a} x - y2^a$

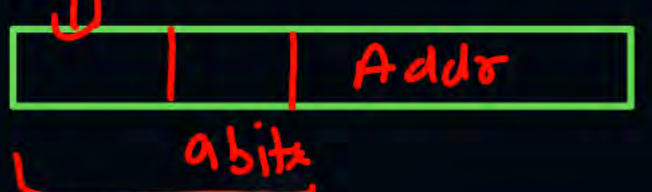
D $2^i - 2^a x - y2^a$

Ans [C]



2 Addr. Inst

1-Addr. Inst



# of 2-Addr. = x

# of 1- " = y

$$(2^{i-2a} - x)2^a - y)$$

maximum possible 0-Addr. Inst

$$= (2^{i-2a} - x)2^a - y) \times 2^a$$

$$= (2^{i-2a} - x)2^{2a} - y \cdot 2^a$$

$$= \underline{2^i - 2^{2a} \cdot x - 2^a \cdot y}$$

[NAT]

#Q. Consider there are 4 types of instructions in system:

Type 1: One opcode and 2 registers

Type 2: One opcode and 1 register

Type 3: One opcode and 1 memory address

Type 4: One opcode, 1 register and 1 memory address

Number of registers in CPU= 128

Maximum instruction length: 32bits (Variable length instructions supported)

Total Instructions: Type-1: 15, Type-2: 20, Type-3: 12, Type-4: 14

Maximum memory address size = 19 bits

#y Reg = 128

Reg. field = 7 bits

total ins^{n} = T1 + T2 + T3 + T4

= 15 + 20 + 12 + 14

= 35 + 26

= 61

≈ 61

Opcode = 6 bits

T-1

6 7 7

Size = 6 + 7 + 7 = 20 bits

T2

6 bit 7 bit

Size = 6 + 7 = 13 bits

T3.

6 bits 26 bits. ×19 bit = 6+19 = 25 bits

32

T4.

6 bits 7 bits 32 − (13) = 19 bits

6 + 7 + 26 = 39 bits ✗

[NAT]

3

#Q. Consider a register-based architecture system which can support maximum 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?

```
t1 = X + Y
t2 = Z * 2
t3 = t2 + A
t4 = t3 - t1
t5 = t4 + t3
```

$R_1 \leftarrow X$
$R_2 \leftarrow Y$
(t1) $R_1 \leftarrow R_1 + R_2$
$R_2 \leftarrow Z$
(t2) $R_2 \leftarrow R_2 \times 2$
$R_3 \leftarrow A$
(t3) $R_2 \leftarrow R_2 + R_3$
$R_3 \leftarrow R_2$
(t4) $R_3 \leftarrow R_3 - R_1$
$R_2 \leftarrow R_3 + R_2$

Note: X, Y, A and Z are memory operands; and consider first operand as destination operand and there is no any optimization done by compiler.

THANK - YOU