

Computer Science & Information Technology

Computer Organization & Architecture

Instruction & Addressing Modes

DPP: 1

- Q1** A relative branch mode type instruction is stored in memory starting from address 240. The branch is made to an address 140. What should be the value of relative address field of the instruction, if each instruction is stored on 2 memory locations?

16b @ PC

Note: All numbers are in decimal

- Q2** Consider the following:

1. Operation code
2. Source operand reference
3. Result operand reference
4. Next instruction reference

Which of the above are typical elements of machine instructions?

- (A) 1, 2 and 3 only
 (B) 1, 2 and 4 only
 (C) 3 and 4 only
 (D) 1, 2, 3 and 4

- Q3** Which addressing mode helps to access table data in memory efficiently?

- (A) Indirect mode
- (B) Immediate mode
- (C) Auto-increment or Auto-decrement mode
- (D) Index mode

- Q4** An addressing mode in which the location of the data is contained within the mnemonic, is known as?

- (A) Immediate addressing mode
- (B) Implied addressing mode
- (C) Register addressing mode
- (D) Direct addressing mode

- Q5** The addressing modes used for source operand in the following instructions are respectively?

- R1 \leftarrow #5
R1 \leftarrow M[5000]
R1 \leftarrow M[R2]
(A) Implied, direct, register
(B) Implied, direct, register indirect
 (C) Immediate, direct, register indirect
(D) Immediate, direct, register

- Q6** Consider a PC-relative mode type branch instruction which takes branch on address 720 in memory. The instruction has offset value 160. What is the starting address of this instruction in memory, if each instruction is stored in memory on 4 locations?

Note: All numbers are in decimal

- Q7** Consider the system in which in fetch cycle complete instruction is fetched. Which of the following addressing modes do(es) not require memory access for operand after fetch cycle?

- (A) Register Mode
(B) Register Indirect Mode
(C) Indirect Mode
(D) Indexed Mode

- Q8** 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?

47



[Android App](#)

| [iOS App](#)

| [PW Website](#)

Q9 Consider a system which supports 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 _____ bytes? **2200**

Q10 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

Q11 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 _____ Mbytes?

Q12 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? **1024**

Q13 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? **512**

Q14 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 _____ bytes?

Q15 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) $2^i - 2^a x - y2^a$

Q16 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)



[Android App](#)

| [iOS App](#)

| [PW Website](#)

Total Instructions: Type-1: 15, Type-2: 20, Type-3:

12, Type-4: 14

Maximum memory address size = _____ bits

- Q17** 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?

$$t1 = X + Y$$

$$t2 = Z * 2$$

$$t3 = t2 + A$$

$$t4 = t3 - t1$$

$$t5 = t4 + t3$$

3

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.

- Q18** Consider a register-memory 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?

$$t1 = X + Y$$

$$t2 = Z * X$$

$$t3 = t2 + t1$$

$$t4 = t3 - Y$$

$$t5 = t4 + t3$$

U

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.

- Q19** Consider a single AC-based architecture system which can support maximum 1-address instructions. For this system the following intermediate code is going to be converted in machine code. Minimum how many general-purpose registers are required in system so that the code can run without register spill?

$$t1 = X + Y$$

$$t2 = t1 * X$$

$$t3 = Y + Z$$

$$t4 = t3 - t2$$

$$t5 = t3 - t4$$

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.



[Android App](#)

[iOS App](#)

[PW Website](#)

Answer Key

Q1 -102~102

✓ Q2 (A)

✓ Q3 (C)

✓ Q4 (B)

Q5 (C)

Q6 556~556

✓ Q7 (A)

✓ Q8 2~2

✓ Q9 4000~4000

✓ Q10 (C)

✓ Q1 64~64

✓ Q12 1024~1024

Q13 508~508

Q14 128~128

✓ Q15 (C)

✓ Q16 19~19

✓ Q17 3~3

✓ Q18 2~2

Q19 2~2



[Android App](#)

| [iOS App](#)

| [PW Website](#)

Hints & Solutions

Q1 Text Solution:

PC value when the given address is in execution
 $= 240 + 2 = 242$
 Target address given = 140
 In PC relative mode target address = PC value + relative address
 $140 = 242 + \text{relative address}$
 Relative address = $140 - 242 = -102$

Q2 Text Solution:

Within an instruction operation code (opcode), source and destination operand references are specified in general. But next instruction reference is not because for that CPU maintains a register program counter (PC).

Q3 Text Solution:

To access table data (contiguous data) efficiently Auto-increment or Auto-decrement mode are used because using single instruction of these modes, entire table can be accessed without changing the instruction.

Q4 Text Solution:

Mnemonic means the character short form of opcodes, like for addition ADD, for subtraction SUB etc. So the addressing mode in which operand is specified within opcode itself is implied mode.

Q5 Text Solution:

$R1 \leftarrow \#5$, in this instruction source value 5 is mentioned in instruction, hence it is immediate mode

$R1 \leftarrow M[5000]$, in this instruction source value is taken from memory and memory address 5000 is mentioned in instruction hence it is direct mode

$R1 \leftarrow M[R2]$, in this instruction source value is taken from memory and memory address is given indirectly in register R2, hence it is register indirect mode.

Q6 Text Solution:

In PC relative mode target address = PC value + relative address

$$720 = PC + 160$$

$$PC = 720 - 160$$

$$PC = 560$$

PC value will be the address of next instruction when the current instruction is in execution, and current instruction is stored in memory on 4 locations, hence starting address of current instruction = $560 - 4 = 556$

Q7 Text Solution:

In register mode operand is present in CPU register, hence memory access is not required for operand.

In register indirect mode the register will provide memory address where the operand is stored, hence memory access is required.

In Indirect mode the operand is taken from memory only.

In indexed mode the address of operand is obtained by adding base address in index register value but operand is obtained from memory only.

Q8 Text Solution:

Number of words in memory = $2MB/4B = 512K = 2^{19}$, hence memory address size = 19-bits

Instruction format will be like:

Opcode	add.1	add.2
9	19	19

Instruction length = $9 + 19 + 19 = 47$ bits


[Android App](#)
[iOS App](#)
[PW Website](#)

To store 47 bits, 2 addresses in memory required.
Because in memory as given in question at one address one word of 4bytes (32 bits) only can be stored.

Q9 Text Solution:

Number of words in memory = $2\text{MB}/4\text{B} = 512\text{K} = 2^{19}$, hence memory address size = 19-bits

Instruction format will be like:

Opcode	add.1	add.2
9	19	19

Instruction length = $9 + 19 + 19 = 47$ bits

To store 47 bits, 2 addresses in memory required.

Because in memory as given in question at one address one word of 4bytes (32 bits) only can be stored.

To store 500 instructions, $500 \times 2 = 1000$

locations are required. Hence $1000 \times 4\text{Bytes} = 4000$ bytes space is needed in memory.

Q10 Text Solution:

Number of words in memory = $256\text{k} = 2^{18}$, hence address size = 18 bits

Number of registers = 64, hence register field has 6 bits

Number of modes = 8, hence bits for modes = 3 bits

Instruction is stored on exactly 1 word in memory.

Hence instruction length = 32 bits

Instruction format will be as follows:

32			
opcode	mode	Register	Memory address
3	6	18	

Bits in opcode = $32 - (3+6+18) = 5$ bits

Q11 Text Solution:

Number of distinct instructions supported = 240, hence number of bits in opcode = 8 bits

Instruction is stored on exactly 1 word in memory.

Hence instruction length = 32 bits

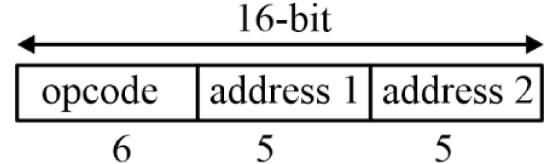
Number of bits for address = $32 - 8 = 24$ bits

Number of addresses in memory = 2^{24}

Size of memory = $2^{24} * 32\text{ bits} = 2^{24} * 4\text{ bytes} = 2^{26}\text{ bytes} = 64\text{ Mbytes}$

Q12 Text Solution:

2-address instruction format:

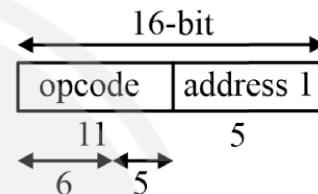


Max number of opcode combinations = $2^6 = 64$

Used opcodes = 32

Unused = $64 - 32 = 32$

1-address instruction format

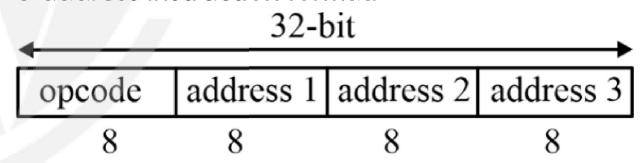


Maximum number of 1-address instructions = $32 \times 2^5 = 1024$

Q13 Text Solution:

Assume there are x 2-address instructions used.

3-address instruction format:

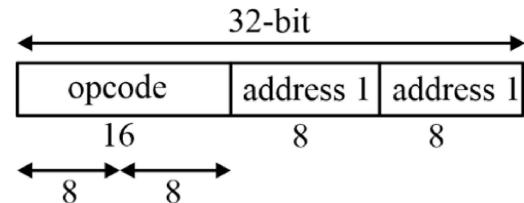


Maximum 3 address instructions = $2^8 = 256$

Used 3 address instructions = 254

Unused opcodes = $265 - 254 = 2$

2-address instruction format:



Maximum 2 address instructions = $2^*2^8 = 512$

Used 2 address instructions = x

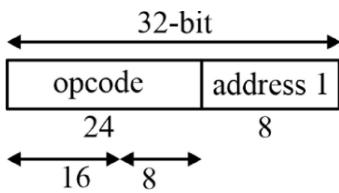
Unused opcodes = $512 - x$



[Android App](#)

| [iOS App](#)

| [PW Website](#)



1-address instruction format:

$$\text{Maximum 1 address instructions} = (512 - x) \times 2^8$$

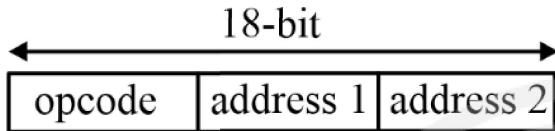
$$\text{Given 1-address instructions} = 1024,$$

$$\text{Hence } (512 - x) \times 2^8 = 1024$$

$$x = 508$$

Q14 Text Solution:

2-address instruction format:



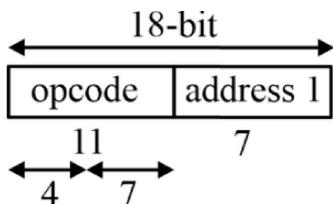
As given number of 2 address instructions = 7, hence opcode must have atleast 3 bits, but if we use 3 bits opcode then for both the addresses 15-bits are remaining, and 15-bits can't be distributed in 2 equal parts. Hence we will start with opcode as 4-bits, so that each address can have 7-bits equally.

$$\text{Maximum 2 address instructions} = 2^4 = 16$$

$$\text{Used 2 address instructions} = 7$$

$$\text{Unused opcodes} = 16 - 7 = 9$$

1-address instruction format:

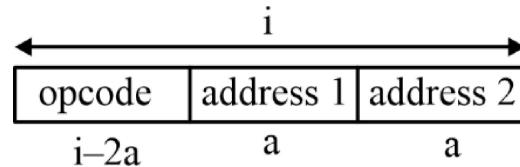


$$\text{Maximum 1 address instructions} = 9 \times 2^7 = 1152, \text{ which is matching with given data question.}$$

Hence memory address length is 7-bits. Memory size = 2^7 bytes = 128 bytes

Q15 Text Solution:

2-address instruction format:

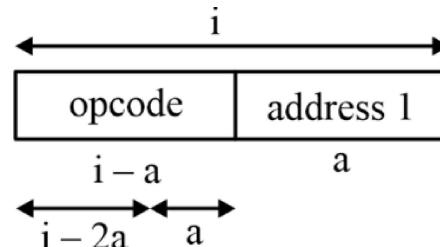


$$\text{Maximum 2 address instructions} = 2^{i-2a}$$

$$\text{Used 2 address instructions} = x$$

$$\text{Unused opcodes} = (2^{i-2a} - x)$$

1-address instruction format:

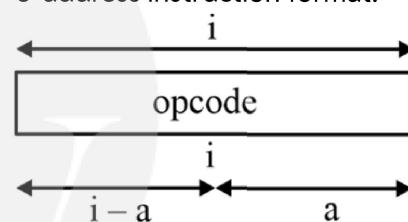


$$\text{Maximum 1 address instructions} = (2^{i-2a} - x) \times 2^a = 2^{i-a} - x2^a$$

$$\text{Used 1 address instructions} = y$$

$$\text{Unused opcodes} = 2^{i-a} - 2x^a - y$$

0-address instruction format:



$$\text{Maximum 0 address instructions} = (2^{i-a} - x2^a - y) \times 2^a = 2^i - x2^{2a} - y2^a$$

Q16 Text Solution:

Number of registers = 128, hence register field in instruction will have 7 bits

As given in question that the variable length instructions are supported, hence for all types of instructions, opcode size will be fixed.

Total number of instructions = 15 + 20 + 12 + 14 = 61, hence opcode will have 6 bits

Type 1 Instruction format

Instruction length of type 1 = 6+7+7=20 bits

Opcode	Register-1	Register-2
6	7	7



[Android App](#)

| [iOS App](#)

| [PW Website](#)

Type 2 Instruction format

Opcode	Register-1
6	7

Instruction length of type 2 = $6+7=13$ bits

Type 3 Instruction format

opcode	Memory address
6	

Here memory address is not known.

Type 4 Instruction format

opcode	Register	Memory address
6	7	

Here also memory address is not known. If assuming type 3 uses maximum size instruction of 32-bits, then memory address size will be = $32 - 6 = 26$ bits.

But 26-bits memory address will make type 4 instruction size = $6 + 7 + 26 = 39$ bits which is larger than given maximum instruction size of 32 bits. Hence Type 4 should use maximum instruction size of 32 bits and memory address will be length = $32 - (7+6) = 19$ bits.

Based on it type 3 instruction length will be = $6 + 19 = 25$ bits

Q17 Text Solution:

Following will be instructions generated for the given basic block:

$R1 \leftarrow X$
 $R2 \leftarrow Y$
 $R1 \leftarrow R1 + R2$
 $R2 \leftarrow Z$
 $R2 \leftarrow R2 * 2$
 $R3 \leftarrow A$

$R2 \leftarrow R2 + R3$

$R3 \leftarrow R2$

$R3 \leftarrow R3 - R1$

$R3 \leftarrow R3 + R2$

Total number of registers needed = 3

Q18 Text Solution:

Following will be instructions generated for the given basic block:

$R1 \leftarrow X$

$R1 \leftarrow R1 + Y$

$R2 \leftarrow Z$

$R2 \leftarrow R2 * X$

$R2 \leftarrow R2 + R1$

$R1 \leftarrow R2$

$R2 \leftarrow R2 - Y$

$R2 \leftarrow R2 + R1$

Total number of registers needed = 2

Q19 Text Solution:

Following will be instructions generated for the given basic block:

$AC \leftarrow X$

$AC \leftarrow AC + Y$

$AC \leftarrow AC * X$

$R1 \leftarrow AC$

$AC \leftarrow Y$

$AC \leftarrow AC + Z$

$R2 \leftarrow AC$

$AC \leftarrow AC - R1$

$R1 \leftarrow AC$

$AC \leftarrow R2$

$AC \leftarrow AC - R1$

Total number of general-purpose registers needed apart from Accumulator = 2



[Android App](#)

| [iOS App](#)

| [PW Website](#)