1 + 1 = 2
Fetch Write

(iv) AND R1 R2
1 - Fetch.

v Increament A
1 + 1 + 1 = 3.
MR MW

11. Base Register Addressing mode
Numericals.

1. How many memory references are required to execute the following instructions?
(i) Add (R1) R3
where R3 is the destination.
Register indirect ; To fetch the data from memory
Fetch
1 + 1 = 2.

(ii) SUB #600, R5 , R5 is Destination
Memory References = 1

2. Write the number of memory referenes required for executing the following instruction.
(i) ADD R1, (R2)+
Auto increment write.
1 + 1 + 1 = 3.
F Fetch operand

(i) SUB #10, R2
Memory References = 1

(iii) MOV R1, 20(R3, R4)
M.R. = 1 + 1 = 2

(iv) AND R1, R2
M.R = 1

(v) Increment A → A ← [A] + 1
= 1 + 1 + 1 = 3

Ans3 Registers R1 and R2 of a computer contains the decimal value 100 and 200. What are the effective address of memory operand in each of the following instruction?

(i) LOAD 20(R2), R1

EA = [R2] + 20 = 200 + 20 = 220

Given
R1 = 100
R2 = 200

(ii) Move 300, R5
EA = 300

(iii) ADD (R1), R2, [R1] = 100
EA = 100

(iv) MUL (R1) +, R5 (Auto increment)
EA = [R1] = 100, R1 = [R1] + 1 = 101

If it will be byte addressability then it may be
100+2 - for 16 bit length
100+4 = 104 for 32 bit length
If nothing given, default value is 1.

Numerical:

(1) A machine has a 32-bit architecture with 1-word long instructions. It has 60 registers, each of which is 32 bits long. It needs to support 45 instructions, which have an immediate operand in addition to two register operands. Assuming that the immediate operand is an unsigned integer, what is the maximum value of the immediate operand?

Note: To represent n combination in binary, to find the no of bits required, we can find the $\log_2 n$.

Ans

opcode	imm. operand	Reg1	Reg2
6	14	6	6

32

immediate operand = 32 - (6+6+6) = 14

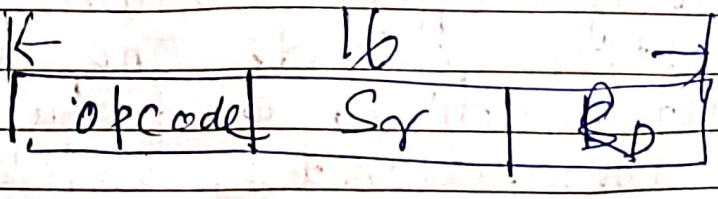
Unsigned = +ve number,

45 instruction, so opcode = 6 bits.
($2^6 = 64$)
$2^5 = 32 < 45$ (so 5 is not considered)
So Register = $2^6 = 64$.
So Length = 6

Magnitude of Maximum Value will be $2^{14} - 1 = 16383$. all bits will be used to represent a value. as no sign bit is required and numbering starts from 0. so the max value will be max − 1.

Qs-2 A general purpose register organization computer has a 16 bit instruction consisting opcode, source register and a destination register. It supports 7 no of arithmetic operations and 6 no of logical operations. Find the total number of maximum registers present in the system.

Ans:



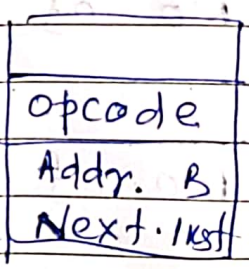
Sr, Source register,
SD, Destination Register

So each instruction
Length = 5 bytes (40 bits)

∴ Amount of memory required
= no of Instr? x size of each instr?
= 100 X 5B = 500 Bytes

Qs 4. What is a two word instruction is stored in a location A. The operand part of instruction holds B If the addressing mode is relative, the operand is available in which location.

Ans Relative Mode.
EA = updated value of PC + offset
= A+2+B



Updated Value of PC = Address of the next instruction i-e the address of the instruction following the branch instruction.

Qs 5 A relative mode branch type instruction is stored in memory at an address 750. The branch is made to an address 500. What would be the value of the relative address field of the instruction?

Ans

Relative Mode

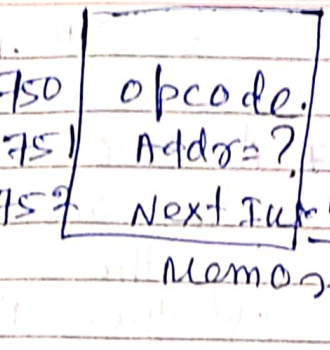
EA = Updated value of PC + offset

500 = 752 + offset

offset = 500 − 752

= −252

Address field value = −252.



Qs 6) An instruction is stored at location 200 with its address field having the value 10. A processor register R10 contains the value 210 which is also used as an Index register. Evaluate the effective addr. of the operand, if the addressing mode of the instruction is,

(i) Direct

(ii) Register direct.

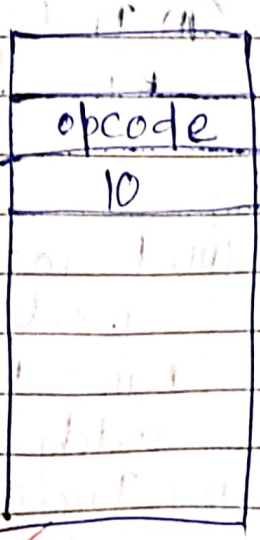
(iii) Register Indirect.

(iv) Indexed.

Ans (i) Direct Addressing Mode.

EA = Value of address field

= 10



(ii) Register Direct Addressing Mode.

EA = Ri = Name of the reg

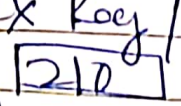
= R10

(iii) Register Indirect Mode

EA = [Ri] = [R10] = 210

(iv) Index Addressing Mode

EA = [Index Reg] + offset (value of the addr. field)



= 210 + 10 = 220

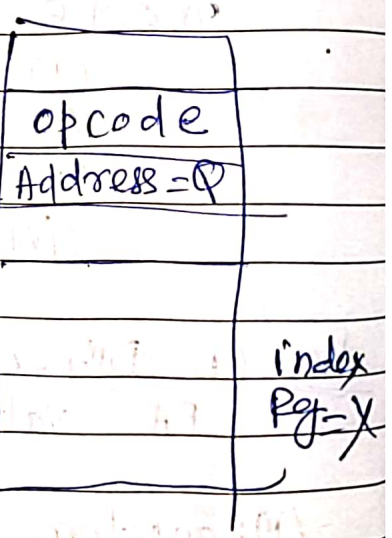
Q7 A two-word Instruction is stored in memory at an address designated by the symbol P. The address field of the instruction (stored at P+1) is designated by the symbol Q. The operand used during the execution is stored at an address symbolized by EA. An index Register contains the value X. State how EA is calculated from the other address if the addressing mode of the instruction is direct, indirect relative and indexed.

Ans: (i) Direct Addressing mode:
EA = Value of the address field = Q.

(ii) Indirect Mode.
EA = [value of the address field]
= [Q]

(iii) Relative Addressing Mode:
EA = Updated [PC] + offset (value of the address field) = P+2+Q.

(iv) Index Addressing mode.
EA = [Index Reg] + offset (value of addr. field)
= X + Q.



Q8 A two word Instruction LOAD is stored at location 1000 with its address field at location 1001. The address field has the value 2000 and the value stored at 2000 is 5000 and at 5000 is 6500. The words stored at 2200, 3002 and 3500 are 3500, 4000 respectively. An Index register has value 200. Evaluate the effective address and operand if addressing mode of the instruction is as follows.

(i) Memory Indirect Addressing Mode.
(ii) Relative Addressing Mode.
(iii) Index Addressing Mode.

Ans (i) Memory Indirect Addressing Mode.

EA = [value of the address field]
[2000] = 5000
Operand = [5000] = 6500

1000	LOAD
1001	addr = 2000
1002	← Updated value of PC
2000	5000
2200	3500
3002	4000
5000	6500

Index 200

(ii) Relative addressing Mode.

EA = Updated [PC] + offset (value of the address field)
= 1002 + 2000 = 3002
Operand = [3002] = 4000

(iii) Index Addressing Mode.

EA = [Index Reg] + offset (value of the addr. field)

$= 200 + 2000 = 2200$

operand = [2200] = 3500.

Q9 An instruction is kept in memory at an address 300 and the memory address 301 occupies the address field of the instruction which is shown below. The opcode is used to add the content of accumulator with an operand. The content of accumulator is 100 and the content Register R_5 is 400. Find out the content of accumulator and effective address of operand if the addressing mode is

Ans

(i) Immediate.

EA = NULL as operand is part of the instr.

[Acc] = [Acc] + Imm operand
(Value of the addr. field)
= 100 + 500 = 600

(ii) Direct

EA = Value of Address field
= 500

Acc = [Acc] + mem [500]
= 100 + 600 = 700.

300	ADD
301	addr. = 500
400	700
401	456
500	600
600	800

Acc = 100

R_5 = 400

(iii) Register direct.

EA = Ri = Name of the register = Rs

[ACC] = [ACC] + [Rs]

= 100 + 400 = 500

(iv) Indirect. Memory

EA = [value of the address field

= mem [500] = 600

[ACC] = [ACC] + mem [EA]

= 100 + 800 = 900

(v) Register Indirect.

EA = [Ri] = [Rs] = 400

[ACC] = [ACC] + mem [EA]

= 100 + mem [400]

= 100 + 700 = 800

11) Consider the following instruction sequence where registers R_1, R_2 and R_3 are general purpose registers and Memory [X] denotes the content of the memory location X

Instr	operation	Inst Size.
MOV R_1, (5000)	$R_1 \leftarrow$ M[5000]	4
MOV R_2, (R_3)	$R_2 \leftarrow$ M[R_3]	4
ADD R_2, R_1	$R_2 \leftarrow R_1 + R_2$	2
MOV (R_3), R_2	M[R_3] $\leftarrow R_2$	4
INC R_3	$R_3 \leftarrow R_3 + 1$	2
DEC R_1	$R_1 \leftarrow R_1 - 1$	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 contents of the Register R_3 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 numerals are in decimal format. Assume that the memory is byte addressable.

After the execution of the Program, the content of memory location 3010 is

Ans. [3000] + R3

3000 - 50
⋮
3010 50
5000 -10

R1=10, R2=50
R3=50 → R2=50
R2+R1 = 50+10=60=R2
3000=60
3001=59 3006=54
3002=58 3007=53
3003=57 3008=52
3004=56 3009=51
3005=55 3010=50

11. If PC = 2004, after executing the instruction pointed by PC, what will be the updated value of PC?
(Initial value R1=10, R2=100)

Ans Result of addition of R1 and R2 is 110 ≠ 0

BNZ is a conditional branch instruction, it will jump to the instruction specified in it's label field, if the result of the previous instruction is NOT equal to zero.
PC = [PC]+4 = 2004+4 = 2008.
By the time the instruction at 2004 is executed, the PC will be incremented to point to the next instruction in sequence.

Hence PC = 2008+1000
= 3008.

2000 ADDR1,R2
2004 BRZ 1000
2008 next instruction

12. If PC = 2004, after executing the instruction pointed by PC, what will be the updated value of PC? (Initial value of $R_1 = 10$, $R_2 = 100$)

memory Location	Instruction
2000	ADD R_1, R_2
2004	BZ 1000
2008	Next Instruction

Ans:

Result of addition R_1 and R_2 is $110 \neq 0$

BZ is a conditional branch instruction, it will jump to the instruction specified in its label field, if the result of the previous instruction is equal to zero.

By the time the instruction at 2004 is executed, The PC will be incremented to point to the next instruction in sequence Hence PC = 2008, as the branch instruction is not true.

13. If PC = 2004, after executing the instruction pointed by PC, what will be the updated value of PC? (initial value of $R_1 = 10$, $R_2 = -100$)

Memory Location	Instruction
2000	ADD R_1, R_2
2004	Branch >0 1000
2008	Next instruction.

Ans $R_1 + R_2 = -90$, not greater than 0

Branch >0 is a conditional branch instruction, it will jump to the instruction specified in it's label field, if the result of the previous instruction greater than zero.

Here condition is not satisfied.

By the time the instruction 2004 is executed the PC will be incremented to point to the next instruction in sequence i.e 2008.

Hence PC = 2008 as the branch condition is not ~~true~~.

14. If PC = 2048 and at 2048 memory location "JUMP 1000" instruction is there, After execting this instruction, what will be the value of PC?

Ans

Memory location	Instruction
2048	JUMP 1000
2052	Next instruction

JUMP is same JMP.
= go to L1

Ans Jump is an unconditional branch instruction, irrespective of the result of the previous instruction it will jump to the instruction specified in its label field.
(NO condition Checking)

By the time instruction at 2048 is exected, the PC will be incremented to point to the next instruction in sequence i.e 2052

Hence PC = 2052 + 1000
= 3052.

15. An instruction is kept in memory at an address 401 occupies the address field of the instruction which is shown below. The opcode is used to add the content of accumulator with an operand. The content of accumulator is 100 and the content of register R600 is 500. What will be the effective address of the operand and the content of accumulation after the execution of the instruction, if the addressing mode is (i) Indirect (ii) Register Indirect

Ans

400 opcode.

401 600

500 800

600 700

700 900

R 600 → 500

(i) Indirect Addressing Mode:

EA = [value of the address field]
= [600] = 700

(ii) Indirect Addressing Mode.

operand = mem[EA] = mem[700]
= 900

Hence [Acc] = 900 + 100 = 1000

(ii) Register Indirect:

EA = [Regi] = [R600] = 500

operand = mem[EA] = mem[500] = 800

Hence = 800 + 100 = 900

16 Consider the code given below. What will be the offset for the instruction BGTZ to branch to the location labeled Loop? All the instructions are 4 bytes in length.

BGTZ – Branch on greater than 0.

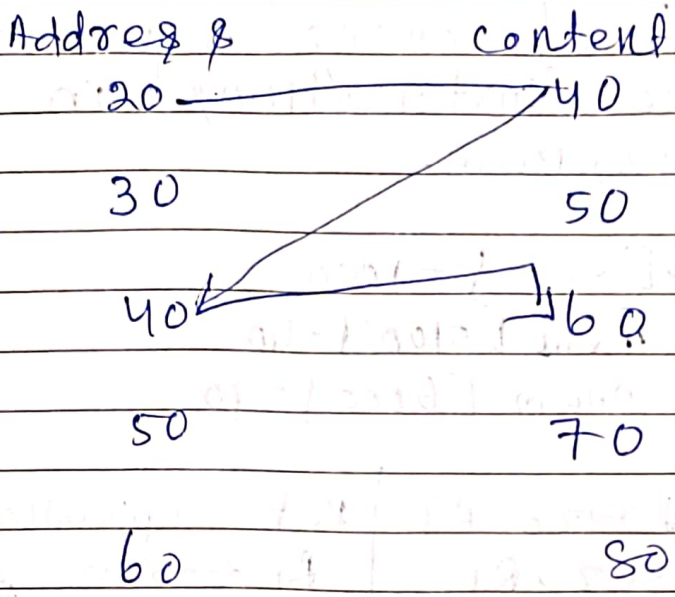
```
1000    MOV #10, R1
1004    MOV #LOCA, R2
1008    CLEAR R3
1012 LOOP:  ADD (R2)+, R3
1016        DEC R1
1020        BGTZ ?
1024
```

TARGET = UPDATED [PC] + offset

1012 = 1024 + offset

offset = 1012 − 1024 = −12

(17) Consider the below given memory map.



which of the followeney instruction will load 60 in the accumulator?

(a) LOAD #40
(b) LOAD 60
(c) LOAD (20) ✓
(d) LOAD (40)

(18) The contents of memory locations 5000, 5100 and 6000 are 1000, 40 & 90 respectively before the following program is executed. [Here, the 2nd operand is the destination]

```
MOV #5000, R1      — 1
MOV 100(R1), R2    — 2
ADD R2, 6000       — 3
ADD (R1)+, R2      — 2
MOV R2, (5000)     — 3
```

M. Ref.

what will be the contents of memory locations 1000, 5000, 6000 and register

R_1 and R_2 after the Program is executed? Find the number of memory references required for each of the instructions in the above program.

Ans Given $[5000] = 1000$

mem $[5100] = 40$

mem $[6000] = 90$

~~MOV #5000, R_1~~	Ref. operations
MOV #5000, R_1	$R_1 \leftarrow 5000$
MOV 100(R_1), R_2	$R_2 \leftarrow$ mem [100+5000], $R_2 = 40$
ADD R_2, 6000	$40 + 90 = 130 \rightarrow 6000$
ADD (R_1)+, R_2	$R_2 \leftarrow$ mem $[R_1] + [R_2]$
	$R_2 \leftarrow 1000 + 40 = 1040$
	$R_1 \leftarrow 5001$
MOV R_2, (5000)	(5000) $\rightarrow$ 1000
	$R_2 = 1040$
	hence mem [1000] $\leftarrow$ 1040

$R_1 = 5001$

$R_2 = 1040$

1000 = 1040

(19) Write the assembly language code the following Pseudo code using the addressing modes known to you / applicable for the given situation. Here P is a pointer to an integer.

```
P = 1000;
*P = 10
P++;
d = *P + 20
```

Ans

```
MOVE  #1000, R0
MOVE  R0, P
MOVE  #10, R1
MOVE  R1, (P)
INC   P
MOV   #20, R2
ADD   (P), R2
MOVE  R2, D
```

here bracket is used because P is a pointer.

20 Match each of the high level language statements given on the lefthand side with the most natural addressing mode from those listed on the righthand side.

1. A[I] = B[J];
2. While [*A++];
3. int temp = *X;

(a) Indirect addressing
(b) Indexed Addressing
(c) Autoincrement

Match columns:

(A)	B
Indirect	Relocatable code
Index	Passing array as a parameter
Base Register	Array
Auto increment	Ex: *(A++)

21. Write the equivalent instructions for zero Address organization & one address Organization of the following instructions:

$R_1 = P$ ← MOV P, R_1
$P - Q \rightarrow R_1$ SUB Q, R_1
$\frac{P-Q}{R} = R_1$ DIV R, R_1
$\left(\frac{P-Q}{R}\right) * S$ MUL S, R_1
MOV R_1, X

General Purpose organisation & 2 addr. Instruction
$R_1 \leftarrow P$, $R_1 \leftarrow P-Q$
$X \leftarrow R_1 \leftarrow (P-Q)/R * S$

Ans Zero address organization.

```
PUSH P
PUSH Q
SUB
PUSH R
DIV
PUSH S
MUL
POP X
```

one Address organisation.

LOAD	P	$AC \leftarrow m[P]$
SUB	Q	$[Acc] - Q \rightarrow [Acc] = P - Q$
DIV	R	$\frac{P-Q}{R} \rightarrow Acc$
MUL	S	$\left(\frac{P-Q}{R}\right) S$
STORE	X	$X = \left(\frac{P-Q}{R}\right) S.$

Basic operational concept.

Examples:

① ADD R_1, R_2

steps:

1. [PC] → MAR
2. Generate the 'Read' signal.
3. Wait for the MFC (Memory function to complete)
4. [PC]++
5. [MDR] → IR

5. Decode the instruction.
6. Execute the instruction. (ALU will perform the addition on the contents of R_1 and R_2 and Result of addition will be stored into R_2.

② ADD A, R_1

m[A] + [R_1] → R_1

steps.

1. [PC] → MAR
2. Generate the (Read signal)
3. Wait for the MFC (memory function to complete)
4. [MDR] → IR
5. Decode the instruction
6. Fetch the operand stored

at memory Location A

i. Address part of IR
ii. Generate the R → MAR. Read signal.
iii. wait for the MFC (memory function to complete)
iv. [MDR] → Input of ALV

v. Execute the instruction. ALV will perform the addition on the contents of memory location A & R1 and result of addition will be stored into R1

ADD A, R1 → 2(1+1) Memory references.

Example-3:

(3) ADD R1, A: → m[A] ← m[A] + [R1]

Steps:
1. [PC] → MAR
2. Generate the Read signal.
3. Wait for the MFC (Memory function to complete [PC]++
4. [MDR] → IR
5. Decode the instruction
6. Fetch the operand stored at memory

Location A.

i. Address part of IR → MAR
ii. Generate the Read signal
(iii) wait for the MFC (Memory function to complete)
(iv) [MDR] - Input of ALU

7. Execute the Instruction
ALU will perform the addition on the contents of memory Location A and R1

8. Store the result into the memory Location A.

1. Address part of IR → MAR
2. MDR ← Result from ALU
3. Generate the write signal
4. wait for the MFC
(Memory function to complete)

Memory References

1 + 1 + 1 = 3.

O.F. ↓ M.R ↓ M.W.

- To Fetch the Instruction.
- To Fetch the operand from memory location A.
- To store the result into memory Location A.

4. (a) ADD R1, R2.
How many memory references required.
Answer - 1

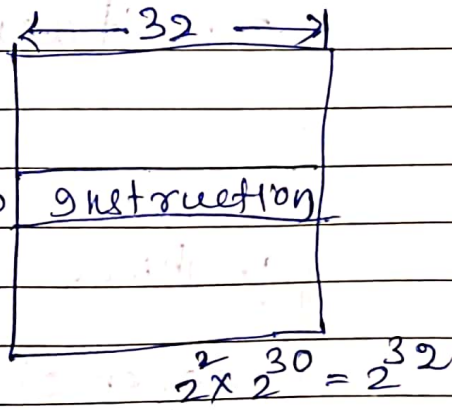
(b) MOV R1, A. (O.F. M.W)
Answer - 2 (1 + 1)

(c) MOV A, R1 (O.F. M.R.)
(1+1)

Numerical:

1. A processor is connected to 4k x 32 bit memory module. A program is kept in 100th address of the memory and the maximum length of each instruction of the Program is of 32 bits. Then find out size of MAR, MDR, IR and also the content of PC?

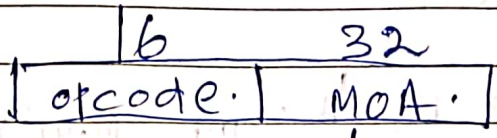
Ans →
[PC] = PC
MAR = 32 bits
MDR = 32 bits.
IR = 32 bits



2. A processor has 48-bit instructions composed of two fields: The first two bytes contain the opcode and the remainder a memory operand. How many bits are

are needed for the program counter and the instruction Register?

Ans.



IR = 48 bits.
[PC] = 32 bits.

(CPU Organization)

1. Single Accumulator organization (One address Instruction)
2. General Register Organization. (Two & Three address Instruction)
3. Stack organization (zero address Instruction)

1. Three address instruction.

General Format

opcode Destination SRC1, SRC2

Maybe opcode SRC1, SRC2, Destination

Note → The order of operands vary from architecture to Architecture.

Numerical - 1

If a 8Ghz computer takes 7 clock cycles for ALU instructions, 11 clock cycles for branch instructions and 6 clock cycles for data transfer instructions. Then find the total time taken by the computer to execute the program that consists of 10 ALU instructions, 5 branch instruction and 5 data transfer instructions.

Solution:

Total Numbers instructions

$$N = 10+5+5 = 20$$

Total cycles $= 7\times10 + 11\times5 + 6\times5$

$$= 70 + 55 + 30 = 155$$

$$S = CPI = 155/20$$

$R = 8$ Ghz.

$$\text{So } T = \frac{N \cdot S}{R} = (155/20) \times 20 \times \frac{1}{(8\times10^9)} \text{ sec}$$

$$= 19.375 \text{ nsec.}$$

Numerical-2

Suppose a Program (or a Program task) takes 1 billion instructions to execute on a processor running at 2 Ghz. Suppose also that 50% of the instructions execute in 3 clock cycles, 30% execute in 4 clock cycles and 20% executes in 5 clock cycles. What is the execution time for the program or task?

Ans

Total number of Instruction

$$N = 10^9$$

value	Frequency	Product
3	50% = 0.5	1.5
4	30% = 0.3	1.2
5	20% = 0.2	1.0

$$S = CPI = 1.5 + 1.2 + 1 = 3.7$$

$$R = 2 \text{ Ghz}$$

$$T = \frac{S \times N}{R} = \frac{3.7 \times 10^9}{2 \times 10^9} \text{ sec}$$

$$= 1.8 \text{ sec}$$

3. A program is running on an 8 Mhz Processor. Assume that 50% instructions perform an ALU operation, 30% instructions perform memory operations and 20% instructions perform Branching. Further assume that instructions performing an ALU take 4 clock cycles, instructions performing memory operation take 9 clock cycles and instructions performing Branching take 7 clock cycles. What is the total time taken by the Program?

Solution –

Total number of instructions
$N = 100$

Values	Frequency	Product
4	50% = 0.5	2.0
9	30% = 0.3	2.7
7	20% = 0.2	1.4

$S = CPI = 6.1$, $R = 8$ Mhz

$$CPU\ T = \frac{N \times S}{R} = \frac{100 \times 6.1}{8 \times 10^6}\ sec$$

$$= 76.25\ \mu sec.$$