1.

I. Cache lines = $cache\ size / block\ size$

= 512∗2^10 / 4∗4

= 2^15

2. 16 = 2^4

Associative mapping cache

28 bits	4bits

3. Cache lines = 2^15

Direct mapping cache

13 bits	15 bits	4 bits

4. when cache is 8-way set associative, sets of cache = 2^15 / 8

= 2^12 sets

5. Cache sets = 2^12

8- way set associative mapping

16 bits	12 bits	4 bits

6. 0x B1AC95F9 = 0b 10110001101011001001010111111001

Tag 1011000110101100	Set 100101011111	Word 1001

2.

X =(A*B-C) + (A/C)

1. Stack based

PUSH A
PUSH B
MUL
PUSH C
SUB
PUSH A
PUSH C
DIV
ADD
POP X

2. Accumulator based

LOAD A
DIV C
STORE P
LOAD A
MUL B
SUB C
ADD P
STORE X

3. Memory-memory based

<u>3 operands</u>

DIV N, A, C
MUL L, A, B
SUB M, L, C
ADD X, M, N

<u>2 operands</u>

MOV P, A
DIV P, C
MOV R, A
MUL R, B
SUB R, C
ADD R, P

4. **Register-Register based**

LOAD R1, A
LOAD R2, B
LOAD R3, C
DIV R4, R1, R3
MUL R5, R1, R2
SUB R5, R5, R3
ADD R4, R5, R4
STORE X, R4

3.

-6 = (00110) => 110102 = M
9 = 010012 = Q
-6 * 9 =>

A	Q	Q_{-1}	M	Cycle
00000	01001	0	11010	Initial state
00110	01001	0	11010	First cycle
00011	00100	1	11010	
11101	00100	1	11010	Second cycle
11110	10010	0	11010	
11111	01001	0	11010	Third cycle
00101	01001	0	11010	Forth cycle
00010	10100	1	11010	
11100	10100	1	11010	Fifth cycle
11110	01010	0	11010	

Answer: 1111001010

reverse 2's compliment =- 54

4.

[MAR] <- [PC] Transfer the address from the PC to MAR.

[MBR] <- [MBR] Read the memory into the MBR.

[IR] <- [MDR] Copy the instruction from the MBR to the instruction register.

[PC] <- [PC]+1 Program counter is incremented.

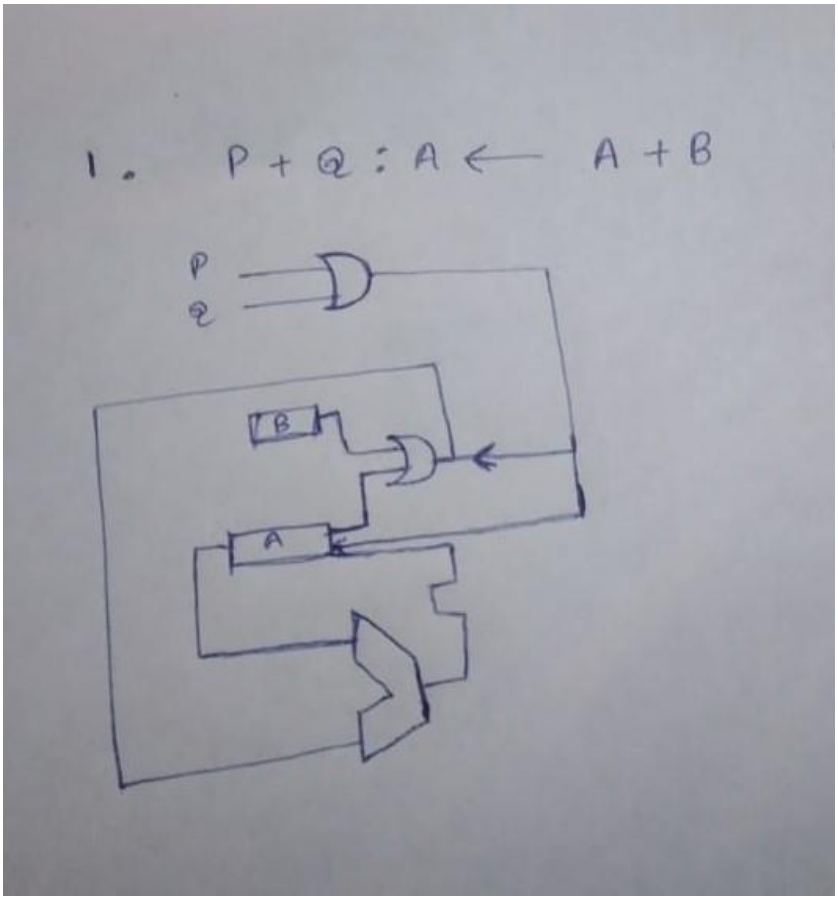
[MAR] <- IR (operand) Decode, then execute.

[MBR] <- [MAR] Reading the memory.

AC<- MBR Transfer to the Accumulator.

5.

1.



2.