COMPUTER ORGANIZATION AND ARCHITECTURE

ASSIGNMENT –2

1. Consider the instruction formats of the basic computer. For each of the following 16-bit instructions, give the equivalent four-digit hexadecimal code and explain in your own words what it is that the instruction is going to perform.

   a. 0001 0000 0010 0100 b. 1011 0001 0010 0100 c. 0111 0000 0010 0000

   (a) 0001 0000 0010 0010 = $(1024)_{16}$

   ADD $(024)_{16}$

   ADD content of M [024] to AC → **ADD 024**

   (b) 1 011 0001 0010 0100 = $(B124)_{16}$

   I STA (124)6

   Store AC in M [M [124]] → **STA @124**

   (c) 0111 0000 0010 0000 = $(7020)_{16}$

   Register Increment AC → **INC**

2. Write a program to evaluate the arithmetic statement:

$$X = \frac{A - B + C * (D * E - F)}{G + H * K}$$

   a. Using a general register computer with three address instructions.

   b. Using a general register computer with two address instructions.

   c. Using an accumulator type computer with one address instructions.

   d. Using a stack organized computer with zero-address operation instructions.

   a) Three address instructions:

SUB R1, A, B	R1←M [A] - M [B]
MUL R2, D, E	R2←M [D] * M [E]
SUB R2, R2, F	R2← R2 – M [F]
MUL R2, R2, C	R2← R2*M [C]
ADD R1, R1, R2	R1← R1+R2
MUL R3, H, K	R3← M [H] + M [K]
ADD R3, R3, G	R3← R3+ M [G]
DIV X, R1, R3	X← R1/R3

b) Two address instructions:

MOV R1, A	R1←M [A]
SUB R1, B	R1←R1-M [B]
MOV R2, D	R2←M [D]
MUL R2, E	R2←R2*M [E]
SUB R2, F	R2←R2 – M [F]
MUL R2, C	R2←R2*M[C]
ADD R1, R2	R1←R1+R2
MOV R3, H	R3←M [H]
ADD R3, G	R3←R3+M [G]
DIV R1, R3	R1←R1/R3
MOV X, R1	M[X]←R1

c) One Address instructions:

LOAD A	AC← M [A]
SUB B	AC←AC-M [B]
STORE T	M [T]←AC
LOAD D	AC←M [D]
MUL E	AC←AC*M [E]
SUB F	AC←AC-M [F]
MUL C	AC←AC*M[C]
ADD T	AC←AC+M [T]
STORE T	M [T]←AC
LOAD H	AC←M [H]
MUL K	AC←AC*M [K]
ADD G	AC←AC+M [G]
STORE T1	M [T1]←AC
LOAD T	AC←M [T]
DIV T1	AC←AC/M [T1]
STORE X	M[X]←AC

d) Zero address instructions:

RPN: AB-CDE*F-*+GHK*+/

PUSH A	TOS←A
PUSH B	TOS←B
SUB	TOS←(A-B)
PUSH C	TOS←C
PUSH D	TOS←D
PUSH E	TOS←E
MUL	TOS← (D*E)
PUSH F	TOS← F
SUB	TOS←((D*E)-F)
MUL	TOS←C*((D*E)-F)
ADD	TOS←((A-B)+ C*((D*E)-F)
PUSH G	TOS← G
PUSH H	TOS←H
PUSH K	TOS←K
MUL	TOS←(H*K)
ADD	TOS←G+(H*K)
DIV	TOS←((A-B)+ C*((D*E)-F)/(G+(H*K))
POP X	M[X]←TOS

3. An instruction at address 021 in the basic computer has I = 0, an operation code of the AND instruction, and an address part equal to 083 (all numbers are in hexadecimal). The memory word at address 083 contains the operand B8F2 and the content of AC is A937. Go over the instruction cycle and determine the contents of the following registers at the end of the execute phase: PC, AR, DR, AC, and IR. Repeat the problem six more times starting with an operation code of another memory-reference instruction.

	PC	AR	DR	AC	IR
Initial	021	—	—	A937	—
AND	022	083	B8F2	A832	0083
ADD	022	083	B8F2	6229	1083
LDA	022	083	B8F2	B8F2	2083
STA	022	083	—	A937	3083
BUN	083	083	—	A937	4083
BSA	084	084	—	A937	5083
ISZ	022	083	B8F3	A937	6083

4. The content of PC in the basic computer is 3AF (all numbers are in hexadecimal). The content of AC is 7EC3. The content of memory at address 3AF is 932E. The content of memory at address 32E is 09AC. The content of memory at address 9AC is 8B9F.

   a. What is the instruction that will be fetched and executed next?

   b. Show the binary operation that will be performed in the AC when the instruction is executed.

   c. Give the contents of registers PC, AR, DR, AC, and IR in hexadecimal and the values of E, I, and the sequence counter SC in binary at the end of the instruction cycle.

3AF	932E
32E	09AC
9AC	8B9F

AC = 7EC3

(a) 9 = (1001)

1 001

I=1 ADD

ADD @32E → AC←AC+ M[M[32E]]

7EC3+8B9F

b) AC = 7EC3 (ADD)

DR = 8B9F

0A62 E=1

c) PC = 3AF + 1 = 3BO IR = 932E

AR = 7AC E = 1

DR = 8B9F I = 1

AC = 0A62 SC = 0000

5. Convert the following numerical arithmetic expression into reverse Polish notation and show the stack operations for evaluating the numerical result.

   (3 + 4)*[10*(2 + 6) + 8]

   RPN: 3 4+10 2 6 +*8+*

STACK											
						6					
					2	2	8		8		
		4		10	10	10	10	80	80	88	
	3	3	7	7	7	7	7	7	7	7	616

OPERATION	PUSH(3)	PUSH(4)	ADD	PUSH(10)	PUSH(2)	PUSH(6)	ADD	MUL	PUSH(8)	ADD	MUL

6. The memory unit of a computer has 256K words of 32 bits each. The computer has an instruction format with four fields: an operation code field, a mode field to specify one of seven addressing modes, a register address field to specify one of 60 processor registers, and a memory address. Specify the instruction format and the number of bits in each field if the in instruction is in one memory word.

   $256\ K = 2^8 \times 2^{10} = 2^{18}$

op code	Mode	Register	Address

   5 3 6 18 = 32

   Address = 18 bits
   Mode = 3 bits
   Register = 6 bits
   27 bits
   op code 5 bits
   32 bits

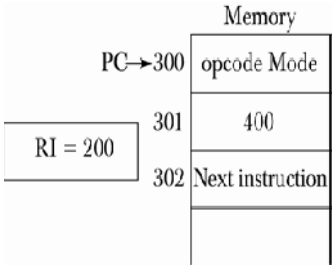
7. A relative mode branch type of instruction is stored in memory at an address equivalent to decimal 750. The branch is made to an address equivalent to decimal 500.
   a. What should be the value of the relative address field of the instruction (in decimal)?
   b. Determine the relative address value in binary using 12 bits. (Why must the number be in 2's complement?)
   c. Determine the binary value in PC after the fetch phase and calculate the binary value of 500. Then show that the binary value in PC plus the relative address calculated in part (b) is equal to the binary value of 500.

   (a) Relative address = 500 – 751 = – 251
   (b) 251 = 000011111011; – 251 = 111100000101
   (c) PC = 751 = 001011101111; 500 = 000111110100
   PC = 751 = 001011101111
   RA = – 251 = +111100000101
   EA = 500 = 000111110100

8. An instruction is stored at location 300 with its address field at location 301. The address field has the value 400. A processor register R1 contains the number 200. Evaluate the effective address if the addressing mode of the instruction is (a) direct; (b) immediate; (c) relative; (d) register indirect; (e) index with R1 as the index register.

(a)direct addressing:

Direct addressing means that the address field contains the address of memory location the instruction is supposed to work with (where an operand "resides").
Effective address would therefore be 400.



(b) immediate addressing
Immediate addressing means that the address field contains the operand itself.
Effective address would therefore be 301.

(c) relative addressing
Relative addressing means that the address field contains offset to be added to the program counter to address a memory location of the operand.
Effective address would therefore be 302 + 400 = 702.

(d) register indirect addressing
Register indirect addressing means that the address of an operand is in the register. The address field in this case contains just another operand.
Effective address would therefore be in R1 = 200.

(e) indexed addressing with R1 as index register
In indexed absolute addressing the effective address is calculated by taking the contents of the address field and adding the contents of the index register.
Effective address would therefore be 400 + R1 = 400 + 200 = 600.