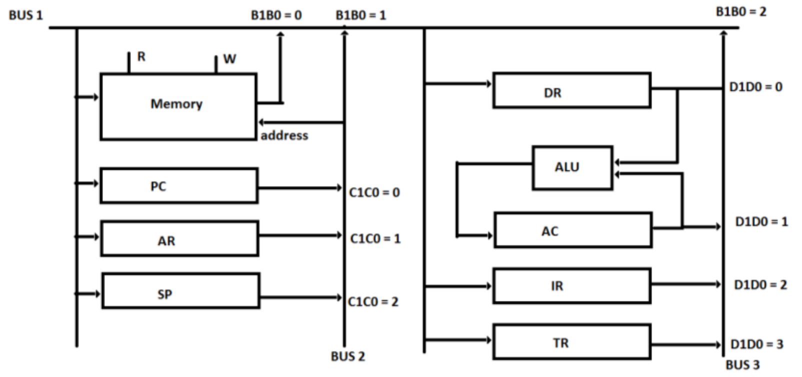
A, B, C and D should be solved according to the following architecture and information.



There are 3 16-bit common busses (BUS1, BUS2 and BUS3). BUS2 and BUS3 are connected to BUS1. All register except AC, can be loaded from BUS1.

PSH: Instruction decrements the value of the stack register (SP) by ones, and then write the content into the memory cell.

PULL: Instruction loads a value into DR from the memory cell whose address be pointed by SP, and then increments the value of the stack register (SP) by one.

15	14 12	11 0
I	Opcode	Address

For Part C use the following information:

The hardware based control unit, the opcode bits from IR is connected an opcode decoder and output signals from K0 to K13.

For Part D use the following information:

The microinstruction format of the software based control unit is as follows:

3	3	3	2	2	7
F1	F2	F3	CD	BR	AD

F1, F2, F3: Microoperation fields
CD: Condition for branching
BR: Branch field
AD: Address field

Microoperations for F1, F2 and F3 are gives as following:

F1	Microoperation	Symbol
000	None	NOP
001	AC ← AC + DR	ADD
010	AC ← 0	CLRAC
011	AC ← AC + 1	INCAC
100	AC ← DR	DRTAC
101	AC ← DR(0-10)	DRTAR
110	AC ← PC	PCTAR
111	M[AR] ← DR	WRITE

F2	Microoperation	Symbol
000	None	NOP
001	AC ← AC - DR	SUB
010	SP ← SP + 1	INCSP
011	SP ← SP - 1	DECSP
100	DR ← M[AR]	READ
101	DR ← AC	ACTDR
110	DR ← DR + 1	INCDR
111	DR(0-10) ← PC	PCTDR

F3	Microoperation	Symbol
000	None	NOP
001	AC ← AC ⊕ DR	ADD
010	AC ← $\overline{AC}$	CLRAC
011	PC ← SP	SPTPC
100	SP ← AR	ARTSP
101	PC ← PC + 1	INCPC
110	PC ← AR	ARTPC
111	AR ← SP	SPTAR

Consider for conditional branches (BR) and condition field (status bit) CD as follows:

CD	Condition	Symbol	Comments
00	Always = 1	U	Unconditional branch
01	DR(15)	I	Indirect address bit
10	AC(15)	S	Sign bit of AC
11	AC = 0	Z	Zero value in AC

BR	Symbol	Function

00	JMP	CAR ← AD if condition = 1
		CAR ← CAR + 1 if condition = 0

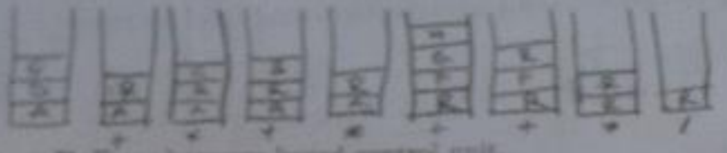
01	CALL	CAR ← AD, SBR ← CAR + 1 if condition = 1
		CAR ← CAR + 1 if condition = 0

10	RET	CAR ← SBR (Return from subroutine)
11	MAP	CAR(2-5) ← DR(11-14), CAR(0,1,6) ← 0

Part A (a) (2pts) Convert the following arithmetic expression from infix to reverse Polish notation: $\frac{A*(B+C*(D+E))}{F*(G+H)}$

A DE + C * B + * FGH + * /

(b) (2pts) What is typically pushed to the stack during a routine (function) call? R is result symbol of operation



* Operands are pushed until see any operator.

Part B For a harware-based control unit,

(a) (5 pts) Write the fetch and decode cycles using Register Transfer Language (RTL).

T_0 : IR ← M[PC] , PC ← PC + 1

T_1 : AR ← IR(0 - 11) , I ← IR(15) , Decode (IR(11 - 14))

(b) (5 pts) Write the values of relevant control signals (B_1B_0, C_1C_0, D_1D_0, LD, INC, CLR, R, W etc.) at each cycle of fetch and decode.

Fetch : B_1B_0 = 00 T_0 = 1

C_1C_0 = 00

LD(IR) = 1

INC(PC) = 1

R = 1

Decode : D_1D_0 = 10 T_1 = 1

B_1B_0 = 10

LD(AR) = 1

Part C For a hardware based control unit, each register have control signal for load (LD), increment (INC) and clear (CLR). However, they do not have any control signal for decrementing.

Hint: Value of the register can be decremented to transfer its value to AC, clearing and increasing the value of the DR such that DR ← 1, complementing the value of the DR and adding DR to AC. However, the content of the DR should be preserved and then should be transferred to the stack memory.

(a) (5 pts) Write the stack operation of PSH (Opcode=101) and PULL (Opcode = 110) using RTL.

(b) (5 pts) Write the value of control signals ($C_1C_0, D_1D_0, LD, INC, CLR, R, W\ etc.$) for each cycle.

Part D (10 pts) Write a symbolic microprogram for *PSH* (Opcode=101) and *PULL* (Opcode=110) using the information given above.
Beware of the changed mapping algorithm: $CAR[3-5] \leftarrow IR[12-14], CAR[0,1,\overset{,2}{6}] \leftarrow 0$
It is possible to decrement SP register by using *DECSP* (for this question only).
There is no need to convert the microprogram into hexadecimal machine codes.
There is no need to implement fetch& decode cycles. Just assume that there are subroutines for fetch (FETCH), and indirect address decoding (INDRCT) available for your convenience and use these subroutines.

$\underbrace{0(101)000}_{\text{mapping}} \rightarrow (40)_{10}$

$\underbrace{0110000}_{\text{mapping}} \rightarrow (48)_{10}$

```
        ORG 40
PSH :   NOP    NOP    NOP     I   CALL   INDRCT
        NOP    DECSP  NOP     U   JMP    NEXT
        NOP    NOP    SPTAR   U   JMP    NEXT
                              U   JMP    FETCH
        WRITE  NOP    NOP     U   JMP    FETCH
        NOP    NOP    NOP     U   JMP    FETCH
        NOP    NOP    NOP     U   JMP    FETCH
        NOP    NOP    NOP     U   JMP    FETCH
        NOP    NOP    NOP
-------------------------------------------------
        ORG 48
PULL :  NOP    NOP    NOP     I   CALL   INDRCT
                              U   JMP    NEXT
        NOP    NOP    SPTAR   U   JMP    NEXT
        NOP    READ   NOP     U   JMP    FETCH
        NOP    INCSP  NOP     U   JMP    FETCH
               NOP            U   JMP    ''
               NOP            U   JMP    ''
               NOP            U   JMP    ''
               NOP
```

For any unwanted circumstances for protection Like try-catch block