June 30, 2010

COMPUTER ENGINEERING DEPARTMENT

COE 205

COMPUTER ORGANIZATION & ASSEMBLY PROGRAMMING

Final Exam

First Semester (091)

Time: 7:30 -10:00 AM

Student Name : _KEY_____

Student ID. : _____

Question	Max Points	Score
Q1	15	
Q2	8	
Q3	22	
Q4	40	
Q5	15	
Total	100	

Dr. Aiman El-Maleh

[15 Points]

(Q1) Fill the blank in each of the following:

(1) When an interrupt occurs, the CPU suspends its execution of the current program, and transfers control to an Interrupt Service Routine (ISR) that will provide the requested service by the interrupt.

(2) Three main differences between interrupts and procedures include: Interrupts can be initiated by both software and hardware while procedures can be initiated only by software, Interrupt mechanism provides an efficient way to handle unanticipated events, Interrupts are identified by numbers while procedures are identified by names.

(3) Interrupts initiated by the keyboard belong to interrupts of the following type: Maskable Hardware Interrupts.

(4) Interrupts initiated due to division by 0 belong to interrupts of the following type: Processor Interrupts of type Fault.

(5) Interrupts initiated due to execution of a break point in debugging mode belong to interrupts of the following type: Software Interrupts.

(6) Maskable Hardware interrupts can be masked.

(7) The Interrupt Flag (IF) controls whether maskable interrupts are delayed or not.

(8) The CPU identifies the interrupting hardware through its number placed on the data bus.

(9) An example of an interrupt Fault is divide error or segment-not-present.

(10) In real mode, when the CPU executes the instruction INT 0Fh, it performs following actions:

1. Push FLAGS register onto the stack,
2. Clear interrupt and trap flags to disable further interrupts
3. Push CS register onto the stack,
4. Push IP register onto the stack,
5. Load CS register with the 16-bit at memory address from IDT at offset 3E
6. Load IP register with the 16-bit from IDT at offset 3C

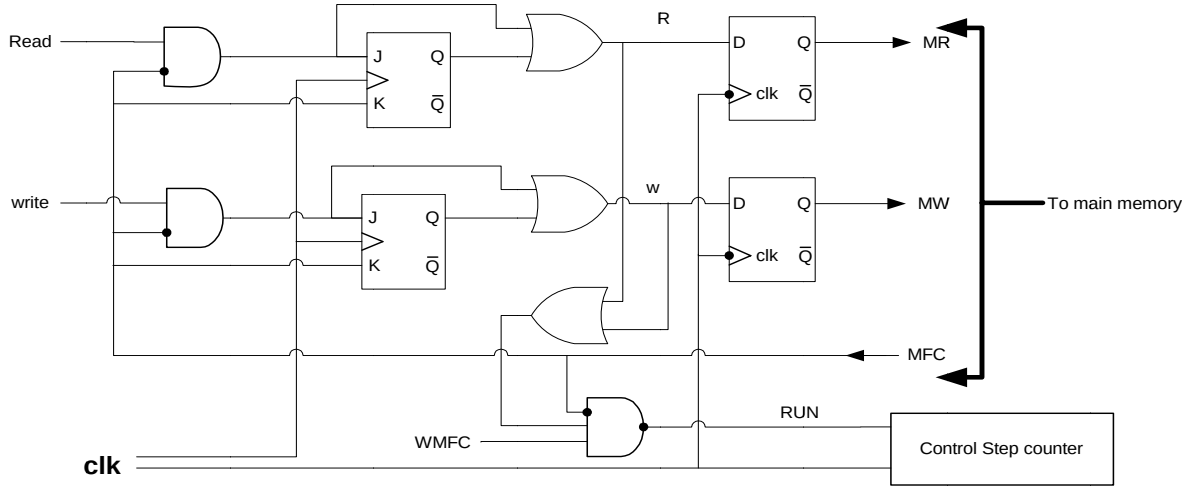
(11) To send control back to the interrupted program, the interrupt service routine should end up with the instruction IRET.

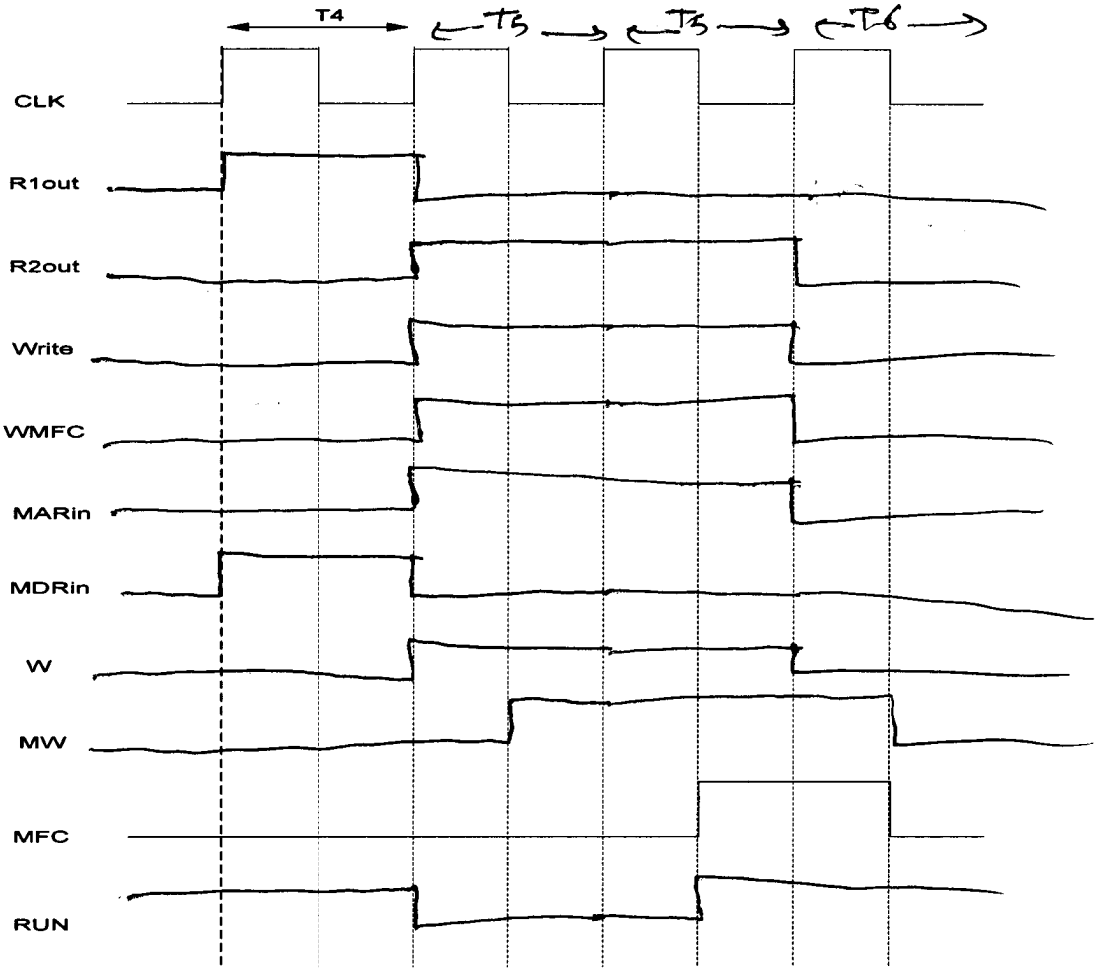
[8 Points]

(Q2) Given the CPU-Memory interface circuit shown below, complete the given timing diagram assuming the following given control sequnce as shown below:

T4 R1out, MDRin,
T5 R2out, MARin, Write, WMFC

Note that the control unit is assumed to be rising-edge triggered.





[22 Points]

(Q3) It is required to design an 8-bit CPU the has four 8-bit registers, namely R0, R1, R2, and R3. Assume that the data bus and address bus are 8-bits. The CPU has 8-bit instructions with the following foramt:

4 bits	2 bits	2 bits
OPCODE	Rdst	Rsrc

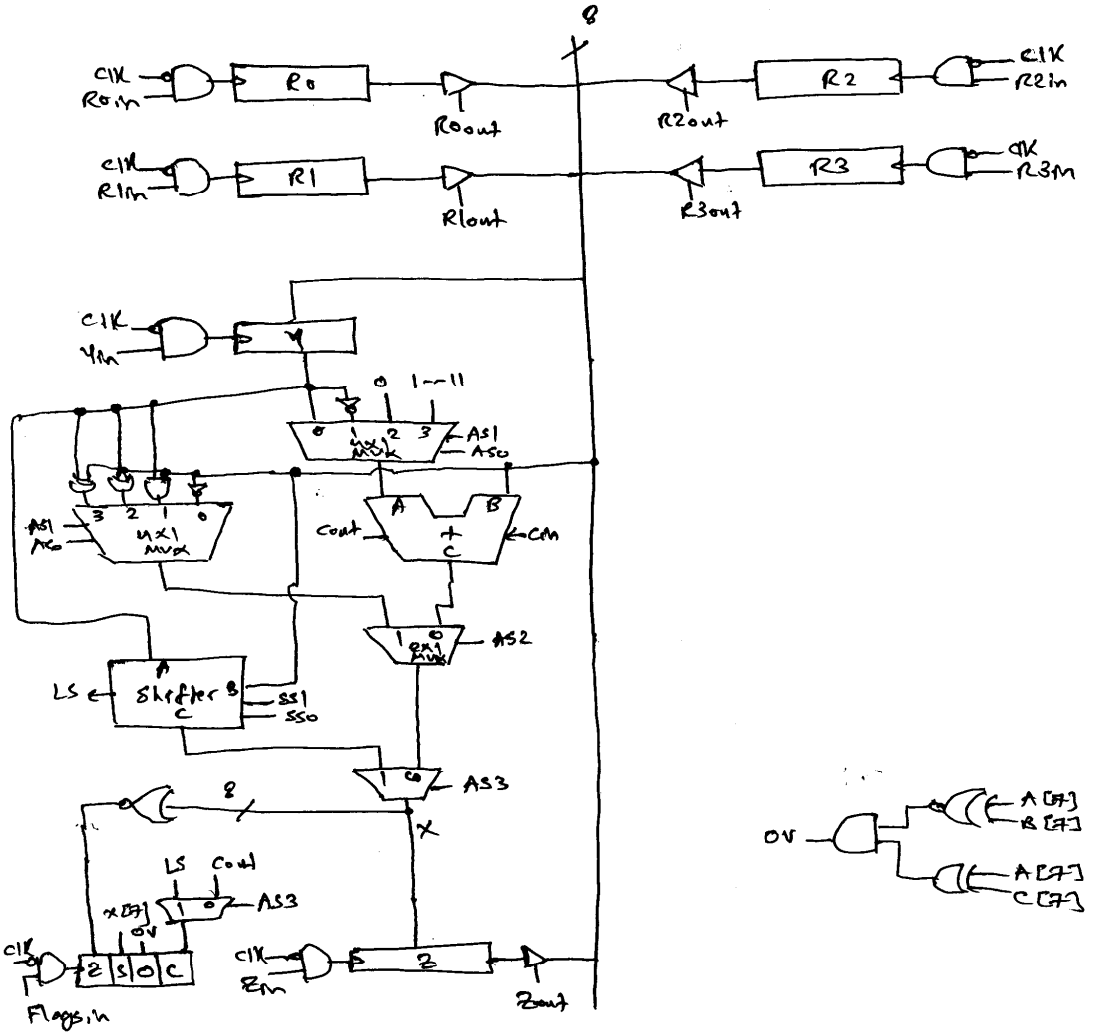
The CPU has the following set of instructions:

Instruction	Register Transfer
MOV Rdst, Rsrc	Rdst ← Rsrc
XCHG Rdst, Rsrc	Rdst ← Rsrc; Rsrc ← Rdst
ADD Rdst, Rsrc	Rdst ← Rdst + Rsrc
SUB Rdst, Rsrc	Rdst ← Rdst – Rsrc
INC Rdst	Rdst ← Rdst + 1
DEC Rdst	Rdst ← Rdst – 1
SHL Rdst, Rsrc	Rdst ← shift left Rdst by Rsrc
SHR Rdst, Rsrc	Rdst ← shift right Rdst by Rsrc
ROL Rdst, Rsrc	Rdst ← rotate left Rdst by Rsrc
ROR Rdst, Rsrc	Rdst ← rotate right Rdst by Rsrc
NOT Rdst	Rdst ← NOT Rdst
AND Rdst, Rsrc	Rdst ← Rdst AND Rsrc
OR Rdst, Rsrc	Rdst ← Rdst OR Rsrc
XOR Rdst, Rsrc	Rdst ← Rdst XOR Rsrc

Design a **single-bus data path** for this CPU. Clearly illustrate all design details and all the required control signals. Assume that you can only use the following combinational logic blocks in addition to basic gates like AND, OR, INV, XOR, MUX and Tri-state Buffers:

- An 8-bit **Adder** that has the inputs A[7:0], B[7:0] and Cin, and produces the Sum in C[7:0] and Cout.
- An 8-bit **Shifter** that has the inputs A[7:0] for specifiying the input to be shifted, B[2:0] to specify the amount of shift to be performed and two select lines SS1 and SS0 to determine the required operation as follows: SS1=0, SS0=0 to shift left, SS1=0, SS0=1 to shift right, SS1=1, SS0=0 to rotate left, and SS1=1, SS0=1 to rotate right. The resulting operand is produced on the output C[7:0]. The last bit shifted or rotated is produced on the output signal LS.

Assume that the CPU has a flags register consisting of 4 flags: sign flag, carry flag, overflow flag, and zero flag. Assume that all instructions affect the 4 flags except, the mov and xchg instructions, similar to 8086 instructions. Design the data path to minimize the number of clock cycles needed for the execution of the specified instructions.


B
CLK
R0in
R0
R0out
R2out
R2
CLK
R2in
CLK
R1in
R1
R1out
R3out
R3
CLK
R3in
CLK
Yin
Y
AS1
AS0
4x1 MUX
A
B
Cout
+
C
Cin
AS2
LS
Shifter
SS1
SS0
AS3
X
LS
Cout
AS3
OV
CLK
Flagsin
Z
S
O
C
CLK
Zin
Z
Zout
OV
A[7]
B[7]
A[7]
C[7]

[40 Points]

(Q4) Assume that a CPU has **16 instructions** with an opcode of 4 bits, seven **16-bit** general purpose registers namely AX, BX, CX, DX, SI, DI and SP, a **16-bit data bus**, and a **16-bit address bus**. Assume that all the instructions are 16-bit. The CPU has an **Arithmetic and Logic Unit (ALU)** with inputs A and B and output C, that can perform any of the following functions shown below based on the three selection lines AS2, AS1, and AS0:

AS2 AS1 AS0	Operation
000	C=A+B
001	C=A-B
010	C=A+1
011	C=A-1
100	C=B
101	C=A+2
110	C=A-2
111	C=NOT A

The CPU has also a **Shift Unit** that can perform shifting as shown below based on the two selection lines SS1 and SS0:

SS1 SS0	Operation
00	No shift
01	W=Shift logic right(X) by M bits
10	W=Shift logic left(X) by M bits
11	W=Shift arithmetic right(X) by M bits

Assume that the IR, PC, MAR, and MDR registers are also **16-bit regiters**. Also assume that the instructions for this processor follow the syntax and semantics and addressing modes of the 8086 processor. Also assume that registers play the same role as assumed in the 8086 processor.

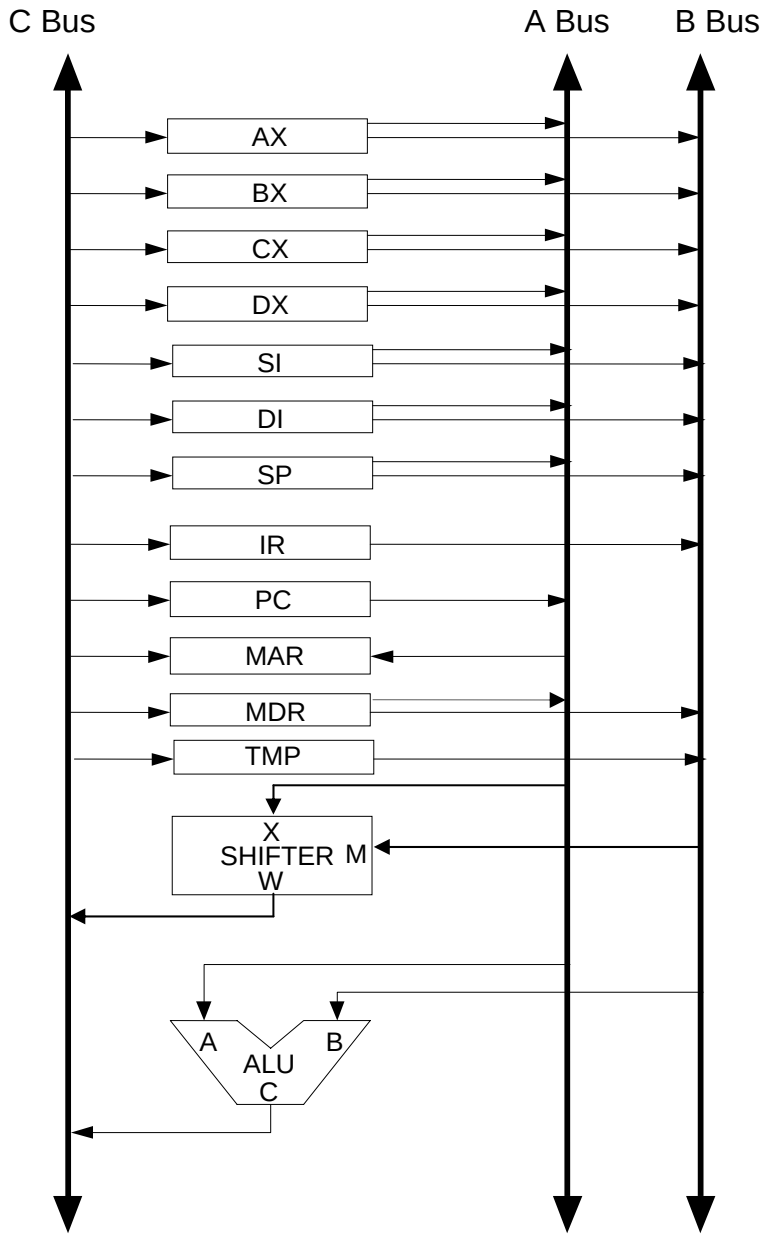
Assume that the data path of this processor is implemented using a **three-bus architecture** as shown in the next page.

(i) Determine the **maximum speed** at which this design will work given the following delay parameters:

Name	Parameter	Delay
Tri-state buffer propagation time	t_q	50 ps
Bus propagation time	t_{bp}	150 ps
ALU delay	t_{ALU}	400 ps
Shifter delay	$t_{shifter}$	250 ps
Flip-flop propagation time	t_{ff}	150 ps
Flip-flop setup time	t_{su}	100 ps
Flip-flop hold time	t_h	50 ps

The clock speed should be greater than the longest delay in the data path

- ⇨ $T \geq tg+tbp+talu+tg+tbp+tsu+tff = 50+150+400+50+150+100+150=1050$ ps
- ⇨ $F=1/T \leq 1/1050$ ps = 952.4 MHZ



Data Path Design

(ii) Write the minimum number of control steps required for fetching an instruction from memory.

T1 PCout, MARin,A, AS2, AS0 ALU(C=A+2), ALUout, PCin, Read, WMFC

T2 MDRout,B, AS2 ALU(C=B), ALUout, IRin

(iii) Write the minimum number of control steps required for the execution of each of the following instructions:

a. **MOV [BX], 20**

T3 BXout,A, MARin,A, IRout, AS2 ALU(C=B), ALUout, MDRin, Write,

WMFC

T4 End

b. **SUB AX, [BX]**

T3 BXout,A, MARin,A, Read, WMFC

T4 AXout,A, MDRout,B, AS0 ALU(C=A-B), ALUout, AXin, End

c. **CALL Next**

T3 SPout,A, AS2, AS1 ALU(C=A-2), ALUout, Spin, MARin,C

T4 PCout, Shifter(W=X) , Wout, MDRin, Write, WMFC

T5 PCout, IRout, ALU(C=A+B), ALUout, PCin, End

d. **RET**

T3 SPout,A, MARin,A, Read, AS2, AS0 ALU(C=A+2), ALUout, Spin, WMFC
T4 MDRout,B, AS2 ALU(C=B), ALUout, PCin, End

e. **LOOPNE Next**

T3 CXout,A, AS1, AS0 ALU(C=A-1), ALUout, CXin, if (ZF=1) End
T4 PCout, IRout, ALU(C=A+B), ALUout, if (CX≠ 0) PCin, End

f. **JNE Next**

T3 PCout, IRout, ALU(C=A+B), ALUout, if (ZF=0) PCin, End

g. **SHL AX, CX**

T3 AXout,A, CXout,B, SS1 Shifter(shift left), Wout, AXin, End

(iv) Based on the fetch and execution control sequence of the seven instructions given in this question, show the logic equation required for generating the signal **PCin.**

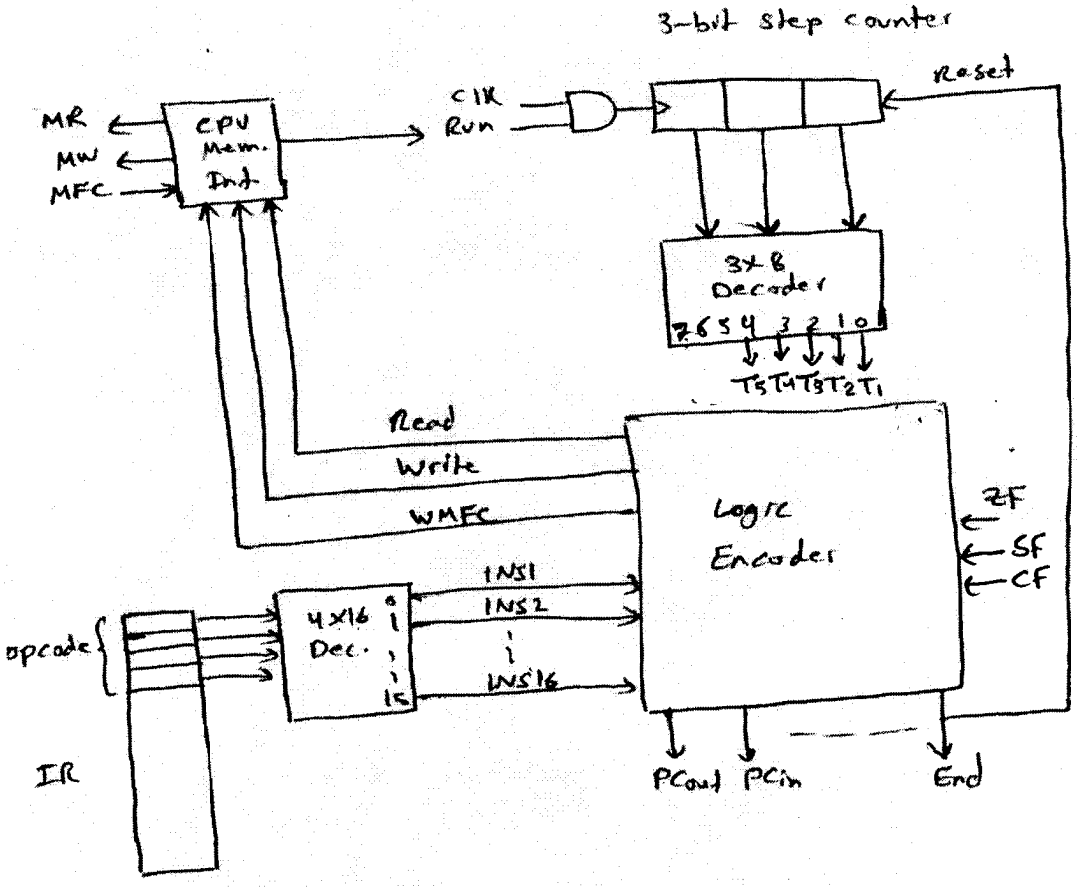
PCin = T1 + T5.CALL + T4. RET+T4.LOOPNE.CXZ' +T3.JNE.ZF'

Note that it is assumed here that the signal CXZ is 1 if the content of register CX is 0.

[15 Points]

(Q5) Using the CPU described in the previous question and the given data path, it is required to implement the control unit for this CPU using both hardwired and microprogrammed approach. Assume that the number of signals that have to be generated by the control unit to control the data path and the memory interface circuit for this CPU is **35** signals.

(i) Assume that the maximum number of control steps required for the fetch and execution of any of the 16 instructions is **5 control steps**. Show the block diagram of the **hardwired control unit** organization for this CPU indicating all the necessary components and signals. Assume that the flags needed for the execution of the instruction are the zero, sign and carry flags. Clearly indicate the size of the various components.



(ii) Assume that the control unit for this CPU is to be implemented using a microprogrammed approach and that the number of different control words to be stored in the control store for all the microroutines is **24 CWs**. Also, assume that the microprogrammed control unit design should support unconditional branch (UBR), branch on zero (BRZ), branch on not zero (BRNZ), branch on negative (BRN), branch on not negative (BRNN), branch on carry (BRC) and branch on no carry (BRNC).

a. Show the format and size of the control word using **horizontal control store**. What is the **size of the control store** needed?

Since the number of CWs = 24, the Ubranch address = 5 bits

CW Format:

MS1, MS0, UBR, BRZ, BRNZ, BRN, BRNN, BRC, BRNC, 35 control signals, 5 bits for Ubranch address

⇒ 49 bits

Size of control store = 24 x 49 bits

b. Show the block diagram of the **microprogrammed control unit** organization for this CPU indicating all the necessary components and signals. Clearly indicate the size of the various components.