Jan. 26, 2011

COMPUTER ENGINEERING DEPARTMENT

COE 205

COMPUTER ORGANIZATION & ASSEMBLY PROGRAMMING

Final Exam

First Semester (101)

Time: 7:30 -10:00 AM

Student Name : KEY_____

Student ID. : _____

Question	Max Points	Score
Q1	16	
Q2	8	
Q3	22	
Q4	40	
Q5	14	
Total	100	

Dr. Aiman El-Maleh

[16 Points]

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

(1) 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.

(2) The three main types of processor interrupts are: traps, faults and aborts.

(3) Hardware interrupts initiated at the NMI pin are non-maskable while those initiated at the INTR pin are maskable.

(4) The interrupt flag controls whether maskable interrupts are delayed or not.

(5) Segment-not-present error is an example of a processor fault interrupt.

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

(7) In protected mode, the address of the interrupt service routine of interrupt 15h is obatined from offset 15h*8=A8h within interrupt descriptor table (IDT).

(8) In protected mode, when an interrupt occurs, the following actions are taken by the CPU:

1. Push EFLAGS register onto the stack,
2. Clear interrupt and trap flags to disable further interrupts
3. Push CS register onto the stack,
4. Push EIP register onto the stack,
5. Load CS register with the 32-bit at memory address from IDT at offset n*8+4.
6. Load EIP register with the 32-bit from IDT at offset n*8.

.

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

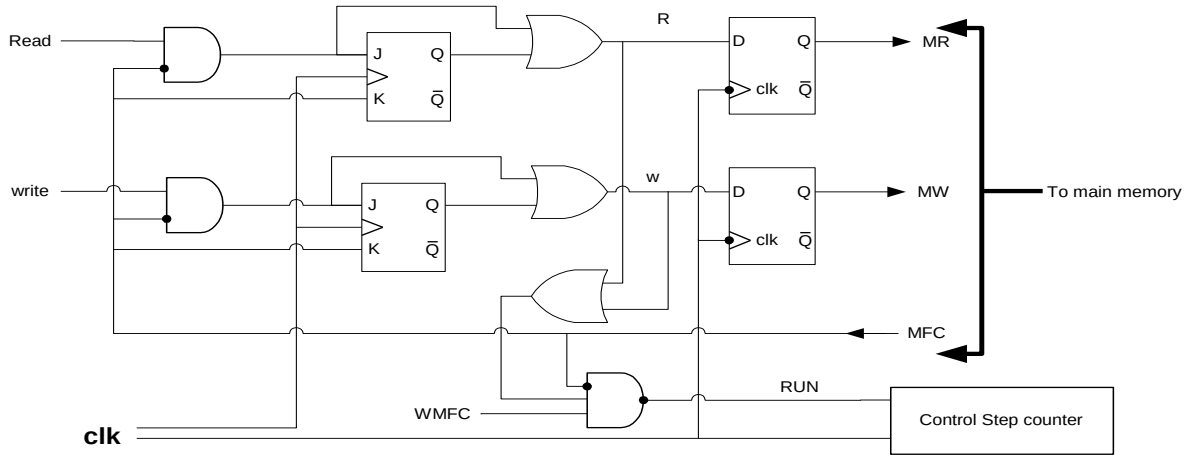
(10) If the trap flag is set, the CPU automatically generates a type 1 (single step) interrupt after executing each instruction.

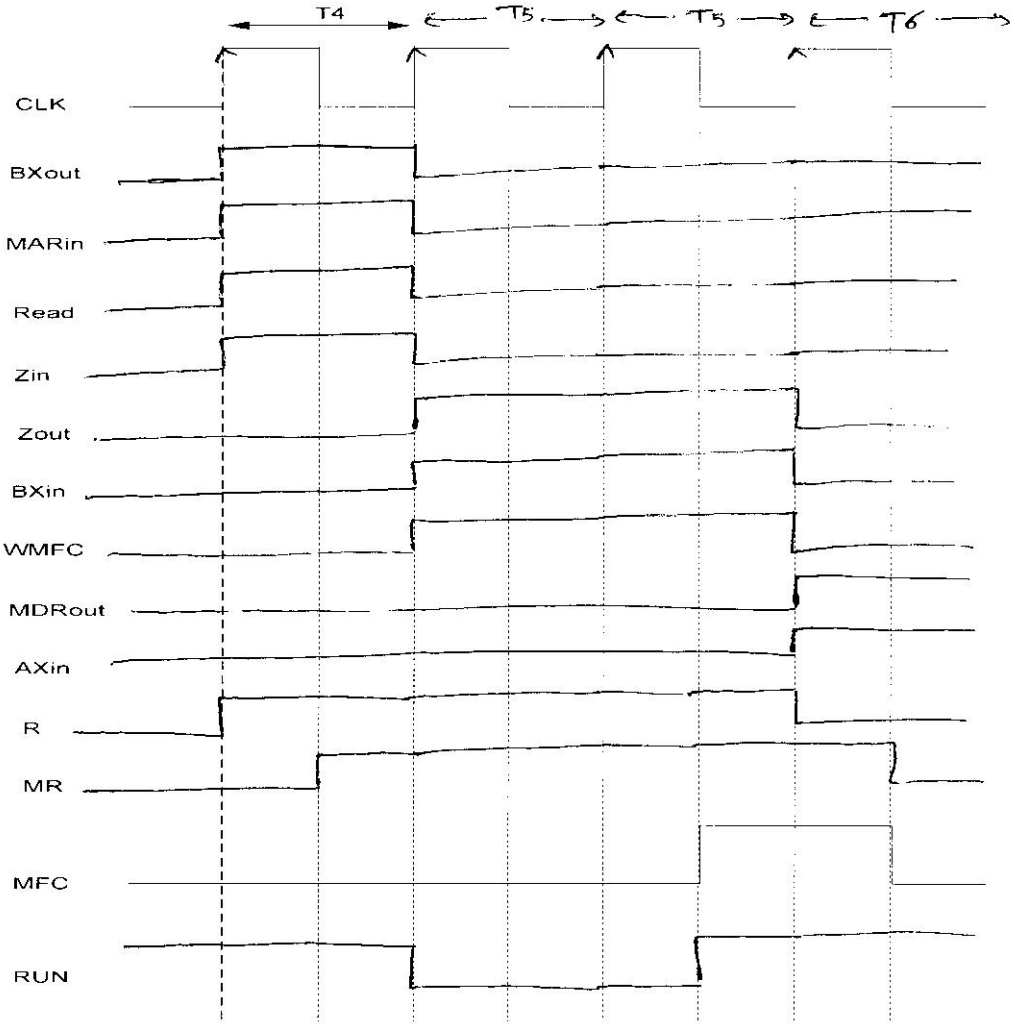
[8 Points]

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

T4 BXout, MARin, Read, Zin
T5 Zout, BXin, WMFC
T6 MDRout, AXin

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

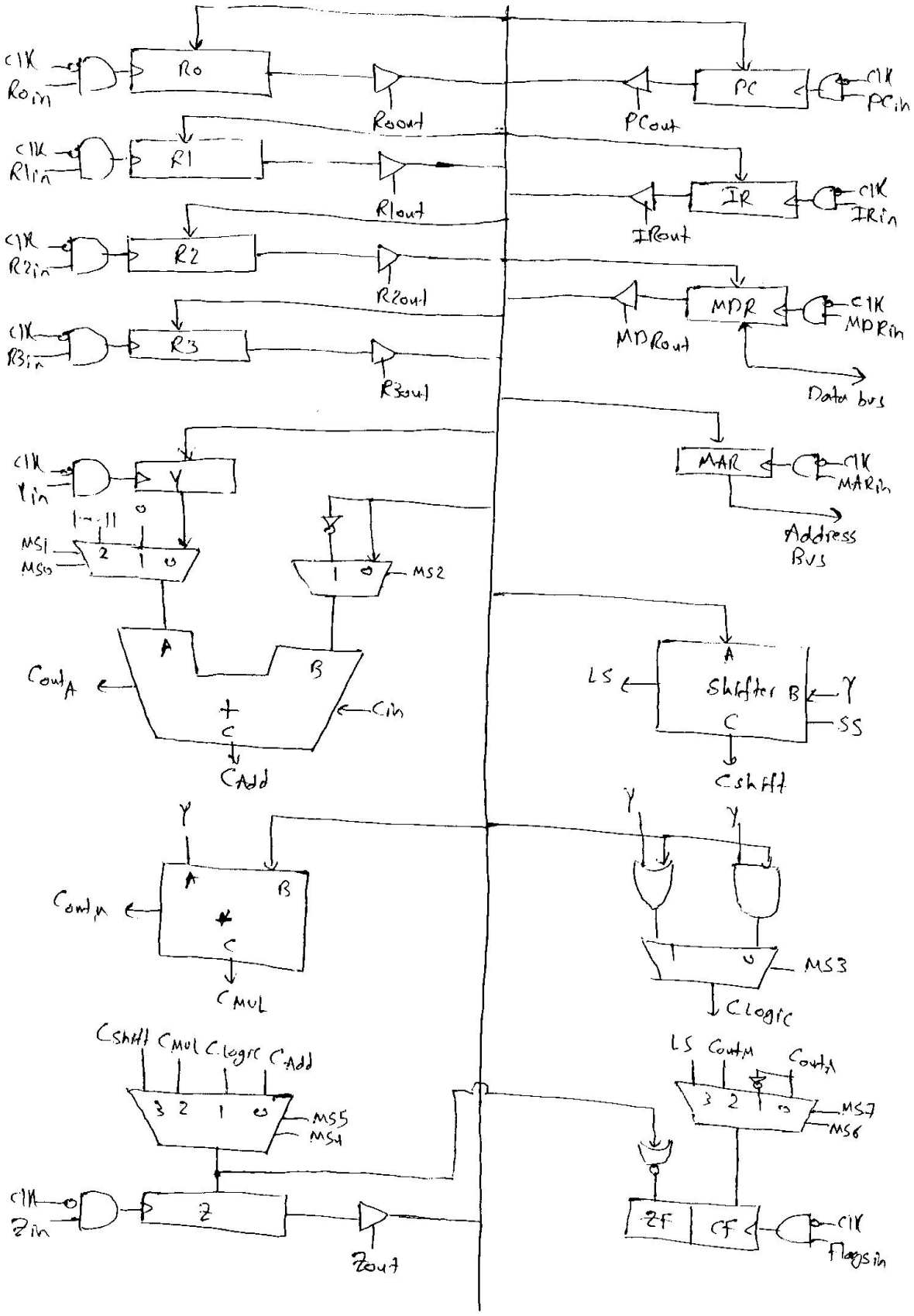
The CPU has the following set of instructions:

Instruction	Register Transfer	Effect on Flags
MOV Rdst, Rsrc	Rdst ← Rsrc	**N**
XCHG Rdst, Rsrc	Rdst ← Rsrc; Rsrc ← Rdst	**N**
ADD Rdst, Rsrc	Rdst ← Rdst + Rsrc	**Y**
SUB Rdst, Rsrc	Rdst ← Rdst – Rsrc	**Y**
INC Rdst	Rdst ← Rdst + 1	**Y**
DEC Rdst	Rdst ← Rdst – 1	**Y**
SHL Rdst, Rsrc	Rdst ← shift left Rdst by Rsrc	**Y**
SHR Rdst, Rsrc	Rdst ← shift right Rdst by Rsrc	**Y**
AND Rdst, Rsrc	Rdst ← Rdst AND Rsrc	**Y**
OR Rdst, Rsrc	Rdst ← Rdst OR Rsrc	**Y**
MUL Rdst, Rsrc	Rdst ← Rdst * Rsrc	**Y**
PUSH Rdst	Rdst is saved on stack	**N**
POP Rdst	Rdst is poped from stack	**N**
JMP Rsrc	IP ← Rsrc	**N**
JE Rsrc	If (ZF) IP ← Rsrc	**N**
JNE Rsrc	If (!ZF) IP ← Rsrc	**N**

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, NOR, 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 **Multiplier** that has the inputs A[7:0], B[7:0], and produces the Product in C[7:0] and Cout. Cout=1 means that the result does not fit in 8-bits.
- 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 one select line SS to determine the required operation as follows: SS=0 to shift left, and SS=1 to shift right. The resulting operand is produced on the output C[7:0]. The last bit shifted is produced on the output signal LS.

Assumat that R3 is the stack pointer. Assume that the CPU has a flags register consisting of 2 flags: Carry Flag (CF) and Zero Flag (ZF). Assume that flags are affected as indicated in the table above. Note that the Carry Flag for SUB and DEC instructions should represent a borrow.


CLK
R0in
R0
R0out
R1in
R1
R1out
R2in
R2
R2out
R3in
R3
R3out
Yin
Y
MS1
MS0
MS2
A
B
+
C
CoutA
Cin
CAdd
PC
PCout
PCin
IR
IRout
IRin
MDR
MDRout
MDRin
Data bus
MAR
MARin
Address Bus
Shifter
LS
SS
CShift
CoutM
MS3
CLogic
CMul
MS5
MS4
Zin
Z
Zout
ZF
CF
Flagsin
MS7
MS6

[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

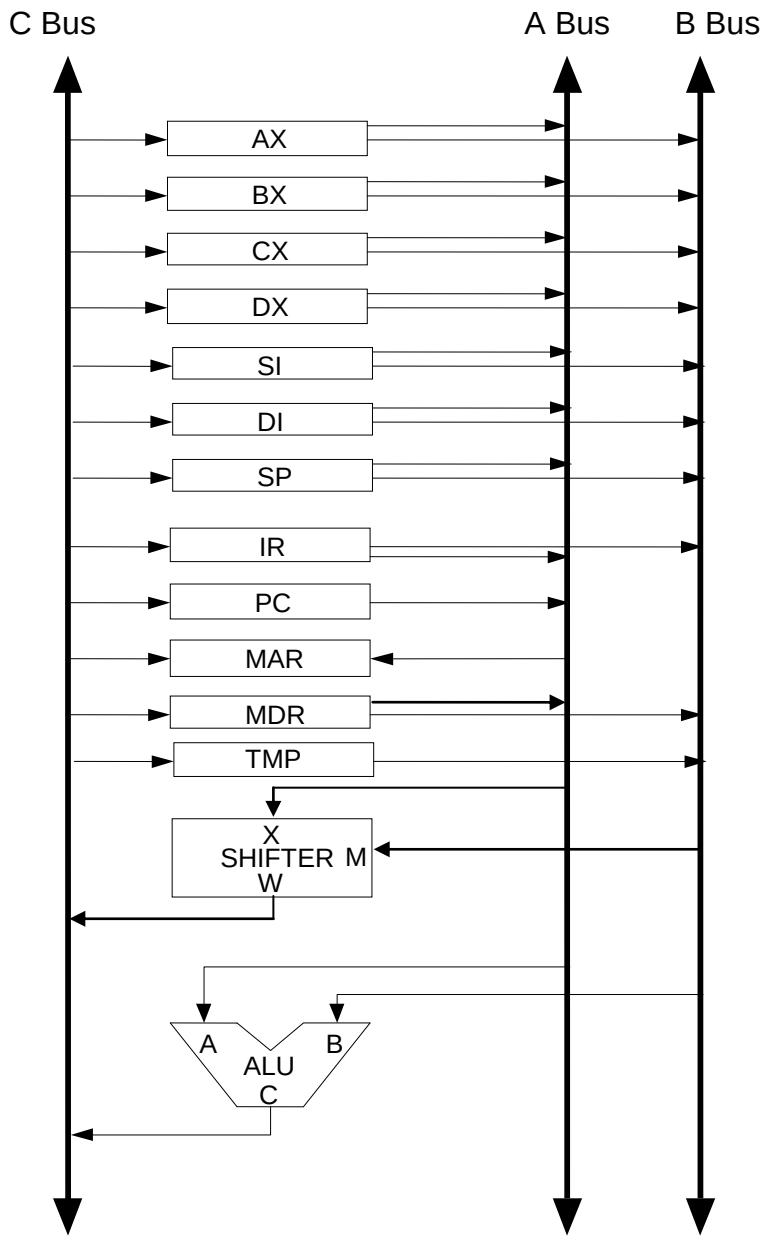
Assume that the IR, PC, MAR, and MDR registers are also **16-bit regiters**. Also assume that the instructions for this processor follow the syntax and semantics and addressing modes of the 8086 processor. Also assume that registers play the same role as assumed in the 8086 processor.

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

(i) 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



Data Path Design

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

a. **XCHG AX, BX**

T3 AXout,B, AS2 ALU(C=B), ALUout, TEMPin

T4 BXout,B, AS2 ALU(C=B), ALUout, AXin

T5 TEMPout, AS2 ALU(C=B), ALUout, BXin, End

b. **ADD I, -10**

T3 IR(offset)out,A, MARin,A, Read, WMFC
T4 MDRout,A, IR(const.)out,B, ALU(C=A+B), ALUout, MDRin, Write, WMFC
T5 End

c. **INC Array[BX]**

T3 IR(offset)out,A, BXout,B, ALU(C=A+B), ALUout, MARin,C, Read, WMFC
T4 MDRout,A, AS1 ALU(C=A+1), ALUout, MDRin, Write, WMFC
T5 End

d. **POP AX**

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

e. **LOOPNE Next**

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

Note that CZ is 1 when the C bus contains 0.

f. **CALL MTest**

T3 SP,out,A, AS2, AS1 ALU(C=A-2), ALUout, Spin, MARin,C
T4 PCout, Shifter(no shift), Wout, MDRin, Write, WMFC
T5 PCout, IRout,B, ALU(C=A+B), ALUout, PCin, End

g. **JZ Next**

T3 PCout, IRout,B, ALU(C=A+B), ALUout, if (ZF=1) PCin, End

h. **SAR AX, BX**

T3 AXout,A, BXout,B, SS1, SS0 Shifter(shift arithmetic right), Wout, AXin, End

(iii) Based on the fetch and execution control sequence of the eight instructions given in this question, show the logic equation required for generating the signals **PCin** and **End.**

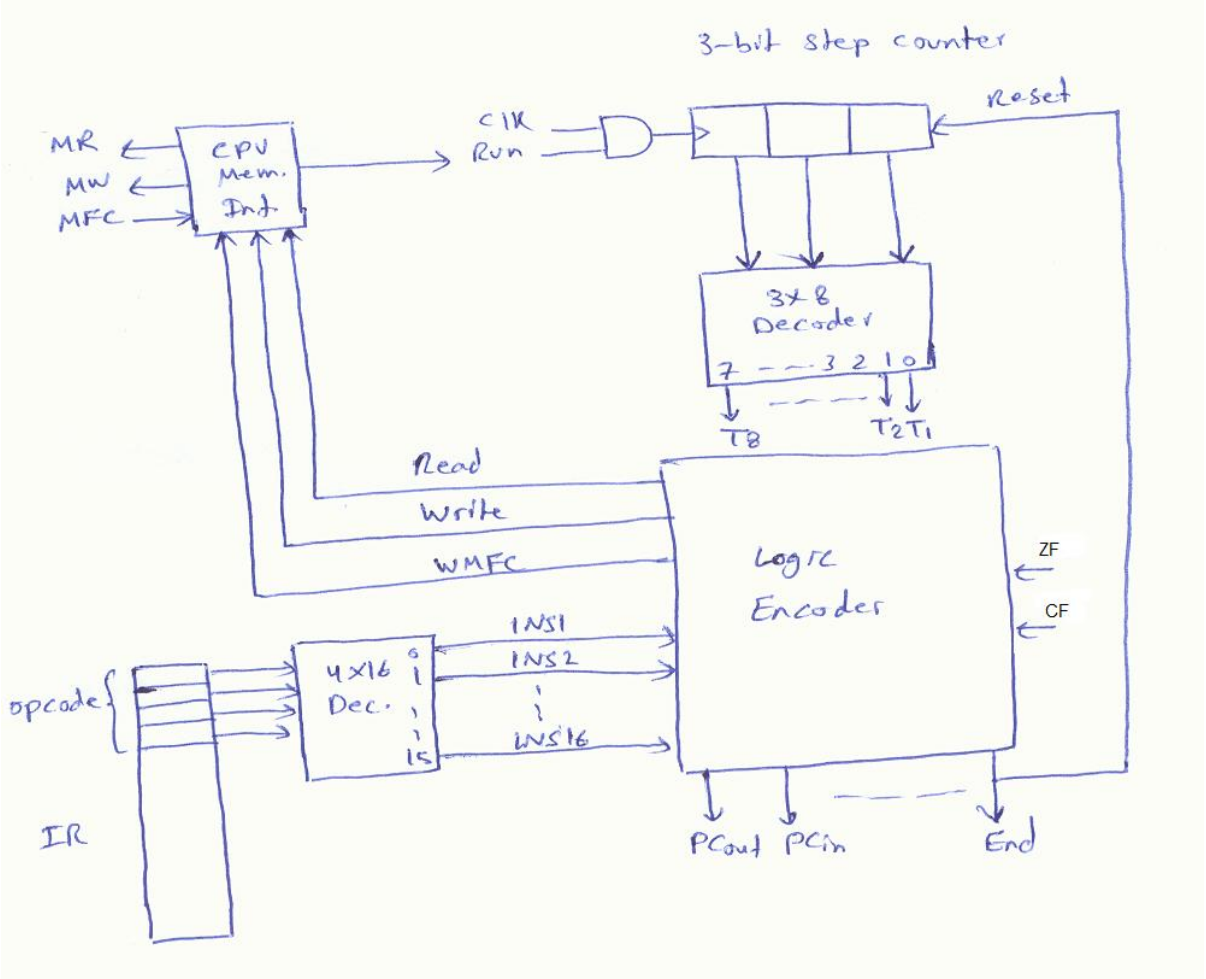
PCin = T1 + T4.LOOPNE + T5.CALL + T3.JZ.ZF

End = T5.XCHG + T5.ADD + T5.INC + T4.POP + T3.(ZF+CZ).LOOPNE + T4.LOOPNE + T5.CALL + T3.JZ + T3.SAR

[14 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 approaches. 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 **30** signals.

(i) Assume that the maximum number of control steps required for the fetch and execution of any of the 16 instructions is **8 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 instructions are the zero 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 **50 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 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=50, the uBranch address=6 bits. Thus, CW format will be: MS1, MS0, UBR, BRZ, BRNZ, BRC, BRNC, 30 control signals, 6 bits for uBranch address = 7+30+6=43 bits.
Size of control store = 50*43=2150 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.