June 24, 2009

COMPUTER ENGINEERING DEPARTMENT

COE 205

COMPUTER ORGANIZATION & ASSEMBLY PROGRAMMING

Final Exam

Second Semester (082)

Time: 7:00 -9:30 PM

Student Name : _KEY____

Student ID.   : ____

Question	Max Points	Score
Q1	18	
Q2	10	
Q3	72	
Total	100	

Dr. Aiman El-Maleh

[18 Points]

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

(1) Interrupts can be initiated by by both software and hardware while procedures can be initiated by software.

(2) Interrupts are identified by numbers.

(3) Interrupt types include software interrupts, hardware interrupts, processor interrupts or exceptions.

(4) Exceptions can be classidfied into faults, traps, aborts.

(5) There are 256 different interrupts.

(6) Interrupt flag controls whether maskable interrupts are delayed or not.

(7) In real mode, the address of the interrupt service routine of interrupt 21h is obatined from offset 4*21h=84h within interrupt descriptor table (IDT).

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

- Push EFLAGS register onto the stack,
- Clear interrupt and trap flags to disable further interrupts
- Push CS register onto the stack,
- Push EIP register onto the stack,
- Load CS register with 16-bit at memory address from IDT
- Load EIP register with 32-bit from IDT

(9) When the IRET instruction is executed in protected mode, the CPU performs the following steps:

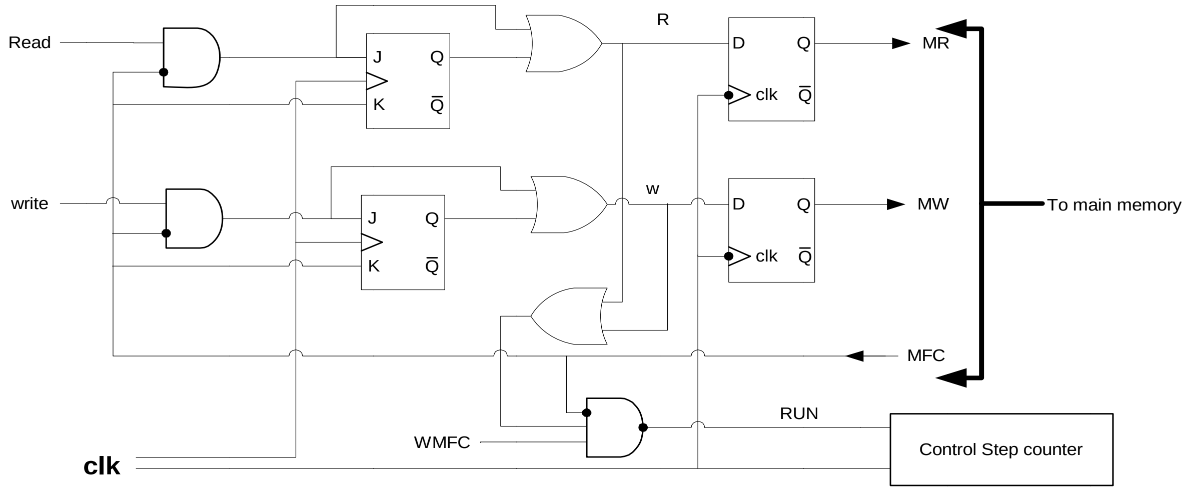
- Pop 32-bit from the top of the stack into the EIP register,
- Pop 16-bit from the top of the stack into the CS register,
- Pop 32-bit from the top of the stack into the EFLAGS register.

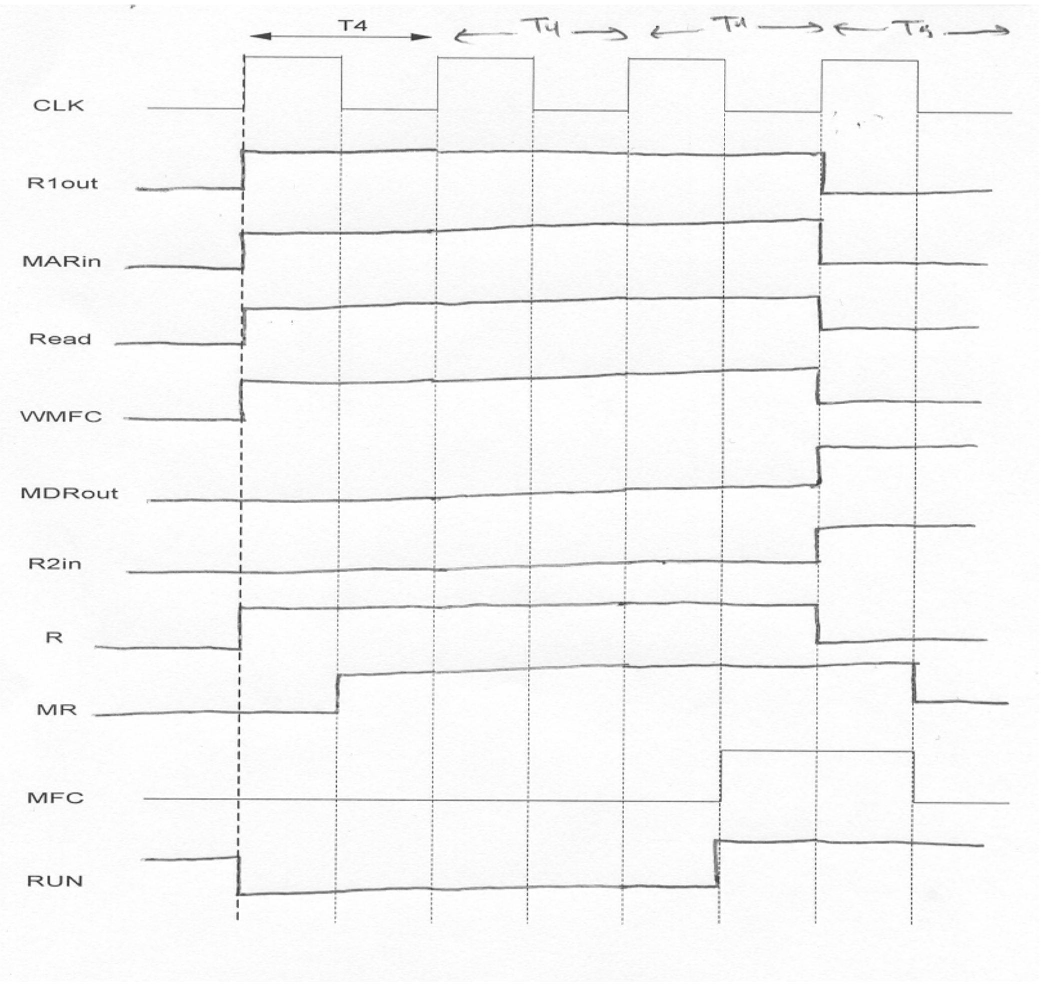
[10 Points]

(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, MARin, READ, WMFC
T5 MDRout, R2in

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





[72 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	Impact on ZF
MOV Rdst, Rsrc	Rdst ← Rsrc	Unchanged
XCHG Rdst, Rsrc	Rdst ← Rsrc; Rsrc ← Rdst	Unchanged
ADD Rdst, Rsrc	Rdst ← Rdst + Rsrc	Changed
SUB Rdst, Rsrc	Rdst ← Rdst – Rsrc	Changed
INC Rdst	Rdst ← Rdst + 1	Changed
DEC Rdst	Rdst ← Rdst – 1	Changed
SHL Rdst, Rsrc	Rdst ← Rdst << Rsrc	Changed
SHR Rdst, Rsrc	Rdst ← Rdst >> Rsrc	Changed
JMP Rdst	IP ← Rdst	Unchanged
JZ Rdst	If (ZF=1) IP ← Rdst	Unchanged
JNZ Rdst	If (ZF=0) IP ← Rdst	Unchanged
LOOP Rdst, Rsrc	Rsrc ← Rsrc-1; If (Rsrc≠ 0) IP ← Rdst	Changed

(i) Design a **three-bus data path** for this CPU. Clearly illustrate all design details and all the required control signals. Assume that the CPU has only a Zero Flag (ZF) affected by the executed instructions as indicated in the above table. 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 ShiftDir to determine whether to shift left (ShiftDir=0) or shift right (ShiftDir=1). The shifted operand is produced on the output C[7:0].

Design the data path to minimize the number of clock cycles needed for the execution of the specified instructions. Show the impact on the ZF in your design.

(ii) Write the minimum number of control steps required for fetching an instruction from memory for this CPU, assuming asynchronous memory transfer.

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

a. XCHG Rdst, Rsrc

b. SUB Rdst, Rsrc

c. INC Rdst

d. SHR Rdst, Rsrc

e. JNZ Rdst

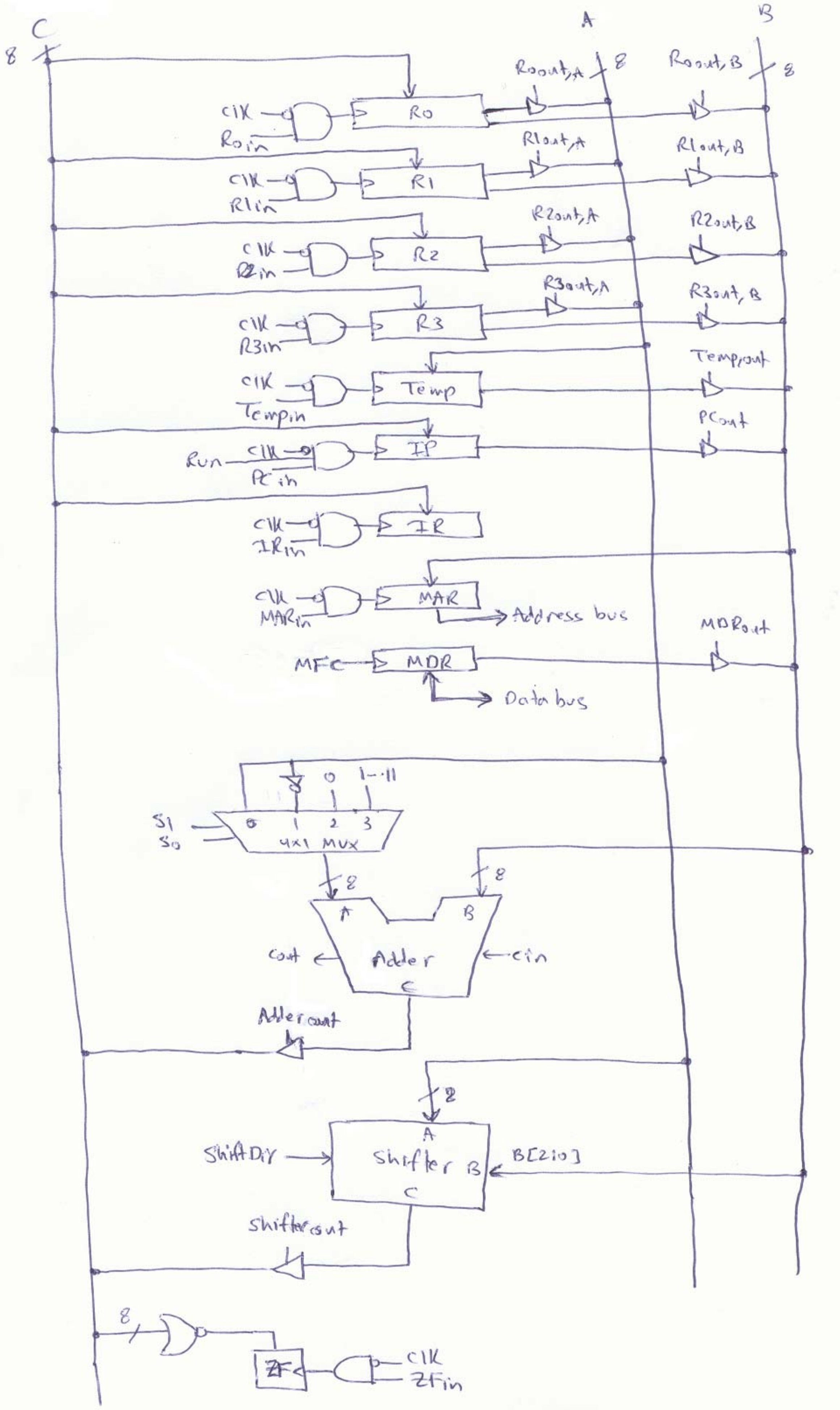
f. LOOP Rdst, Rsrc

(iv) Determine the **maximum speed** at which your CPU design will work given the following delay parameters:

Name	Parameter	Delay
Tri-state buffer delay	T_{buff}	50 ps
INV delay	T_{inv}	40 ps
AND delay	T_{and}	100 ps
OR delay	T_{or}	100 ps
MUX delay	T_{mux}	150 ps
XOR delay	T_{xor}	150 ps
Bus propagation time	t_{bp}	400 ps
ADDER delay	t_{ALU}	500 ps
Shifter delay	$t_{shifter}$	300 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

(v) Show the block diagram of the **hardwired control unit** organization for this CPU indicating all the necessary components and signals. Clearly indicate the size of the various components. Derive the equations for the END signal and all the signals that control register R1.

(vi) 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. Show the format and size of the control word. Show the microroutine of the JNZ instruction (show only the bits set to 1). Assume that the fetch microroutine is at address 0 and the address of JNZ microroutine is at address E. What is the **size of the control store** needed?


C
8
A
B
R0out,A
8
R0out,B
8
clk
R0in
R0
R1out,A
R1out,B
clk
R1in
R1
R2out,A
R2out,B
clk
R2in
R2
R3out,A
R3out,B
clk
R3in
R3
Temp,out
clk
Tempin
Temp
PCout
Run
clk
PC in
IP
clk
IRin
IR
clk
MARin
MAR
Address bus
MDRout
MFC
MDR
Data bus
0
1..11
S1
S0
0 1 2 3
4x1 MUX
8
8
A
B
Cout
Adder
cin
C
Adderout
8
A
ShiftDir
Shifter
B
B[2:0]
C
Shifterout
8
ZF
clk
ZFin

(ii) Fetch Control Sequence:

T1 PCout, MARin, S1, Cin, Adderout, PCin, Read, WMFC

T2 MDRout, S1, Adderout, IRin

(iii) Execution Control Sequence:

a. XCHG Rdst, Rsrc

T3 Rdstout, A, Tempin, Rsrcout, B, S1, Adderout, Rdstin

T4 Tempout, S1, Adderout, Rsrcin, End

b. SUB Rdst, Rsrc

T3 Rdstout, B, Rsrcout, A, S0, Cin, Adderout, Rdstin, End, ZFin

c. INC Rdst

T3 Rdstout, B, S1, Cin, Adderout, Rdstin, End, ZFin

d. SHR Rdst, Rsrc

T3 Rdstout, A, Rsrcout, B, ShiftDir, Shifterout, Rdst,in, ZFin, End

e. JNZ Rdst

T3 Rdstout, B, S1, Adderout, if (ZF=0) Then IPin, End

f. Loop Rdst, Rsrc

T3 Rsrcout, B, S1, S0, Adderout, Rsrcin, ZFin

T4 Rdstout, B, S1, Adderout, if (ZF=0) Then IPin, End

(iv) The pulse width needs to be $\geq t_{buff} + t_{bp} + t_{inv} + t_{mux} + t_{adder} + t_{buff} + t_{bp} + \underset{(=t_{OR})}{t_{NOR}} + t_{su}$

$= 50 + 400 + 40 + 150 + 500 + 50 + 400 + 100 + 100$

$= 1790$ ps

However, since when the clk goes from High to Low, there will be a delay across the AND gate (for capturing signals) that can be subtracted from the pulse width delay.

This delay $= t_{inv} + t_{AND} = 40 + 100 = 140$ ps.

Thus, pulse width $= 1790 - 140 = 1650$ ps.

Clock Period (T) $\geq$ Pulse width $+ T_{ff}$

$\geq 1650 + 150 = 1800$ ps

clock frequency $\leq \frac{1}{T} = \frac{1}{1800\text{ps}} = 556$ MHZ

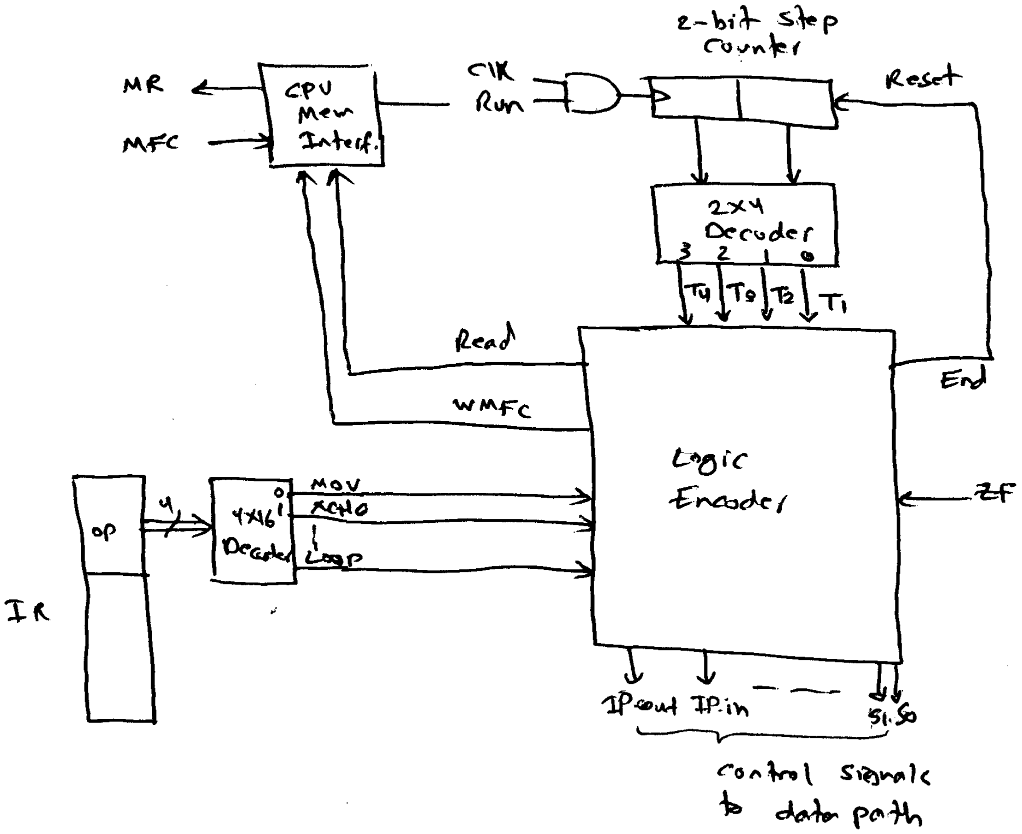
(v) Equation for the END Signal:

END = T4 . XCHG + T3 . SUB + T3 . INC

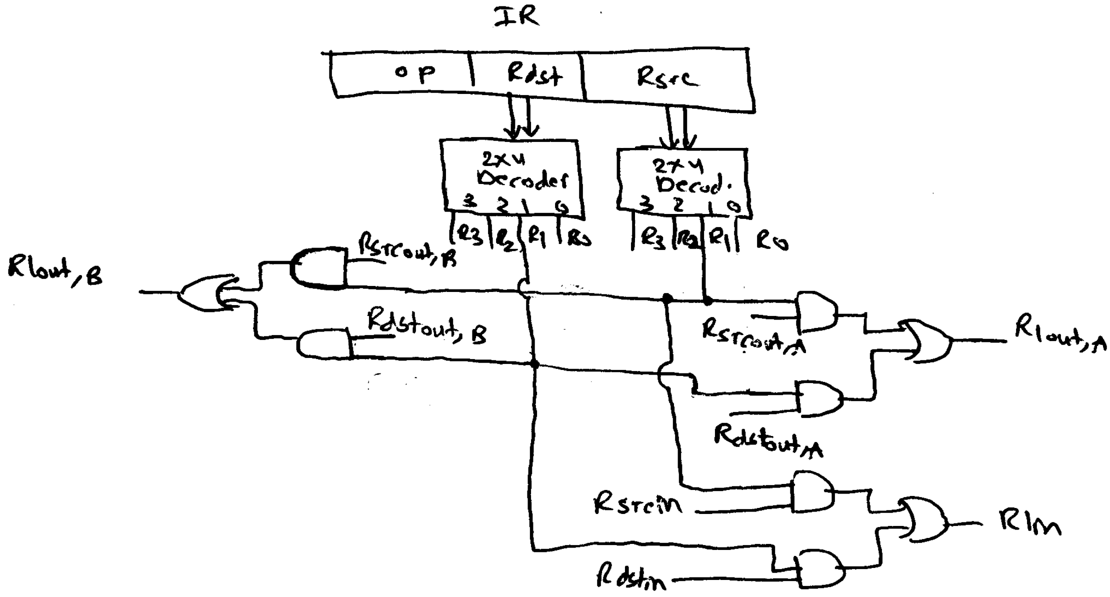
+ T3 . SHR + T3 . JNZ + T4 . Loop

+ - - -

Hardwired Control Unit Design:



Generating Control Signals for R1:



(vi) Microprogrammed Control Unit:

Number of control signals in the design is 23. So, we need 23 bits, one bit for each control signal.

Next, we determine the number of control words that we need to have in the control store as follows:

Fetch	:	2 cws
MoV	:	1 cw
XCHG	:	2 cw's
Add	:	2 cw's
sub	:	2 cw's
INC	:	1 cw
DEC	:	1 cw
SHL	:	1 cw
SHR	:	1 cw
JMP	:	1 cw
JNZ	:	2 cw
JZ	:	2 cw
Loop	:	3 cw's
Total		21 cw's

Thus, we need 5 bits for the microprogram counter and microbranch address.

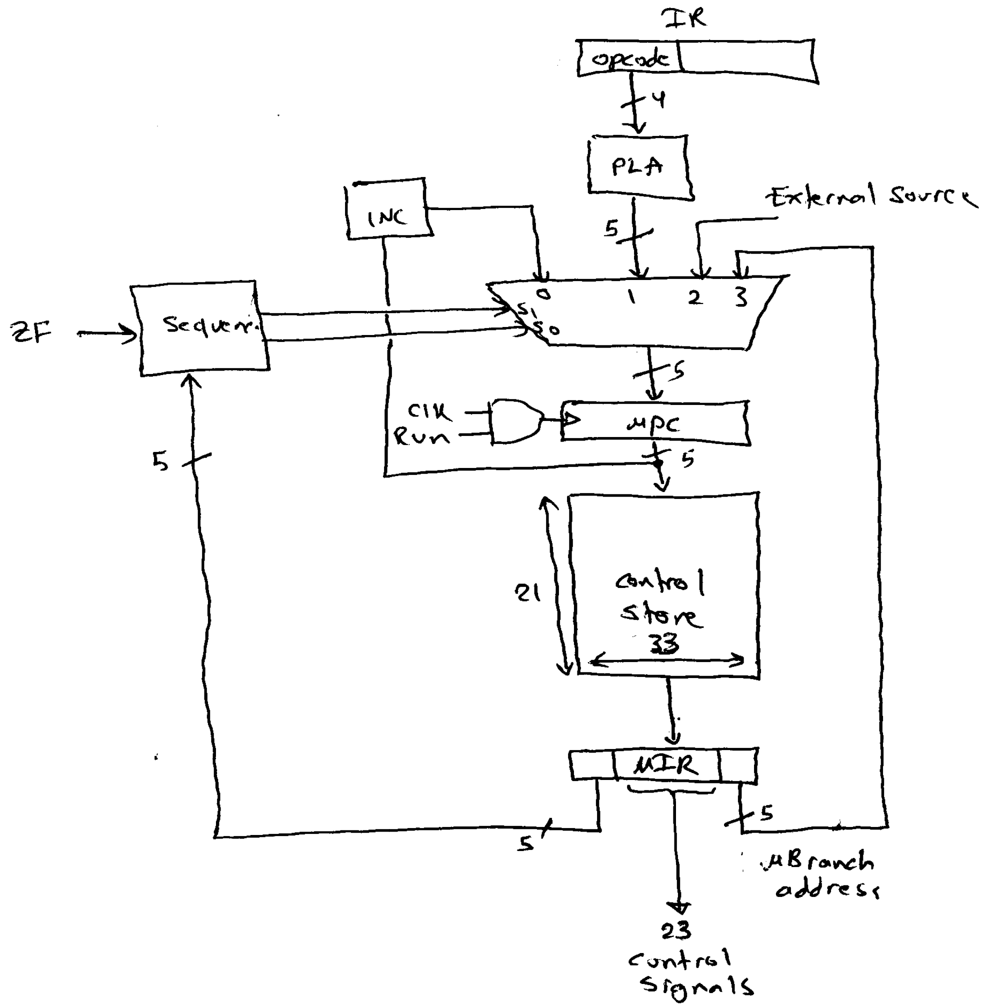
cw Format:

MSl, MSo, UBr, BrZ, BrNZ, 23 bits for the
control signals, 5 bits for μBranch address

$\Rightarrow$ $5 + 23 + 5 = 33$ bits

Thus, the size of the control store is

21×33 bits.



μ Routine for JNZ ;

	CW					23 control sig.				5bits
Address	MS1	MS0	BrU	BrZ	BrNZ	Rdstout,B	S1	Adderout	IPin	μBran address
E	1	1	0	1	0	0	0	0	0	0
F	1	1	1	0	0	1	1	1	1	0