Roll no:

CS321: Quiz 3

Q1: Consider two different implementations, P1 and P2, of the same instruction set. There are different classes of instructions (A, B, C, D, and E) in the instruction set. The clock rate and CPI of each class is given below.

Machine	Clock	CPI A	CPI B	CPI C	CPI D	CPI E
P1	1 GHz	1	1	2	3	2
P2	1.5Ghz	1	2	3	4	3

Assume that peak performance is defined as the fastest rate that a computer can execute any instruction sequence. What are the peak performances of P1 and P2 expressed in instructions per second?

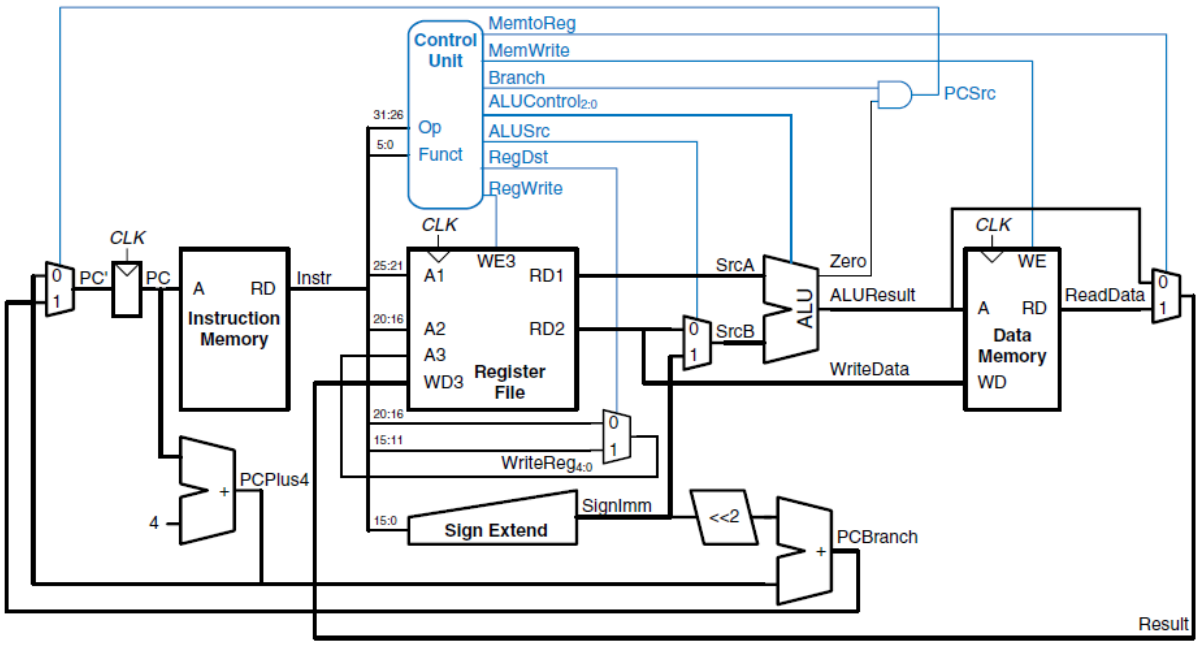
(10 points)

Ans:

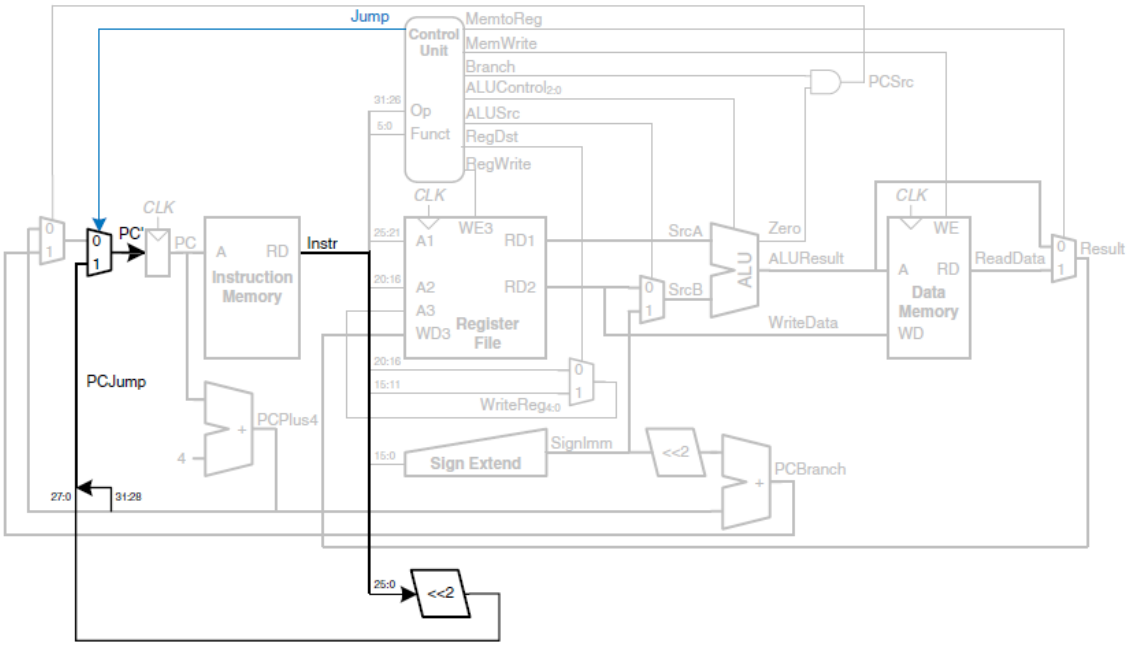
P1 (Using CPA A)= 1G/s x 1 instr/cycle = 1 G instr/s

P1 (Using CPA B)= 1.5G/s x 1 instr/cycle = 1.5 G instr/s

Q5: A single-cycle MIPS processor structure is shown in Figure**.** We wish to add instruction **j** (jump) to the single cycle datapath . Add any necessary datapaths and control signal to the single cycle datapath of the Figure . **(10 points)**


Control Unit
MemtoReg
MemWrite
Branch
ALUControl2:0
ALUSrc
RegDst
RegWrite
Op
Funct
31:26
5:0
PCSrc
CLK
PC'
PC
A
RD
Instruction Memory
Instr
25:21
20:16
A1
A2
A3
WD3
WE3
RD1
RD2
Register File
SrcA
SrcB
ALU
Zero
ALUResult
WriteData
WE
WD
Data Memory
ReadData
20:16
15:11
WriteReg4:0
PCPlus4
4
15:0
Sign Extend
SignImm
<<2
PCBranch
Result

Ans:


Jump
Control Unit
MemtoReg
MemWrite
Branch
ALUControl2:0
ALUSrc
RegDst
RegWrite
Op
Funct
31:26
5:0
PCSrc
CLK
PC'
PC
A
RD
Instruction Memory
Instr
25:21
20:16
A1
A2
A3
WD3
WE3
RD1
RD2
Register File
SrcA
SrcB
ALU
Zero
ALUResult
WriteData
WE
WD
Data Memory
ReadData
Result
20:16
15:11
WriteReg4:0
PCJump
PCPlus4
4
15:0
Sign Extend
SignImm
<<2
PCBranch
27:0
31:28
25:0
<<2

Convert the following High-Level Code to MIPS Assembly Code

```
if (i == j)
f = g + h;
else
f =f - i;
```

(Assume: # $s0 = f, $s1 = g, $s2 = h, $s3 =i, $s4 = j)

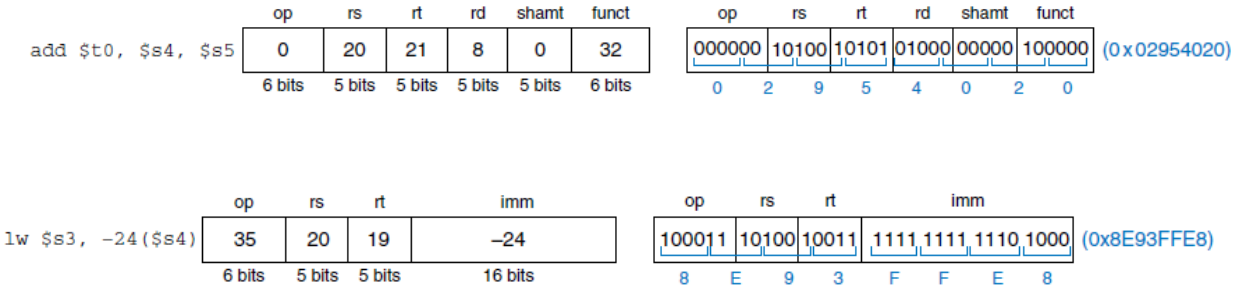
Sample Ans:

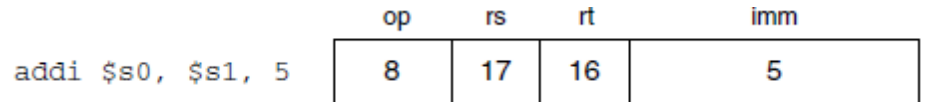
```
bne $s3, $s4, L1        # if i =j, skip if block
add $s0, $s1, $s2       # if block: f = g + h
L1:
sub $s0, $s0, $s3       # f =f - i
```

Translate the following assembly language statement into machine language.
add $t0, $s4, $s5
lw $s3, -24($s4)
`addi $s7, $s1, -15`

($t0, $s4, and $s5 are registers 8, 20, and 21; add has an opcode of 0 and a funct code of 32); lw has an opcode of 35; addi has an opcode of 8)

Ans:






```
addi $s7, $s1, -15        (s7=24, s1=17)
 op
```

Ans: 001000 10001 10111 1111 1111 1111 0001

Q6: Suppose that one of the following control signals in the single-cycle MIPS processor has a *stuck-at-0 fault*, meaning that the signal is always 0, regardless of its intended value. What instructions would malfunction? Why?

(15 points)

(a) *RegWrite*
(b) *ALUOp*1
(c) *MemWrite*

Ans:
(a) R-type, lw, addi
(b) R-type
(c) sw

How many cycles are required to run the following program on the multicycle MIPS processor? What is the CPI of this program?

```
   add $s0, $0, $0              # i = 0

  add $s1, $0, $0               # sum = 0

   addi $t0, $0, 10             # $t0 =10

loop:

    slt $t1, $s0, $t0           # if (i <10), $t1 = 1, else $t1 = 0
```

```
    beq $t1, $0, done          # if $t1 == 0 (i >= 10), branch to done
    add $s1, $s1, $s0          # sum = sum + i
    addi $s0, $s0, 1           # increment i
  j loop
done:
```

Ans:

-Initialization: addi, add, addi. A 3 instruction sequence that on the multicycle processor takes 4 + 4 + 4 = 12 cycles.

-Main loop iteration: slt, beq, add, addi, j. A 5 instruction sequence that takes 4 + 3 + 4 + 4 + 3 = 18 cycles. As it will be executed 10 times (for □ = 0, 1, ..., 9) this totals to 50 instructions or 180 cycles.

- Exit loop iteration: slt, beq. When □ = 10 these two instructions from the loop will be executed, the branch taken, and the program terminated. These last 2 instructions take 4 + 3 = 7 cycles.

So, our total cyles for this program is 12 + (10 □ 180) + 7 = 199 cycles.

(b) What is the CPI of this program?

CPI is cycles per instruction. We already have cycles from above, but need to compute the dynamic instruction total 3 + (10 x 5) + 2 = 55. So CPI = 199/55 = 3.62.