For all questions, please show your step-by-step solution to get full-credits.

Question 1: Assume a 5 stages microprocessor, calculate the number of cycles of below instructions for

1) no pipeline system
2) pipelined system with no control dependency (perfect branch predictor)
3) pipelined system with control dependency (no branch predictor)

Assuming only 1 adder module, 1 pipelined multiplier module, and 2 memory module. Assume that ALU and memory modules take 1 cycle and multiplier modules takes 3 cycles.

Add = add instruction using adder module
Mul = multiply instruction using multiplier module
Ld = load instruction using memory module
Beq = branch if equal instruction using adder module

```
1 Mul R1, R2, R3
2 Add R5,R1,R4
3 Mul R1, R2, R3
4 Mul R4, R5, R6
5 Ld  R1,[R2]
6 Add  R2, R3,R4
7 Add R3, R4,R5
8 BEQ R1, jmp_address   (not branch)
9 Mul R1,R2,R3
10 Add R4, R5,R6
```

Question 2: calculate the IPC of below code for following case

1) Pipeline system
2) Scoreboard technique
3) Tomasulo algorithm

assuming that you have 2-wide machine with 1 adder, 1 Multiplier, 1 Divider, and 1 Load/Store unit.

```
1. Mul R1,R2,R3
2. Div R4,R5,R6
3. Add R7,R8,R9
4. Ld    R2,[R1]
5. Ld    R3,[R4]
6. Mul R1,R2,R3
7. Div R4,R1,R6
8. Sub R7,R4,R9
9.  Add R7, R8, R9
10. Add R4, R5,R6
```

Question 3: Indicate the type of dependency/conflict (or no dependency) for below codes assuming only 1 adder module, 1 multiplier module, and 1 memory module for 5 stages pipelined microprocessor. Assume that ALU and memory modules take 1 cycle and multiplier modules (no pipeline) takes 3 cycles. Note that for each question, there could be more than one type of dependency. If there is no dependency then just state that there is no dependency.

Add = add instruction using adder module
Mul = multiply instruction using multiplier module
Ld = load instruction using memory module
Beq = branch if equal instruction using adder module

a.
```
Mul R1, R2, R3
Add R5,R1,R4
```

b
```
Mul R1, R2, R3
Mul R4, R5, R6
```

c
```
Ld  R1,[R2]
Add  R2, R3,R4
Add R3, R4,R5
Add R4, R5,R6
```

D
```
Mul R1,R2,R3
BEQ R1, jmp_address
```

Question 4: branch predictor

Given the set of instructions below, decide the miss prediction rate assuming that branch address is known in the following case:

1) bimodal branch predictor
2) global branch predictor
3) local branch predictor

Assume that R1 = 1, R2 = 0, R3= 1, R4 =0, R5=10

```
       MOV R1, 1
Addr1: BEQ  R1, R2, Addr2
       ADD R2,R2,1
       BNE  R1, R2,  Addr1
Addr2: ADD R1, R1,1
       BEQ R1, R2 , Addr1
       ADD R4, R4,1
       BNE R4,R5, Addr1
```

Question 5: load/store unit

For the following case, will it be any problem if we execute load before the store in the same basic block. If it has the problem in which iteration and how to address it?

Assume that R1 = 16, R2 = 0, R3= 1, R4 =0, R5=10, R6=10

```
Addr1: STR R3, 0[R1]
       LDR R6, 0[R2]
       ADD R6,R6,1
       ADD R2, R2,8
       ADD R1, R1,4
       ADD R4, R4,1
       BNE R4,R5, Addr1
```

Memory address 0 : 1
Address 4 : 2
Address 8 : 3
Address 12: 4
Address 16: 5
Address 20: 6
Address 24: 7
Address 28: 8
Address 32: 9
Address 36: 10
Address 40: 11

Address 44: 12
Address 48: 13
Address 52: 14
Address 56: 15
Address 60: 16
Address 64: 17
Address 68: 18
Address 72: 19
Address 76: 20
Address 80: 21
Address 84: 22

Question 6: cache/memory

Given below load pattern, calculate the number of cache miss of a directed-mapped 4-set cache with the block size of 32 bytes and the system is byte-addressable.

0xDC800
0xDC810
0xDC820
0xAC800
0xAB800
0xAD800
0xDC800
0xAC800
0xAB800
0xAD808
0xDC80A
0xAC804
0xAB808
0xAD80C

If you're asking to improve the system by adding a 2way associative cache to reduce the number of cache miss, which configuration you will suggest (victim cache, miss cahe, etc,) and what is the new miss rate?

Question 7: LRU access

Given below load pattern, calculate the number of cache miss of a cache with 2 set and 4 ways associative with the block size of 32 bytes and the system is byte-addressable for following case

FIFO algorithm
Pseudo LRU tree algorithm
Pseudo LRU algorithm with one victim way (no next victim). When random chose the way, always select the way one as victim.

0xAB00
0xAC00
0xAD00
0xAE00
0xAB00
0xAF00
0xAA00
0xAB00
0xAC00
0xAB00
0xB000
0xB100
0xAB00
0xAC00

Question 8: Choose the system with the most efficient (performance/cost), when cost of the system is considered by silicon area for the following code.

Assume that R3=0, R4 =0, R5=100000000. The system is 5 stages pipeline.
Control path (chose one)
For control units without pipeline, the die size is 5.
Control unit with pipeline, the die size is 7.
Tomasulo hardware die area is 20 mm^2.

Adder module die area is 15 mm^2. (1 cycle latency)

Multiplier (chose one)
Non-pipeline Multiplier area is 20 mm^2.
Pipelined Multiplier area is 40 mm^2.

Branch predictor (chose one)
none
Bi-modal branch predictor die area is 3 mm^2.
Global branch predictor die area 7 mm^2.

You can decide whether to have cache or not:
I-Cache die area is 20 mm^2. (1 cycle latency)
D-Cache die area is 20 mm^2. (1 cycle latency)

Memory access latency is 300 cycles.

```
        MOV R1, 1
Addr1:  LDR R2,0[R3]
        BEQ  R1, R2, Addr2
        ADD R2,R2,1
        MUL R6,R7,R8
        BNE  R1, R2,  Addr1
Addr2:  ADD R1, R1,1
        BEQ R1, R2 , Addr1
        ADD R4, R4,1
        BNE R4,R5, Addr1
```

Any suggestion to improve the system?

Question 9: Calculate total memory access latency for below memory requests?

Assume that RAS =2cycles, CAS= 2cycles, data transfer = 2cycles, PRE = 2 cycles. PRE is issued to close the page before openning the new page.

FSB = 2 MHz, CPU = 2 GHz.

a) Pattern 1: A0 B0 C0 D0 A1 B1 C1 D1 A2 B2 C3 D2 A3 B3 C3 D3

b) Pattern 2: A0 A1 A2 A3 B0 B1 B2 B3 C0 C1 C2 C3 D0 D1 D2 D3

What’re the CPU latency different and why?

Question 10: Your friend told you that introducing many level of cache can hurt the performance. Is this statement true or false? If it is true, can you give a reason or example how can it happen?