

Student Name: _____

Question 1: Memory Hierarchies (25 points)

Memory blocks are mapped to caches in various ways. The cache configuration or structure determines the mapping. The memory address of the block and the cache configuration determine the placement of the block in the cache. There are three basic cache configurations: direct mapped (one-way set associative), set associative, and fully associative (n -way set associative).

- A. (9 points) Compare and contrast the three cache configurations in terms of:
- hardware resources - which configuration(s) need more resources and what are those resources?
 - access time - which configuration(s) require more access time and why?
 - potential hit rate - which configuration(s) have (potentially) better hit rates and why? Is this true for all workloads?
- B. (3 points) Assume a direct-mapped cache, a two-way set-associative cache, and a fully-associative cache, each with storage for (the tags and data associated with) 8 blocks total. For the memory block with block address 20 (in decimal base), indicate all possible locations in the cache to which this block may be mapped by drawing a picture of each cache and marking an "X" in the targeted cache block.
- C. (4 points) Give short definitions of *hit time*, *miss rate*, *miss penalty*. What is the formula for the *average memory access time* for a three-level cache?
- D. (9 points) Indicate one technique each that is used by modern computer systems to reduce (a) cache time hit; (b) cache miss rates; (c) cache miss penalties. For each technique, briefly explain why they can be effective.

Student Name: _____

Question 2: Dynamic branch prediction (25 points)

Several processors use dynamic scheduling whereby hardware rearranges the instructions execution to reduce stalls. Frequent branches and jumps demand that we attach potential stalls from control dependences. Hardware mechanisms can be used to dynamically predict branches and jumps, i.e., allow the processor to resolve the outcome of a branch early, though preventing control dependences.

A. (6 points) Briefly describe the 1-bit and 2-bit prediction schemas and outline their major performance shortcoming.

B. (6 points) Briefly describe the *correlating predictor schema* or *two-level predictor schema*. Explain the meaning of the values “2” and “4” in a (2,4) predictor. In other words, what does “2” represent? And what does “4” represent?

C. (6 points) Advanced mechanisms for dynamic branch prediction include *branch-target buffer* also called *branch-target cache*. This mechanism can provide 0-cycle unconditional branches and sometimes 0-cycle conditional branches. Briefly explain this technique and explain why it can provide 0-cycle unconditional branches.

D. (7 points) Suppose we have a pipelined architecture that has 7 stages and on a particular benchmark, it has the following dynamic instruction mix.

Instruction Type	Dynamic Frequency	Cycle Count
LOADS	24%	7
STORES	15%	7
ALU	37%	7
BRANCHES (taken)	19%	7
BRANCHES (not taken)	5%	7

Suppose we are using dynamic branch prediction and that each misprediction costs a 2-cycle penalty. Assume there are no additional hazards. What branch prediction accuracy rate (in %) is needed in order to achieve a throughput of 0.90 instructions per cycle? (Show work)

Student Name: _____

Question 3: Five-stage pipelining with bypass (or forwarding) hardware and delayed branches (25 points)

Pipelining is an implementation technique whereby multiple instructions are overlapped in execution and exploit parallelism among the instructions in a sequential instruction stream. The potential overlapping among instructions is called instruction-level parallelism (ILP). Structural, data, and control hazards can cause stalls that limit the level of ILP in a program. Stalls can ultimately affect the value of the cycles per instruction, or CPI.

A. (6 points) Briefly describe the three different types of hazards, i.e., structural, data, and control hazards in terms of what conditions cause them. Write the formula of the CPI in a pipelined processor.

B. (7 points) Consider pipelining code that performs the dot product of two integer vectors in memory. The two vectors, X and Y, are 100 words long, beginning at addresses 1000 and 2000 respectively. Here is the pseudo-code for the routine:

```
(I1)      R0 <- 0                ; initialize result
(I2)      R1 <- 1000            ; initialize X pointer
(I3)      R2 <- 2000           ; initialize Y pointer
(I4)      R3 <- 100             ; load element counter
(I5) Loop  R4 <- (R1)           ; load X element
(I6)      R5 <- (R2)           ; load Y element
(I7)      R6 <- R4 * R5         ; compute product
(I8)      R0 <- R0 + R6         ; add to result
(I9)      R1 <- R1 + 4          ; increment X pointer
(I10)     R2 <- R2 + 4          ; increment Y pointer
(I11)     R3 <- R3 - 1          ; decrement element counter
(I12)     if R3 != 0, goto Loop ; loop if more elements
```

Consider this routine executing on a five-stage pipelined processor (i.e., Instruction Fetch or IF, Instruction Decode or ID, Execution or EX, Memory Access or MEM, and Write Back or WB). In this implementation of the pipelined architecture, the branch predicate is determined in the EX stage. Also assume there is no bypass (or forwarding) hardware or delayed branches. However, registers are written at the start of the cycle and read at the end of the cycle. If the instructions are executed in the order specified above, determine how many stall cycles are required for each instruction in the loop (i.e., how many bubbles must be inserted BEFORE each instruction to avoid hazards). For instruction (I5), determine how many bubbles are produced by the branch instruction (I12).

C. (6 points) Now assume bypass hardware is provided, but the execution order remains the same. What is the new number of required stalls before each instruction? Note that instruction (I5) should be handled as in the previous part.

D. (6 points) Now assume bypass hardware is no longer available, but we are able to reorder the arithmetic operations to avoid data hazards. Reorder the instructions to eliminate data hazards. Add NO-OP instructions if necessary.

Student Name: _____

Question 4: Multiple-issue processors (25 points)

The CPI can be decreased to less than one only if we can issue multiple instructions per clock cycle. Multiple-issue processors allow multiple instructions to issue in a clock cycle and can come in three major flavors:

- Statically scheduled superscalar processors
- VLIW (very long instruction word) processors
- Dynamically scheduled superscalar processors

A. (4 point) Describe the three scheduling techniques in terms of (1) number of instructions issued per clock; (2) hazard detection, i.e., hardware or software; and (3) scheduling, i.e., static or dynamic.

B. (3 points) Two different approaches can be used to issue multiple instructions per clock in a dynamically scheduled processor. Briefly describe the two approaches. What approach is used in modern processors?

C. (10 points) Consider a superscalar microprocessor with a superscalar factor of six. It has a design similar to the PowerPC, using reservation stations, a renaming buffer, and a reorder buffer to support out-of-order execution. The functional units have the following latencies (measured in number of cycles):

IntegerALU: 1 cycle
FP multiply: 5 cycles
Memory: 10 cycles
FP divide: 10 cycles
FP add/sub: 10 cycles
Branch: 1 cycle

Suppose that the superscalar pipeline is empty and the following instructions are issued. F3 holds value 80, F4 holds value 4, and F5 holds value 2.

Inst #	Instruction
(i1)	ADDF F1, F3, F4
I2	MULTF F2, F3, F5
I3	ADDF F3, F1, F2
I4	ADDF F2, F4, F3
I5	DIVF F4, F2, F4
I6	MULTF F3, F4, F5

Note that each instruction deals with floating point registers and has the following structure:
<operation> <destination>, <source 1>, <source2>

There are 3 reservation stations for the FPAdd functional unit, 2 reservation stations for the FPMultiply functional unit, and 1 reservation station for the FPDivide functional unit. What is the status of the reservation stations and renaming buffer once all the instructions have issued and I2 has completed execution and its results have been broadcast? Assume that none of the other functional units have finished execution, yet; only the MULTF for I2 has finished execution.

Student Name: _____

Note that you do not need to copy the reservation station and the reorder buffer on a separate sheet: a copy of the tables is provided to you on a separate sheet of paper. Attach the separate sheet to your answer sheets if you answer this question.

Reservation station:

Name	Busy	Operation	Source 1 (value or RB)	Source 2 (value or RB)	Destination
FPAdd1					
FPAdd2					
FPAdd3					
FPMultiply1					
FPMultiply2					
FPDivide1					

Reorder buffer:

Name	Busy	Instructions	Destination	Value
1				
2				
3				
4				
5				
6				

D. (8 points) Suppose the processor is executing the following code fragment:

```

Inst #   Instruction
I1       DIVF F0, F1, F2
I2       MULTF F3, F4, F5
I3       ADDF F0, F7, F8
I4       SUBF F9, F10, F11
I5       SUBF F12, F13, F14
I6       MULTF F15, F16, F17
    
```

Suppose the reorder buffer has the state shown in the first 5 columns of this table:

Inst #	Register #	RB #	Validity	Status	Write to register file?	Order in which results are written
I1	F0	#1	Y	P		
I2	F3	#2	Y	C		
I3	F0	#3	Y	C		
I4	F9	#4	Y	E		
I5	F12	#5	Y	P		
I6	F15	#6	Y	C		
			N			
			N			

The meaning of the status codes are: C = execution has completed, P = execution is pending, E = an exception occurred, M = mispredicted instruction.

Fill in the rightmost 2 columns in the table above. Indicate with a "Yes" which instructions write their result back to the register file. For those instructions that do write to the register file, indicate

Student Name: _____

in what order they write their results. (If an instruction's results are not written back to the register file, then put a "No" and an "X" in the last 2 columns.)

Note that you do not need to copy the table on a separate sheet: a copy of the table is provided to you on a separate sheet of paper. Attach the separate sheet to your answer sheets if you answer this question.

Question 4: Multiple-issue processors (supplement sheet)

Question 4.C

Reservation station:

Name	Busy	Operation	Source 1 (value or RB)	Source 2 (value or RB)	Destination
FPAdd1					
FPAdd2					
FPAdd3					
FPMultiply1					
FPMultiply2					
FPDivide1					

Reorder buffer:

Name	Busy	Instructions	Destination	Value
1				
2				
3				
4				
5				
6				

Question 4.D

Inst #	Register #	RB #	Validity	Status	Write to register file?	Order in which results are written
I1	F0	#1	Y	P		
I2	F3	#2	Y	C		
I3	F0	#3	Y	C		
I4	F9	#4	Y	E		
I5	F12	#5	Y	P		
I6	F15	#6	Y	C		
			N			
			N			