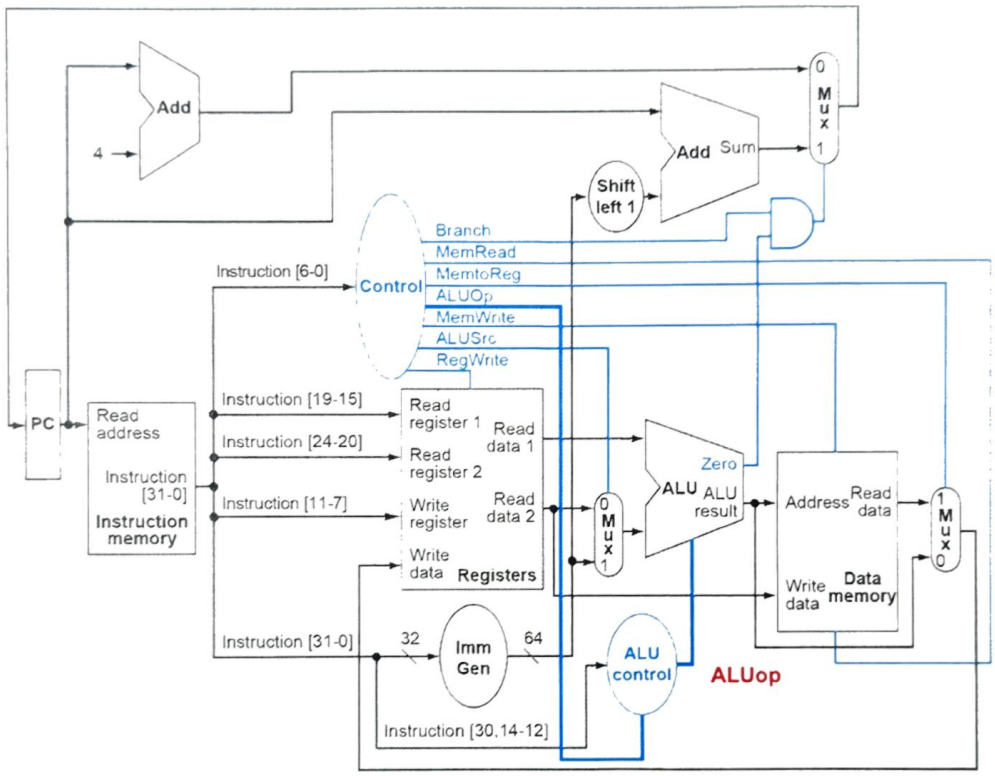
NATIONAL TSING HUA UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE
CS 4100: Computer Architecture
Spring 2024, Final Examination

ID: _____ **Name:** _____

☆ **You must list all the calculation work.**

1. (10%) Consider the single-cycle processor which is shown below.



Instruction	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	ALUOp0
R-format	0	0	1	0	0	0	1	0
ld	1	1	1	1	0	0	0	0
sd	1	X	0	0	1	0	0	0
beq	0	X	0	0	0	1	0	1

(a) (3%) Among the three instructions: **ld**, **add**, and **beq**, which instruction(s) could fail to operate correctly if ALUSrc is always set to 1?

(b) (3%) Suppose the Mux originally controlled by MemtoReg is now controlled by ALUOp1. Describe how to modify the processor so that it can still operate correctly without adding any new gates.

(c) (4%) Consider adding the R-format instruction `slt rd, rs1, rs2` to the processor, which

has the same opcode as **add**, **sub**, **and**, and **or**, but has a different funct3. From Chapter 3, we know that the ALU can support the set-less-than operation by setting the ALUop (i.e., the 4-bit control signal of the ALU) to 0111. Can we modify only the "ALU control" to make the processor work correctly for the **slt** instruction as well as the original instructions? Justify your answer.

2. (6%) Assume the clock period is 10 ns and no pipeline stalls occur.
   (a) (3%) If a 4-stage pipelined processor takes 200 ns to execute S instructions, how long will it take to execute 3×S instructions?
   (b) (3%) Consider a pipelined processor which has N stages. If it takes 500 ns to execute M instructions and 1700 ns to execute 4×M instructions, what are N and M, respectively?

3. (5%) What is structural hazard? Why can the five-stage pipelined processor avoid structural hazards?

4. (20%) Consider the following sequence of instructions executed on the five-stage pipelined processor with both the forwarding unit and hazard detection unit. Assume that the execution starts in clock cycle 1.

```
and x25, x21, x25
sd x25, 0(x21)
ld x24, 4(x21)
ld x23, 0(x24)
sub x23, x23, x22
add x21, x23, x22
```

   (a) (15%) Indicate in which clock cycle(s) the processor forwards correct data to the EX stage, and determine the number of clock cycles required to complete the execution.
   (b) (5%) Assume that both the forwarding unit and hazard detection unit are not present in the processor. Show how to insert the minimum number of NOP (no operation) instructions to ensure correct execution and determine the number of clock cycles required to complete the execution.

5. (6%) Consider the sequence of branch outcomes: NT, T, NT, NT, T, T, T, NT, for a branch instruction, where T denotes taken and NT denotes not-taken.
   (a) (2%) What is the accuracy rate of the always-not-taken predictor for this sequence of branch outcomes?
   (b) (2%) Consider a 1-bit dynamic predictor which starts at the NT state. What is the accuracy rate of the predictor for this sequence of branch outcomes?
   (c) (2%) Consider a 2-bit dynamic predictor which starts at the "strongly predict taken" state. What is the accuracy rate of the predictor for this sequence of branch outcomes?

6. (4%) Consider the following sequence of instructions executed on the five-stage pipelined processor, and assume that the execution starts in clock cycle 1.

```
add x20, x21, x22
beq x21, x22, LABEL
ld x21, 0(x22)
```

Suppose the second instruction is detected to have an invalid target address, causing an exception in the ID stage. What instructions will appear in the IF, ID, EX, and MEM stages, respectively, in clock cycle 4? Note that each instruction in your answer should be one chosen from the given three instructions, the NOP instruction, and the first instruction of the exception handler.

7. (4%) Given each of the following techniques or components, determine whether it is associated primarily with a software-based approach, a hardware-based approach, or both, to exploring instruction-level parallelism (ILP). For each question, you only need to answer S (for software-based approach), H (for hardware-based approach), or B (for both approaches).

   (a) (1%) Multiple issue

   (b) (1%) Dynamic scheduling

   (c) (1%) Reorder buffer

   (d) (1%) Register renaming

8. (5%) Consider a 12-bit SEC/DEC code protected by using a Hamming single error-correcting code with an additional parity bit as the right-most bit, p_{12}. Suppose that the received pattern is `1100_0111_1101`.

   (a) (1%) How many bits are the data bits in this codeword?

   (b) (2%) Show the decoding process.

   (c) (2%) Are there any errors in this received pattern and can you correct or detect them?

9. (10%) Consider designing a cache architecture of a specific processor whose base CPI without memory stalls is 1.5, assuming all references hit in the primary cache. The main memory access delay is 400 cycles. The analysis of a two-level cache system shows that the miss rate per instruction at the L-1 cache is 10%. The designers also figured that the local miss rate at the L-2 cache varies depending on its associativity. That is, the miss rate is $\left(\frac{5\%}{1+n}\right)$ where $n \in \{1, 2, 4, 8\}$ for four different cache designs (e.g., $n = 1$ for direct-mapped; $n = 2$ for 2-way set associative; $n = 4$ for 4-way set associative; $n = 8$ for 8-way set associative). Additionally, the access delay of the L-2 cache depends on the associativity too, i.e., the access cycles are $\left(20 + \left\lfloor\frac{n}{2}\right\rfloor\right)$. For example, the access cycles are 20 for the direct-mapped cache and 21 for the 2-way set associative design.

   (a) (2%) What is the effective CPI with only the L-1 cache?

   (b) (8%) Suppose the architect wants to reduce the effective CPI to 4.1 or lower. Which of the four different L-2 cache designs can be used to satisfy the requirement?

10. (20%) Consider a computer system that has a 1024-byte 2-way set associative cache with LRU replacement, write-back, and write-allocate policy. Suppose a cache block has 16 words. The physical memory address has 16 bits.

    (a) (2%) What is the bit-width of the tag field in each cache block?

    (b) (18%) Given the initial content of Set 0 and Set 1 in the following table with the valid bit (V=1 indicates a valid block; V=0 or a blank field indicates an invalid block) and dirty bit (D=1 indicates a dirty block), ignoring the reference bit and data fields. The rest of the cache is empty. For each of the following memory accesses of the block addresses in decimal, indicate if a **hit** or **miss** occurs. For each write-back due to a block replacement, you must indicate **the write-**

back block address. Also, draw the final content of the cache.

i. Write Block Address $(24)_{10}$
ii. Read Block Address $(77)_{10}$
iii. Write Block Address $(52)_{10}$
iv. Read Block Address $(80)_{10}$
v. Write Block Address $(65)_{10}$

	Way 0			Way 1		
Set	V	D	Tag	V	D	Tag
0	1	1	7	1	0	3
1	1	0	5			
⋮	⋮	⋮	⋮	⋮	⋮	⋮

11. (10%) Consider a byte-addressable virtual memory system in which each virtual address is 48 bits, each physical address is 40 bits, the page size is 4K bytes, and the TLB is direct-mapped with 2 entries. Suppose that the following code is executed:

```
long int A[2][2048], B[2048], C[2];
for (i=0; i<2; i++)
     for(j=0; j<2048; j++)
          C[i] += A[i][j] * B[j];
```

Note that for `long int` data type, an element is 4 bytes. Suppose that the array `A[][]` is allocated in a row-major fashion starting at the virtual address `0x000003C62000`, i.e., `A[0][0]` is at `0x000003C62000`, `A[0][1]` is at `0x000003C62004`, and so on. The array `B[]` is allocated consecutively in the order of `B[0]`, `B[1]`, and so on. Assume that `C[0]` and `C[1]` are preloaded to registers `x10` and `x11` so that we don't need to consider their memory accesses here.

When the loop starts to execute, all the data of `A[][]` and `B[]` is loaded into the physical memory. So, the page table has the mapping from their virtual page numbers to the physical page numbers. Also, assume that the TLB is empty at this moment.

(a) (2%) How many pages are allocated to store the arrays `A[][]` and `B[]`?

(b) (4%) How many TLB misses will occur when executing the code?

(c) (4%) Suppose that you can specify the virtual memory allocation of the arrays `A[][]` and `B[]`. And the virtual address space that you can utilize is limited from `0x000003C62000` to `0x000003C68FFF`. Explain how to rearrange the virtual memory allocation to reduce the number of TLB misses to 6. You can also revise the code when necessary.