**Part 1: Knowledge Base (5 points)**

1. List and briefly explain various ways in which an instruction pipeline can deal with conditional branch instructions.

Various ways in which an instruction pipeline can deal with conditional branch instructions:

a) Branch Delay Slot: In this approach, the instruction immediately following a branch instruction is executed before the branch is taken or resolved. This helps in utilizing the pipeline stages effectively, as the instruction in the delay slot is independent of the branch outcome.

b) Branch Prediction: Branch prediction techniques attempt to predict the outcome of a branch instruction before it is resolved. This prediction allows the pipeline to continue executing instructions speculatively, based on the predicted branch outcome. If the prediction is correct, the pipeline continues smoothly; otherwise, the incorrectly predicted instructions need to be flushed or discarded.

c) Branch Target Buffer (BTB): A BTB is a cache-like structure that stores the target address of recently executed branch instructions. When a branch instruction is encountered, the BTB is checked to retrieve the predicted target address. If the prediction is correct, the pipeline can fetch instructions from the predicted target address without stalling.

1. What are some typical characteristics of a RISC instruction set architecture?

Typical characteristics of a RISC instruction set architecture (ISA) include:

a) Simple Instructions: RISC ISAs have simple instructions that perform basic operations. Each instruction typically executes in a single clock cycle.

b) Load/Store Architecture: RISC ISAs follow a load/store architecture, where data is explicitly loaded from memory into registers before being operated upon and stored back to memory.

c) Fixed Instruction Format: RISC ISAs often have a fixed-length instruction format, which simplifies the instruction fetch stage in the pipeline.

d) Large Number of Registers: RISC ISAs typically have a large number of general-purpose registers, enabling more operations to be performed using registers instead of memory.

e) Compiler-Friendly: RISC ISAs are designed to be easily optimized by compilers, with a focus on reducing memory accesses and instruction count.

1. Briefly define the following terms:

a) True Data Dependency: A true data dependency occurs when the result of one instruction depends on the result of a previous instruction. The dependency prevents the instructions from being executed out of order because the correct data must be available before the dependent instruction can proceed.

b) Procedural Dependency: A procedural dependency refers to a dependency between instructions that arises due to the order of execution mandated by the program's control flow. It occurs when the correctness of the program depends on the sequential execution of instructions.

c) Resource Conflicts: Resource conflicts occur when multiple instructions require the simultaneous use of the same hardware resource, such as a register or an execution unit. These conflicts can lead to stalls or delays in the pipeline.

d) Output Dependency: An output dependency, also known as a write-after-write (WAW) dependency, occurs when two instructions write to the same destination register. The correct execution order needs to be maintained to ensure the correct value is stored in the register.

e) Antidependency: An antidependency, also known as a write-after-read (WAR) dependency, occurs when an instruction reads from a register that a subsequent instruction writes to. This dependency restricts the execution order to ensure the correct data is read by the dependent instruction.

1. What is the purpose of an instruction window?

The purpose of an instruction window is to hold a set of instructions that are candidates for execution in a pipelined processor. The instruction window allows the processor to examine and select instructions for execution based on various factors such as dependencies, resource availability, and branch outcomes. It helps in reordering and scheduling instructions to maximize the utilization of execution resources and improve overall performance. The instruction window allows for out-of-order execution, where instructions can be executed ahead of their original program order as long as the dependencies are satisfied.

**Part 2: Multiple Choice (5 points)**

1. \_\_\_\_\_\_\_\_\_\_ are bits set by the processor hardware as the result of operations.

A. MIPS B. Condition codes

C. Stacks D. PSWs

1. The \_\_\_\_\_\_\_\_\_ contains the address of an instruction to be fetched.

A. instruction register B. memory address register

C. memory buffer register D. program counter

1. The \_\_\_\_\_\_\_\_ determines the opcode and the operand specifiers.

A. decode instruction B. fetch operands

C. calculate operands D. execute instruction

1. \_\_\_\_\_\_\_\_\_ is a pipeline hazard.

A. Control B. Resource

C. Data D. All of the above

1. A \_\_\_\_\_\_\_\_ hazard occurs when there is a conflict in the access of an operand location.

A. resource B. data

C. structural D. control

1. A \_\_\_\_\_\_\_\_\_ is a small, very-high-speed memory maintained by the instruction fetch stage of the pipeline and containing the *n* most recently fetched instructions in sequence.

A. loop buffer B. delayed branch

C. multiple stream D. branch prediction

1. The \_\_\_\_\_\_\_\_\_ is a small cache memory associated with the instruction fetch stage of the pipeline.

A. dynamic branch B. loop table

C. branch history table D. flag

1. The ARM architecture supports \_\_\_\_\_\_\_ execution modes.

A. 2 B. 8

C. 11 D. 7

1. \_\_\_\_\_\_\_\_\_ determines the control and pipeline organization.

A. Calculation B. Execution sequencing

C. Operations performed D. Operands used

1. \_\_\_\_\_\_\_\_\_ is the fastest available storage device.

A. Main memory B. Cache

C. Register storage D. HLL

1. A \_\_\_\_\_\_\_\_ instruction can be used to account for data and branch delays.

A. SUB B. NOOP

C. JUMP D. all of the above

1. A tactic similar to the delayed branch is the \_\_\_\_\_\_\_\_\_, which can be used on LOAD instructions.

A. delayed load B. delayed program

C. delayed slot D. delayed register

1. The MIPS R4000 uses \_\_\_\_\_\_\_\_ bits for all internal and external data paths and for addresses, registers, and the ALU.

A. 16 B. 32

C. 64 D. 128

1. All MIPS R series processor instructions are encoded in a single \_\_\_\_\_\_ word format.

A. 4-bit B. 8-bit

C. 16-bit D. 32-bit

1. A \_\_\_\_\_\_\_\_\_ architecture is one that makes use of more, and more fine-grained pipeline stages.

A. parallel B. superpipelined

C. superscalar D. hybrid

1. The R4000 can have as many as \_\_\_\_\_\_\_ instructions in the pipeline at the same time.

A. 8 B. 10

C. 5 D. 3

1. The superscalar approach can be used on \_\_\_\_\_\_ architecture.

A. RISC B. CISC

C. neither RISC nor CISC D. both RISC and CISC

1. The essence of the \_\_\_\_\_\_\_\_ approach is the ability to execute instructions independently and concurrently in different pipelines.

A. scalar B. branch

C. superscalar D. flow dependency

1. Which of the following is a fundamental limitation to parallelism with which the system must cope?

A. procedural dependency B. resource conflicts

C. antidependency D. all of the above

1. The situation where the second instruction needs data produced by the first instruction to execute is referred to as\_\_\_\_\_\_\_

A. true data dependency B. output dependency

C. procedural dependency D. antidependency

1. The instructions following a branch have a \_\_\_\_\_\_\_\_\_ on the branch and cannot be executed until the branch is executed.

A. resource dependency B. procedural dependency

C. output dependency D. true data dependency

1. \_\_\_\_\_\_\_\_ refers to the process of initiating instruction execution in the processor’s functional units.

A. Instruction issue B. In-order issue

C. Out-of-order issue D. Procedural issue

1. Instead of the first instruction producing a value that the second instruction uses, with \_\_\_\_\_\_\_\_\_\_\_ the second instruction destroys a value that the first instruction uses.

A. in-order issue B. resource conflict

C. antidependency D. out-of-order completion

1. \_\_\_\_\_\_\_\_\_\_ exists when instructions in a sequence are independent and thus can be executed in parallel by overlapping.

A. Flow dependency B. Instruction-level parallelism

C. Machine parallelism D. Instruction issue

1. \_\_\_\_\_\_\_\_\_ is determined by the number of instructions that can be fetched and executed at the same time and by the speed and sophistication of the mechanisms that the processor uses to find independent instructions.

A. Machine parallelism B. Instruction-level parallelism

C. Output dependency D. Procedural dependency

1. Which of the following is a hardware technique that can be used in a superscalar processor to enhance performance?

A. duplication of resources B. out-of-order issue

C. renaming D. all of the above

1. SMPs, clusters, and NUMA systems fit into the \_\_\_\_\_\_\_\_ category of computer systems.

A. SISD B. MIMD

C. SIMD D. MISD

1. A \_\_\_\_\_\_\_\_\_ problem arises when multiple copies of the same data can exist in different caches simultaneously, and if processors are allowed to update their own copies freely, an inconsistent view of memory can result.

A. cache coherence B. cluster

C. failover D. failback

1. A \_\_\_\_\_\_\_\_\_\_ is an instance of a program running on a computer.

A. process B. process switch

C. thread D. thread switch

1. A \_\_\_\_\_\_\_\_ is a dispatchable unit of work within a process that includes a processor context and its own data area for a stack.

A. process B. process switch

C. thread D. thread switch