CSC 310 ASSIGNMENT

Briefly explain each of the following architectures:

1. RISC

2. CISC

3. VLIW.

Answers

1. RISC (reduced Instruction Set Computer)

A processor architecture that shifts the analytical process of a computational task from the execution or runtime to the preparation or compile time. By using less hardware or logic, the system can operate at higher speeds. RISC cuts down on the number and complexity of instructions, on the theory that each one can be accessed and executed faster, and that less semiconductor real estate is required to process them. The result is that for any given semiconductor technology, a more powerful microprocessor can be produced with RISC than with complex instruction set computer (CISC) architectures. This simplification of computer instruction sets gains processing efficiencies.

That theme works because all computers and programs execute mostly simple instructions. RISC has five design principles:

• Single-cycle execution

• Hard-wired control, little or no microcode

• Simple instructions, few addressing modes

• Load and store, register-register design

• Efficient, deep pipelining

1. CISC (Complex Instruction Set Computing)

This is a type of microprocessor design. The CISC architecture contains a large set of computer instructions that range from very simple to very complex and specialized. Though the design was intended to compute complex instructions in the most efficient way, it was later found that many small, short instructions could compute complex instructions more efficiently. This led to a design called Reduced Instruction Set Computing (RISC), which is now the other major kind of microprocessor architecture. Intel Pentium processors are mainly CISC-based, with some RISC facilities built into them, whereas the PowerPC processors are completely RISC-based.

1. VLIW (Very long instruction word)

It refers to [instruction set architectures](https://en.wikipedia.org/wiki/Instruction_set_architecture) designed to exploit [instruction level parallelism](https://en.wikipedia.org/wiki/Instruction_level_parallelism) (ILP). Whereas conventional [central processing units](https://en.wikipedia.org/wiki/Central_processing_unit) (CPU, processor) mostly allow programs to specify instructions to execute in sequence only, a VLIW processor allows programs to explicitly specify instructions to execute in [parallel](https://en.wikipedia.org/wiki/Parallel_computing). This design is intended to allow higher performance without the complexity inherent in some other designs. The traditional means to improve performance in processors include dividing instructions into sub steps so the instructions can be executed partly at the same time (termed pipelining), dispatching individual instructions to be executed independently, in different parts of the processor ([superscalar](https://en.wikipedia.org/wiki/Superscalar) architectures), and even executing instructions in an order different from the program ([out-of-order execution](https://en.wikipedia.org/wiki/Out-of-order_execution)). These methods all complicate hardware (larger circuits, higher cost and energy use) because the processor must make all of the decisions internally for these methods to work. In contrast, the VLIW method depends on the programs providing all the decisions regarding which instructions to execute simultaneously and how to resolve conflicts. As a practical matter, this means that the [compiler](https://en.wikipedia.org/wiki/Compiler) (software used to create the final programs) becomes far more complex, but the hardware is simpler than in many other means of parallelism.