**OYEBADE OLUWAROTIMI**

**17/SCI01/074**

**CSC 310**

**ASSIGNMENT**

1. **RISC (Reduced Instruction Set Computer)**

RISC (reduced instruction set computer) is a microprocessor that is designed to perform a smaller number of types of computer instructions so that it can operate at a higher speed (perform more millions of instructions per second, or MIPS). Since each instruction type that a computer must perform requires additional transistors and circuitry, a larger list or set of computer instructions tends to make the microprocessor more complicated and slower in operation.

The RISC concept has led to a more thoughtful design of the microprocessor. Among design considerations are how well an instruction can be mapped to the clock speed of the microprocessor (ideally, an instruction can be performed in one clock cycle); how "simple" an architecture is required; and how much work can be done by the microchip itself without resorting to software help.

1. **CISC (Complex Instruction Set Computer)**

Stands for "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)**

Very long instruction word (VLIW) refers to instruction set architectures designed to exploit instruction level parallelism (ILP). Whereas conventional central processing units (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. This design is intended to allow higher performance without the complexity inherent in some other designs.

VLIW executes operations in parallel, based on a fixed schedule, determined when programs are compiled. Since determining the order of execution of operations (including which operations can execute simultaneously) is handled by the compiler, the processor does not need the scheduling hardware that the three methods described above require. Thus, VLIW CPUs offer more computing with less hardware complexity (but greater compiler complexity) than do most superscalar CPUs.