**RISC and CISC Comparisons and Contrasts**

1. **Complexity:** Both RISC and CISC architectures aim to simplify the task of programming, but RISC architectures have a simpler instruction set with fewer instructions, while CISC architectures have a larger and more complex instruction set.
2. **Execution time:** Both RISC and CISC architectures aim to execute programs quickly, but RISC architectures tend to have faster execution times due to their simpler instruction set.
3. **Power consumption:** Both RISC and CISC architectures need to be energy-efficient, but RISC architectures tend to have lower power consumption than CISC architectures.
4. **Instruction decoding:** Both RISC and CISC architectures require instruction decoding, but CISC architectures require more complex decoding than RISC architectures.
5. **Memory access:** Both RISC and CISC architectures need to access memory, but CISC architectures tend to have better memory access capabilities due to their more complex instruction set.
6. **Compiler dependence:** Both RISC and CISC architectures can benefit from compiler optimization, but RISC architectures are more dependent on the compiler for optimization than CISC architectures.
7. **Function calls:** Both RISC and CISC architectures support function calls, but RISC architectures use a simpler calling convention than CISC architectures.
8. **Cost:** Both RISC and CISC architectures have costs associated with them, but RISC architectures tend to be more expensive than CISC architectures due to their specialized hardware requirements.
9. **Code portability:** Both RISC and CISC architectures can be portable across different computer systems, but RISC architectures tend to be more portable than CISC architectures.
10. **Code density:** Both RISC and CISC architectures can achieve high code density, but CISC architectures generally have a higher code density than RISC architectures.