Features of RISC:

Feature 1

RISC processors have a fixed instruction size. In a CISC microcontroller such as the 8051, instructions can be 1, 2 or even 3 bytes.

For example, look at the following instruction in the 8051:

```
CLR C          ; 1-byte instruction
ADD Accumulator, #mybyte ; 2 byte instruction
LJMP target_address      ; 3-byte instruction
```

This var... the task of

This variable instruction size makes the task of the instruction decoder very difficult because of the size of the incoming instruction is never known.

Feature 2

One of the major characteristics of RISC architecture is a large number of registers. All RISC architecture have at least 32 registers.

One advantage of a large number of registers is that it avoids the need for a large stack to store

Microsoft Whiteboard

to store parameters.

FA

Settings menu

Feature 3

RISC processors have a small instruction set. RISC processor have only basic instructions such as ADD, SUB, MUL, LOAD STORE AND OR and so on.

The limited number of instructions is one of the criticism leveled at the RISC processors because it makes the job of Assembly language programmers much more [illegible] to CISC

Microsoft Whiteboard

much more tedious and difficult compared to CISC assembly language programming

This is one reason that RISC is used more commonly in high-level language environment such as C programming language rather than assembly language.

Feature 4:

The m[illegible] of the RISC

Feature 4:

The most important characteristic of the RISC processor is that more than 95% of instructions are executed with only one clock pulse, to the ~~cost~~ contrast to CISC instruction.

Feature 5

Because CISC has such a long number of instructions, each with so many different addressing modes, microinstruction (micro code) are used to implement them.

Microsoft Whiteboard

feature 6

RISC use load/store architecture. In CISC microprocessor data can be manipulated while it is still in memory.

For example, in instructions such as:

ADD Reg, Memory CISC

The processor must bring the contents of the external memory location into the CPU, add it to the contents of the register, then move the result back to the external memory location.

Microsoft Whiteboard

The problem is there might be a delay in accessing the data from external memory.

In RISC, instructions can only load from external memory into registers or store registers into external memory locations. There is no direct way of doing arithmetic and logic operations between a register and the contents of external memory locations.

LDI R_{20}, 0X60

L

Microsoft Whiteboard

```
LDI   R20, 0x60
LDI   R21, 0x46
ADD   R20, R21
```