Lec 02
05 August 2022

CISC Vs RISC

CISC	RISC
• instr. variable size	• fixed size (32-bit in ARM)
• Multiple fetch cycle	- one cycle for fetching

In CISC processors instructions are different in size.

In 8086 instructions are 1B, 2B, 3B, 4B, 5B and 6B.

In 8051 instructions are 1B, 2B, 3B.

But in RISC processors all instructions are in same size. In ARM all the instr. are 32bit (4Byte).

As the size of instructions in CISC processors, they might require more than one cycle for fetching for bigger instructions.

As the size of instructions in RISC processors are same so they all always take one cycle for fetching an instruction.

CISC	RISC
• operations on registers and memory	• Mainly registers.

* In CISC, all the operations can be done both on Registers and Memory.

In RISC exeepting load and store, all the arithmatic and logic operations are done only on ~~memory~~ registers.

* Operations on memory is slowers than operations on registes.

* operations occursing using memory kills pipelining. when a processing executing an instruction, it can at the same time it can not fetch the instruction from the memory. So pipelining gets disturbed.

- For example we can say 8 in 2086 has a pipelining system. It can do fetching and executing instructions simultaneously.

• Many addressing modes	• Less addressing modes
• Few registers	• Lots of registers

But it is true we when the operations are done through Registers.

If the fetching occurs directly from memory, processor gets involved in fetching and the execution gets a break in pipelining.

So overall true pipeling is found in RISC processors

In CISC one pipelining is worked with disturbance.

As CISC it works both register and memory, it requires more addressing modes.

RISC works only with registers. it requires less addressing modes.

In ARM7 has 32 regs.

In PIC18 has 4096 regs.

CISC	RISC
• Instruction set large	• Reduced
• Instruction set complex	• Simple instruction set
• Decoding complex	• Decoding simple
• Multiple execution cycle	• One execution cycle

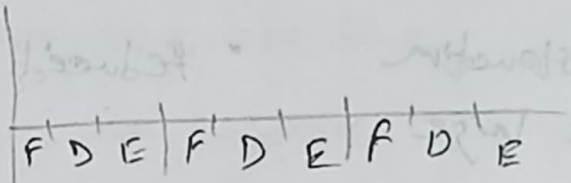
working with both Mem and regs it needs large no. of instruction set.
But in RISC it only works with regs, it does not need to have so many instructions.
So it is called reduced instruction set.

For the same reason above mentioned above, instrs in CISC is complex and instrs in RISC is simple.

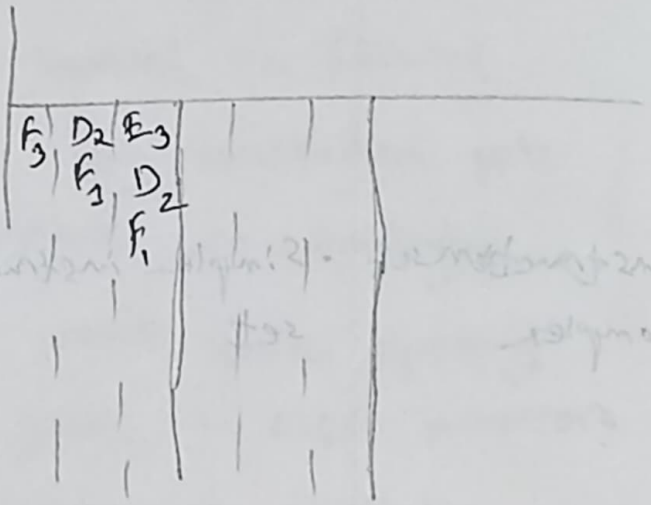
As there are more instructions in CISC, there requires many opcodes. So many opcodes makes the decoding complex.
For opposite reason the decoding is simple in RISC.

CISC	RISC
• Para pipelining	• Excellent pipiling

Normal processor:



pipelined processors (3 stage)



In pipelining, when the fetching, decoding, executing for every instructions are same size. it gives the perfect image of pipeling. and that is possible for RISC processors.

In CISC processors different instructions take different time for execution. So fetching, decoding and execution are not same for every instruction.

CISC	RISC
• Flexible but slower	• rigid but faster
• Microprocessor	• Microcontroller

So it creats bubble in pipelining.

It gives a poor image of pipelining.

For general purpose use, microprocessor is used for flexibility. and to get flexibility MP is based on CISC.

On them other hand in applinces, performance is first priority and operations are fixed. So there is used microcontrollers, and most of them are based on RISC.

example (ARM7, PIC18)