RISC Vs CISC

دانشگاه آزاد اسلامی

Islamic Azad University

Subject: RISC Vs CISC

Professor: Amir Masoud Bidgoli

by: Ghazal Ghaderiniri

1396 Azar

Contents

- What are RISC and CISC processors?
- What are CISC and RISC used for?
- History of CISC and RISC
- CISC Architecture
  - Advantages of CISC Architecture
  - Disadvantages of CISC Architecture
  - Examples of CISC processor
- RISC Architecture
  - The pipelining technique
  - Examples of RISC processors
  - Advantages of RISC Architecture
  - Disadvantages of RISC Architecture
- RISC Vs CISC Table
- RISC VS CISC Architecture Table
- Is CISC faster than RISC?
- Summation
- Modern Day Advancement
- References

What are RISC and CISC processors?

- Most desktop or laptop computers use CISC (complex instruction set computing) architecture made by Intel or AMD.

- Smartphones and tablets use RISC (reduced instruction set computing) ARM architecture.

- The key differences between the two CPUs are: Instructions - RISC has fewer instructions than CISC.

What are CISC and RISC used for?

A complex instruction set computer (CISC /pronounce as ˈsisk'/) is a computer where single instructions can execute several low-level operations (such as a load from memory, an arithmetic operation, and a memory store) or are capable of multi-step operations or addressing modes within single instructions, as its name ...

History of CISC and RISC

- 1950s IBM instituted a research program
- 1964 Release of System/360
- Mid-1970s improved measurement tools demonstrated on CISC
- 1975 801 project initiated at IBM’s Watson Research Center
- 1979 32-bit RISC microprocessor (801) developed led by Joel Birnbaum
- 1984 MIPS developed at Stanford, as well as projects done at Berkeley
- 1988 RISC processors had taken over high-end of the workstation market
- Early 1990s IBM’s POWER (Performance Optimization With Enhanced RISC) architecture introduced w/ the RISC System/6k
- AIM (Apple, IBM, Motorola) alliance formed, resulting in PowerPC

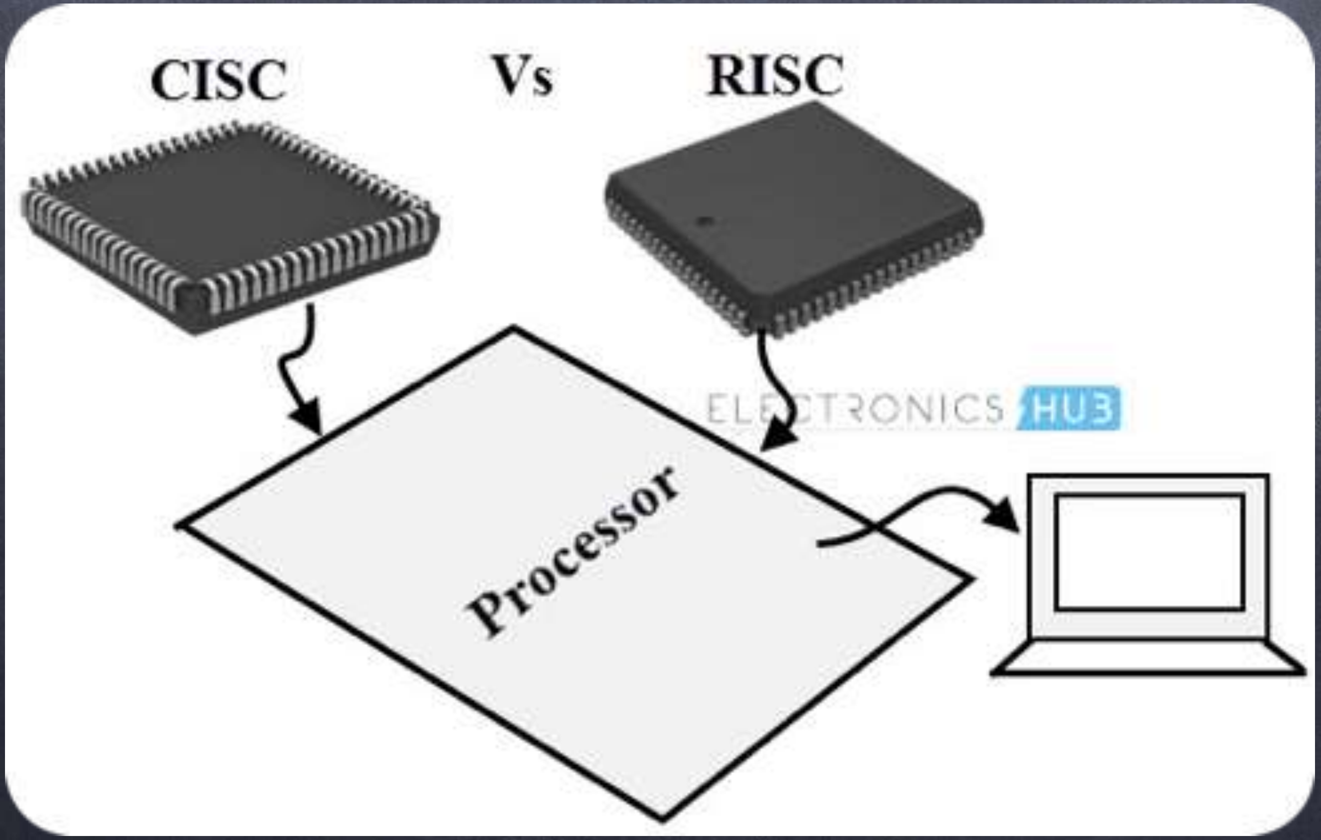
Instruction set architecture is a part of processor architecture, which is necessary for creating machine level programs to perform any mathematical or logical operations. Instruction set architecture acts as an interface between hardware and software. It prepares the processor to respond to the commands like execution, deleting etc given by the user.

The performance of the processor is defined by the instruction set architecture designed in it. As both software and hardware are required for functioning of a processor, there is dilemma in deciding which should play a major role. Major firms like Intel argues that hardware should play a major role than software. While, Apple’s argument is that software should play a major role in processors architecture.

The two major instruction sets architectures

1) CISC => Complex Instruction Set Computing

2) RISC => Reduced Instruction Set Computing



What is CISC?

CISC is an acronym for Complex Instruction Set Computer and are chips that are easy to program and which make efficient use of memory. Since the earliest machines were programmed in assembly language and memory was slow and expensive, the CISC philosophy made sense, and was commonly implemented in such large computers as the PDP-11 and the DECsystem 10 and 20 machines.

Most common microprocessor designs such as the Intel 80x86 and Motorola 68K series followed the CISC philosophy.

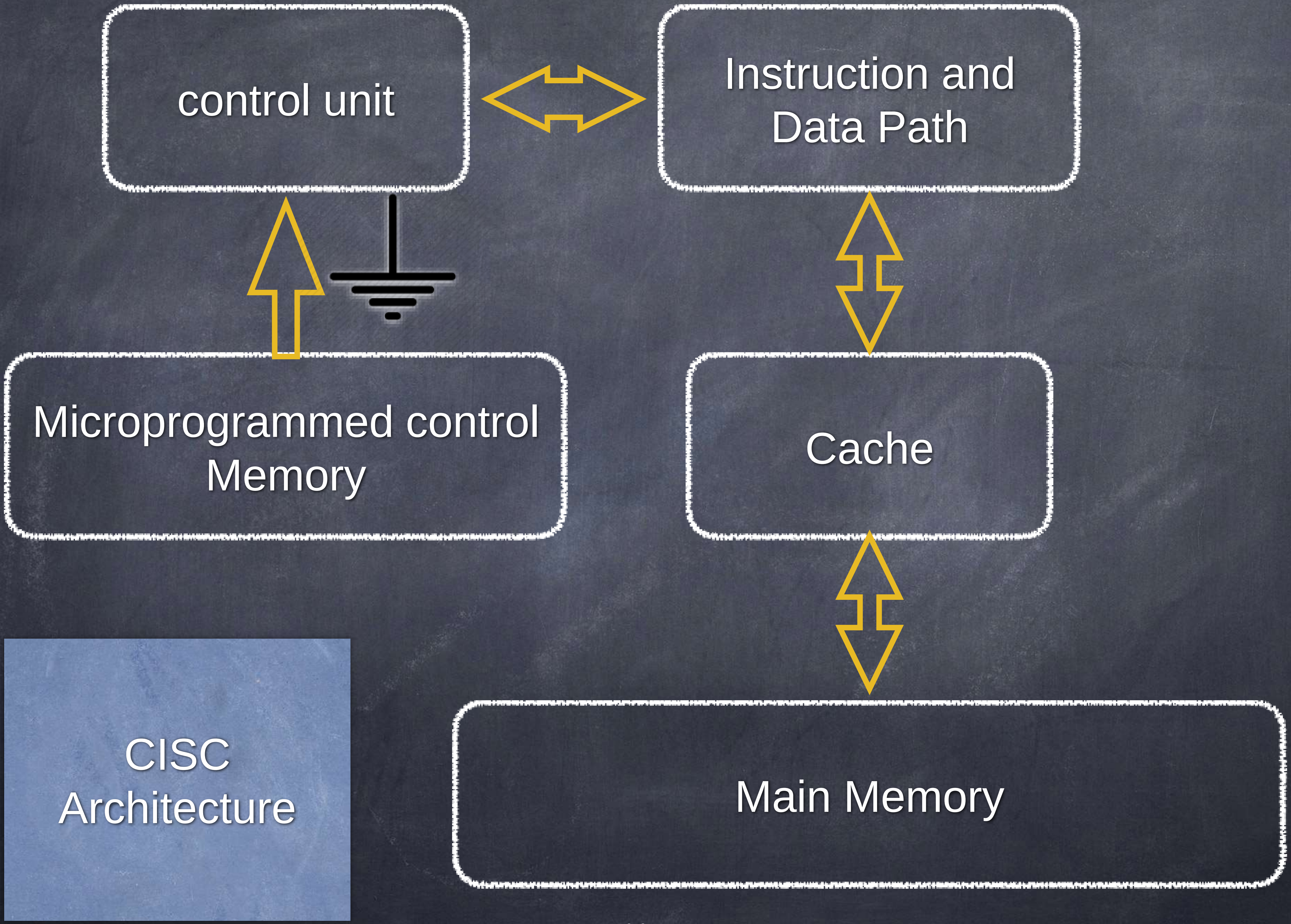
But recent changes in software and hardware technology have forced a re-examination of CISC and many modern CISC processors are hybrids, implementing many RISC principles.

CISC was developed to make compiler development simpler. It shifts most of the burden of generating machine instructions to the processor. For example, instead of having to make a compiler write long machine instructions to calculate a square-root, a CISC processor would have a built-in ability to do this.

CISC Architecture

In the early days machines were programmed in assembly language and the memory access is also slow. To calculate complex arithmetic operations, compilers have to create long sequence of machine code.

This made the designers to build an architecture , which access memory less frequently and reduce burden to compiler. Thus this lead to very power full but complex instruction set.



control unit

Instruction and
Data Path

Microprogrammed control
Memory

Cache

Main Memory

CISC
Architecture

CISC architectures directly use the memory, instead of a register file. The above figure shows the architecture of CISC with micro programmed control and cache memory.

This architecture uses cache memory for holding both data and instructions. Thus, they share the same path for both instructions and data.

CISC has instructions with variable length format. Thus, the number of clock cycles required to execute the instructions may be varied.

Instructions in CISC are executed by micro program which has sequence of microinstructions.

Let us see an example: Addition of two numbers can be calculated as follows.

ADD 1:1 , 2:2

- The above shown instruction is divided into a number of micro instructions.
- Initally,it stores the data in two separate registers then decodes finally, execute
- This result is stored in MBR register.
- After this the two registers were cleared automatically.
- Thus to execute all these steps a complex circuitry is required.

Advantages of CISC Architecture

- Microprogramming is easy to implement and much less expensive than hard wiring a control unit.
- It is easy to add new commands into the chip without changing the structure of the instruction set as the architecture uses general-purpose hardware to carry out commands.
- This architecture makes the efficient use of main memory since the complexity (or more capability) of instruction allows to use less number of instructions to achieve a given task.
- The compiler need not be very complicated, as the micro program instruction sets can be written to match the constructs of high level languages.

Disadvantages of CISC Architecture

- A new or succeeding versions of CISC processors consists early generation processors in their subsets (succeeding version). Therefore, chip hardware and instruction set became complex with each generation of the processor.
- The overall performance of the machine is reduced because of slower clock speed.
- The complexity of hardware and on-chip software included in CISC design to perform many functions.

Examples of CISC processor

1 IBM 370/168

2 Intel 80486

3 VAX 11/780

What does RSIC mean?

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).

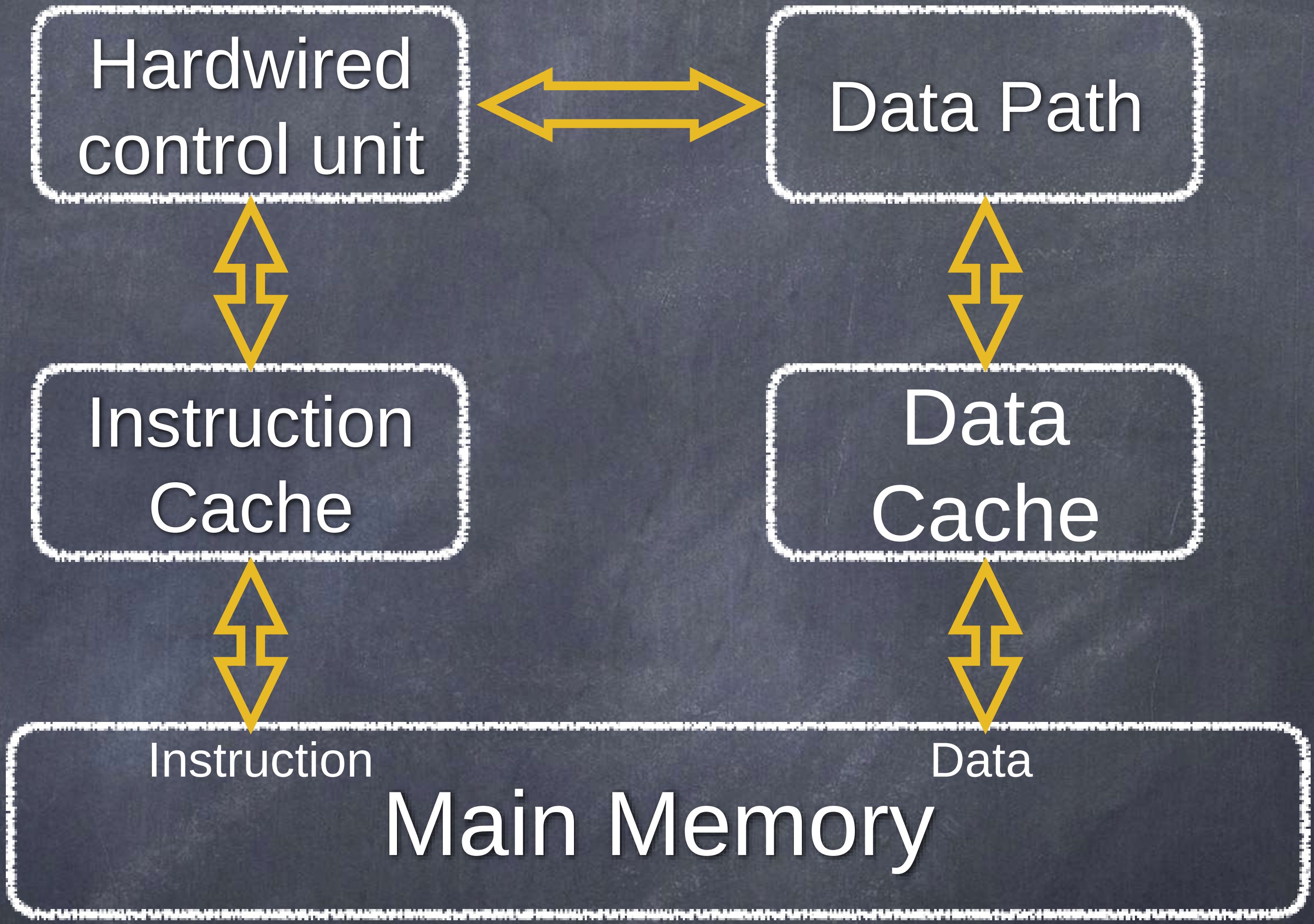
RISC Architecture

Although CISC reduces usage of memory and compiler, it requires more complex hardware to implement the complex instructions.

In RISC architecture, the instruction set of processor is simplified to reduce the execution time. It uses small and highly optimized set of instructions which are generally register to register operations.

The speed of the execution is increased by using smaller number of instructions .This uses pipeline technique for execution of any instruction.

The figure is the architecture of RISC processor, which uses separate instruction and data caches and their access paths also different. There is one instruction per machine cycle in RISC processor.



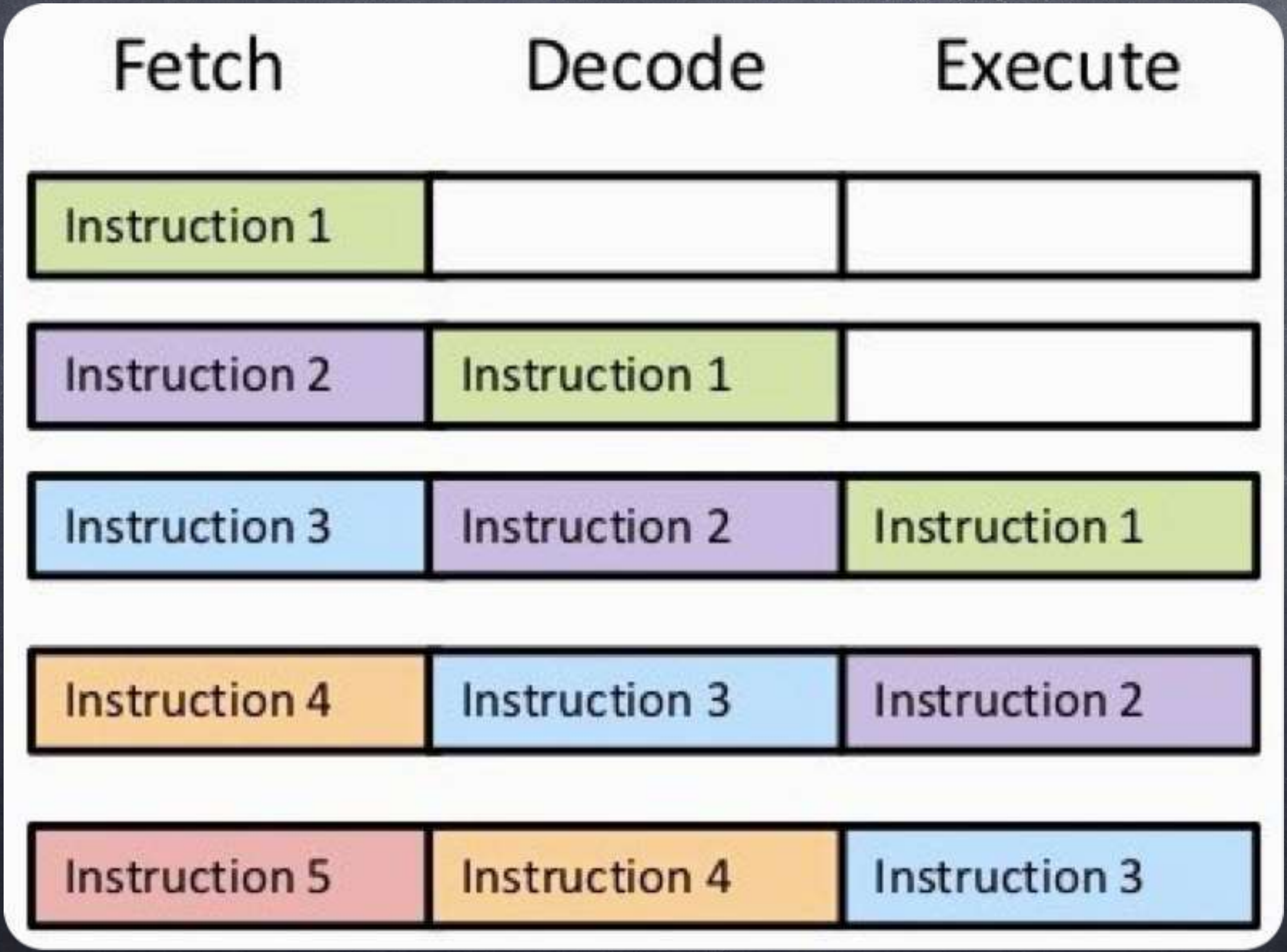
The pipelining technique

The pipelining technique allows the processor to work on different steps of instruction like fetch, decode and execute instructions at the same time.

Generally, execution of second instruction is started, only after the completion of the first instruction. But in pipeline technique, each instruction is executed in number of stages simultaneously.

When the first stage of first instruction is completed, next instruction is enters into the fist stage. This process continuous until all the instructions are executed.

execution of instructions in pipelining technique.



Fetch	Decode	Execute
Instruction 1		
Instruction 2	Instruction 1	
Instruction 3	Instruction 2	Instruction 1
Instruction 4	Instruction 3	Instruction 2
Instruction 5	Instruction 4	Instruction 3

RISC Pipelines

A RISC processor pipeline operates in much the same way, although the stages in the pipeline are different. While different processors have different numbers of steps, they are basically variations of these five, used in the MIPS R3000 processor:

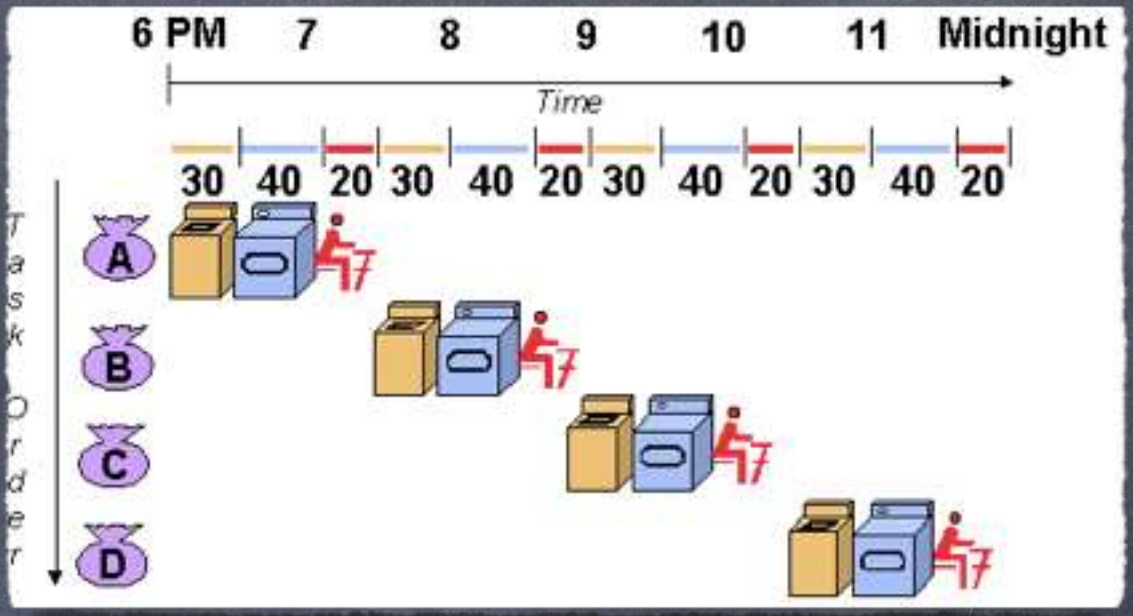
- fetch instructions from memory

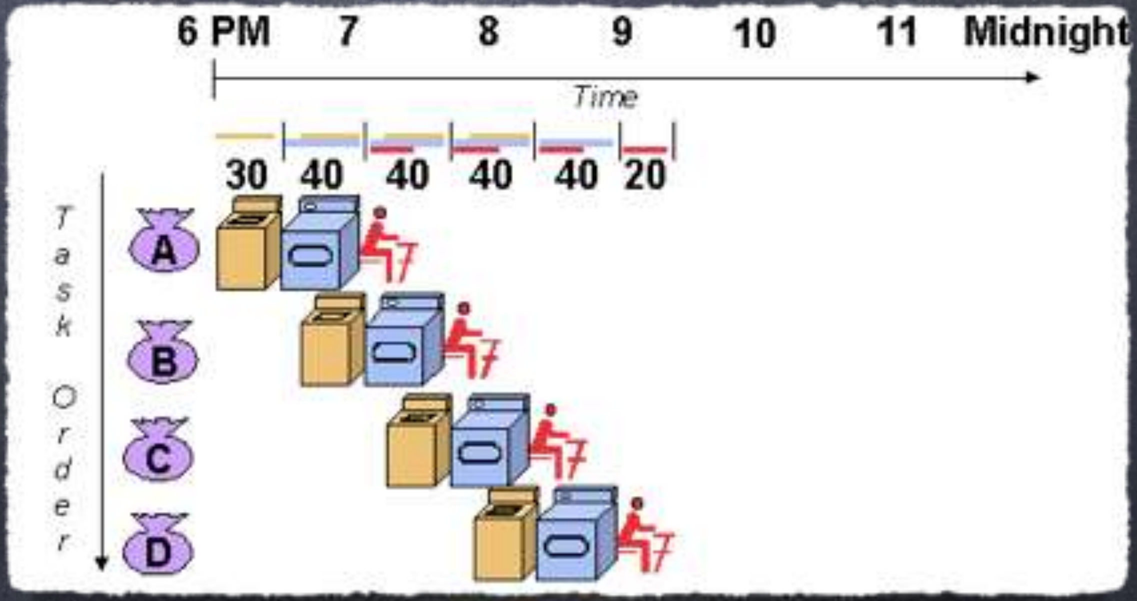
- read registers and decode the instruction

- execute the instruction or calculate an address

- access an operand in data memory

- write the result into a register





Let us see an example : Adding of two numbers can be as shown below.

  - LOAD A, 2:3
  - LOAD B, 5:2
  - SUM A, B
  - STORE 2:3, A

- From the above it can be seen that Addition requires four steps in RISC processor
- Though many instructions were required in RISC, time taken by these instructions to complete execution is same as time required to execute “ADD” in CISC.
- The design of the control unit is also simple due to the limited number of instructions.

Examples of RISC processors:

This architecture include alpha, AVR, ARM, PIC, PA-RISC, and power architecture.

Advantages of RISC Architecture

- The performance of RISC processors is often two to four times than that of CISC processors because of simplified instruction set.
- This architecture uses less chip space due to reduced instruction set. This makes to place extra functions like floating point arithmetic units or memory management units on the same chip.
- The per-chip cost is reduced by this architecture that uses smaller chips consisting of more components on a single silicon wafer.
- RISC processors can be designed more quickly than CISC processors due to its simple architecture.
- The execution of instructions in RISC processors is high due to the use of many registers for holding and passing the instructions as compared to CISC processors.

Disadvantages of RISC Architecture

- The performance of a RISC processor depends on the code that is being executed. The processor spends much time waiting for first instruction result before it proceeds with next subsequent instruction, when a compiler makes a poor job of scheduling instruction execution.

- RISC processors require very fast memory systems to feed various instructions. Typically, a large memory cache is provided on the chip in most RISC based systems.

RISC Vs CISC

CISC	RISC
CISC architecture gives more importance to hardware	RISC architecture gives more importance to Software
complex instructions	Reduced instructions
it access memory directly	It requires registers
Coding in CISC processor is simple.	coding in RISC processor requires more number of lines
As it consists of complex instructions , it take multiple cycles to execute	It consists of simple instructions that take single cycle to execute.
Complexity lies in microprogram	complexity lies in compiler

RISC VS CISC Architecture

RISC	CISC
Single - clock	Multi - clock
Reduce instructions	Complex instructions
No microcode	Complicated microcode
Data exilicitly accessed	Memory to memory operations
Easier to validate	Difficult to validate
larger code sizes	smaller code sizes
Low cycles /second	High cycles / second
More transistors on memory registers	More transistors fore complex instructions
Pipelining friendly	Compiler friendly
Emphasis on software	Emphasis on hardware

Is CISC faster than RISC?

Modern processors are pretty much all RISC. Even CISC instruction sets (x86-64) are translated to RISC microcode on chip prior to execution. ... RISC CPUs generally run at faster clock speeds than CISC because max clock period is dictated by the slowest step of the pipeline (more complex instructions are slower)

Summation

As memory speed increased, and high-level languages displaced assembly language, the major reasons for CISC began to disappear, and computer designers began to look at ways computer performance could be optimized beyond just making faster hardware.

One of their key realizations was that a sequence of simple instructions produces the same results as a sequence of complex instructions, but can be implemented with a simpler (and faster) hardware design. (Assuming that memory can keep up.) RISC (Reduced Instruction Set Computers) processors were the result.

CISC and RISC implementations are becoming more and more alike. Many of today’s RISC chips support as many instructions as yesterday's CISC chips. And today's CISC chips use many techniques formerly associated with RISC chips.

To some extent, the argument is becoming moot because CISC and RISC implementations are becoming more and more alike. Many of today's RISC chips support as many instructions as yesterday's CISC chips. And today's CISC chips use many techniques formerly associated with RISC chips.

Modern Day Advancement

CISC and RISC Convergence
State of the art processor technology has changed significantly since RISC chips were first introduced in the early '80s. Because a number of advancements are used by both RISC and CISC processors, the lines between the two architectures have begun to blur. In fact, the two architectures almost seem to have adopted the strategies of the other. Because processor speeds have increased, CISC chips are now able to execute more than one instruction within a single clock. This also allows CISC chips to make use of pipelining. With other technological improvements, it is now possible to fit many more transistors on a single chip.

This gives RISC processors enough space to incorporate more complicated, CISC-like commands. RISC chips also make use of more complicated hardware, making use of extra function units for superscalar execution. All of these factors have led some groups to argue that we are now in a "post-RISC" era, in which the two styles have become so similar that distinguishing between them is no longer relevant. However, it should be noted that RISC chips still retain some important traits. RISC chips stricly utilize uniform, single-cycle instructions. They also retain the register-to-register, load/store architecture. And despite their extended instruction sets, RISC chips still have a large number of general purpose registers.

References

- https://www.electronicshub.org/risc-and-cisc-architectures/
- http://www.edgefxkits.com/blog/what-is-risc-and-cisc-architecture/
- http://cse.stanford.edu/class/sophomore-college/projects-00/risc/
- http://www.visionengineer.com/comp/why_cisc.shtml
- http://www.visionengineer.com/comp/why_risc.shtml
- http://www.embedded.com/story/OEG20030205S0025
- http://encyclopedia.laborlawtalk.com/PowerPC
- http://www.sunderland.ac.uk/~ts0jti/comparch/ciscrisc.htm
- http://www.heyrick.co.uk/assembler/riscvcisc.html
- http://www.aallison.com/history.htm