Overview of Modern Processor Architecture

Parallel Execution Capabilities

JinHui Lin

ShenZhen University

2025-06-02

What Affects CPU Performance?

Instruction Count to be executed

This depends on:

- Program objectives
- ISA(Instruction Set Architecture)
- Code quality
- Programming language used
- Compiler behavior
- etc.

What Affects CPU Performance?

Instruction Count to be executed

This depends on:

- Program objectives
- ISA(Instruction Set Architecture)
- Code quality
- Programming language used
- Compiler behavior
- etc.

Not within the scope of today's discussion

What Affects CPU Performance?

Clock Frequency

This depends on:

- Front-end CPU design
- Back-end design
- Manufacturing process
- etc.

What Affects CPU Performance?

Clock Frequency

This depends on:

- Front-end CPU design
- Back-end design
- Manufacturing process
- etc.

Not within the scope of today's discussion

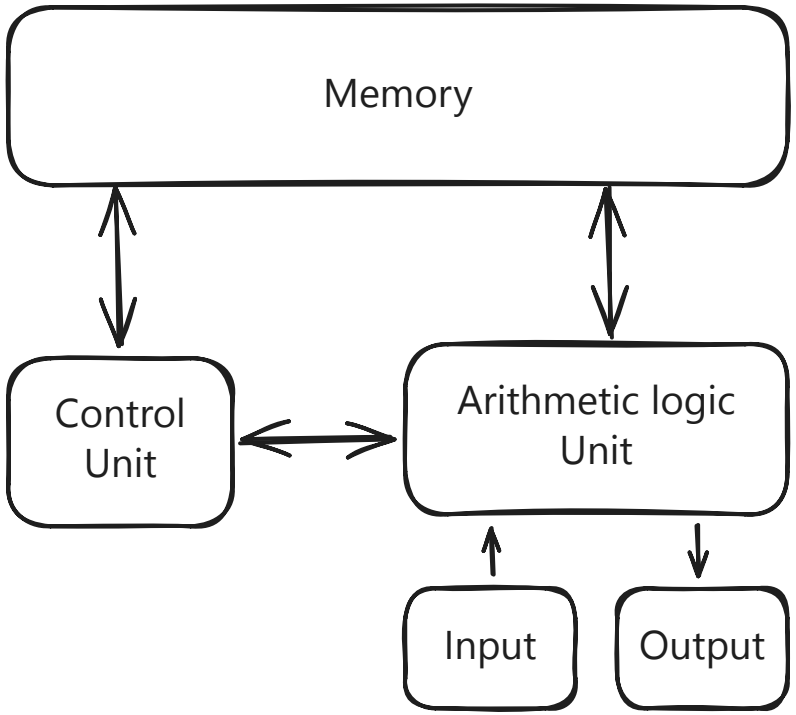
What Affects CPU Performance?

IPC (Instructions Per Cycle)

The number of instructions that can be executed per cycle

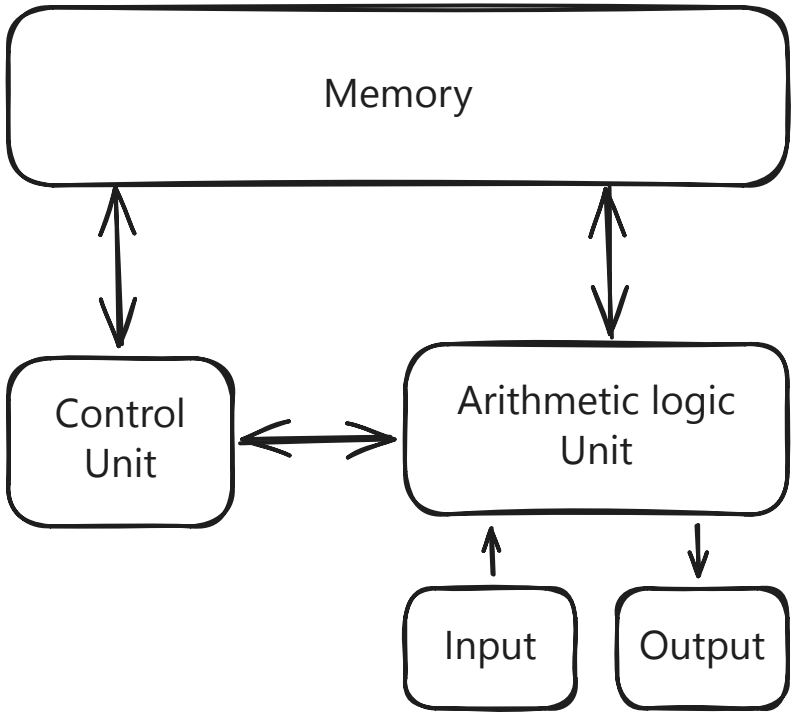
This essentially equates to—“the CPU’s ability to parallelize serial instructions”

Basic Computer Architecture



Classical Von Neumann Computer Architecture

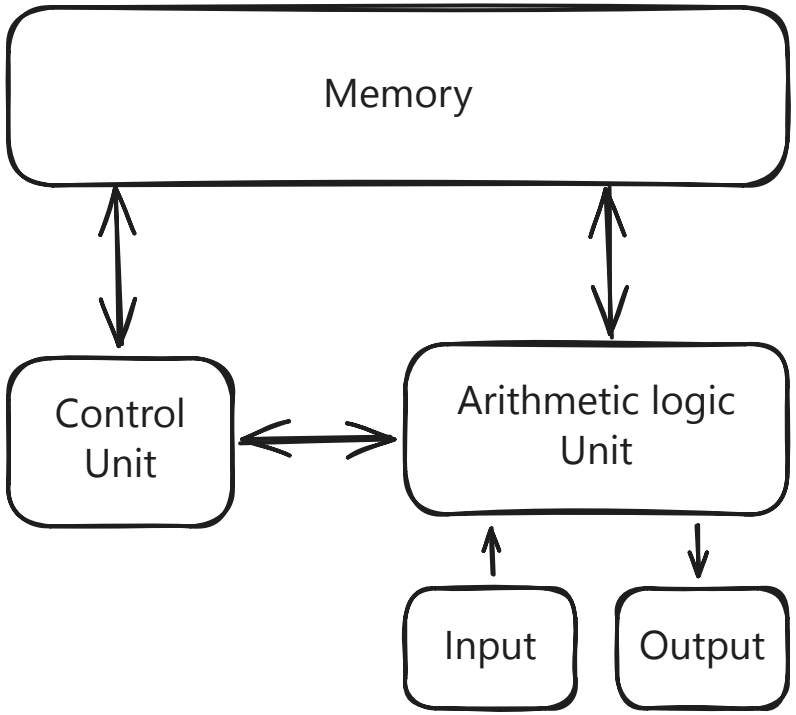
Basic Computer Architecture



Classical Von Neumann Computer Architecture

Even today, this architecture can still be used to explain computer architecture.

Basic Computer Architecture

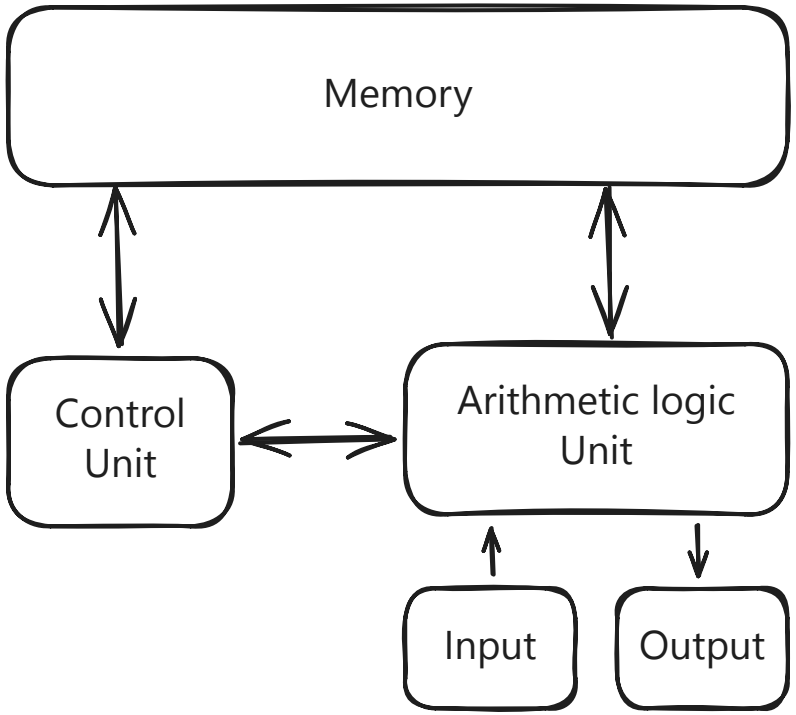


Classical Von Neumann Computer Architecture

Even today, this architecture can still be used to explain computer architecture.

Although there are minor differences in some areas (such as internal registers, cache, MMIO, etc.)

Basic Computer Architecture



Classical Von Neumann Computer Architecture

Even today, this architecture can still be used to explain computer architecture.

Although there are minor differences in some areas (such as internal registers, cache, MMIO, etc.)

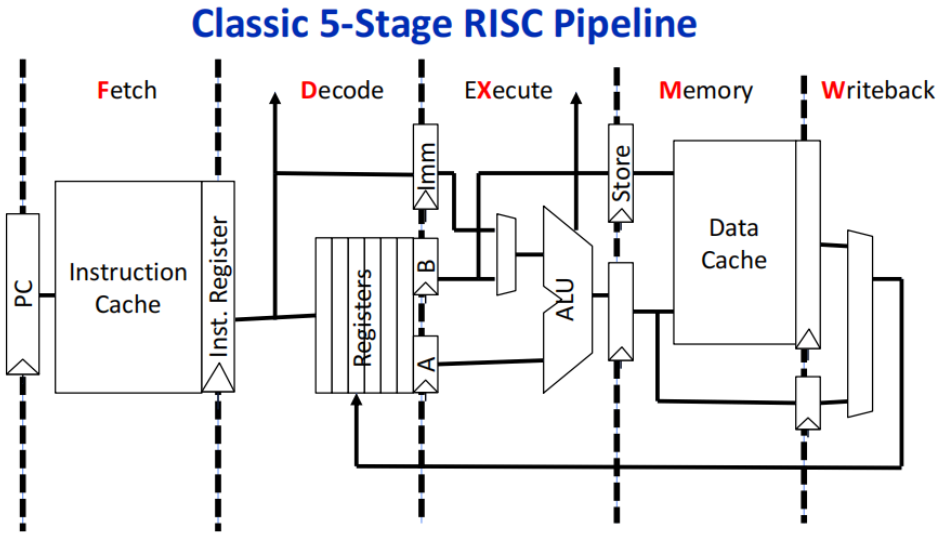
Its operation process is essentially a cycle of “fetch->execute”

Pipelining

Pipelining is a classic CPU parallelization acceleration technique. By dividing CPU execution into multiple stages and executing these stages simultaneously, it increases frequency without significantly reducing IPC, thereby improving performance.

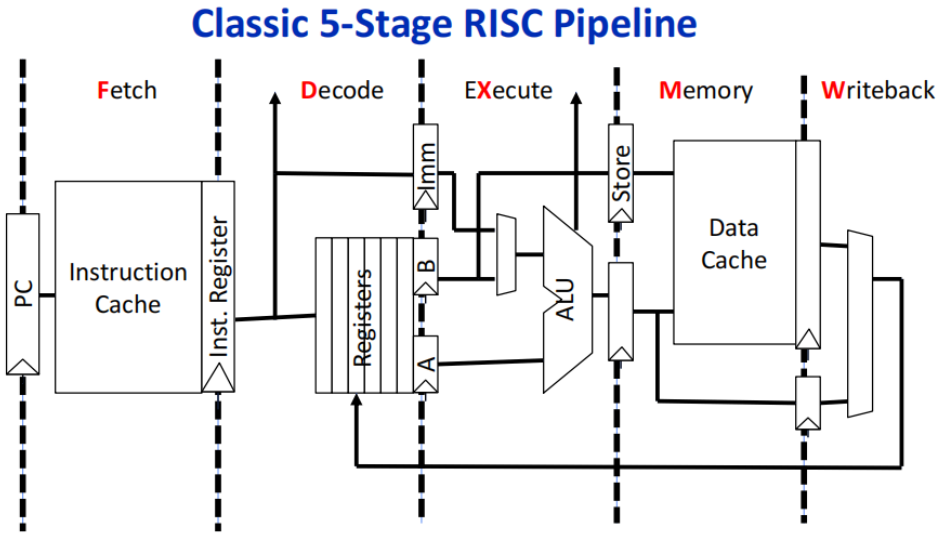
Modern CPUs rarely do not use pipelining technology. Even the microcontroller (MCU) in your washing machine likely has at least a two-stage pipeline (Arm CortexM0+).

Pipelining



A classic pipeline stage division

Pipelining

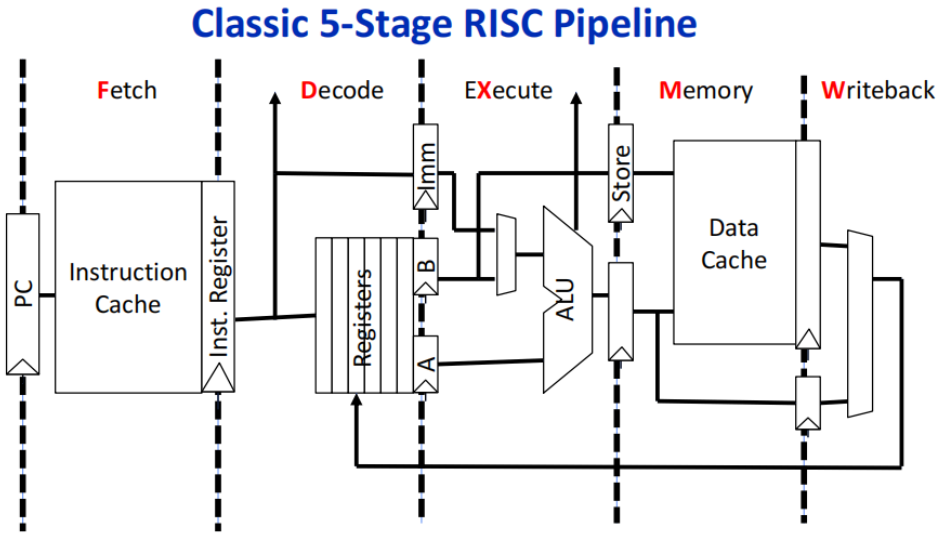


A classic pipeline stage division

However, pipelining cannot be executed arbitrarily.

Some special instruction and instrcution sequence may have data dependencies, which can lead to data hazards.

Pipelining



A classic pipeline stage division

However, pipelining cannot be executed arbitrarily.

Some special instruction and instrcution sequence may have data dependencies, which can lead to data hazards.

The solution is through speculative execution, guessing the jump result, and flushing the pipeline if incorrect.

Branch Prediction

If speculative execution is correct, it can significantly improve performance, so the accuracy of speculation becomes crucial.

Branch Prediction

If speculative execution is correct, it can significantly improve performance, so the accuracy of speculation becomes crucial.

Branch prediction algorithms:

- Static branch prediction
  Determines branch prediction direction at compile time.
- Dynamic branch prediction
  Records the historical jump results of branch instructions and uses these historical results to predict the next jump.

Out-of-Order Execution

By reordering instructions, data hazards are avoided in advance.

Out-of-Order Execution

By reordering instructions, data hazards are avoided in advance.

Basic idea of out-of-order execution:

- Add an instruction queue
- Reorder it to avoid hazards
- After execution, write back in the original order

Superscalar

Superscalar refers to the CPU's ability to execute multiple instructions in the same cycle.

Superscalar

Superscalar refers to the CPU's ability to execute multiple instructions in the same cycle.

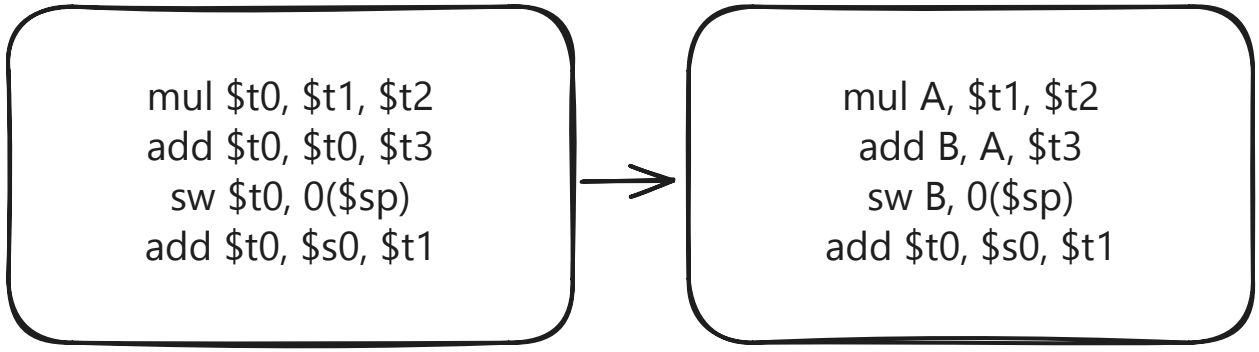
Basic idea of superscalar:

- Fetch multiple instructions simultaneously and push them into the instruction queue
- Based on instruction dependencies, push non-dependent instructions into the backend for simultaneous execution
- After execution, write back in the original order

Register Renaming

Register renaming is another way to avoid hazards.

The basic idea is shown in the figure:



Register Renaming

Tomasulo algorithm:

- Maintain a renaming table for each logical register
- When an instruction needs to write to a register, allocate a new physical register
- Update the renaming table, mapping the logical register to the new physical register
- Subsequent instructions reading this logical register will use the latest physical register
- When an instruction completes, release the physical register that is no longer needed

Summary

Key optimization technologies in modern processor architecture:

Summary

Key optimization technologies in modern processor architecture:

- Pipelining: Divides instruction execution into multiple stages for parallel execution
- Branch prediction: Reduces pipeline stalls by predicting branch jump direction
- Out-of-order execution: Reorders instructions to avoid data hazards
- Superscalar: Executes multiple instructions in the same cycle
- Register renaming: Avoids register hazards through physical register mapping

Thank You

If you have any questions, please feel free to discuss.