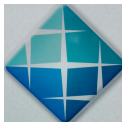


Programação de Microcontroladores

Aula 02

Evandro J.R. Silva¹

¹ Bacharelado em Ciência da Computação
Estácio Teresina



Sumário

- 1 Introdução
- 2 CISC
- 3 RISC
- 4 CISC x RISC
- 5 Arquitetura de Memória
 - Arquitetura de Von Neumann
 - Arquitetura de Harvard
- 6 Arquitetura dos Microcontroladores
- 7 FIM

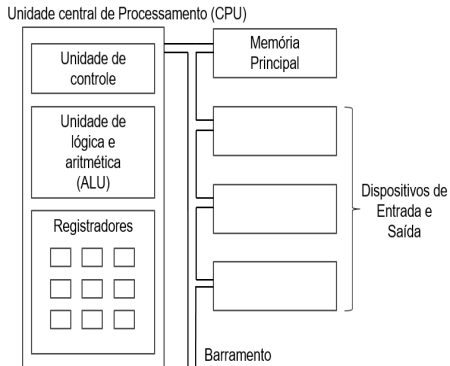
Introdução

Introdução

- Os diversos microcontroladores desenvolvidos ao longo dos anos, assim como os microprocessadores, podem ser distintos em relação ao conjunto de instruções e a arquitetura de memória.
- Portanto, microcontroladores podem ser do tipo RISC ou CISC, ou de arquitetura Harvard ou Von Neumann.

Introdução

Recordando: organização básica de um computador

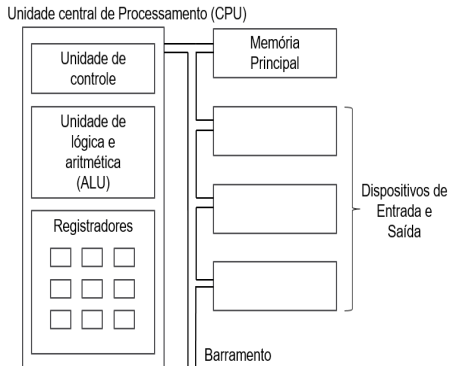


Introdução

Recordando: organização básica de um computador

■ Unidade de Controle

- Busca instruções na Memória Principal e determina o seu tipo: registrador-registrador / registrador-memória



Introdução

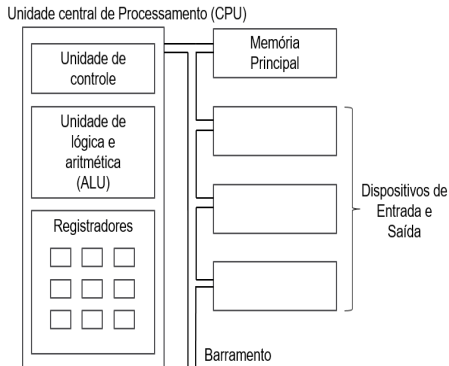
Recordando: organização básica de um computador

■ Unidade de Controle

- Busca instruções na Memória Principal e determina o seu tipo: registrador-registrador / registrador-memória

■ ALU

- Efetua as operações aritméticas e booleanas para executar uma instrução.



Introdução

Recordando: organização básica de um computador

■ Unidade de Controle

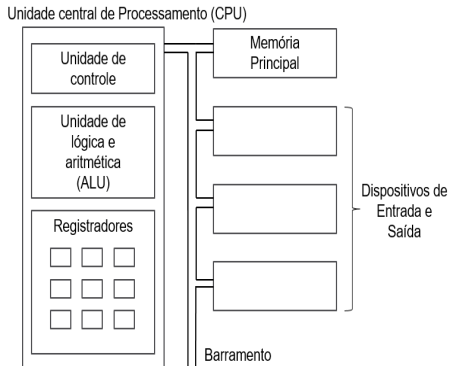
- Busca instruções na Memória Principal e determina o seu tipo: registrador-registrador / registrador-memória

■ ALU

- Efetua as operações aritméticas e booleanas para executar uma instrução.

■ Registradores

- Pequenas memórias de alta velocidade utilizadas para armazenar valores temporários e para controle de informações.



Introdução

- Execução de uma instrução de maneira simples:

Introdução

- Execução de uma instrução de maneira simples:
 - 1 Trazer a próxima instrução da memória;

Introdução

- Execução de uma instrução de maneira simples:
 - 1 Trazer a próxima instrução da memória;
 - 2 Atualizar o registrador com a próxima instrução a ser buscada;

Introdução

- Execução de uma instrução de maneira simples:
 - 1 Trazer a próxima instrução da memória;
 - 2 Atualizar o registrador com a próxima instrução a ser buscada;
 - 3 Determinar o tipo de instrução que foi buscada;

Introdução

- Execução de uma instrução de maneira simples:
 - 1 Trazer a próxima instrução da memória;
 - 2 Atualizar o registrador com a próxima instrução a ser buscada;
 - 3 Determinar o tipo de instrução que foi buscada;
 - 4 Se a instrução utilizar uma *palavra* na memória, determinar onde ela está;

Introdução

- Execução de uma instrução de maneira simples:
 - 1 Trazer a próxima instrução da memória;
 - 2 Atualizar o registrador com a próxima instrução a ser buscada;
 - 3 Determinar o tipo de instrução que foi buscada;
 - 4 Se a instrução utilizar uma *palavra* na memória, determinar onde ela está;
 - 5 Trazer a *palavra* para um registrador, se necessário;

Introdução

- Execução de uma instrução de maneira simples:
 - 1 Trazer a próxima instrução da memória;
 - 2 Atualizar o registrador com a próxima instrução a ser buscada;
 - 3 Determinar o tipo de instrução que foi buscada;
 - 4 Se a instrução utilizar uma *palavra* na memória, determinar onde ela está;
 - 5 Trazer a *palavra* para um registrador, se necessário;
 - 6 Executar a instrução;

Introdução

- Execução de uma instrução de maneira simples:
 - 1 Trazer a próxima instrução da memória;
 - 2 Atualizar o registrador com a próxima instrução a ser buscada;
 - 3 Determinar o tipo de instrução que foi buscada;
 - 4 Se a instrução utilizar uma *palavra* na memória, determinar onde ela está;
 - 5 Trazer a *palavra* para um registrador, se necessário;
 - 6 Executar a instrução;
 - 7 Retornar à etapa 1.

Introdução

- Com a popularização dos computadores
 - Uma mesma arquitetura para uma *família* de computadores

Introdução

- Com a popularização dos computadores
 - Uma mesma **arquitetura** para uma *família* de computadores
 - Mesmo conjunto de instruções;
 - Máquinas diferentes capazes de executar os mesmos programas!

Introdução

- Com a popularização dos computadores
 - Uma mesma arquitetura para uma *família* de computadores
- Famílias mais conhecidas atualmente
 - Intel x86 e x86-64
 - AMD Zen
 - ARM

CISC

CISC

- Até o fim da década de 70
 - As memórias eram muito lentas;
 - Ler instruções era lento;
 - Não há como evitar a busca de instrução;
 - Solução?

CISC

- Até o fim da década de 70
 - As memórias eram muito lentas;
 - Ler instruções era lento;
 - Não há como evitar a busca de instrução;
 - Solução?
 - Reduzir a busca de instruções!

CISC

- CISC : *Complex Instruction Set Computer* ou *Computador com Conjunto de Instruções Complexo*
 - Diversos modos de endereçamento;
 - Registradores de uso geral com vários modos de acesso;
 - Uso de micro-instruções complexas

CISC

- CISC : *Complex Instruction Set Computer* ou *Computador com Conjunto de Instruções Complexo*
 - Diversos modos de endereçamento;
 - Registradores de uso geral com vários modos de acesso;
 - Uso de micro-instruções **complexas**
 - Uma única instrução poderia realizar várias tarefas;

CISC

- CISC : *Complex Instruction Set Computer* ou *Computador com Conjunto de Instruções Complexo*
 - Diversos modos de endereçamento;
 - Registradores de uso geral com vários modos de acesso;
 - Uso de micro-instruções **complexas**
 - Uma única instrução poderia realizar várias tarefas;
 - Instruções parecidas com as programadas em mais alto nível (simplifica a tarefa do compilador);

CISC

- CISC : *Complex Instruction Set Computer* ou *Computador com Conjunto de Instruções Complexo*
 - Diversos modos de endereçamento;
 - Registradores de uso geral com vários modos de acesso;
 - Uso de micro-instruções **complexas**
 - Uma única instrução poderia realizar várias tarefas;
 - Instruções parecidas com as programadas em mais alto nível (simplifica a tarefa do compilador);
 - Tamanho do código é reduzido, logo, menos memória necessária para armazenar programa;

CISC

- CISC : *Complex Instruction Set Computer* ou *Computador com Conjunto de Instruções Complexo*
 - Diversos modos de endereçamento;
 - Registradores de uso geral com vários modos de acesso;
 - Uso de micro-instruções **complexas**
 - Uma única instrução poderia realizar várias tarefas;
 - Instruções parecidas com as programadas em mais alto nível (simplifica a tarefa do compilador);
 - Tamanho do código é reduzido, logo, menos memória necessária para armazenar programa;
 - Uma menor quantidade de instruções teoricamente implica em um menor tempo de execução.

CISC

■ PORÉM:

CISC

- PORÉM:
 - As instruções complexas são pouco exploradas;

CISC

■ PORÉM:

- As instruções complexas são pouco exploradas;
- Otimizar o código é muito mais difícil com instruções complexas do que com instruções simples;

CISC

■ PORÉM:

- As instruções complexas são pouco exploradas;
- Otimizar o código é muito mais difícil com instruções complexas do que com instruções simples;
- Na prática, um programa CISC não é notavelmente menor que programas escritos com instruções simples;

RISC

RISC

- RISC: *Reduced Instruction Set Computer* ou *Computador com Conjunto de Instruções Reduzido*

RISC

- RISC: *Reduced Instruction Set Computer* ou *Computador com Conjunto de Instruções Reduzido*
 - Formato de instruções simples;

RISC

- RISC: *Reduced Instruction Set Computer* ou *Computador com Conjunto de Instruções Reduzido*
 - Formato de instruções simples;
 - Uma instrução por ciclo;

RISC

- RISC: *Reduced Instruction Set Computer* ou *Computador com Conjunto de Instruções Reduzido*
 - Formato de instruções simples;
 - Uma instrução por **ciclo**;
 - Processo de passar dois operandos pela ALU e armazenar o resultado;
 - Processadores atuais operam numa taxa entre 3 e 4 GHz, ou seja, entre 3 e 4 bilhões de ciclos por segundo!

RISC

- RISC: *Reduced Instruction Set Computer* ou *Computador com Conjunto de Instruções Reduzido*
 - Formato de instruções simples;
 - Uma instrução por ciclo;

- Operações registrador-registrador (possibilitado pela quantidade maior de registradores gerais);

RISC

- RISC: *Reduced Instruction Set Computer* ou *Computador com Conjunto de Instruções Reduzido*
 - Formato de instruções simples;
 - Uma instrução por ciclo;

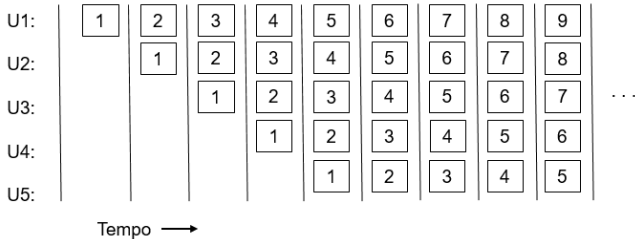
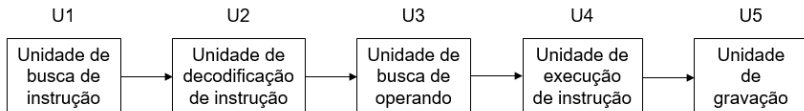
- Operações registrador-registrador (possibilitado pela quantidade maior de registradores gerais);
- Modos de endereçamento simples. Além disso, o acesso à memória apenas com as instruções **load** e **store**;

RISC

- RISC: *Reduced Instruction Set Computer* ou *Computador com Conjunto de Instruções Reduzido*
 - Formato de instruções simples;
 - Uma instrução por ciclo;

 - Operações registrador-registrador (possibilitado pela quantidade maior de registradores gerais);
 - Modos de endereçamento simples. Além disso, o acesso à memória apenas com as instruções **load** e **store**;
 - Mais facilidade na utilização de *pipeline*.

RISC



CISC x RISC

CISC x RISC

- Não necessariamente a arquitetura RISC é sempre a melhor escolha!

CISC x RISC

- Não necessariamente a arquitetura RISC é sempre a melhor escolha!
- Comparação de desempenho: cálculo de *speedup*
 - $S = 100 * (T_C - T_R) / T_R$, onde
 - S — *Speedup*;
 - T_C — Tempo de execução com CISC;
 - T_R — Tempo de execução com RISC.

CISC x RISC

- Não necessariamente a arquitetura RISC é sempre a melhor escolha!
- Comparação de desempenho: cálculo de *speedup*
 - $S = 100 * (T_C - T_R) / T_R$, onde
 - S — *Speedup*;
 - T_C — Tempo de execução com CISC;
 - T_R — Tempo de execução com RISC.
 - $T = NI * CPI * P$, onde
 - T — Tempo de execução/processamento;
 - NI — Número de Instruções;
 - CPI — quantidade de Ciclos de clock Por Instrução;
 - P — Período, ou o tempo de cada ciclo.

CISC x RISC

■ Supondo:

- Um processador de 4MHz, ou seja, 4 milhões de ciclos por segundo, logo $P = 250\text{ns}$;
- *CPI* do CISC como 10 ciclos por instrução;
- *CPI* do RISC como 1 ciclo por instrução;
- Código CISC (*NI*) com 1283 instruções e seu equivalente em RISC (*NI*) com 10243 instruções.

CISC x RISC

■ Supondo:

- Um processador de 4MHz, ou seja, 4 milhões de ciclos por segundo, logo $P = 250ns$;
- CPI do CISC como 10 ciclos por instrução;
- CPI do RISC como 1 ciclo por instrução;
- Código CISC (NI) com 1283 instruções e seu equivalente em RISC (NI) com 10243 instruções.

■ $T_C = NI * CPI * P \quad \therefore \quad T_C = 1283 * 10 * 250 = 3.207.500ns$

CISC x RISC

■ Supondo:

- Um processador de 4MHz, ou seja, 4 milhões de ciclos por segundo, logo $P = 250ns$;
- CPI do CISC como 10 ciclos por instrução;
- CPI do RISC como 1 ciclo por instrução;
- Código CISC (NI) com 1283 instruções e seu equivalente em RISC (NI) com 10243 instruções.

$$■ T_C = NI * CPI * P \quad \therefore \quad T_C = 1283 * 10 * 250 = 3.207.500ns$$

$$■ T_R = NI * CPI * P \quad \therefore \quad T_R = 10243 * 1 * 250 = 2.560.750ns$$

CISC x RISC

■ Supondo:

- Um processador de 4MHz, ou seja, 4 milhões de ciclos por segundo, logo $P = 250ns$;
- CPI do CISC como 10 ciclos por instrução;
- CPI do RISC como 1 ciclo por instrução;
- Código CISC (NI) com 1283 instruções e seu equivalente em RISC (NI) com 10243 instruções.

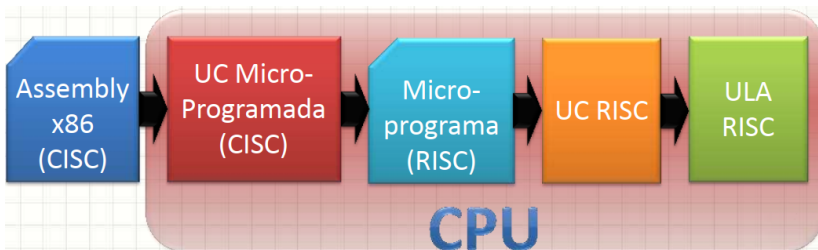
$$■ T_C = NI * CPI * P \quad \therefore \quad T_C = 1283 * 10 * 250 = 3.207.500ns$$

$$■ T_R = NI * CPI * P \quad \therefore \quad T_R = 10243 * 1 * 250 = 2.560.750ns$$

$$■ S = 100 * (T_C - T_R) / T_R \quad \therefore \quad S \approx 25,3\%$$

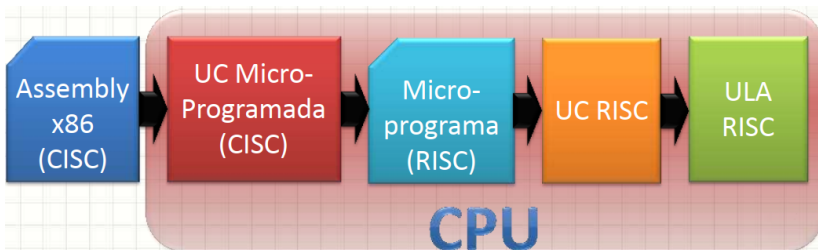
CISC x RISC

- Não é um ganho significativo.



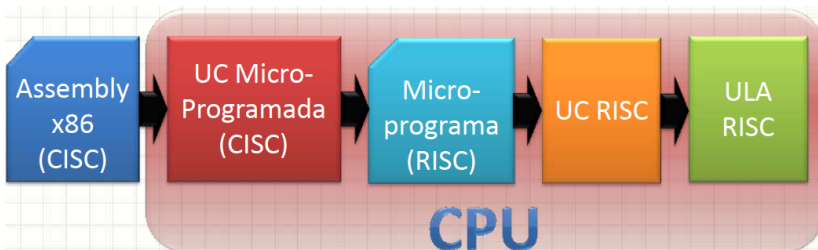
CISC x RISC

- Não é um ganho significativo.
- Porém uma CPU RISC é **menor**, **gasta menos**, é **mais barata**, e pode ser acelerada mais facilmente.



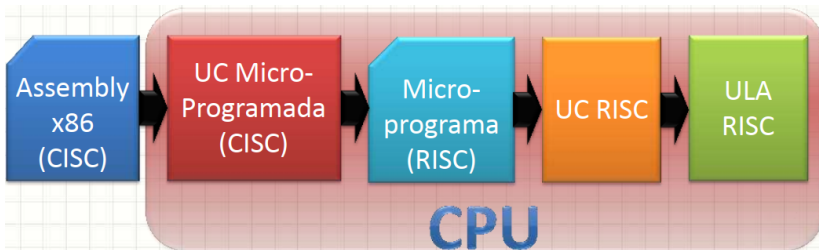
CISC x RISC

- Não é um ganho significativo.
- Porém uma CPU RISC é **menor**, **gasta menos**, é **mais barata**, e pode ser acelerada mais facilmente.
- E quando houver um uma instrução complexa, porém relativamente bastante utilizada?



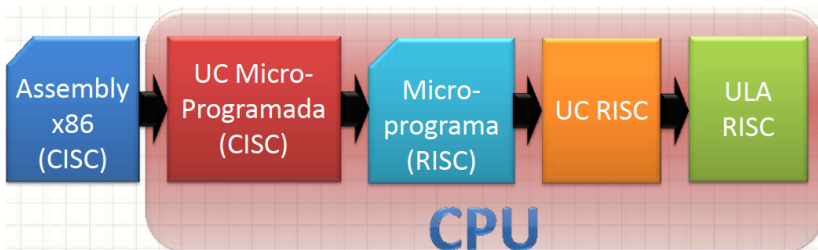
CISC x RISC

- Não é um ganho significativo.
- Porém uma CPU RISC é **menor**, **gasta menos**, é **mais barata**, e pode ser acelerada mais facilmente.
- E quando houver um uma instrução complexa, porém relativamente bastante utilizada?
- Como manter a compatibilidade com o hardware já existente?



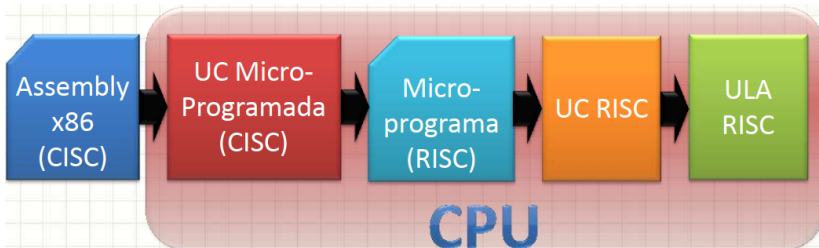
CISC x RISC

- Não é um ganho significativo.
- Porém uma CPU RISC é **menor**, **gasta menos**, é **mais barata**, e pode ser acelerada mais facilmente.
- E quando houver um uma instrução complexa, porém relativamente bastante utilizada?
- Como manter a compatibilidade com o hardware já existente?
- Um projeto CISC pode se beneficiar da inclusão de alguns recursos RISC, e **vice-versa**.



CISC x RISC

- Não é um ganho significativo.
- Porém uma CPU RISC é **menor**, **gasta menos**, é **mais barata**, e pode ser acelerada mais facilmente.
- E quando houver um uma instrução complexa, porém relativamente bastante utilizada?
- Como manter a compatibilidade com o hardware já existente?
- Um projeto CISC pode se beneficiar da inclusão de alguns recursos RISC, e **vice-versa**.
- Exemplo:



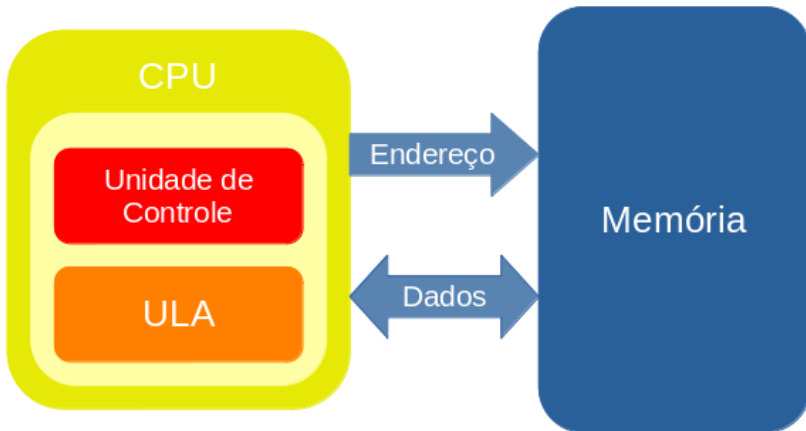
Arquitetura de Memória

Arquitetura de Memória

Arquitetura de Von Neumann

Arquitetura de Von Neumann

- Tanto os dados como as instruções de um programa são armazenados na mesma memória, e são transportados pelo mesmo barramento.
- Gargalo de memória: só é possível transmitir um de cada vez, o dado ou a instrução.

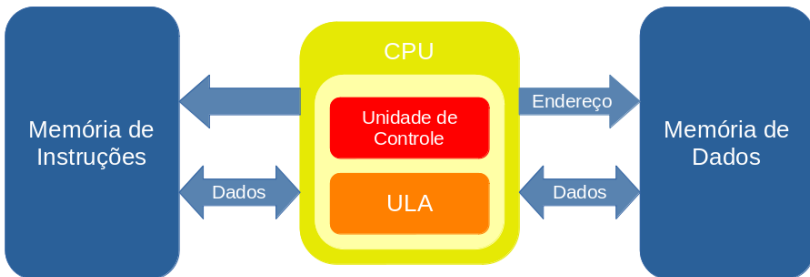


Arquitetura de Memória

Arquitetura de Harvard

Arquitetura de Harvard

- As instruções são armazenadas em uma memória separada dos dados.
- Instruções e dados podem ser transportados ao mesmo tempo, pois utilizam diferentes barramentos.



- Sua aplicação é comum em DSPs (*Digital Signal Processor*), microprocessadores especializados em processamento digital de sinal (digitalização de sinais analógicos).
- São utilizados no processamento de áudio/vídeo, Ressonância Magnética, *Smart Watches*, e até em assistentes domésticos como a Alexa.

Arquitetura de Harvard

- Essa ideia/princípio foi adaptada na Arquitetura de Von Neumann, nos computadores mais modernos.
- As Caches de nível 1 ainda podem ser divididas em duas: cache de instruções e cache de dados.

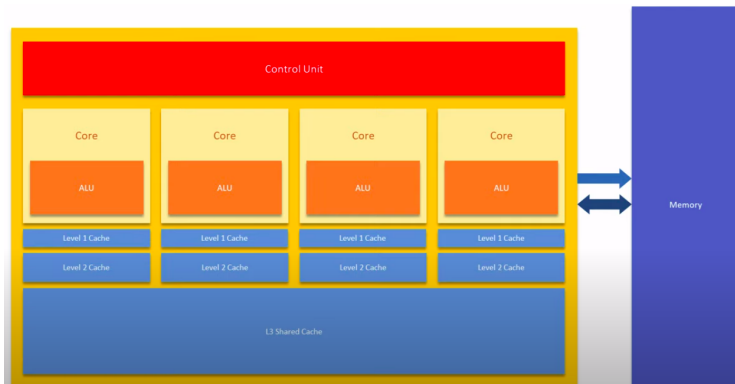


Figura: Fonte: <https://www.youtube.com/watch?v=4nY7mNHLrLk>

Arquitetura dos Microcontroladores

Arquitetura dos Microcontroladores

- Lista com alguns microcontroladores comuns por marca (lista completa [aqui](#)).
 - ARM
 - Famílias de microprocessadores RISC licenciados para uso em microcontroladores.
 - Atmel
 - Família AVR — microcontroladores RISC de 8 bits e Arquitetura de Harvard modificada.
 - Famílias básicas: ATmega e ATtiny.
 - Comumente utilizado nas placas Arduino.
 - Epson
 - Família S1C6x — CISC e Arquitetura de Harvard.
 - Microchip Technology
 - PIC (*Peripheral Interface Controller, Controlador de Interface Periférica*)
 - Conjunto de famílias com Arquitetura de Harvard e suporte para RISC.
 - Raspberry Pi Foundation
 - RP2040 — 32-bit dual ARM Cortex-M0+.

FIM