



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Sistemas Embarcados: (ELF74)

Prof: DaLuz



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Sistemas Embarcados:

Restrições Temporais

Sistema de tempo Real

Sensores

Atuadores

Microntroladores

Custo envolvido

Eficiência Energética

Não Customizável

Interconectividade

... * ...



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Segmentos:

Equip. Consumidor

Sis. Telecomunicações

Automação Industrial

Transporte

Equi. Médicos

Entretenimento



Ref. *



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Segmentos:

Aviônico – A350



Ref. *



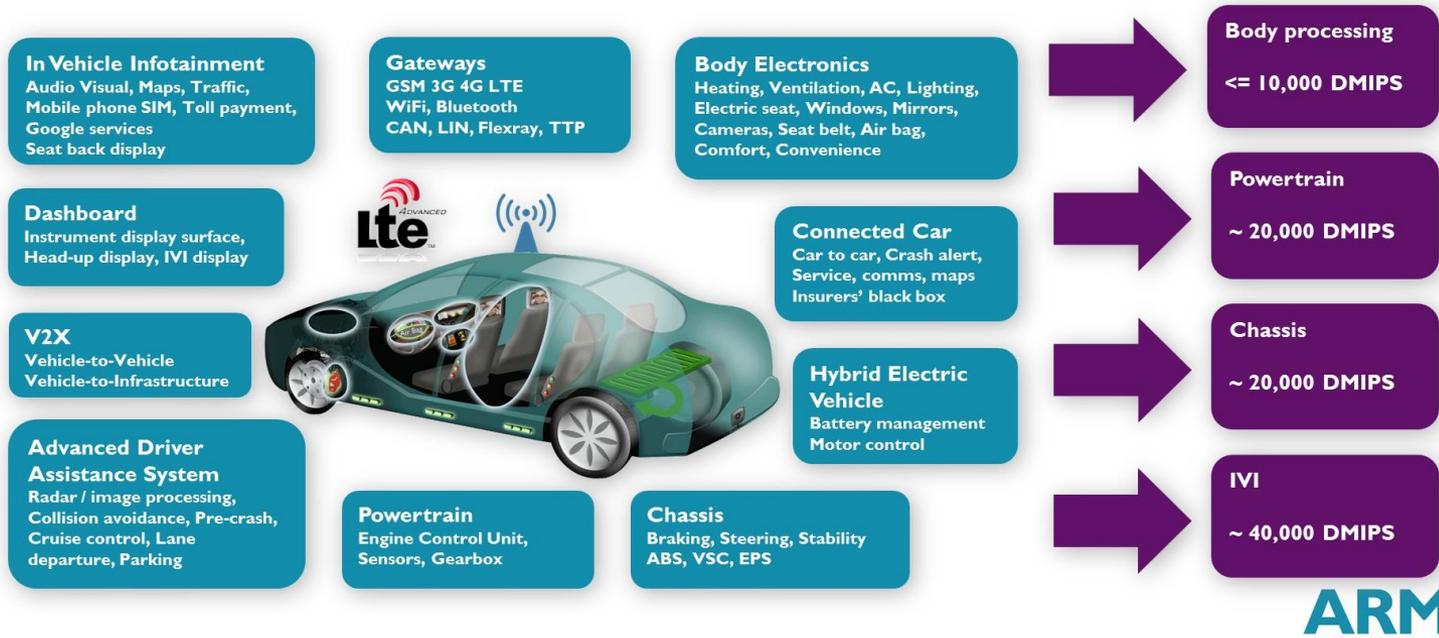
- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Segmentos:

Automotivo

Automotive ECUs Controllers by 2020

- Between 25 and 100 individual ECUs
- With distributed sensors and motor controllers.



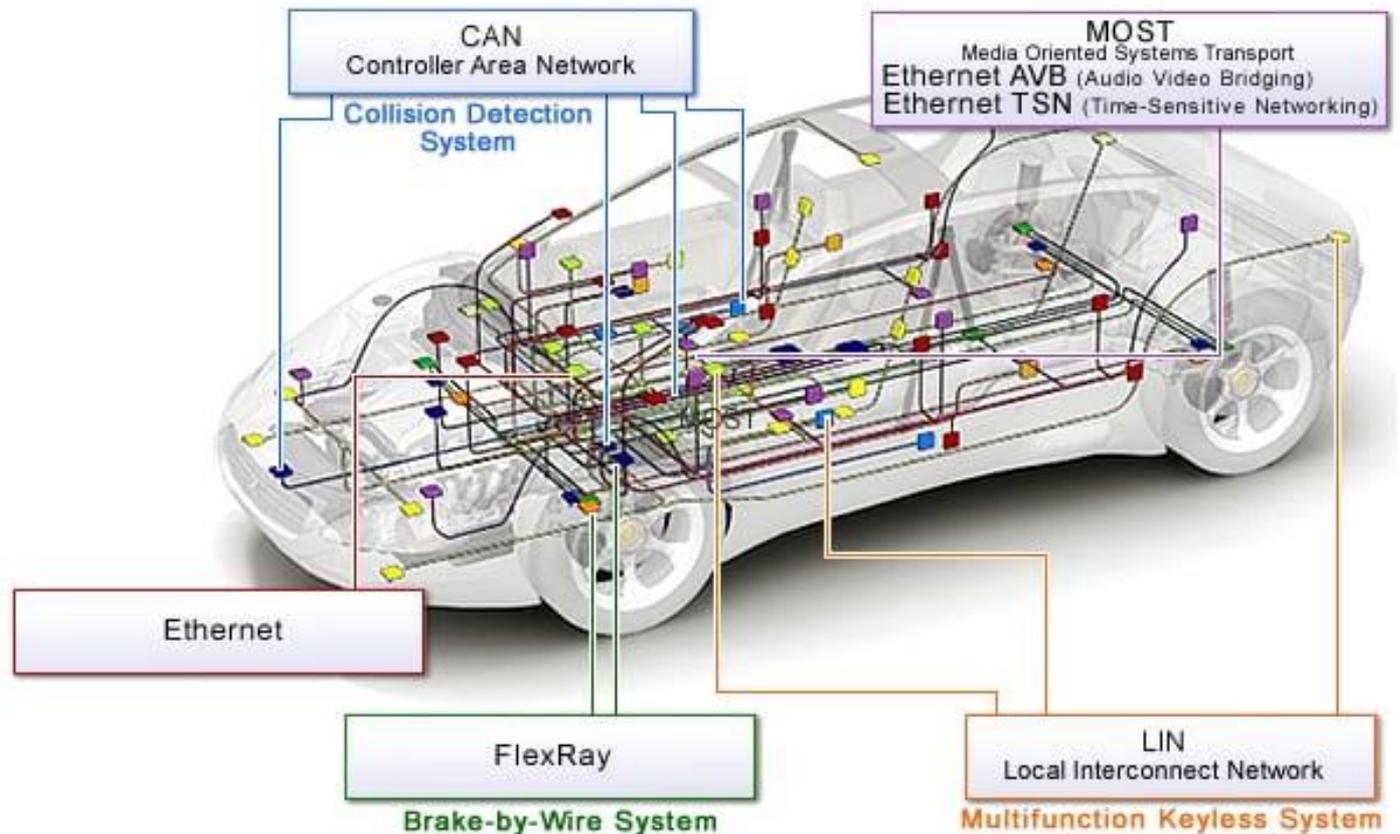
Ref. *



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Segmentos:

Automotivo



Ref. *



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Segmentos:

Equip. Médico-Hospitalares



*Ref. **

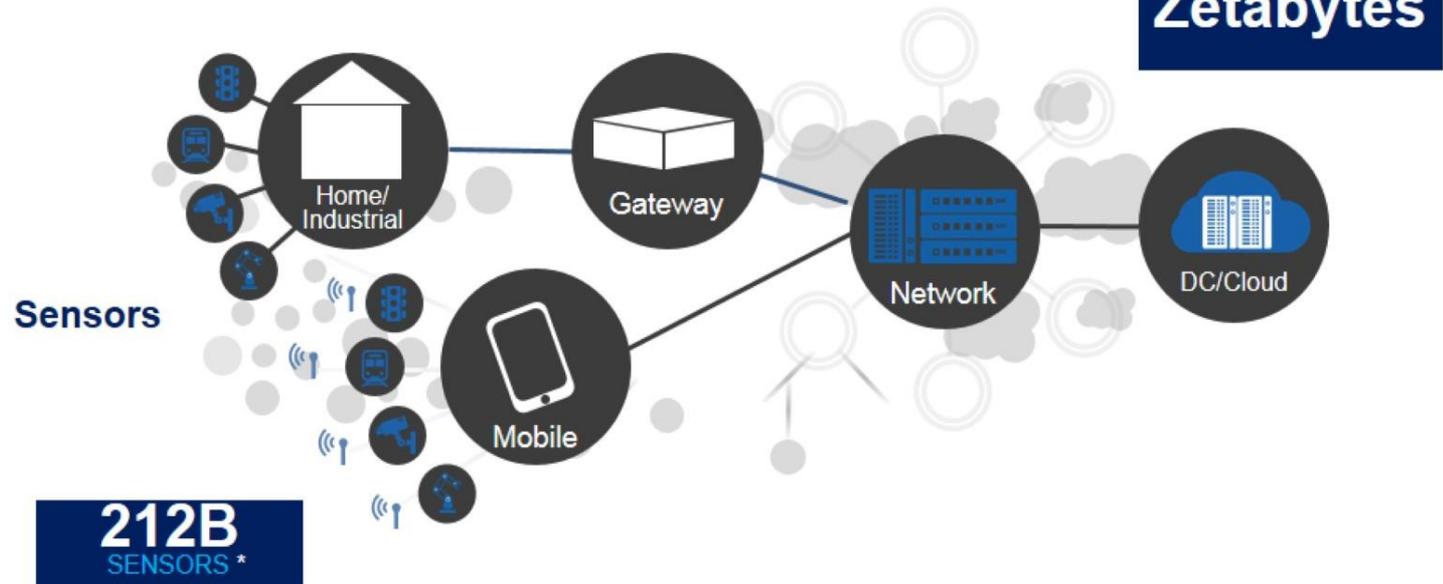


- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Segmentos:

IoT – *Internet of Things*

The Internet of Things is Everywhere



Ref. *

Segmentos:

Ex: eVTOL – Taxis Aéreos



www.airway.com.br/embraerx-revela-novo-conceito-de-taxi-aereo-urbano/

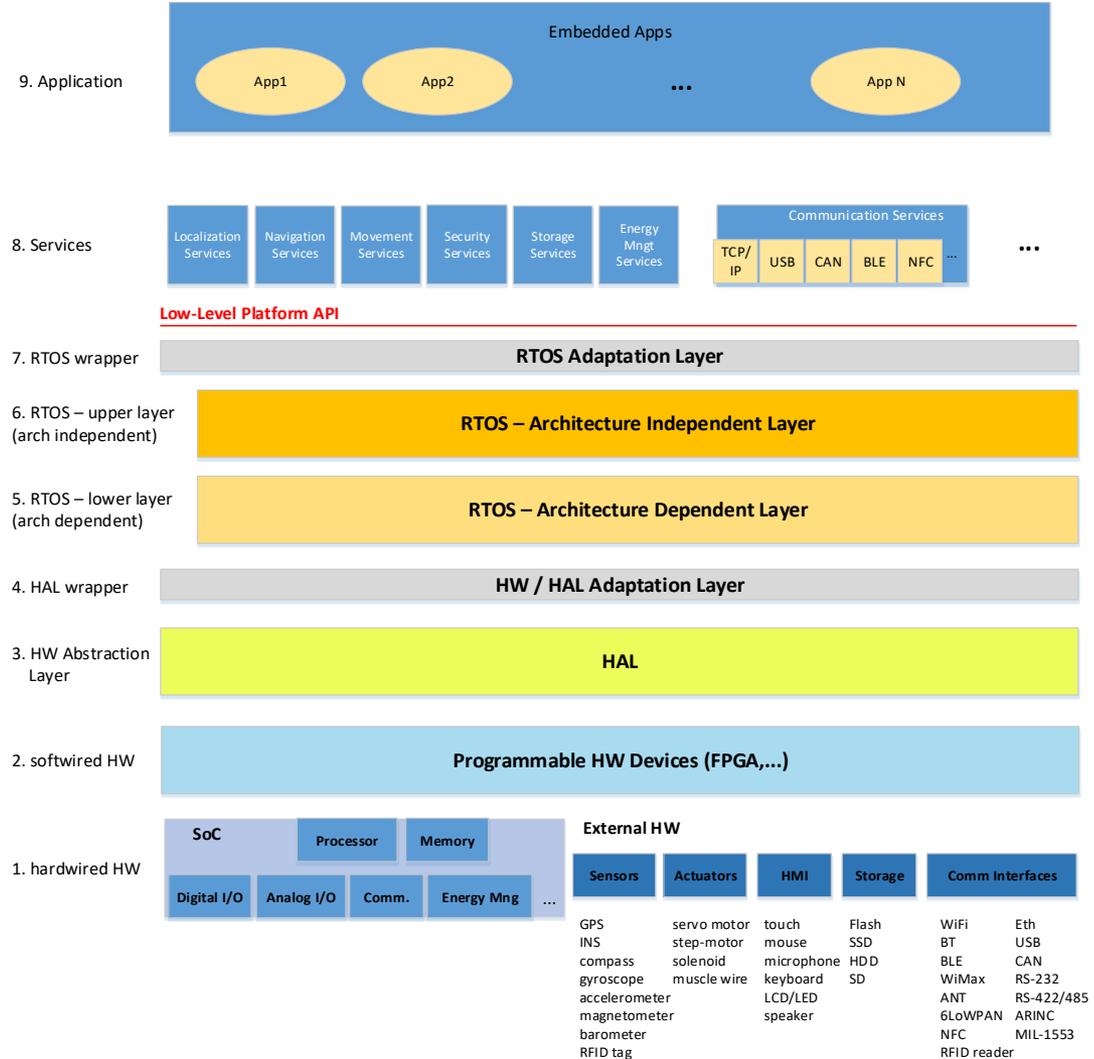
UTFPR

- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:



Ref. *



- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura:

1-Hardwired HW

Esta camada é composta por dispositivos:

- 📖 **MCU**
- 📖 **Memória**
- 📖 **I/O**
- 📖 **Conectores**
- 📖 **PCB**
- 📖 **Sensores / Atuadores**
- 📖 **Interface Homem - Máquina**



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

2-Softwired HW

Esta camada é composta por dispositivos:

- 📖 **FPGA**
- 📖 **CPLD**
- 📖 **GAL**
- 📖 **PAL**
- 📖 **PLA**
- 📖 **ROM extra**



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

3-Hardware Abstraction Layer

Esta camada é composta por *software*:

- 📖 **Device Drivers**
- 📖 **SPI**
- 📖 **I2C**
- 📖 **API**
- 📖 **Bibliotecas**
- 📖 **Escritas em C / Assembly**



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

4-Adaption Layer / Wrapper

Esta camada é composta por *software*:

- ☐ **Interfaceamento comum ≠ fabricantes**
- ☐ **Converte um padrão comum ↔ fabricantes**
- ☐ **Utiliza geralmente Funções e Expr. regulares**
- ☐ **Camada de Padronização**
- ☐ **Camada de Convesão: Genérica ↔ Específica**



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

5 e 6-RTOS

Esta camada é composta por *software*:

- ☐ **Real time operating System**
- ☐ **Pouca memória utilizada**
- ☐ **Funções de suporte reduzida ...**
- ☐ **Dividido: Parte dependente de arquitetura**
- ☐ **Parte independente de arquitetura**



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

7-RTOS *Adaption Layer*

Esta camada é composta por *software*:

- ☐ **Tradução da API do RTOS por uma padrão:**
Ex: CMSIS-RTOS
- ☐ **Provê estrutura de nível superior**
- ☐ **Reduz a adaptação de código em migrações**
- ☐ **Dividido: Parte dependente de arquitetura**
- ☐ **Parte independente de arquitetura**



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

8-Serviços

Esta camada é composta por *software*:

- ☐ **Reutilização de código, bibliotecas**
- ☐ **Stacks: Tcp/IP, Usb, Gps, Segurança, etc**
- ☐ **São de fácil portabilidade**
- ☐ **Compiladas para diferentes plataformas**



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

9-Aplicação

Esta camada é composta por *software*:

- ☐ **Camada de mais alto nível**
- ☐ **Programas rodam aqui**
- ☐ **Podemos ter diversos programas rodando em cooperação. Programação concorrente**
- ☐ **Concorrência é gerenciada pelo RTOS**



- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura:

CISC

Complex Instruction Set Computer



- Ênfase do Hardware
- Maior qt. de instruções
- Operações com memória nas intruções.
- Alta taxa de ciclos / s, para pouco código
- Maior qt. de transistores para armazenar intruções complexas

x86

RISC

Reduced Instruction Set Computer



- Ênfase do *Software*
- Menor qt. de instruções que CISC
- Execução otimizada de *calls func.*
- Menor qt. modos de endereçamento
- Utilização: larga escala de *pipelining*
- Maior qt. transistores para Regist.

ARM

MIPS

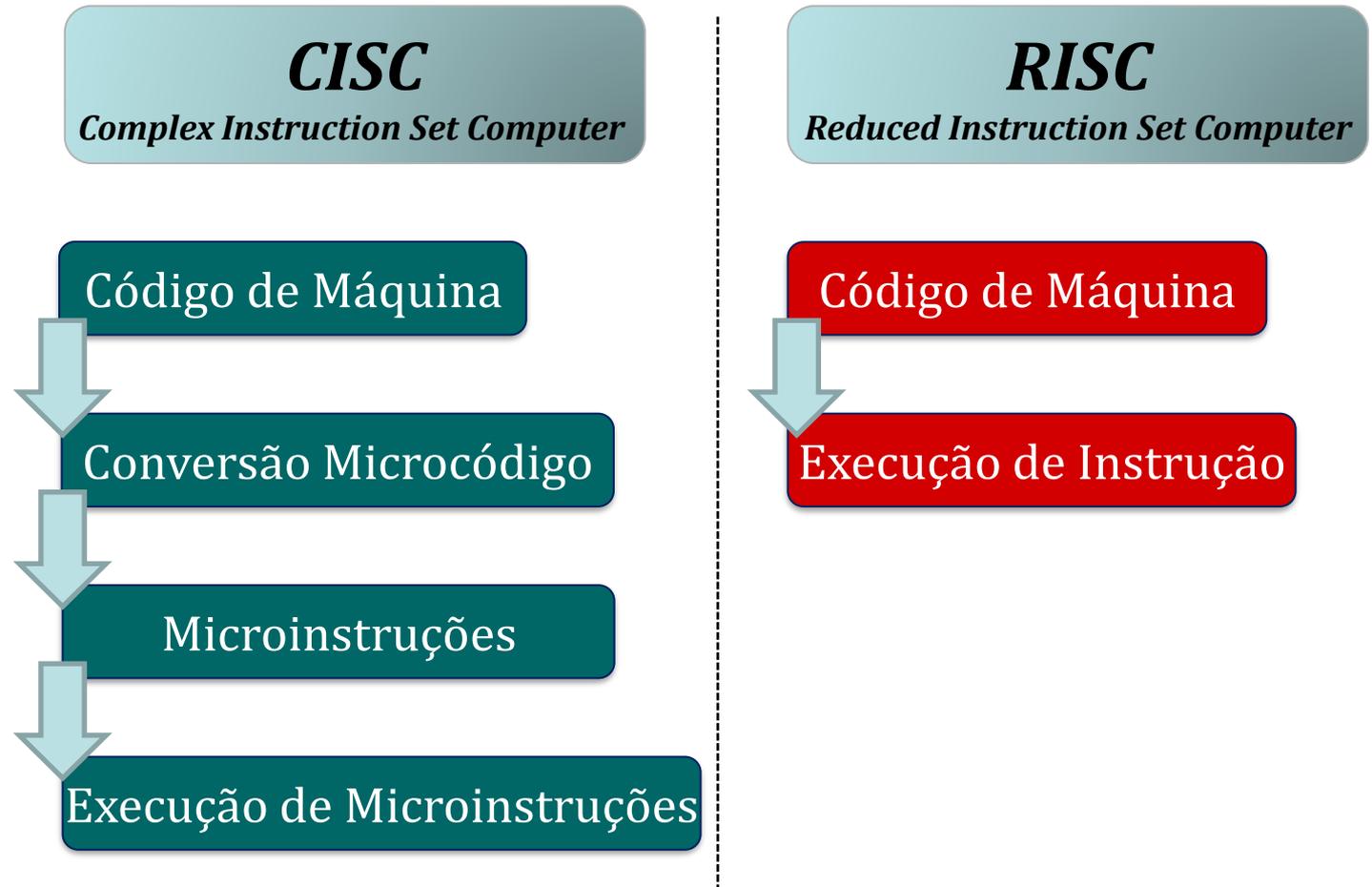
RISC-V

<https://pt.wikipedia.org/wiki/RISC>



- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura:





- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

As arquiteturas puras são raras atualmente:

- ☐ HCS08 da Freescale (**CISC**)
- ☐ PIC da Microchip (**RISC**)

x86 é **CISC**, mas suas instruções são decodificadas em microinstruções **RISC**

Principais Processadores atualmente RISC:

ARM

MIPS

RISC-V

PowerPC

SPARC

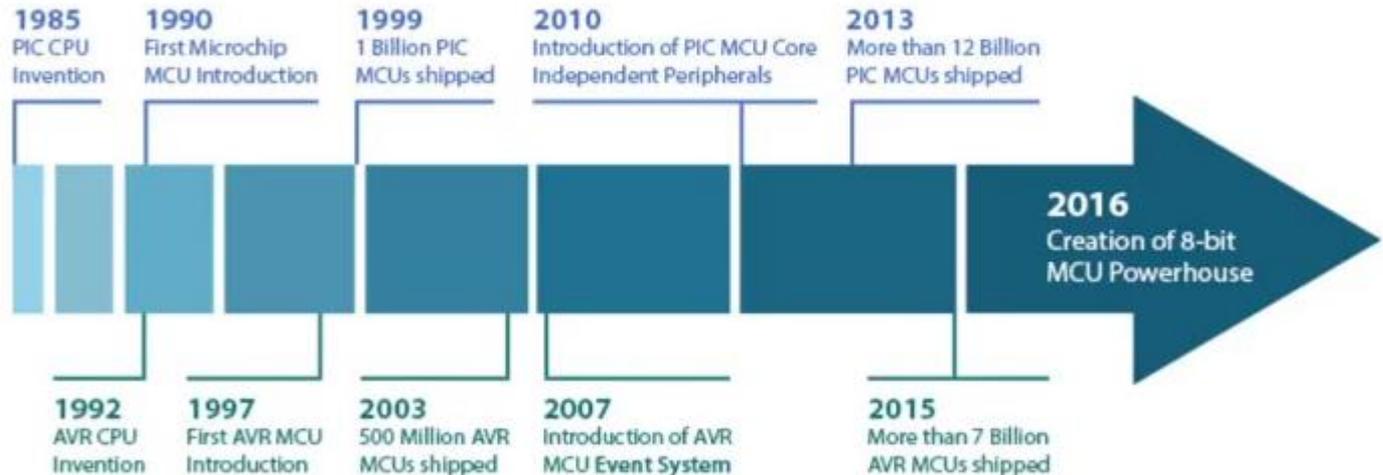
Etc.



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

PIC® MCU



AVR® MCU

Ref. *



- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura:



*available for STM32MP151C and STM32MP151F only

Ref. *



- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura:

Exemplo: Processador ColdFire:

- 📖 Baseado no Processador 68K(CISC) – Atari, Apple Lisa, Macintosh, etc.
- 📖 *Variable-length reduce instruction set (VL-RISC)*
- 📖 Operações baseadas em memória (CISC)



Ref. *



- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura:

*“ColdFire occupies the “**sweet spot**” **between** complex instruction set computing (**CISC**) and conventional fixed-length reduced instruction set computing (**RISC**), striking an optimum balance between code density and transistor count. By using Motorola’s unique **Variable-Length (VL) RISC** technology, ColdFire combines the simplicity and optimal performance of a 32-bit RISC architecture with a memory-saving variable-length instruction set.”*



- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura:

Premissas da Abordagem RISC

Reduced Instruction Set Computer

- ❏ Maior desempenho pode ser obtido se cada instrução ocupar exatamente uma palavra de memória.
- ❏ Esta restrição reduz a complexidade e o número de instruções que podem ser incluídas na CPU.



- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura:

Premissas da Abordagem RISC

Reduced Instruction Set Computer

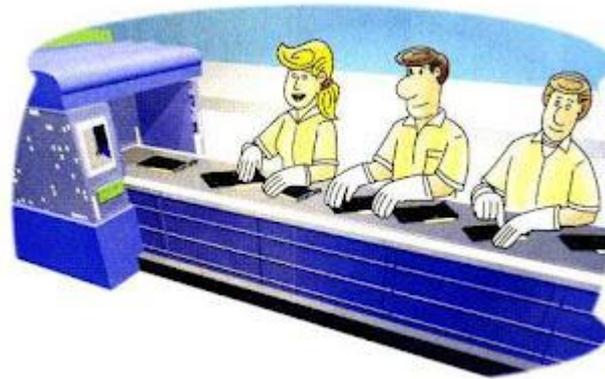
- ▣ Todos os operandos de instruções aritméticas e lógicas devem estar nos registradores do processador.
- ▣ Propício para implementar unidades de processamento nas quais várias operações necessárias para processar uma sequência de instruções são executadas em ***pipeline***.



- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura: Pipeline

- ▣ Conceito de **“linha de produção”**.
- ▣ A organização em fila de processos eleva a **“produtividade”**.



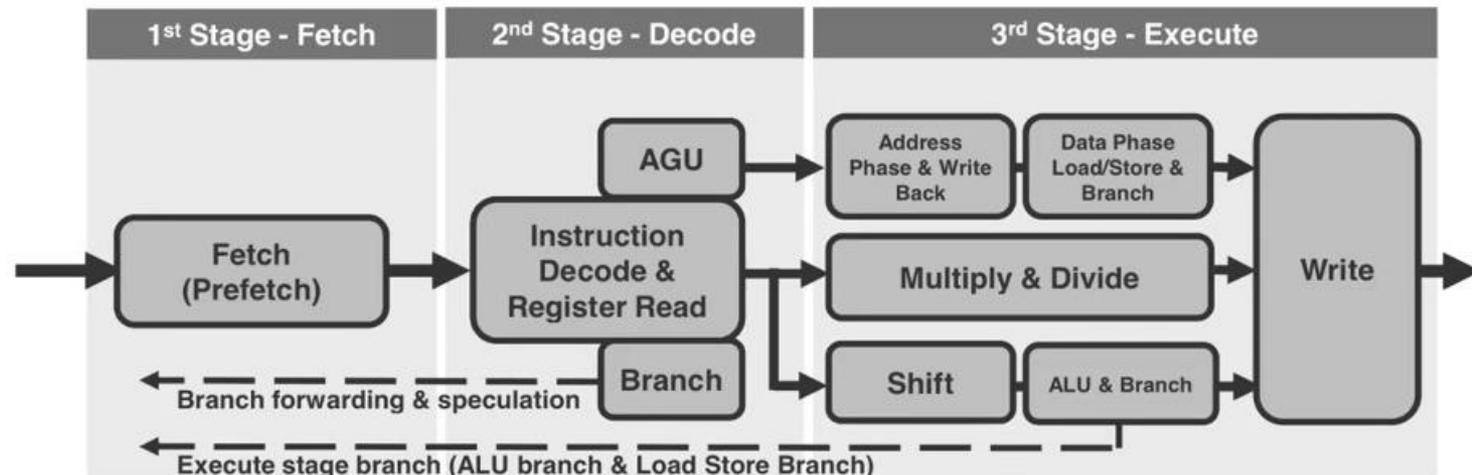
*Ref. **



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura: Pipeline

- ☐ Cortex-M4 possui 3 estágios de *pipeline*: 1:Fetch, 2:Decode, 3:Execute.
- ☐ Cortex-M4 faz mais em cada estágio para aumentar a performance.



Ref. *

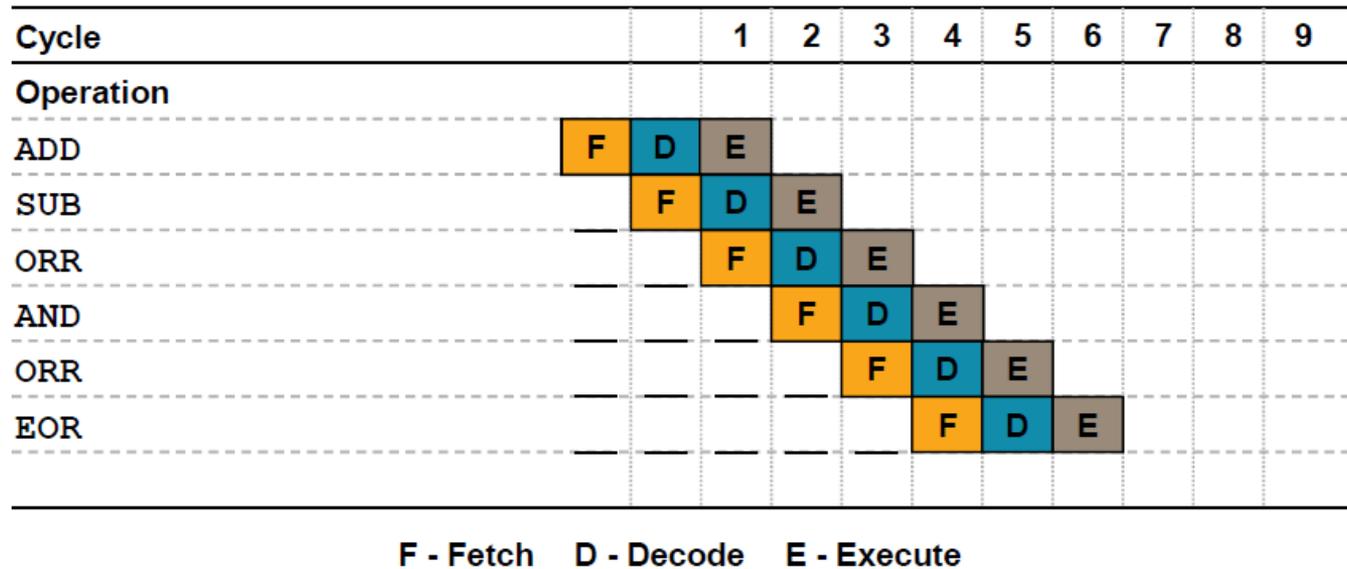


- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

Pipeline

- ☐ Cortex-M4 possui 3 estágios de *pipeline*: 1:Fetch, 2:Decode, 3:Execute.



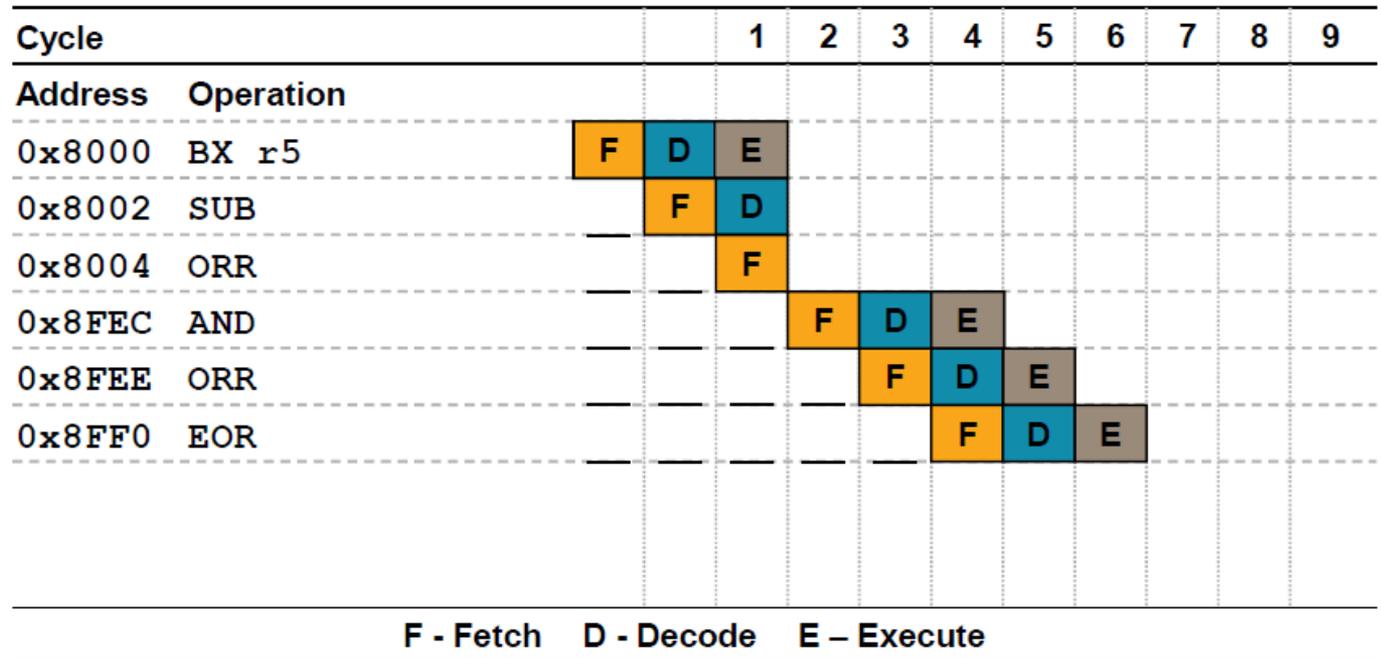
Ref. *



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura: Pipeline

☐ Cortex-M4 – flush após um salto



Ref. *



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

Pipeline

- ☐ Cortex-M4 – A arquitetura **Harvard** permite o acesso simultâneo entre as memórias de **código** e **dados**.

Cycle		1	2	3	4	5	6	7	8	9
Operation										
ADD		F	D	E						
SUB			F	D	E					
STR				F	D	Ea	Ed			
STR					F	D	Ea	Ed		
ORR					F	D	E			
EOR						F	D	E		

F - Fetch D - Decode E - Execute S - Stall
 Ea - Execute / STR address phase Ed - STR data phase

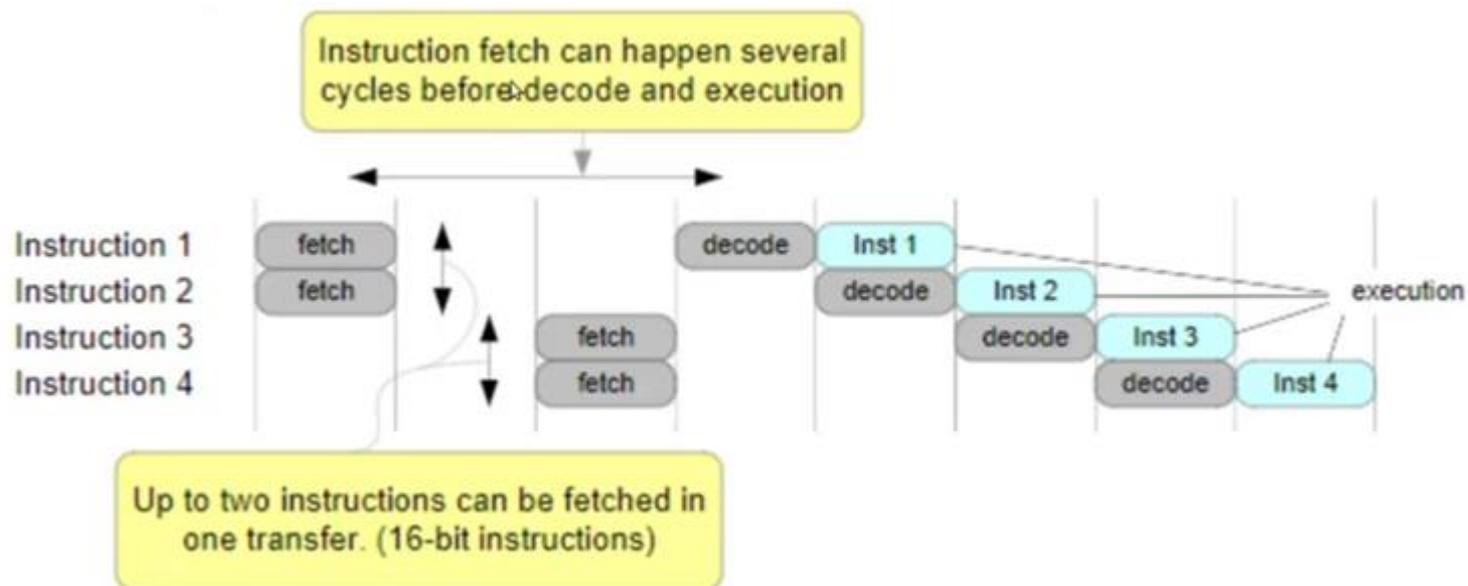
Ref. *



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura: Pipeline

- Estágios no Cortex-M3 e Cortex-M4.



Ref. *



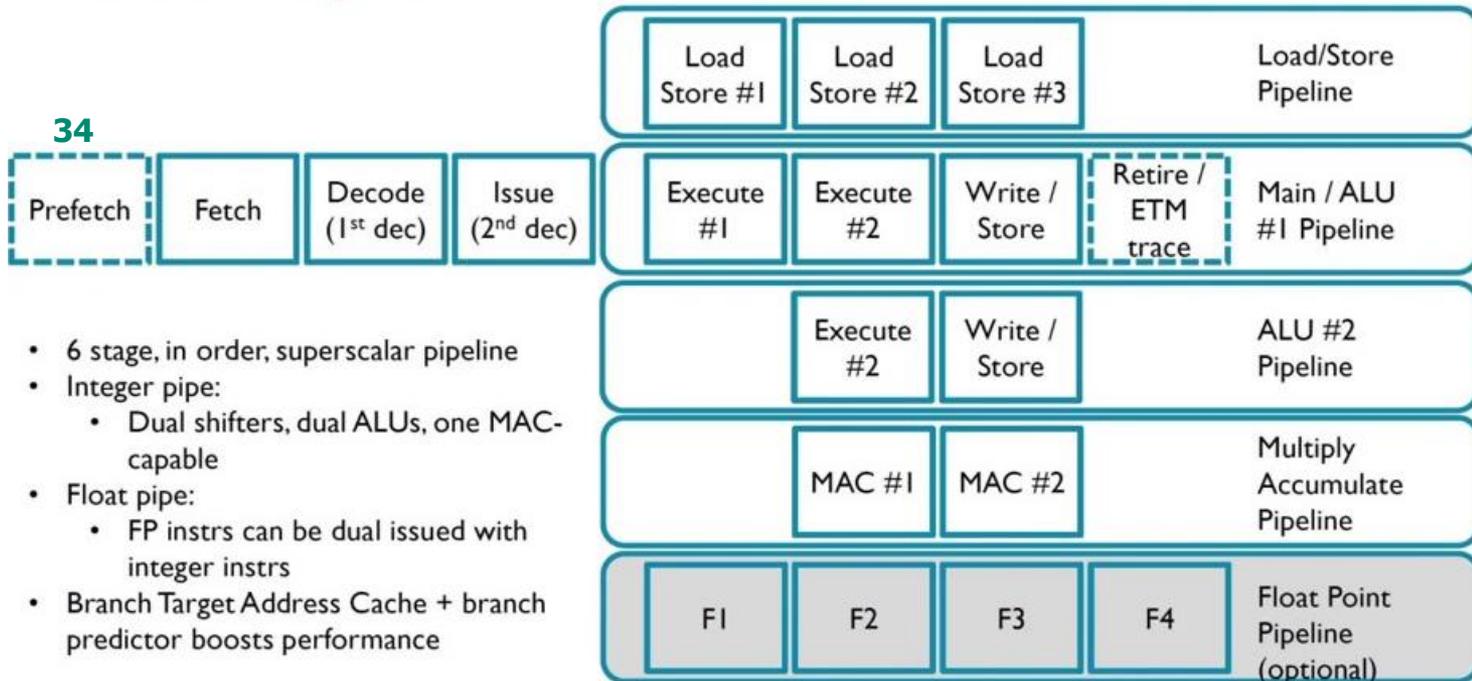
- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura:

Pipeline Superescalar

Estágios no Cortex-M7.

Cortex-M7 Pipeline



- 6 stage, in order, superscalar pipeline
- Integer pipe:
 - Dual shifters, dual ALUs, one MAC-capable
- Float pipe:
 - FP instrs can be dual issued with integer instrs
- Branch Target Address Cache + branch predictor boosts performance

Ref. *

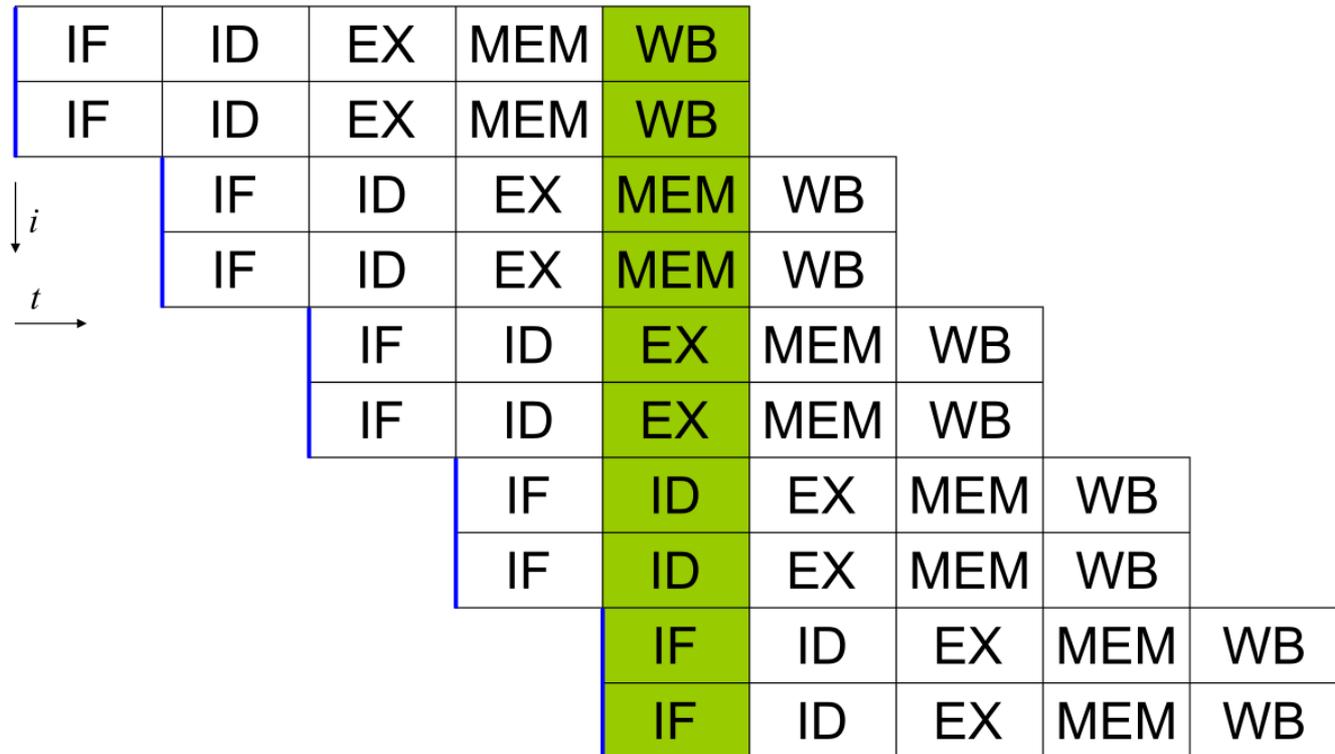


- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Arquitetura:

Pipeline Superescalar

Estágios no Cortex-M7.



Obs: *Dobro de instruções ↔ Uma Thread.*

Ref. *



- Sis. Embarcados
- Segmentos
- **Arquitetura**
- Referências

Arquitetura:

IDE: IAR - Instalação



http://www.elf74.daeln.com.br/Pdfs/IAR_Install.pdf



- Sis. Embarcados
- Segmentos
- Arquitetura
- Referências

Referências:

Extras:

Video:

2b economics.mkv

Aspectos econômicos na produção de microcontroladores

Video:

3a history.mkv

Histórico da ARM

☒ * Refs ↔ Renesas.com, Pixabay.com, wikimedia.org, flickr, community.arm.com, Undergraduated course Renesas (Prof. Douglas P. B. Renaux e Robson Linhares), ytchannel Gustavo W. Dernardin.