

# PSI3441 – Arquitetura de Sistemas Embarcados

---

## Aula 1 Introdução

---

Escola Politécnica da Universidade de São Paulo

Prof. Gustavo Rehder – [gprehder@usp.br](mailto:gprehder@usp.br)





# Objetivos da Aula

- Descrever as etapas do projeto
- Listar tipos de CPUs e diferenças
- Listar diferenças entre arquiteturas de uControladores
- Elencar critérios para escolha de hardware
- Definir requisitos e especificações
- Listar periféricos disponíveis

# o que é um sistema embarcado?



Start the presentation to see live content. Still no live content? Install the app or get help at [PollEv.com/app](https://PollEv.com/app)



# Definição

- Sistema embarcado: combinação de hardware e software projetados para uma função específica.

Exemplos:



Industrial Robots



GPS Receivers



Digital Cameras



DVD Players



Wireless Routers

## Embedded Systems



MP3 Players



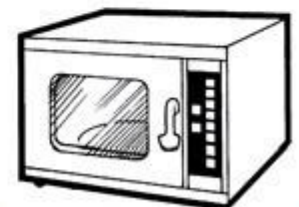
Set top Boxes



Gaming Consoles



Photocopiers

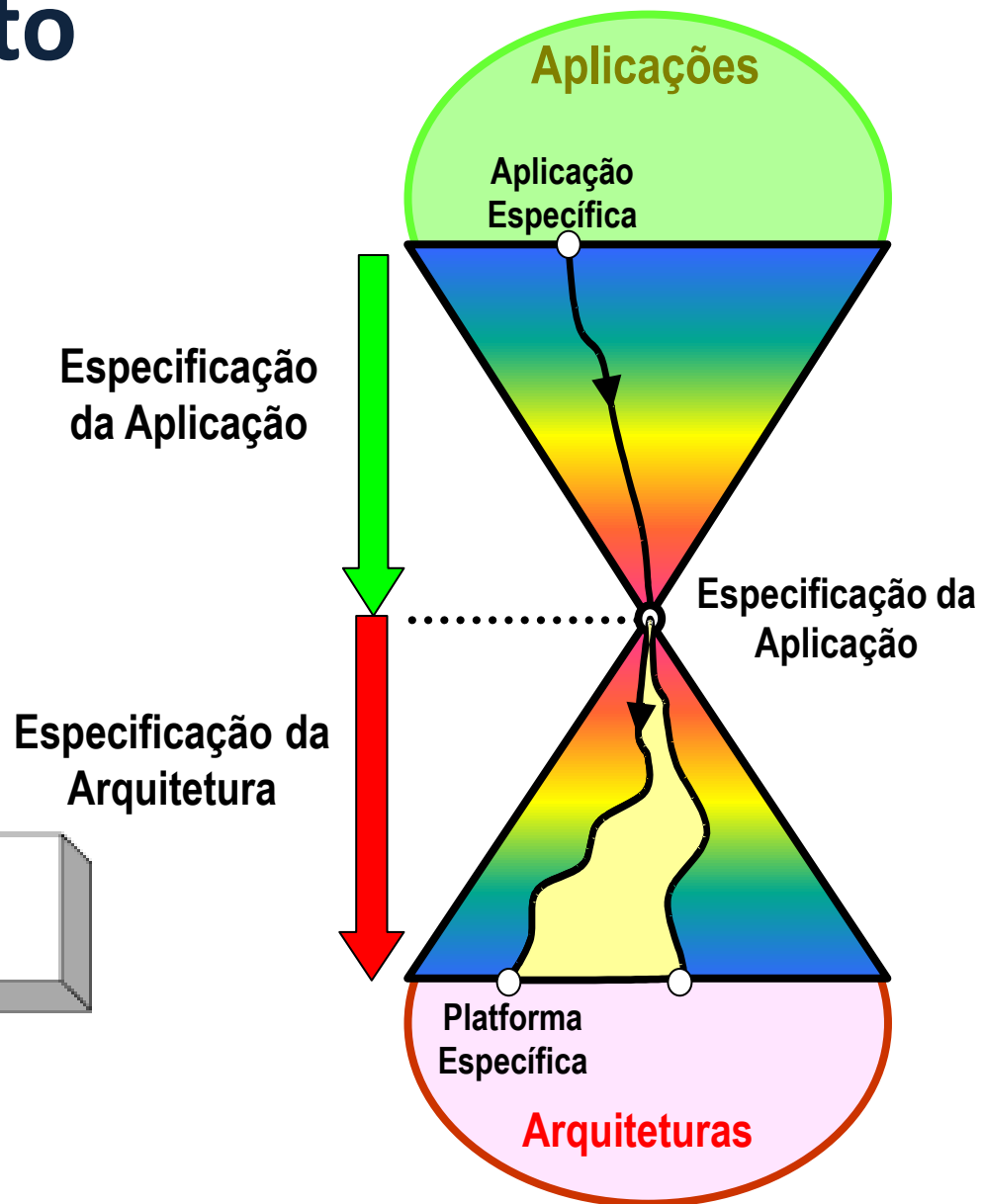
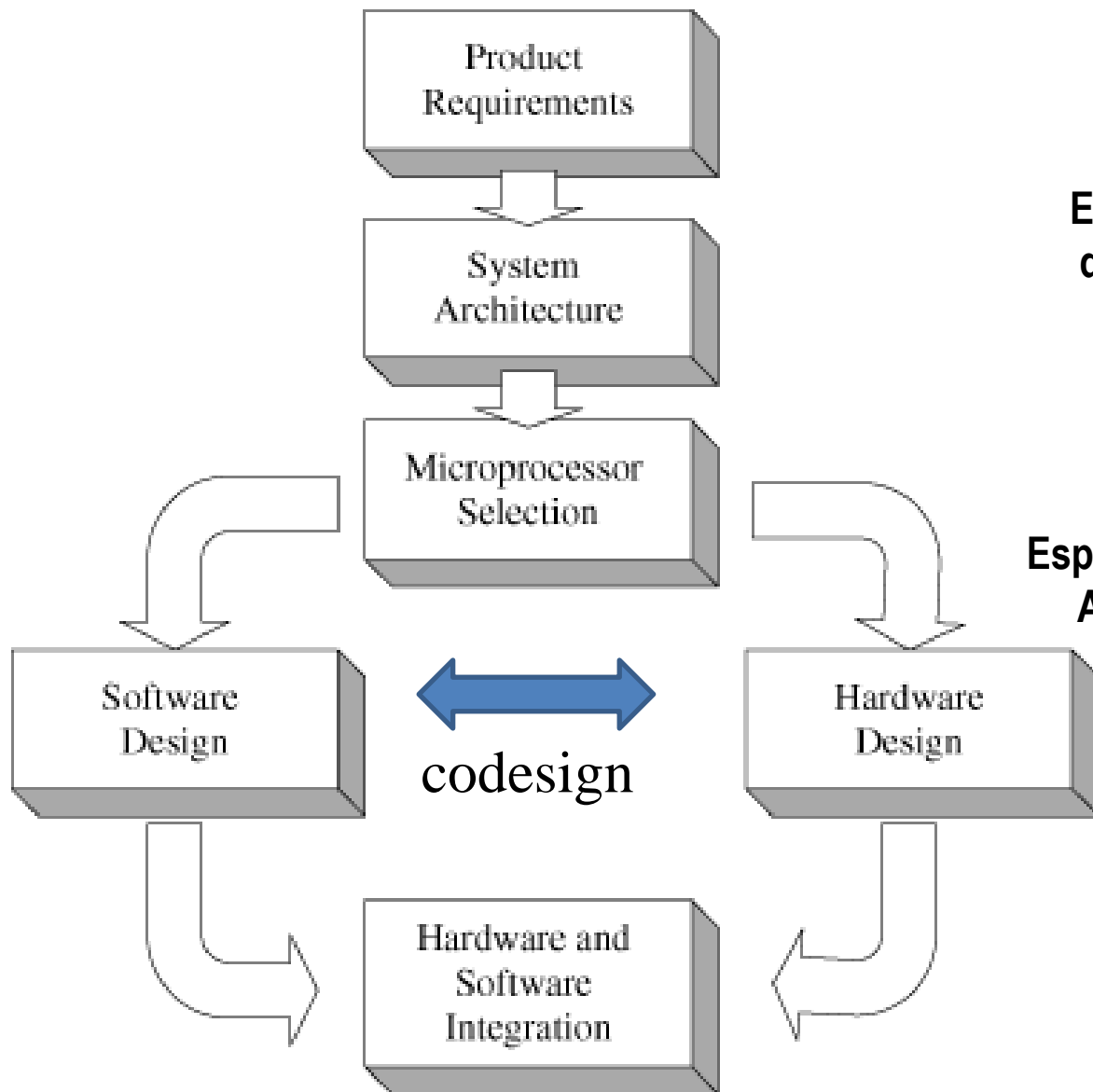


Microwave Ovens



# Projeto de Sistemas Embarcados

## Fluxo de desenvolvimento





# Requisitos – definidos pelo cliente

Definem os objetivos do projeto (What?)

- Funções que o Sistema precisa executar (functional)
  - Entradas
  - Saídas
  - Comunicação
- Velocidade (não funcional)
- Desempenho (não funcional)
- Custo (não funcional)
- Tamanho físico (não funcional)
- Consumo de potência (não funcional)



# Exemplo: Camera digital

- Funcional
  - Capturar Imagem com resolução para imprimir uma foto 8"x10"
  - Exibir Imagem no display
  - Zoom óptico de 4x
  - Transferência de imagem por USB
- Não Funcional
  - Custo (R\$ 500)
  - Peso (800g)
  - Potência consumida (suficiente para durar 1 dia tirando fotos)
  - Tamanho (cabe no bolso)





# Especificações – definidos pelo engenheiro

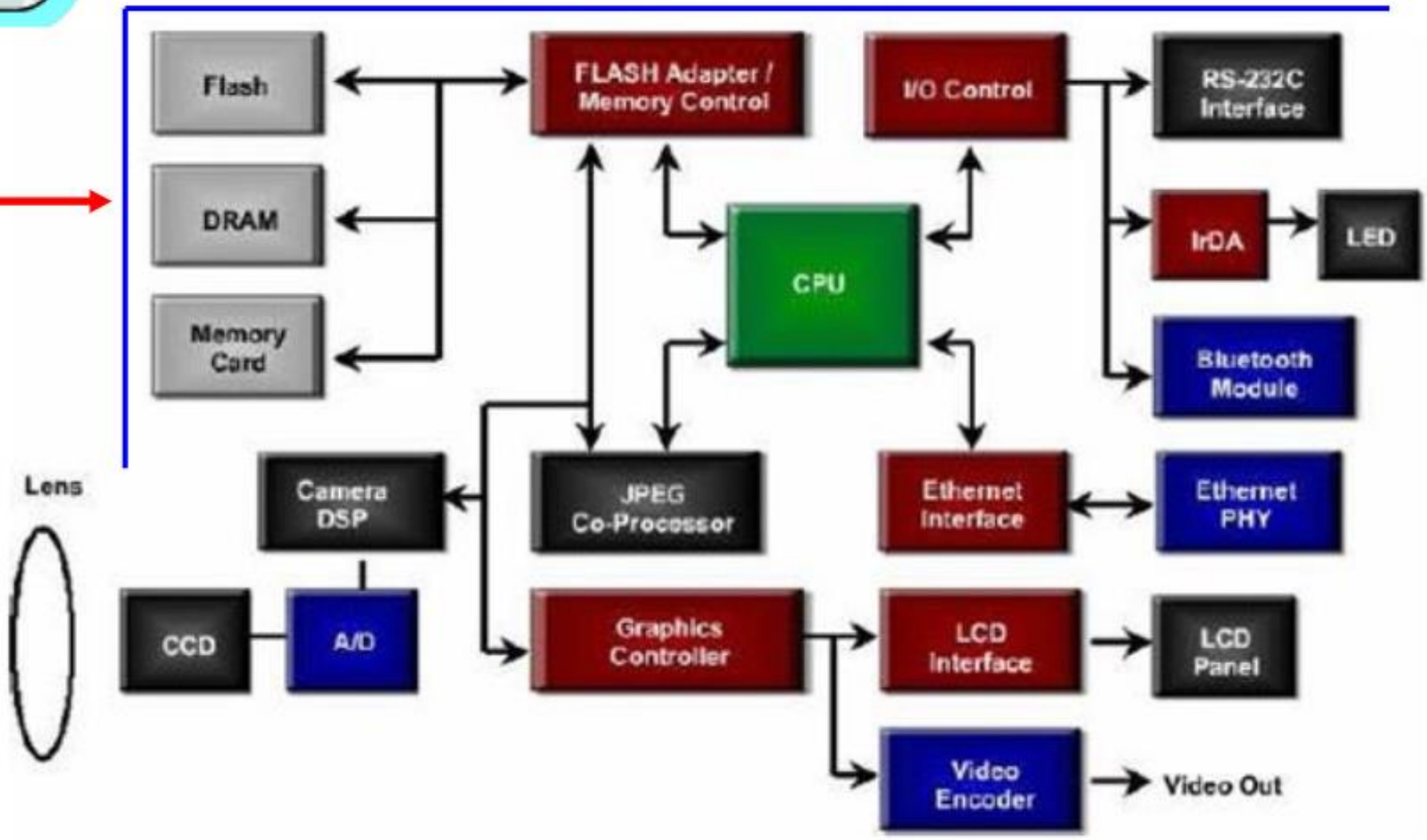
Definem como fazer o projeto para alcançar os objetivos (How?)

Exemplo: Câmera digital

- Sensor CCD de 5 M píxeis com ½ polegada
- Display de 4.3”, resolução 480 x 272
- Cartão de 2 Gb de memória
- Bateria de 2500 mAh



# Digital Camera Block Diagram





# Software

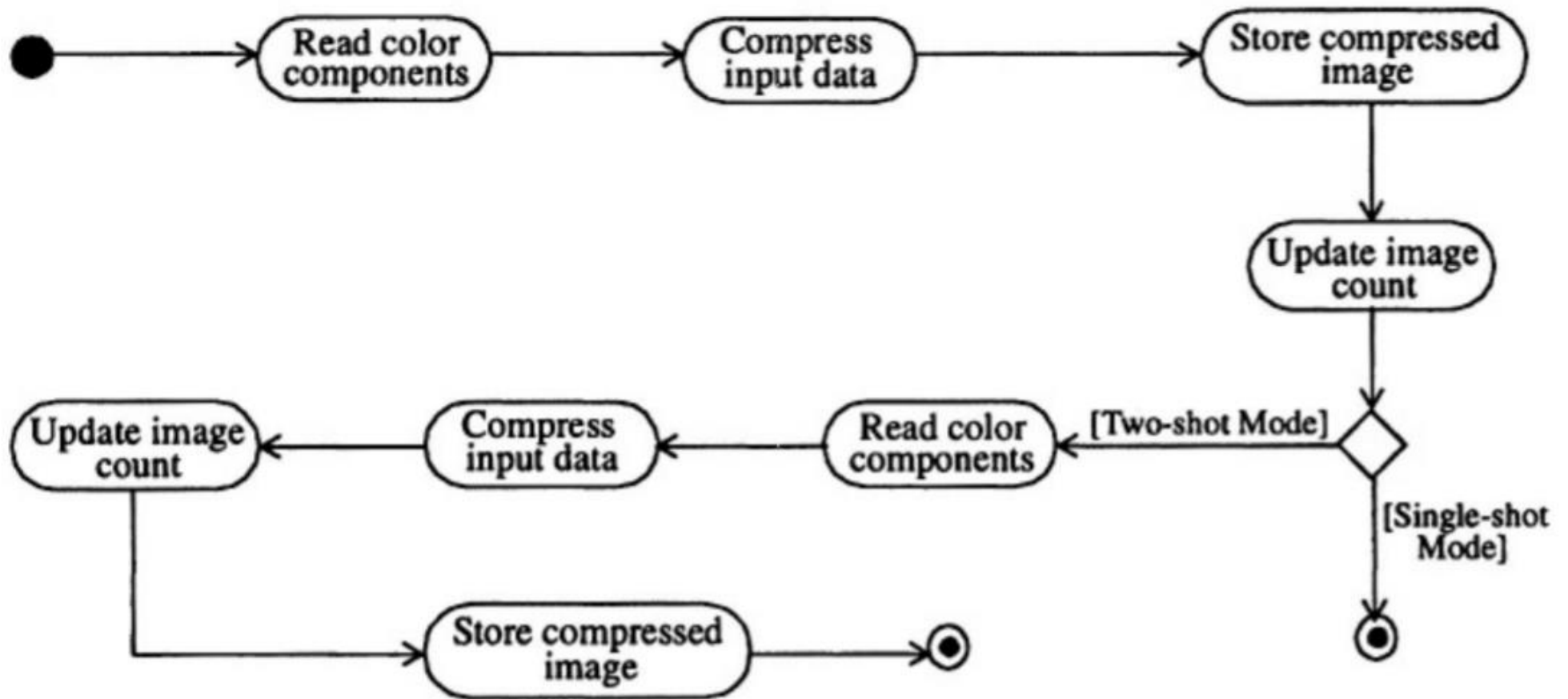


Figure 6-8. The “Take a picture” Activity diagram



# Integração e Teste





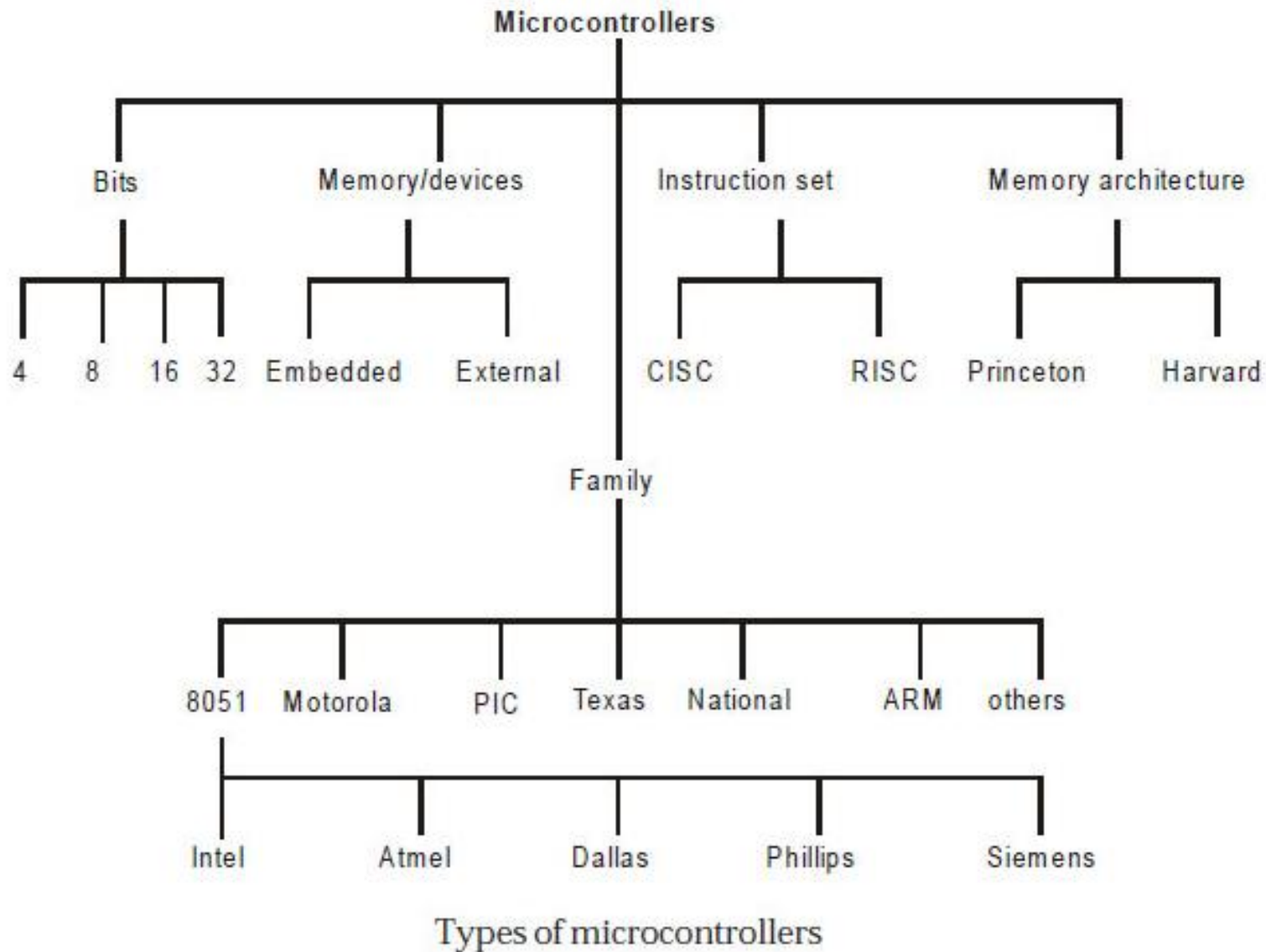
# Definição da CPU

- Processador, **Microcontrolador**, DSP, FPGA, ASIC?
- Arquitetura: Load/Store, Register Memory, Stack?
- Von Neumann ou Harvard?
- RISC ou CISC?
- 8 bits, 16 bits, 32 bits ou 64 bits?
- 64 Mb, 128 Mb, 256 Mb, ... ?
- Velocidade do Clock? 16 MHz, 48 MHz, ... ?
- Memória Externa?
- Periféricos? AD/AD, timers, comunicação serial, ... ?
- Consumo de potência?
- Preço?
- Cache? Pipeline? DMA?

**Necessidade de conhecer a  
arquitetura e funcionamento  
de CPUs**



# Classificação de Microcontroladores

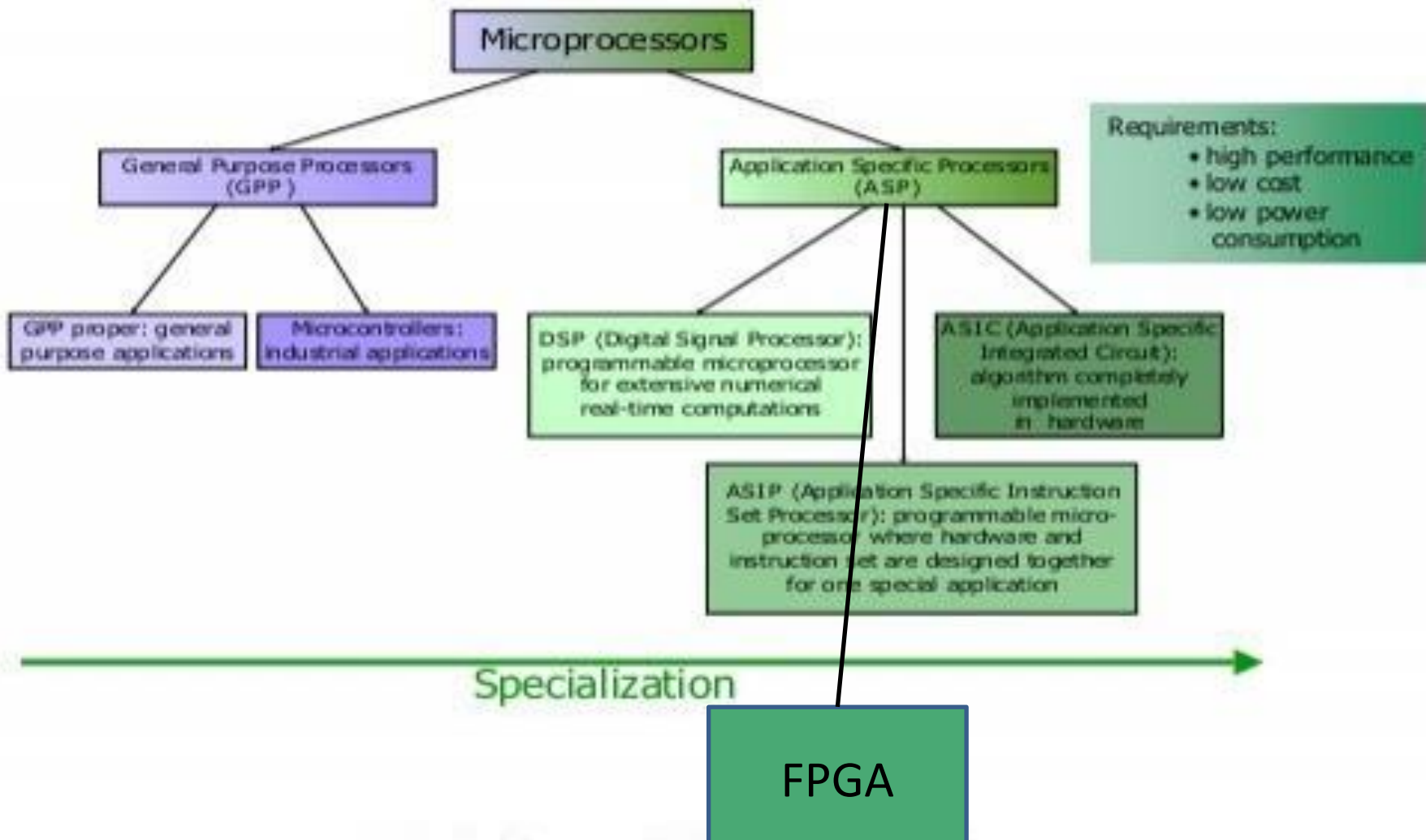


- <http://electro-mate.blogspot.com/2013/07/classification-of-microcontrollers.html>



# Processor, Microcontrolador, DSP, FPGA, ASIC?

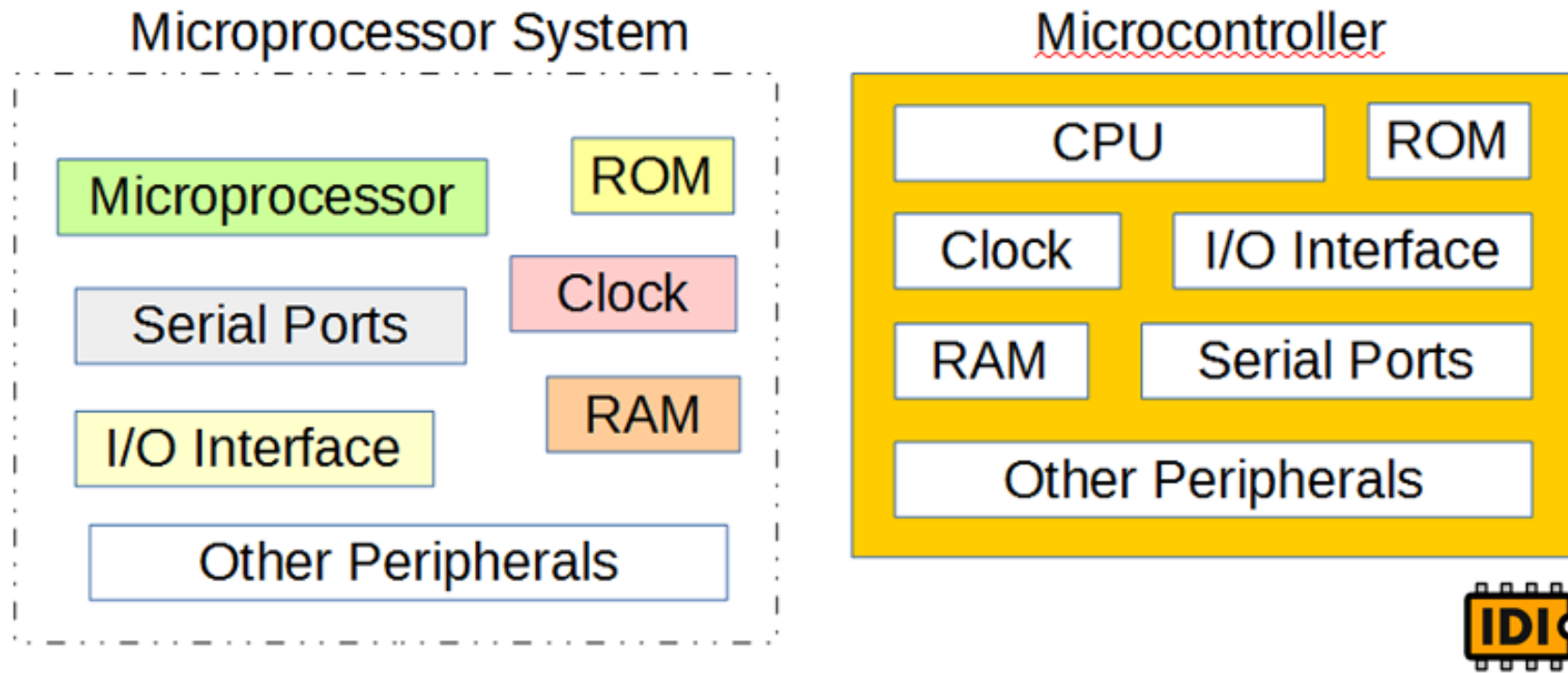
## Classification of Microprocessors







# Processador x Microcontrolador



DSP (Digital Signal Processor) – microprocessador otimizado para processamento digital de sinais

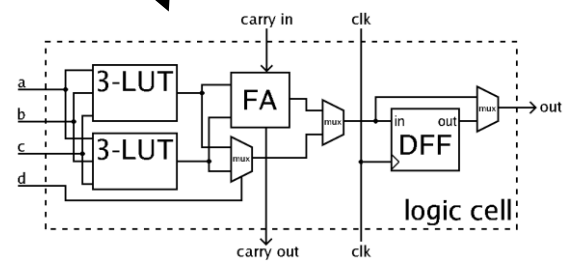
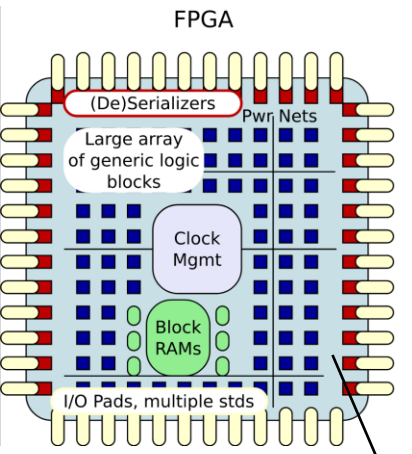
Filtro FIR:

$$y[n] = b_0x[n] + b_1x[n - 1] + \dots + b_Nx[n - N]$$
$$= \sum_{i=0}^N b_i \cdot x[n - i],$$





# FPGA (Field Programmable Gate Array)



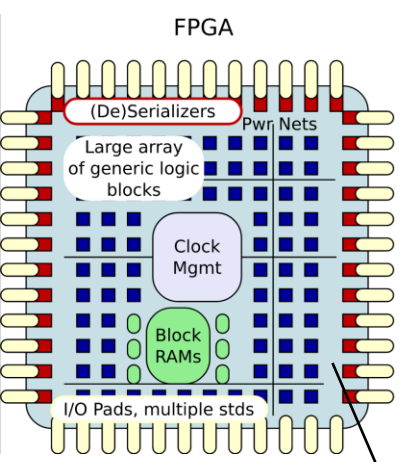


# FPGA (Field Programmable Gate Array) x ASIC (Application Specific Integrated Circuits)

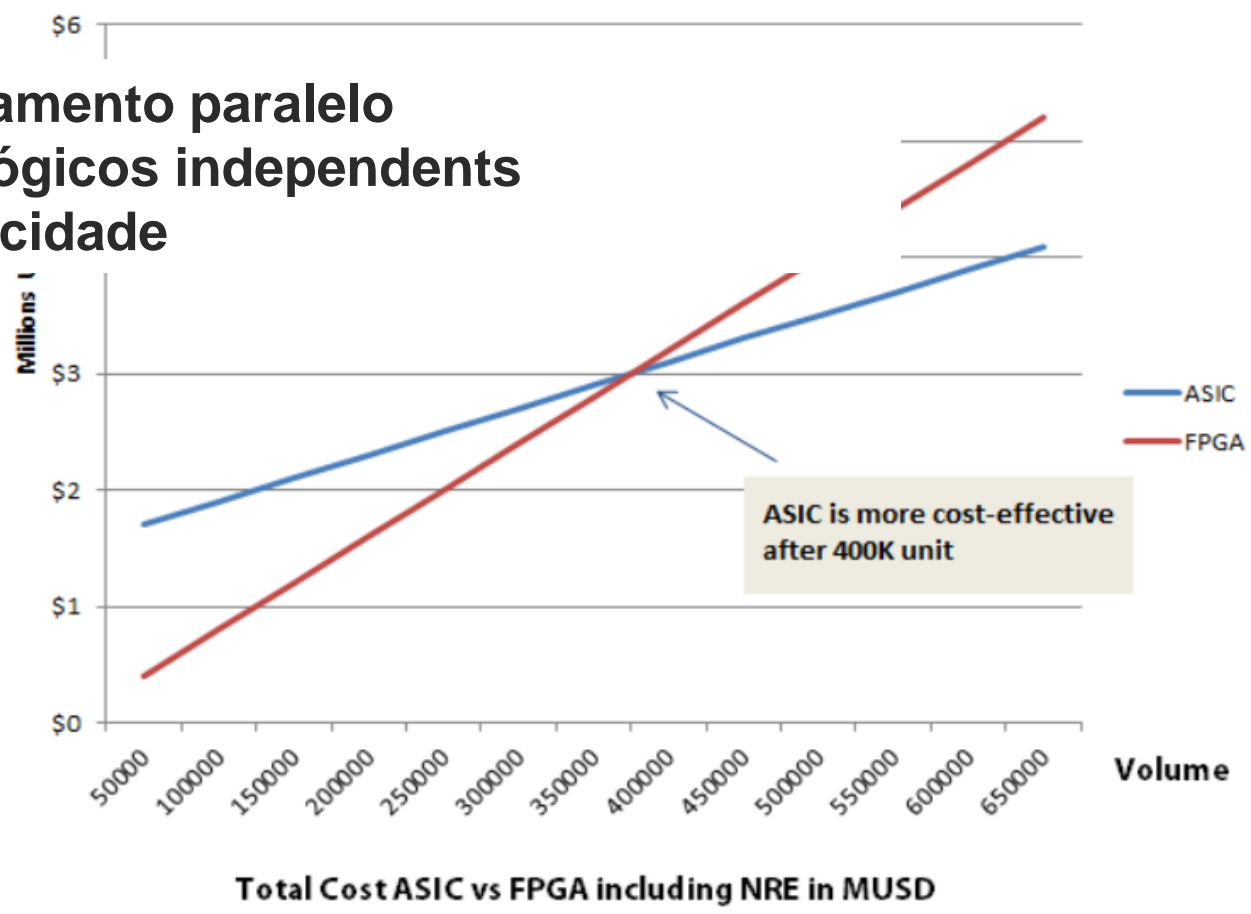
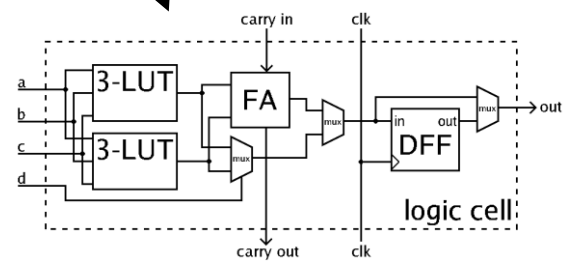
FPGA

ASIC NRE: \$1.5M  
ASIC Unit Cost: \$4  
  
FPGA NRE: \$0  
FPGA Unit Cost: \$8

ASSP (Application Specific Standard Part)



- Processamento paralelo
- Blocos Lógicos independentes
- Alta velocidade





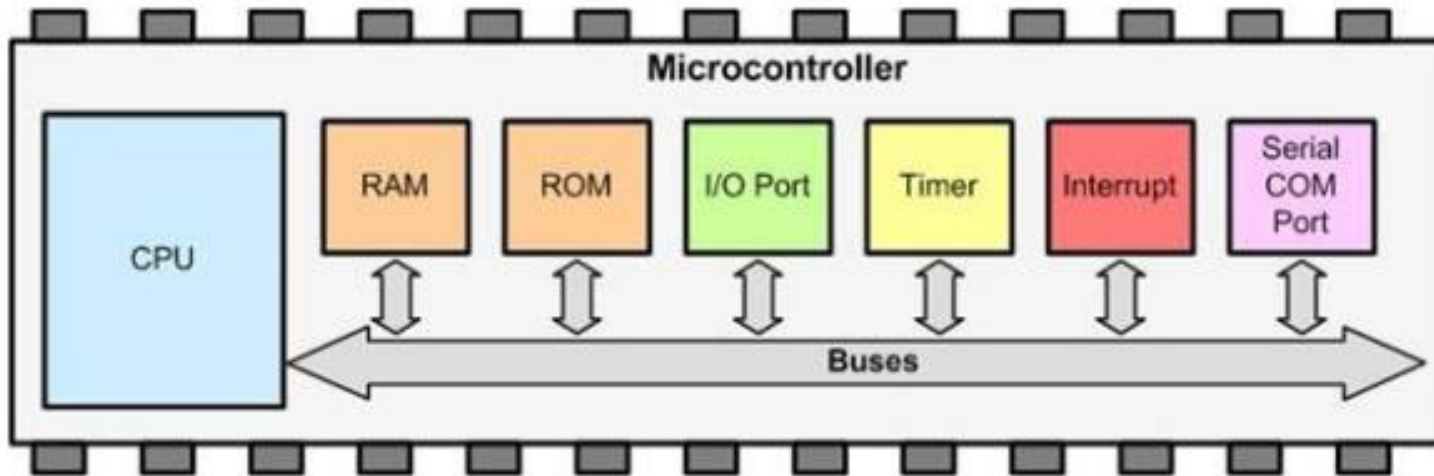
# FPGA vs. DSP

Category (MMAC Performance Range)	Device Cost Range	Minimum cents/MMAC for DSPs	Minimum cents/MMAC for FPGAs
LOW ( <300 MMAC)	< \$10	1.8	-
MED (300~1000 MMAC)	\$10~\$30	1.6	1.4
	\$30~\$100	3	2.8
HIGH ( >1000 MMAC)	\$100~\$300	5.8	2.9
	\$300~\$1000	13.4	4.2
	\$1000~\$10000	-	20

MMAC (millions of multiply-accumulate operations per second)



# μControlador

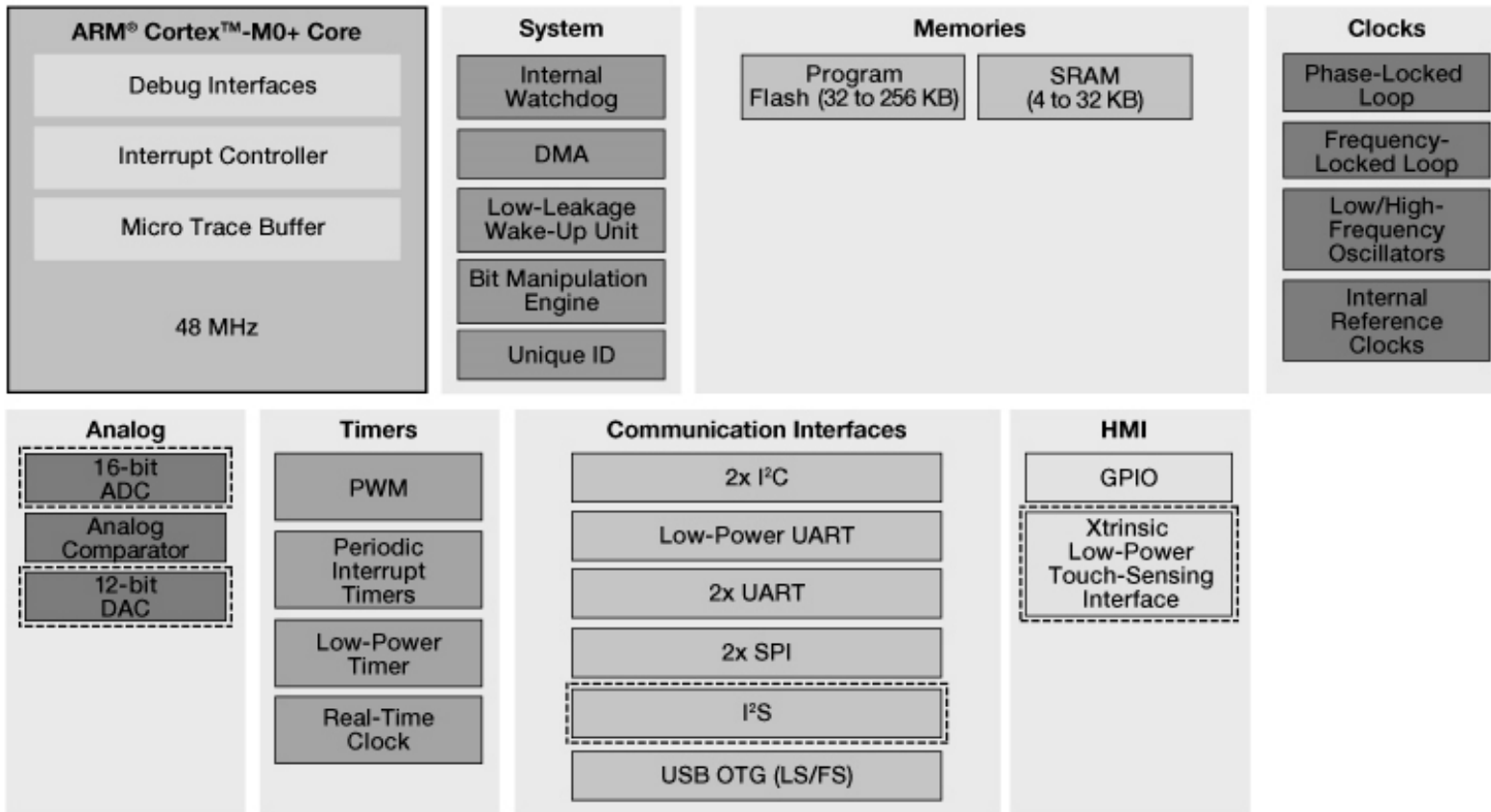


# Pesquise os diferentes tipos de periféricos de um microcontrolador.

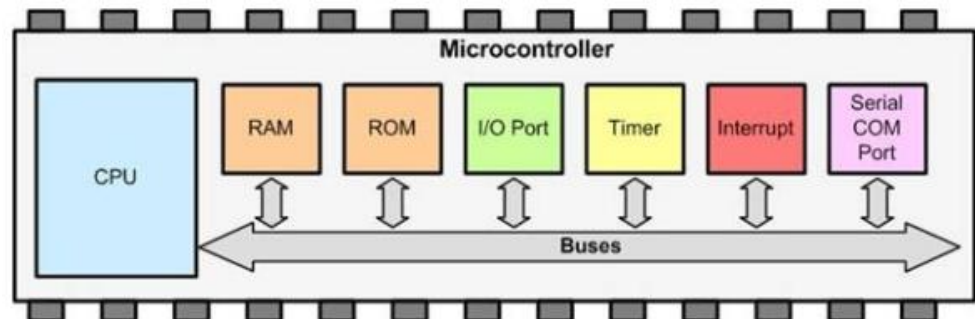


# Microcontrolador KL25Z

## Kinetis KL2x MCU Family Block Diagram

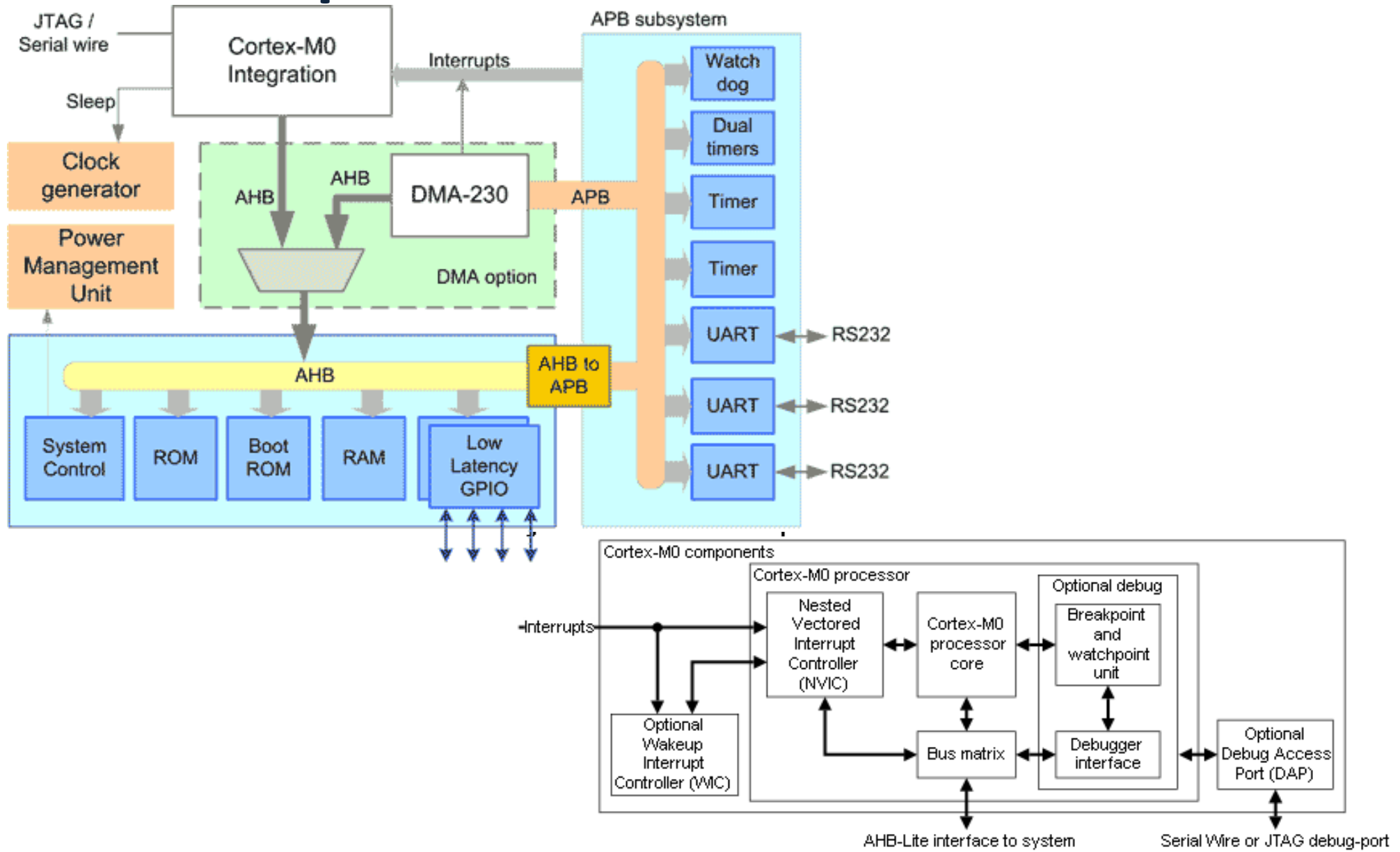


□ Standard    □ Optional





# Microarquitetura do Cortex M0+

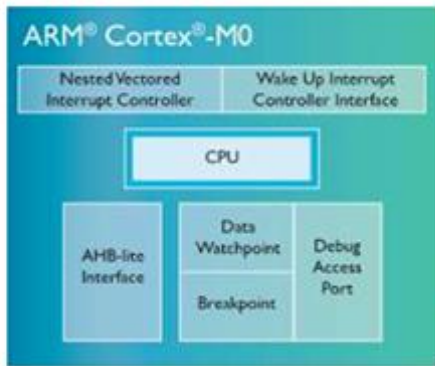




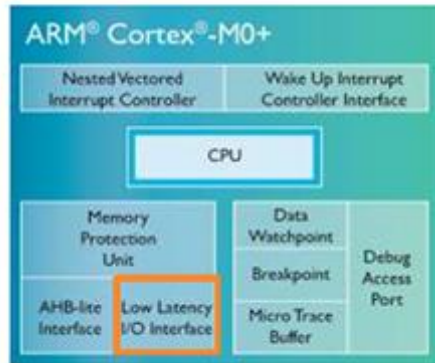
# ARM Cortex M

## ARM Cortex-M Product Line

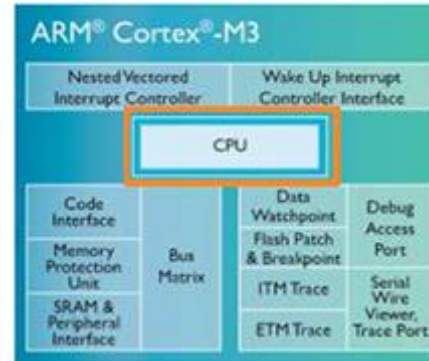
Consistent 32 bit processor architecture across all applications



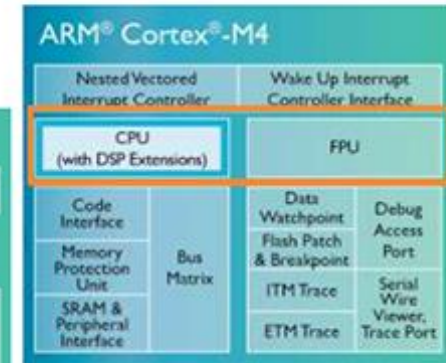
Lowest cost  
Low power



Lowest power  
Outstanding energy efficiency



Performance efficiency  
Feature rich connectivity



Digital Signal Control  
Processor with DSP  
Accelerated SIMD  
Floating point





# Família ARM Cortex-M

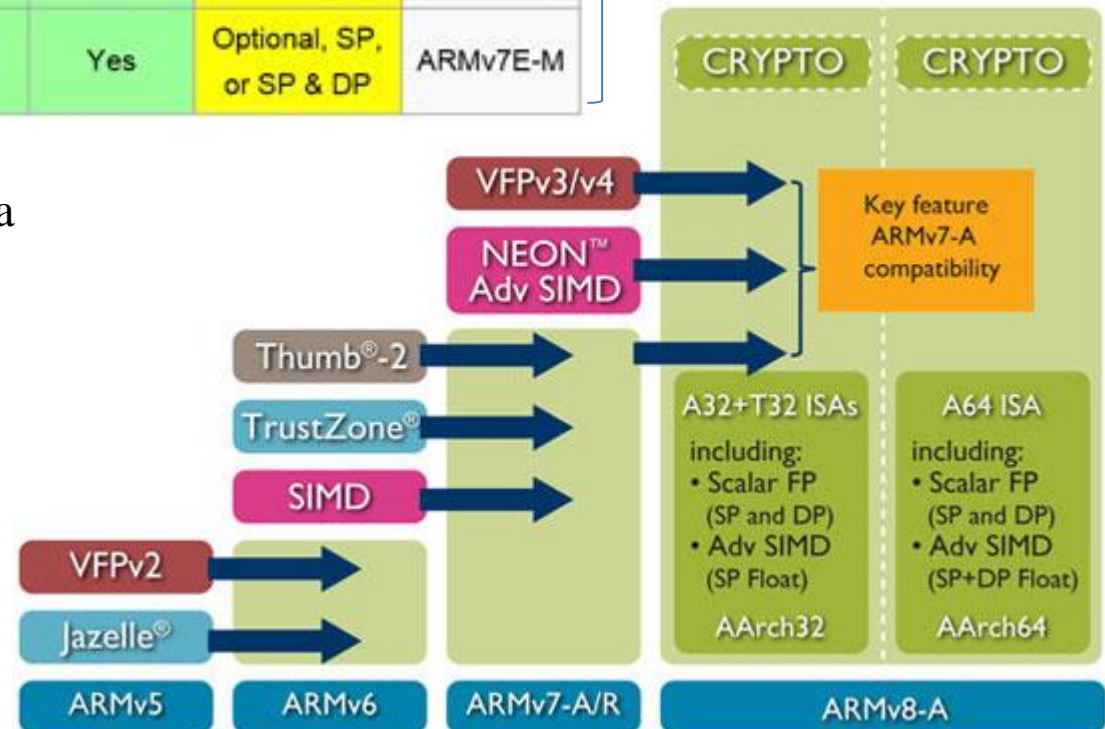
ARM Cortex-M Instruction sets<sup>[6][7]</sup>

ARM Cortex-M	Thumb	Thumb-2	Hardware multiply	Hardware divide	Saturated math	DSP extensions	Floating-Point Unit (FPU)	ARM architecture
Cortex-M0 <sup>[1]</sup>	Entire	Subset	1 or 32 cycle	No	No	No	No	ARMv6-M
Cortex-M0+ <sup>[2]</sup>	Entire	Subset	1 or 32 cycle	No	No	No	No	ARMv6-M
Cortex-M1 <sup>[3]</sup>	Entire	Subset	3 or 33 cycle	No	No	No	No	ARMv6-M
Cortex-M3 <sup>[4]</sup>	Entire	Entire	1 cycle	Yes	Yes	No	No	ARMv7-M
Cortex-M4 <sup>[5]</sup>	Entire	Entire	1 cycle	Yes	Yes	Yes	Optional, SP	ARMv7E-M
Cortex-M7	Entire	Entire	1 cycle	Yes	Yes	Yes	Optional, SP, or SP & DP	ARMv7E-M

Von Neumann

Harvard

Saturated Math → Algoritmos eficientes para processamento de sinais



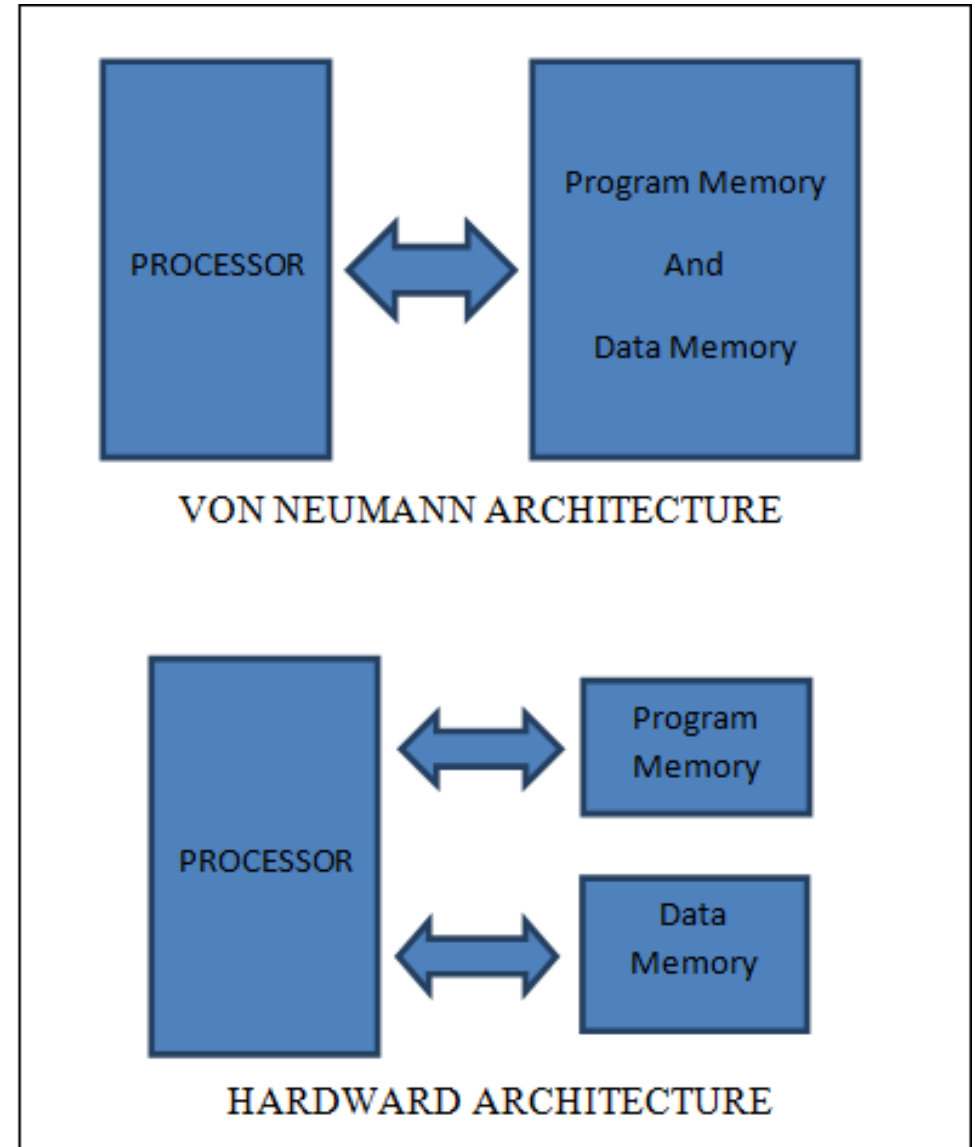


# Von Neumann vs Harvard

- 1 vs 2 Barramentos
  - Custo
  - Complexidade
  - Acesso simultâneo a instrução e dados
  - Implementação em microcontroladores
  - ~~Velocidade~~



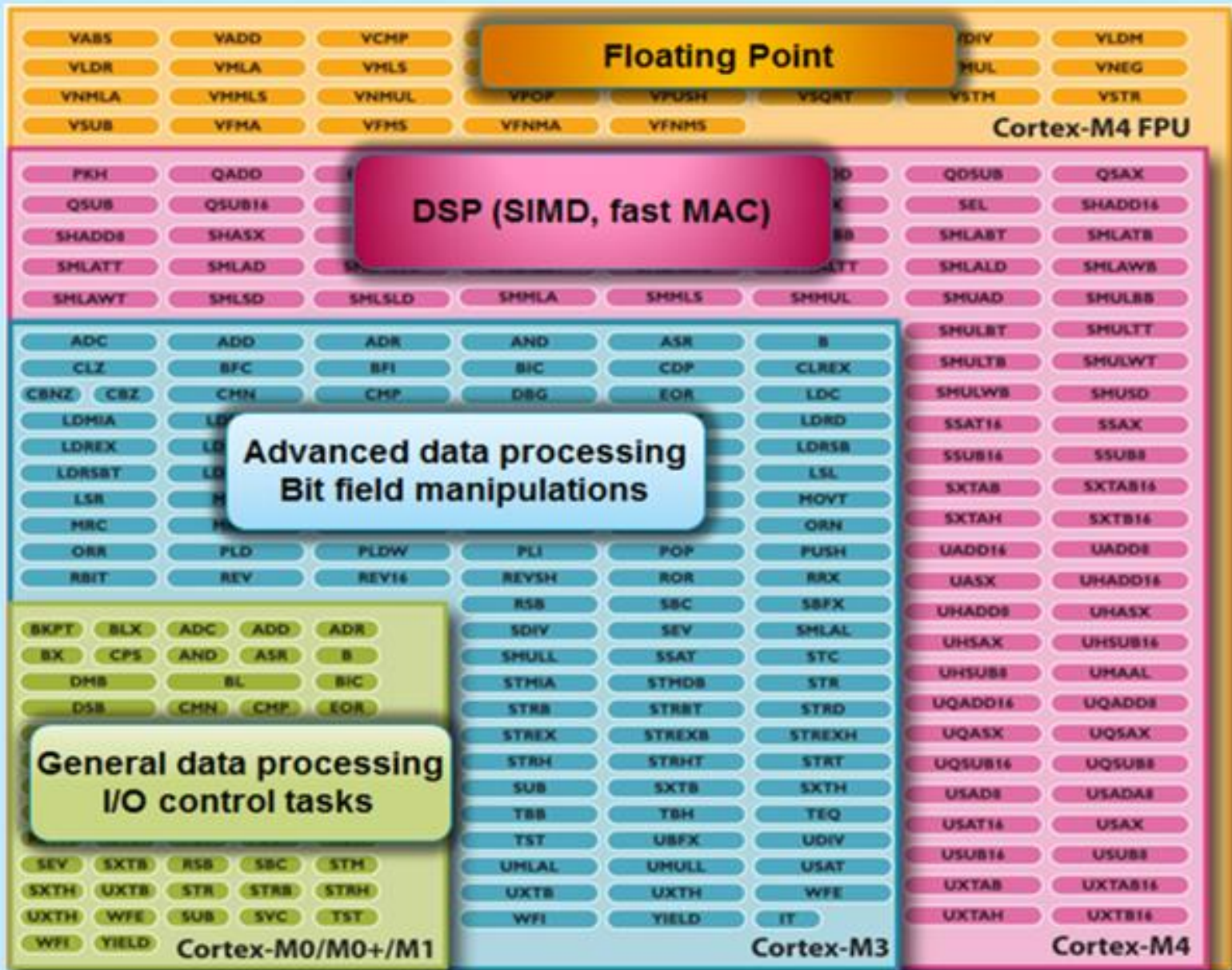
Cache e Pipeline





# Arquitetura RISC (Reduced Instruction Set Computer)

16 Bit Thumb & 16/32 Bit Thumb2









# RISK vs CISC

- RISK (Load-Store)

- Instruções e endereçamento simples
- Instruções executadas em um ciclo de clock
- Tamanho de instrução fixo
- Pipeline
- Complexidade

- CISC (Register-Memory)

- Endereçamento complexo
- Rico em instruções
- Instruções executadas em mais de um ciclo de clock
- Tamanho da instrução de tamanho variável
- Sem pipeline





# Como Escolher um Microcontrolador

1) Faça uma lista das necessidades de hardware e interfaces.

- Utilize o diagrama de blocos da arquitetura do sistema
- Identifique os tipos de comunicação necessários (UART, I2C, SPI, USB, Ethernet, wireless, bluetooth etc.)
- Identifique as entradas e saídas digitais e analógicas

2) Examine a arquitetura do software.

- Necessidade de ponto flutuante
- Operações matemáticas
- DSP
- Velocidade de processamento



# Como Escolher um Microcontrolador

- 3) Escolha a precisão da arquitetura.
  - 8, 16 ou 32 bits?
- 4) Identifique a necessidade de memória
  - Flash, RAM
  - Memória externa
- 5) Pesquise os microcontroladores existentes
  - Experiência pessoal
  - Fabricantes, famílias, documentação
  - Toolchain (IDE, compilador etc.)
  - Suporte



# Como Escolher um Microcontrolador

- 6) Verifique o custo e consumo de potência
- 7) Disponibilidade do componente
  - Estoque
  - Distribuidores
  - Ciclo de vida
- 8) Selecione um kit de desenvolvimento