

MICROPROCESSADORES E MICROCONTROLADORES



MICROCONTROLADORES

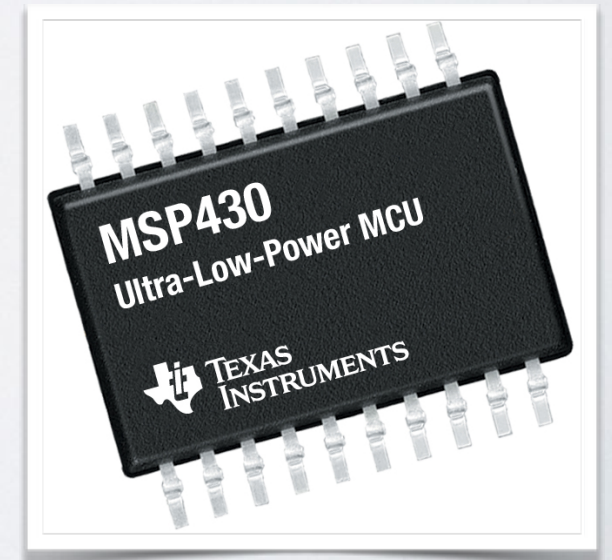
Microprocessador + memória + relógio + ...



MICROCONTROLADORES

Microprocessador + memória + relógio + ...

Computer on a chip



MICROCONTROLADORES

Quanto produtos na sua casa
possuem "computer chips"?

MICROCONTROLADORES

Quantos produtos na sua casa
possuem "computer chips"?



MICROCONTROLADORES

Quantos produtos na sua casa possuem "computer chips"?



MICROCONTROLADORES

Quantos produtos na sua casa
possuem "computer chips"?



MICROCONTROLADORES

Quantos produtos na sua casa
possuem "computer chips"?



MICROCONTROLADORES

Quantos produtos na sua casa
possuem "computer chips"?



**Electronic control units:
Módulos de controle
automotivo**

Airbag

**Portas, janelas, assentos e
luzes**

Motor

Transmissão

Freios

MICROCONTROLADORES

Quantos produtos na sua casa

Sistemas embarcados

"Invisíveis"



**Control units:
controle
ativo**

Airbag

**Portas, janelas, assentos e
luzes**

Motor

Transmissão

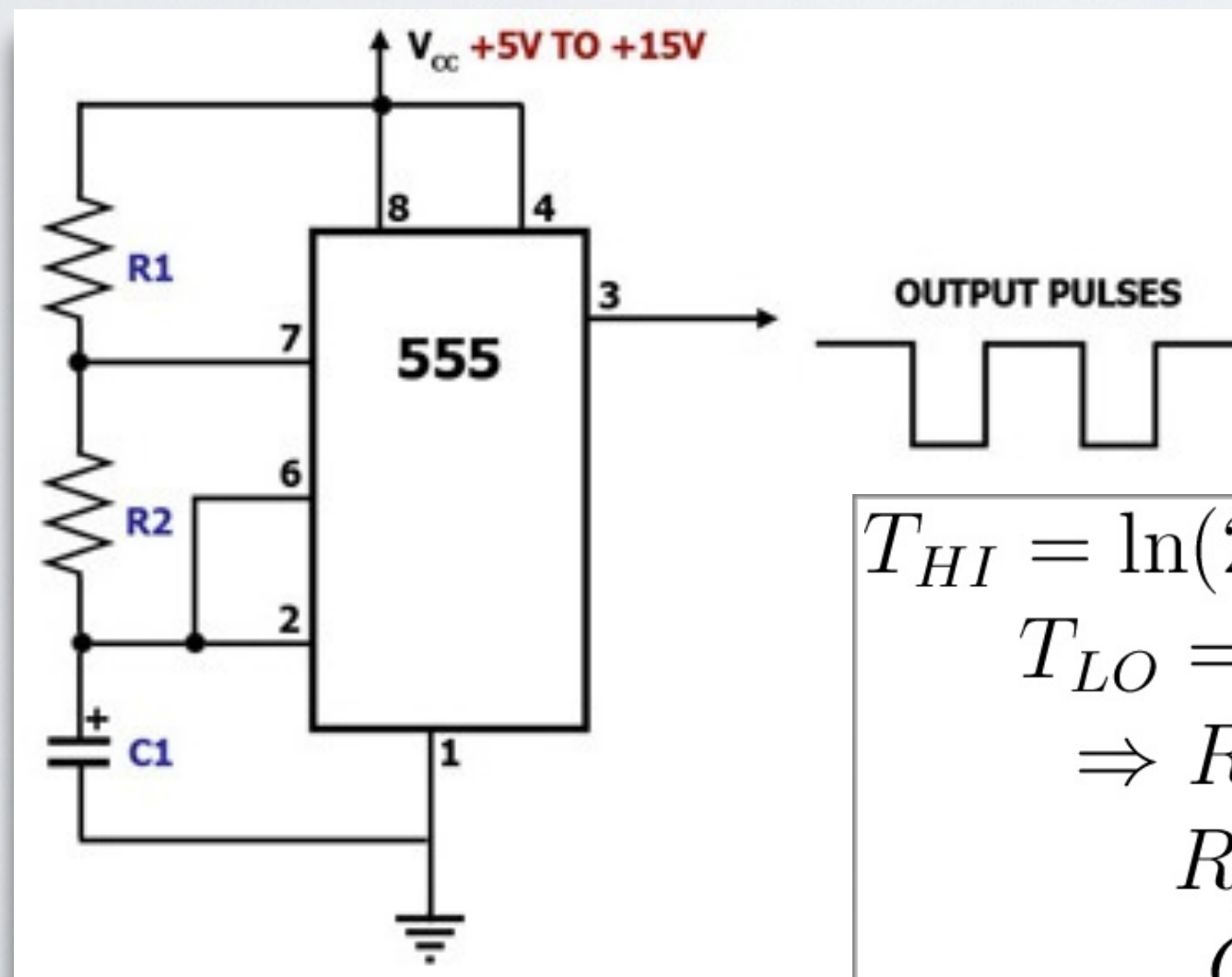
Freios

MICROCONTROLADORES

Exemplo: escova elétrica que pausa a cada 30s, para que o usuário passe a escovar outra área

MICROCONTROLADORES

Exemplo: escova elétrica que pausa a cada 30s, para que o usuário passe a escovar outra área



$$T_{HI} = \ln(2)(R_1 + R_2)C = 29s$$

$$T_{LO} = \ln(2)R_2C = 1s$$

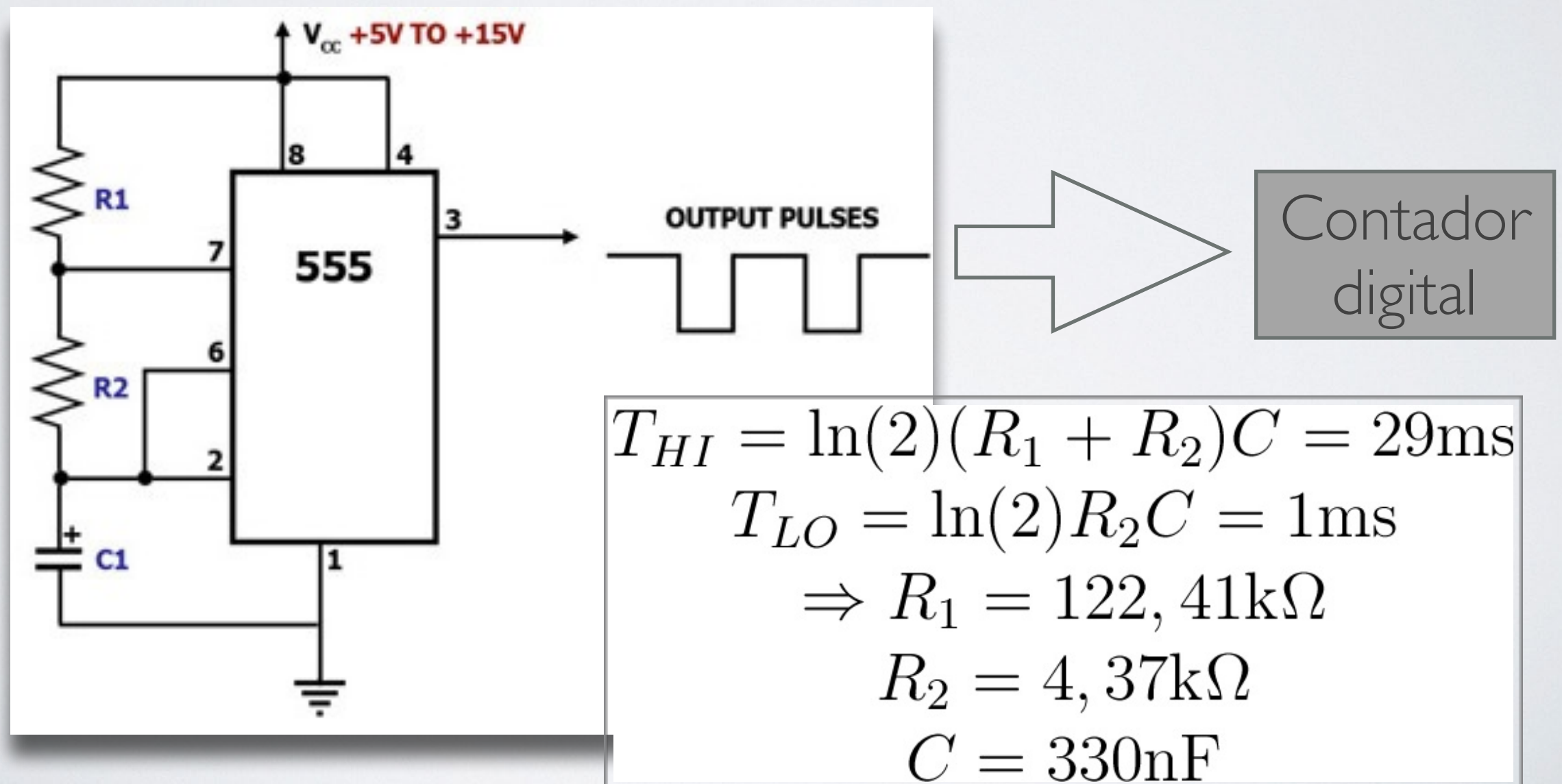
$$\Rightarrow R_1 = 122,41k\Omega$$

$$R_2 = 4,37k\Omega$$

$$C = 330\mu F$$

MICROCONTROLADORES

Exemplo: escova elétrica que pausa a cada 30s, para que o usuário passe a escovar outra área



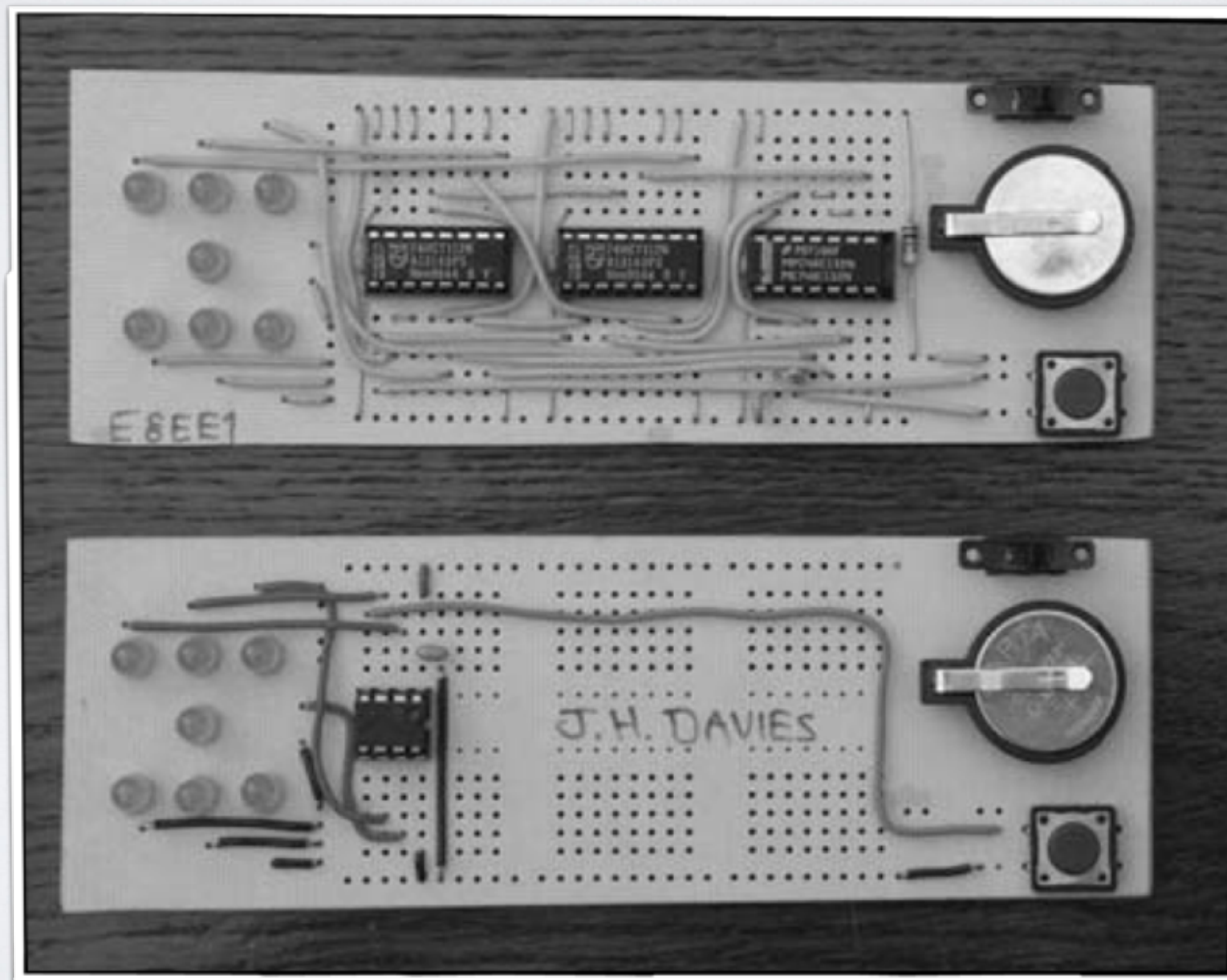
MICROCONTROLADORES

Exemplo: escova elétrica que pausa a cada 30s, para que o usuário passe a escovar outra área



MICROCONTROLADORES

Exemplo: "dado" digital



MICROCONTROLADORES

Como desenvolver sistemas embarcados?

MICROCONTROLADORES

Como desenvolver sistemas embarcados?

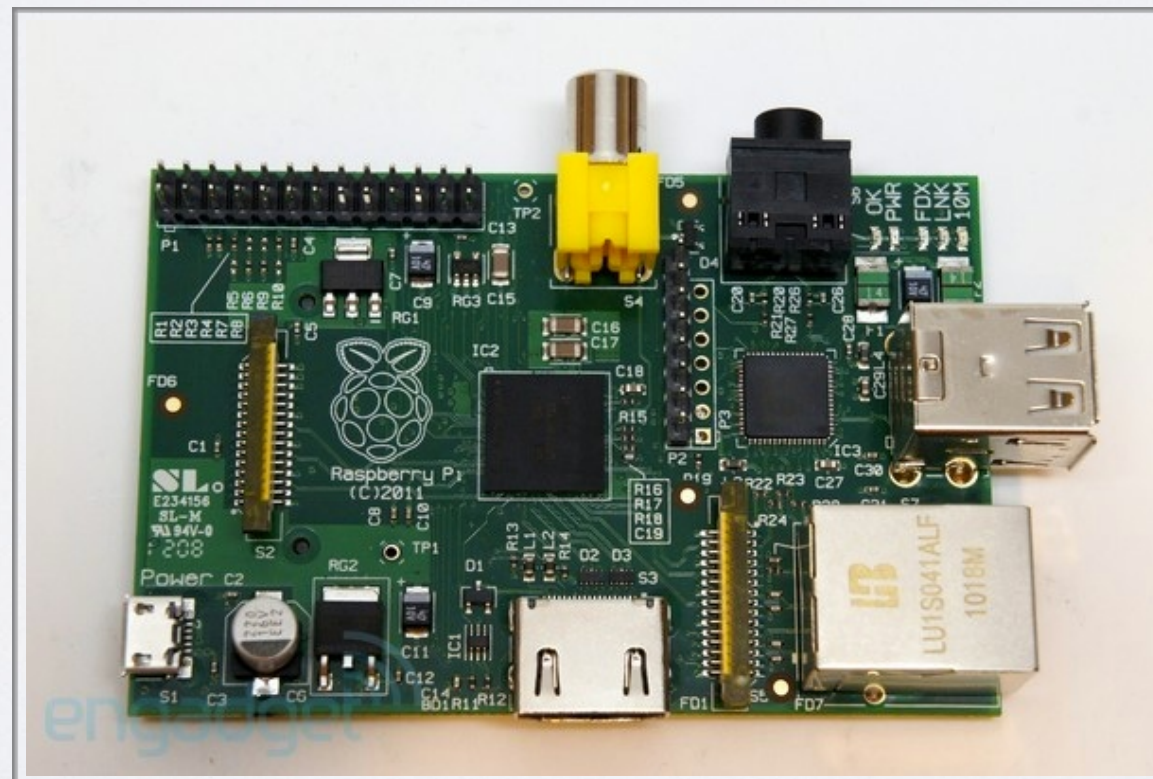
Computadores completos - hardware e software
facilmente disponíveis



MICROCONTROLADORES

Como desenvolver sistemas embarcados?

Computadores completos - hardware e software
facilmente disponíveis

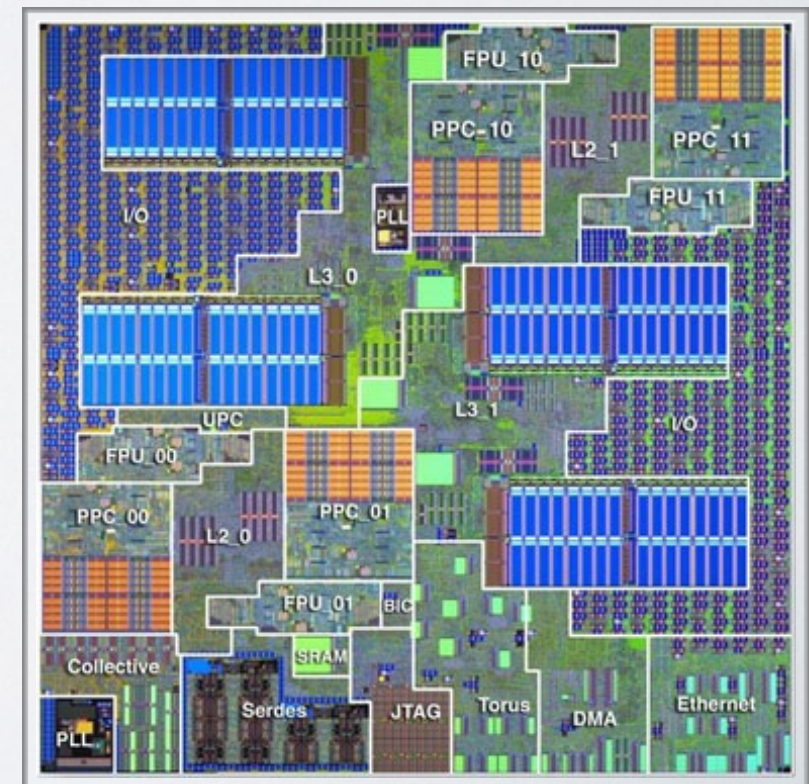
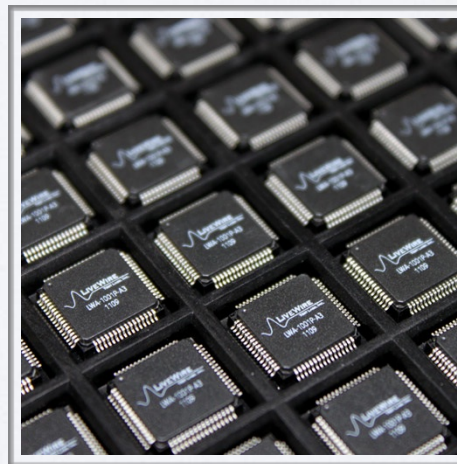
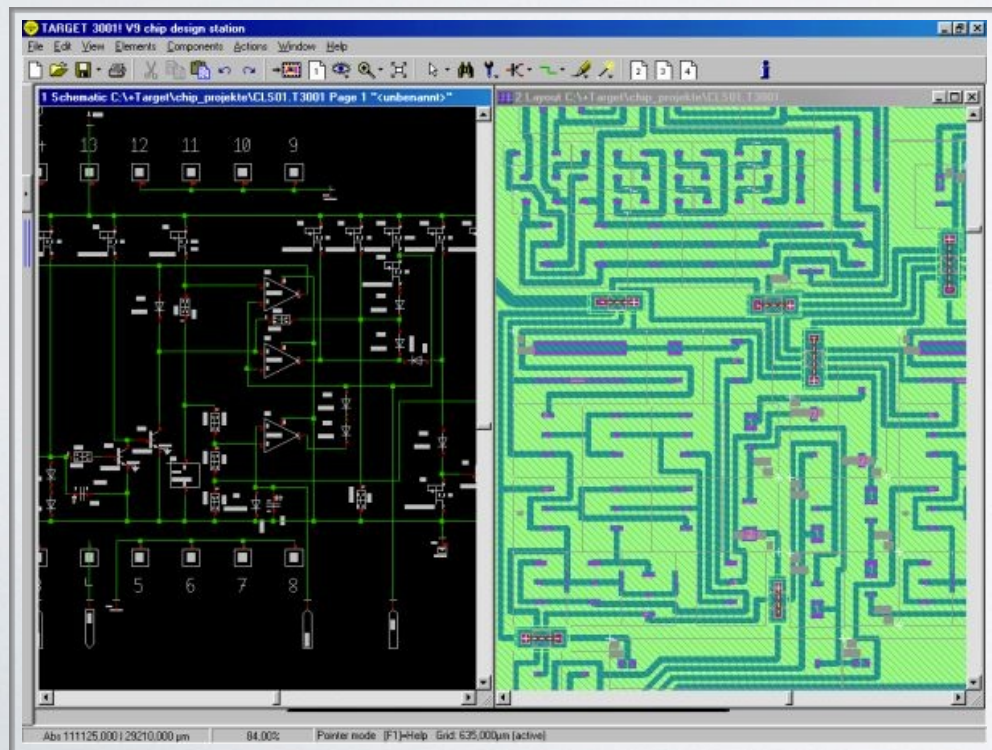


MICROCONTROLADORES

Como desenvolver sistemas embarcados?

ASICs - application-specific integrated circuits

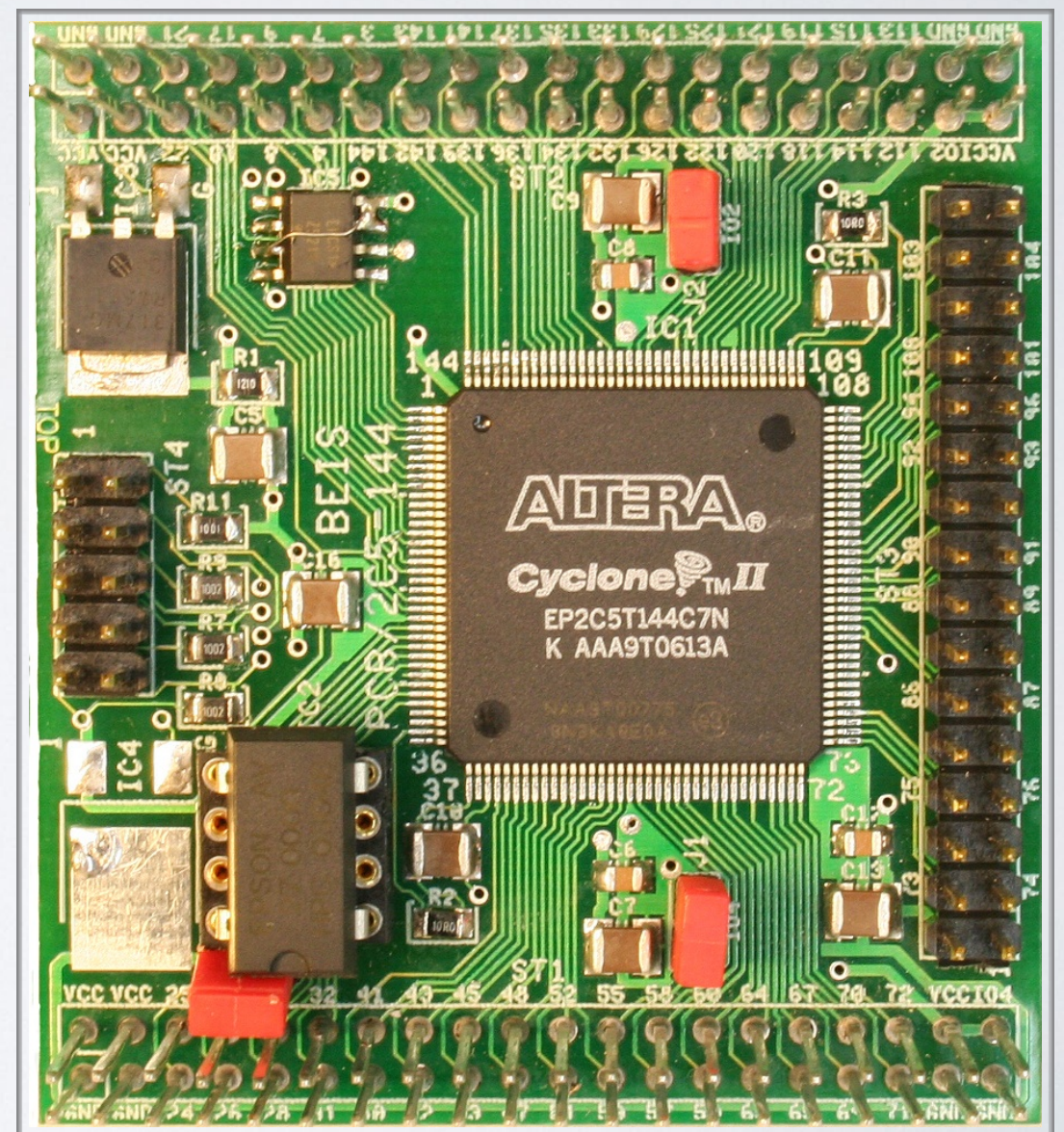
CIs feitos sob medida para a aplicação



MICROCONTROLADORES

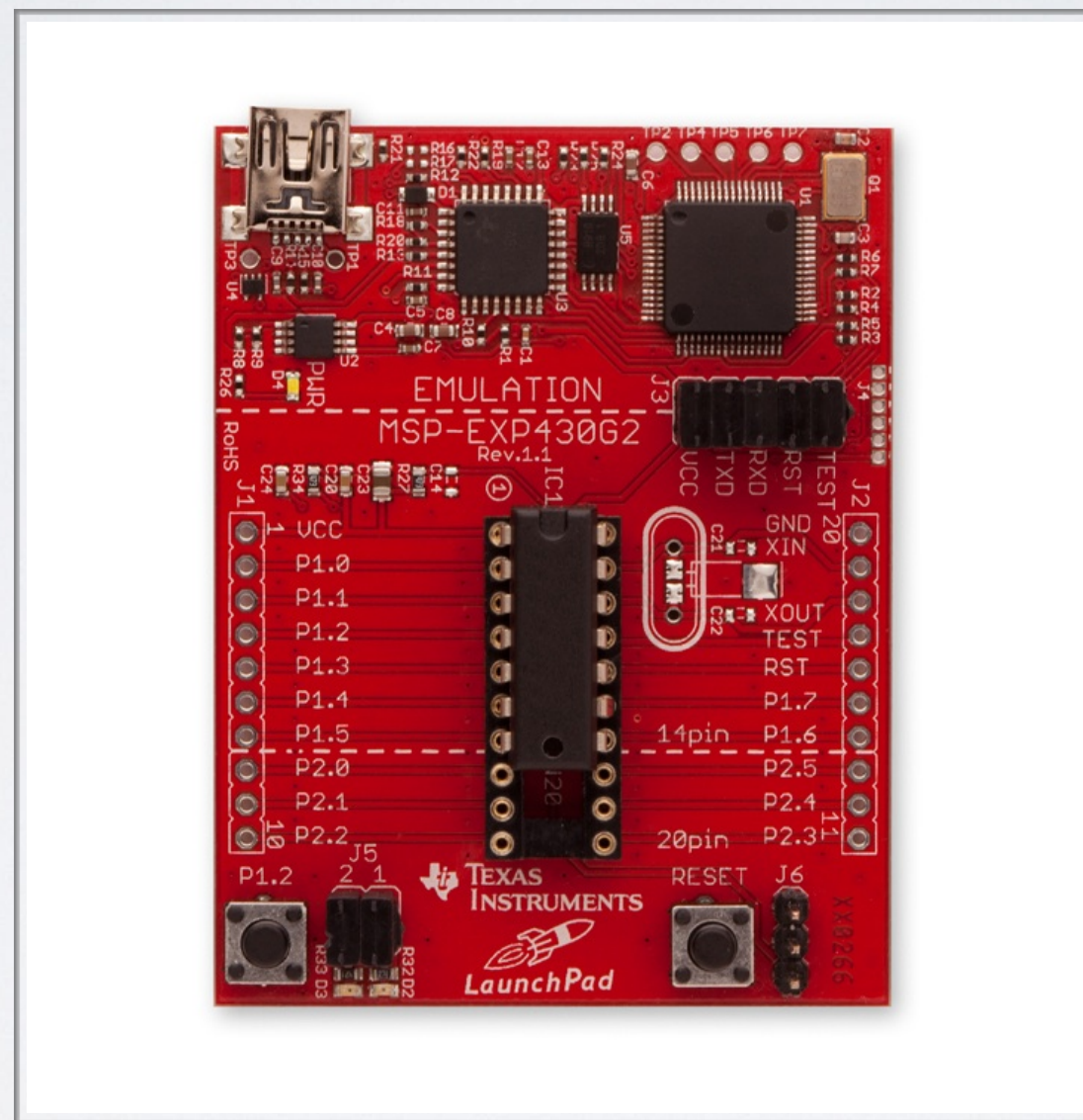
Como desenvolver sistemas embarcados?

FPGAs - portas lógicas e
flip-flops conectados por
linguagem de hardware
(VHDL etc.)



MICROCONTROLADORES

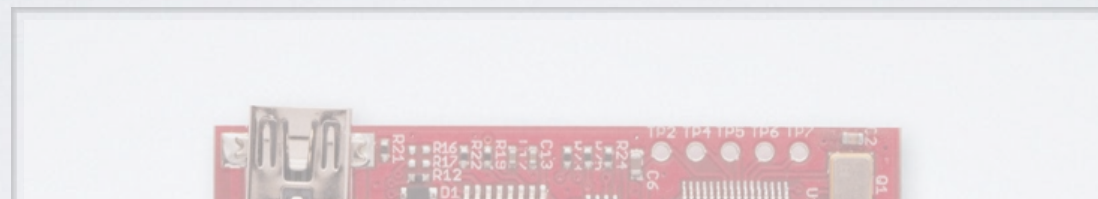
Como desenvolver sistemas embarcados?



Microcontroladores

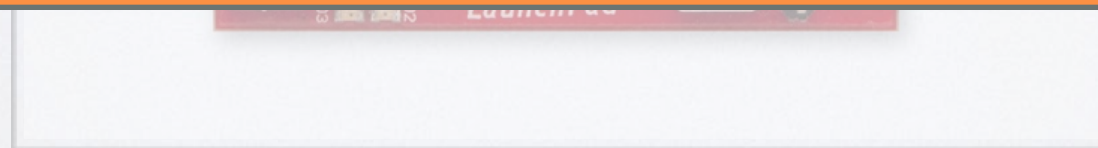
MICROCONTROLADORES

Como desenvolver sistemas embarcados?



Não utiliza sistema operacional, somente um software específico para a aplicação

Ou então utiliza-se um RTOS (real-time operating system), geralmente menor que um kilobyte



Microcontroladores

MICROCONTROLADORES

Como desenvolver sistemas embarcados?

- programa do usuário carrega o RTOS, diferente de sistemas operacionais de computadores, onde o programa é chamado através do SO

Microcontroladores

MICROCONTROLADORES

Memória RAM
(random-access),
para dados

Processador digital (unidade
lógico-aritmética, registradores,
decodificador de instruções etc.)

Memória ROM
(read-only), para o
programa

Barramentos de
dados e de
endereços

Relógio, para
sincronismo
do sistema

Entradas e saídas
digitais

Microcontroladores

MICROCONTROLADORES

Como desenvolver

Interfaces de comunicação: SPI, I2C, RS-232, USB, CAN, ethernet etc.

Memória não-volátil para dados permanentes, como números de série e endereços de rede

Temporizadores, para contar tempo em si, a duração de transições, gerar sinais periódicos etc.

Entradas e saídas analógicas

Watchdog timer, para reiniciar o sistema em caso de falha

Relógio em "tempo real" (hora, dia, mês, ano)

Microcontroladores

MICROCONTROLADORES

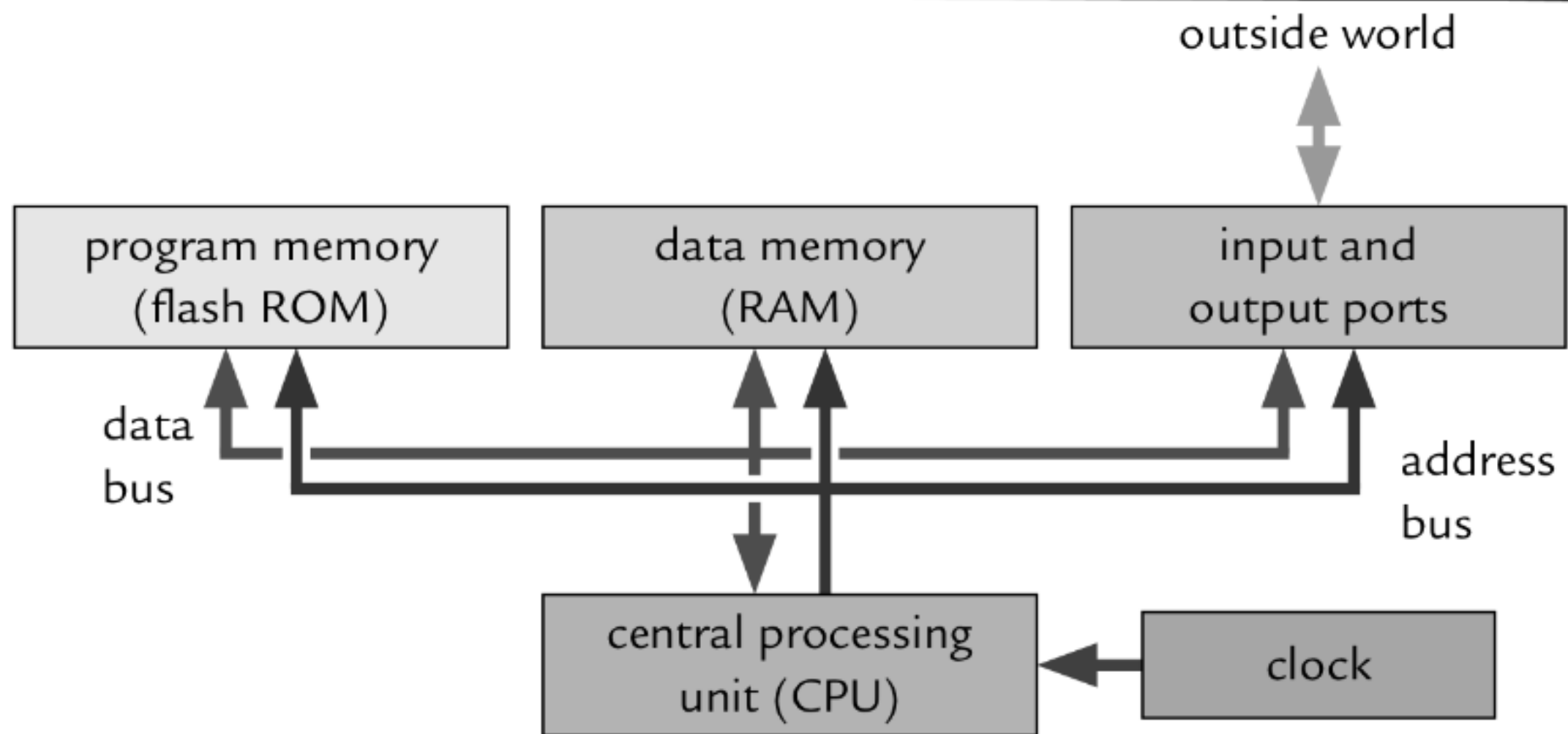
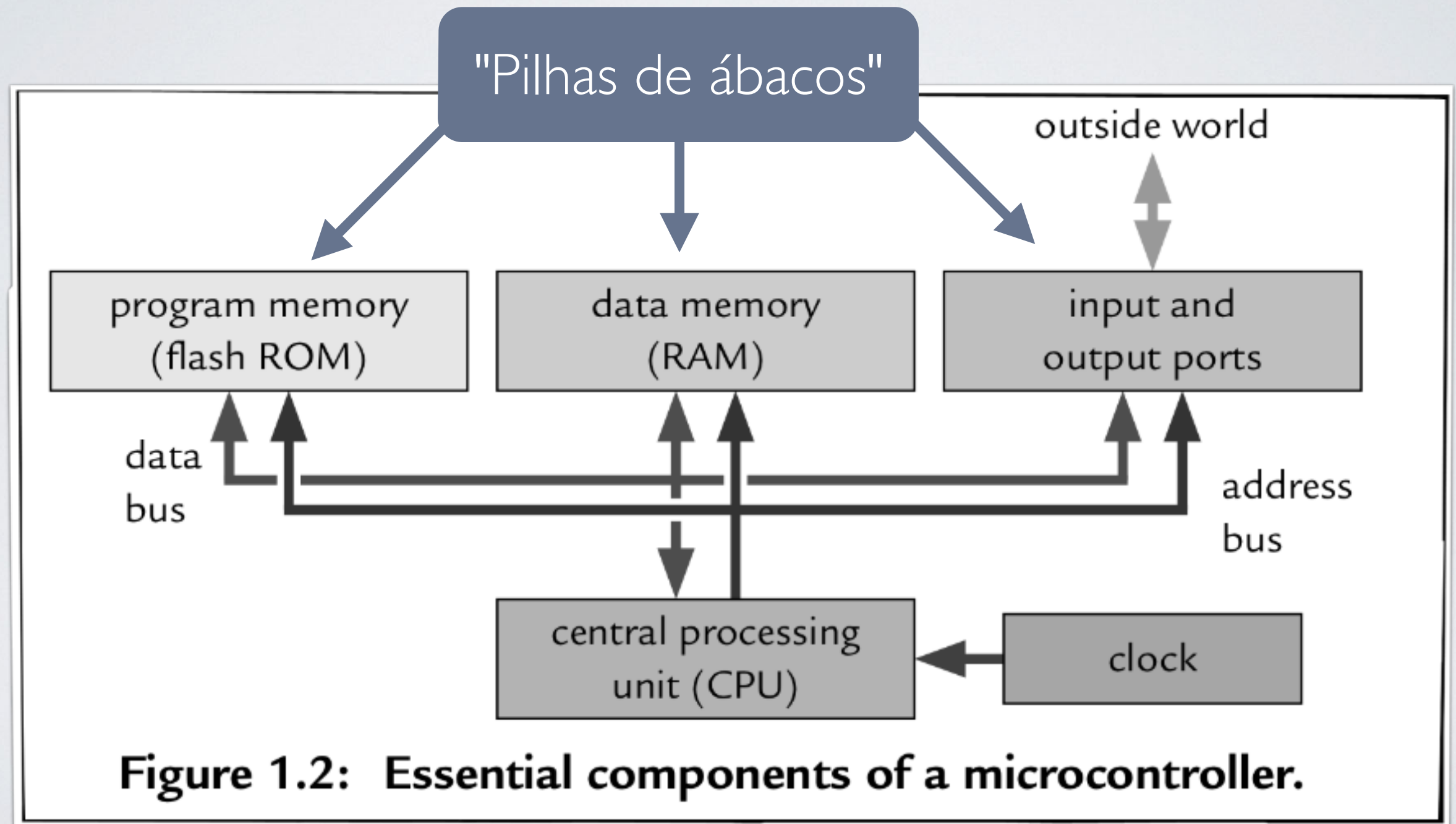
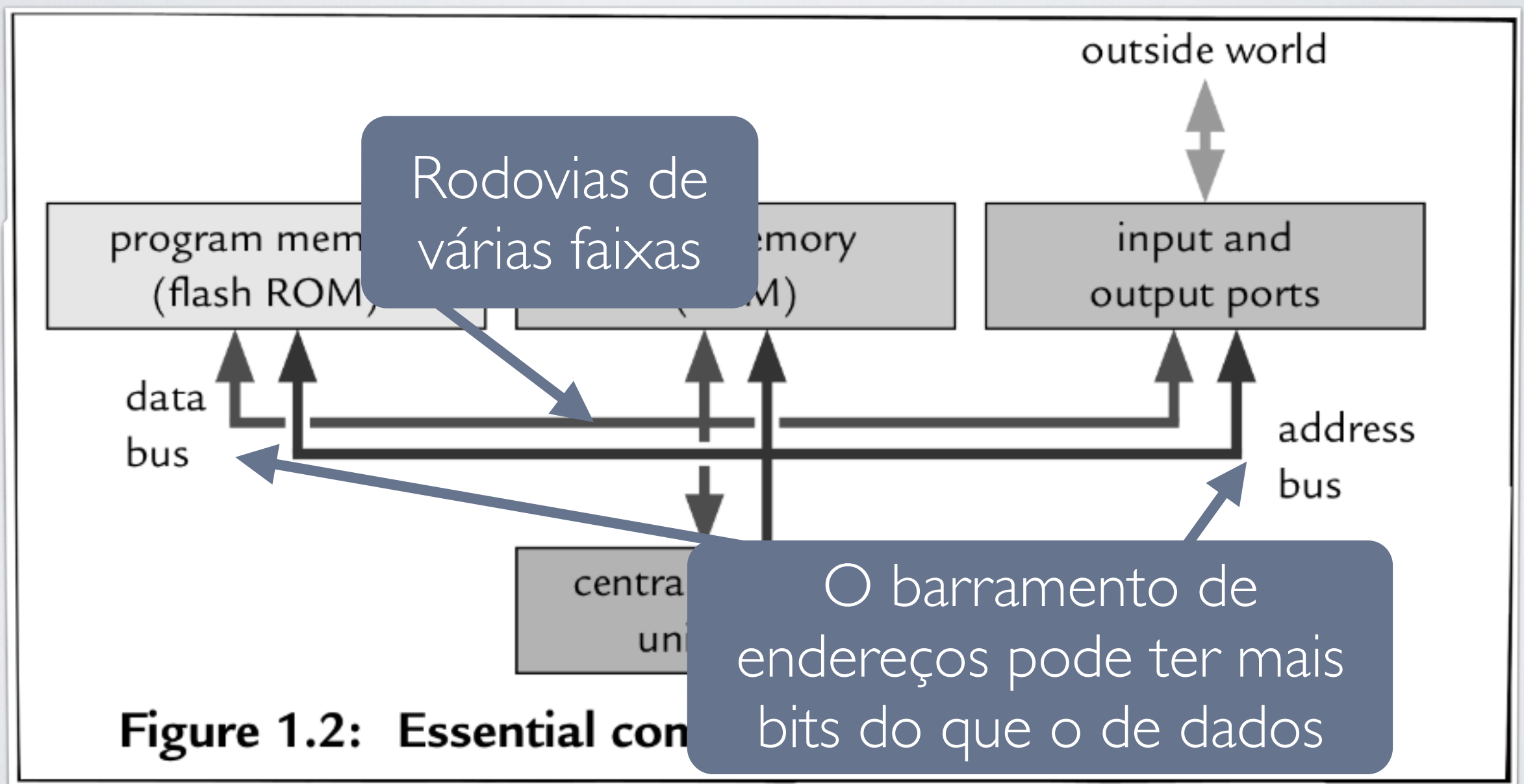


Figure 1.2: Essential components of a microcontroller.

MICROCONTROLADORES



MICROCONTROLADORES



MICROCONTROLADORES

Processadores são caracterizados pelo tamanho dos dados que ele pode processar (8 bits etc.)

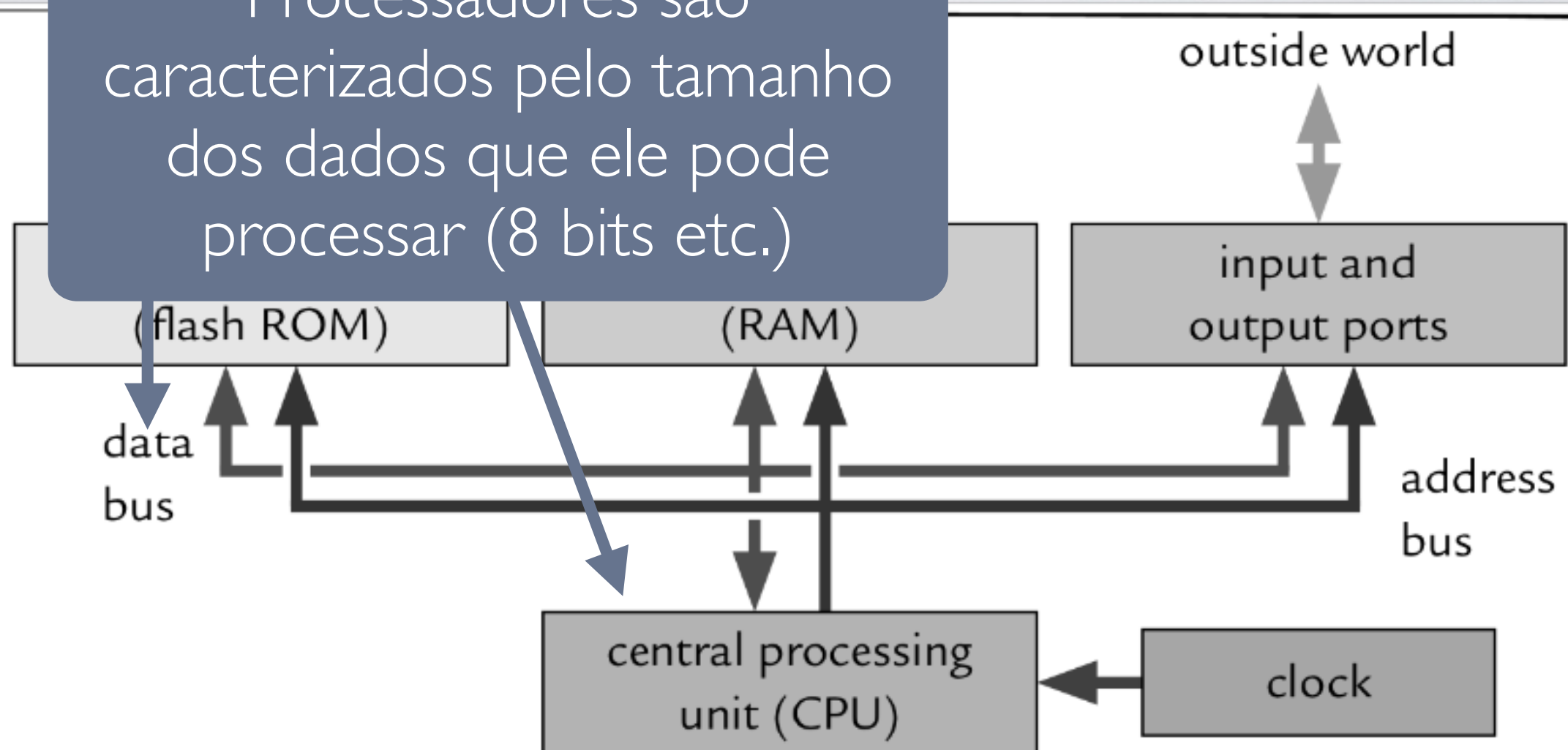


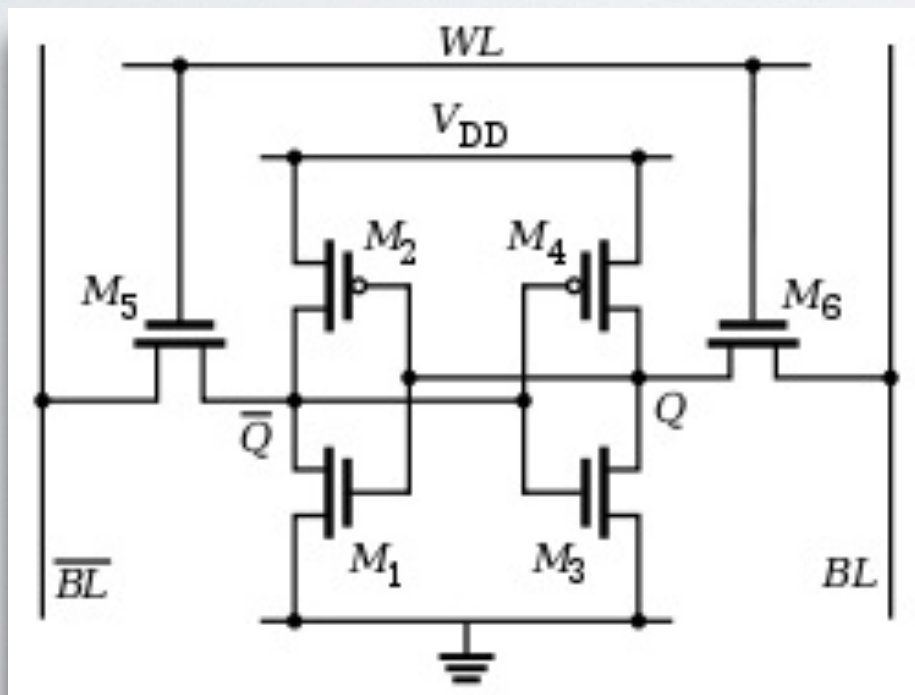
Figure 1.2: Essential components of a microcontroller.

MEMÓRIA

RAM (random-access memory): volátil,
perde o conteúdo quando acaba a energia

MEMÓRIA

RAM (random-access memory): volátil,
perde o conteúdo quando acaba a energia

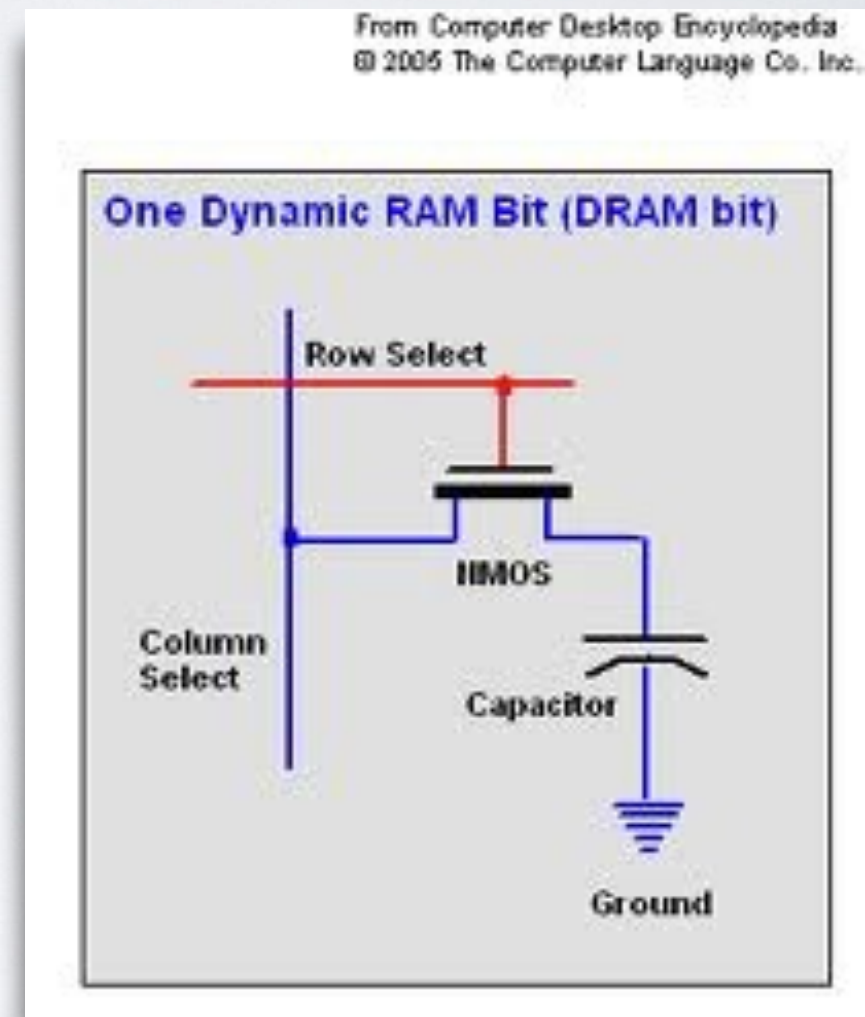


Estática - mantém os dados mesmo quando o clock foi parado. Precisa de 6 transistores

MEMÓRIA

RAM (random-access memory): volátil,
perde o conteúdo quando acaba a energia

Dinâmica - precisa ser
atualizada regularmente.
Precisa de 1 transistor,
ocupando menos espaço



MEMÓRIA

ROM (read-only memory): não-volátil, mantém o conteúdo quando acaba a energia. A escrita é bem mais lenta do que para a memória RAM

MEMÓRIA

ROM (read-only memory): não-volátil, mantém o conteúdo quando acaba a energia. A escrita é bem mais lenta do que para a memória RAM

Masked ROM - dados são escritos durante a manufatura do CI, utilizando uma máscara fotolitográfica.

MEMÓRIA

ROM (read-only memory): não-volátil, mantém o conteúdo quando acaba a energia. A escrita é bem mais lenta do que para a memória RAM

EPROM (electrically programmable ROM) - programável eletricamente, apagada com luz ultravioleta por cerca de 10 minutos

MEMÓRIA

ROM (read-only memory): não-volátil, mantém o conteúdo quando acaba a energia. A escrita é bem mais lenta do que para a memória RAM

OTP (one-time programmable memory) -
EPROM sem a "janela" para luz ultravioleta

MEMÓRIA

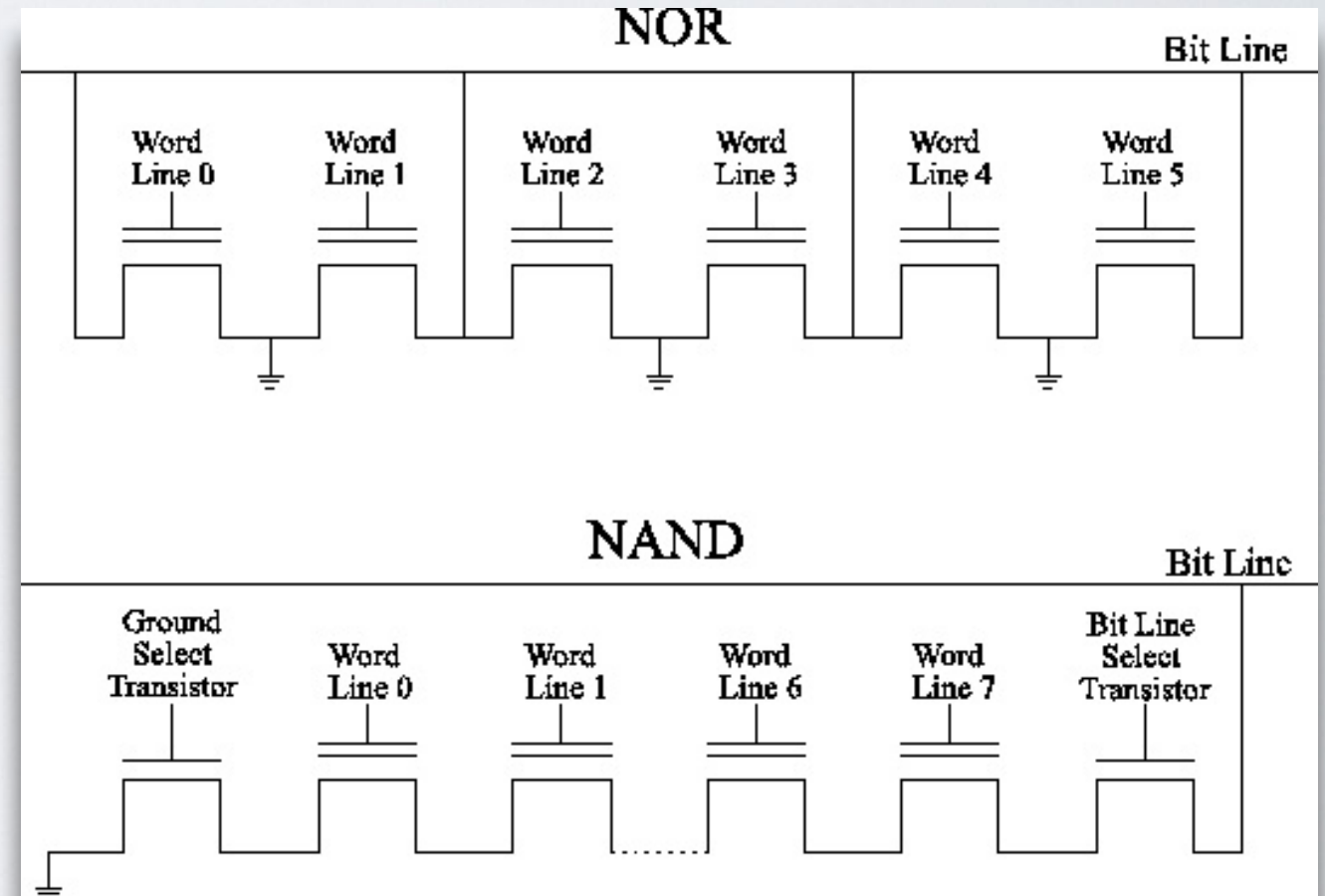
ROM (read-only memory): não-volátil, mantém o conteúdo quando acaba a energia. A escrita é bem mais lenta do que para a memória RAM

Flash - permite a leitura e escrita eletricamente, assim como a EEPROM. A memória flash só pode ser apagada em blocos, e a EEPROM, byte a byte.

MEMÓRIA

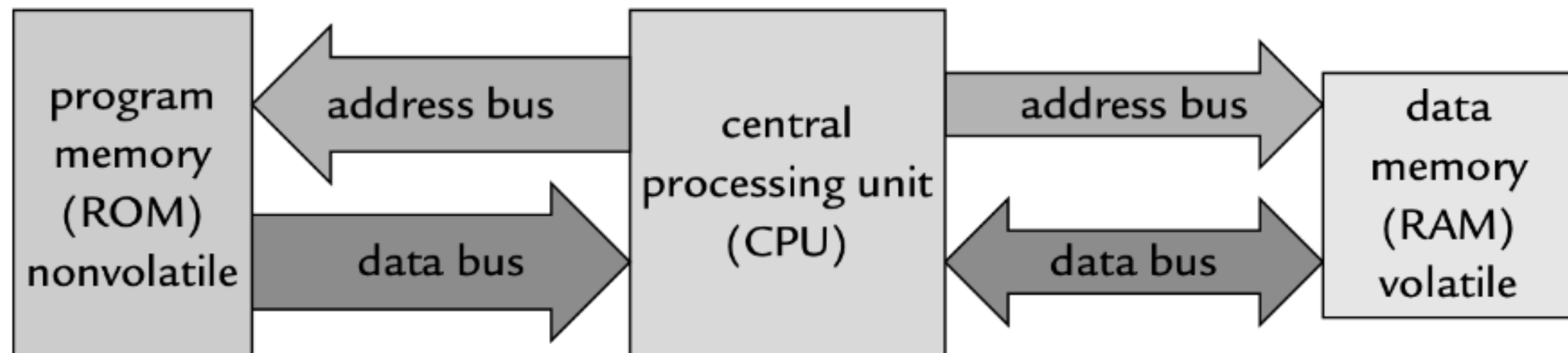
Os MSP430 com a letra F no seu part number vêm com memória flash.

Microcontroladores usam memória flash do tipo NOR, que permite acesso aleatório. Flash NAND só pode ser acessada em blocos.

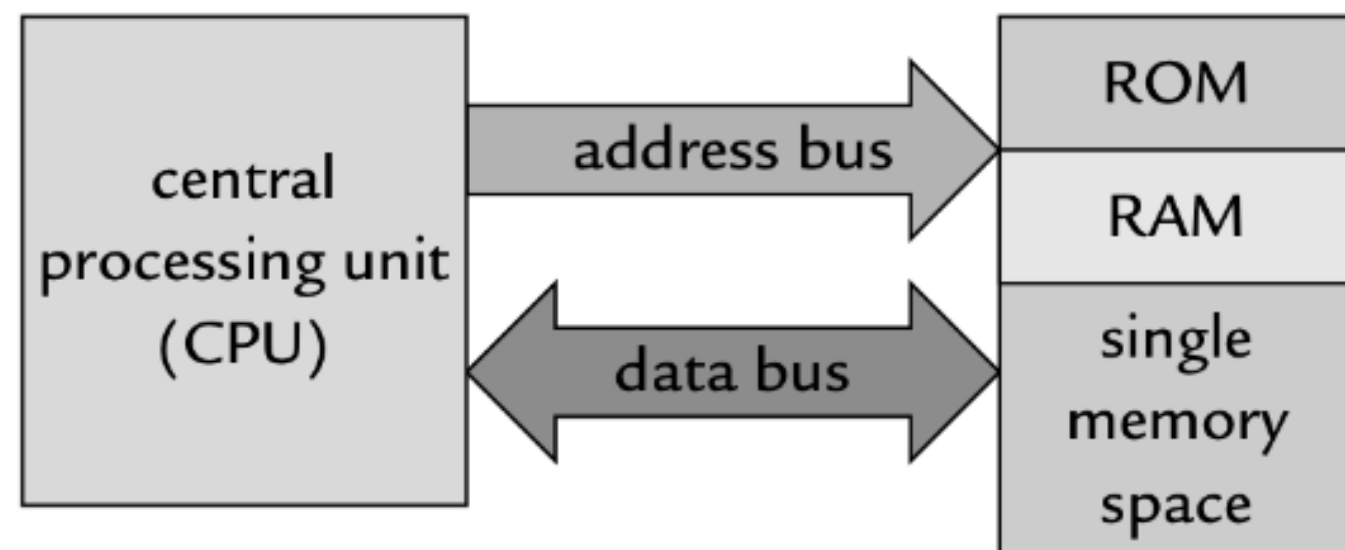


ARQUITETURAS PARA MEMÓRIA

(a) Harvard architecture

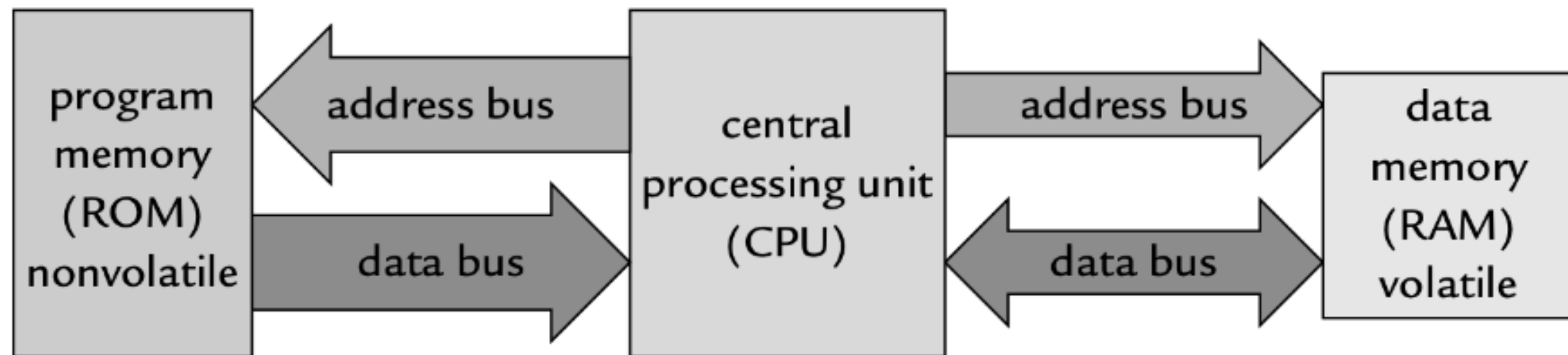


(b) von Neumann architecture



ARQUITETURAS PARA MEMÓRIA

(a) Harvard architecture



PICs, Intel 8051, ARM9

Permite leitura simultânea de dados e do programa

Permite barramentos de dados e programa separadamente otimizados

Guarda dados constantes (look-up tables etc.) na memória de programa, o que requer instruções específicas

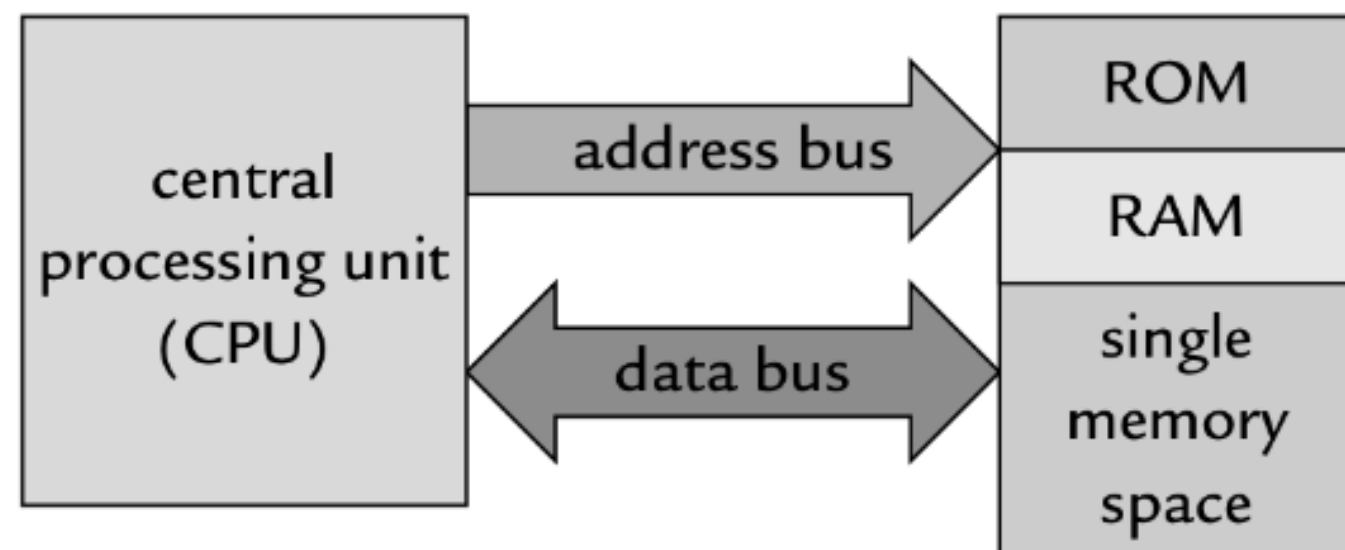
ARQUITETURAS PARA MEMÓRIA

MSP430, Freescale HCS08, ARM7

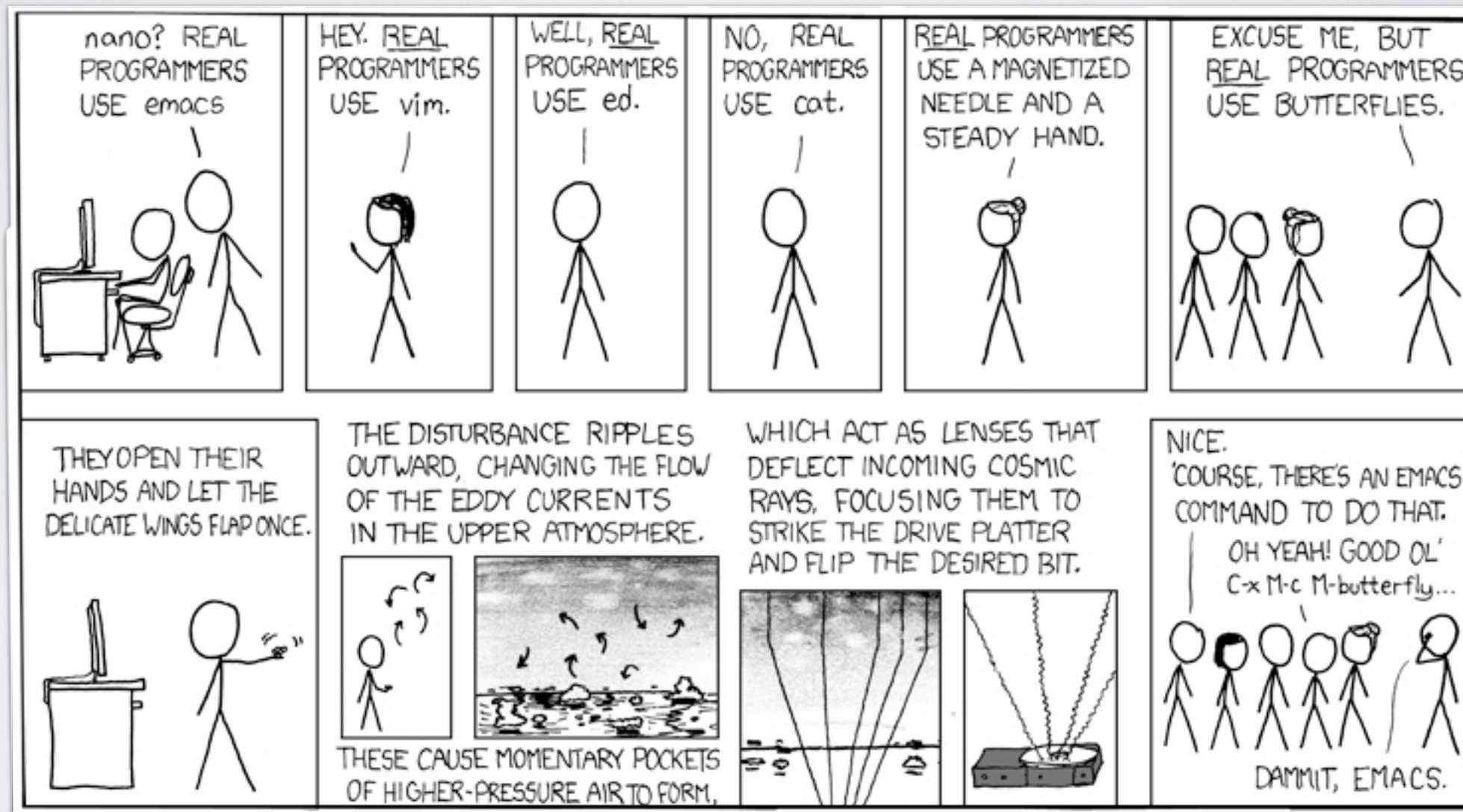
Também conhecida como arquitetura Princeton

Mais simples que a arquitetura Harvard
Menos eficiente

(b) von Neumann architecture



SOFTWARE



SOFTWARE

Microcontroladores não possuem sistema operacional

CPU passa boa parte do tempo interagindo com periféricos

SOFTWARE

Linguagens de programação disponíveis:

Linguagem de máquina: opcodes

(0001110101010111010011010101 - obsoleto!)

SOFTWARE

Linguagens de programação disponíveis:

Assembly: opcodes traduzidos para o inglês

Depende do processador

Algumas operações não existem em C
(rotação bit-a-bit, por exemplo)

SOFTWARE

Linguagens de programação disponíveis:

C: hoje em dia, a compilação para Assembly é bastante eficiente

Permite o uso de funções, estruturas, conversão de tipos de variáveis etc.

SOFTWARE

Linguagens de programação disponíveis:

C++: C orientado a objetos

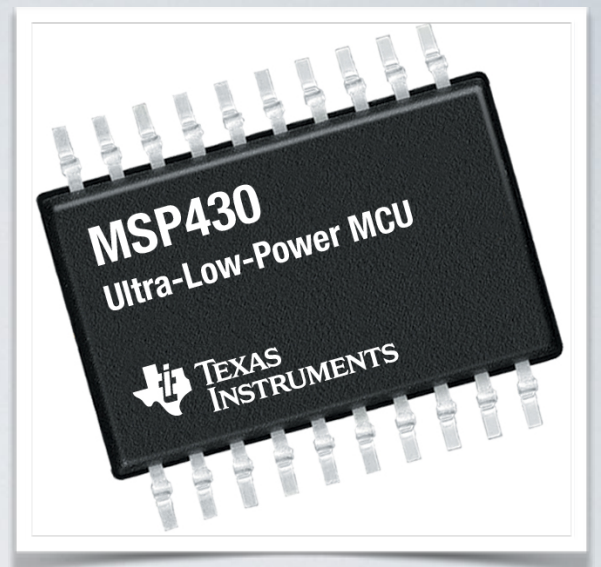
Microcontroladores geralmente trabalham com uma versão simplificada, que mesmo assim pode gerar código ineficiente (Embedded C++)

SOFTWARE

Linguagens de programação disponíveis:

BASIC: disponível para alguns processadores

MSP430



Processador de 16 bits

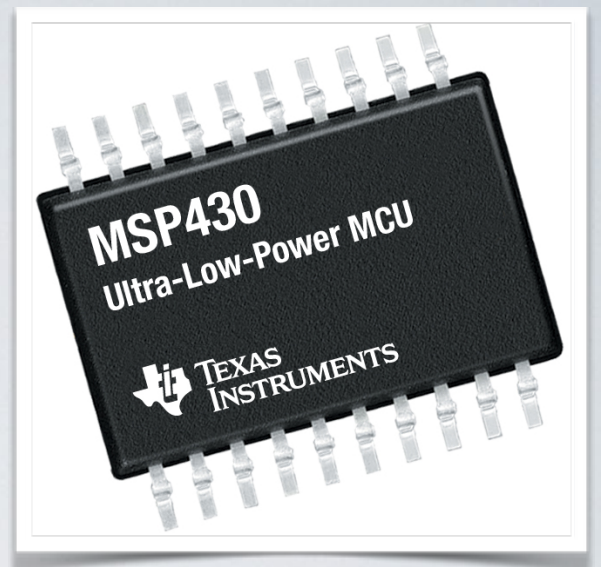
Arquitetura von Neumann

Aplicações de baixo consumo de energia

Barramento de dados e de memória de 16 bits

Registradores de 16 bits

MSP430

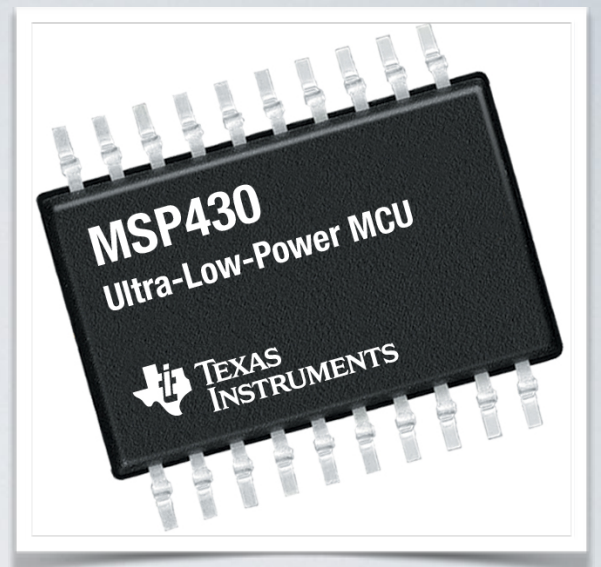


Registradores podem ser usados para variáveis locais, parâmetros para sub-rotinas, endereços etc.

Processador do MSP430 permite aritmética diretamente com valores da memória

CPU pequena e eficiente, muitos registradores

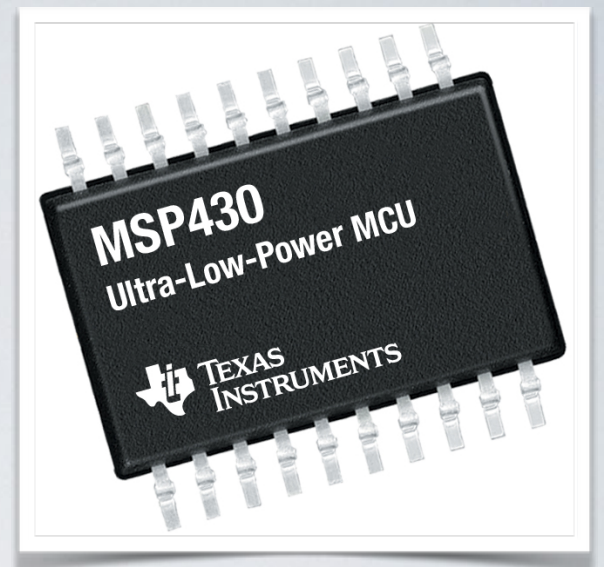
MSP430



Vários modos de baixo consumo (liga/desliga periféricos, retorno rápido/devagar ao modo de consumo completo)

Facilmente colocado em modo de baixo consumo, através de bits no registrador de estado (status register)

MSP430

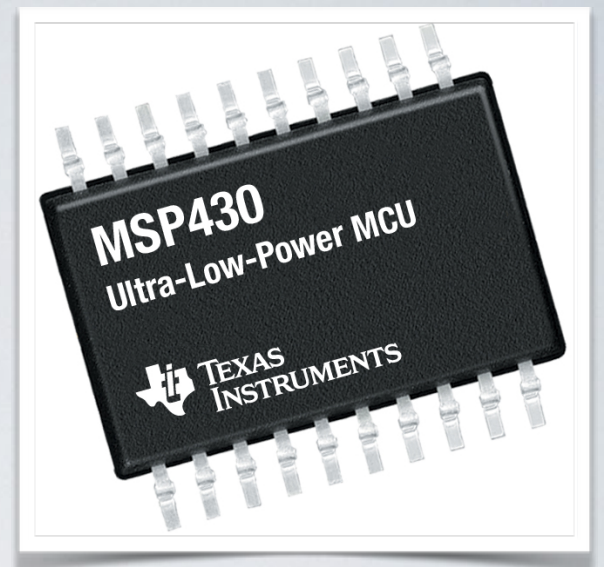


Várias opções de clocks: cristal de 32 KHz, DCO (digitally controlled oscillator), cristal externo etc.

Vários periféricos, sendo que alguns podem funcionar sem a CPU

Alguns MSP430 possuem hardware analógico para aplicações específicas

MSP430



A letra após o nome MSP430 indica o tipo de memória: F para flash, C para ROM

A segunda letra indica MSPs para aplicações específicas: E para eletricidade, W para água e G para sinais que requerem um estágio de ganho

MSP430



MSP430x1xx: amplo conjunto

MSP430F2xx: CPU pode trabalhar em 16MHz (o dobro dos anteriores), não necessita de cristal para o clock de baixa frequência, oferece resistores de pull-up e pull-down, oferece várias entradas analógicas

MSP430x3xx: obsoleto