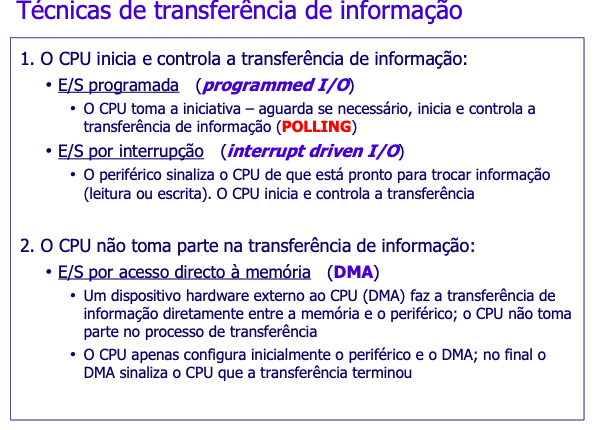
**AC2-TEORICO**

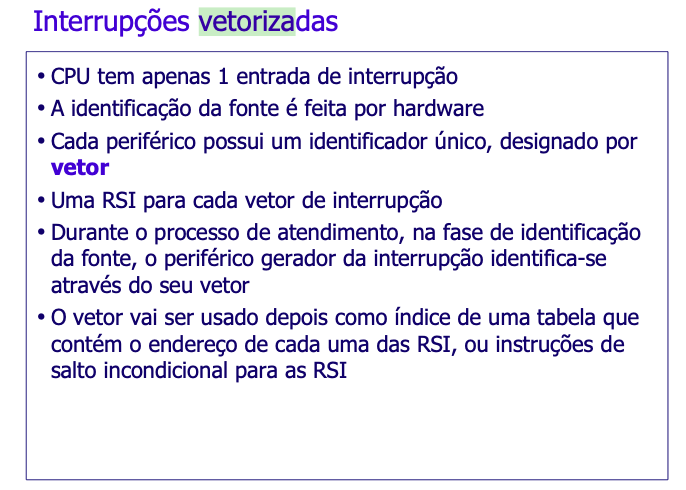
Técnicas de transferência de informação entre periféricos e memoria:

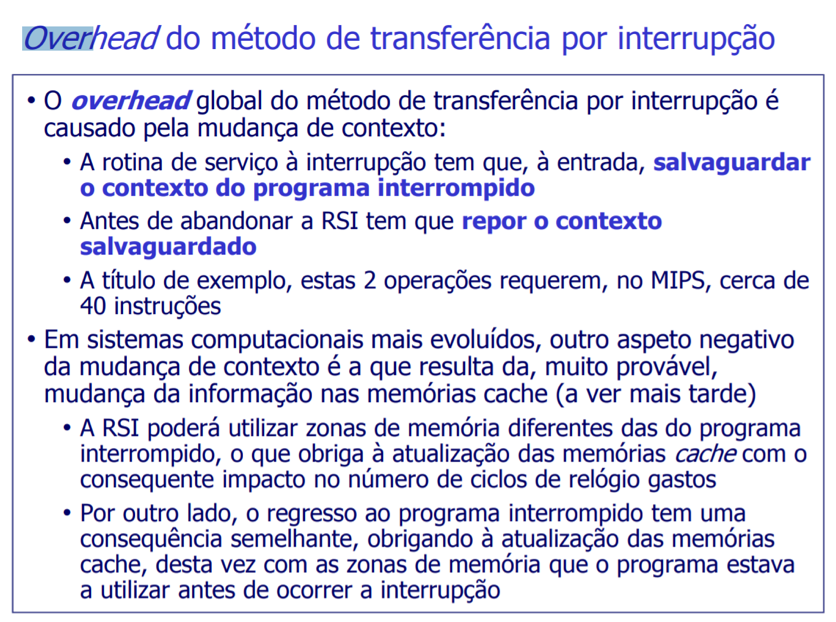
* E/S programada( programed i/o)
* E/S por interrupção ( interrupt driven i/o)
* E/S por acesso direto a memoria( DMA)

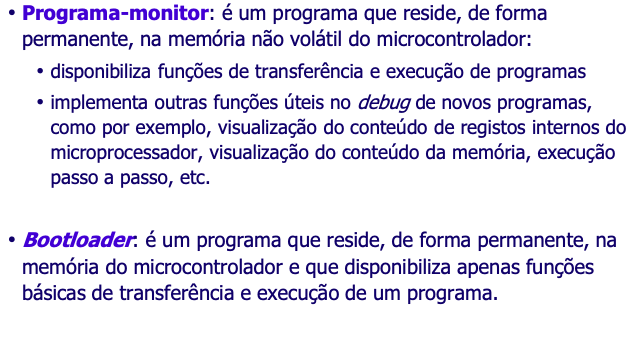
**Prologo**: Salvaguarda do contexto do programa que foi interrompido.

**Epilogo**: reposição do contexto do programa que foi interrompido.







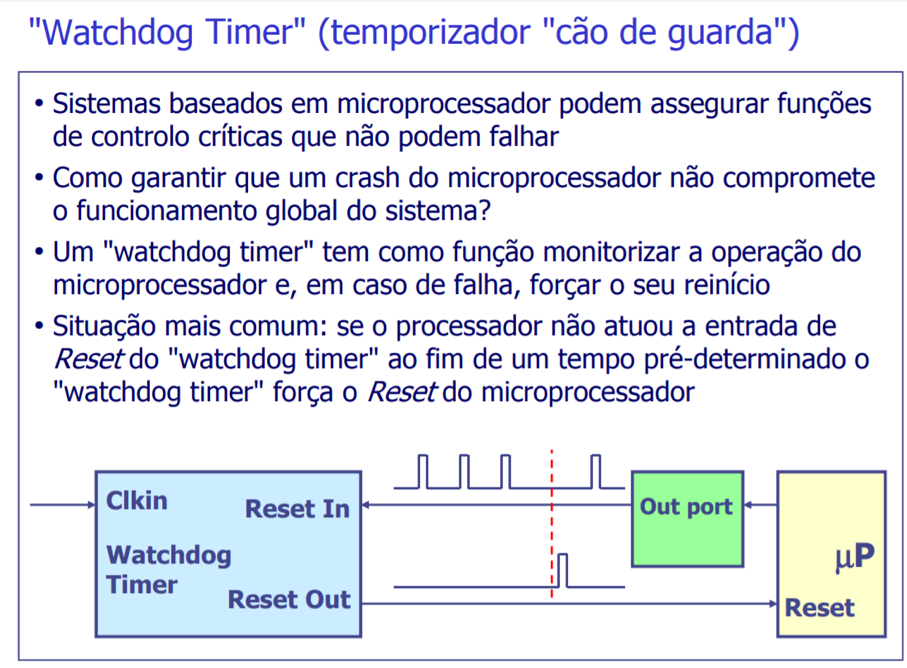
**Microcrontoladores-processadores**

Uma imagem com texto

Descrição gerada automaticamente

**Timers**

Timers:



**Sistema Computacional**

Byte-adressable : uma word de 32 bits é armazenada em 4 posicoes de memoria consecutivas de 1 byte.

**DMA**

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

**Ciclos com transferências múltiplas:**

Uma imagem com texto

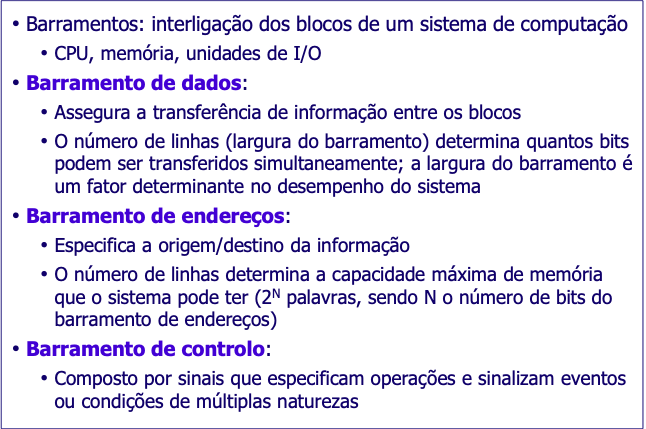
Descrição gerada automaticamente

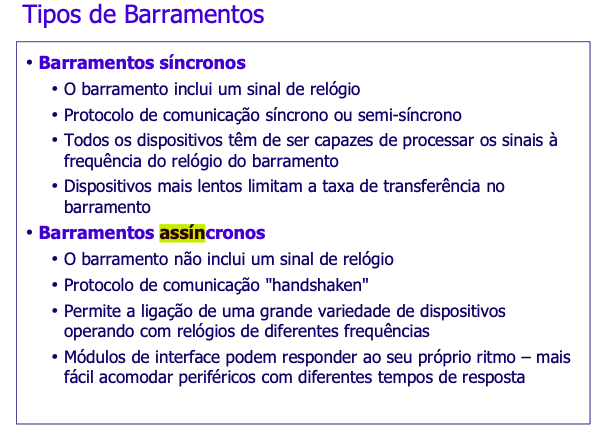
**Organização de Barramentos**

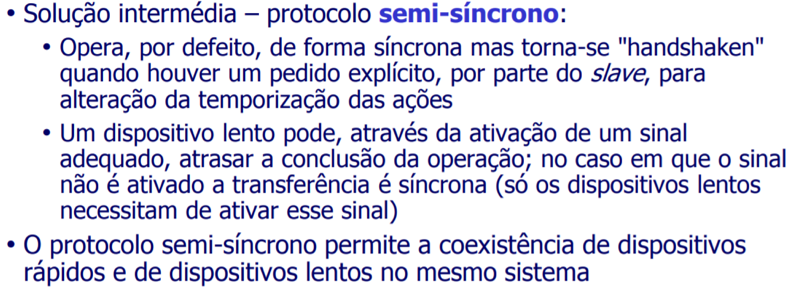
Uma imagem com mesa

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamente

****





Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

**Barramento PCI( peripheral component interconnect)**

* Barramento paralelo
* Barramento “multi-master” com arbitragem centralizada
* Permite que dispositivos ligados ao barramento sejam automaticamente detetados e configurados
* O aumento da fraquencia de relogio não é solução
* Incapaz de suportar transferências com largura de banda e latência garantidos.

**Interface SPI**

Descricao geral:

* Arquitetura "Master-Slave"  
  • O sistema só pode ter um master  
  • O master é o único dispositivo no sistema que pode controlar o relógio
  + Um master pode estar ligado a vários slaves: para cada comunicação, apenas 1 slave é selecionado pelo master
  + Comunicação bidirecional "full-duplex"

**Barramento CAN**

Descricao geral:

• O CAN é um barramento "multi-master": qualquer nó do barramento pode produzir informação e iniciar uma transmissão

• cada tipo de mensagem tem um ID unico

• Uma vez que dois nós podem querer aceder simultaneamente ao barramento para transmitir, tem que haver uma forma de arbitrar o acesso

• Comunicação bidirecional "half-duplex"

• A informação produzida é encapsulada em tramas

“bit-stuffing”:

• Por cada 5 bits iguais é inserido 1 de polaridade oposta

• Garante tempo máximo entre transições da linha de dados, assegurando que há transições suficientes para manter os relógios dos nós sincronizados.

**Interface RS-232**

O desvio da frequência entre os sinais de relogio do transmissor e recetor podem originar um erro. Esse erro cresce de forma diretamente proporcional ao comprimento da trama.

**Interface I2C**

Descrição geral:

• Transferência série bidirecional, half-duplex, orientada ao byte

• As transferências envolvem sempre uma relação master/slave

• Master pode ser transmissor ou recetor ("master-transmitter" ou "master-receiver")

• Cada dispositivo ligado ao barramento é endereçável por software usando um endereço único previamente atribuído

• Endereços de 7 bits ("standard mode") - alguns endereços reservados, 112 disponíveis.

* + O primeiro byte de uma comunicação contem a informação do endereço do slave e da operação a realizar.

**USB**

Descricao geral:

• "Hot plugging": Possibilidade de ligar e desligar fisicamente dispositivos sem necessidade de desligar a alimentação do sistema ou efetuar a sua reinicialização

• Os dados são transmitidos em modo diferencial em par entrançado (nível lógico na receção é discriminado pela diferença de tensão entre os dois sinais)

• O USB usa comunicação síncrona orientada ao bit. Um conjunto consecutivo de bits é organizado numa mensagem (packet)

• Os bits são transmitidos continuamente com um ritmo imposto por um relógio que segue para o recetor embebido nos dados (relógio codificado - uma das tarefas do recetor é a recuperação desse relógio)

Transferencia de controlo:

* + Informação a receber/transmitir é pequena

Transferencia “bulk”:

• Transferência de volumes de dados elevados, com instantes de transmissão irregulares

• Permitem a transmissão de dados com garantias de entrega, usando mecanismos de retransmissão em caso de deteção de uma situação de erro (CRC – Cyclic Redundancy Check)

Transferencia isócrona:

• Permitem a transferência de informação com uma largura de banda garantida (quantidade de dados / unidade de tempo) e latência limitada. Ou seja, é assegurado um ritmo de transmissão sustentado.

**Memoria do tipo RAM**

SRAM – Static RAM:

• Vantagens: • Rápida

• Informação permanece até que a alimentação seja cortada

• Inconvenientes:  
• Implementações típicas: 6 transistores / célula  
• Baixa densidade, elevada dissipação de potência

• Custo/bit elevado

DRAM – Dynamic RAM :

• Vantagens:  
• Implementações típicas: (1 transistor + 1 condensador) / célula • Alta densidade, baixa dissipação de potência  
• Custo/bit baixo

• Inconvenientes:  
• Informação permanece apenas durante alguns mili-segundos (necessita

de refresh regular – daí a designação "dynamic")  
• Mais lenta (pelo menos 1 ordem de grandeza) que a SRAM

**Device Drivers**

• Um programa que permite a outro programa (aplicação, SO) interagir

com um dado dispositivo de hardware

• Implementa a camada de abstração e lida com as particularidades do dispositivo controlado

• Como o Device Driver tem de lidar com os aspetos específicos da implementação física, o seu fornecimento é sempre assegurado pelo fabricante

• Abstração, uniformização de acesso, independência entre aplicações/SO

e o hardware.

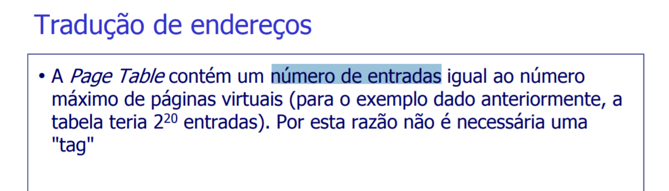
**Memoria Virtual**

Uma imagem com texto

Descrição gerada automaticamentese ativo , reside em memoria física

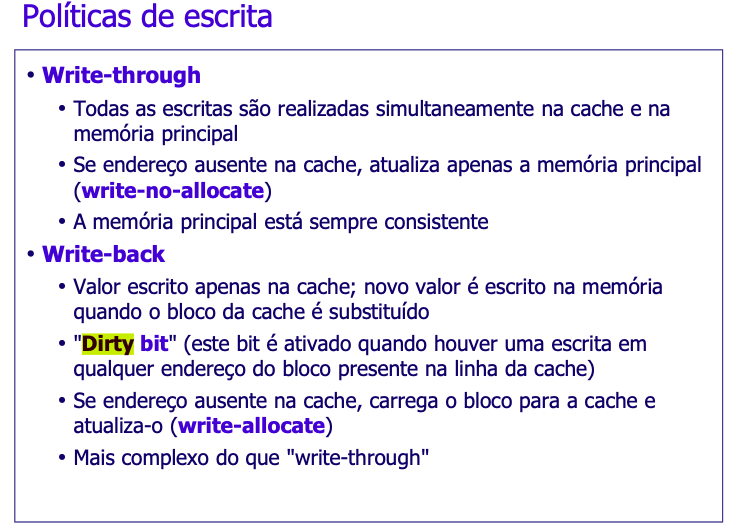
Uma imagem com texto

Descrição gerada automaticamente



**Cache**

Tacess = hitRatio \* hitTime + (1-hitRatio)\*(hitTime+tMem) onde tMem é o que esta na memoria principal.



3 formas de organizar os sistemas de cache:

Cache totalmente associativa ("fully associative") = no de linhas  
Cache com mapeamento direto ("direct mapped") - 1 comaparador ou linha Cache parcialmente associativa ("set associative") = associatividade

