

ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES DISPOSITIVOS DE ENTRADA E SAÍDA

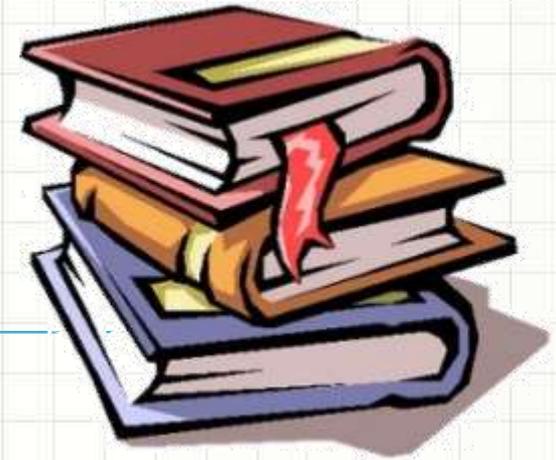
Prof. Dr. Daniel Caetano

2011 - 2

Visão Geral

- 1 • O que é um Dispositivo
- 2 • Ligando o Dispositivo de E/S
- 3 • E/S por Polling
- 4 • E/S por Interrupção
- 5 • E/S por DMA
- 6 • Dispositivos Comuns de E/S

Material de Estudo



Material

Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/aulas/aoc/>
(Aula 7)

Apresentação

<http://www.caetano.eng.br/aulas/aoc/>
(Aula 7)

Material Didático

Arquitetura e Organização de Computadores, páginas
191 a 238

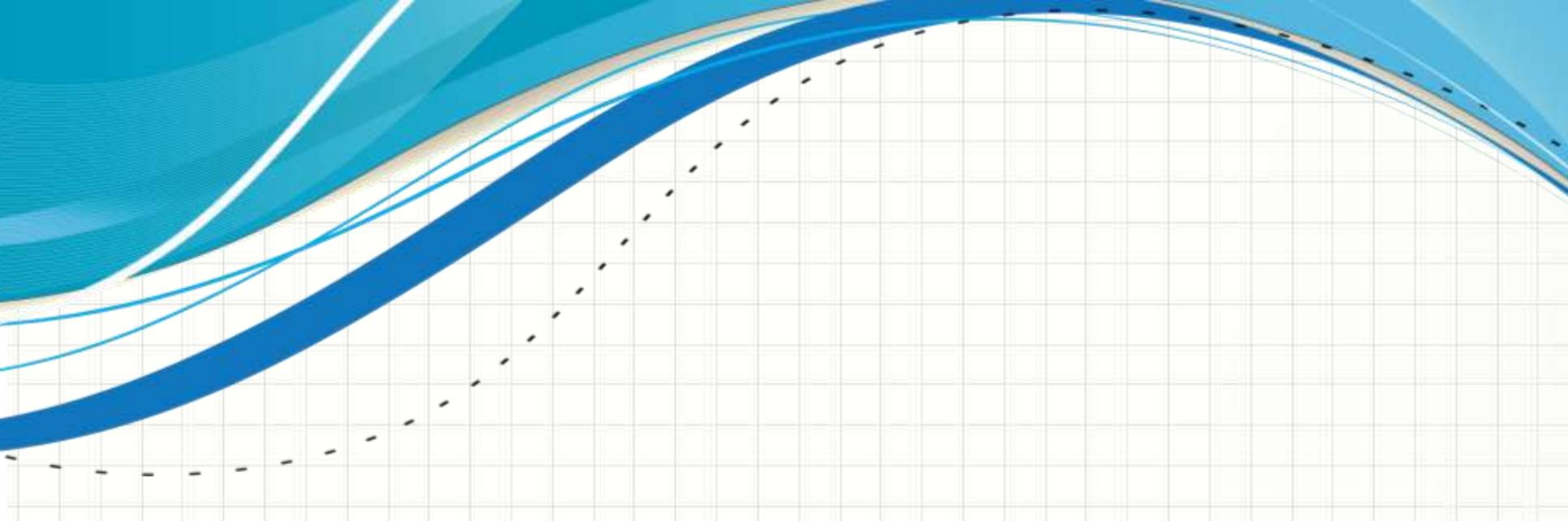
Biblioteca Virtual

-

Lembretes

- **AV1!**

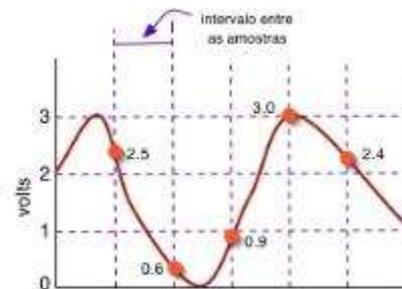




O QUE É UM DISPOSITIVO?

O que é um dispositivo?

- Simples: converte informações \leftrightarrow sinais
- Dispositivos de Entrada
 - Converte informações do mundo em sinais elétricos (números)

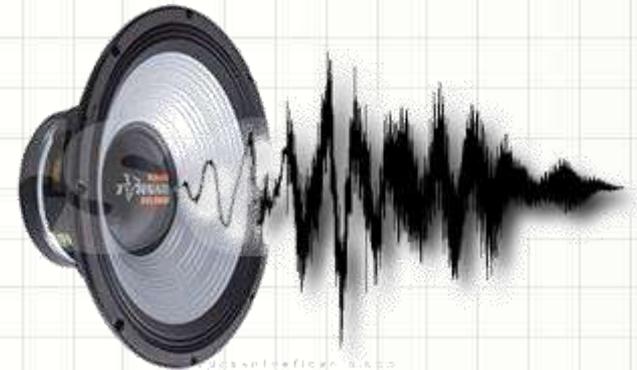
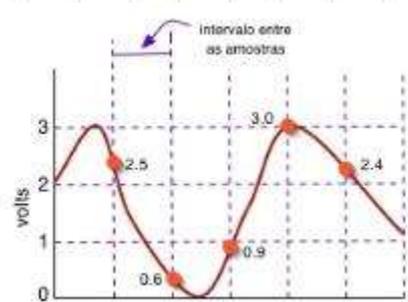


10011001
10011110
10101100
10111001
11001010
11001111
11010011
10111101

O que é um dispositivo?

- Dispositivos de Saída
 - Converte sinais (números) em informações do mundo

10011001
10011110
10101100
10111001
11001010
11001111
11010011
10111101



O que é um dispositivo?

- Usualmente: pensam que são únicos
 - Não entendem endereços
 - Possuem apenas um “pino” de ativação
 - O tamanho da palavra é diferente da CPU
- CPU: apenas um “pino” de dispositivo: IORQ
 - Input/Output ReQuest

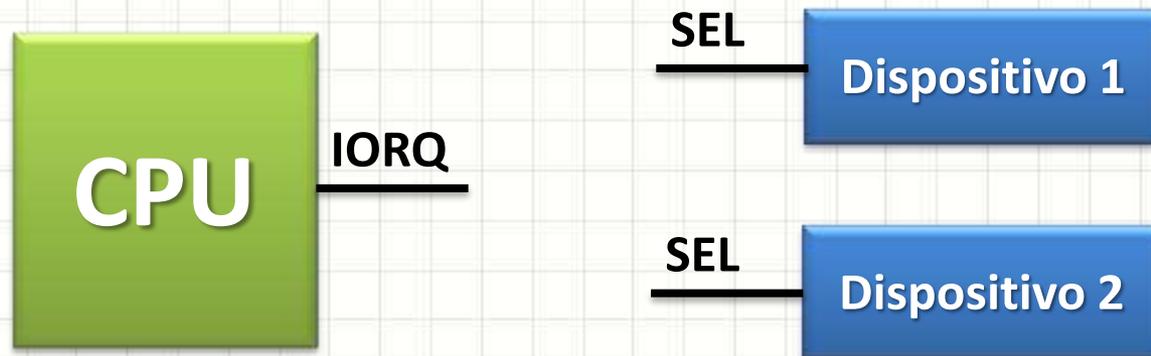


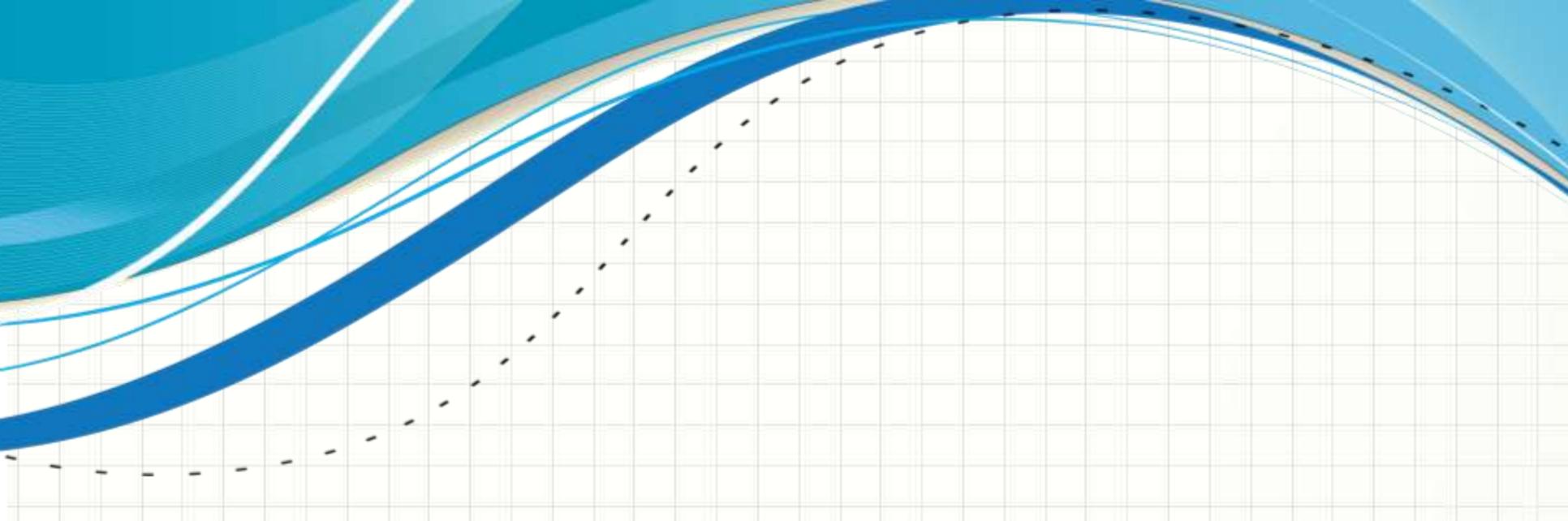
O que é um dispositivo?

- Usualmente: pensam que são únicos

Como Resolver?

- CPU: apenas um pino de dispositivo: IORQ
 - Input/Output ReQuest





DECODIFICADOR DE ENDEREÇOS

Decodificador de Endereços

- Memória: é acionada por MREQ, A_0-A_n , R/W
- E/S: IORQ, R/W... e A_0-A_n
 - Como assim?



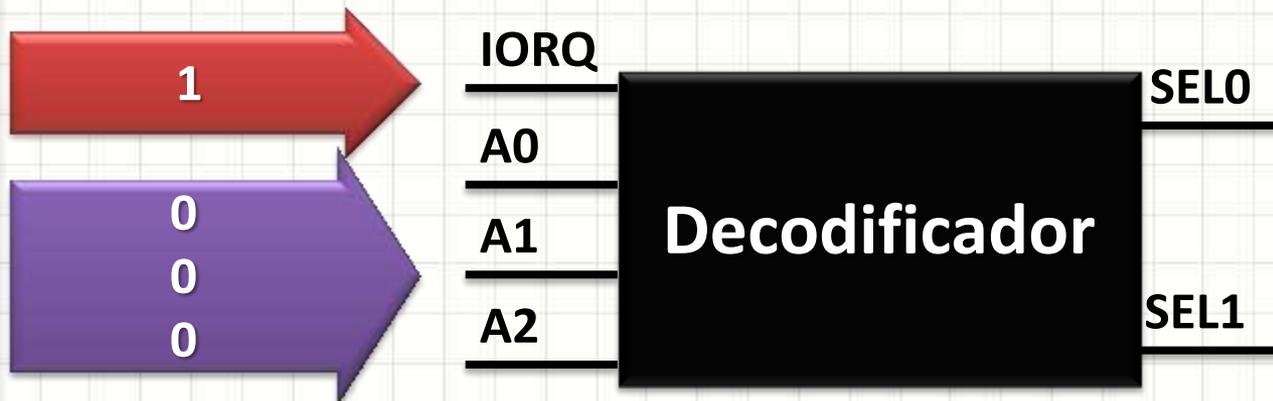
Decodificador de Endereços

- Como funciona o decodificador?



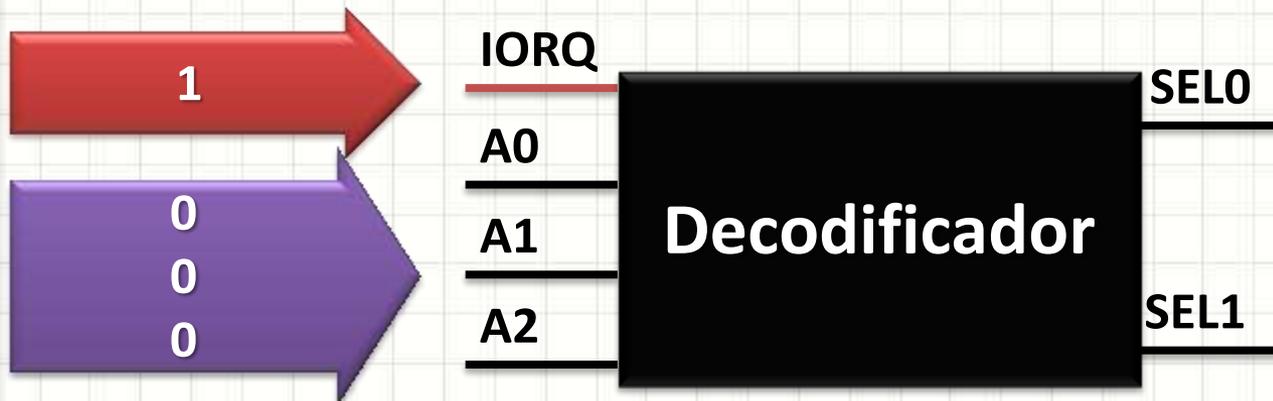
Decodificador de Endereços

- Como funciona o decodificador?
- Exemplo: ativar dispositivo no “endereço” 0



Decodificador de Endereços

- Como funciona o decodificador?
- Exemplo: ativar dispositivo no “endereço” 0



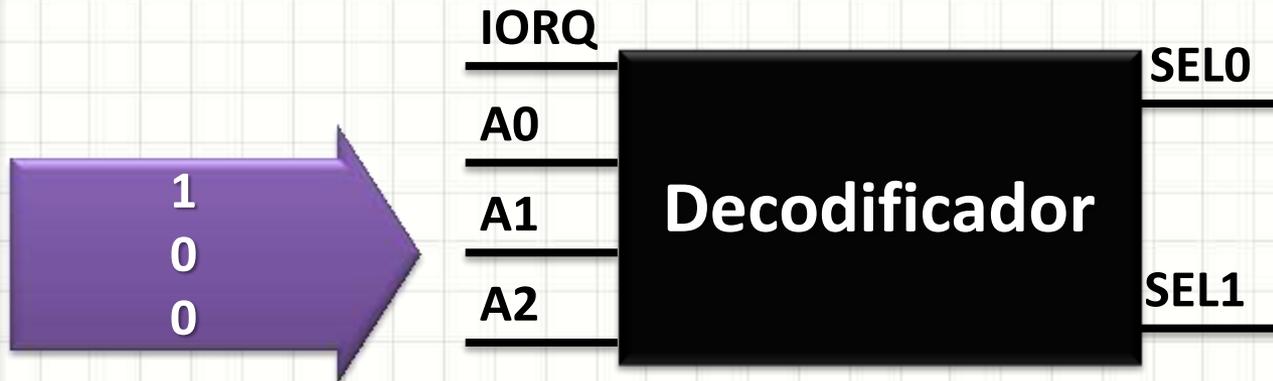
Decodificador de Endereços

- Como funciona o decodificador?
- Exemplo: ativar dispositivo no “endereço” 0



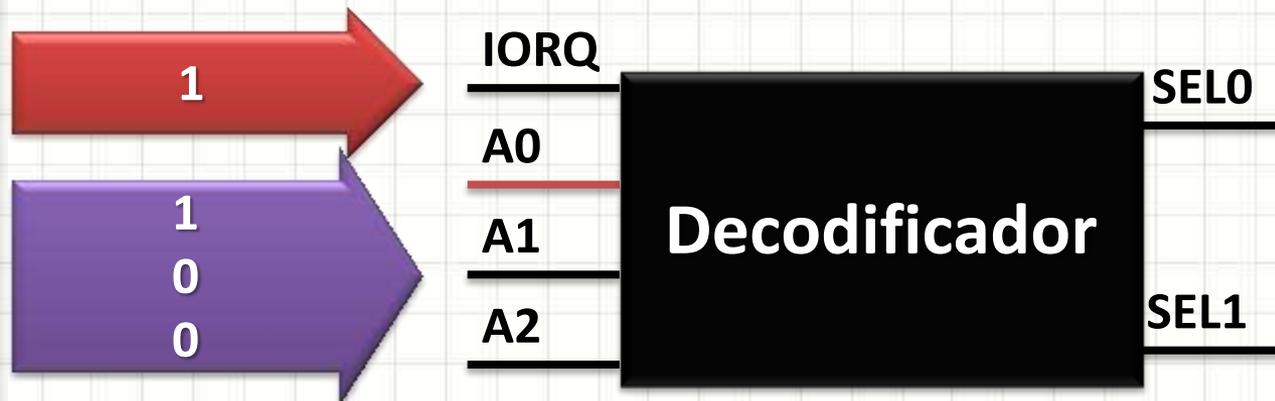
Decodificador de Endereços

- Como funciona o decodificador?
- Exemplo: ativar dispositivo no “endereço” 1



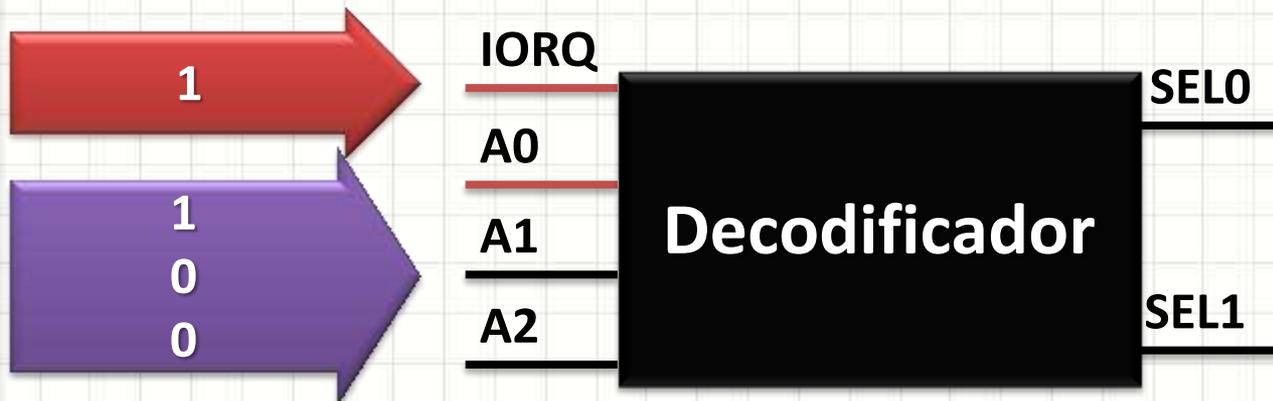
Decodificador de Endereços

- Como funciona o decodificador?
- Exemplo: ativar dispositivo no “endereço” 1



Decodificador de Endereços

- Como funciona o decodificador?
- Exemplo: ativar dispositivo no “endereço” 1



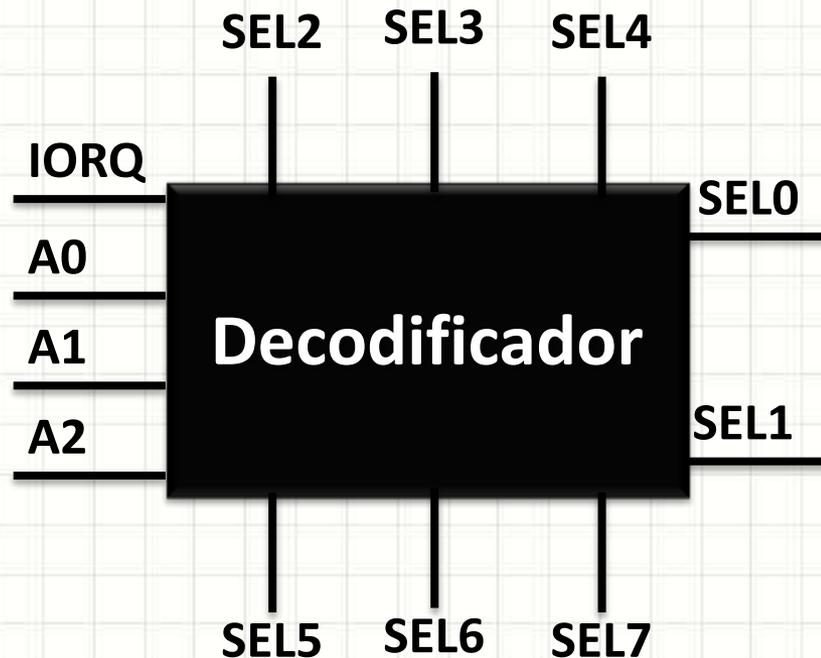
Decodificador de Endereços

- Como funciona o decodificador?
- Exemplo: ativar dispositivo no “endereço” 1



Decodificador de Endereços

- Quantos dispositivos esse decodificador suporta?



Decodificador de Endereços

- Q
SU

**3 bits =
8 dispositivos**



Decodificador de Endereços

- Decodificador: conjunto de circuitos lógicos
 - Dispositivo é ativado para endereço específico
- Cada dispositivo é, usualmente, “encaixado” em um único endereço
- “Endereço” de dispositivo: **porto** ou **porta**
- Dispositivos simples:
 - barramento de dados: direto
 - barramento de controle: direto
- E nos dispositivos mais complexos?

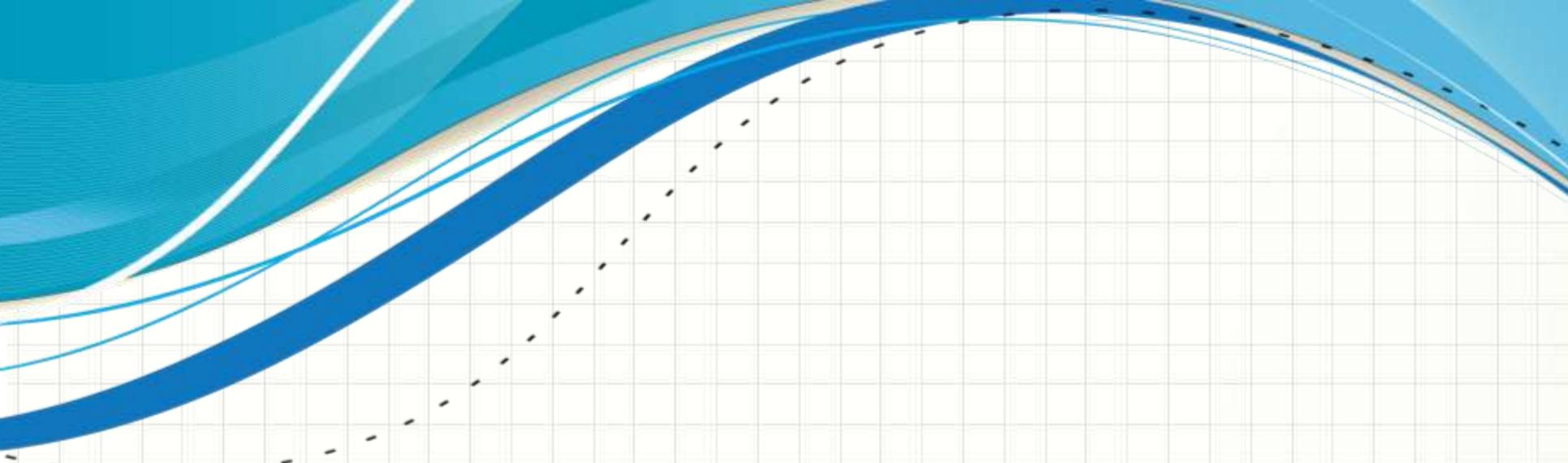
Controlador de Dispositivos

- Dispositivos mais complexos
 - Exigem mais que um decodificador
 - Exemplo: harddisk
- Existe uma **placa controladora** ou **interface controladora de dispositivo**



Controlador de Dispositivos

- Controladores
 - Podem integrar vários dispositivos
 - Permitem interações entre dispositivos e memória
 - Podem estar integrados aos dispositivos (!)
- Como a CPU fica nessa história?
 - Existem três formas de controlar dispositivos
 - *Polling* (E/S Programada)
 - Interrupções
 - DMA (*Direct Memory Access*)



E/S POR *POLLING*

E/S por *Polling*

- *Polling* ou E/S Programada
 - *Poll*: consulta
 - A CPU precisa verificar constantemente os dispositivos
- Exemplo:
 - Placa de rede recebeu dado
 - CPU precisa lê-lo antes que o próximo chegue!
 - Impressora imprimiu uma letra
 - CPU precisa fornecer a próxima

E/S por *Polling*

- Como a CPU sabe que chegou a hora...
 - De ler um novo dado
 - De escrever um novo dado?
- Precisa **consultar** o dispositivo: *polling*
- A consulta é cíclica, de tempos em tempos
 - Dispositivo 1: precisa de algo?
 - Dispositivo 2: precisa de algo?
 - ...
 - Dispositivo n: precisa de algo?
 - Dispositivo 1: precisa de algo?
 - ...

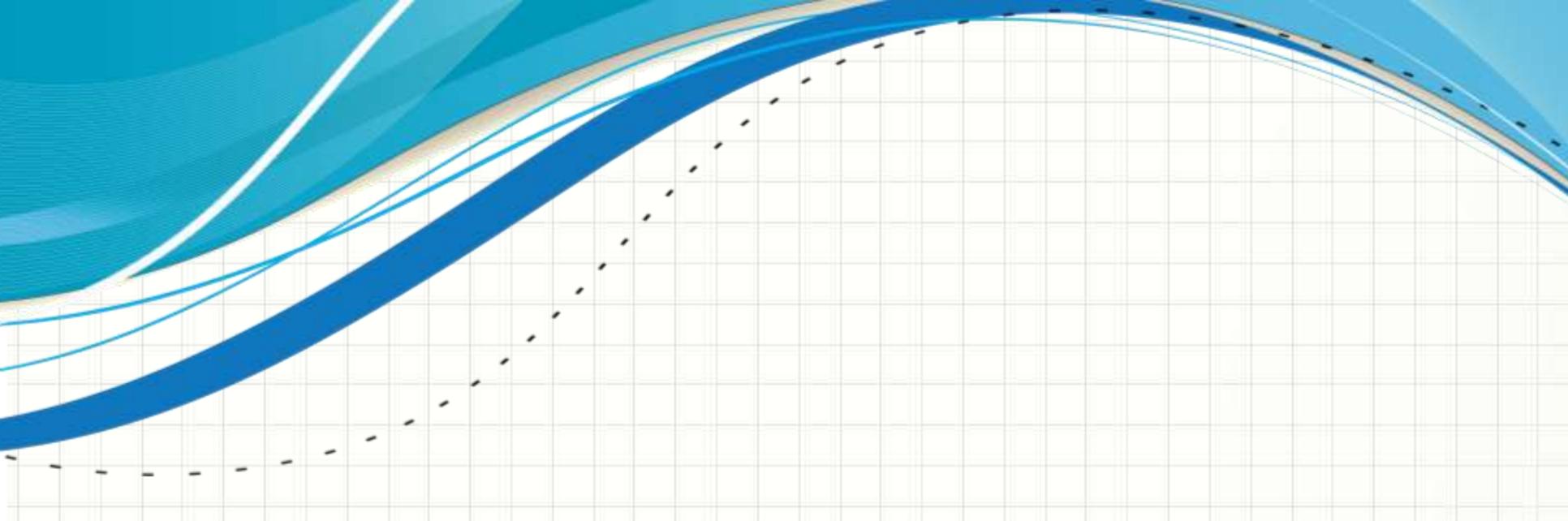
E/S por *Polling*

- Vantagens:
 - Controle total da CPU
 - CPU escolhe quando consulta o quê
- Problemas:
 - Tempo de CPU gasto com a consulta
 - Gasto de CPU com transferências $\text{Disp.} \leftrightarrow \text{Mem}$
 - Perda de dados, se a CPU não atender rapidamente um dispositivo (ex.: rede)

E/S por *Polling*

- Analogia:
 - CPU: Cozinheiro e Garçom de um restaurante
 - Dispositivos: Clientes
- Se o cozinheiro é também o garçom
 - Para de cozinhar para atender os clientes
 - Para de cozinhar para levar pratos para os clientes
 - Pode perder clientes se demorar demais para atendê-los





E/S POR INTERRUPÇÃO

E/S por Interrupção

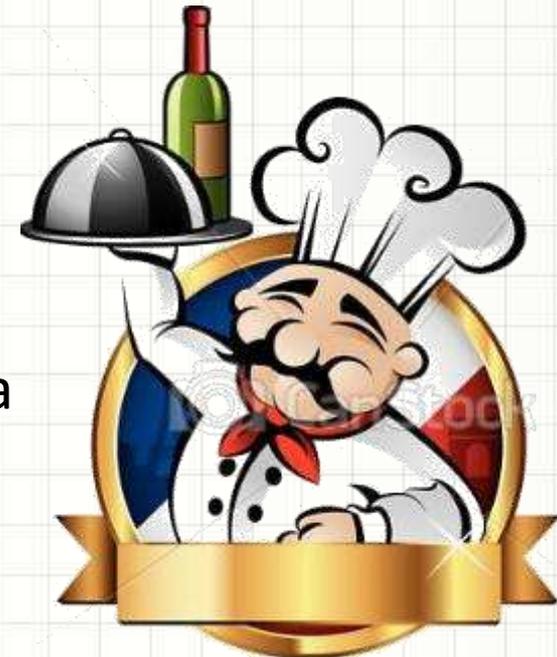
- Interrupção: Dispositivo **interrompe** CPU
 - Sempre que dispositivo precisa de algo
 - CPU não precisa fazer consulta
- A Interrupção é um “pino” da CPU (IRQ)
- Sinal da IRQ: barramento de controle
- Exemplo:
 - Placa de rede recebeu dado
 - Placa de rede dispara interrupção da CPU
 - Acabaram os dados da placa de som
 - Placa de som dispara interrupção da CPU

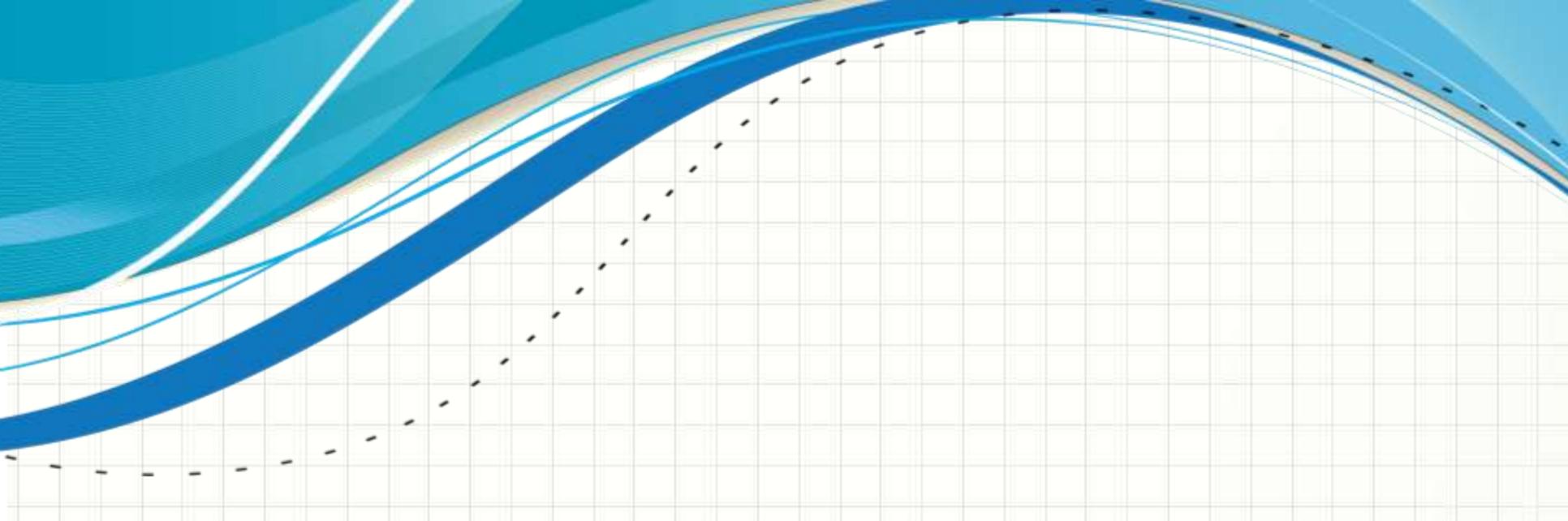
E/S por Interrupção

- CPUs modernas costumam ter várias IRQs
 - PIC: Programmable Interrupt Controller (16)
 - APIC: Advanced PIC (256)
- Vantagens:
 - CPU não precisa consultar dispositivos
- Problemas:
 - Perda (parcial) de controle pela CPU (EI/DI)
 - Gasto de CPU com transferências Disp.↔Mem

E/S por Interrupção

- Analogia:
 - CPU: Cozinheiro e Garçom de um restaurante
 - Dispositivos: Clientes **com sineta**
- Com a sineta
 - Para de cozinhar para atender os clientes **só quando ouve a sineta**
 - Para de cozinhar para levar pratos para os clientes
 - Os clientes **tocam a sineta** quanto precisam ser atendidos (não há perda)





E/S POR DMA

E/S por DMA

- *DMA: Direct Memory Access*
 - Acesso Direto à Memória
 - CPU configura o DMA para transferir dados
 - CPU não precisa fazer a transferência
- O DMA é quase um dispositivo em si mesmo
- Exemplo:
 - Placa de rede recebeu um pacote de dados
 - CPU configura DMA para copiar esses dados na RAM
 - Acabaram os dados da placa de som
 - CPU configura DMA para transferir dados da RAM para placa de som

E/S por DMA + *Polling*

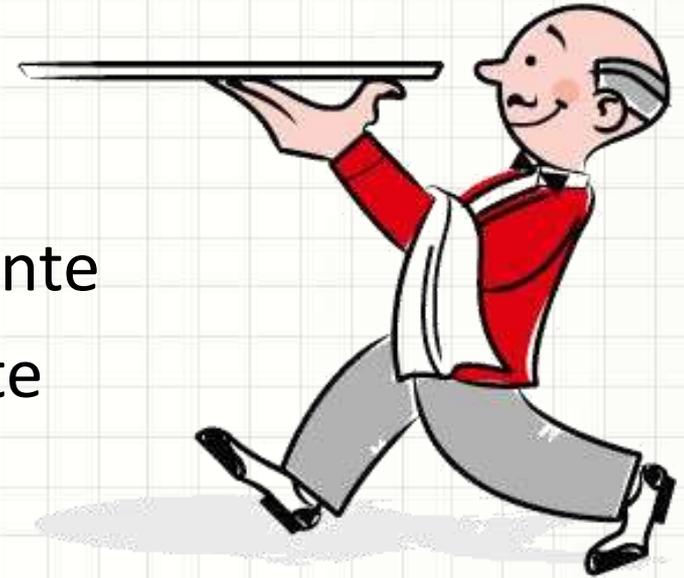
- Vantagens:
 - Controle total da CPU
 - CPU escolhe quando consulta o quê
 - CPU não gasta tempo com transferências
- Problemas:
 - Tempo de CPU gasto com a consulta
 - Perda de dados, se a CPU não atender rapidamente um dispositivo (ex.: rede)

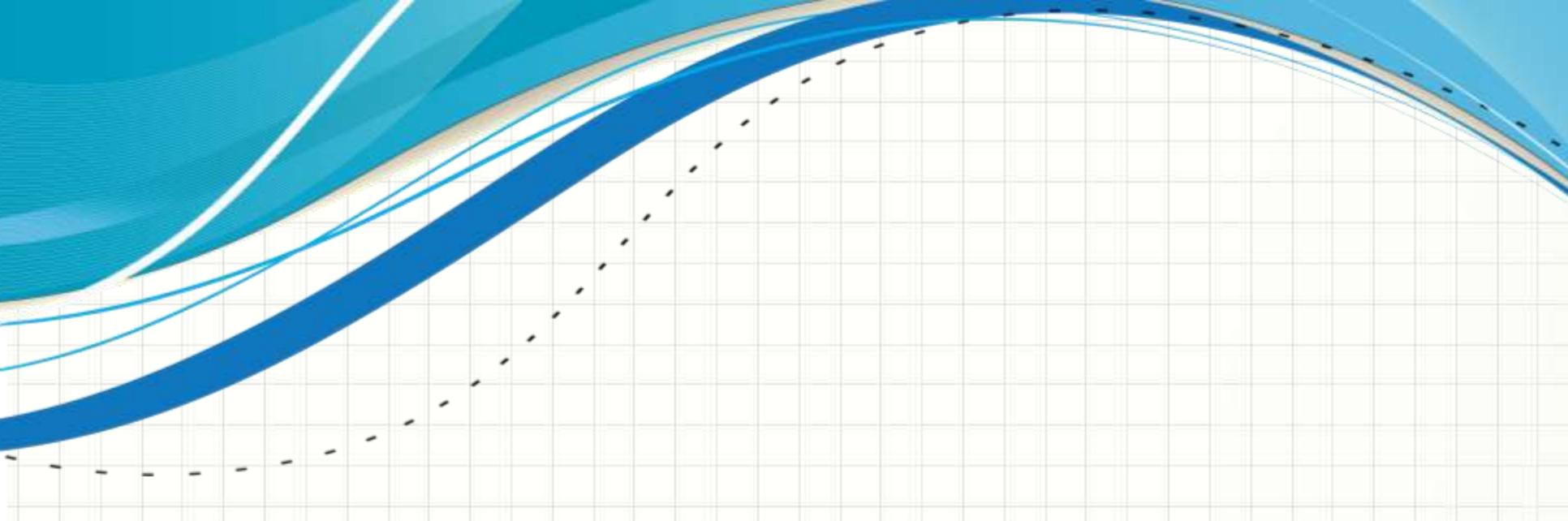
E/S por DMA + Interrupção

- Vantagens:
 - CPU não precisa consultar dispositivos
 - CPU não precisa gastar tempo com transferências
- Problemas:
 - Perda (parcial) de controle pela CPU (EI/DI)
 - Custo do circuito DMA

E/S por DMA

- Analogia:
 - CPU: Cozinheiro do restaurante
 - DMA: Garçon do restaurante
 - Dispositivos: Clientes
- Com o garçom
 - É o garçom que atende os clientes, o cozinheiro **não para** de cozinhar
 - É o garçom que leva os pratos aos clientes

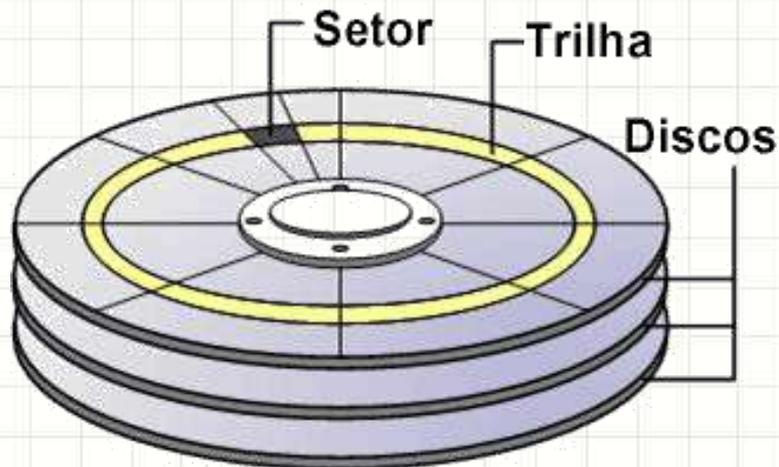




DISPOSITIVOS COMUNS DE E/S

Hard Disk

- Similar aos finados disquetes, mas...
 - Vários discos (de alumínio ou vidro)
 - Cobertos por Óxido de Ferro (magnetizado – 0, 1)
 - Fixos a um eixo, giram de 3.000 a 10.000 rpm
 - Regiões acessíveis: disco (head), setor, trilha (cylinder)
 - Setores: usualmente 512 bytes

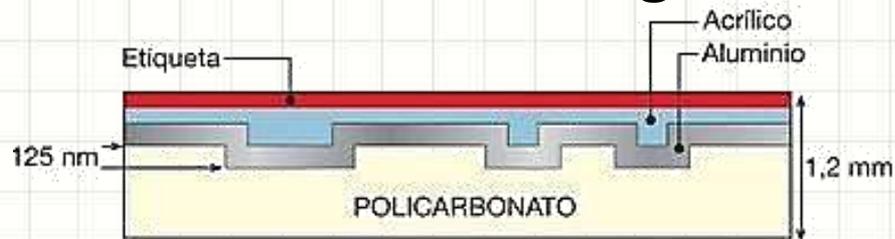


- 6 Cabeças
- 1 por vez
- Leitura: setor?
- Sistema de arquivos

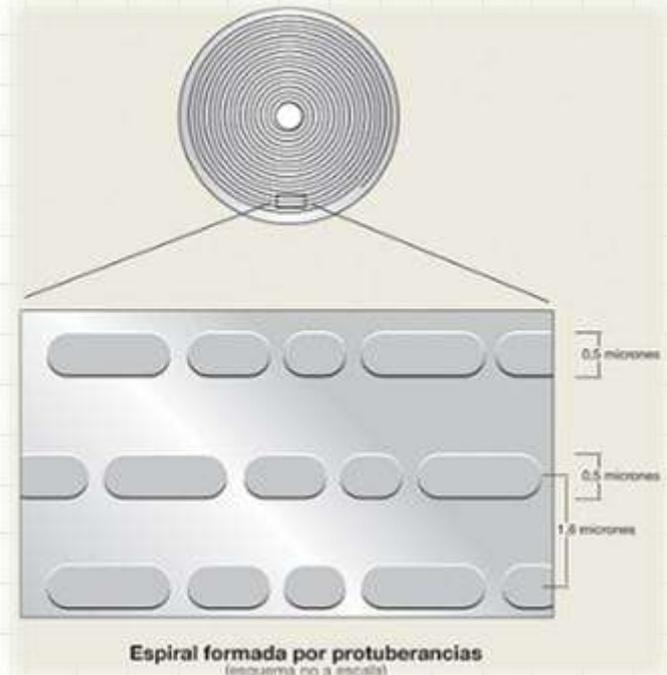
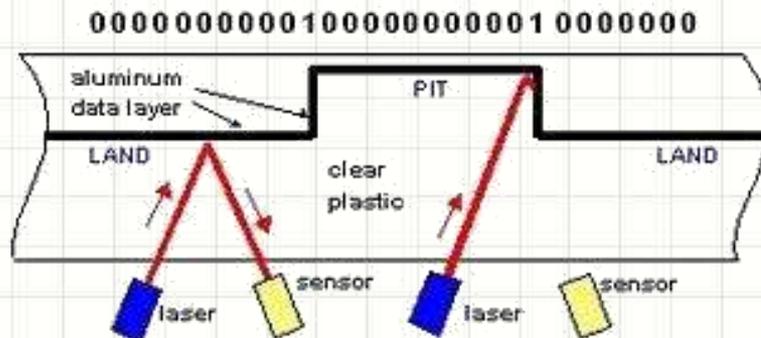
Discos Ópticos

- Discos Plásticos

- Camada de alumínio “entalhada” (0, 1)
- Leitura: cabeça dispara laser e tenta coletá-lo
- Dados armazenados em espiral
- Velocidade de giro: mais lenta em direção à borda



Corte transversal de un CD
(esquema no a escala)

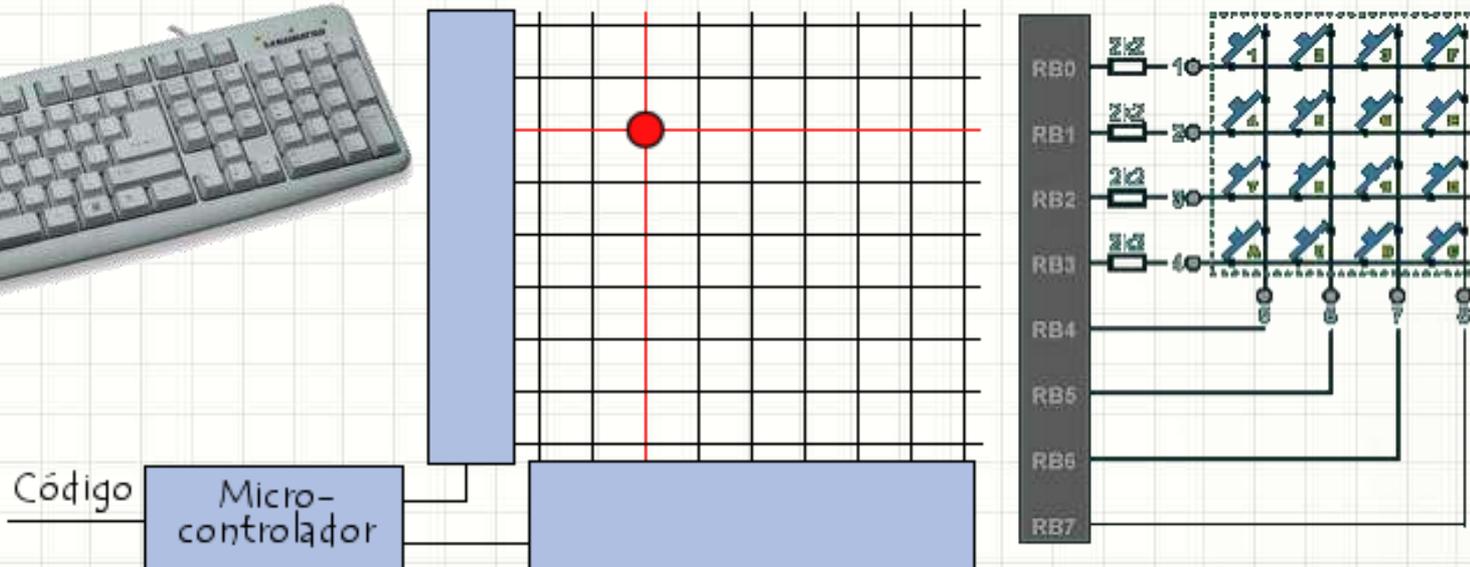


Espiral formada por protuberancias
(esquema no a escala)

Teclados

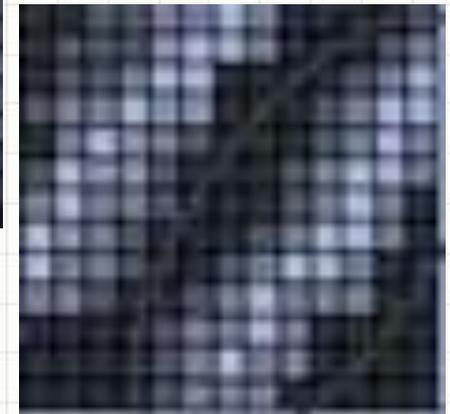
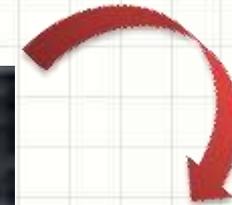
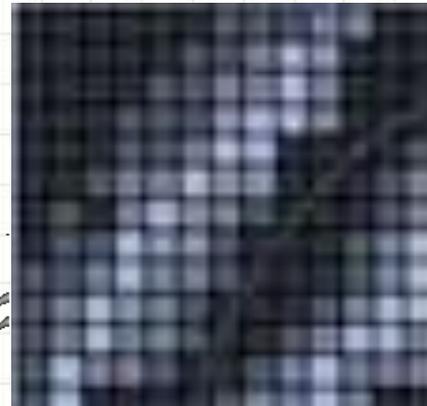
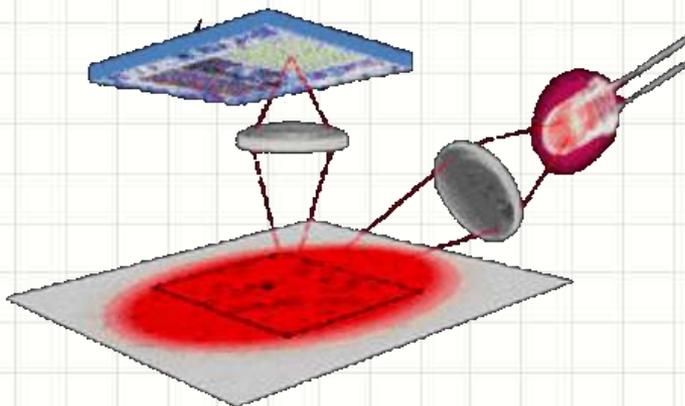
- Matriz Eletrônica

- Simplificado: A → Linha 04, coluna 02 → 0402
- Teclas modificadoras enviadas separadamente
- Mapa de Teclado: Linha/Coluna → Código
- US-International, ABNT-2 etc.



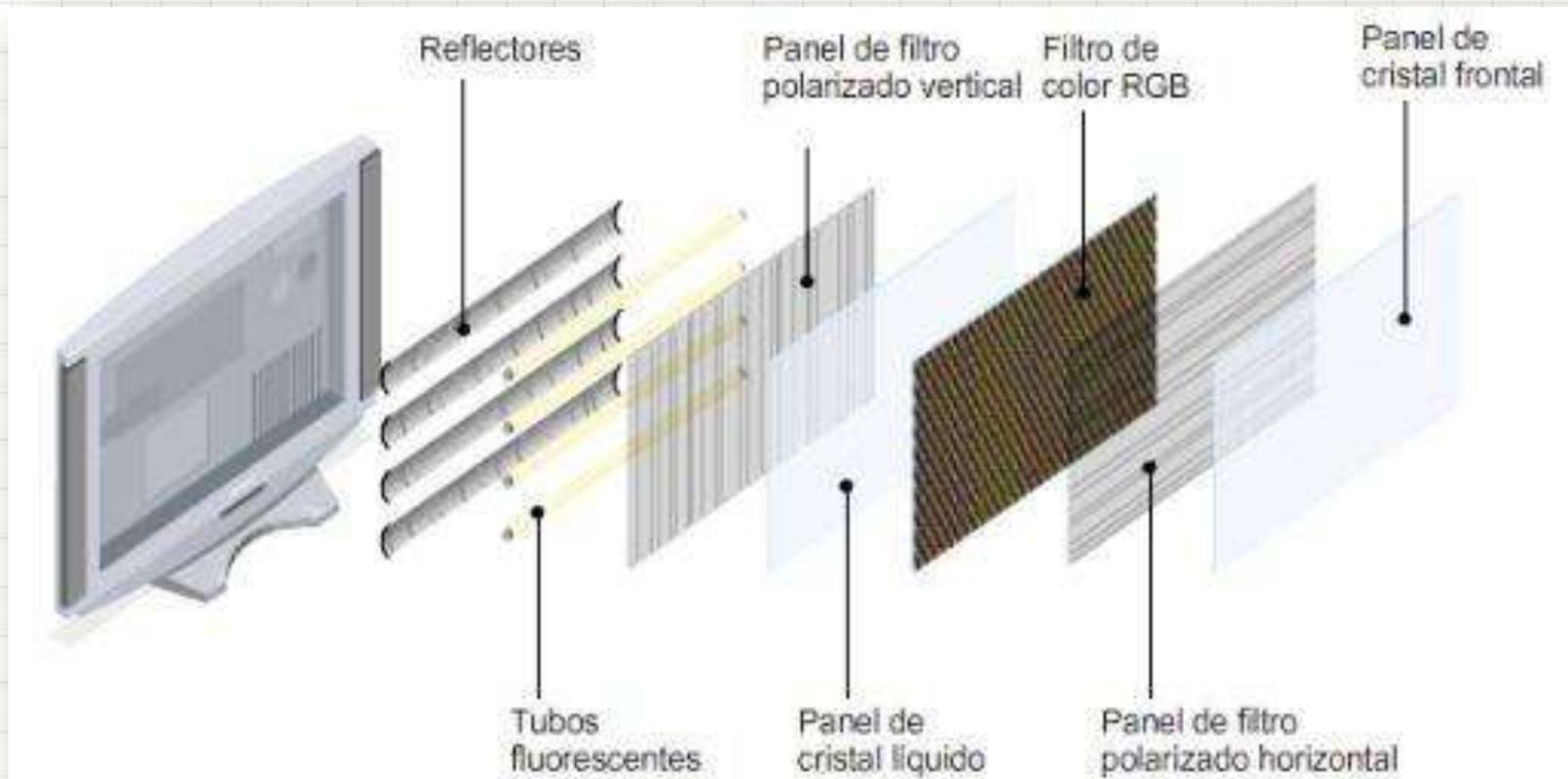
Mouse Óptico

- Fotos da Base de Apoio
 - Emite luz e digitaliza imagem: intervalos regulares
 - Detecção de deslocamento entre imagens
 - Codificação em um valor para o mouse



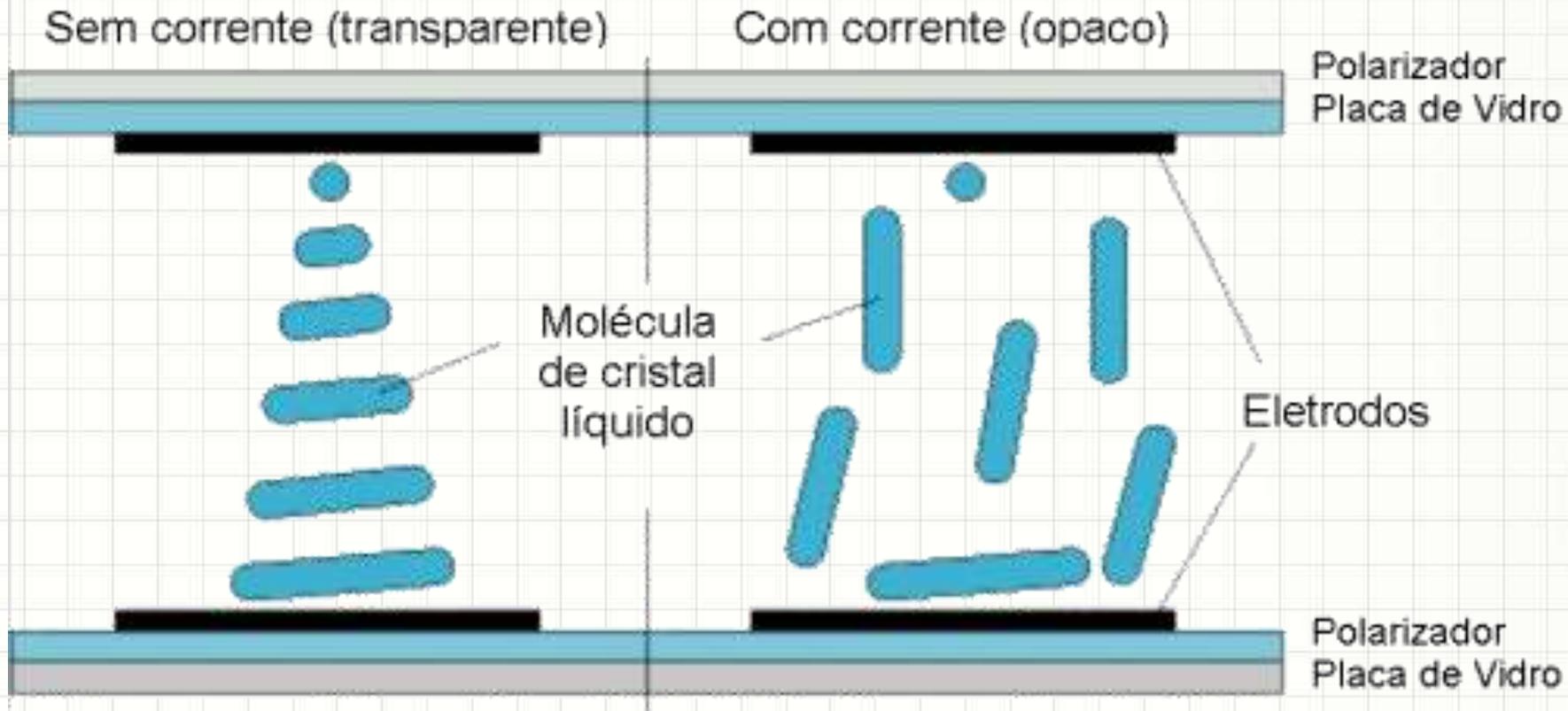
Monitores de Vídeo LCD

- Elementos Fundamentais:
 - emissor, polarizador vert., tela de cristal líquido, plástico filtro colorido, polarizador horizontal



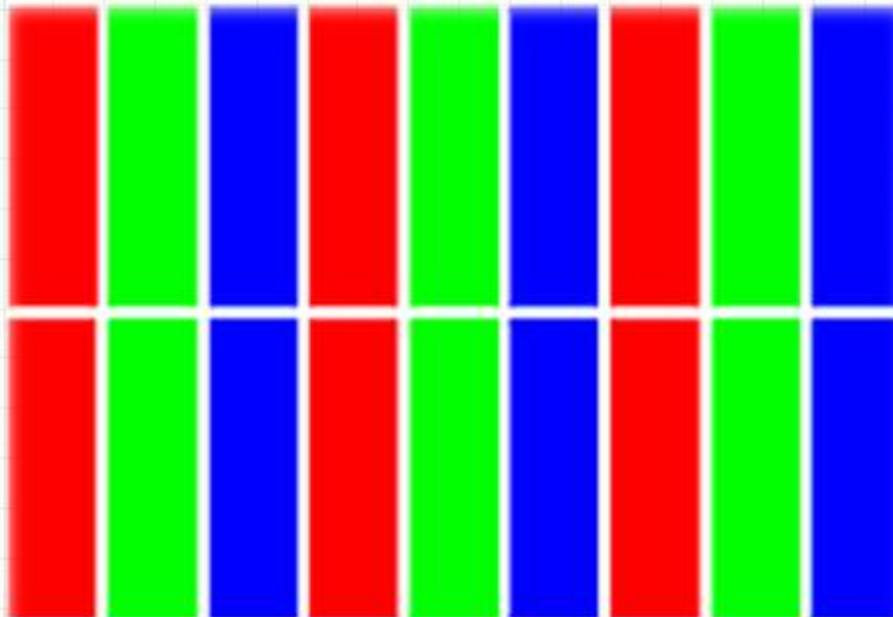
Monitores de Vídeo LCD

- Emissor, tela de cristal e polarizadores controlam quanta luz passa

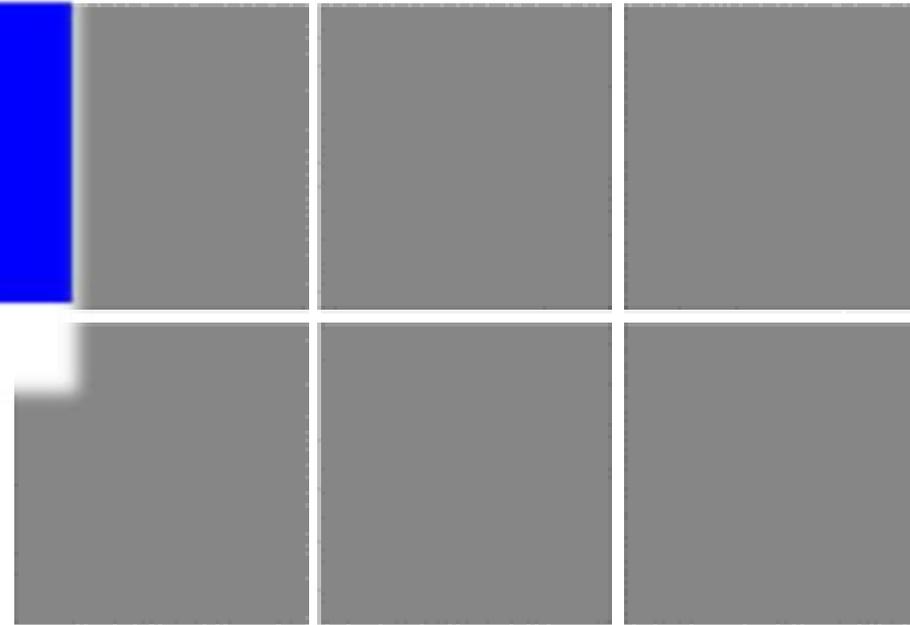


Monitores de Vídeo LCD

- O filtro controla a cor de cada “subpixel”



LCD Colorido

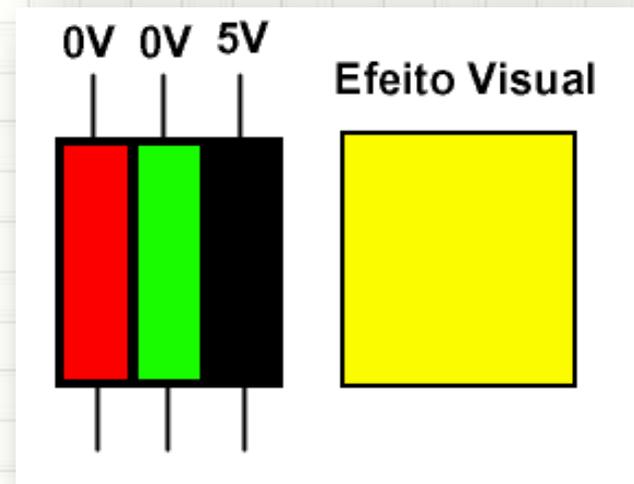
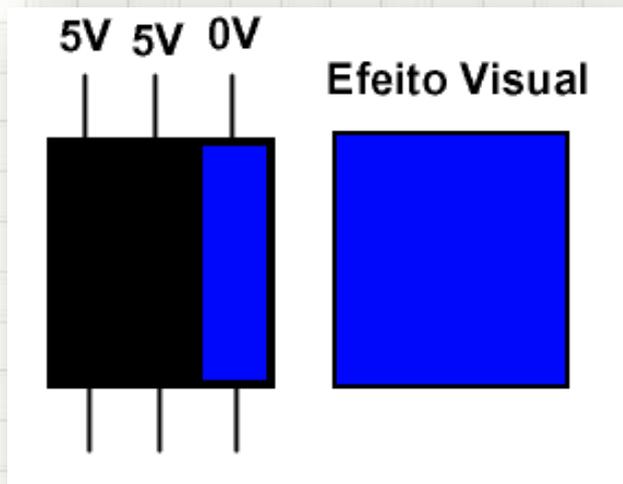
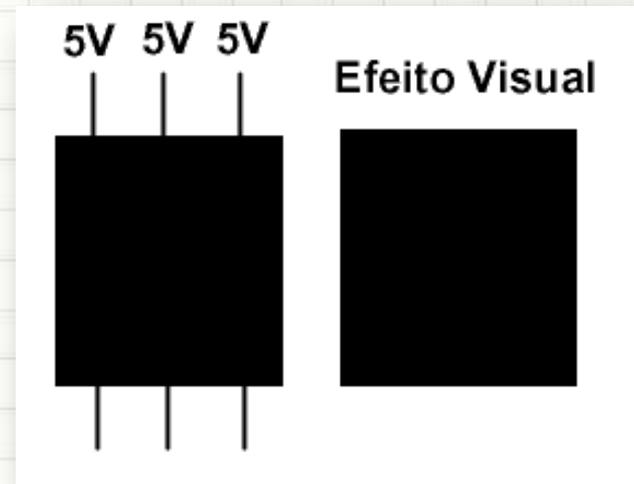
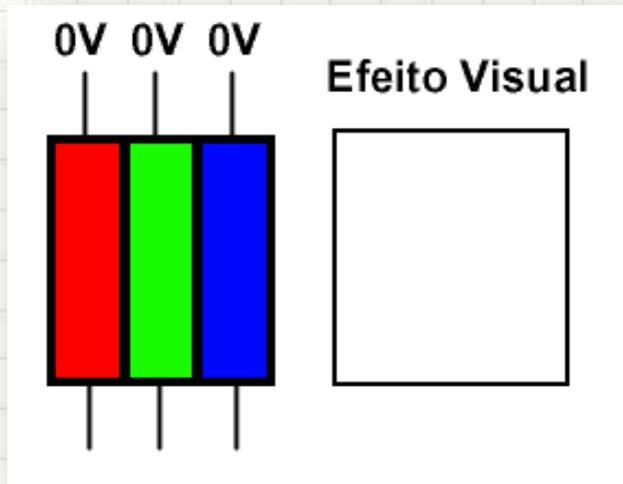


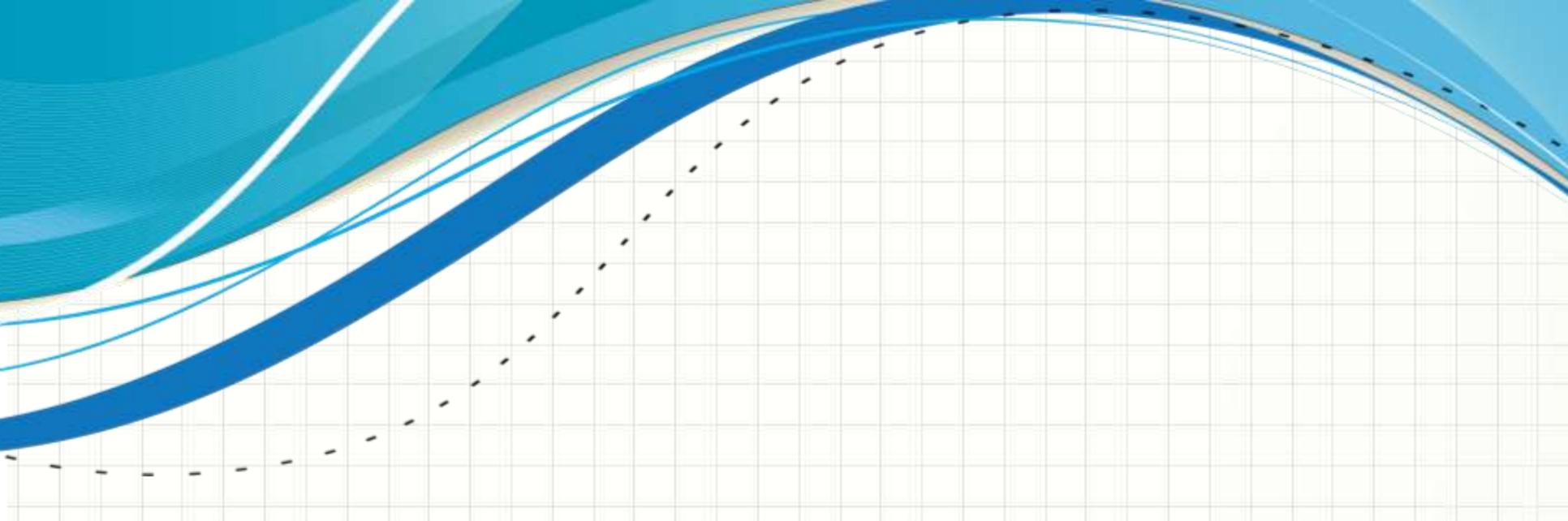
Pixel

LCD Monocromático

Monitores de Vídeo LCD

- Configurações e Efeitos





CONCLUSÕES

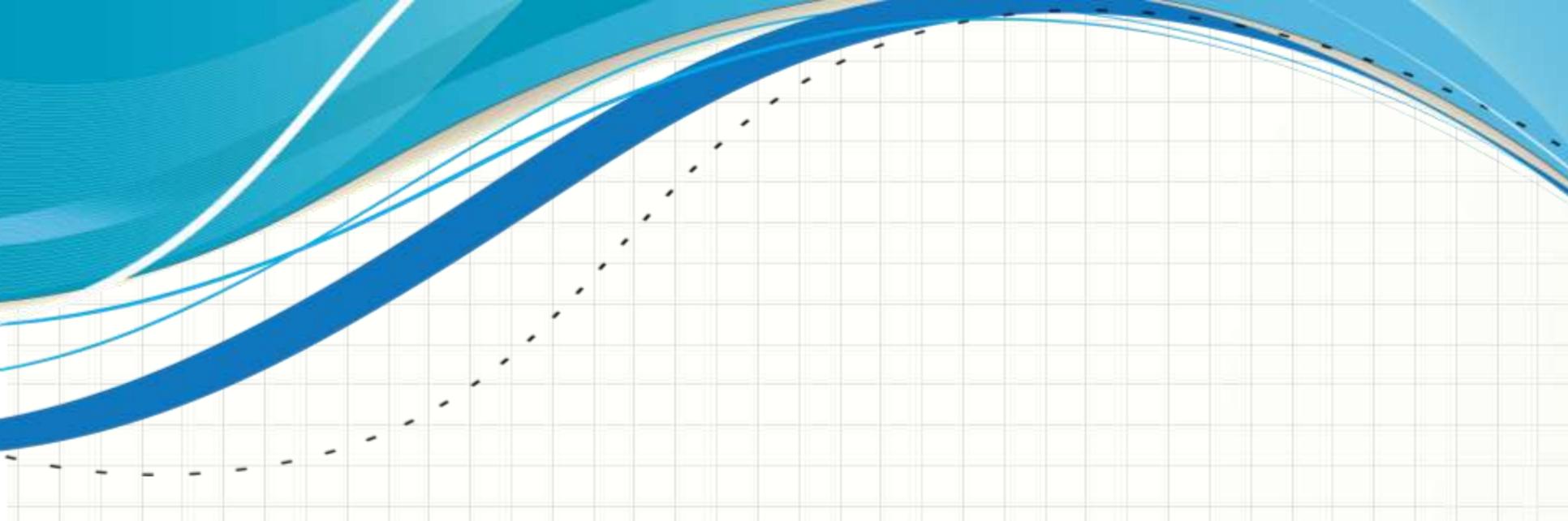
Resumo

- Os dispositivos são interconectados ao computador para converter sinais em informações e vice-versa
- O acesso aos dispositivos é feito pelo barramento, com auxílio do decodificador de endereços
- Dispositivos mais complexos exigem uma interface controladora
- Cada dispositivo lida com os sinais de uma maneira diferente
- **TAREFA**
– AV1!

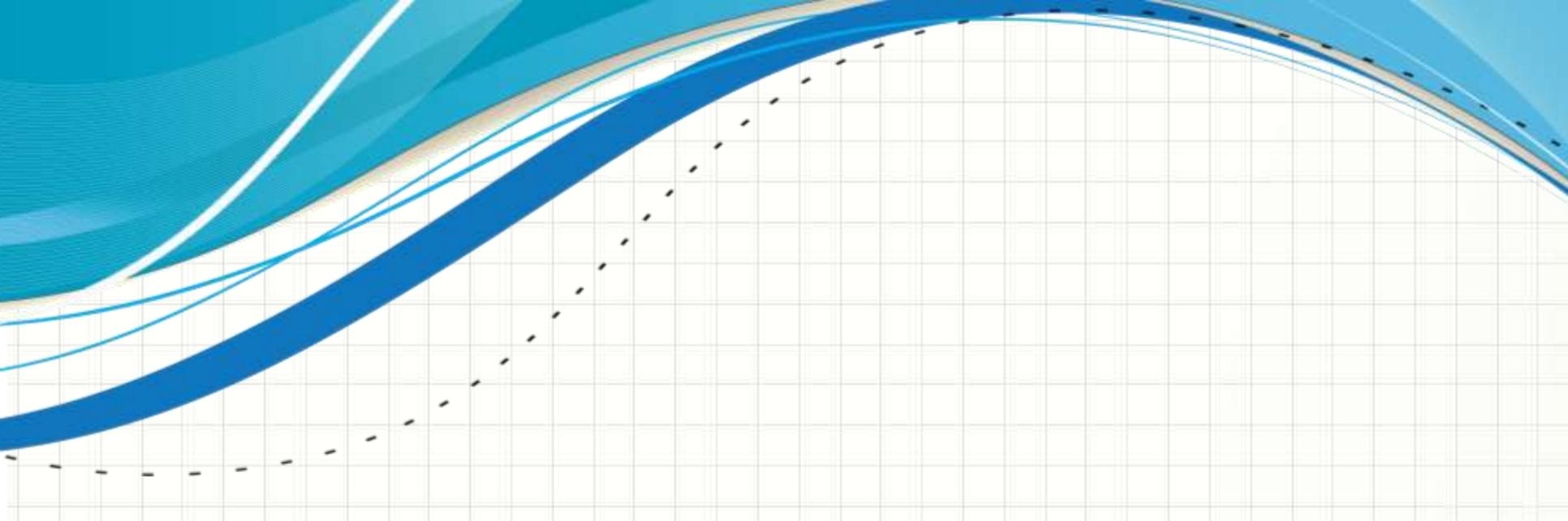
Próxima Aula



- Avaliação AV1!
- E depois?
 - Quem gerencia a memória?
 - E a CPU?
 - Sistemas Operacionais!
 - O que os SOs precisam?



PERGUNTAS?



**BOM DESCANSO
A TODOS!**