Arquitetura e Organização de Computadores

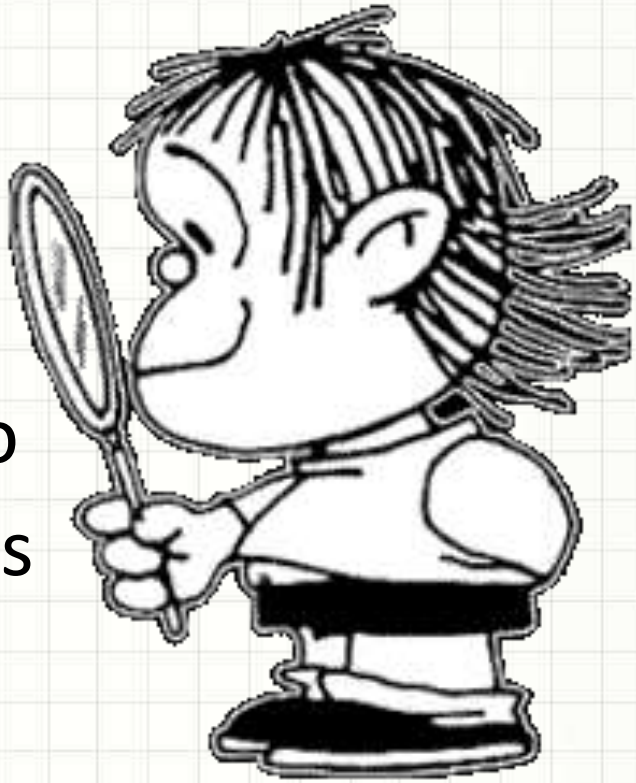
Dispositivos de Entrada e Saída

Prof. Dr. Daniel Caetano

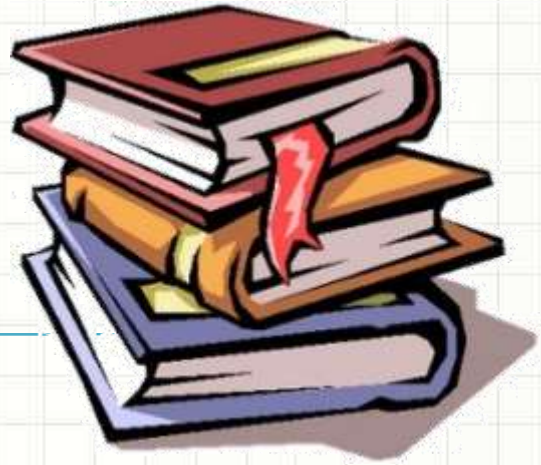
2012 - 1

Objetivos

- Compreender o que é um dispositivo
- Compreender o mecanismo de acionamento de um dispositivo
- Formas de controlar e interagir com um dispositivo
- Conhecer alguns dispositivos comuns
- **AV1**!



Material de Estudo

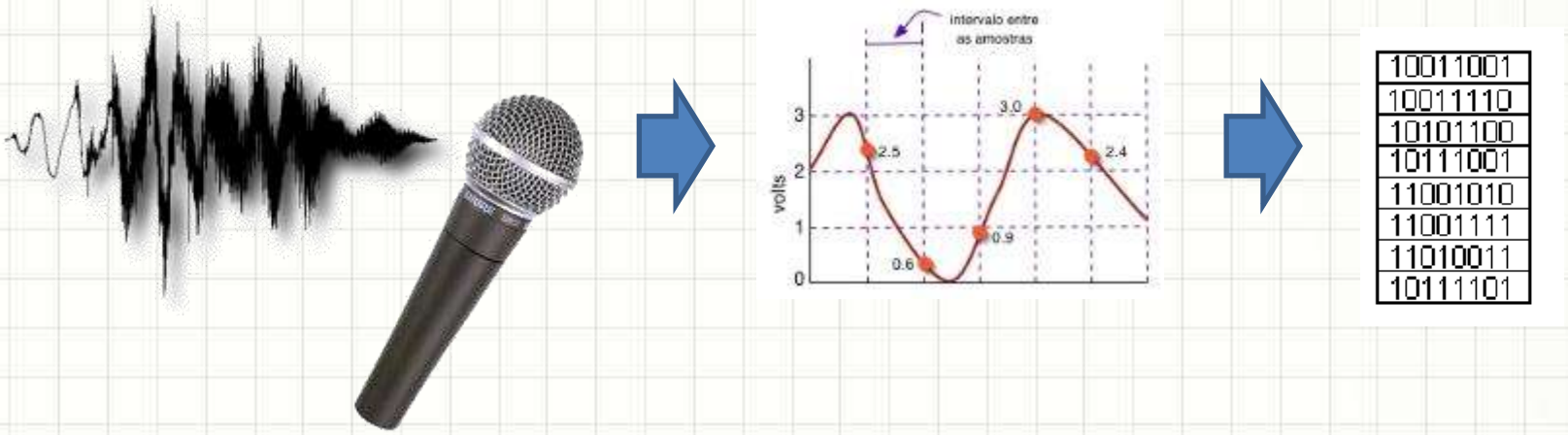


Material	Acesso ao Material
Notas de Aula	http://www.caetano.eng.br/aulas/aoc/ (Aula 7)
Apresentação	http://www.caetano.eng.br/aulas/aoc/ (Aula 7)
Material Didático	Arquitetura e Organização de Computadores, páginas 191 a 238
Biblioteca Virtual	-

O Que é um Dispositivo?

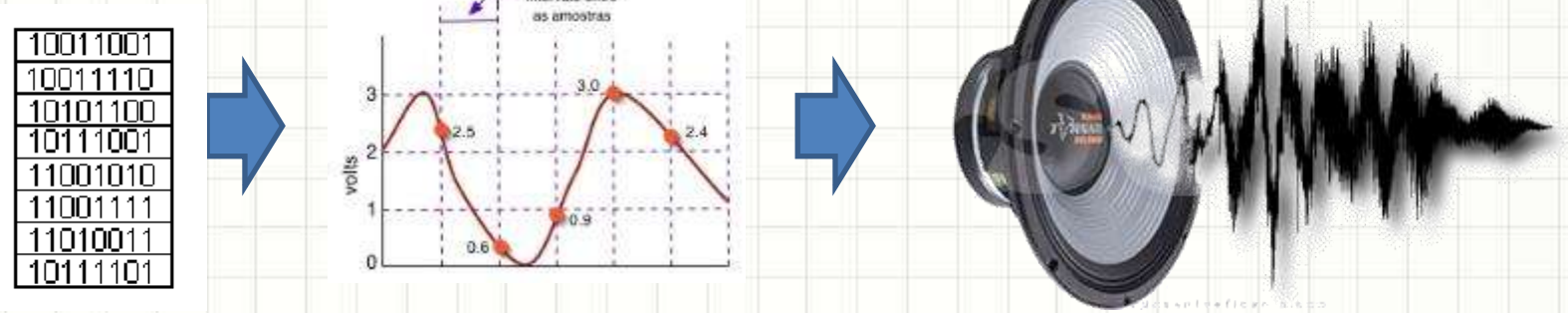
O que é um dispositivo?

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



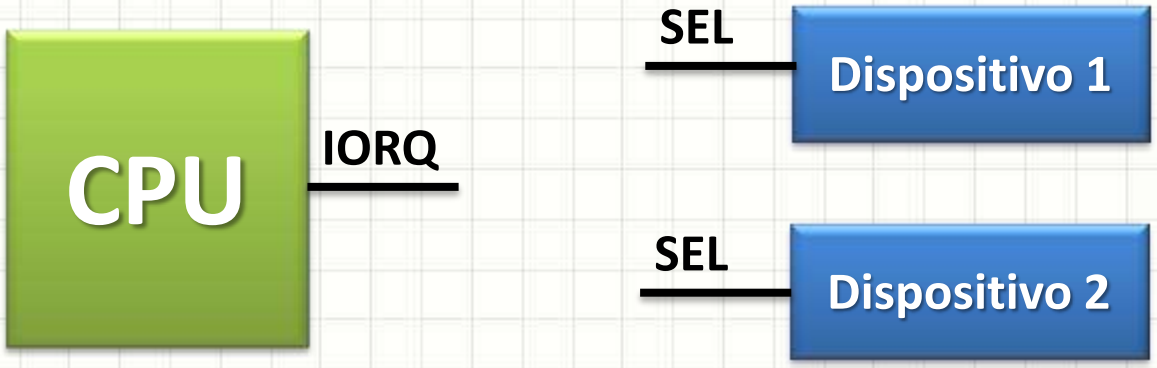
O que é um dispositivo?

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



O que é um dispositivo?

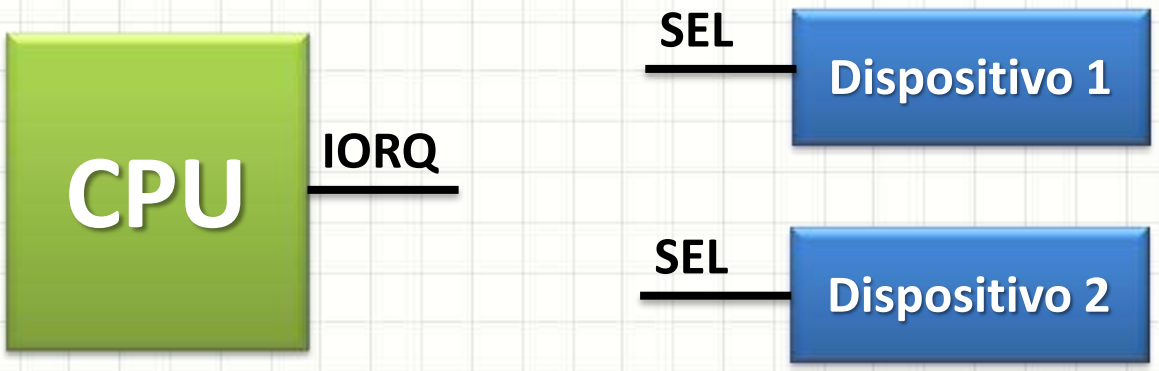
- Usualmente: pensam que são únicos
  - Não entendem endereços
  - Possuem apenas um “pino” de ativação
  - O tamanho da palavra é diferente da CPU
- CPU: apenas um “pino” de dispositivo: IORQ
  - Input/Output ReQuest



O que é um dispositivo?

- Usualmente: pensam que são únicos
  - –
  - –
  - –
- CPU: apenas um "pino" de dispositivo: IORQ
  - – Input/Output ReQuest

Como Resolver?



Decodificador de Endereços

Decodificador de Endereços

- Em princípio: “fios” do barramento de endereço estão “à toa”
- Que tal numerar os dispositivos...
  - ...e selecionar o dispositivo pelo “endereço”?
- Para isso funcionar:
  - Circuito que
    - Identifique um endereço
    - Acione o dispositivo correto
- Decodificador de Endereços

Decodificador de Endereços

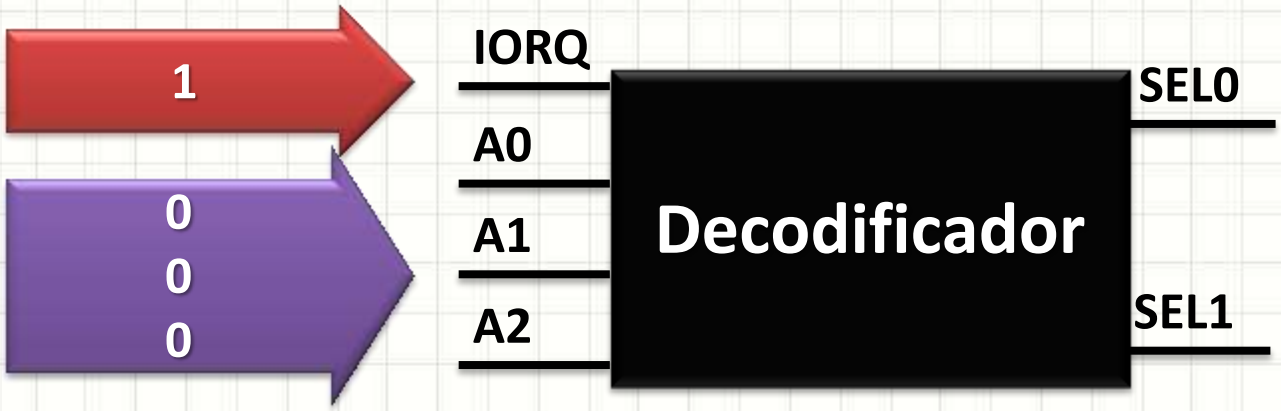
- Memória: **MREQ**, **R/W** e **A_0-A_n**
- E/S: **IORQ**, **R/W**... e **A_0-A_n**
- Observe quem faz a “ponte”!

Decodificador de Endereços

- Como funciona o decodificador?

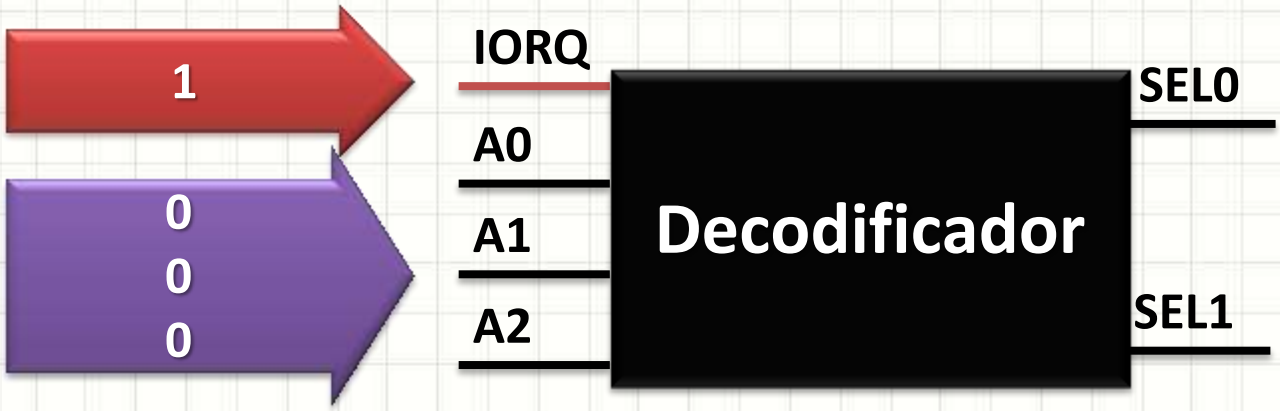
Decodificador de Endereços

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



Decodificador de Endereços

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

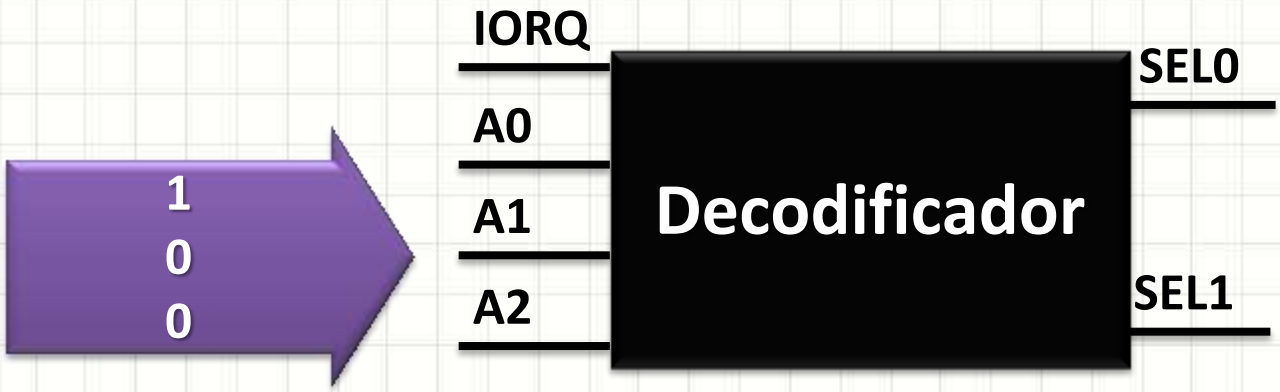


Decodificador de Endereços

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

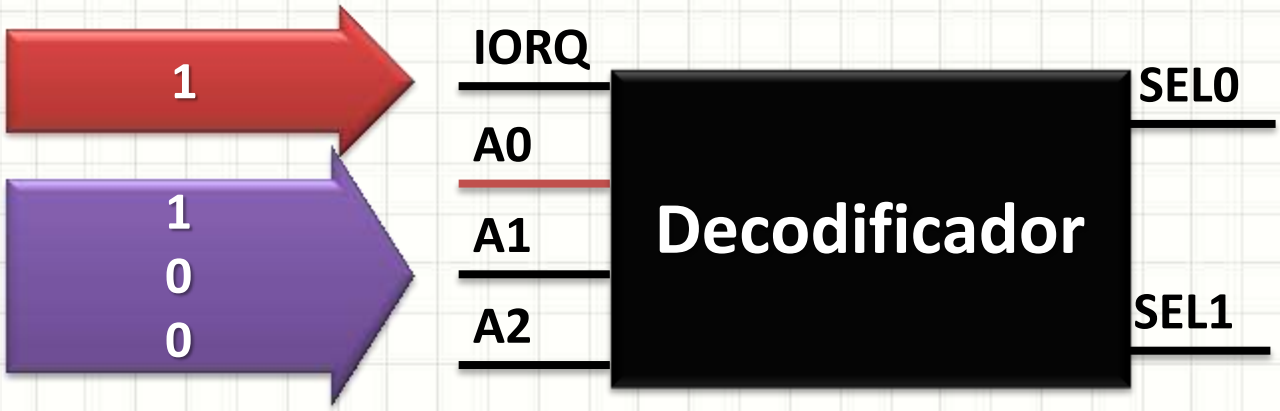
Decodificador de Endereços

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



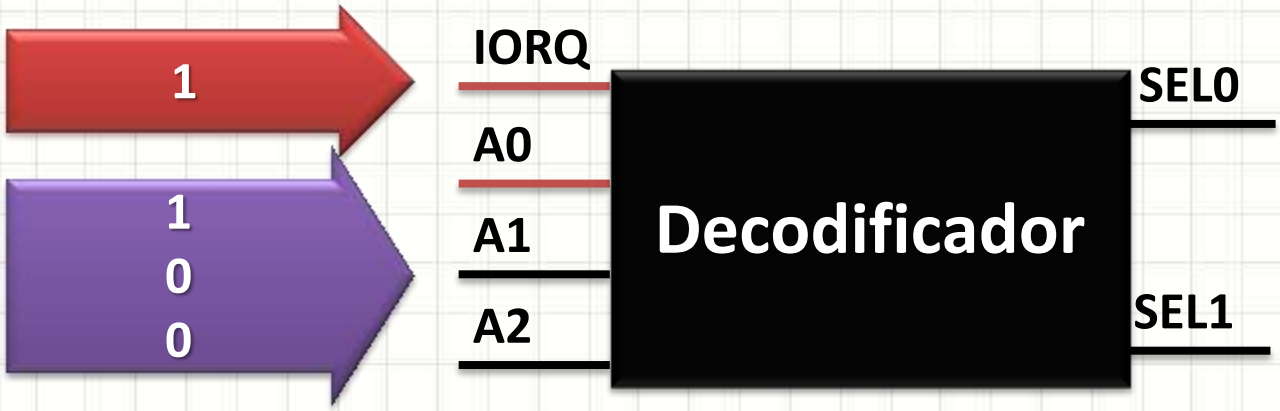
Decodificador de Endereços

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



Decodificador de Endereços

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

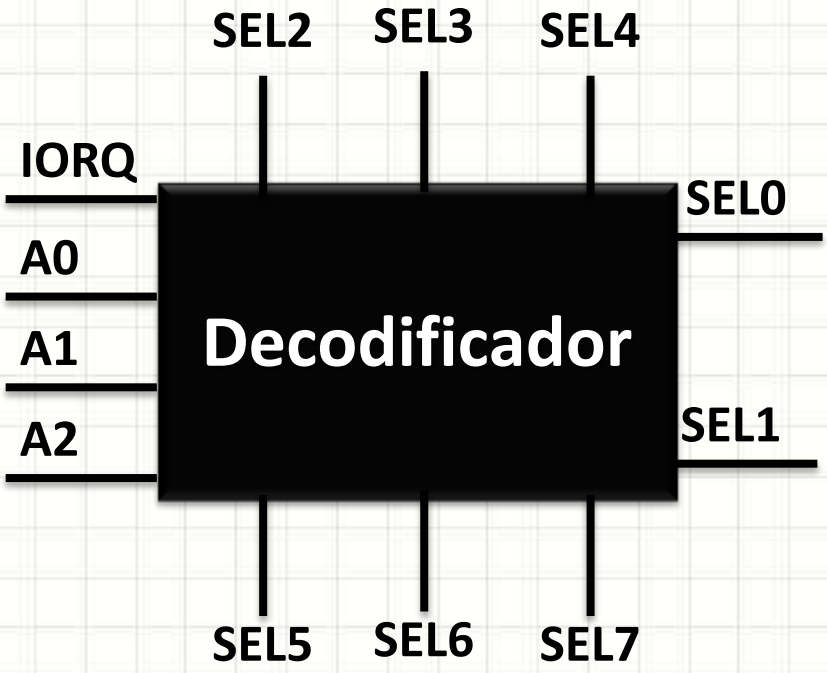


Decodificador de Endereços

- 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
  - Cada endereço específico ativa um dispositivo
- "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, vídeo...
- 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 (!)

Controlador de Dispositivos

- O controle ainda é da CPU?
  - Claro!
- Como a CPU fica nessa história?
- Como ela solicita algo?
- Como ela sabe o que está acontecendo?
- 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*

- 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 *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 Disp.↔Mem
  - Perda de dados, se a CPU não atender rapidamente um dispositivo (ex.: rede)

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

- 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 Interrupção

- CPUs modernas costumam ter várias IRQs
  - PIC: Programable 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

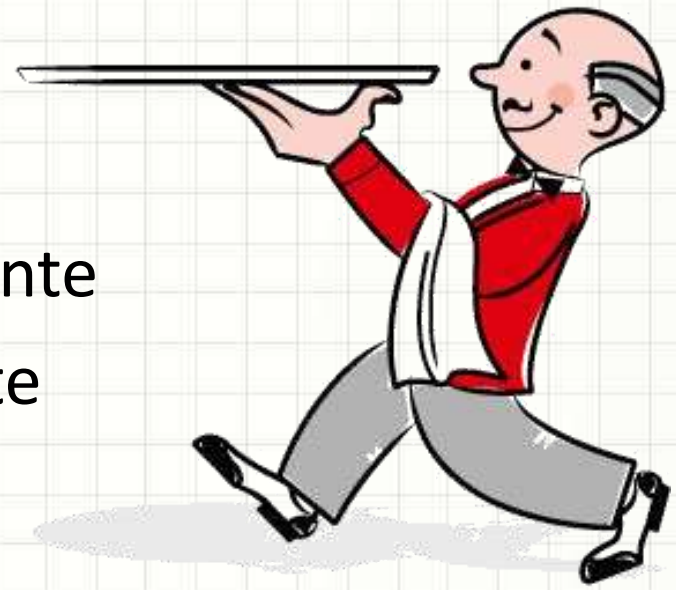
E/S POR DMA

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
- DMA: praticamente um dispositivo
  - Apenas executa as transferências
- 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

- Analogia:
  - CPU: Cozinheiro do restaurante
  - DMA: Garçom 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

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)

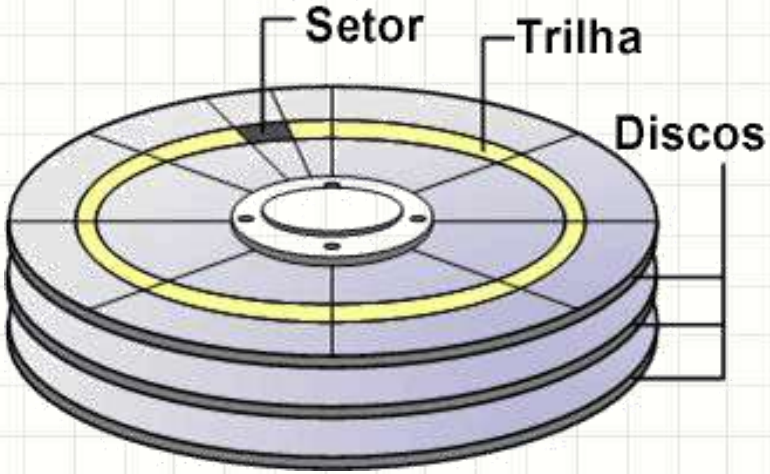
E/S por DMA + Interrupção

- 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

Dispositivos Comuns de E/S

Hard Disk

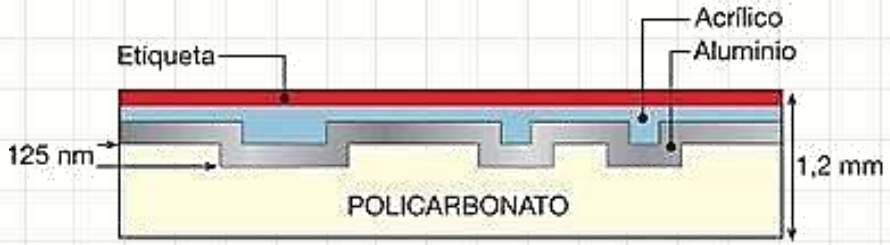
- Similar aos finados disquetes, mas...
  - Vários discos (de alumínio ou vidro)
  - Cobertos por Óxido de Ferro (magnetizado – 0, 1)
  - Fixos a um eixo, giram de 3.000 a 10.000 rpm
  - Regiões acessíveis: disco (head), trilha (cylinder), setor
  - Setores: usualmente 512 bytes



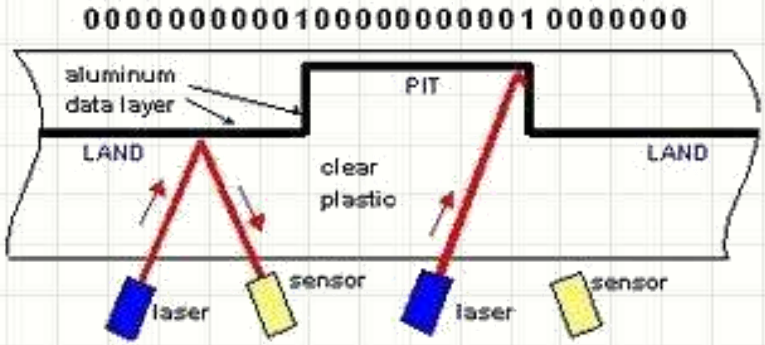
- 6 Cabeças
- 1 por vez
- Leitura: setor?
- Sistema de arquivos

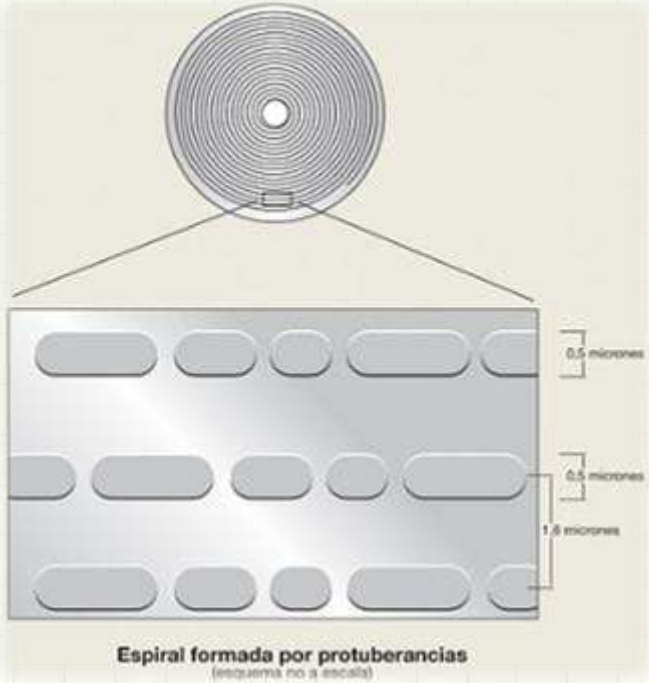
Discos Ópticos

- 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)

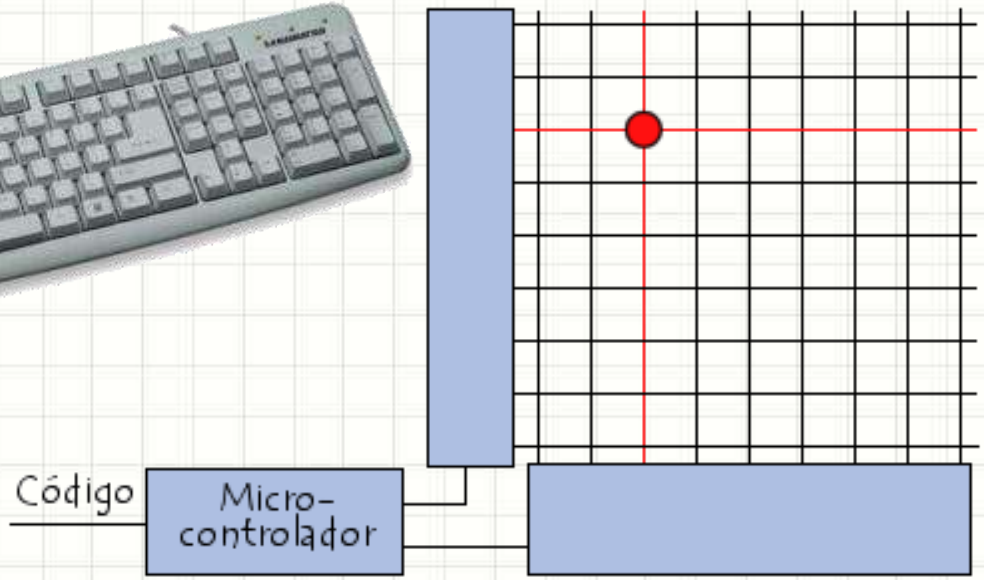


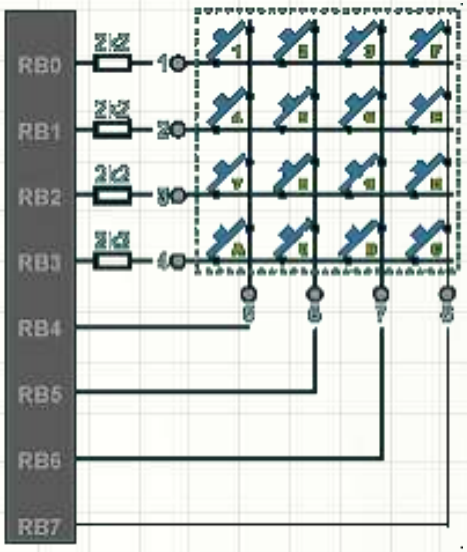


Espiral formada por protuberancias
(esquema no a escala)

Teclados

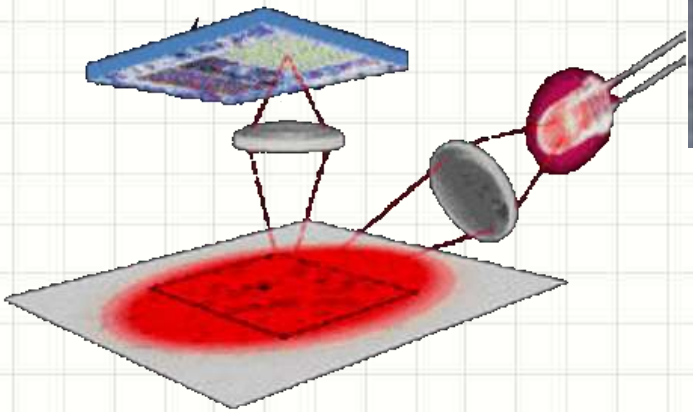
- Matriz Eletrônica
  - Simplificado: A → Linha 04, coluna 02 → 0402
  - Teclas modificadoras enviadas separadamente
  - Mapa de Teclado: Linha/Coluna → Código
  - US-International, ABNT-2 etc.

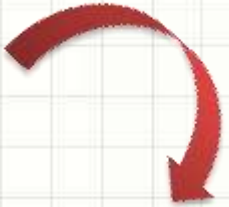




Mouse Óptico

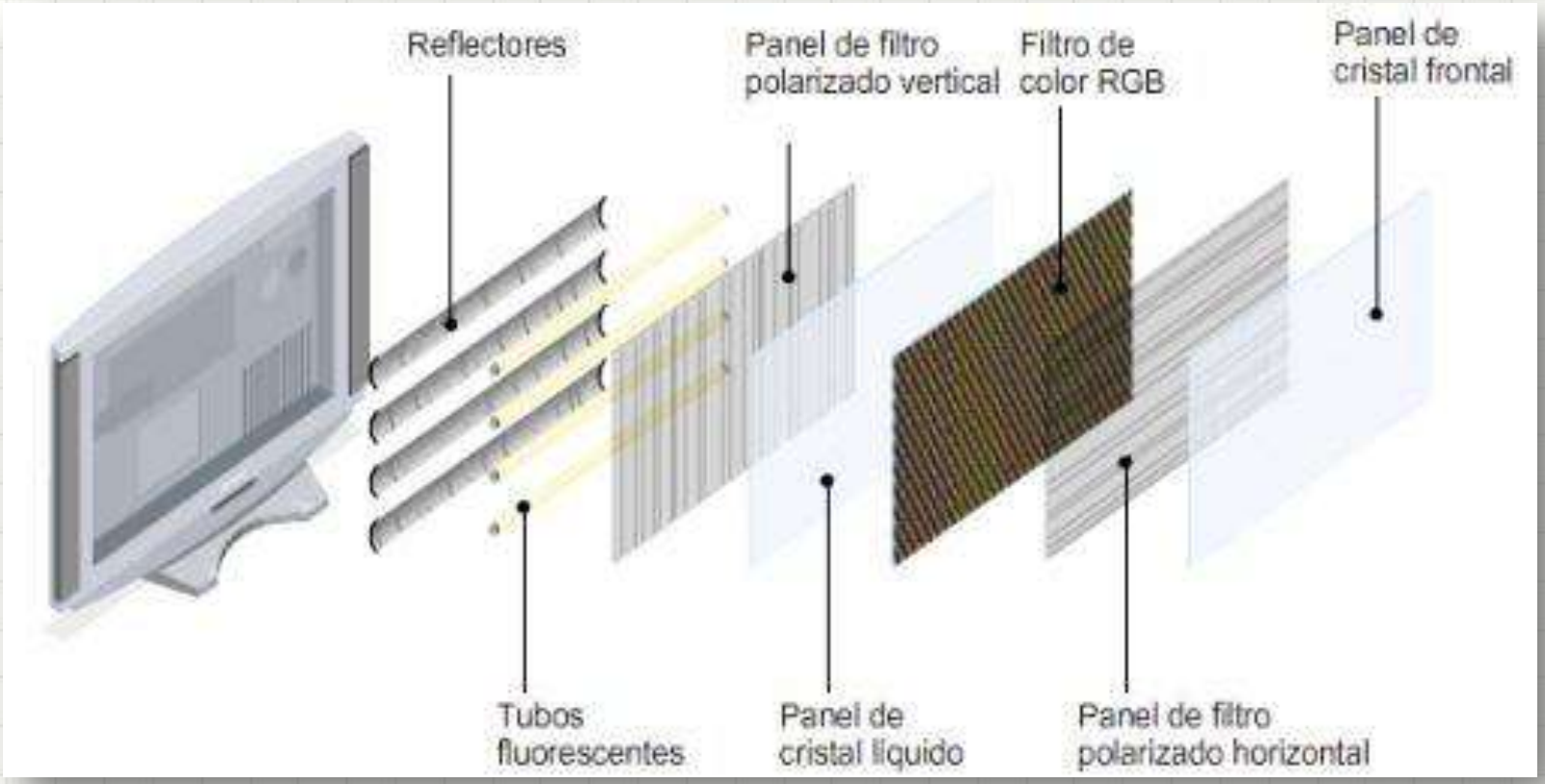
- Fotos da Base de Apoio
  - Emite luz e digitaliza imagem: intervalos regulares
  - Detecção de deslocamento entre imagens
  - Codificação em um valor para o mouse





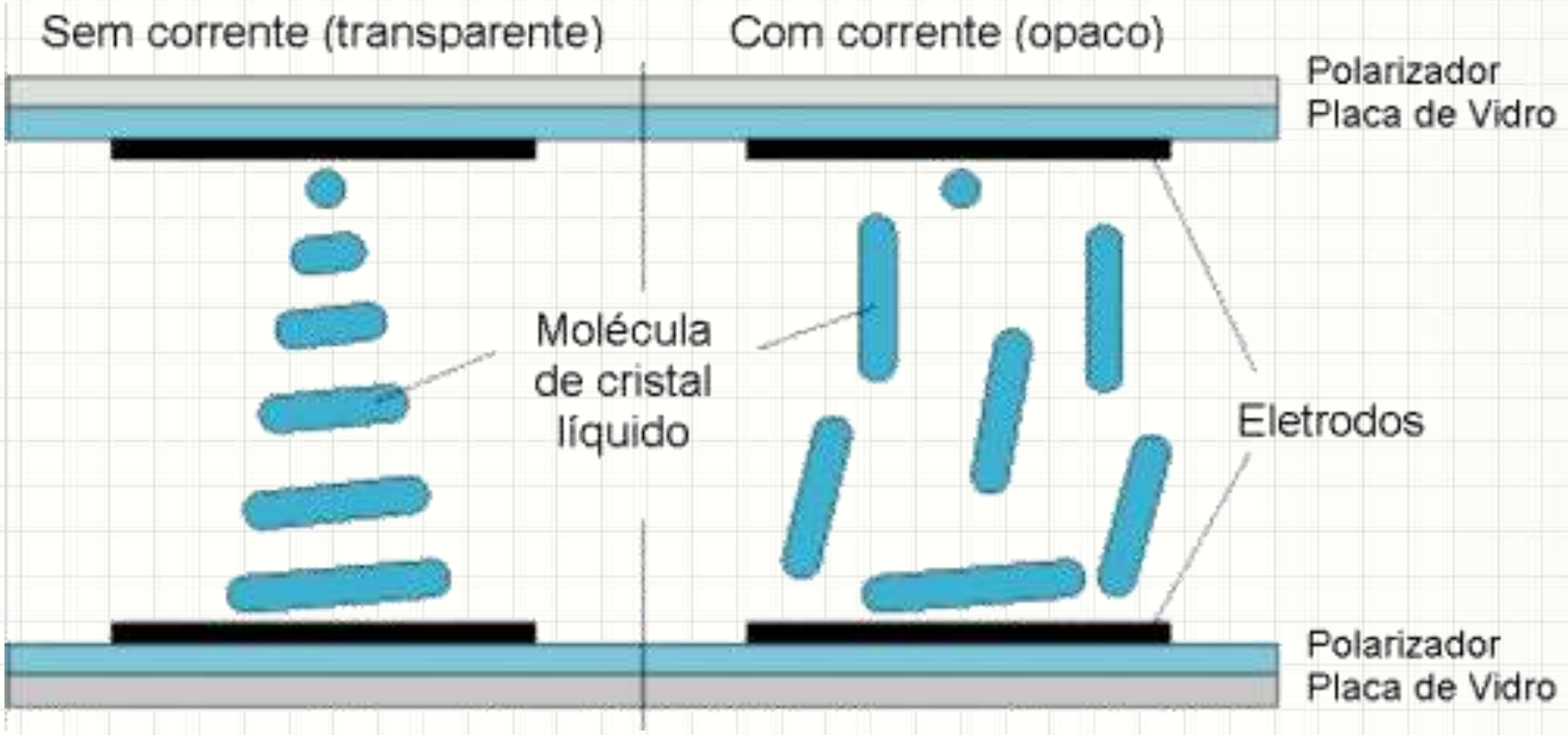
Monitores de Vídeo LCD

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



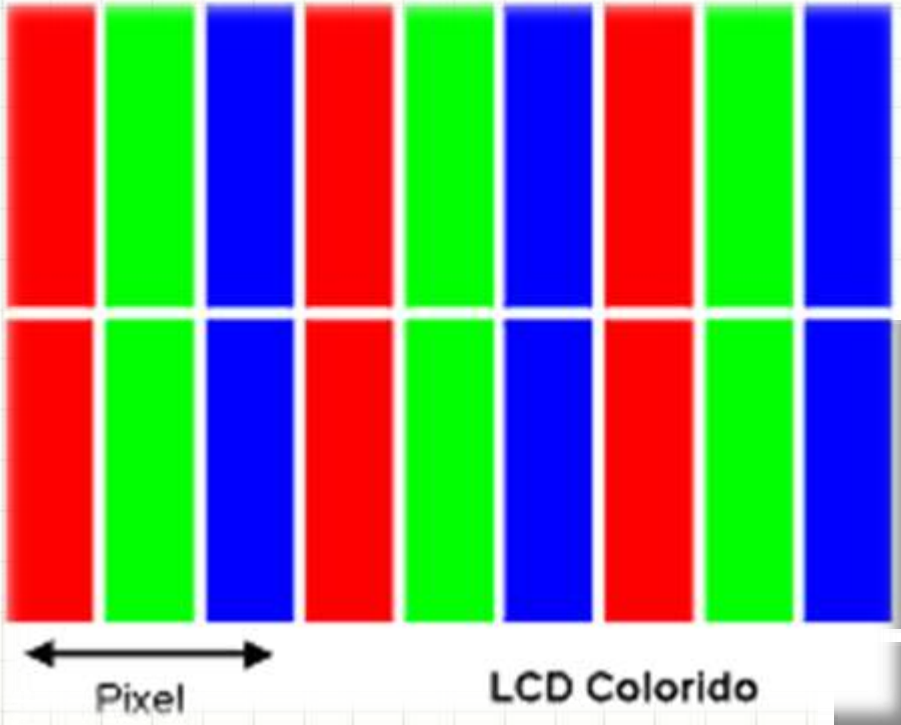
Monitores de Vídeo LCD

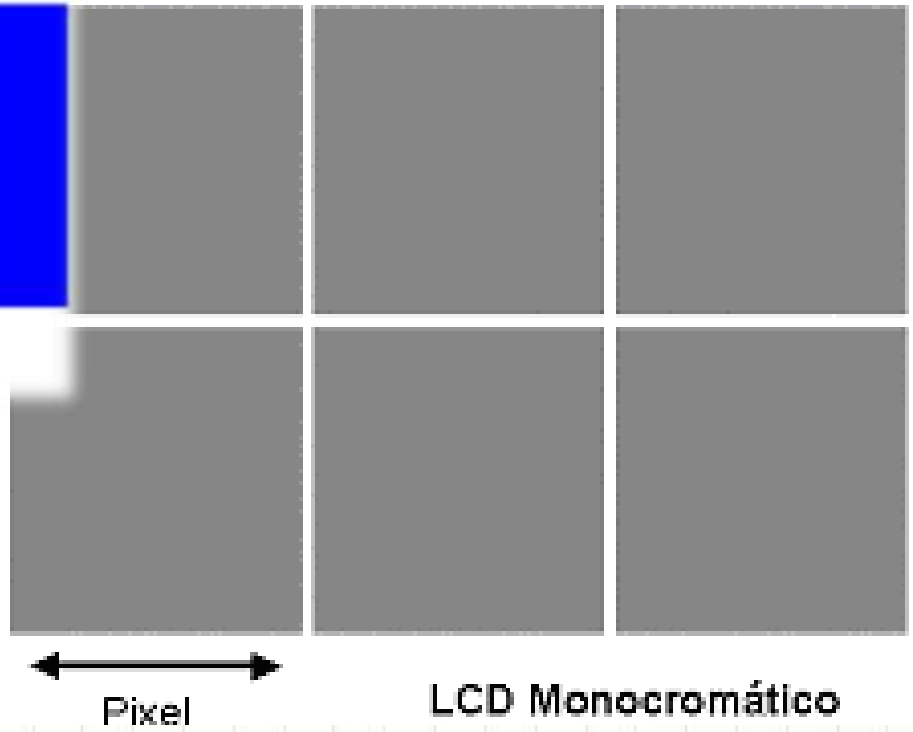
- Emissor, tela de cristal e polarizadores controlam quanta luz passa



Monitores de Vídeo LCD

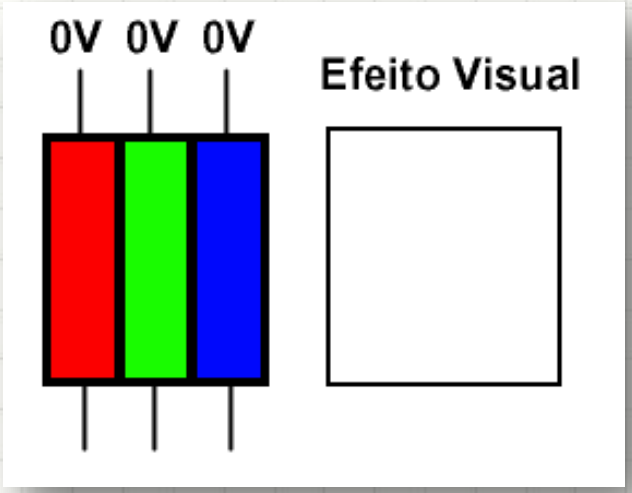
- O filtro controla a cor de cada “subpixel”

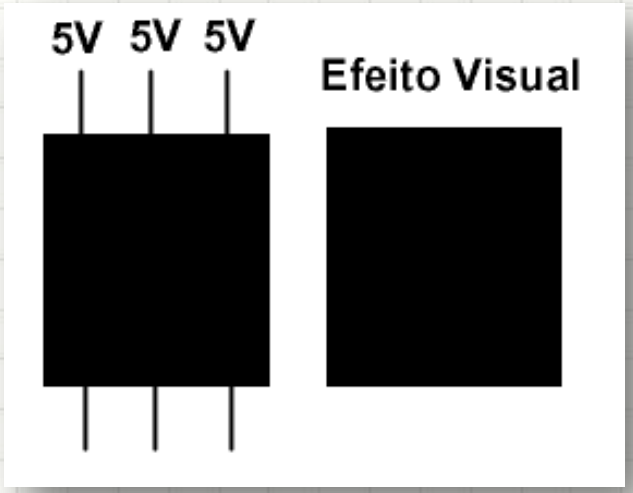


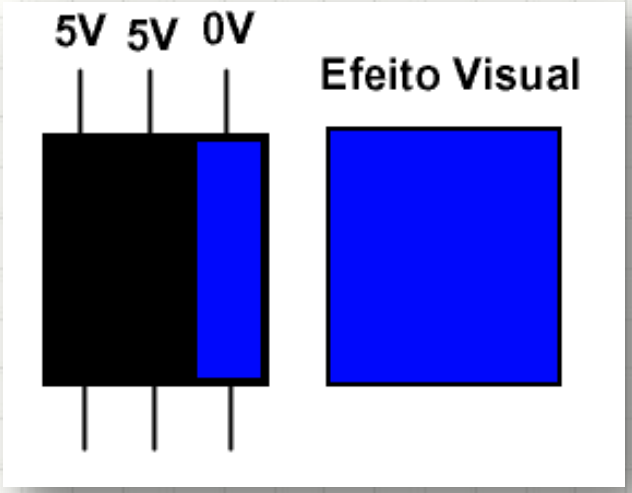


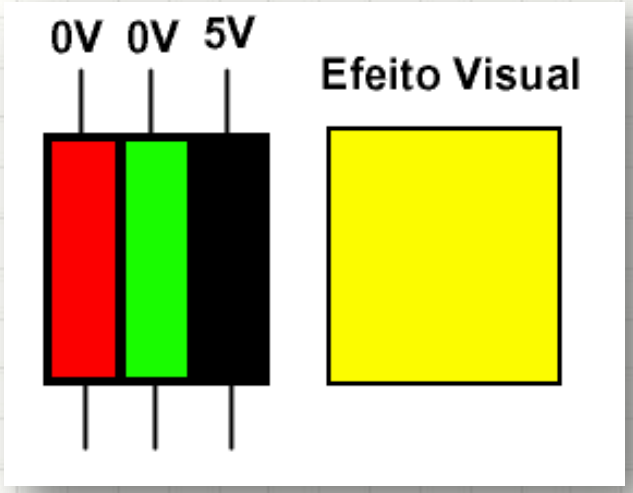
Monitores de Vídeo LCD

- 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!