Microcontroladores e Interfaces

3º Ano – Eng. Electrónica Industrial

Carlos A. Silva

2º Semestre de 2004/2005
http://www.dei.uminho.pt/lic/mint

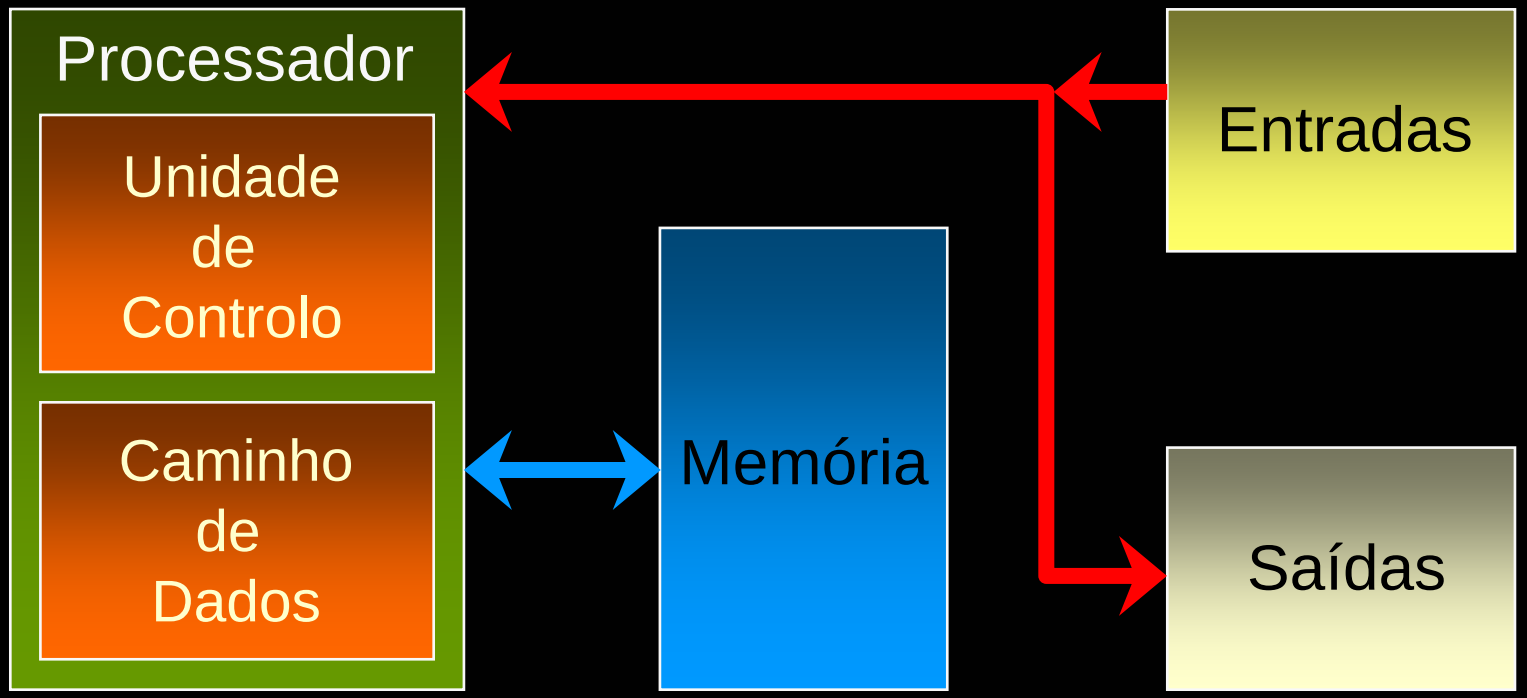
Assunto: Busses
Aula #11

O que é um *bus*

BUS: Traduz-se como *barramento* e não autocarro !!!

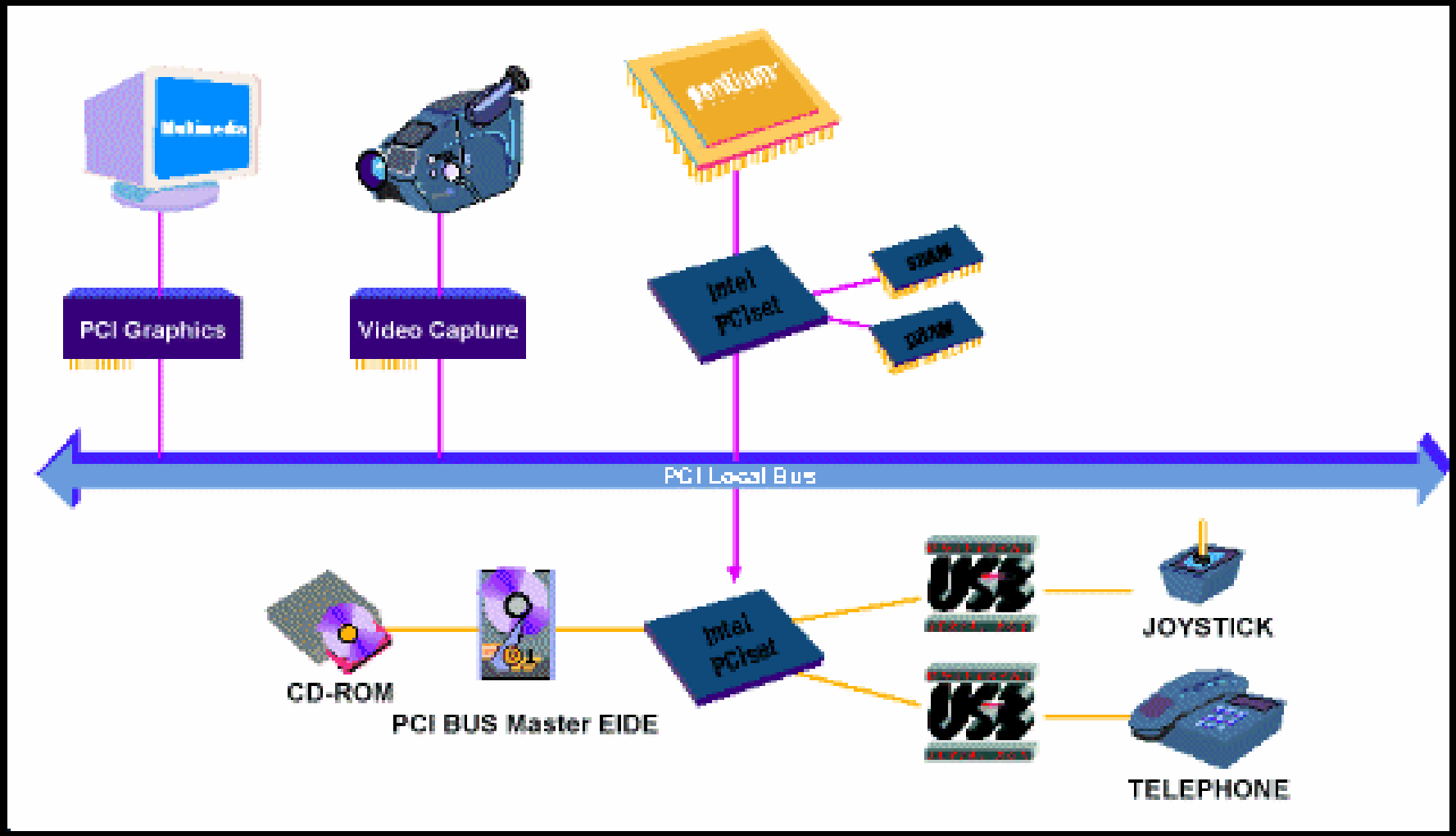
- **O barramento é**
  - Uma ligação partilhada para comunicação.
  - Conjunto de linhas usado para conectar múltiplos sub-sistemas

O que é um *bus*

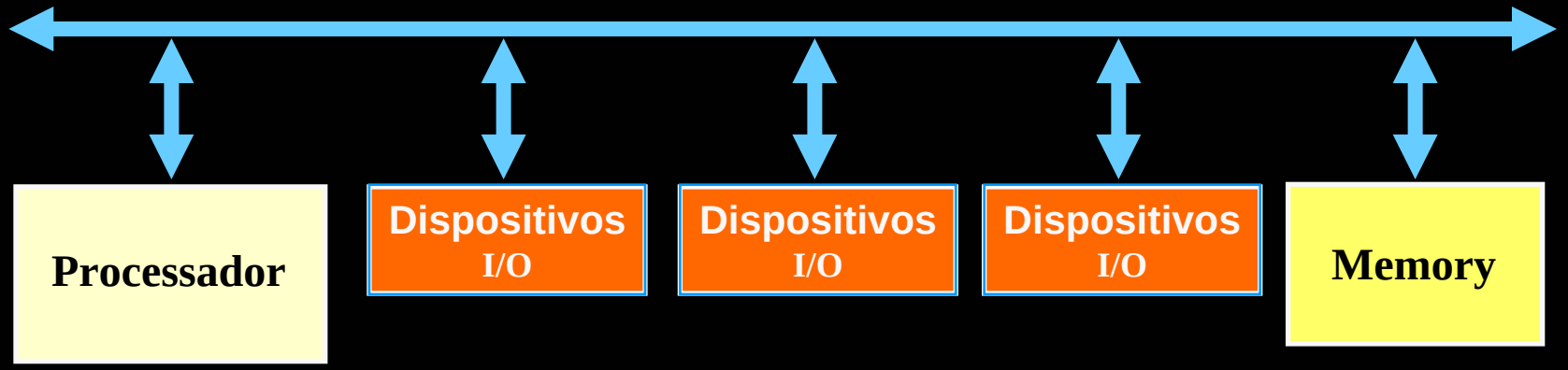


- **O barramento também é usado como uma ferramenta fundamental na composição de sistemas complexos**
  - Permite a abstracção da ligação de sub-sistemas

Barramentos

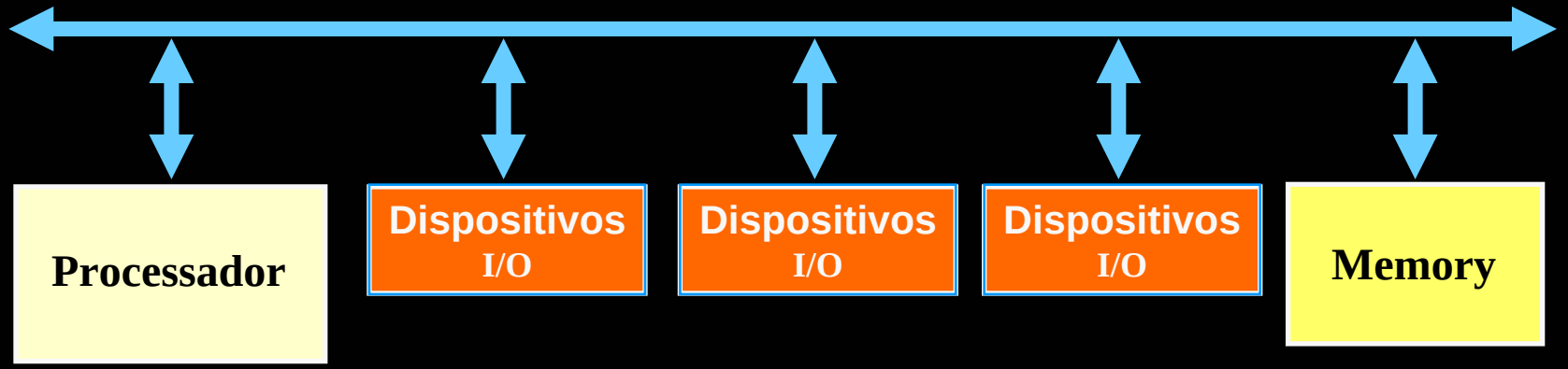


Vantagens do barramento



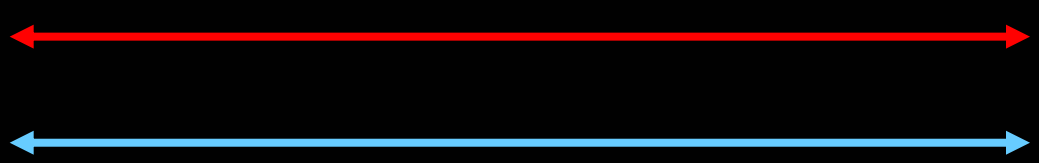
- **Versatilidade**
  - Novos dispositivos podem ser adicionados
  - Periféricos podem ser movidos entre computadores que usem o mesmo *standard* para o barramento (*p. ex., USB*)
- **Baixo Custo**
  - Um simples conjunto de fios pode ser partilhado de diversas maneiras

Desvantagens do barramento



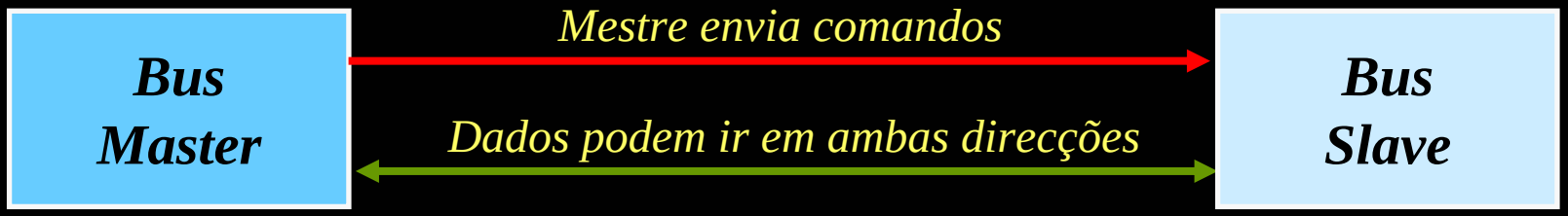
- **O barramento cria um *afunilamento* na comunicação**
  - A largura de banda do barramento pode limitar o *throughput* máximo do sistema I/O
- **A velocidade máxima do barramento será grandemente limitada pelos seguintes factores:**
  - Comprimento do barramento
  - Número de dispositivos 'pendurados' no barramento
  - A necessidade de suportar uma gama de dispositivos com
    - Grande variabilidade de latência
    - Grande variabilidade na velocidade de transferência de dados

Organização geral de um barramento



- **Linhas de Controlo**
  - Sinalizam pedidos (*'requests'*) e confirmações (*'acknowledgements'*)
  - Indicam qual é o tipo de dado presente na linha de dados
- **Linha de Dados**
  - Transporta endereços e dados
  - Comandos complexos

Transacção: Topologia Mestre / escravo



- **Uma transacção do barramento envolve duas fases**
  - Envio de um comando (e endereço) – '*request*'
  - Transferência de dados – acção
- **O mestre é responsável por iniciar a transacção ao**
  - Enviar o comando (e endereço)
- **O escravo é aquele que responde ao comando ao**
  - Enviar dados pedidos pelo mestre se este pedir dados
  - Receber dados do mestre se este quiser enviar dados

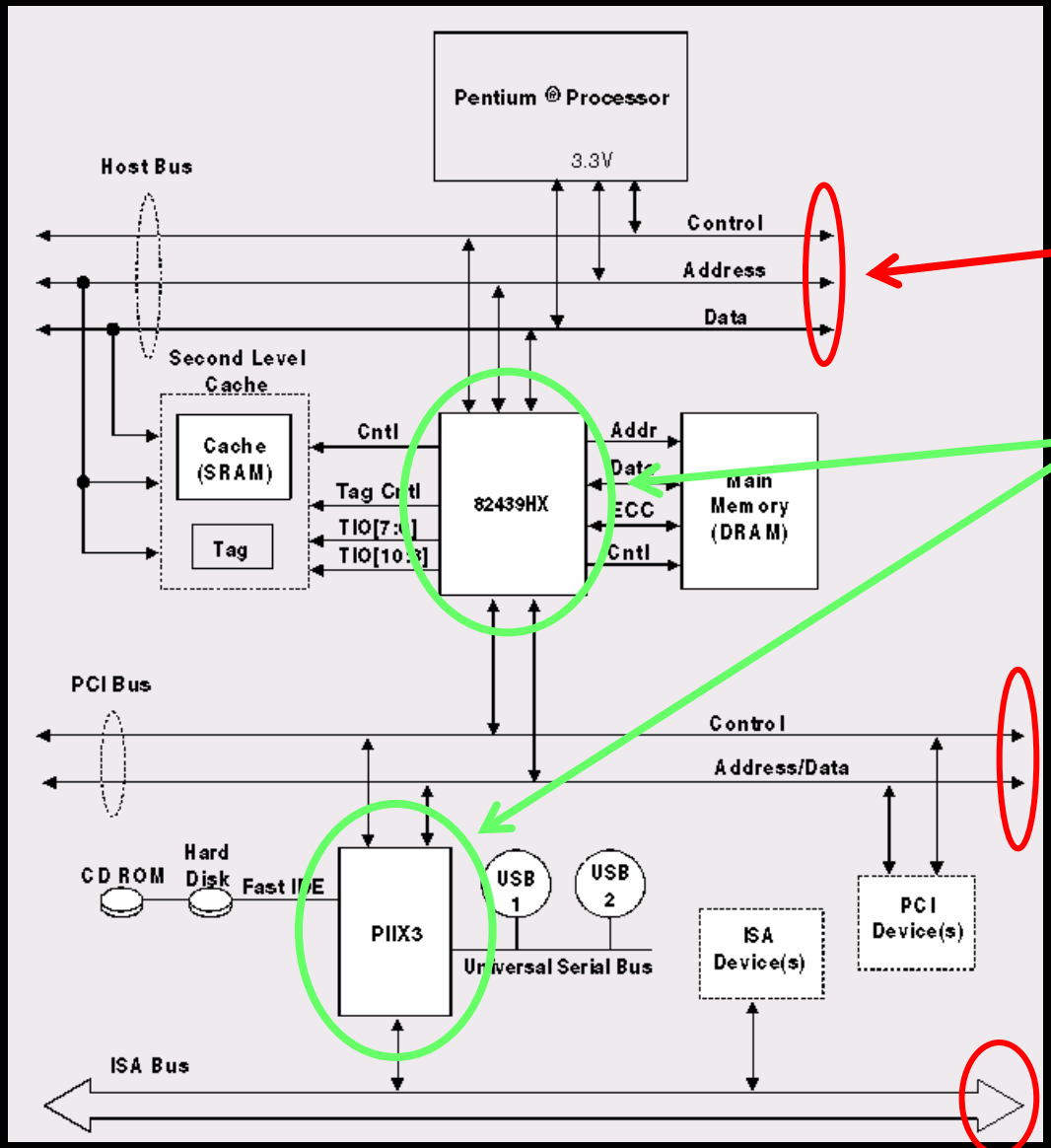
Tipos de barramentos

- **Existem três tipos de barramentos**
- **Barramento processador-memória** (***específico ao projecto do processador***)
  - Curto e de alta velocidade
  - Precisa apenas se adaptar as características do sistema de memória
    - Maximizar a largura de banda da ligação processador-memória
  - Conecta directamente ao processador
  - Optimizado para transferência de blocos de dados da *cache*

Tipos de barramentos

- **Barramento I/O (*standard industrial*)**
  - Geralmente é comprido e lento
  - Tem que ser adaptado a uma larga gama de dispositivos I/O
  - Liga ao barramento processador-memória ou ao barramento *backplane*
- **Barramento *backplane***
  - *Backplane*:
    - Estrutura de ligação dentro do chassis
  - Permite a coexistência entre processadores, memória e dispositivos I/O
  - Vantagem do baixo custo:
    - Um barramento para todos os componentes

Exemplo: Organização do Sistema *Pentium*



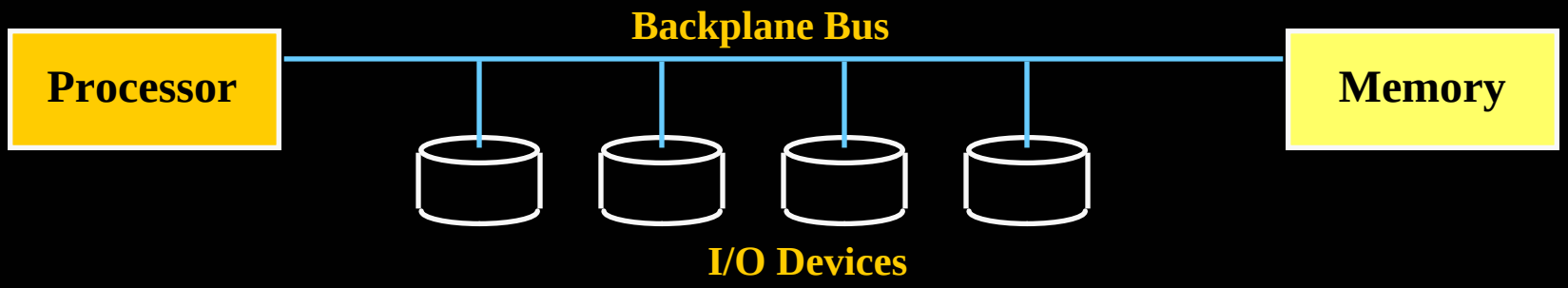
barramento processador-memória

bridge* ou *hub

***barramento PCI* (barramento *backplane*)**

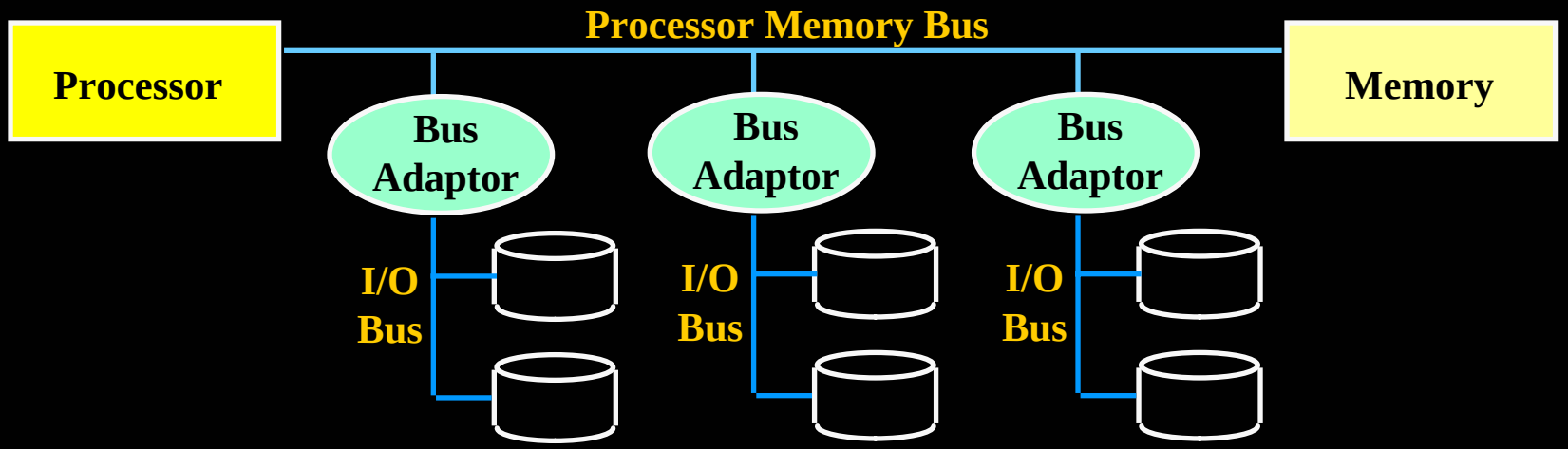
barramento I/O

Sistema com um só barramento: barramento *backplane*



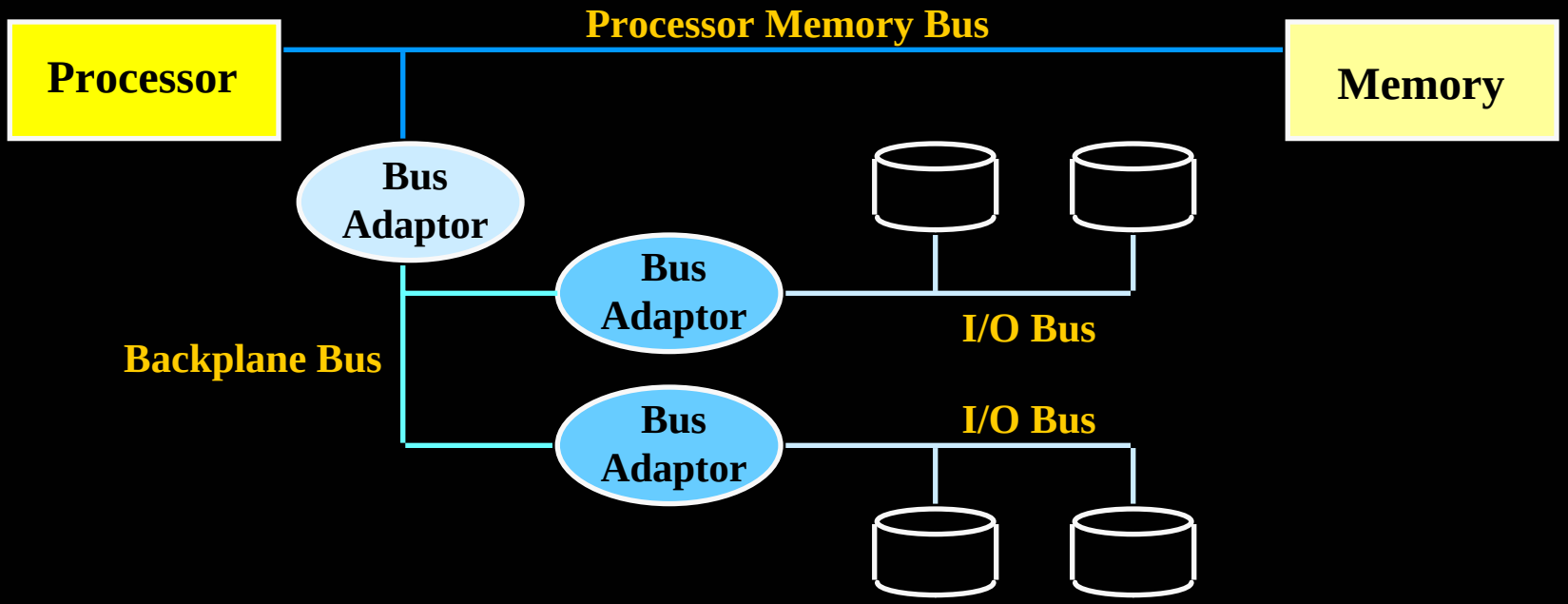
- **Barramento único (barramento *backplane*) é usado para:**
  - Comunicação processador-memória
  - Comunicação entre dispositivos I/O e memória
- **Vantagens:**
  - Simples e de baixo custo
- **Desvantagens:**
  - Lento (*o barramento pode se tornar no maior afunilamento do sistema*)
- **Exemplo:**
  - IBM PC – AT

Sistema com dois barramentos



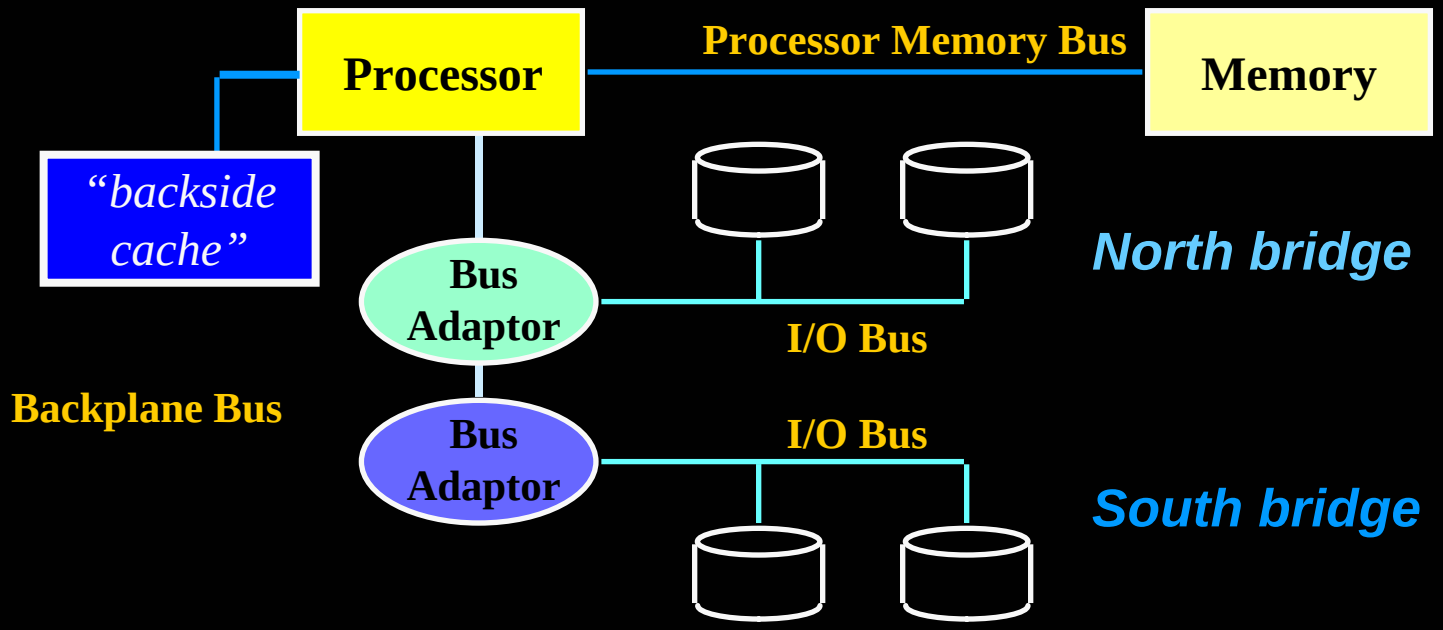
- **O barramento I/O é ligado ao barramento processador-memória através de uma adaptador de barramento (*bridge*)**
  - Barramento processador-memória: usado essencialmente para o tráfico processador-memória
  - Barramento I/O: fornece *slots* de expansão para dispositivos I/O
- **Apple MacIntosh-II**
  - Barramento NuBus: processador, memória e alguns dispositivos de I/O previamente seleccionados
  - Barramento SCCI: responsável pelo resto dos dispositivos I/O

Sistema com três barramentos



- **Um pequeno número de barramentos *backplane* são ligados ao barramento processador-memória**
  - O barramento processador-memória é usado apenas para o tráfego processador-memória
  - Os barramentos I/O são ligados ao barramento *backplane*
- **Vantagem**
  - A carga no barramento do processador é reduzida consideravelmente

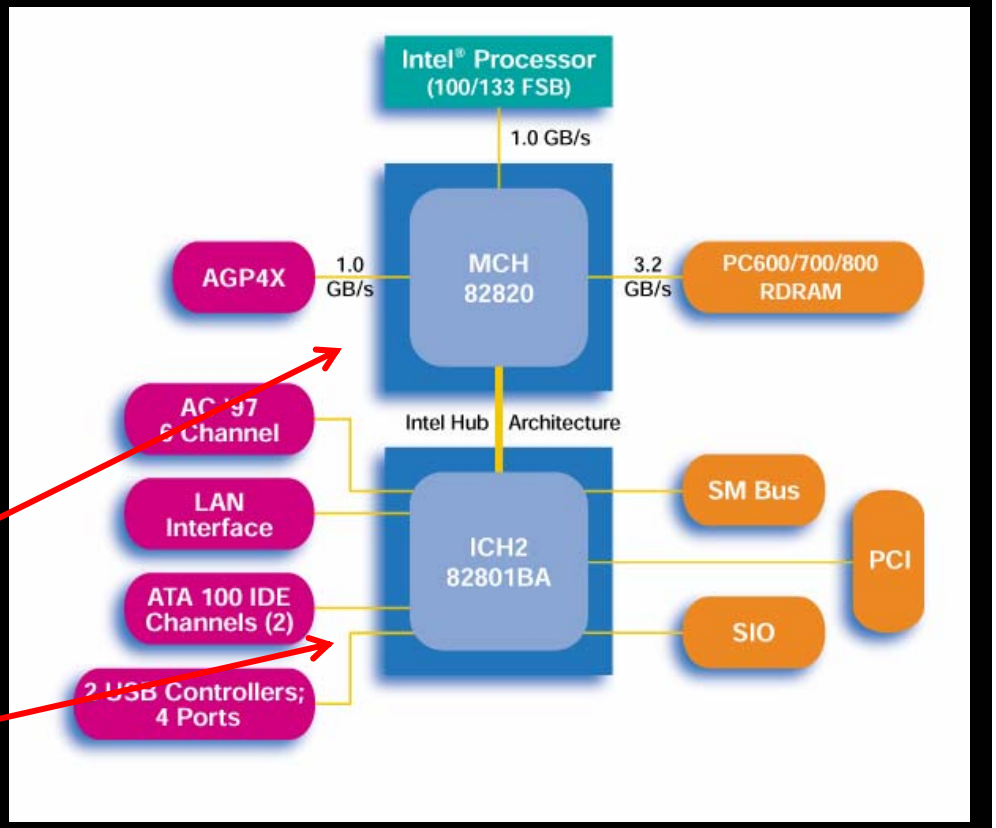
Arquitectura com *bridges* Norte/Sul: Barramentos separados



- **Conjunto de pinos diferentes para diferentes funções**
  - Barramento de memória
  - Caches
  - Barramento gráfico (*frame buffer rápido*)
  - Barramentos I/O são ligados ao barramento *backplane*

Arquitectura com *bridges* Norte/Sul: Barramentos separados

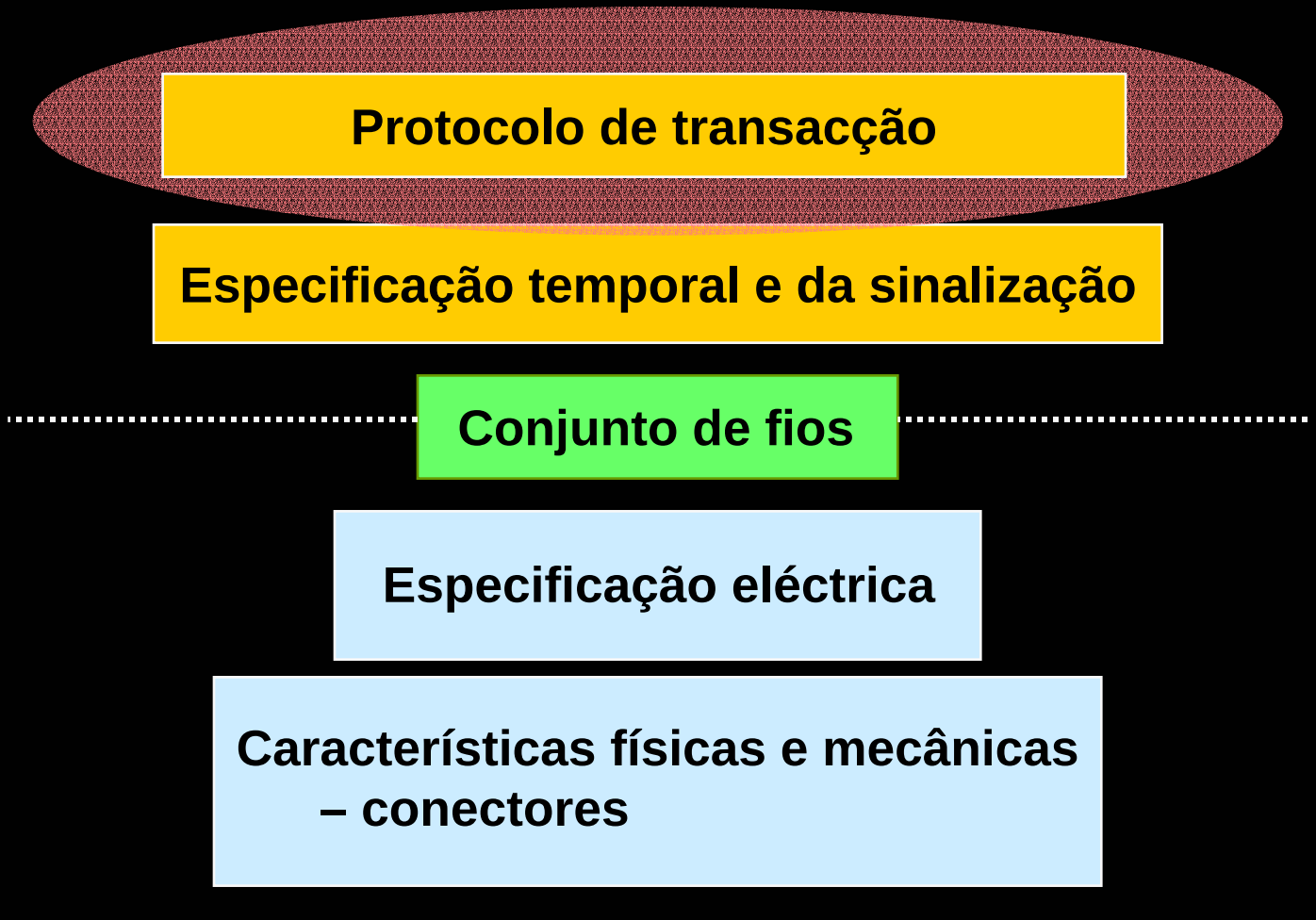
- **Vantagens**
  - Os barramento de podem operar a diferentes velocidades
  - A carga total é muito menor



Exemplo: *Pentium III*

- ***North bridge***
- ***South bridge***

O que define um barramento ?



Barramento síncrono e barramento assíncrono

- **Barramento síncrono**
  - Inclui um sinal de relógio nas linhas de controlo
  - Protocolo de comunicação pré-fixado (relativo ao relógio)
  - Vantagem:
    - Requer pouca lógica e permite velocidades mais elevadas
  - Desvantagens:
    - Cada dispositivo no barramento tem que suportar a mesma velocidade
    - De modo a evitar o skew do sinal de relógio, o comprimento do barramento deve ser curto para poder atingir velocidades elevadas

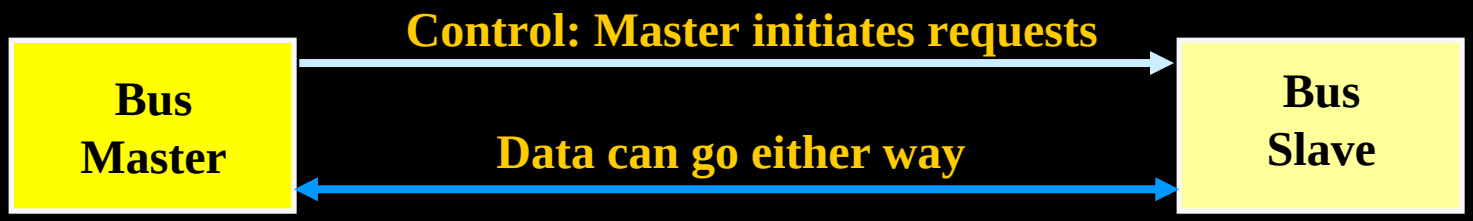
Barramentos síncronos e barramentos assíncronos

- **Barramentos assíncronos**
  - Não tem sinal de relógio
  - Pode acomodar um largo espectro de dispositivos
  - Pode atingir comprimentos maiores sem o problema do *skewing* do sinal de relógio
  - Requer o uso de um protocolo de *handshaking*

Transacção no barramento

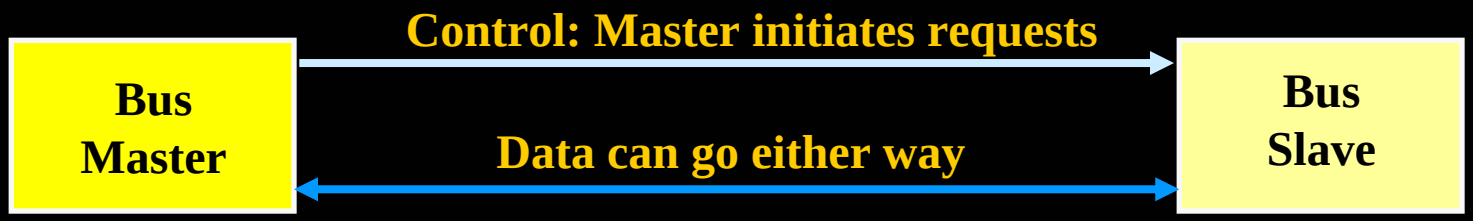
- **Três questões básicas:**
  - Arbitragem
    - Quem terá acesso ao barramento ?
  - Pedido
    - O que se pretende fazer ?
  - Acção
    - O que acontecerá na resposta ?

Arbitragem: Obtendo acesso ao barramento



- **O aspecto mais importante no projecto do barramento:**
  - Como o barramento é reservado pelo dispositivo que pretende usá-lo ?
    - Quem terá acesso ao barramento ?
- **O caos é impedido através da topologia mestre-escravo**
  - Apenas o *bus master* pode controlar o acesso ao barramento
    - Ele inicia e controla todos os pedidos de acesso ao barramento
  - Um escravo apenas responde a um pedido de leitura ou de escrita

Arbitragem: Obtendo acesso ao barramento



- **O sistema mais básico será**
  - O processador é o único mestre do barramento
  - Todos os pedidos de acesso ao barramento são controlados pelo processador
  - Principal limitação:
    - O processador está envolvido em todas as transacções

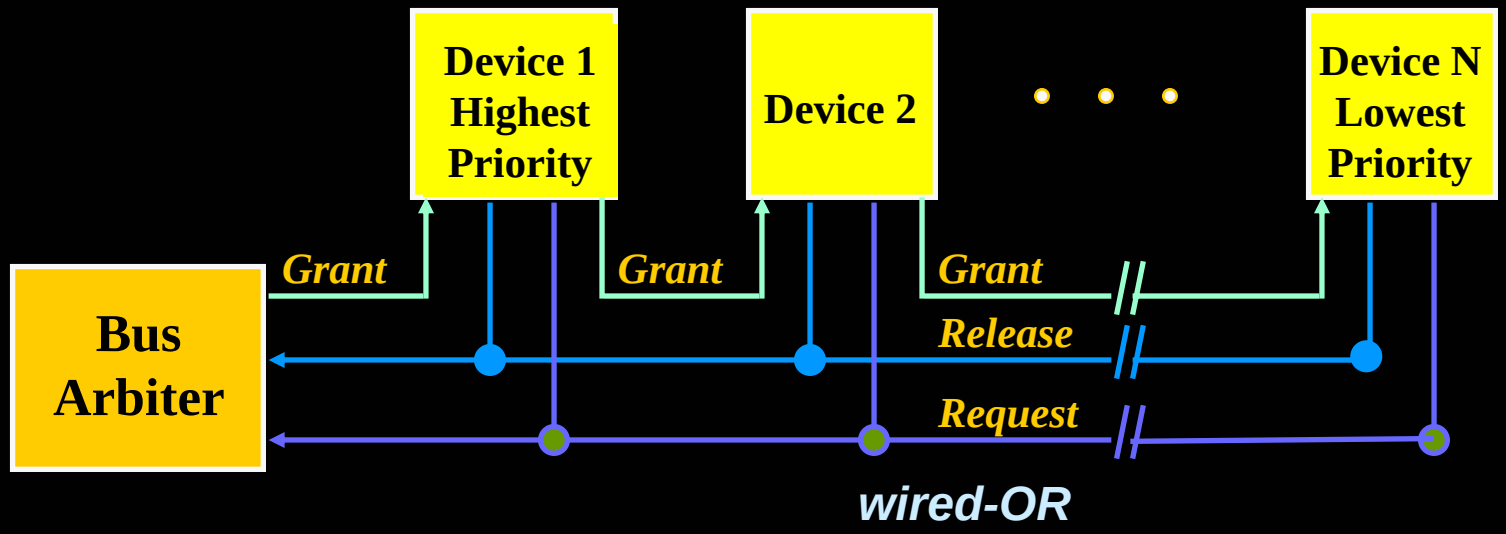
Barramento com múltiplos mestres: Necessidade de arbitragem

- **Esquema de arbitragem do barramento**
  - Um dos mestres do barramento ao querer usar o barramento sinaliza que quer o barramento
  - Um dos mestres do barramento não pode usar o barramento até que este lhe tenha sido cedido
  - O mestre do barramento sinaliza ao árbitro do barramento que terminou de utilizar o barramento
- **O esquema de arbitragem da utilização do barramento procura balancear dois factores:**
  - Prioridade no uso do barramento
    - O dispositivo com maior prioridade deve ser servido primeiro
  - Equidade
    - Mesmo o dispositivo com menor prioridade nunca deve ser completamente impedido de usar o barramento

Barramento com múltiplos mestres: Necessidade de arbitragem

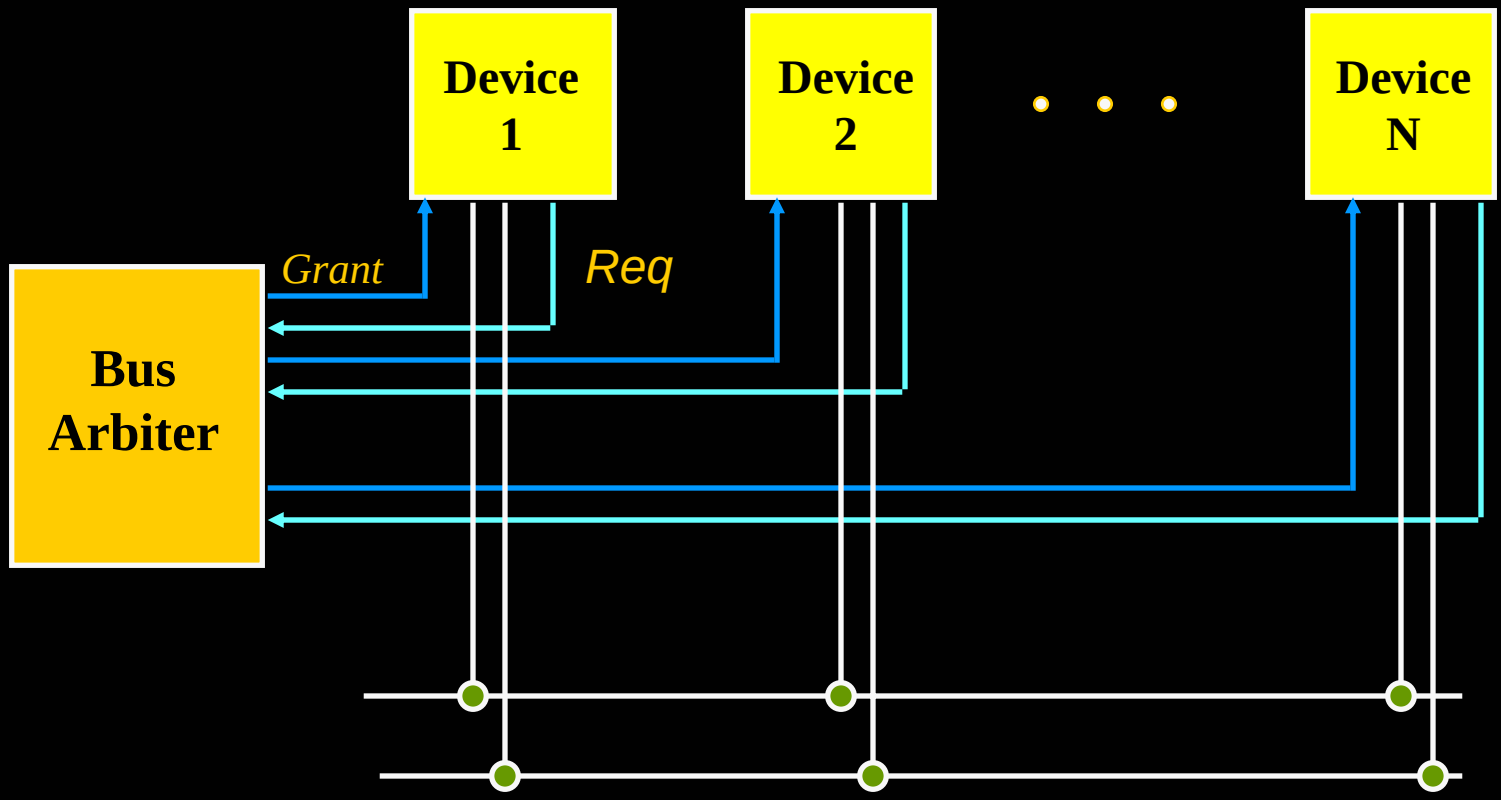
- **Os esquemas de arbitragem do barramento podem ser divididos em quatro grandes classes:**
  - Arbitragem em *daisy chain*
  - Arbitragem paralela centralizada
  - Arbitragem distribuída por auto-selecção:
    - Cada dispositivo que queira o barramento coloca um código no barramento indicando a sua indentidade
  - Arbitragem distribuída por detecção de colisão:
    - Cada dispositivo “simplesmente avança”.
    - Qualquer problema é resolvido a posterior.

Esquema de arbitragem: Arbitragem por *Daisy Chain*



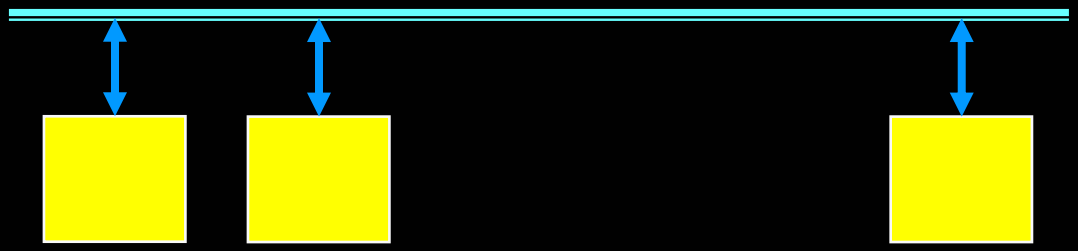
- **Vantagem:**
  - Simplicidade
- **Desvantagem:**
  - Não pode assegurar equidade no acesso ao barramento
    - Um dispositivo de menor prioridade pode não conseguir o acesso ao barramento
  - O uso do *daisy chain* também limita a velocidade do barramento

Esquema de arbitragem: Arbitragem paralela centralizada



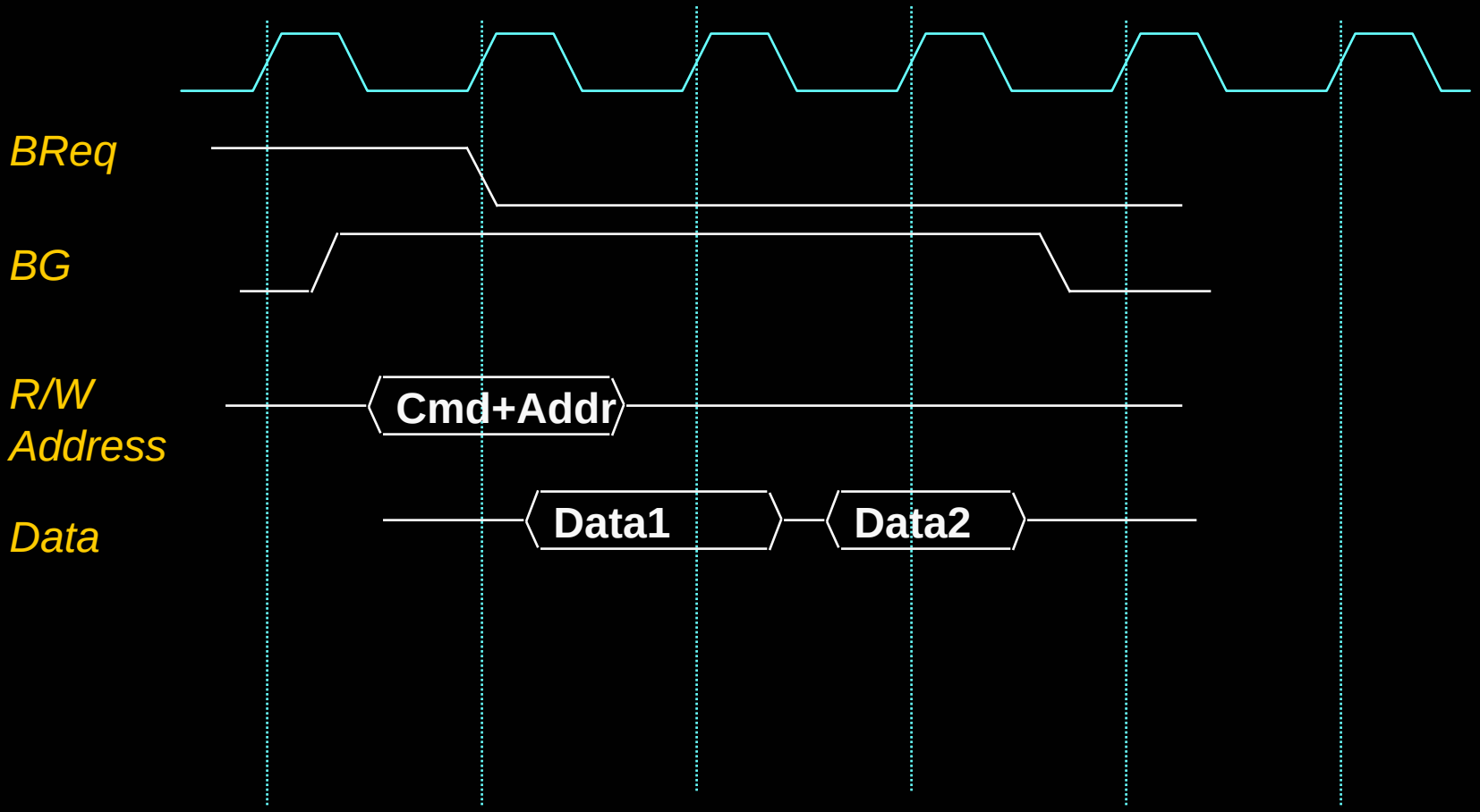
- **É usado praticamente em todos os barramentos processador-memória e em barramentos I/O de elevado desempenho**

Paradigma mais simples de arbitragem do barramento



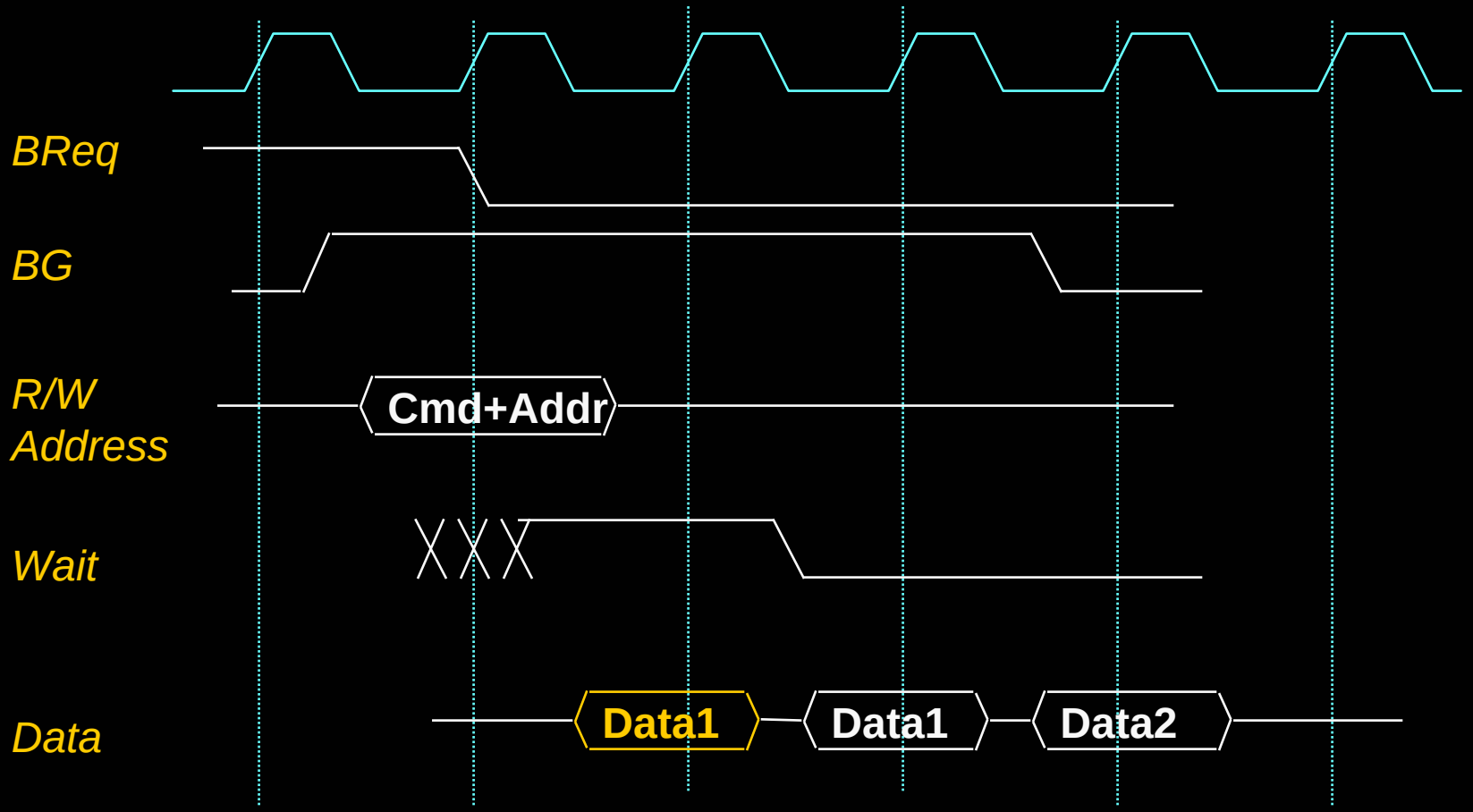
- **Todos os agentes operam sincronamente**
- **Todos podem transaccionar dados à mesma taxa**
- **Trata-se do protocolo mais simples**
  - Apenas tem-se que gerir o dispositivo fonte e o destino

Forma mais simples do protocolo síncrono



- **O barramento de memória geralmente é mais complexo do que este**
  - A memória (dispositivo escravo) pode precisar de tempo para responder

Típico protocolo síncrono



- **O dispositivo escravo indica quando está preparado para a transferência de dados**
- **A taxa de transferência é a do barramento**

Como aumentar a largura de banda do barramento ?

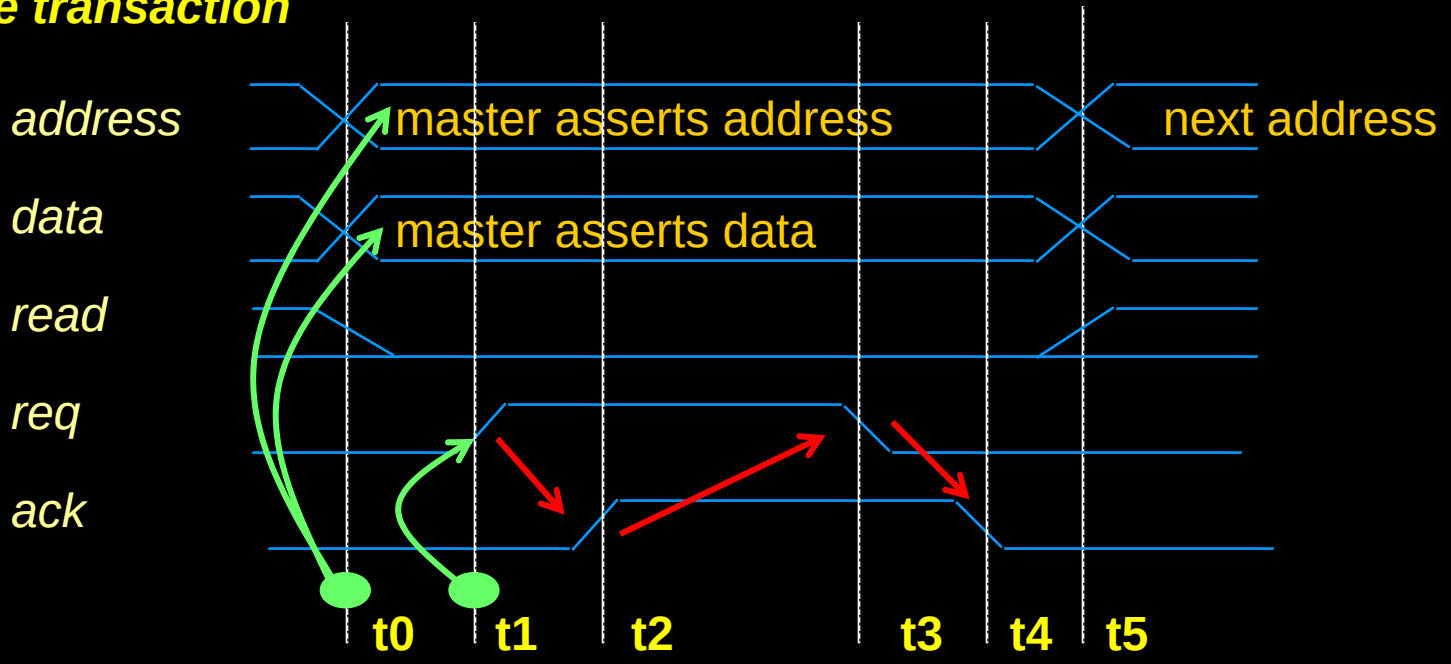
- **Linhas de endereço e de dados: *separadas ou multiplexadas***
  - Os endereços e os dados podem ser transmitidos num único ciclo do barramento se as linhas de endereço e de dados forem distintas
  - Custo
    - mais linhas no barramento
    - aumento da complexidade do barramento
- **Largura do barramento de dados**
  - Ao aumentar a largura do barramento de dados, a transferência de múltiplas palavras requererá menos ciclos do barramento
  - Exemplo:
    - SPARC Station 20: o barramento de dados tem uma largura de 128 bits

Como aumentar a largura de banda do barramento ?

- **Transferência de blocos de dados**
  - Permite que o barramento transfira múltiplas palavras em ciclos
  - Apenas é necessário enviar um endereço no início
  - O barramento não é libertado até que a última palavra tenha sido transferida
  - Custo:
    - Aumento da complexidade
    - Diminuição do tempo de resposta por pedido

Handshake para protocolo assíncrono (4 fases)

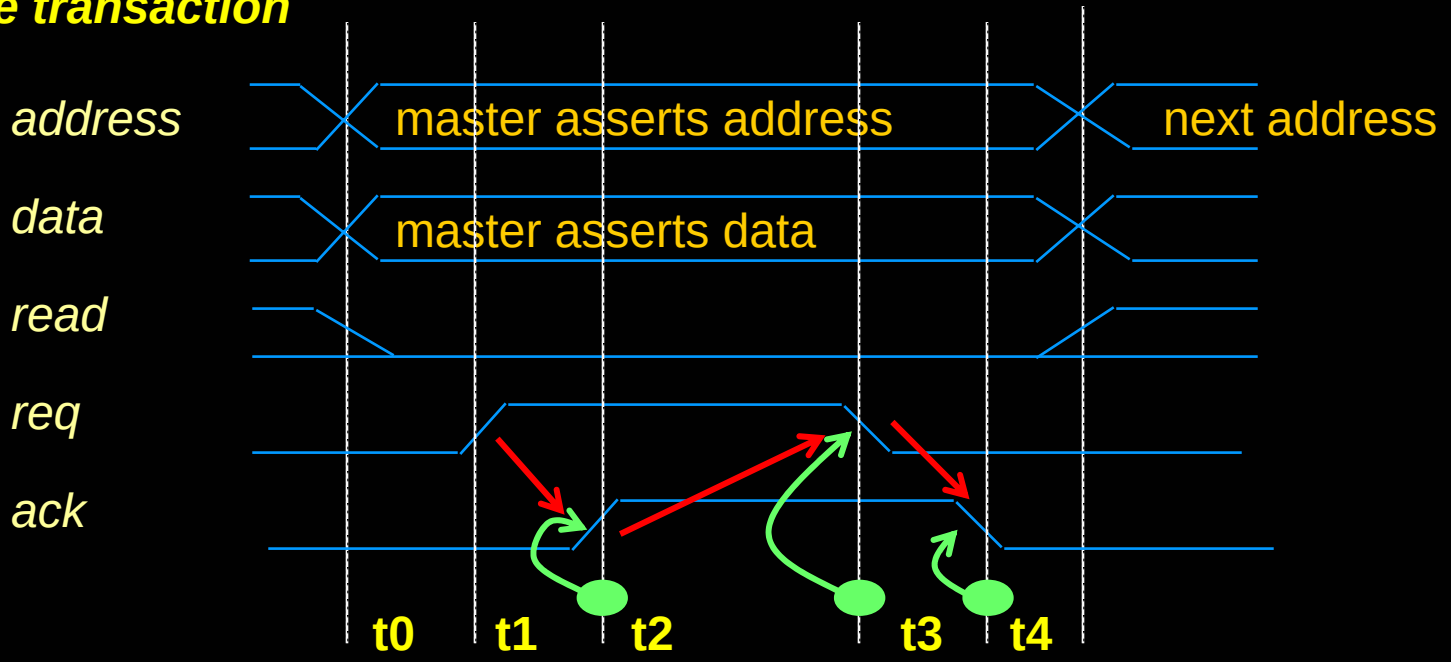
Write transaction



- **t0:** O ***master*** obteve o controlo e emite o endereço, a direcção e os dados. Espera durante um determinado tempo a fim de que os ***slaves*** descodifiquem o destinatário
- **t1:** O ***master*** activa a linha de '***req***' ('*resquest*')

Handshake para protocolo assíncrono (4 fases)

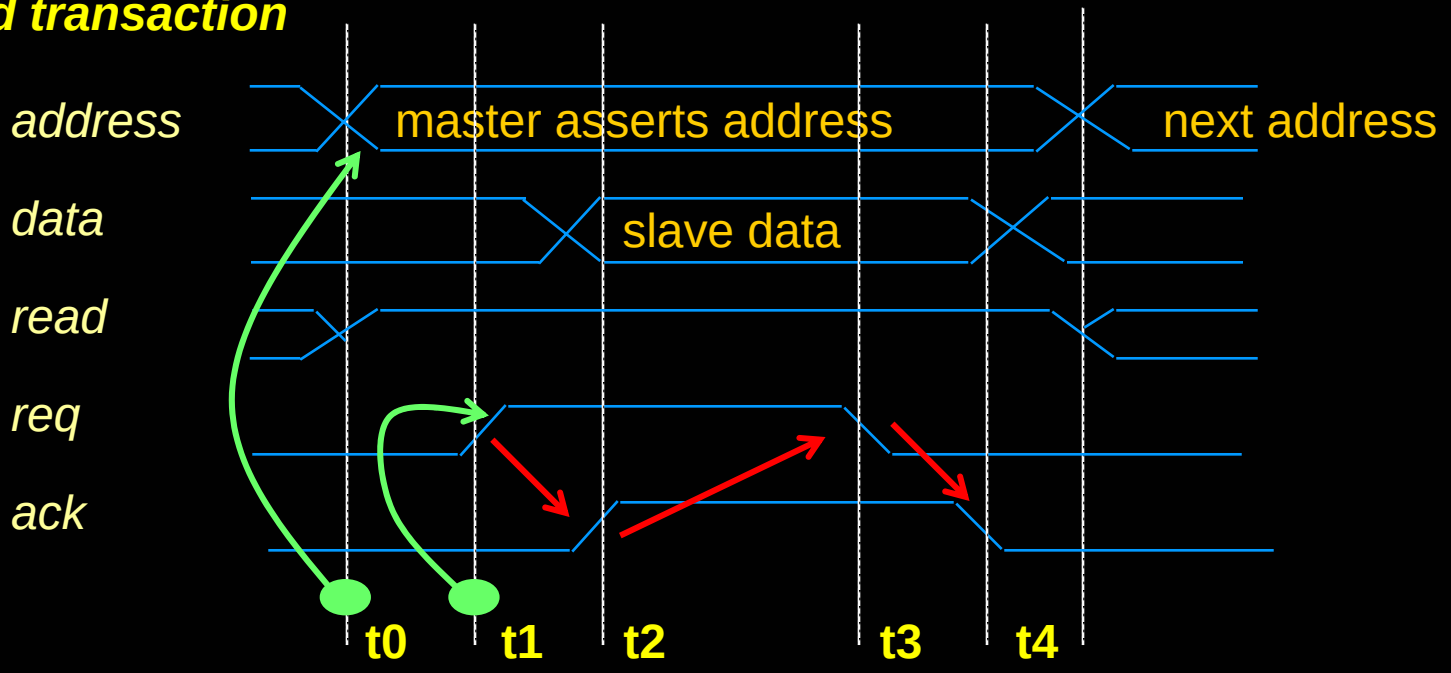
Write transaction



- t2: O ***slave*** activa a linha '***ack***' ('*acknowledge*'), indicando a recepção dos dados
- t3: O master desactiva a linha de '***req***' ('*request*')
- t4: O slave desactiva a linha de '***ack***' ('*acknowledge*')

Transacção na leitura

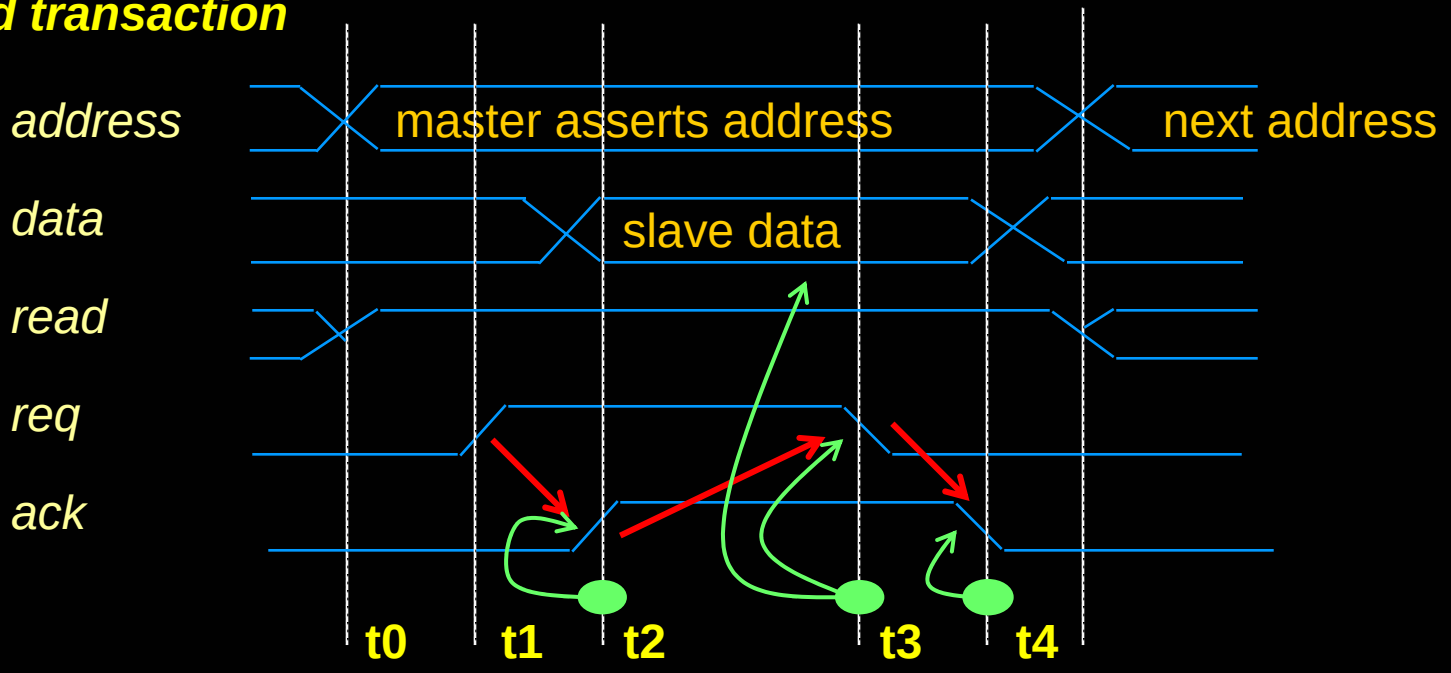
Read transaction



- **t0:** O ***master*** obteve o controlo e emite o endereço e a direcção. Espera durante um determinado tempo a fim de que os ***slaves*** descodifiquem o destinatário
- **t1:** O ***master*** activa a linha de '***req***' ('*resquest*')

Transacção na leitura

Read transaction



- t2**:** O ***slave*** activa a linha '***ack***' ('*acknowledge*'), indicando estar pronto para transmitir dados
- t3: O ***master*** desactiva a linha de '***req***' ('*request*'), os dados são recebidos
- t4: O ***slave*** desactiva a linha de '***ack***' ('*acknowledge*')