Barramentos

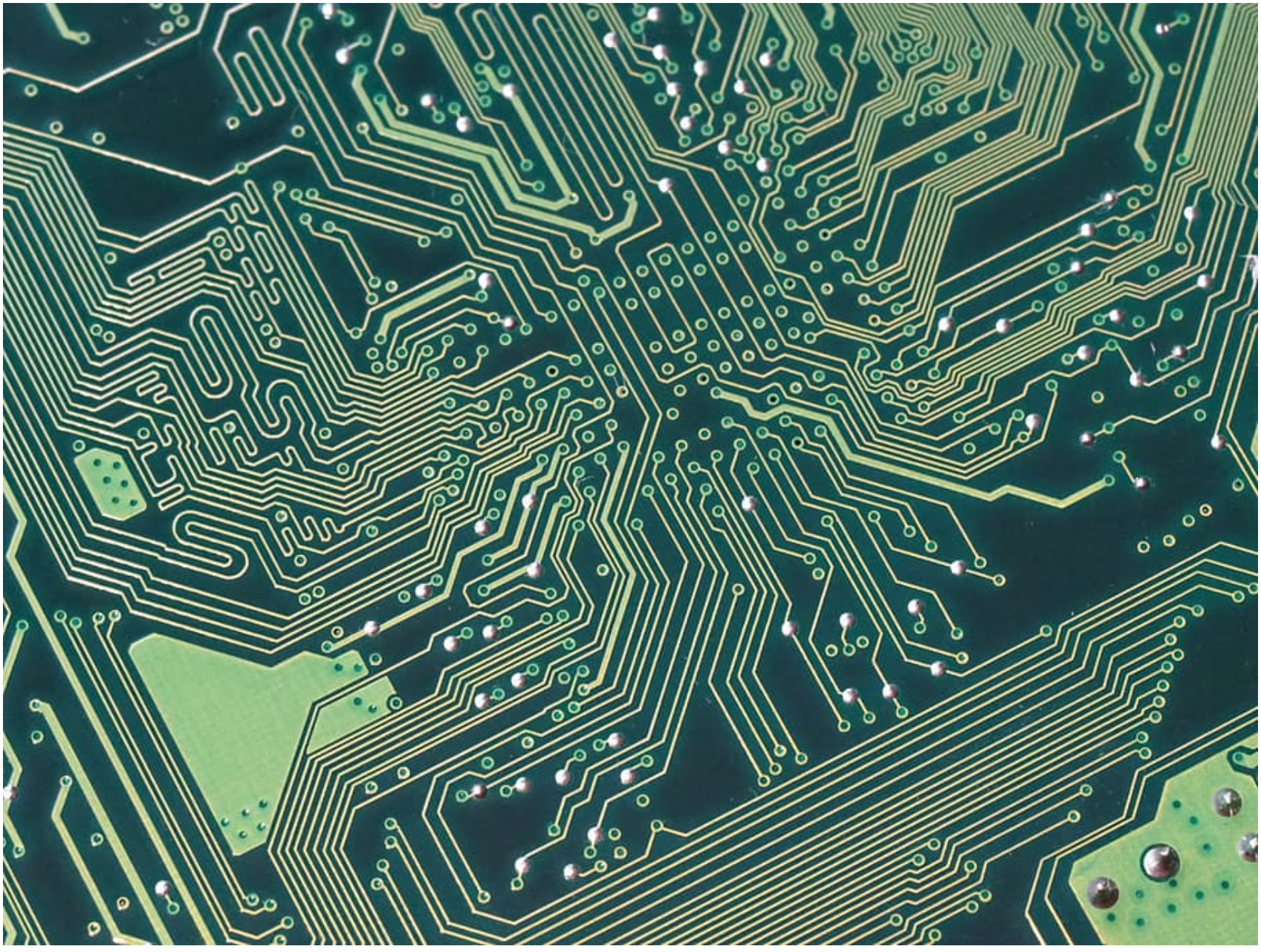
Barramentos, Arbitragem, Entrada e Saída, AMBA

Versão para Engenharia de Computação

PCS - Departamento de Engenharia de Computação e Sistemas Digitais
Escola Politécnica da Universidade de São Paulo

10 de Junho de 2024

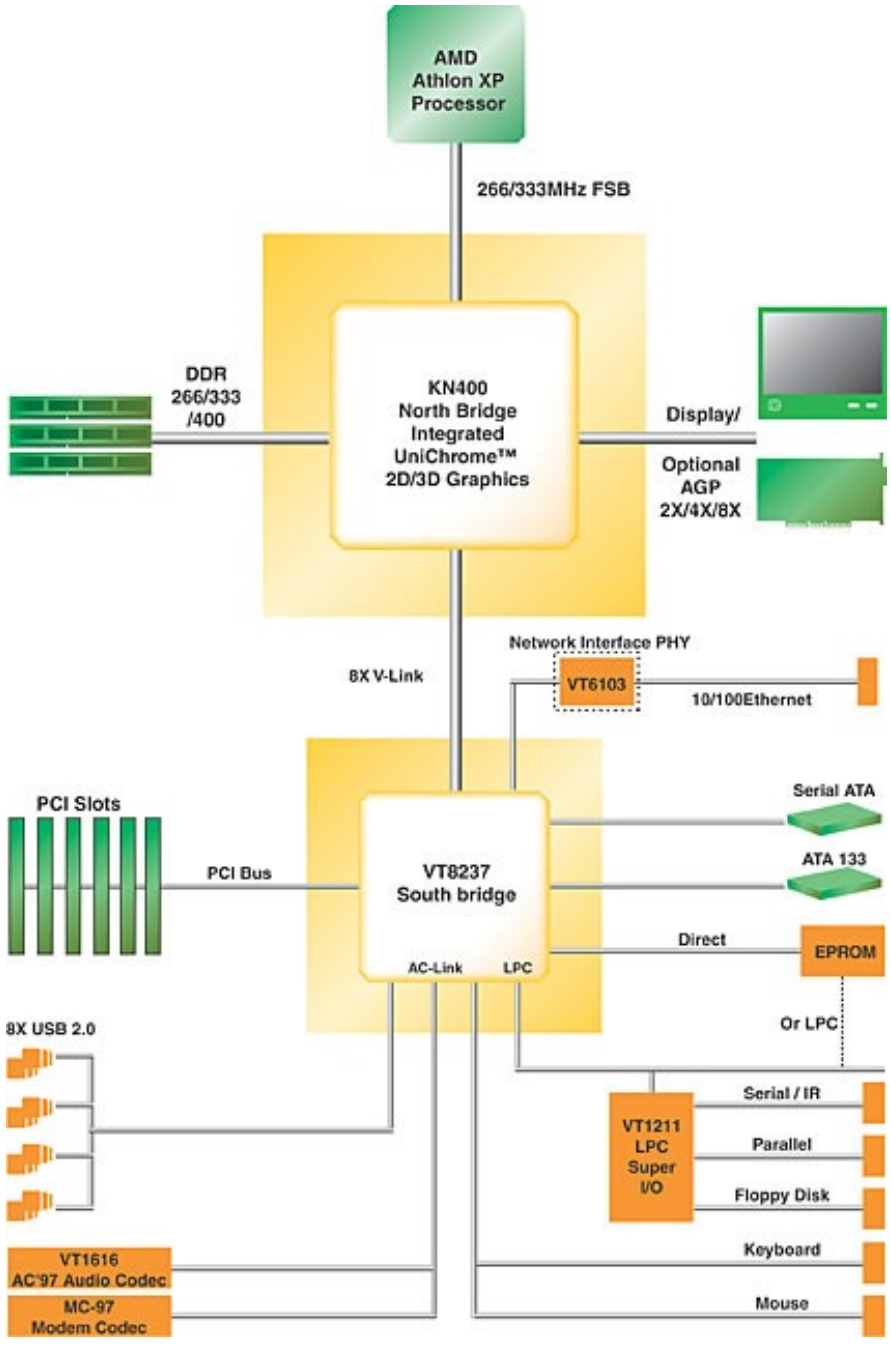
Barramentos



Definição de Barramento

- Um **barramento** é um sistema de comunicação entre componentes em um sistema digital.
- Inclui hardware (fios, cabos, fibra ótica, etc.) e software (protocolos de comunicação).
- Em um computador, conecta o processador à memória e aos periféricos.
- Um barramento tipicamente conecta vários componentes.

Exemplo: North & South Bridge

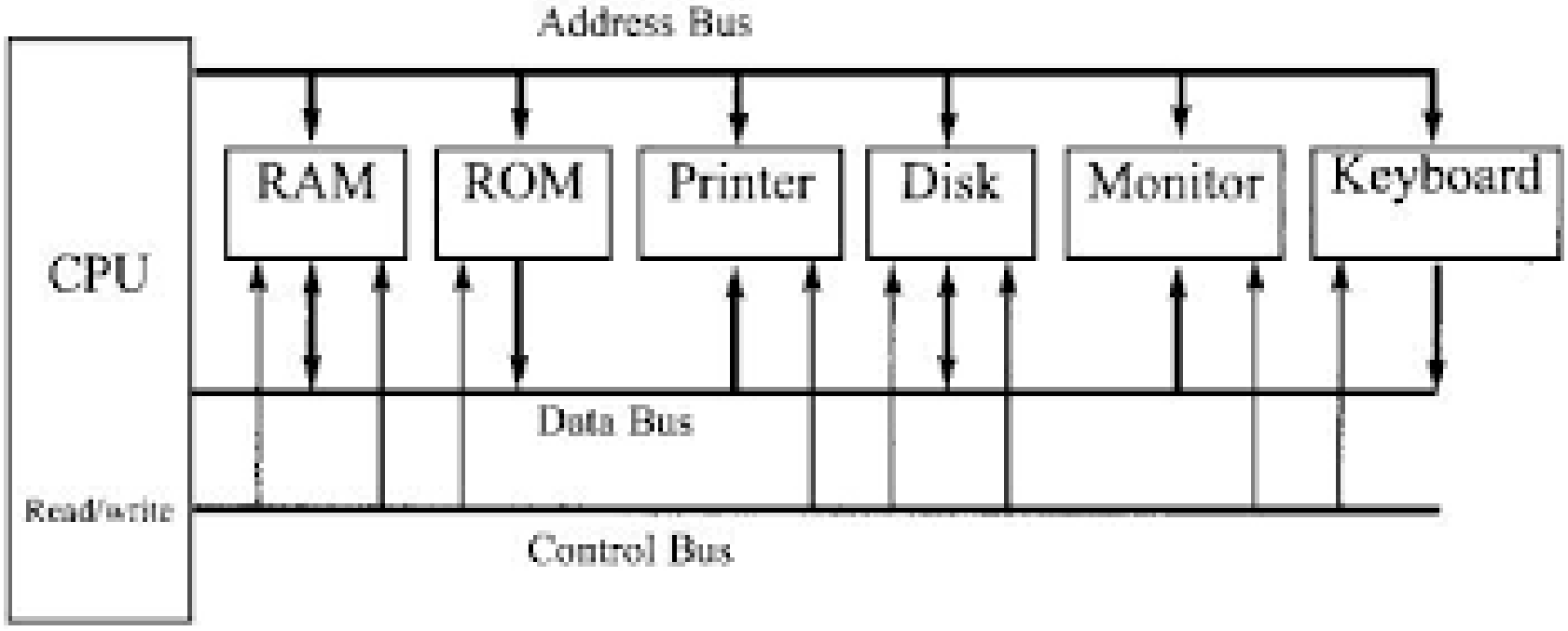


Ativo e Reativo

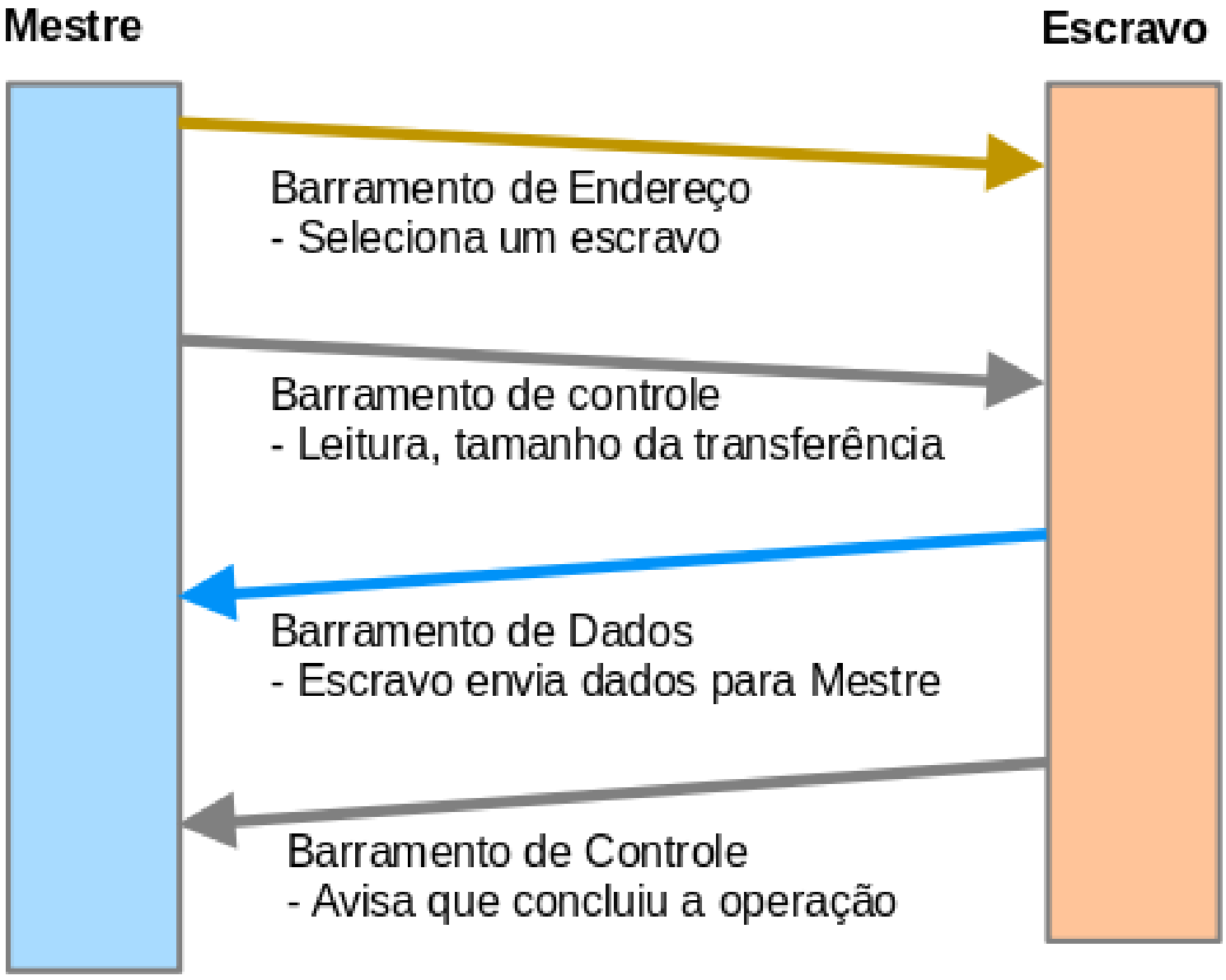
- Os dispositivos que se comunicam por um barramento são classificados como **ativo** ou **reativo**.
- Toda transação envolve um ativo e um reativo.
- Ativo inicia a transação, escolhendo o reativo.

Tipos de Barramento

- **Barramento de Dados:** carrega os dados a serem transferidos.
- **Barramento de Endereço:** usado para o ativo selecionar um reativo.
- **Barramento de Controle:** identifica a transação (read/write), seu modo (simples, rajada) e seu status (em andamento, erro, etc).

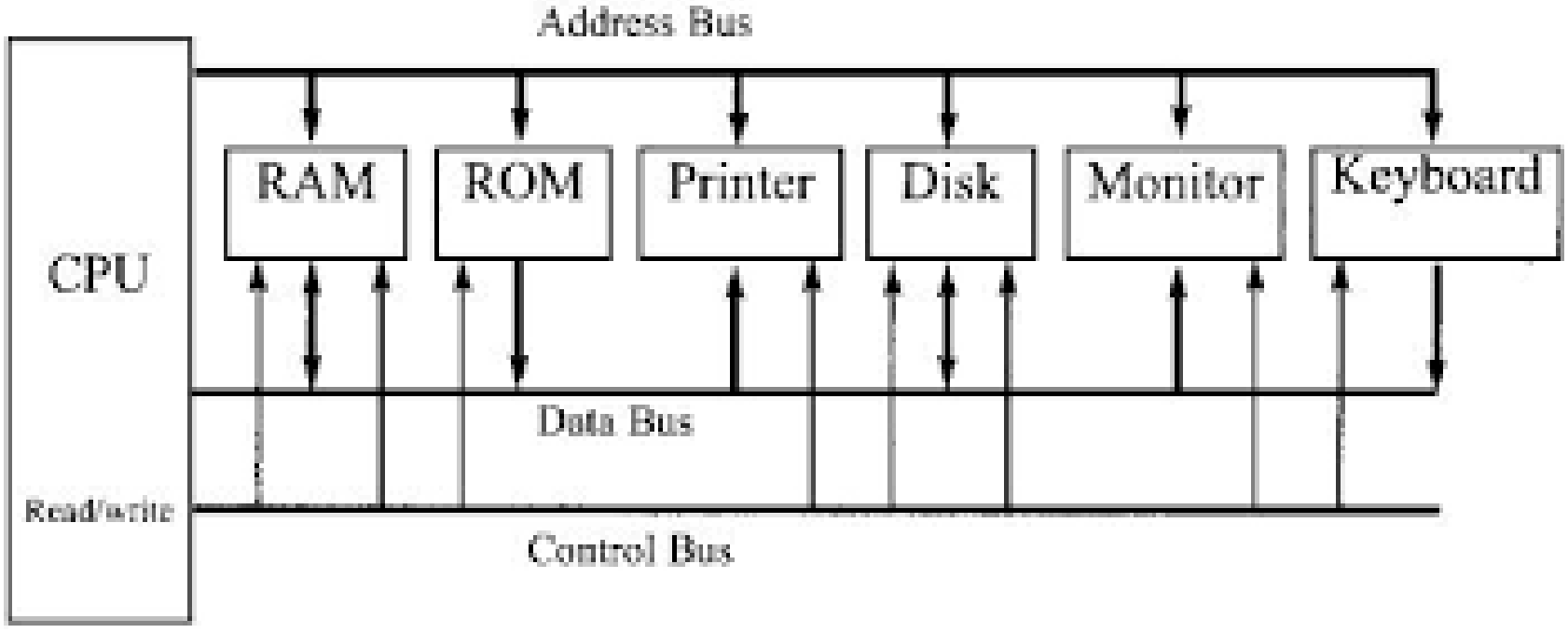


Uma Típica Operação



Uma Típica Operação

```
1  LW X10, 0x0000002C(X0)
2  SW X10, 0x0000002C(X0)
```

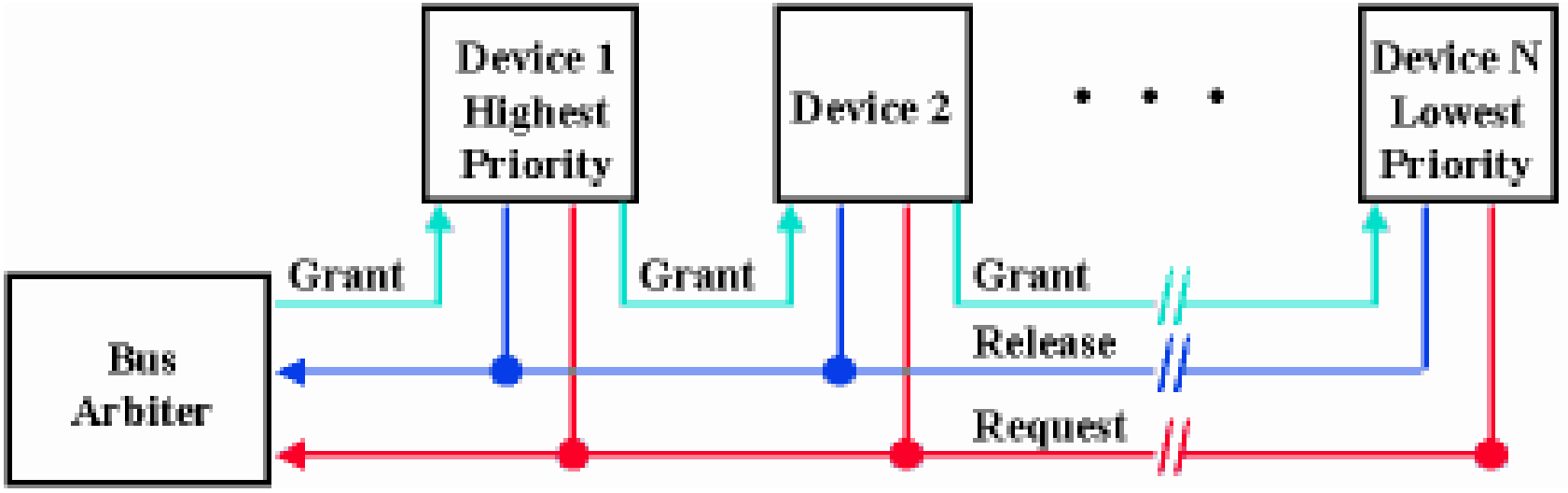


Múltiplos Ativos

- Para um periférico se comunicar diretamente com a memória, sem passar pela CPU, podemos incluir outros ativos no barramento (e.g. DMA).
- Quando há mais de um ativo, apenas um pode usar o barramento em um dado momento.
- A coordenação para que isso occorra é chamada de **arbitragem**.
  - Centralizada: Um único árbitro decide sozinho
  - Distribuída: Todos os ativos participam da decisão

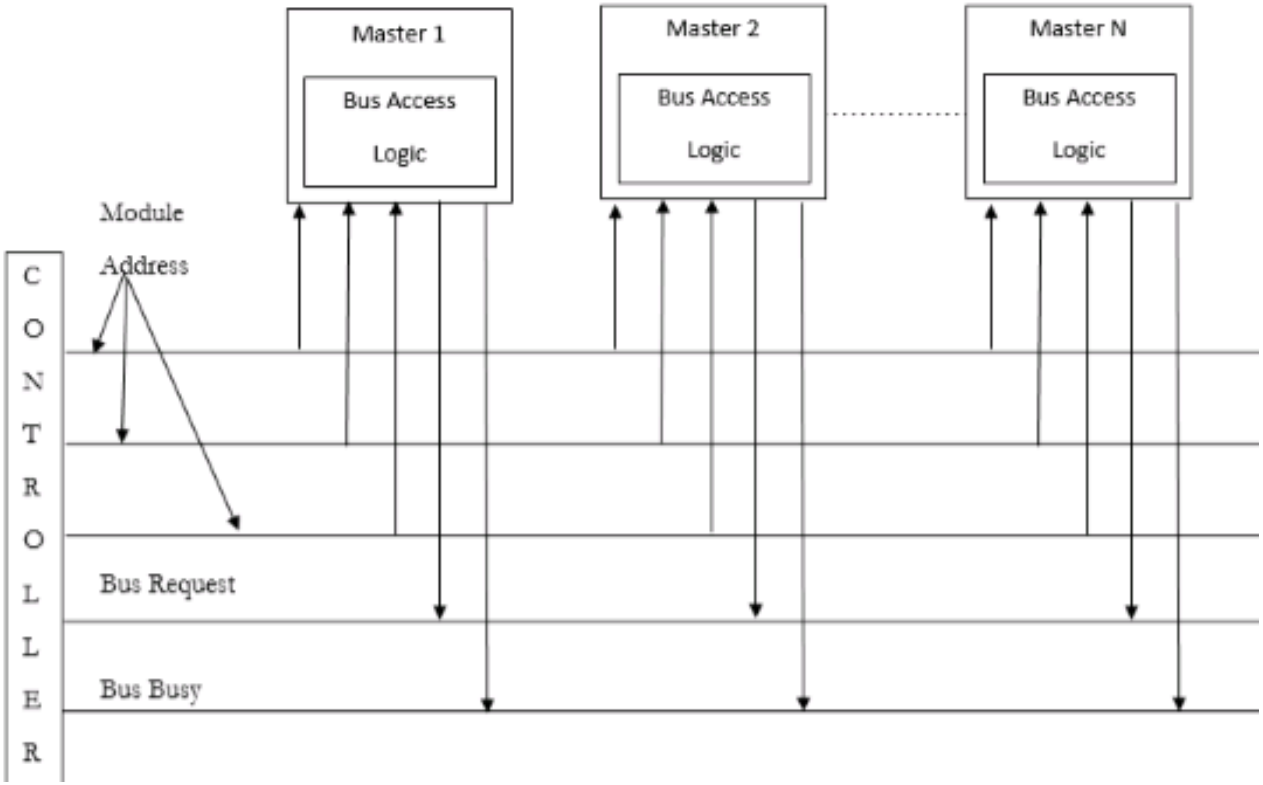
Esquema Daisy Chain de Arbitragem

- Árbitro recebe requisição de controle (Request).
- Controle concedido ao primeiro dispositivo (Grant), que passa adiante se não requisitou.
- Dispositivo que requisitou toma controle se recebe Grant e ativa o sinal Busy.



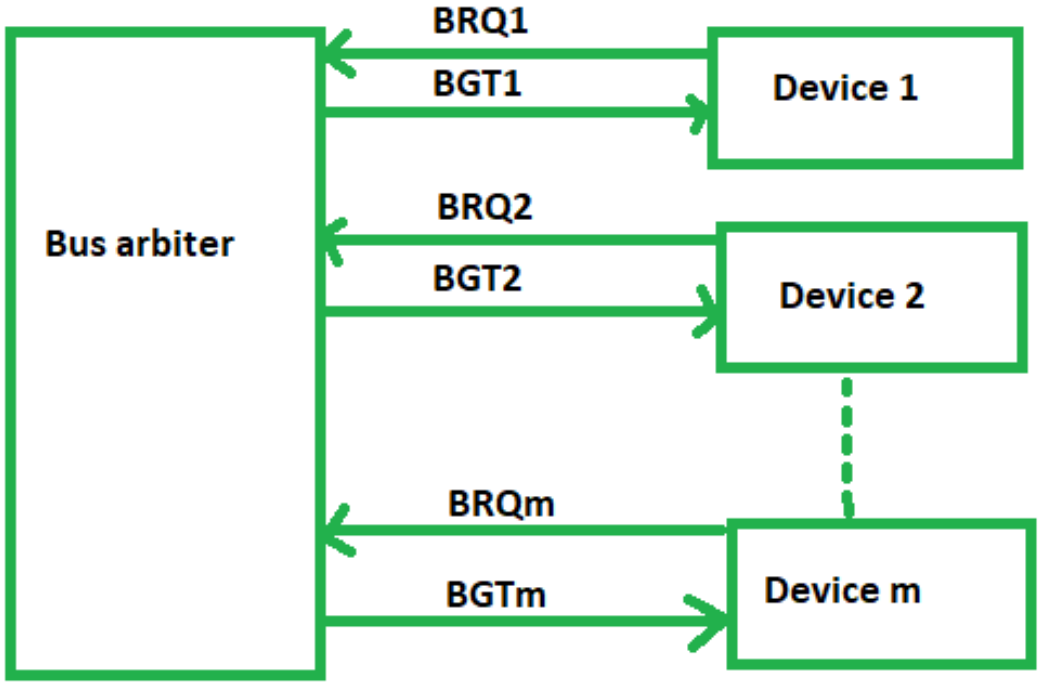
Arbitragem com Polling

- Ao receber um Request, árbitro gera sequência de endereços, variando a ordem.
- Dispositivo do Request toma controle quando reconhece seu endereço.



Arbitragem com Requisições Independentes

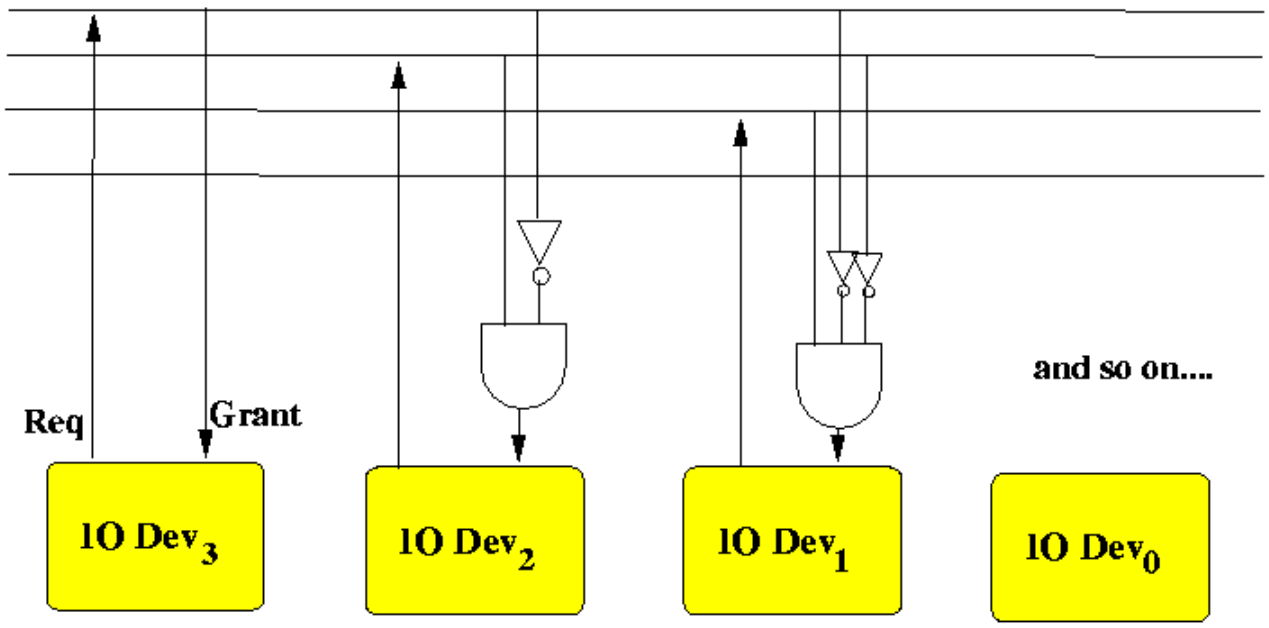
- Um Grant e um Request por dispositivo, prioridade fixa.
- Reposta mais rápida, porém mais fios.



Fixed priority bus arbitration method

Arbitragem Descentralizada

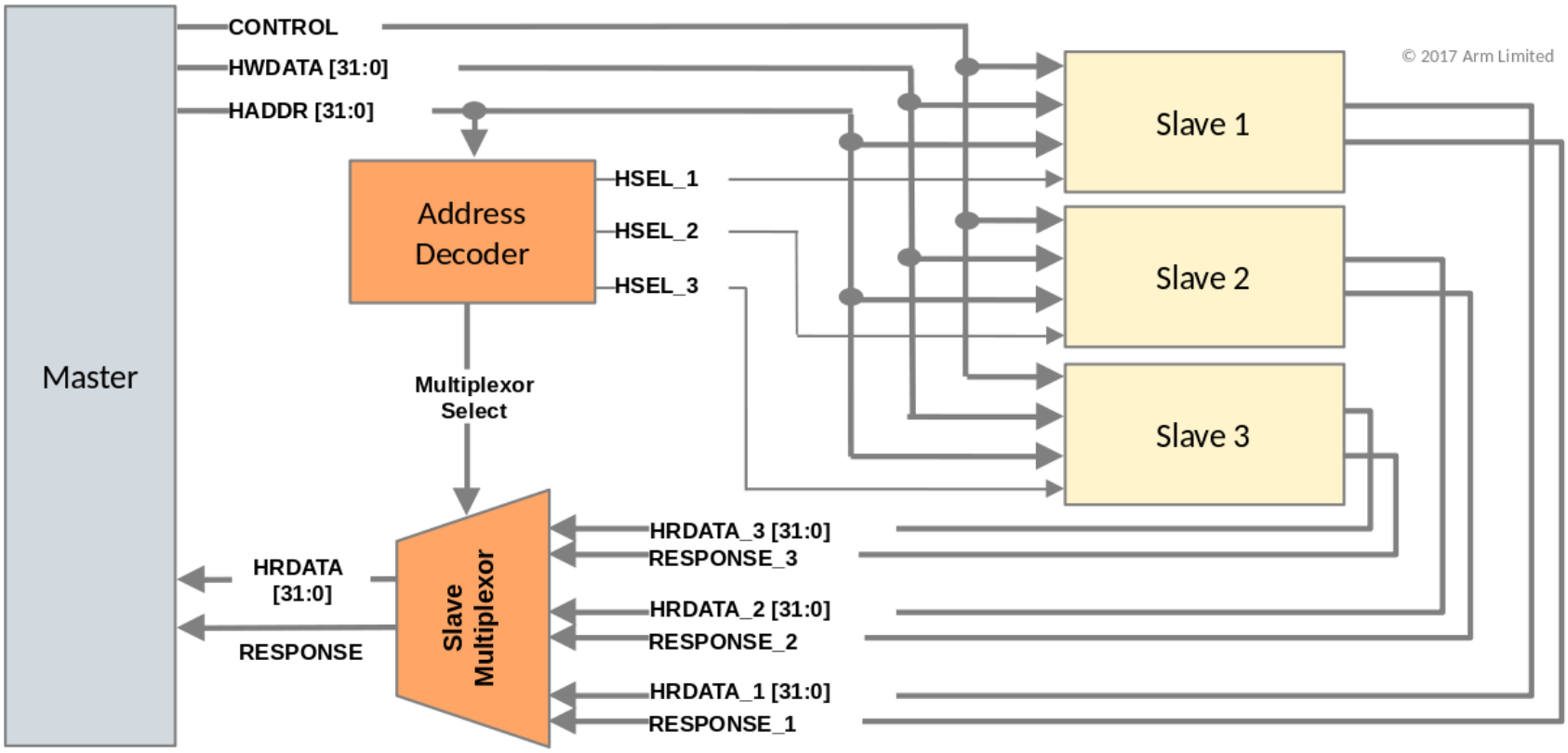
- Um Request por dispositivo, prioridade fixa.
- Cada dispositivo monitora os resquests mais priotários para gerar seu Grant.



Exemplo de Barramento

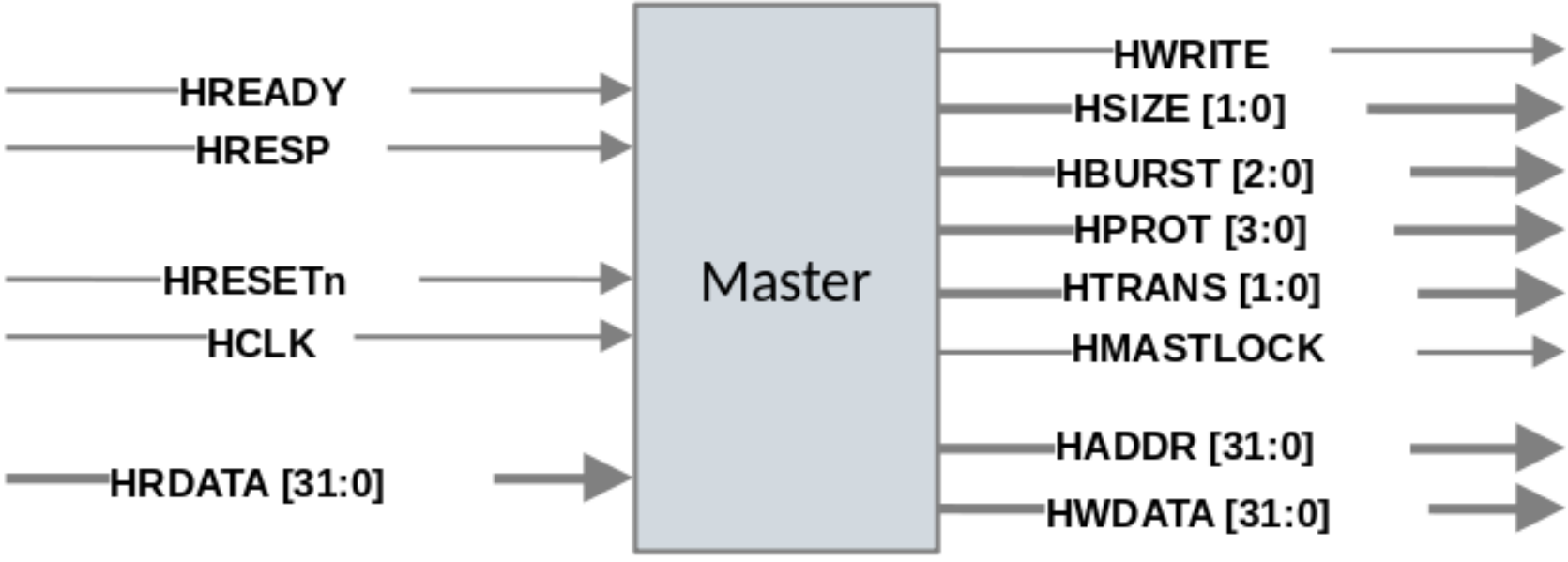
- Para ilustrar o funcionamento de um barramento, consideremos o AMBA 3 AHB-Lite.
- AMBA (Advanced Microcontroller Bus Architecture) é um conjunto de especificações de barramento, padrão aberto da ARM, muito usado na prática.
- AHB (Advanced High-performance Bus) foca em projetos de alto desempenho, fornecendo operações com alta largura de banda.
- AHB-Lite é um subconjunto do AHB, de projeto mais simples.
  - É sugerido para o cenário com apenas um ativo, mas pode suportar mais.

Diagrama de Blocos



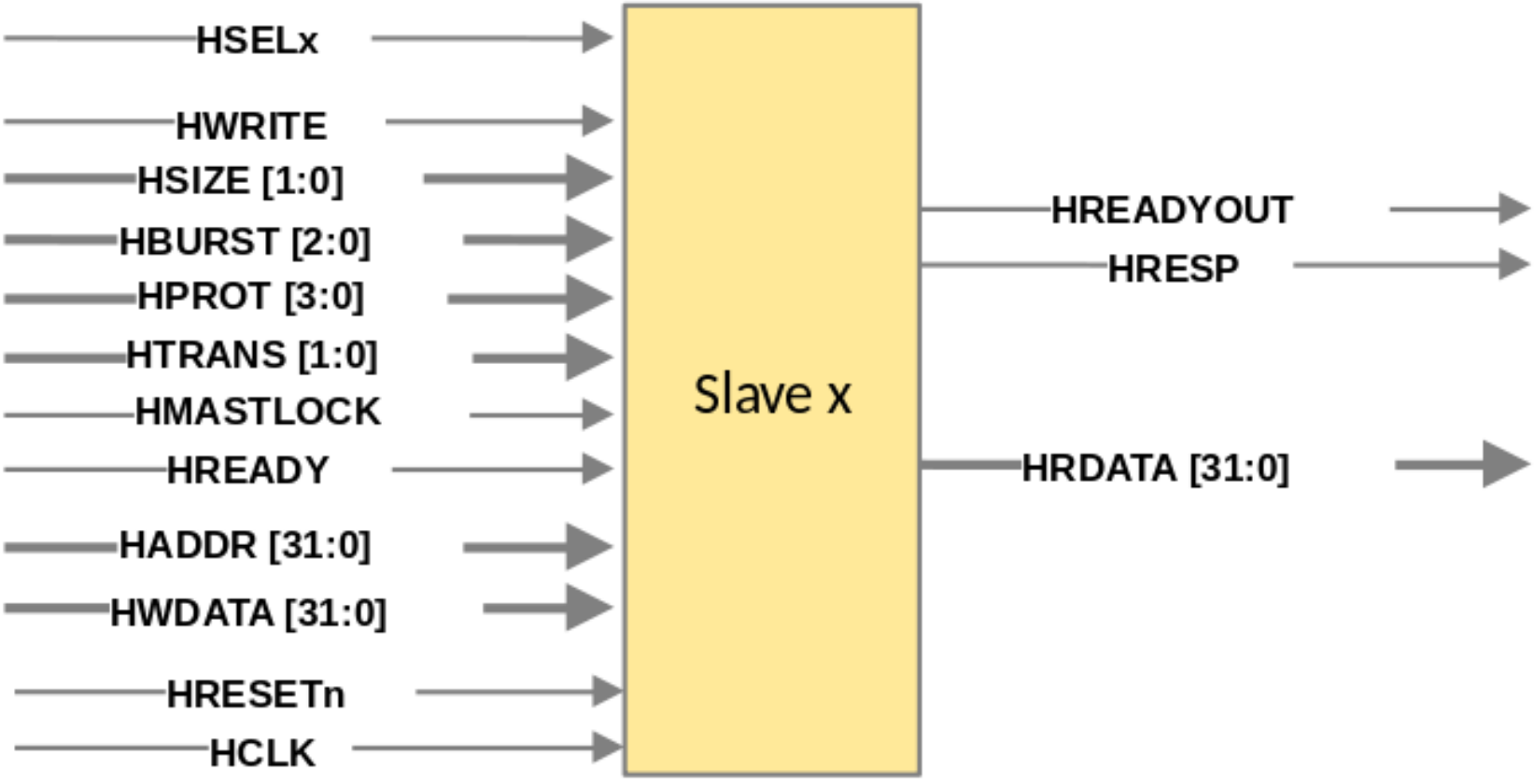
Interface do Ativo

- HRDATA, HWDATA e HADDRESS são os dados lidos, escritos, e o endereço.
- HREADY e HRESP vêm do reativo selecionado.
- HBURST especifica o modo de rajada/simples.
- HSIZE diz o tamanho do dado (1 a 128 bytes).



Interface do Reativo

- SEL é o sinal de seleção.
- HREADYOUT é 1 se o reativo está pronto.
  - HREADY é HREADYOUT do reativo selecionado.
- HRESP diz se houve erro ou não.

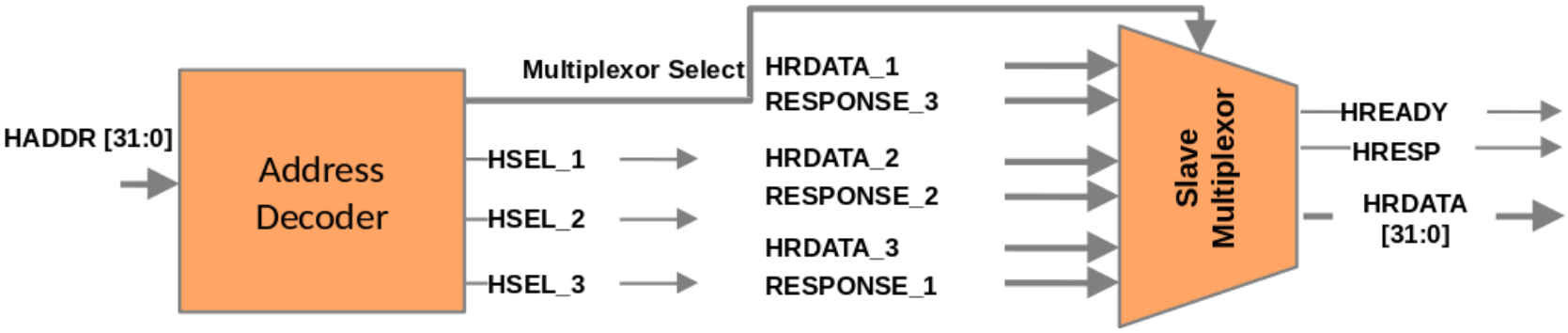


Decodificador de Endereço

- Cada reativo tem um sinal de seleção SEL que corresponde a sua faixa de endereços.
- O **decodificador de endereço** ativa o SEL correspondente ao endereço em sua entrada.
- Outra saída do decodificador controla o Multiplexador, para passar os dados do reativo selecionado.

Multiplexador dos Reativos

- O **Multiplexador** passa para o ativo os dados (na leitura) e os sinais de controle do reativo selecionado.
- HREADY e HRESP são o RESPONSE (HREADYOUT e HRESP) do reativo selecionado.
- Controle do Multiplexador vem do decodificador.



Modos de Operação

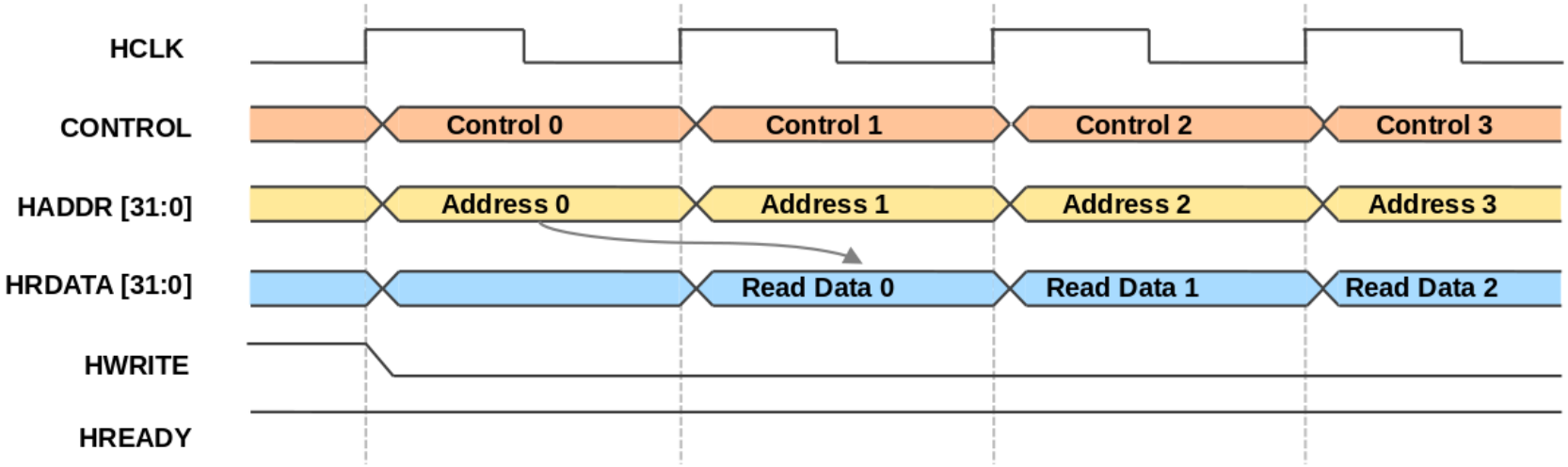
- O AHB-Lite suporta dois grandes tipos de transferência (leitura ou escrita).
- **Simples**: especifica-se um endereço e faz-se uma leitura/escrita.
- **Rajada**: Especifica-se uma série de endereços em um mesmo reativo, correspondendo a uma série de leituras/escritas.

Fases da Transferência

- Cada transferência consiste em duas fases:.
  - **Fase de Endereço**: Ativo especifica o endereço.
  - **Fase de Dados**: No clock seguinte à fase de endereços, os dados são transferidos.
- Fases em *pipeline*: uma transferência está na fase de dados enquanto a seguinte está na fase de endereços.
- Na fase de dados, se HREADY=0, é necessário aguardar o próximo ciclo para transferência.
- Enquanto isso, Ativo entra em um estado de espera (*wait state*).

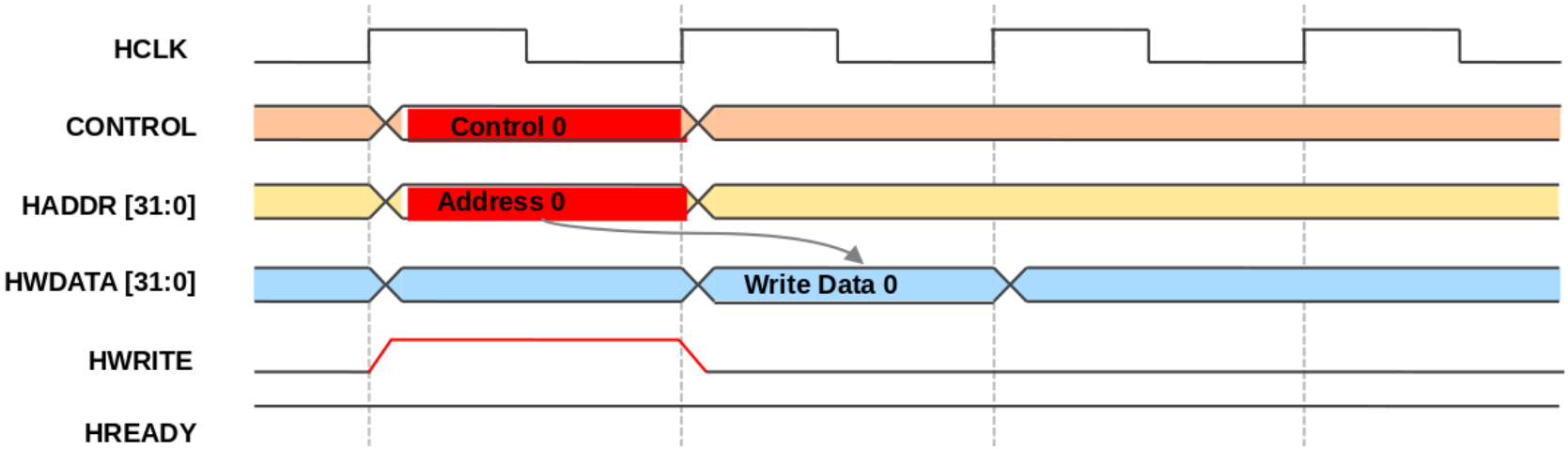
Exemplo de Leituras

- Considere uma sequência de leituras simples.
- Reativo responde HREADY=1 (sem *wait state*).



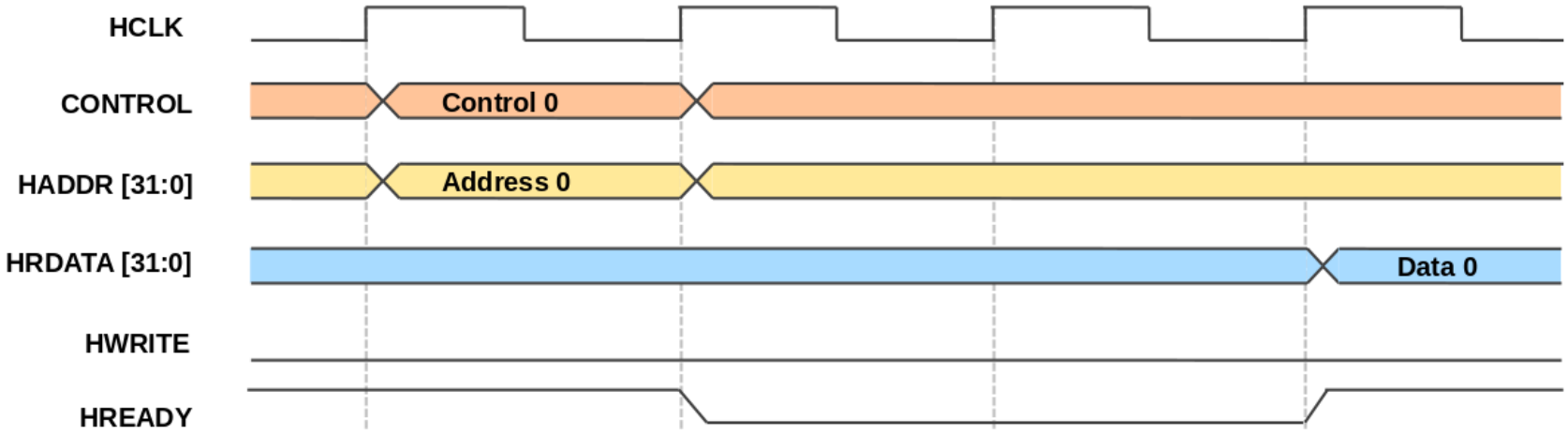
Exemplo de Escrita

- Considere uma escritas simples.
- Reativo responde HREADY=1 (sem *wait state*).



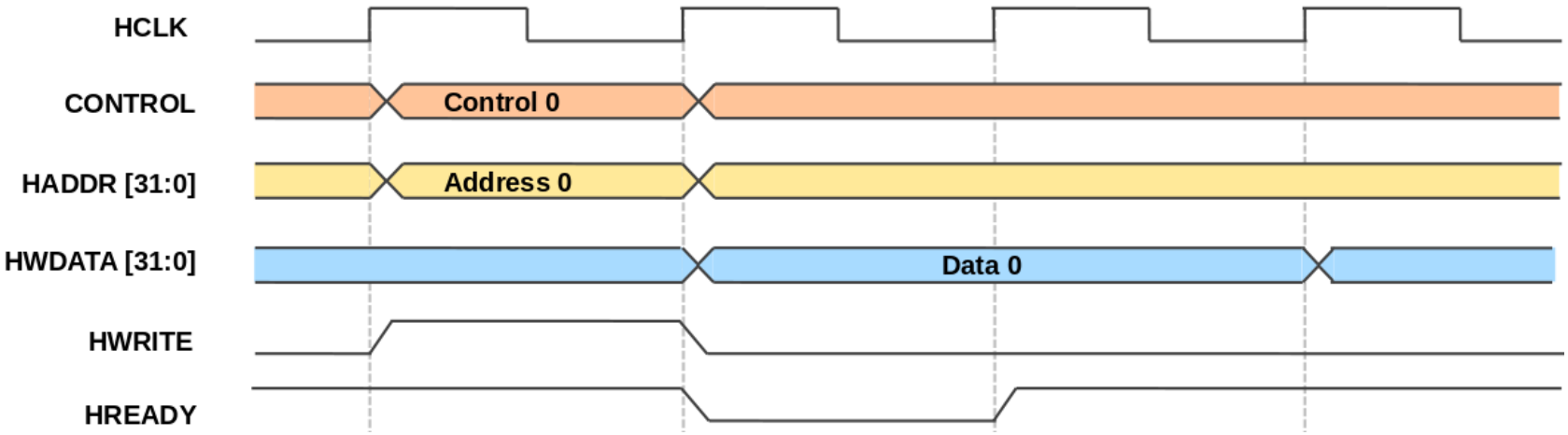
Leitura com Espera

- Se reativo responde HREADY=0, ativo espera (*wait state*), atrasando sua próxima transferência, até que HREADY=1.



Escrita com Espera

- Se reativo responde HREADY=0, ativo espera (*wait state*), atrasando sua próxima transferência, até que HREADY=1.



Rajada com Incremento

- Escrita em rajada, onde endereços são incrementados.