# Interconexão (Bus)

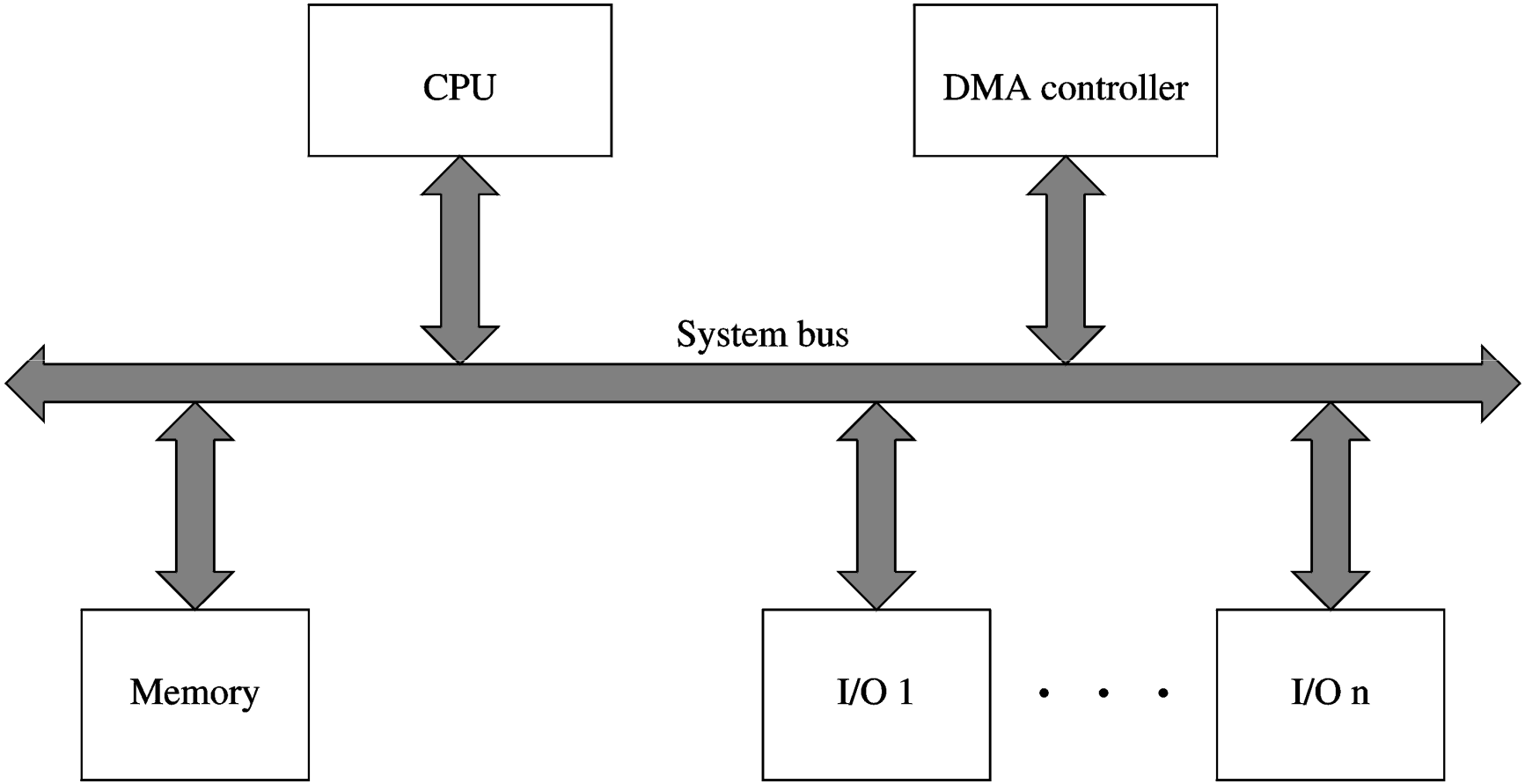
# Buses de Sistema

- Internos (foco deste capítulo)
  - ✓ PCI
  - ✓ AGP
  - ✓ PCMCIA
  - ✓ …
- Externos (mais tarde)
  - ✓ USB
  - ✓ FireWire
  - ✓ …

# Bus

- Operação
  - ✓ Exemplo: escrever um endereço no bus de endereços
- Transacção
  - ✓ O bus é partilhado – é necessário definir como é que os dispositivos ligados podem utilizá-lo
  - ✓ Transacção → sequência de operações para realizar uma dada actividade
    - ✓ Exemplos: leitura, escrita, burst de leitura/escrita
    - ✓ O *mestre* inicia a transacção – o *escravo* alvo responde
    - ✓ Por exemplo em operações sobre memória o CPU é o mestre a memória é o escravo
- Ciclo de Bus
  - ✓ Cada operação pode demorar vários ciclos

# Bus



# Bus

- Compostos por:
  - ✓ Bus de endereços
  - ✓ Bus de dados
  - ✓ Bus de controlo

- Podem ser:
  - ✓ Dedicados
  - ✓ Multiplexados
    - ✓ Por exemplo dados e endereços são comunicados através do mesmo bus
  - ✓ Síncronos
  - ✓ Assíncronos

# Bus de Controlo

- Sinais de controlo:
  - ✓ Leitura/escrita de/para memória
  - ✓ Leitura/escrita de/para controladores de I/O
  - ✓ Pronto (Ready) - Usado pelo escravo para informar o mestre que concluiu o processamento do pedido
  - ✓ Pedido de acesso ao Bus e sua aceitação
  - ✓ Envio de uma interrupção e notificação da sua recepção (a discutir aquando das aulas sobre I/O)
  - ✓ Pedido de operação de DMA e notificação da sua recepção (a discutir aquando das aulas sobre I/O)
  - ✓ Clock - Sincronização do bus
  - ✓ Reset - Inicializar o sistema

# Questões de Desenho

- Largura dos bus de endereços e dados
- Tipo de bus
  - ✓ Dedicado ou multiplexado
- Operações
  - ✓ Leitura, escrita, transferência de blocos, interrupções, …
- Arbitragem
  - ✓ Centralizada ou distribuída
- Temporização
  - ✓ Síncrona ou assíncrona

# Largura do Bus de Dados

- Crítico para a performance
- Não tem de corresponder ao que o ISA mostra ao programador
  - ✓ 8086 (1978) - registos de dados de 16 bits
    - ✓ Bus de dados: 16 bits
  - ✓ 8088 (1979) - registos de dados de 16 bits
    - ✓ Bus de dados: 8 bits
  - ✓ Pentium - registos de dados de 32 bits
    - ✓ Bus de dados: 64 bits
  - ✓ Core 2 - registos de dados de 64 bits
    - ✓ Bus de dados: 64 bits
  - ✓ Itanium - registos de dados de 64 bits
    - ✓ Bus de dados: 128 bits

# Largura do Bus de Endereços

- Determina a quantidade realmente acessível a partir de um programa
- Não tem de corresponder ao que o ISA mostra ao programador
  - ✓ 8086 (1978) - registos de endereços de 16 bits
    - ✓ Bus de endereços: 20 bits
  - ✓ Pentium - registos de endereços de 32 bits
    - ✓ Bus de endereços : 32 ou 48 bits
  - ✓ Core 2 - registos de endereços de 64 bits
    - ✓ Bus de endereços : 48 bits
  - ✓ Itanium - registos de endereços de 64 bits
    - ✓ Bus de endereços : 44 bits

# Tipo de Bus

- Um bus pode ser
  - ✓ Proprietário
    - ✓ Coberto por uma patente e não acessível a outros fabricantes
  - ✓ Normalizado
    - ✓ Obedece a uma especificação publicada;
    - ✓ Permite a inter-operação de equipamentos de diferentes fabricantes , ou seja, permite que no bus existam dispositivos de diferentes fabricantes
  - ✓ A norma define:
    - ✓ As especificações eléctricas exactas
    - ✓ As temporizações dos sinais
    - ✓ A forma como os dados são codificados
    - ✓ Fichas para ligar novos periféricos quando o bus permite extensões

# Tipo de Bus

- Bus dedicado
  - ✓ Buses separados para dados e endereços
  - ✓ Melhor performance
  - ✓ Mais caro

- Bus multiplexado
  - ✓ Bus partilhado para a comunicação de dados e endereços
  - ✓ Melhor utilização
  - ✓ Mais barato

# Operações

- Básicas
  - ✓ Leitura e escrita
- Transferência de blocos
  - ✓ Operar sobre uma sequência de posições de memória contíguas
  - ✓ Exemplo: Carregar uma linha (sequência de posições de memória contíguas) da memória central para uma memória cache
- Leitura-Escrita
  - ✓ Útil para operações atómicas / regiões críticas
- Interrupções
- Etc.

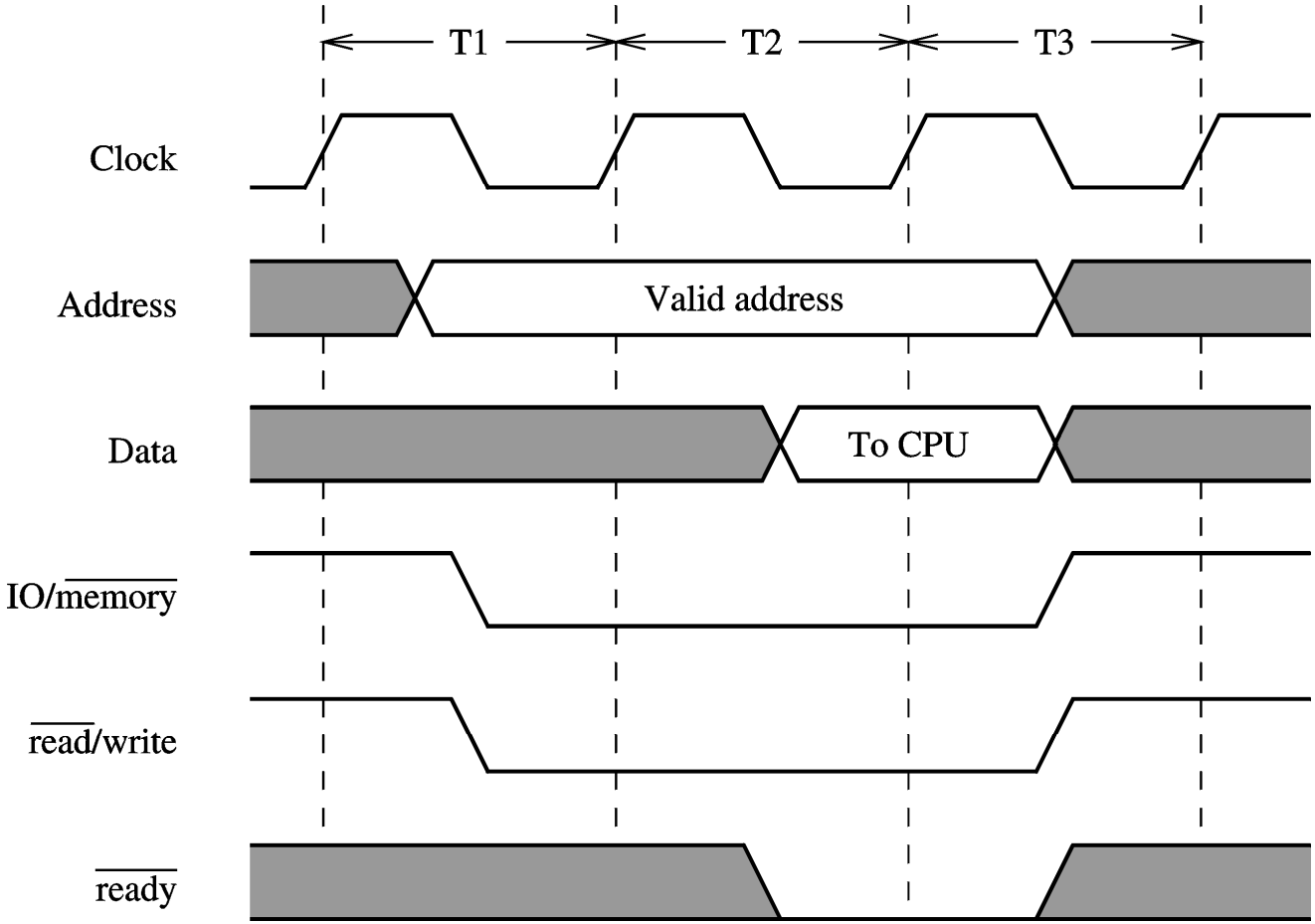
# Bus Síncrono

- Um sinal de relógio sincroniza de forma centralizada todas as operações sobre o bus
  - ✓ Sincronia dada pela subida e descida do sinal de relógio
  - ✓ Fácil de implementar
  - ✓ A maior parte dos buses são síncronos

- As operações podem ou não suportar ciclos de espera

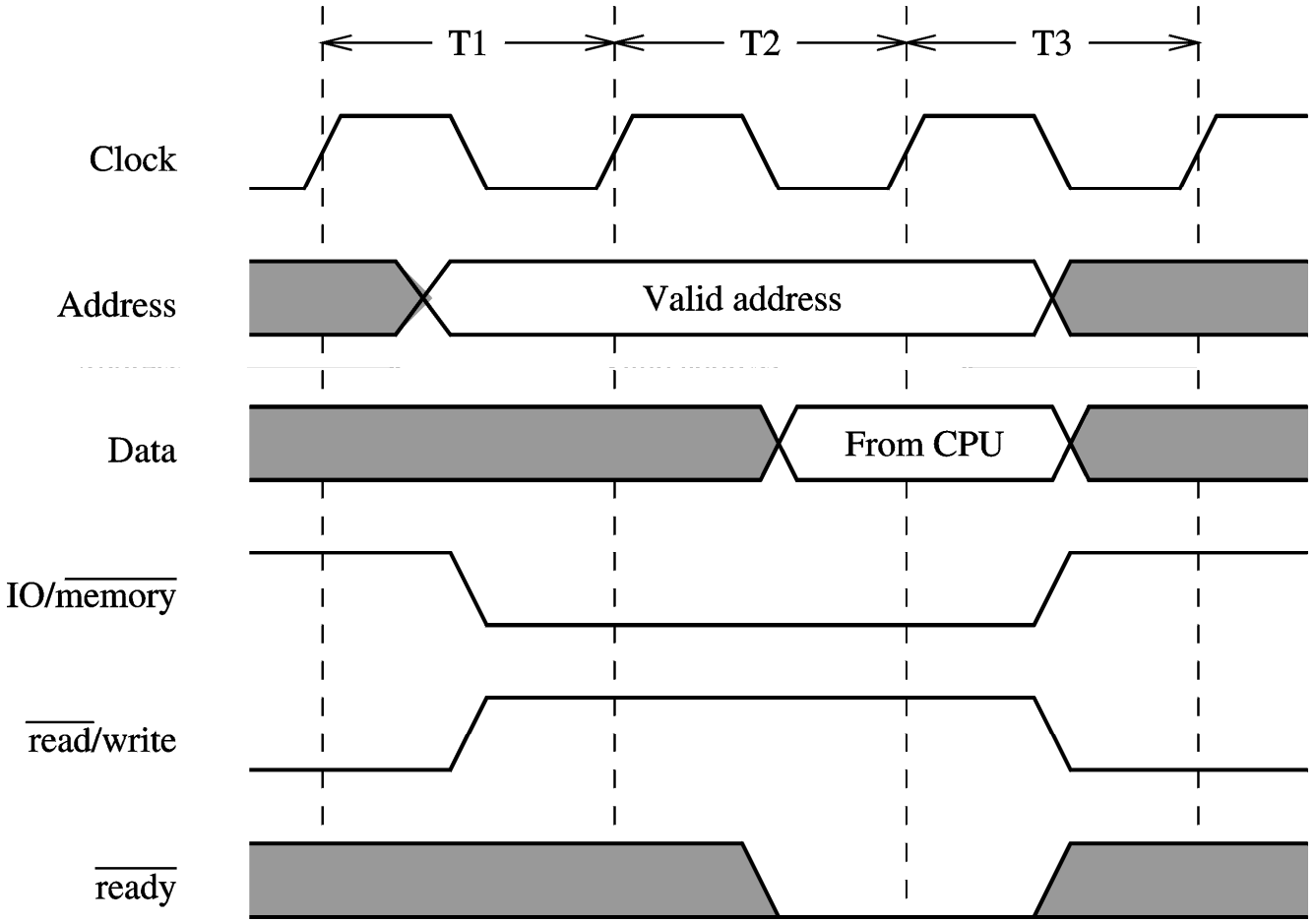
# Bus Síncrono

- Leitura sem ciclos de espera



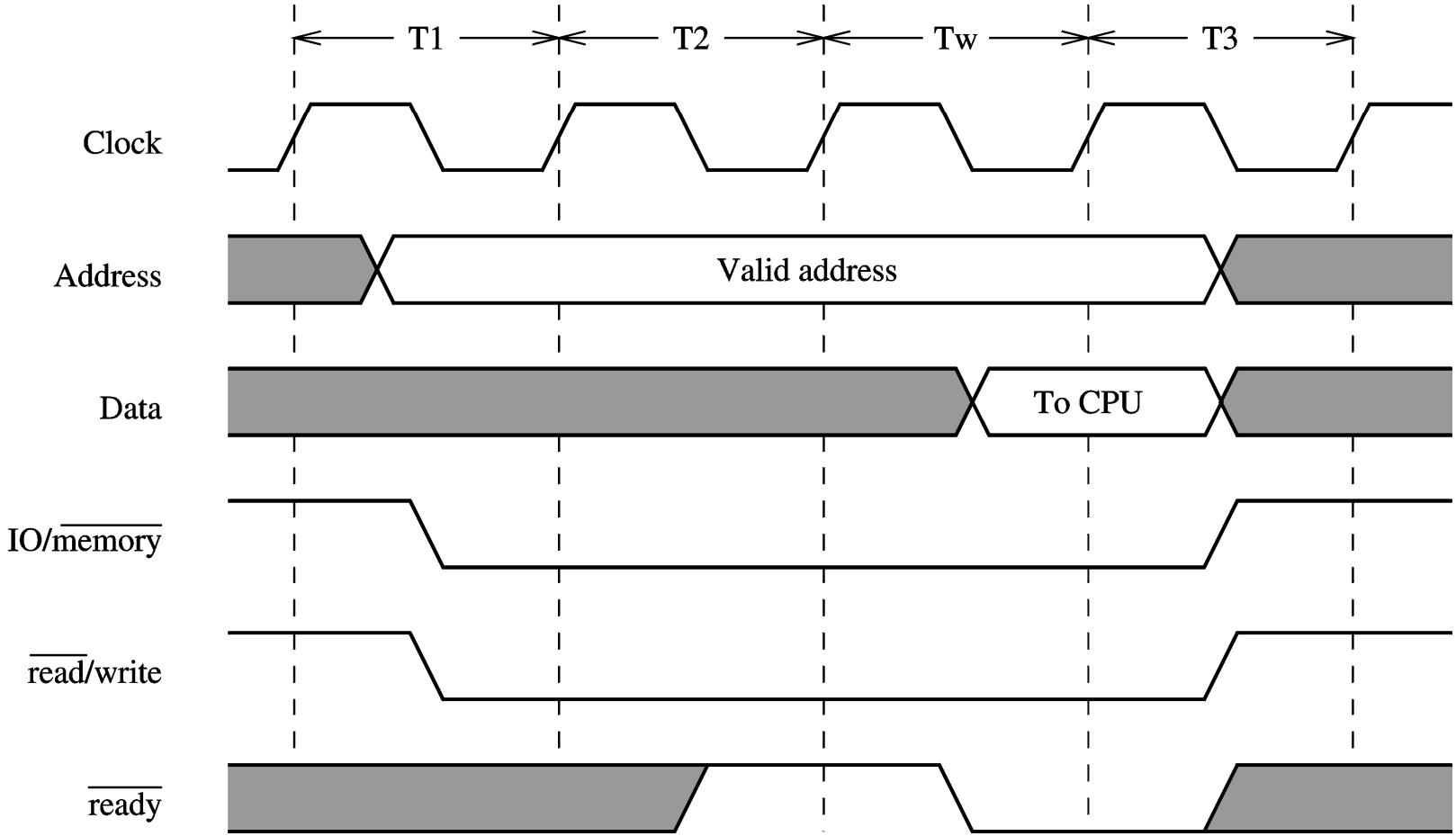
# Bus Síncrono

- Escrita sem ciclos de espera



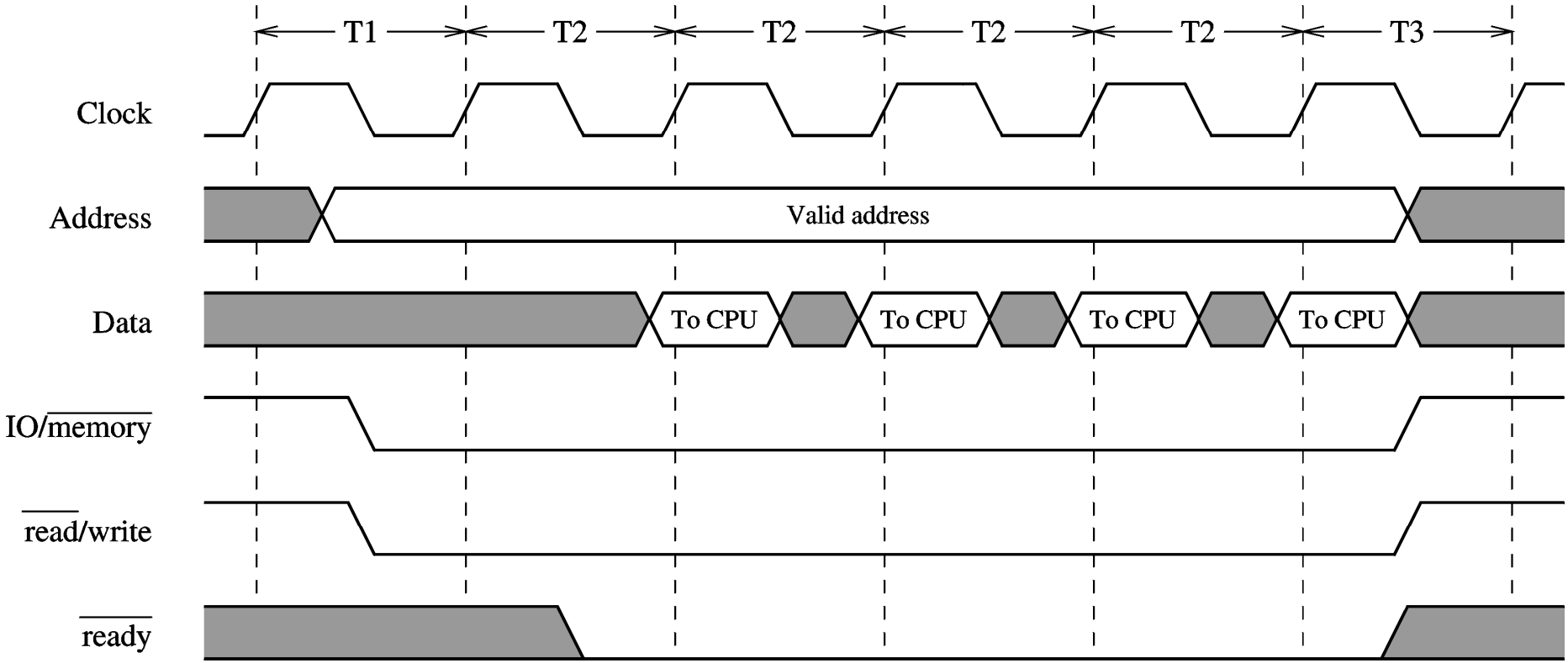
# Bus Síncrono

- Leitura com 1 ciclo de espera



# Bus Síncrono

## Leitura de um bloco



# Bus Assíncrono

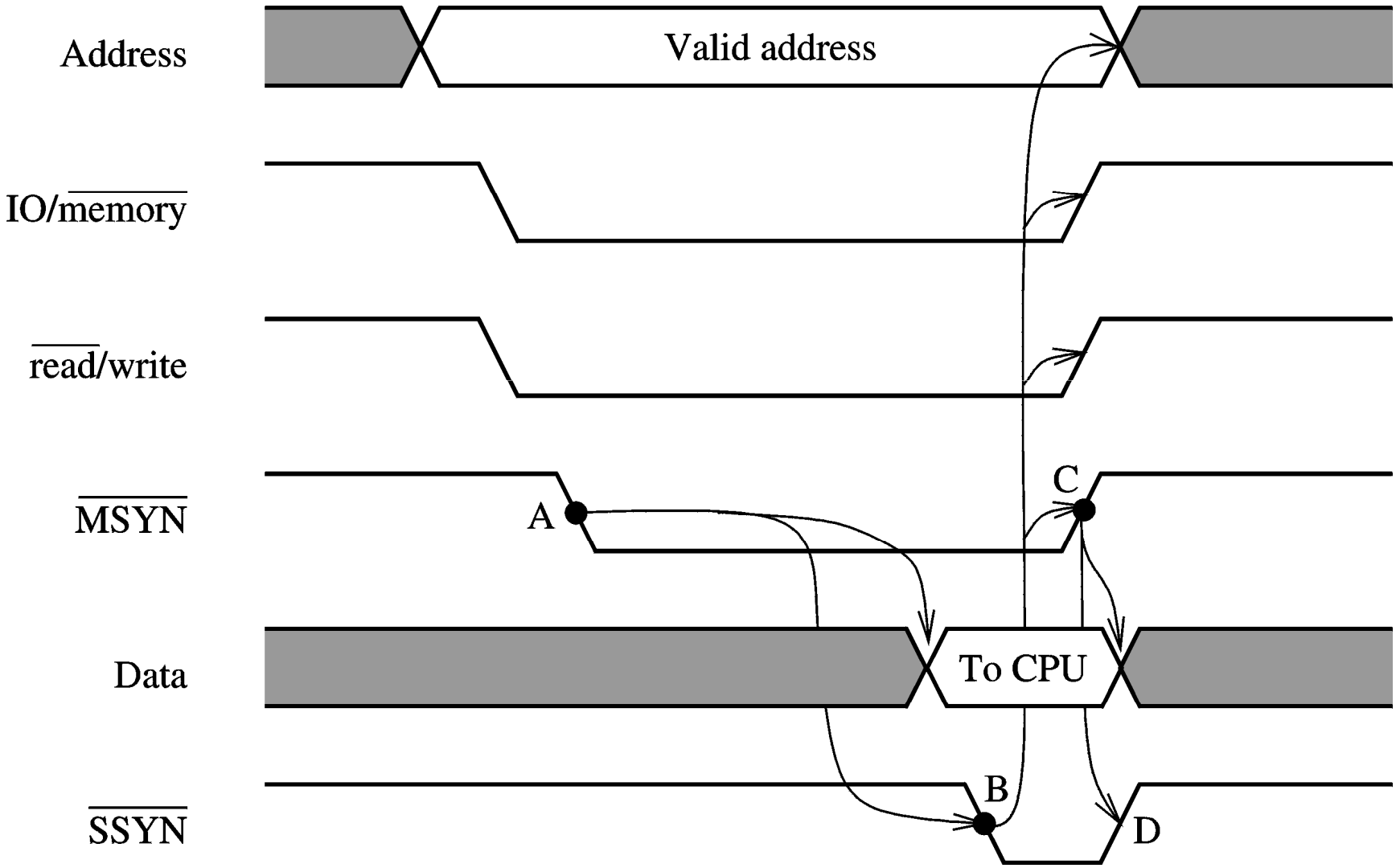
- Não é necessário relógio

- Usa um protocolo de aperto de mão (handshaking)
  - ✓ Sinal de sincronização do mestre – MSYN
  - ✓ Sinal de sincronização do escravo – SSYN

- Vantagem
  - ✓ Ausência de relógio
  - ✓ Mais fléxiveis

- Desvantagem
  - ✓ Complexidade

# Bus Assíncrono



# Arbitragem

- Se mais do que um mestre quer aceder ao bus num dado ciclo é necessário um mecanismo que arbitra o procedimento
- Pode ser feito estaticamente ou dinâmicamente

- A arbitragem estática
  - ✓ A alocação do bus é pré-definida
    - ✓ Exemplo: alocação circular (round-robin)
  - ✓ O acesso ao bus pode ser atribuído a um mestre que não o pretende utilizar na altura
    - ✓ Má gestão
  - ✓ Quase todas as implementações usam arbitragem dinâmica

# Arbitragem Dinâmica

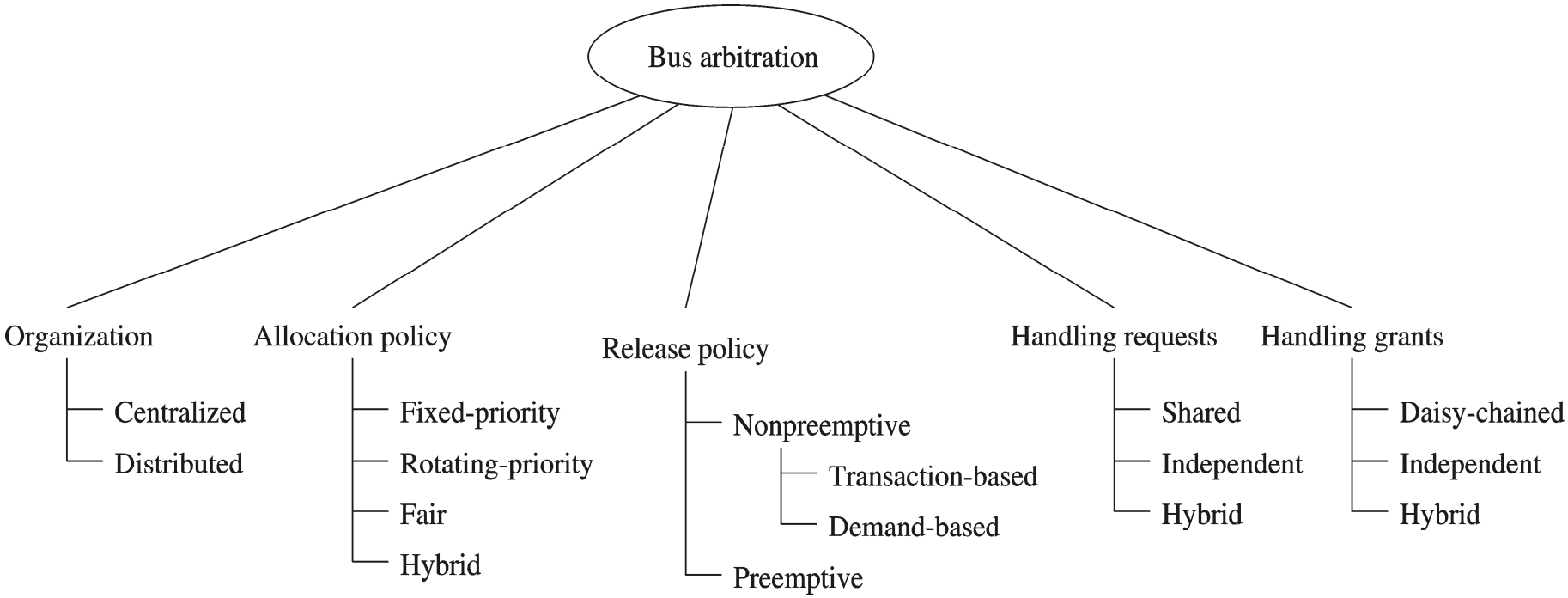
- A atribuição do bus a um mestre é consequência de um pedido prévio
- Cada mestre tem de ter acesso linhas para **pedir** e **obter** autorização para acedir ao bus
- A linha de pedido de acesso serve para informar os restantes mestres da intenção de utilizar o bus
- A autorização é recebida via na linha de obtenção de acesso
- A arbitragem pode ser centralizada ou distribuída

# Arbitragem Dinâmica Centralizada

- Um arbitro central recebe todos os pedidos
- Utiliza uma política de alocação para determinar que mestre deve ter acesso
- Envia a notificação de autorização via as linhas de obtenção
- A libertação do bus também segue uma política pré-determinada

# Arbitragem Dinâmica Distribuída

- A arbitragem é distribuída pelos vários mestres
- Um algoritmo distribuído é utilizado para decidir quem teve ter acesso



# Políticas de Alocação

- Política fixa
  - ✓ A cada mestre é atribuído uma prioridade fixa
  - ✓ O mestre com maior prioridade é o que obtém acesso
  - ✓ Em teoria pode levar inanição (morte pela fome) – starvation
  - ✓ Prioridades podem ser atribuídas de acordo com a importância do serviço
- Política rotativa
  - ✓ A prioridade poder ser uma função do tempo de espera, ou
  - ✓ O mestre que liberta o bus fica com menos proridade que os restantes
  - ✓ Elimina a inanição

# Políticas de Alocação

- Política justa
  - ✓ Políticas que não permitem inanição
    - ✓ Como a política rotativa
  - ✓ Não recorrem a prioridades
  - ✓ Exemplo de políticas desta classe:
    - ✓ Agrupar os pedidos em janelas temporais
      - ✓ Só se progride para a janela seguinte depois de todos os pedidos da janela corrente serem atentidos
    - ✓ Requisitos de tempo de resposta
      - ✓ Um pedido não pode esperar mais do que um dado tempo
      - ✓ No bus PCI a justiça é especificada indicando o tempo máximo para o atraso da autorização
- Política híbrida
  - ✓ Junta as políticas baseadas em prioridades e justiça: PCI

# Políticas de Libertação

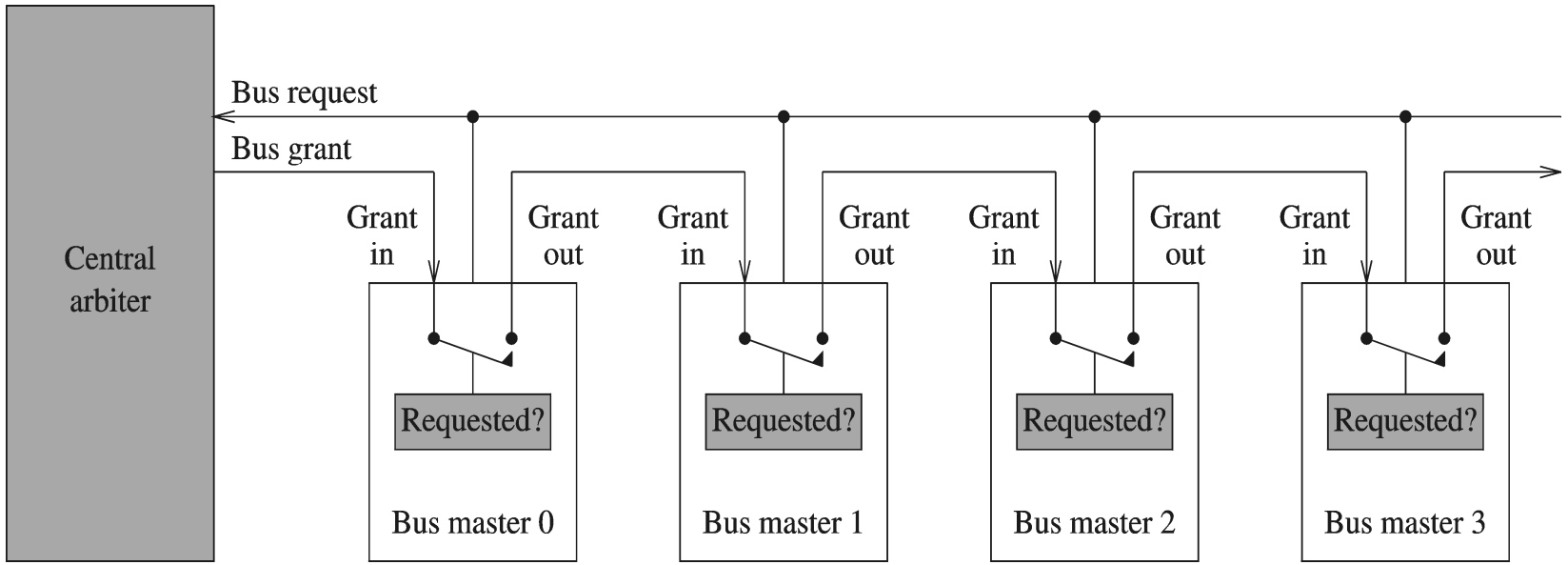
- Específica as condições em que um mestre deve libertar o bus
- Sem preempção
  - ✓ O mestre liberta o bus voluntariamente
  - ✓ Um mestre pode deter o bus por muito tempo
  - ✓ A política pode ser baseada em transacções
    - ✓ O mestre liberta o bus depois de ter terminado uma transacção
    - ✓ Uma nova transacção implica um novo pedido de acesso
    - ✓ Penalização desnecessária se apenas um mestre precisar do bus
      - ✓ Cenário usual em sistemas com apenas 1 processador/core
  - ✓ Um alternativa é a política baseada em pedidos
    - ✓ O mestre liberta o bus depois de ter concluído todos os pedidos pendentes

# Políticas de Libertação

- Com preempção
  - ✓ Caso a transacção seja muito longa é interrompida e o acesso é dado a outro mestre

# Arbitragem Centralizada - Implementação

- Daisy-chaining
  - ✓ Usa apenas uma linha de pedido
  - ✓ O arbitro envia a autorização ao mestre mais próximo
  - ✓ Este, caso não pretenda aceder ao bus, passa a autorização ao próximo



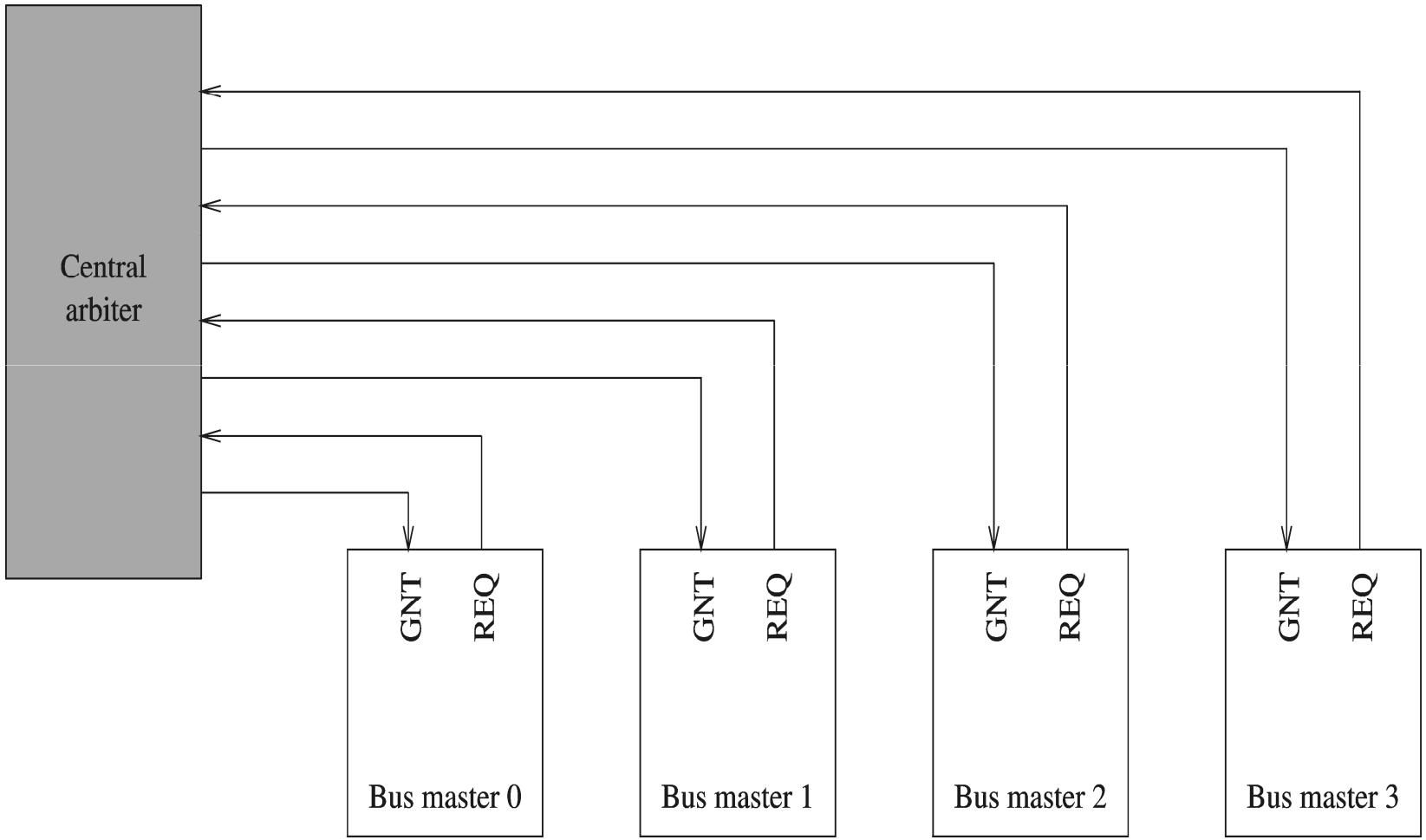
# Arbitragem Centralizada - Implementação

- Daisy-chaining
  - ✓ Vantagem
    - ✓ Fácil de implementar
  - ✓ Desvantagens
    - ✓ Prioridade fixa
    - ✓ Tempo de arbitragem proporcional ao número de mestres
    - ✓ Não tolera falhas

# Arbitragem Centralizada - Implementação

- Pedidos independentes
  - ✓ Arbitro está ligado a cada mestre
  - ✓ Várias políticas podem ser implementadas
  - ✓ Vantagens
    - ✓ Elimina as limitações do Daisy-chaining
  - ✓ Desvantagens
    - ✓ Complexidade
    - ✓ Número de sinais de controlo é proporcional ao número de hosts
  - ✓ Implementação usada pelo PCI

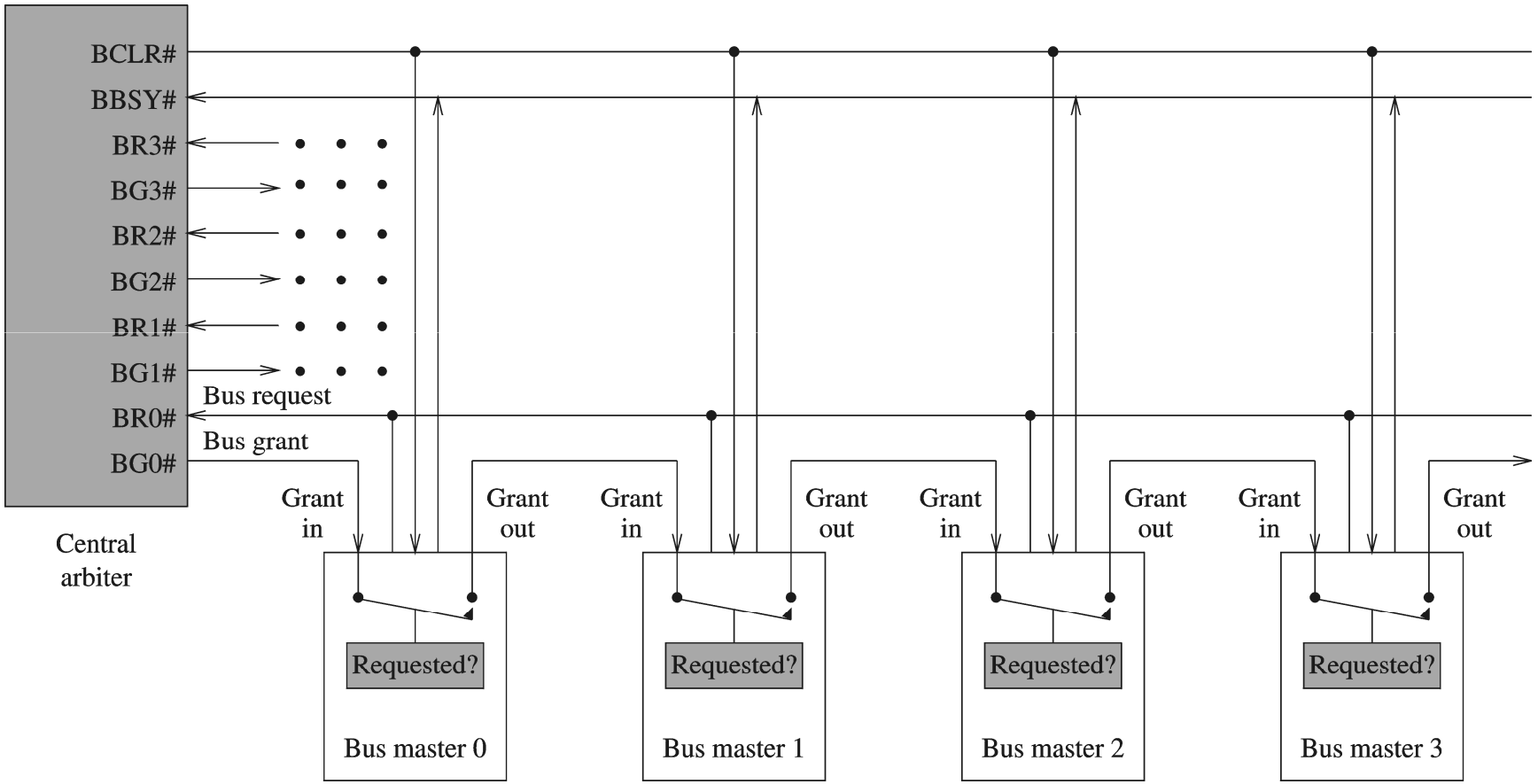
# Arbitragem Centralizada - Implementação



# Arbitragem Centralizada - Implementação

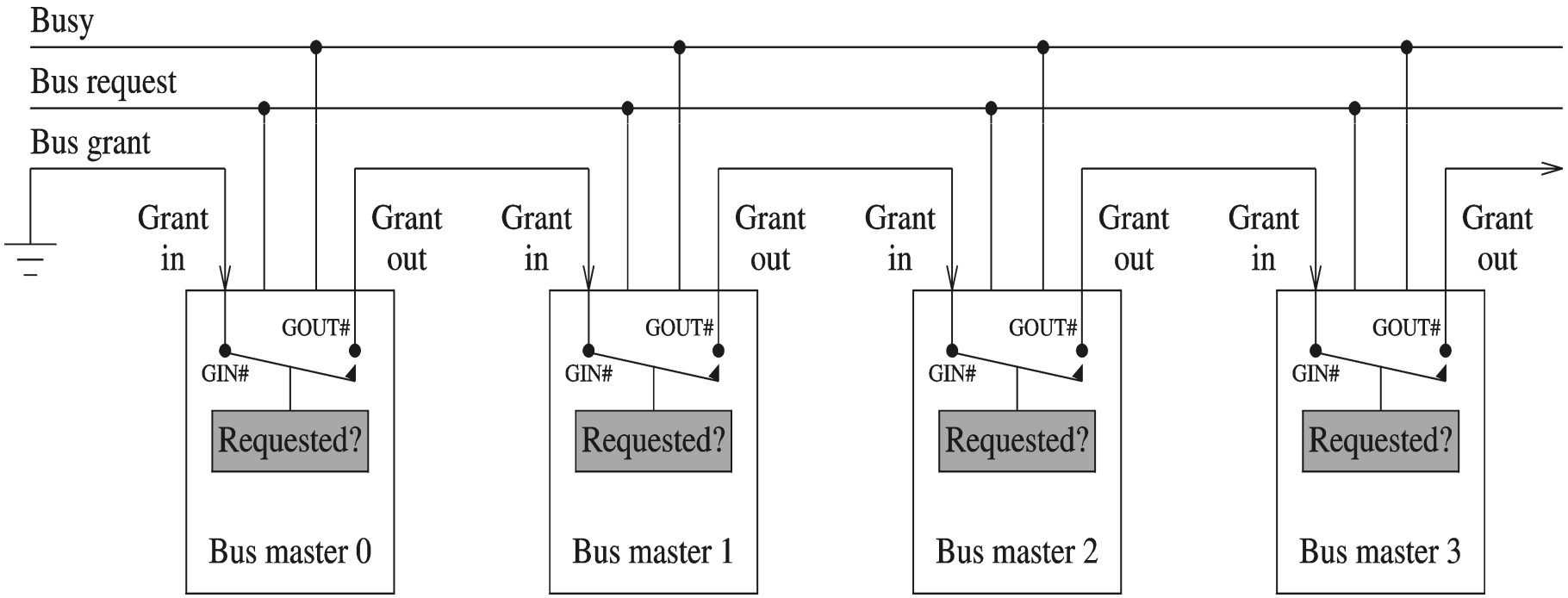
- Híbrido
  - ✓ Combina o melhor do Daisy-chaining e dos pedidos independentes
  - ✓ Os mestres são divididos em classes
  - ✓ Pedidos independentes entre as classes
  - ✓ Daisy-chaining dentro da classe
  - ✓ O VME – bus desenvolvido pela Motorola com grande aceitação em servidores - utiliza esta implementação

# Arbitragem Centralizada - Implementação



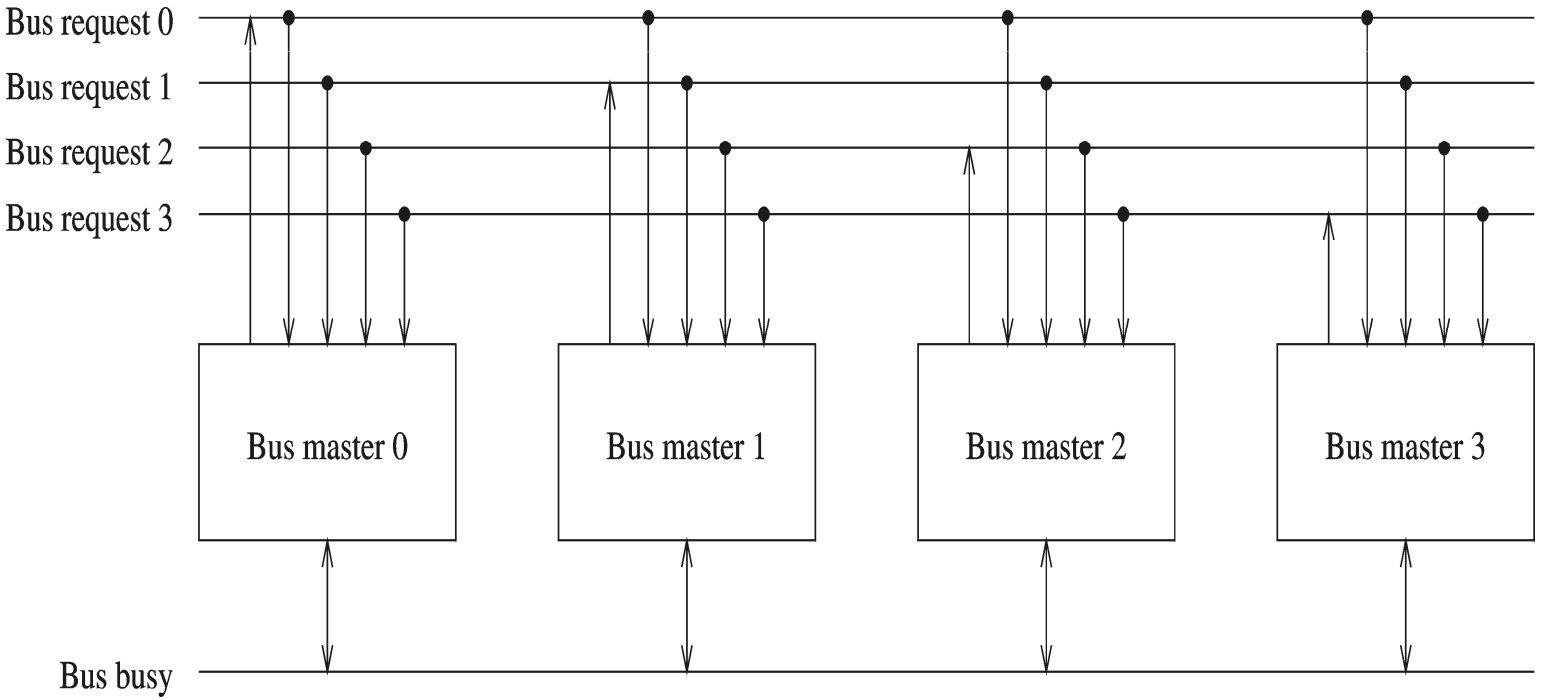
# Arbitragem Distribuída - Implementação

- Daisy-chaining



# Arbitragem Distribuída - Implementação

- Pedidos independentes
  - ✓ Todos os mestres têm acesso a todas as linhas de pedidos
  - ✓ O que tem maior prioridade obtém o acesso
    - ✓ A política de alocação é fixa – precisa de um esquema de redução de prioridade



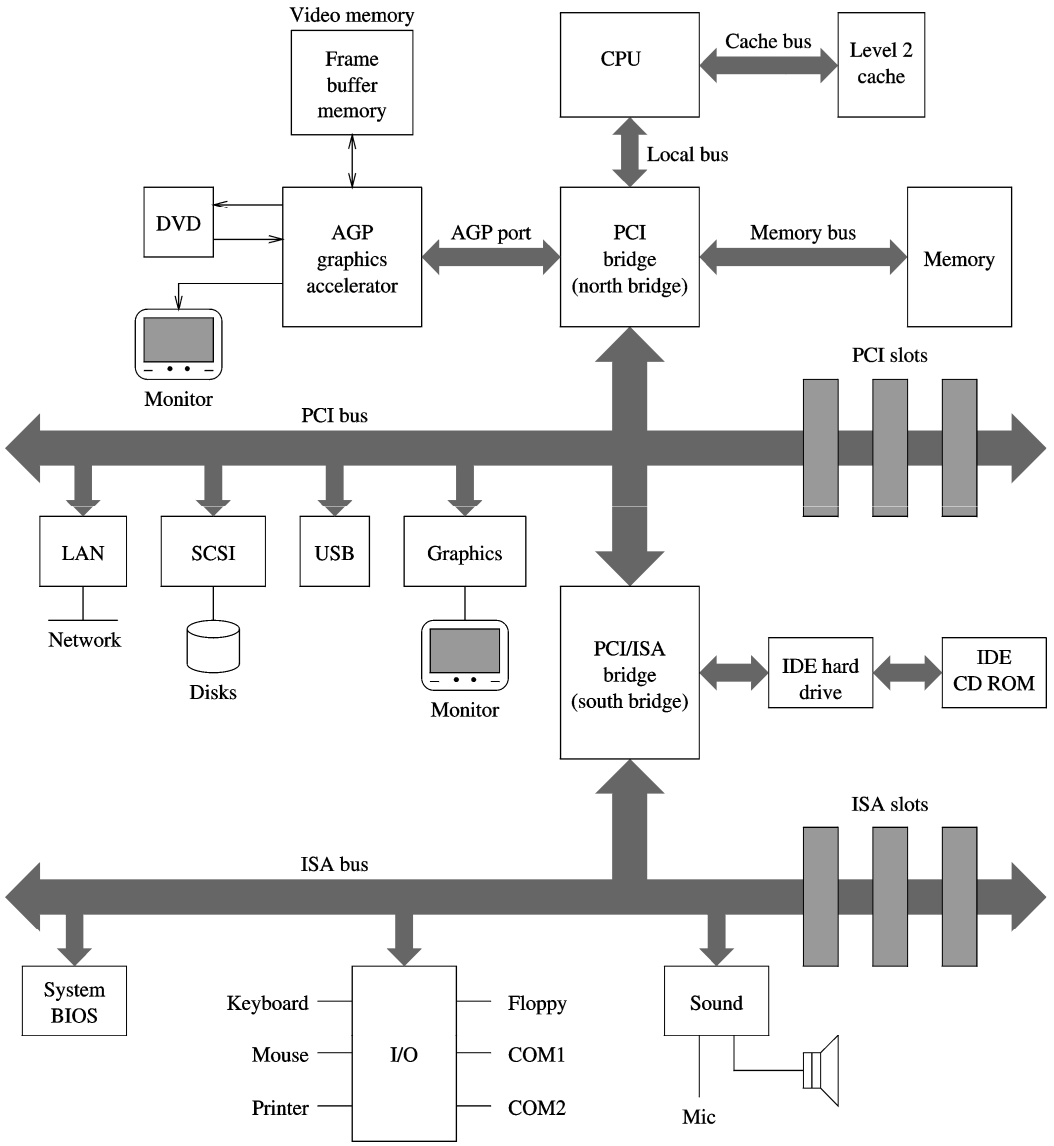
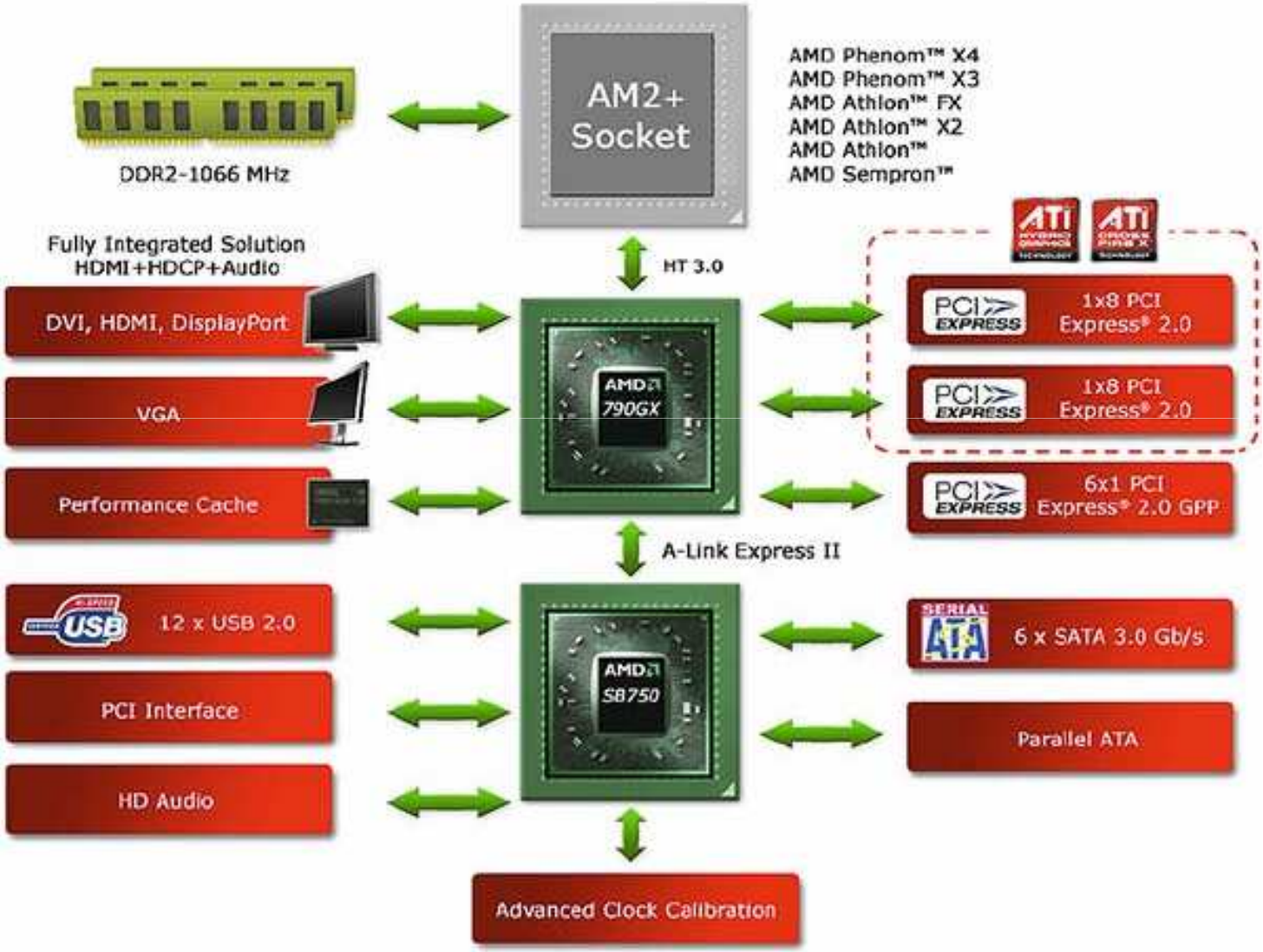
# Exemplos de Buses

- ISA
  - ✓ Bus antigo usado inicialmente para conectar o CPU à memória e mais tarde periféricos
- PCI e PCI-X
  - ✓ Buses para a conexão de periféricos em PCs
- AGP
  - ✓ Suporte para placas gráficas de alta performance
- PCIe
  - ✓ Standard actual para a conexão de periféricos em PCs
  - ✓ Substituiu o PCI, o PCI-X e o AGP
- ExpressCard e o mais antigo PCMCIA (PC Card)
  - ✓ Buses para ligar periféricos em portáteis

# Requisitos de um Bus

- Cada bus tem de suportar a taxa de transferência dos componentes que interliga
  - ✓ Componentes com maiores requisitos de velocidade são ligados na *Northbridge*, os restantes à *Southbridge*
- O FSB (Front-Side Bus) liga o processador aos restantes componentes
- Consideremos um FSB com relógio de 800MHz e uma largura de banda de 8 bytes
  - ✓ O CPU parte esse relógio para definir o seu relógio interno. Se partir em 3 vezes terá um relógio de 2.4GHz
  - ✓ As memórias actuais (DDR) realizam duas transferências por ciclo de bus. Neste caso 2x800 – FSB 1600
    - ✓ A taxa de transferência é de 2*800*8bytes = 12800 Mbytes/s

# Hierarquia de Buses de um PC - Exemplo



# Hierarquia de Buses de um PC actual