

## Tipos de Barramentos *(Bus Types)*

### ↩ Dedicados

- × Linhas de Dados & endereços separadas

### ↩ Multiplexados

- × Linhas compartilhadas
- × linha de controle de endereços válidos ou dados válidos
- × Vantagem - *poucas linhas*
- × Desvantagem
  - ✓ *Controle mais complexo*
  - ✓ *Redução potencial de desempenho, desde que certos eventos que compartilham as mesmas linhas não podem ser realizadas em paralelo*



19

## Arbitragem *(Bus Arbitration)*

### ↩ Mais de um módulo controlando o barramento

- × Por exemplo: CPU e DMA controller

### ↩ Somente um módulo pode controlar o barramento por vez.

### ↩ Arbitragem pode ser *centralizada* ou *distribuída*

↑ *decide quem usa o barramento*

### ↩ Arbitragem Centralizada

- × **Um único dispositivo** de hardware controlando o acesso ao barramento.
  - ✓ Bus Controller
  - ✓ Arbiter
- × Pode ser parte da CPU ou separado dela.

*- Arbitragem -> controlador de barramento*

### ↩ Arbitragem Distribuída

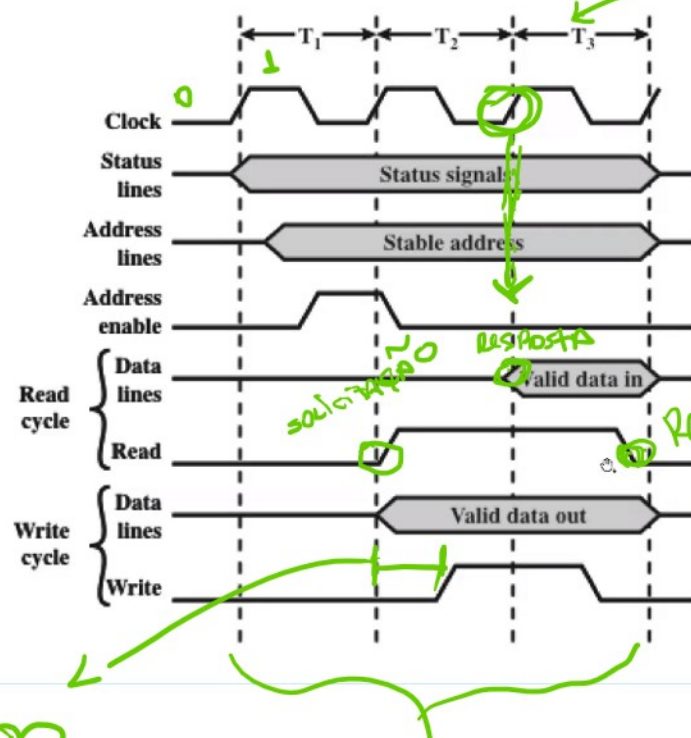
- × **Cada módulo pode reivindicar o barramento.**
- × Controle lógico em todos os módulos.

# Temporização (Timing)

## Coordenação de eventos no barramento

### Síncrono (Synchronous)

- × **Eventos determinados pelo sinal do relógio (clock)**
- × **Barramento de controle inclui linha de clock**
- × **Uma transição de 0-1 é definida como ciclo de barramento**
- × **Todos os dispositivos podem ler a linha de clock**
- × **Usualmente sincronizados na transição.**
- × **Usualmente um único ciclo por evento.**



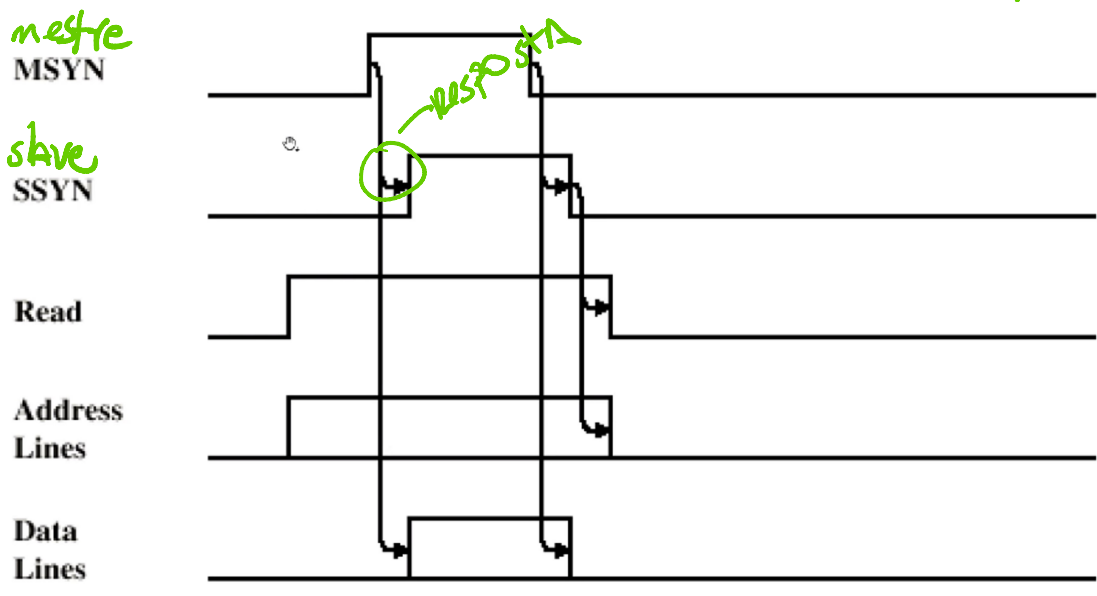
AGUARDANDO ESTABILIZAR

Tempo TOTAL = 3 ciclos de clock de barramento

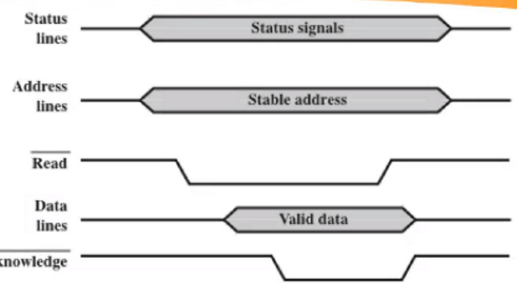
# Asynchronous Timing Diagram

sem sinal de clock

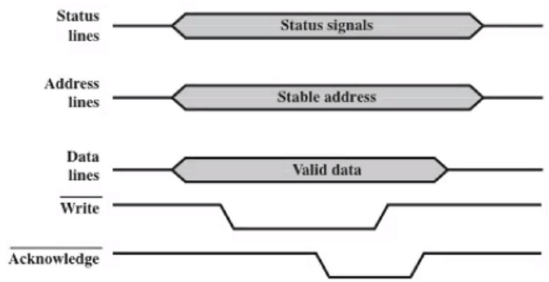
CONTROLE mestre-escravo



Resposta mestre-escravo



(a) System bus read cycle



(b) System bus write cycle