

Organização e Arquitetura de Processadores

Modelos de Comunicação Digital

Leituras Aconselhadas:
Capítulo 7 do Monteiro
Capítulos 3 e 7 do Stallings
Seções 2.4, 3.6, 3.7 do Tanenbaum & Austin
Seções 8.5 a 8.6 do Patterson & Hennessy

Última alteração: 10/06/2022

Prof. Ney Laert Vilar Calazans

Baseado em notas de aulas originais do Prof. Dr. César Marcon

Introdução

- **Modelos atendem características de diferentes sistemas**
 - Sistemas assíncronos
 - Sistemas síncronos
- **Modelos de trocas de dados em nível elétrico levam em consideração **sinais** e **protocolos** necessários para realizar trocas de informações**
- **A implementação do modelo em hardware é um circuito responsável pela interação entre sistemas (por exemplo: periférico e controlador)**
- **O modelo de comunicação pode considerar a direção da mensagem e a simultaneidade**
 - Unidirecional
 - Bidirecional em tempos distintos
 - Bidirecional simultâneo

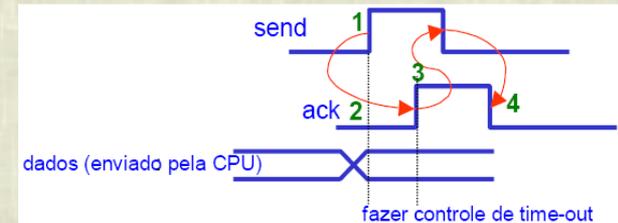
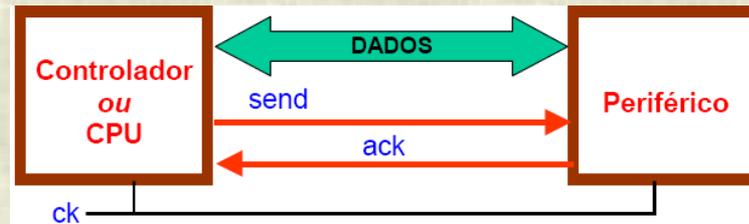
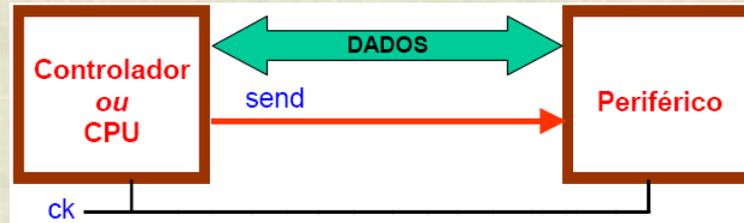
Modelos de Comunicação – Critérios e Taxonomias

• Critério Paralelismo

- Com. Serial / Com. Paralela

• Critério Sincronismo

- Com. Síncrona →
- Com. Semi-síncrona →
- Com. Assíncrona →



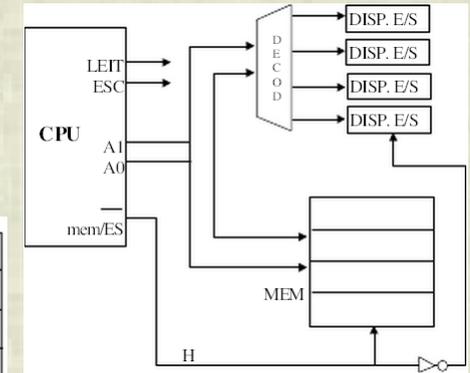
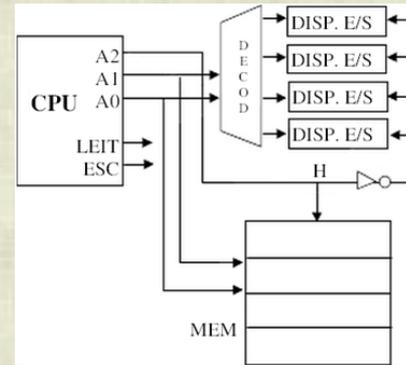
• Critério Sentidos dos Fluxos de Informação

- Com. *Simplex*
- Com. *Half-duplex*
- Com. *Full-duplex*

Tipos de Sistemas de Entrada e Saída (E/S)

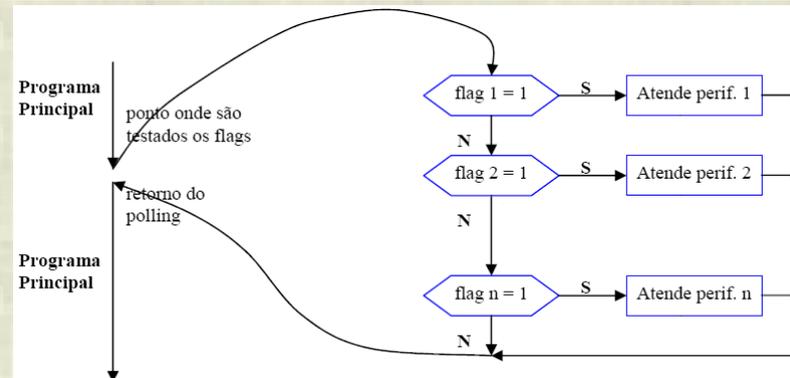
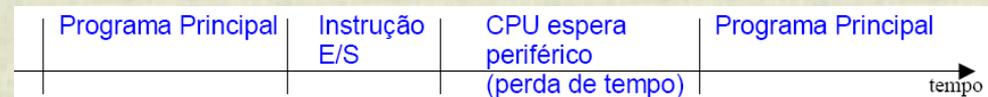
- **Mapeamento de Entrada e Saída (depende do Hw, ou não!)**

- Em memória (sempre viável)
- Em portas de entrada e saída (pressupõe Hw específico)



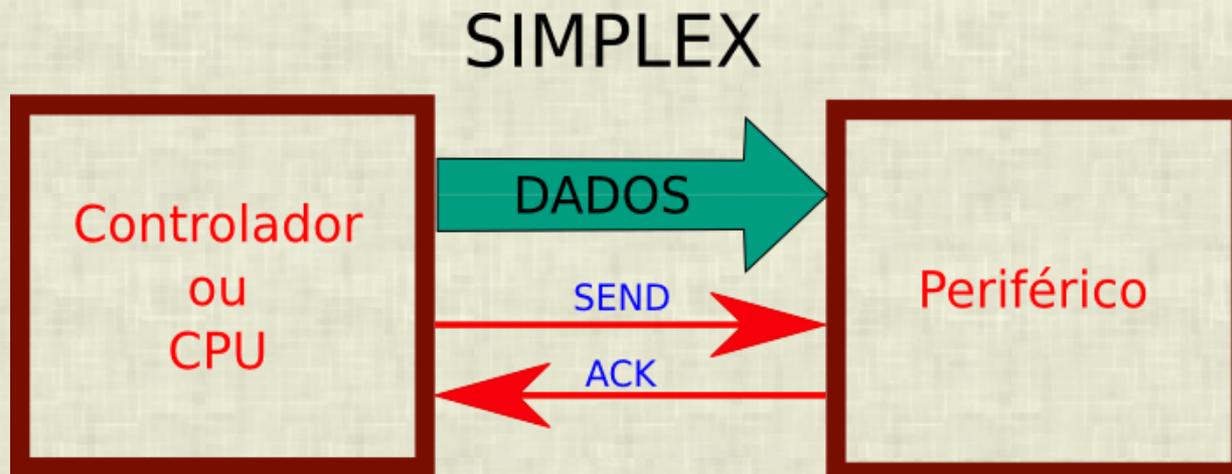
- **Modos de Transferência de Dados**

- Modos E/S programada versus não programada
- E/S Programada
 - Bloqueado
 - *Polling* (inquisição)



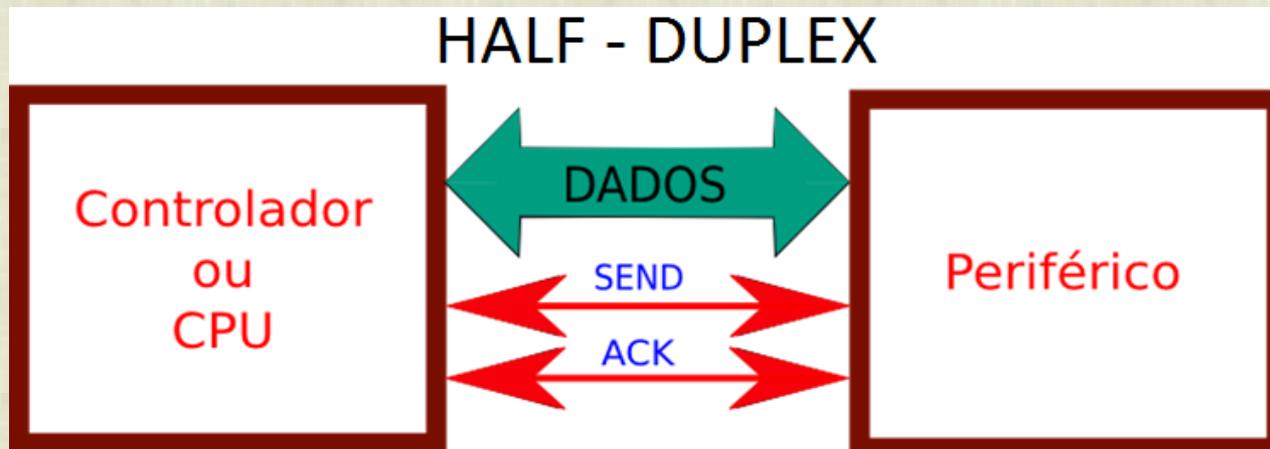
Direção da Comunicação (Simplex)

- **Simplex é uma comunicação unidirecional**
 - Um transmissor
 - Um receptor
- **Exemplos de aplicação**
 - Transmissão de TV
 - Transmissão de rádio
- **Exemplo de implementação com 3 sinais**
 - DADOS: barramento de dados unidirecional
 - SEND: sinal que informa envio de dados para o controlador
 - ACK: sinal que confirma o recebimento dos dados do controlador



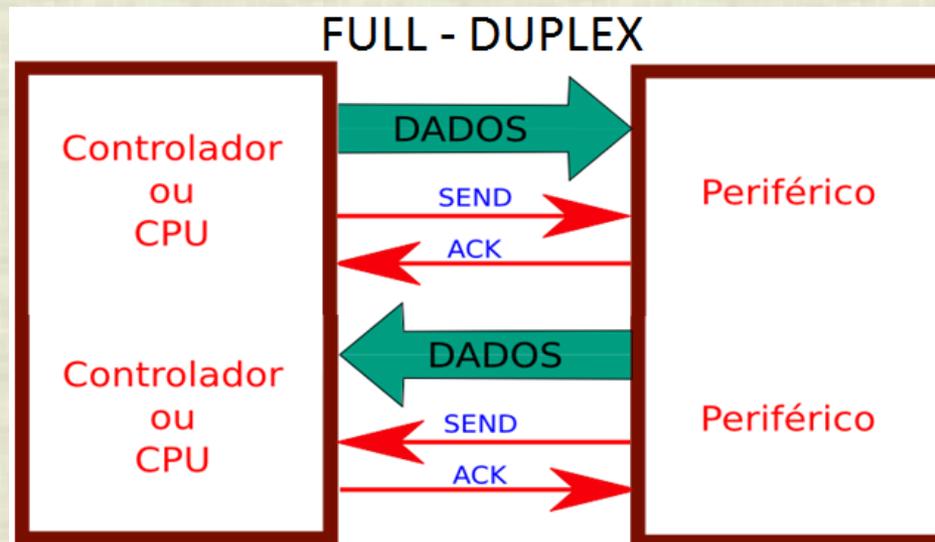
Direção da Comunicação (Half-Duplex ou Semi-Duplex)

- **Half-Duplex ou Semi-Duplex é uma comunicação bidirecional não simultânea**
 - Dois transmissores
 - Dois receptores
 - Compartilhamento do meio físico de dados
- **Exemplo de aplicação**
 - *Walkie-talkie*
- **Exemplo de implementação com 3 sinais**
 - DADOS: barramento de dados bidirecional
 - SEND: Sinal de envio de dados nas duas direções
 - ACK: Sinal que confirma o recebimento dos dados nas duas direções



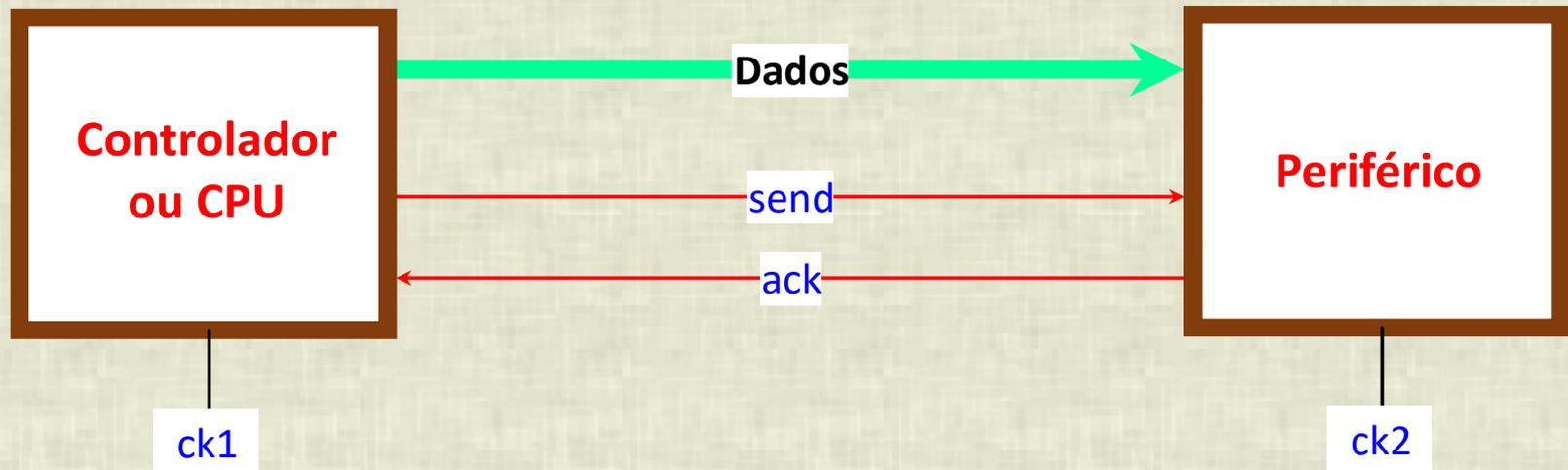
Direção da Comunicação (Full-Duplex)

- **Full-Duplex é uma comunicação bidirecional simultânea**
 - Dois transmissores
 - Dois receptores
 - Meio físico de dados exclusivo de cada comunicação
- **Exemplo de aplicação**
 - Telefonia
- **Exemplo de implementação com 3 sinais**
 - DADOS: Dois barramento de dados. Um para cada direção
 - *SEND*: Sinal de envio de dados nas duas direções
 - *ACK*: Sinal que confirma o recebimento dos dados nas duas direções

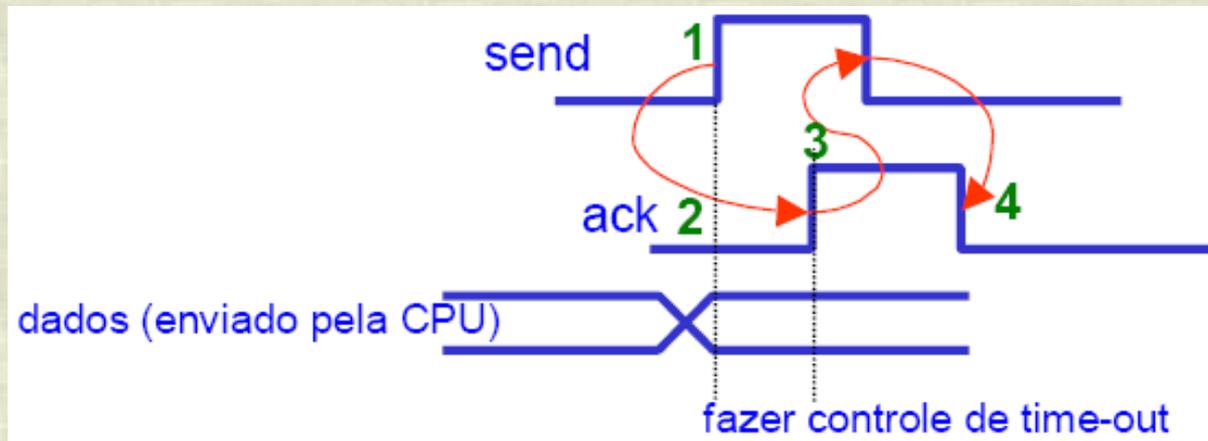


O Modelo Assíncrono

- Dois sistemas computacionais autônomos que não necessitam ter a mesma referência de tempo (relógio)
- Relógios
 - Não são necessariamente sincronizados
 - Podem ter frequências diferentes



O Modelo Assíncrono – Handshake de 4 fases



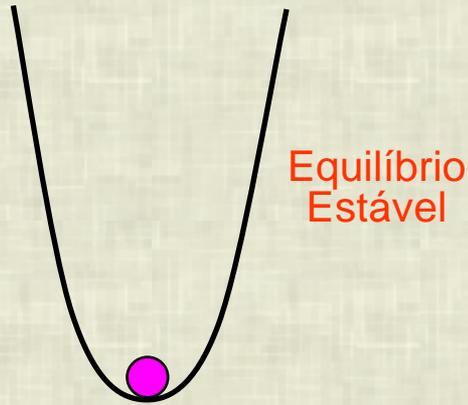
CPU	Periférico
1) Fornece dados e send ↑	1) Monitorando subida do sinal send
2) Fica a esperar subida do sinal ack	2) Quando send ↑ armazena os dados
	3) Uma vez armazenados os dados ack ↑
3) Quando ack ↑ remove send (↓) e os dados	4) Fica a esperar descida do sinal send
	5) Quando send é removido remove-se o ack (↓)
4) Fica a esperar descida do sinal ack (↓)	

O Modelo Assíncrono – Handshake de 4 fases

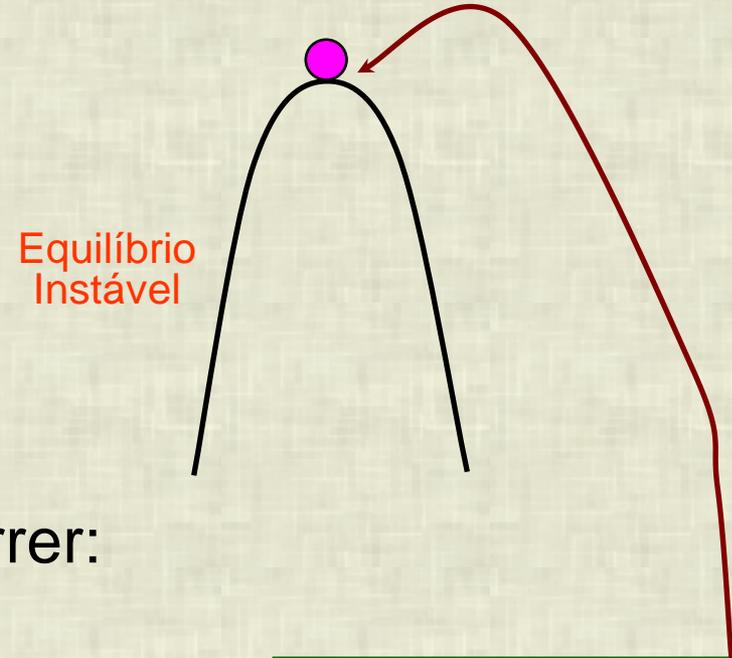
- **Perguntas**

1. Será que deve haver algum mecanismo de controle entre os passos 1 e 2 da CPU? (*em relação à figura*)
 - Porque?
 - Se a resposta for positiva, qual a consequência em não tê-lo?
2. Porque os passos 3 e 4 da CPU são necessários?
 - Qual a consequência em não tê-los?
3. Qual a consequência em ter frequências diferentes no modelo assíncrono? Esboce formas de onda que elucidam a resposta
4. Faça um diagrama de tempos para a CPU e outro para o periférico de forma a implementar uma comunicação unidirecional no sentido CPU -> Periférico
5. Existe algum outro protocolo que poderia ser utilizado para o modelo assíncrono? Se sim, sugira um. Pense em termos de sinais necessários e em tempos de temporização dos mesmos

Uma Palavra sobre Metaestabilidade

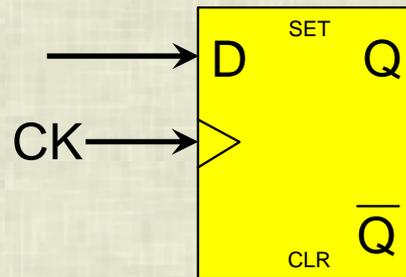


Analogia Mecânica



- Eletricamente, o mesmo pode ocorrer:

Supor que uma transição ocorre em D ao mesmo tempo que uma transição do sinal de relógio!



O que pode acontecer na saída Q?

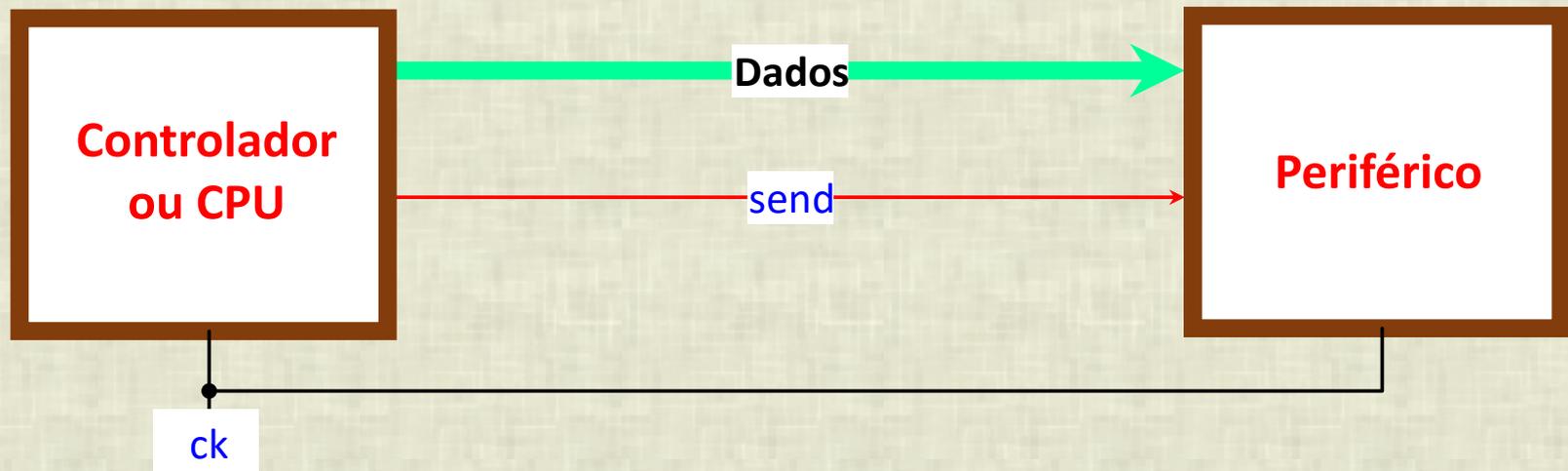
Síncronos versus Não-síncronos

Metaestabilidade

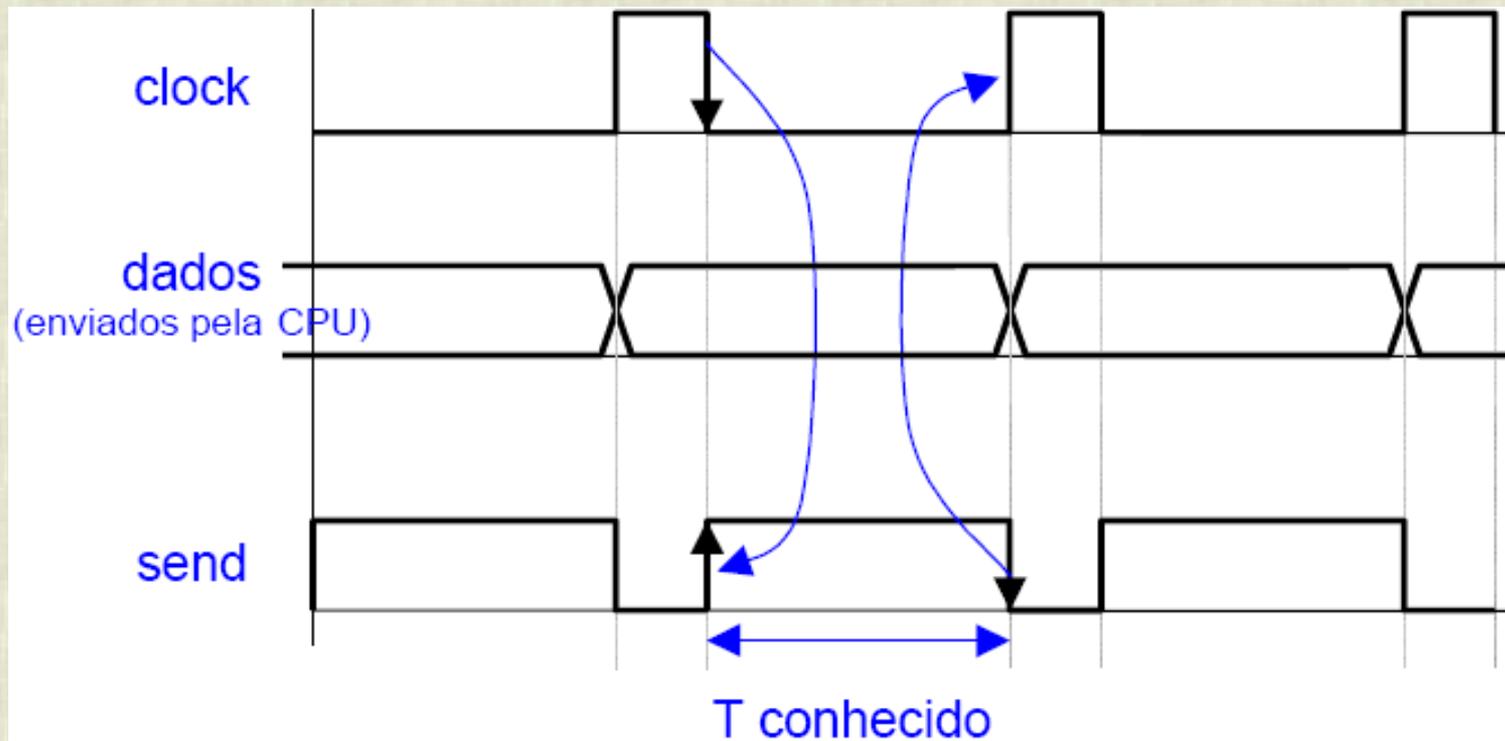
- Eletricamente:
 - Se a entrada D transicionar ao mesmo tempo que o sinal que comanda seu armazenamento em um elemento de memória, coisas *horríveis* podem acontecer
 - O valor finalmente armazenado pode não ser o desejado (**Ruim**)
 - O valor finalmente armazenado (certo ou errado) pode sofrer uma demora arbitrária para aparecer na saída (**PIOR**)
 - O valor pode ser armazenado ora de forma correta, ora incorreta (**MUITO RUIM**)
 - A saída do circuito de armazenamento pode ficar em valor lógico inválido (nem 0, nem 1) por um tempo arbitrário! (**CATASTRÓFICO**)
- Metaestabilidade deve ser evitada a todo custo!

O Modelos de Troca de Dados Síncrono

- **Dois sistemas computacionais autônomos com mesmas referências temporais**
- **Relógio**
 - Mesma frequência
 - Necessariamente sincronizados
- **Tempo para a transferência é conhecido**
 - Não há necessidade de sinal de ack
 - Podem ser vários ciclos de relógio



Um Protocolo para o Modelo Síncrono



1. Quando $ck \uparrow$, CPU coloca dados no barramento
2. Quando $ck \downarrow$, $send \uparrow$ significando que há dado a ser transferido
3. Quando $ck \uparrow$ novamente terminou a transferência ($send \downarrow$)

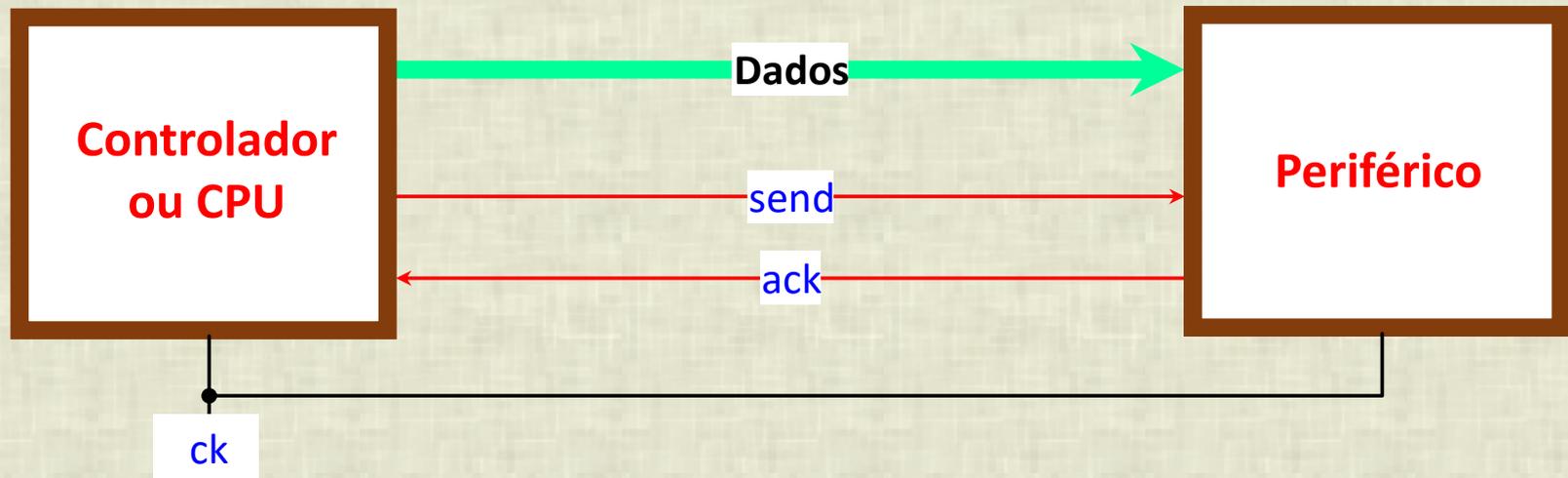
O Modelo Síncrono

- **Perguntas**

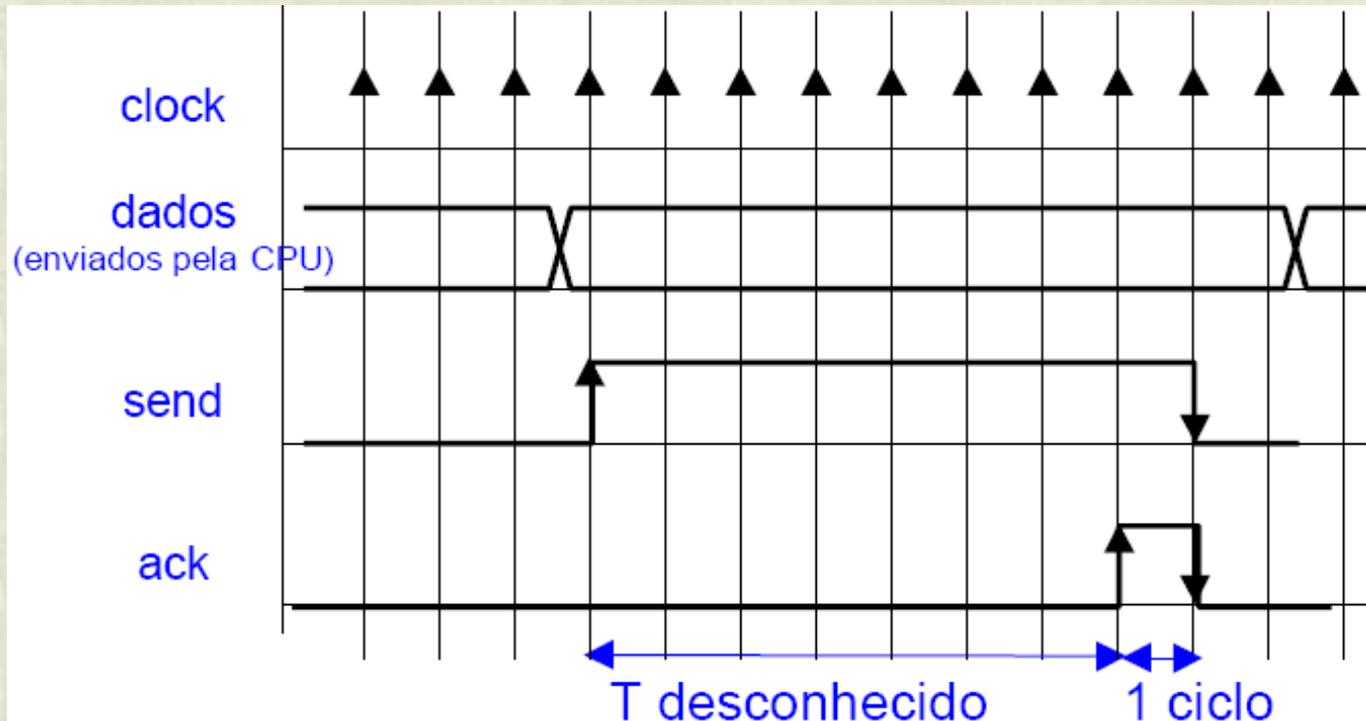
1. Porque o sinal *send* é necessário?
 - Qual a consequência de sua ausência?
2. Porque o sinal *ack* não é necessário?
 - Como os sistemas detectam o fim de uma comunicação?
3. Qual a consequência de não ter sinais de relógio sincronizados?
Analise em termos de fase e frequência
4. Para distâncias muito longas este modelo pode ter problemas?
Quais?
5. Faça o esquema físico e os diagramas temporais para uma comunicação bidirecional full-duplex. Alguma coisa muda deste modelo para uma comunicação *half-duplex*?

O Modelo Semi-Síncrono

- **Dois sistemas computacionais autônomos com mesmas referências temporais, mas com tempo de transferência desconhecido**
 - Transferência pode ocorrer em vários ciclos de relógio
- **Relógio**
 - Mesma frequência
 - Necessariamente sincronizados



O Modelo Semi-Síncrono



- **Procedimento**

1. CPU disponibiliza dados e sincronamente ativa o sinal send (este sinal deve ficar ativo até periférico responder (podem ser diversos ciclos))
2. Na primeira transição do clock, após o periférico ter armazenado os dados, o periférico ativa o sinal ack
3. No ciclo de clock seguinte CPU remove o send e periférico remove o ack

O Modelo Semi-Síncrono

- **Exercícios/Perguntas**

1. Dê um exemplo onde um mecanismo de *timeout* é necessário
 - Qual consequência de não tê-lo?
2. O periférico, no ciclo seguinte à ativação do sinal *ack*, desativa este novamente. Como ele sabe que a CPU já detectou o sinal?
3. Qual a diferença para a transferência assíncrona? Em termos de hardware e em termos de máquina de controle (diagrama de estados de ambas as partes – periférico e CPU)
4. Faça o diagrama temporal de uma comunicação *half-duplex*

Comunicação Serial

- Transmissão da informação bit a bit

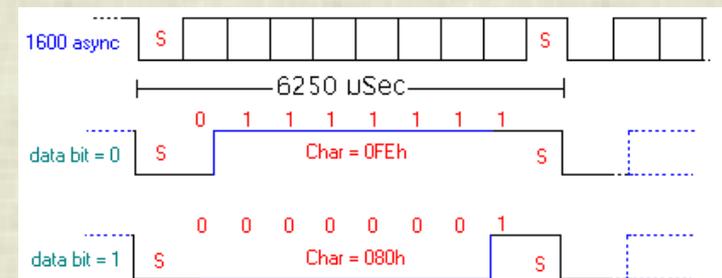
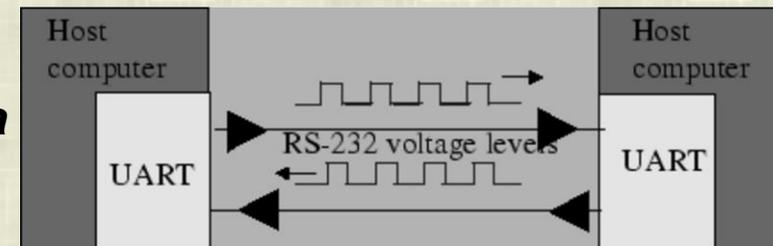
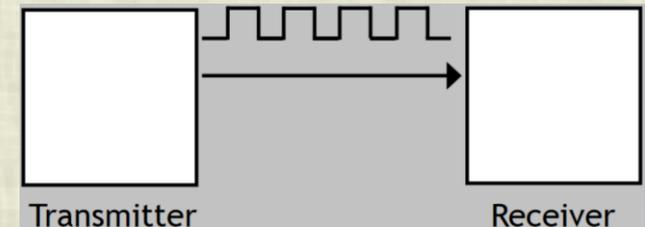
- Modelos

- Sob a mesma linha seguem informação de dados e controle (comunicação assíncrona)
- Uma linha para dados e outras para controle (comunicação síncrona)

- Exemplos: **SPI, RS-232, I2C, USB, FireWire, E-1**

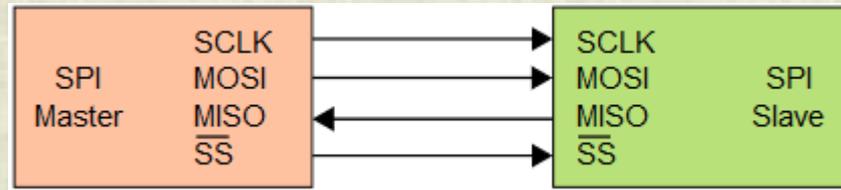
- **RS-232**

- Interface de comunicação serial **assíncrona**
- Longas distâncias de comunicação
- Usado em modems
- Exemplo com uma *Universal Asynchronous Receiver/Transmitter* (UART)
- Exemplo de formas de onda para uma interface serial operando a 1600bps →

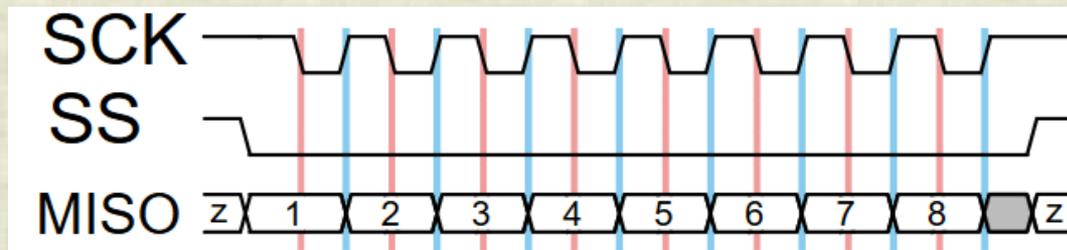


Comunicação Serial

Serial Peripheral Interface (SPI)

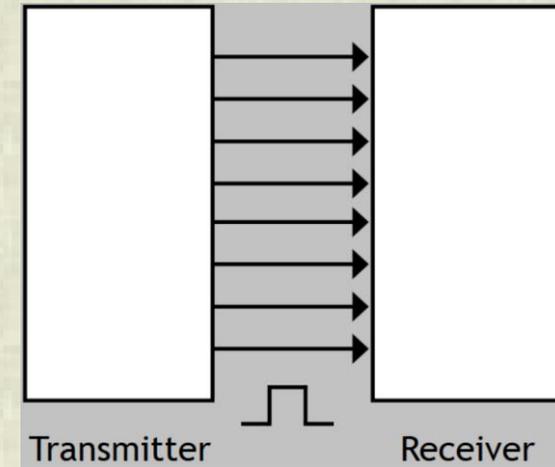


- Interface de comunicação serial **síncrona**
- Curtas distâncias de comunicação
- Sinais
 - SCLK : Serial Clock
 - MOSI : Master Output, Slave Input
 - MISO : Master Input, Slave Output
 - SS : Slave Select (active low)



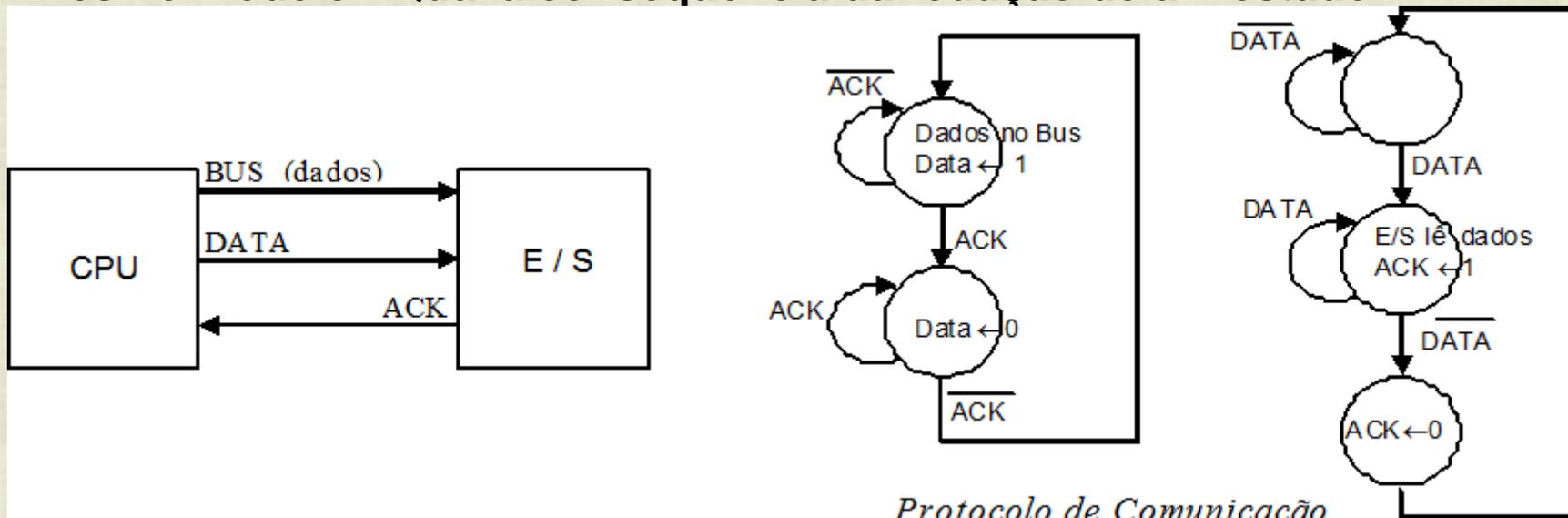
Comunicação Paralela

- **Transmissão da informação com granularidade de palavra**
- **Dados e controle têm linhas exclusivas**
- **Normalmente, tem desempenho e custo maior que a comunicação serial**
- **Exemplos: ISA, ATA, SCSI, PCI**
- **Grande parte das comunicações paralelas foi substituída por versões mais recentes seriais**
 - Exemplo: PCI por PCI Express, ATA por SATA



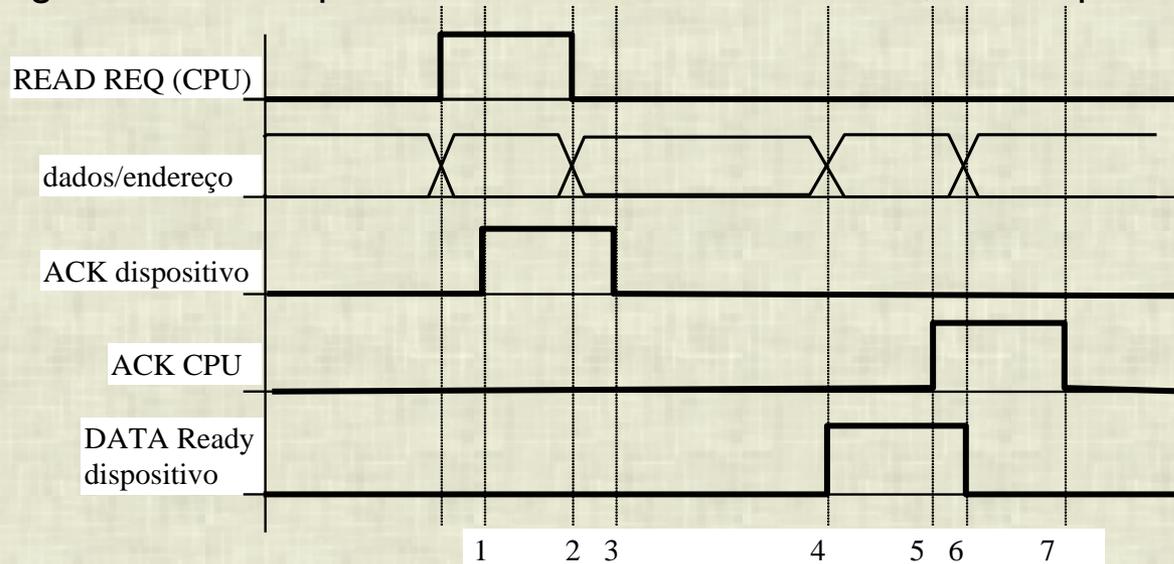
Exercícios

1. Quais são as características do modelo da transferência síncrona, e quais os principais problemas relacionados a este modelo?
2. Faça a mesma análise acima para o modelo de transferência assíncrona.
3. O que diferencia o modelo síncrono do modelo semi-síncrono? Ilustre as diferenças através de um diagrama de tempos e diga quando se aplica um ou outro modelo
4. Explique o protocolo de comunicação abaixo, mostrando o diagrama de tempos para os sinais DATA, ACK e BUS. Este diagrama poderia ser reduzido (ter eliminado um de seus estados) e ainda assim manter o mesmo modelo? Qual a consequência da redução de um estado?



Exercícios

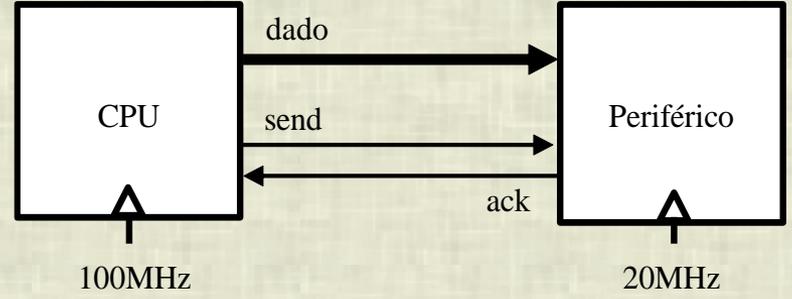
5. Dado o modelo de representação do exercício anterior, produza descrições neste modelo para os demais estilos de comunicação vistos em aula
6. Seja um barramento de 16 bits e uma memória com tempo de acesso de 100ns. Considerando que em um barramento *síncrono* há uma transição a cada período de relógio (50 ns), e em um barramento *assíncrono* há uma transição a cada 40 ns
 - Explique o protocolo ilustrado no diagrama de tempos abaixo
 - Dê a largura de banda para a transferência síncrona, em Mbps (Megabits por segundo)
 - Dê a largura de banda para a transferência assíncrona, em Mbps



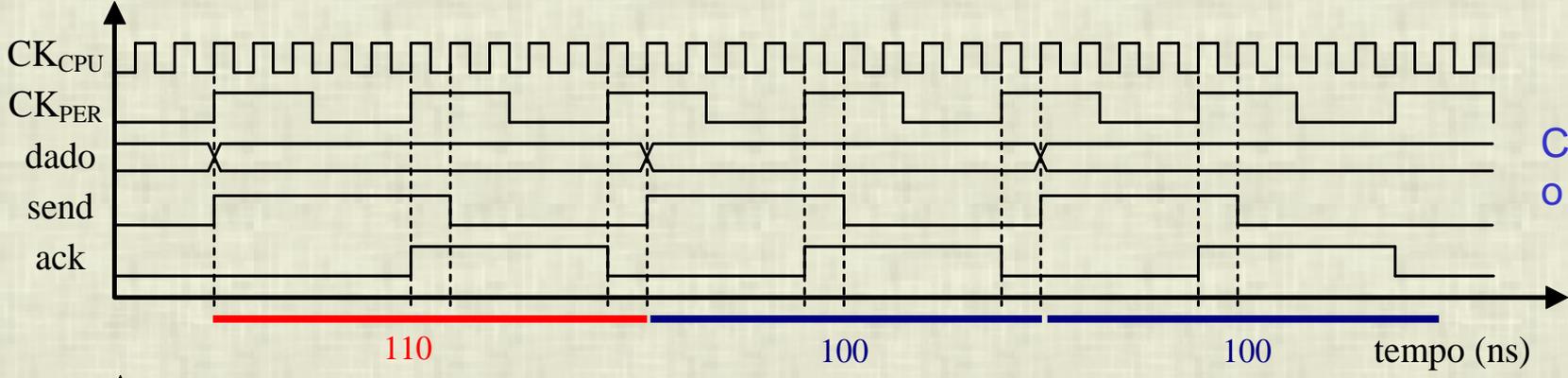
Exercício Resolvido

- 7. No modelo assíncrono de transferência de dados, considerando comunicação unidirecional e sinais de controle *send/ack*, qual o intervalo de tempo teórico para efetuar a transmissão do primeiro dado da CPU para o periférico? Como ficam os demais dados?**
- Apresente o diagrama de blocos e diagramas elétricos justificando a resposta
 - Dados: a CPU tem *clock* de 100 MHz e o periférico tem *clock* de 20 MHz
 - Considere que tanto a CPU quanto o periférico são máquinas síncronas, com registradores sensíveis à borda de subida do *clock*

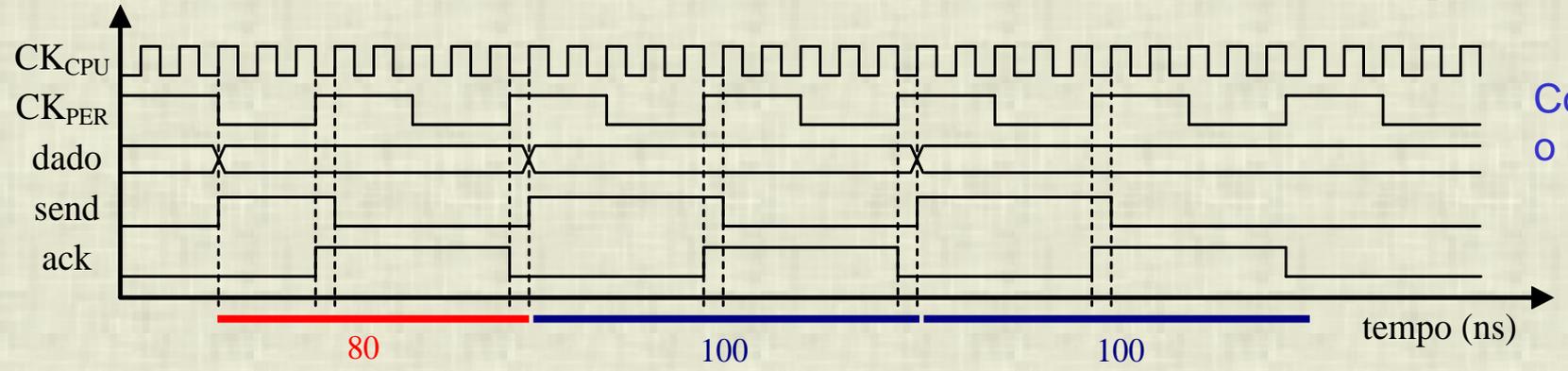
Resposta



7. RESPOSTA



Considerando o pior caso



Considerando o melhor caso

$T_{CPU} = 1/f_{CPU} = 1/100MHz = 1/100 \cdot 10^6 = 10^{-8} = 10 \text{ ns}$

$T_{Periférico} = 1/f_{Periférico} = 1/20MHz = 1/20 \cdot 10^6 = 5 \cdot 10^{-8} = 50 \text{ ns}$

Intervalo considerado para comunicação assíncrona do primeiro dado: [80ns, 110ns]

Tempo para transmissão dos dados subsequentes

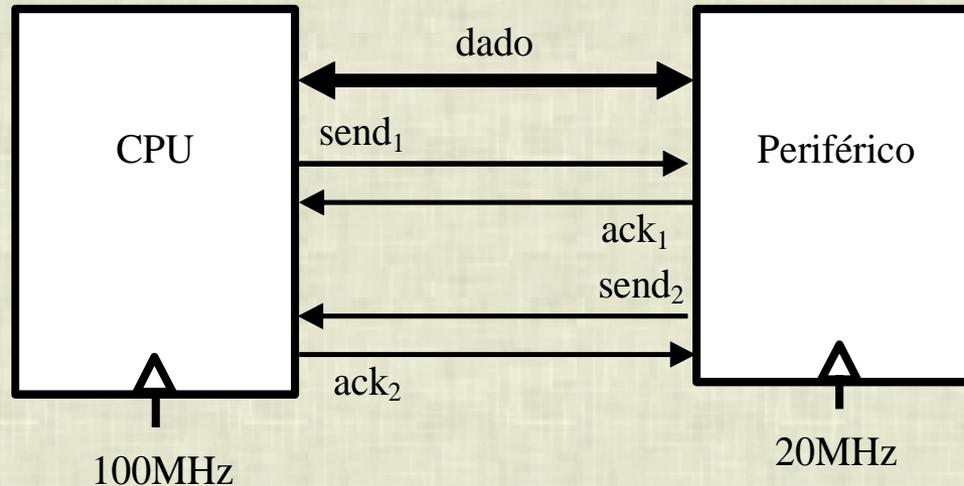
100 ns (dois pulsos de relógio da máquina mais lenta)

Exercício Resolvido

8. Considere os mesmos equipamentos do exercício 7 com uma comunicação bidirecional. Calcule o tempo necessário para a CPU transmitir um pacote de 100 dados e receber um pacote de 2 dados em resposta, contendo informações de *ACK*, *NACK* ou *ABORT*. Considere que após receber o último dado do pacote, são necessários 5 ciclos para o periférico responder

Resposta

8. RESPOSTA



- Como pode se observar no exercício anterior, o tempo necessário para a transmissão do primeiro dado na comunicação assíncrona para este problema está no intervalo [80 ns, 110 ns]. Para os demais dados, o tempo é de 100 ns. Levando em consideração que o pacote leva 5 ciclos para ser analisado pelo periférico, o tempo total levando em consideração comunicação e processamento é dado pela fórmula abaixo:
- Total = intervalo_transmissãoCPU→Periférico + ComputaçãoPeriférico + intervalo_transmissãoPeriférico→CPU
- Total = ([80 ns, 110 ns] + 99 * 100 ns) + 5 * 50 ns + ([80 ns, 110 ns] + 100 ns)
- Total = [9.980 ns, 10.010 ns] + 250 ns + [180 ns, 210 ns]
- Total = [10.420 ns a 10.470 ns]

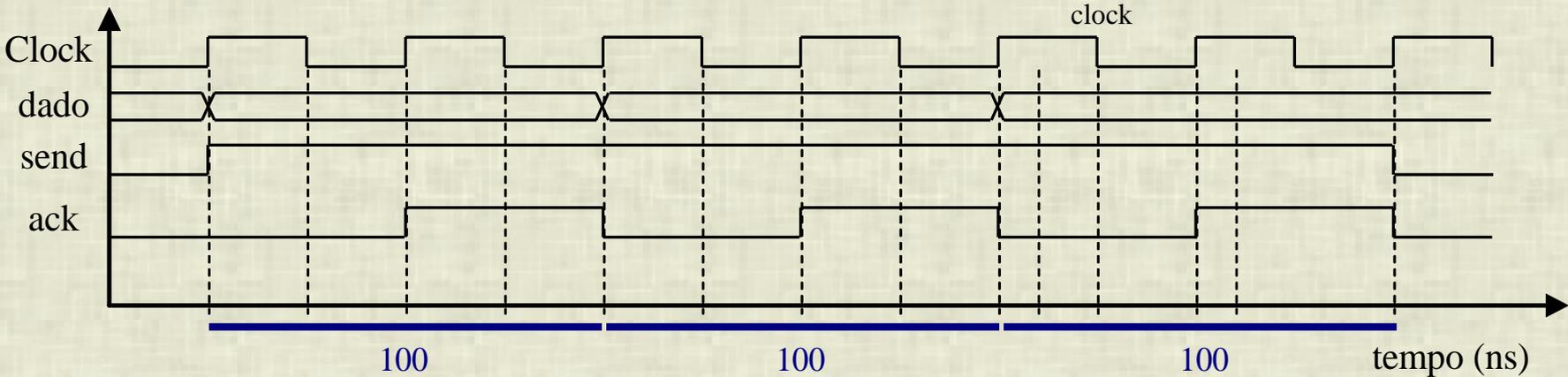
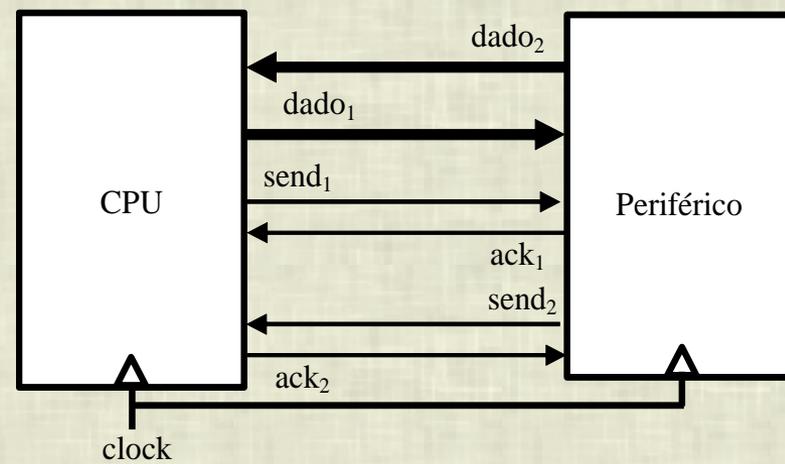
Exercício Resolvido

9. Considere o mesmo sistema do exercício anterior, mas agora a comunicação é conforme o modelo de transferência semi-síncrona. Calcule o novo tempo necessário

- Apresente o diagrama de blocos e o diagrama elétrico justificando a resposta
- Faça os cálculos considerando que a CPU e periférico são controladas por um clock de 100 MHz. Faça o mesmo para clock de 20 MHz
- Para os dois casos acima compare com os resultados do exercício 5

Resposta

9. RESPOSTA



- Ao contrário da implementação assíncrona, o primeiro dado é lido no mesmo tempo dos demais

Para clock de 20 MHz

- Total = intervalo_transmissãoCPU→Periférico + ComputaçãoPeriférico + intervalo_transmissãoPeriférico→CPU
- Total = (100 * 100 ns) + 5 * 50 ns + (2 * 100 ns) = 10.000 ns + 250 ns + 200 ns = 10.450 ns

Para clock de 100 MHz

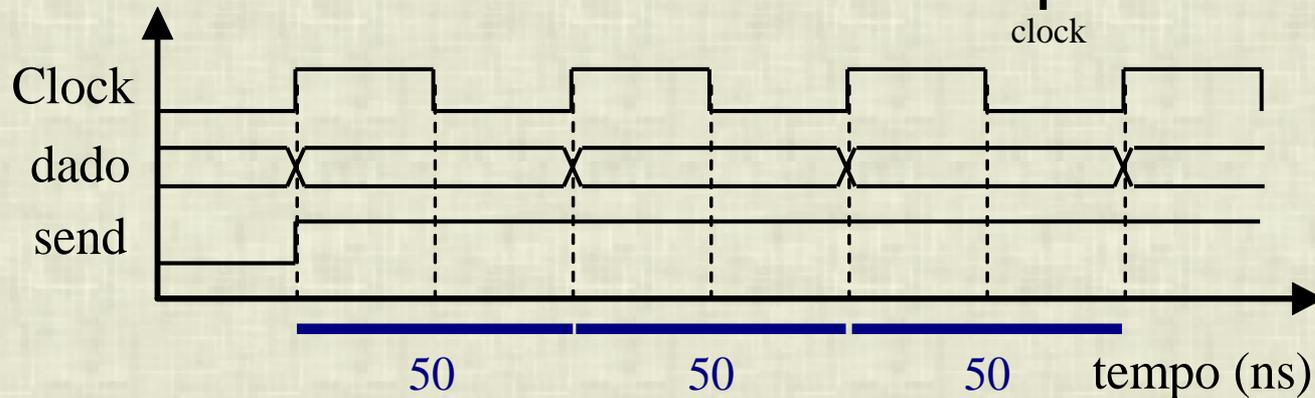
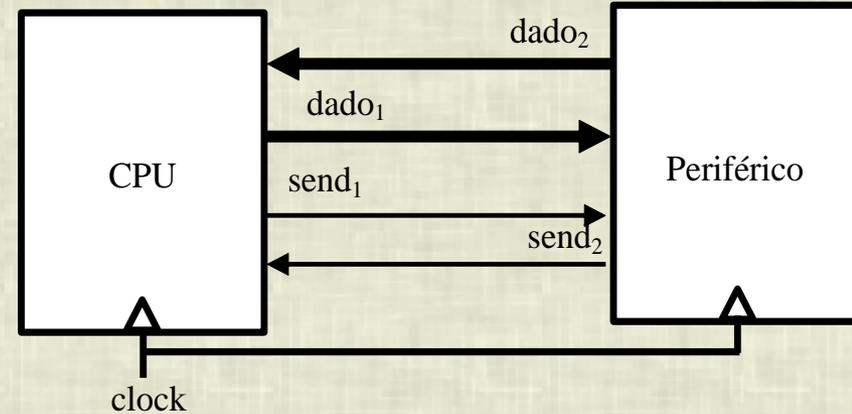
- Total = intervalo_transmissãoCPU→Periférico + ComputaçãoPeriférico + intervalo_transmissãoPeriférico→CPU
- Total = (100 * 20 ns) + 5 * 10 ns + (2 * 20 ns) = 2.000 ns + 50 ns + 40 ns = 2.090 ns

Exercício Resolvido

10. Considere o mesmo sistema do exercício anterior, mas agora a comunicação é síncrona. Calcule o novo tempo necessário

- Apresente o diagrama de blocos e o diagrama elétrico justificando a resposta
- Faça os cálculos considerando que a CPU e periférico são controladas por um clock de 100 MHz. Faça o mesmo assim, mas agora para clock de 20 MHz
- Para os dois casos acima compare com os resultados do exercício acima

Resposta



- Na implementação síncrona, pode ser enviado um dado a cada clock, o que potencializa para a metade do tempo de comunicação

Para clock de 20MHz

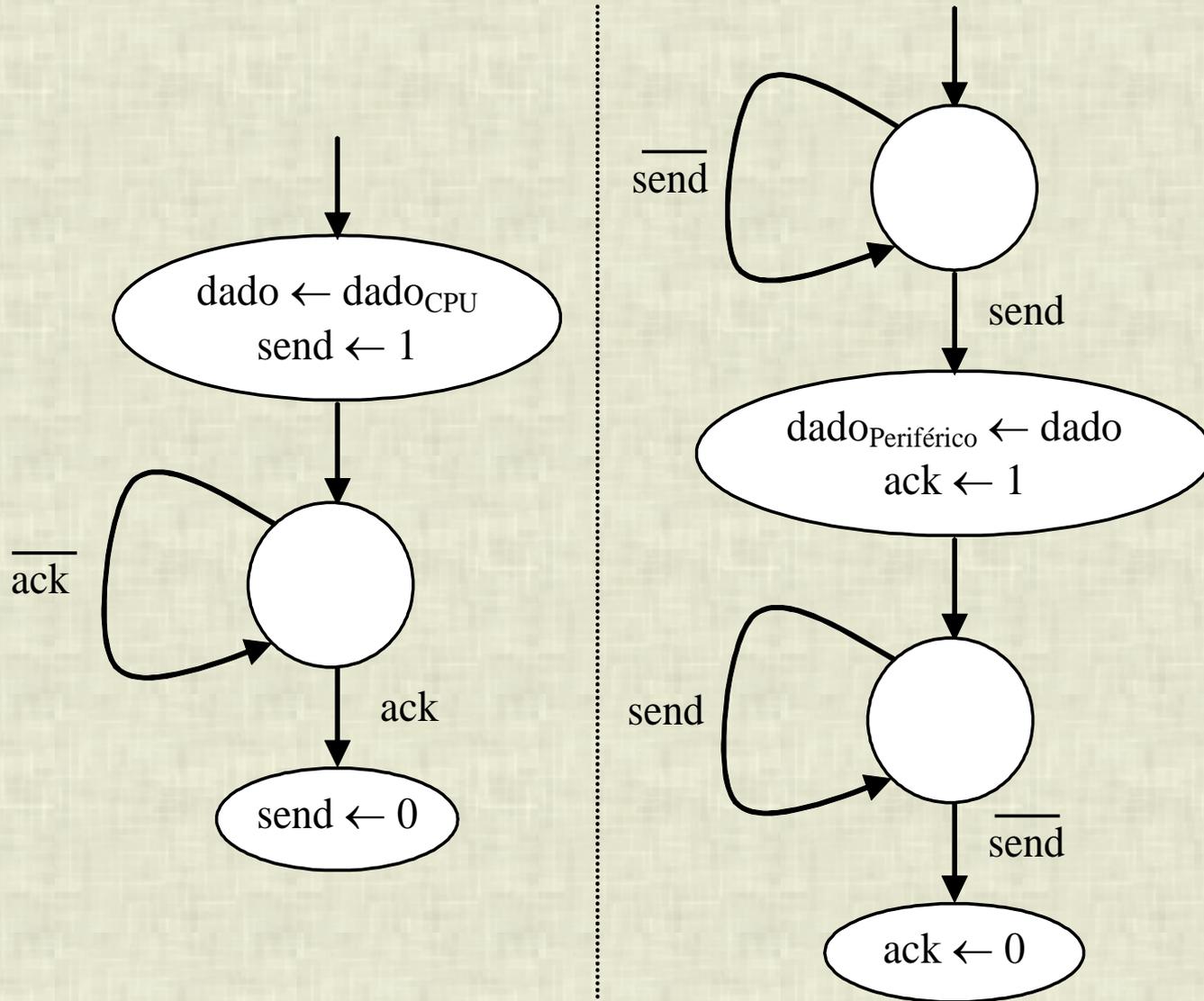
- Total = intervalo_transmissãoCPU→Periférico + ComputaçãoPeriférico + intervalo_transmissãoPeriférico→CPU
- Total = (100*50 ns) + 5 * 50 ns + (2*50 ns) = 5.000 ns + 250 ns + 100 ns = 5.350 ns
- Para clock de 100 MHz
- Total = intervalo_transmissãoCPU→Periférico + ComputaçãoPeriférico + intervalo_transmissãoPeriférico→CPU
- Total = (100 * 10 ns) + 5 * 10 ns + (2 * 10 ns) = 1.000 ns + 50 ns + 20 ns = 1.070 ns

Exercício Resolvido

11. Faça um diagrama de estados para uma comunicação assíncrona unidirecional

Resposta

11. RESPOSTA



Exercícios Adicionais

12. **Faça um diagrama de estados para uma comunicação assíncrona half-duplex**
13. **Faça um diagrama de estados para uma comunicação assíncrona full-duplex**
14. **Faça uma implementação VHDL dos dois exercícios acima**
15. **Faça um diagrama de estados para uma comunicação síncrona half-duplex**
16. **Compare todos os modelos de troca de dados analisando possíveis vantagens e desvantagens**