

# EA075

## Comunicações: Protocolos



Faculdade de Engenharia Elétrica e de Computação (FEEC)  
Universidade Estadual de Campinas (UNICAMP)

Prof. Rafael Ferrari

(Documento baseado nas notas de aula do Prof. Levy Boccato)

# Introdução

- O processo de comunicação entre dispositivos pode ser realizado em camadas:
  - Esta ideia reduz a complexidade do protocolo de comunicação através da divisão em partes de fácil desenvolvimento e compreensão.
  - Estrutura hierárquica na qual uma camada fornece serviços para a camada superior e recebe serviços da camada inferior.
  - Fornece serviços tanto de baixo nível como para alto nível.
    - O baixo nível pode lidar com bits, enquanto o alto nível tende a trabalhar com pacotes de dados.
  - **Camada física:** corresponde ao meio utilizado para o transporte de dados de um dispositivo a outro.

# Introdução

- Modos de comunicação:

- **Simplex**



- **Half-duplex**

- Dados podem fluir nas duas direções, mas apenas em uma direção a cada instante de tempo.



- **Full-duplex**

- Dados podem fluir nas duas direções simultaneamente.



# Introdução

- **Estratégias de comunicação:**

- **Serial:** camada física transporta um bit de cada vez.
- **Paralela:** camada física capaz de transportar vários bits de dados.
- **Sem fio:** não há necessidade de conexões por fios entre os dispositivos para o transporte de dados.

# Introdução

- **Estratégias de comunicação:**
  - **Síncrona:** o receptor inicialmente entra em sincronismo com o relógio do transmissor, o qual é incorporado ao fluxo de dados transmitidos.
    - Isto permite que o receptor mantenha o sincronismo durante a recepção de longas mensagens.
  - **Assíncrona:** o receptor não inicia a detecção até que ele receba um bit de partida, o qual, usualmente, tem o nível de tensão oposto ao de descanso e permite que o receptor entre em sincronismo com o transmissor para os dados subsequentes.

# Introdução

- **Comunicação paralela**

- Múltiplas linhas de dados e de controle (1 bit por linha).
- Tipicamente utilizado para conexão entre dispositivos no mesmo CI ou na mesma placa de circuito impresso.
- Alto custo, grande volume.
  - Isolamento para reduzir a interferência entre as trilhas aumenta o volume.
- Barramento deve ser curto:
  - Trilhas paralelas longas resultam em altos valores de capacitância que necessitam de mais tempo para carregar/descarregar.
  - Problemas de alinhamento entre os dados nas trilhas aumenta com o comprimento da linha.

# Introdução

- **Comunicação serial**

- Baseada no uso de linhas de dados simples (possivelmente com linhas de energia e de controle, além de uma para dados).
- Palavras são transmitidas um bit de cada vez.
- Grandes quantidades de dados podem ser transmitidas ao longo de distâncias maiores.
  - Capacitância média menor – conseqüentemente, mais bits transmitidos por unidade de tempo para o mesmo gasto de energia.
- Mais barata e ocupa um volume menor.
- Interface lógica (circuitos) e protocolos de comunicação mais complexos.
  - É preciso quebrar palavras em bits, sequenciar a transmissão e reagrupar os bits para compor as palavras novamente.
  - Sinais de controle e de dados podem ser multiplexados na mesma linha (trilha), o que aumenta a complexidade do protocolo de comunicação.

# Introdução

- **Comunicação sem fio**

- **Infravermelho:**

- Onda luminosa cuja frequência é menor do que o espectro de luz visível.
- Na geração do sinal, um diodo emite luz infravermelha.
- Um transistor infravermelho detecta o sinal e conduz quando é exposto à luz infravermelha.
- Opção de baixo custo de construção, porém necessita de linha de visada (alcance limitado).

- **Rádio frequência (RF):**

- Ondas eletromagnéticas cujas frequências estão contidas no espectro de rádio.
- Circuitos analógicos e antenas são necessários em ambos os lados da transmissão.
- Não há necessidade de linha de visada.
- O alcance é determinado pela potência de transmissão.

# Introdução

- **Detecção e correção de erros**

- Correção de erro: habilidade do receptor e do transmissor de trabalharem em cooperação para que um evento de erro seja revertido.
  - Usualmente é realizado por um protocolo que inclui reconhecimento / retransmissão.
- Muitos protocolos trazem algum mecanismo para descobrir a ocorrência de erros na transmissão e, até mesmo, para corrigir tais erros.
- **Paridade:** bit extra é enviado juntamente com a palavra para a detecção de erro.
  - Capaz de detectar a ocorrência de erro em um único bit; insensível a um número par de erros de bits.
- **Soma (*checksum*):** palavra extra enviada com o pacote de dados formado por muitas palavras.
  - Exemplo: palavras extras contêm a soma (XOR) de todas as palavras de dados do pacote.

# Protocolos Seriais

- **Serial Peripheral Interface (SPI):**

- Foi desenvolvido pela Motorola para oferecer uma interface simples e de baixo custo entre microcontroladores e *chips* periféricos.
- Usado para interface com memórias, conversores analógico-digital, relógios de tempo real, *displays* de LCD e mesmo com outros processadores.
- SPI é um protocolo síncrono em que todas as transmissões são referenciadas a um relógio comum, gerado pelo mestre (processador).
- O periférico (escravo) usa o *clock* (SCK) para sincronizar a aquisição do fluxo de dados.

# Protocolos Seriais

- **Serial Peripheral Interface (SPI):**

- Quatro fios / pinos

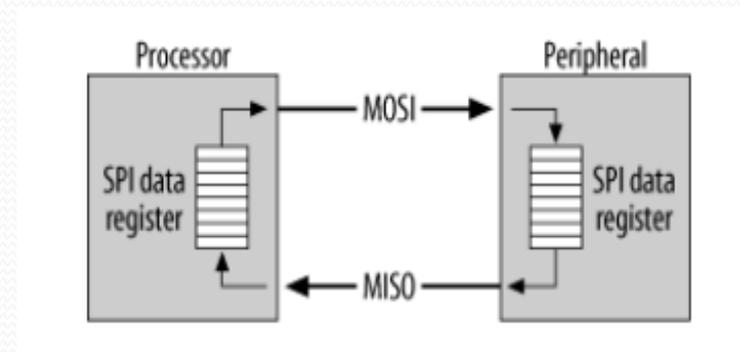


- MOSI (*master out slave in*) – é gerado pelo mestre e recebido pelo escravo.
- MISO (*master in slave out*) – é gerado pelo escravo, mas sua geração é controlada ou acionada pelo mestre.
- SCK – *clock serial*.
- CS' - *chip select*.

# Protocolos Seriais

- **Serial Peripheral Interface (SPI):**

- Vários chips podem estar conectados à mesma interface SPI do mestre.
- O mestre seleciona um escravo para uma transmissão acionando a entrada CS' daquele escravo.
- Tanto o mestre quanto o escravo contêm um registrador de deslocamento. O mestre inicia a transferência de um *byte* ao escrevê-lo neste registrador SPI.
- À medida que o registrador transmite o *byte* para o escravo através da linha MOSI, o escravo transmite o conteúdo de seu registrador de deslocamento de volta para o mestre pela linha MISO. Deste modo, o conteúdo dos dois registradores é trocado.
- Uma operação de leitura pode ser feita simultaneamente com uma operação de escrita no escravo (*full-duplex*).



# Protocolos Seriais

- **Serial Peripheral Interface (SPI):**

- Alguns periféricos (chips de memória com interface SPI, por exemplo) suportam a transferência de múltiplos bytes em sequência. Para isto, o *chip select* deste escravo deve permanecer ativo durante toda a transmissão.
  
- **Quatro modos de operação possíveis de SPI:**

	Clock de fase zero	Clock de fase um
Clock de polaridade ALTA (SCK inativo em 1, ativo em 0)	As saídas são válidas na borda de descida do SCK	As saídas são válidas na borda de subida do SCK
Clock de polaridade BAIXA (SCK inativo em 0, ativo em 1)	As saídas são válidas na borda de subida do SCK	As saídas são válidas na borda de descida do SCK

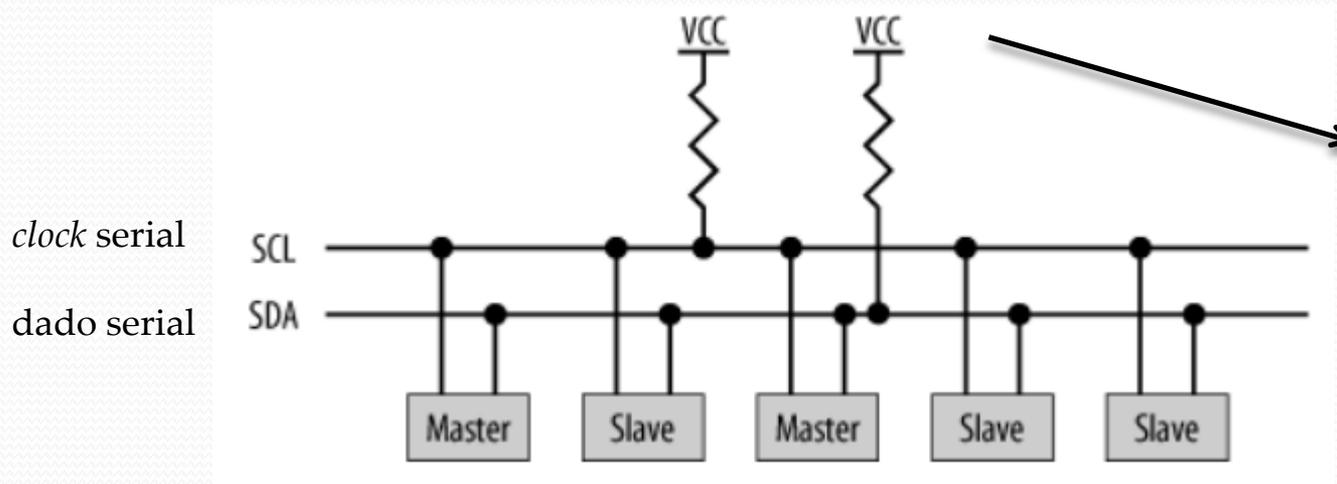
# Protocolos Seriais

- **I<sup>2</sup>C (Inter-Integrated Circuit):**

- Projetado pela Philips Semiconductors há mais de 20 anos, é um protocolo de barramento serial com dois fios.
- O barramento é bidirecional, de baixa velocidade e síncrono com um *clock* em comum.
- Dispositivos podem ser acrescentados ou removidos do barramento I<sup>2</sup>C sem afetar os demais.
- I<sup>2</sup>C é um protocolo de barramento multimestre: mais do que um dispositivo pode assumir o papel de mestre do barramento.
- Cada dispositivo conectado ao barramento I<sup>2</sup>C tem um endereço único e pode operar como um transmissor (mestre do barramento), um receptor (escravo) ou ambos.

# Protocolos Seriais

- I<sup>2</sup>C (Inter-IC):



Resistor de pull-up que garante que as linhas ficam no nível lógico ALTO quando ociosas.

Os dispositivos conectados ao barramento ou deixam a linha em seu nível normal ou forçam seu nível lógico para BAIXO.

- I<sup>2</sup>C compartilha a mesma linha para a transmissão do mestre ao escravo e para a resposta do escravo (multiplexação no tempo).

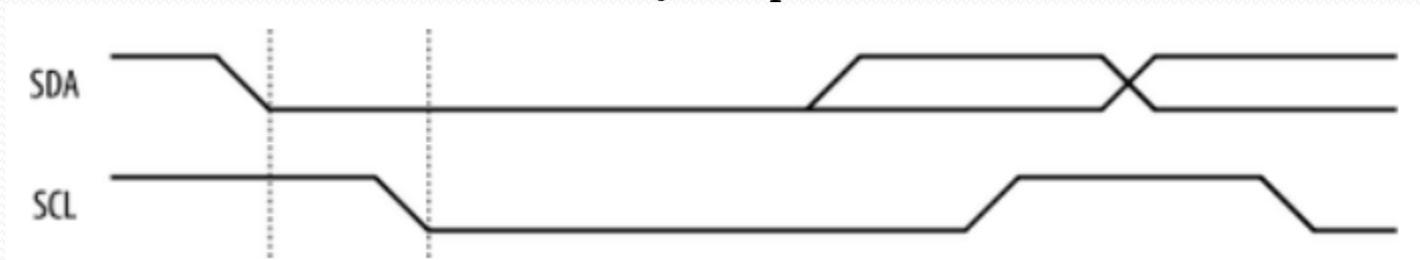
# Protocolos Seriais

- I<sup>2</sup>C (Inter-IC):

- Como é feita a comunicação via I<sup>2</sup>C?

- Quando inativos, SDA e SCL ficam no nível ALTO.
- A transferência **começa** com uma transição de ALTO para BAIXO (borda de descida) do sinal DAS enquanto SCL está ALTO), seguido de SCL indo para BAIXO.

Condição de partida



- Isto indica a todos os receptores no barramento que um pacote de transmissão está em seu início. Enquanto SCL está BAIXO, SDA recebe o valor (ALTO ou BAIXO) do primeiro bit válido de dado.

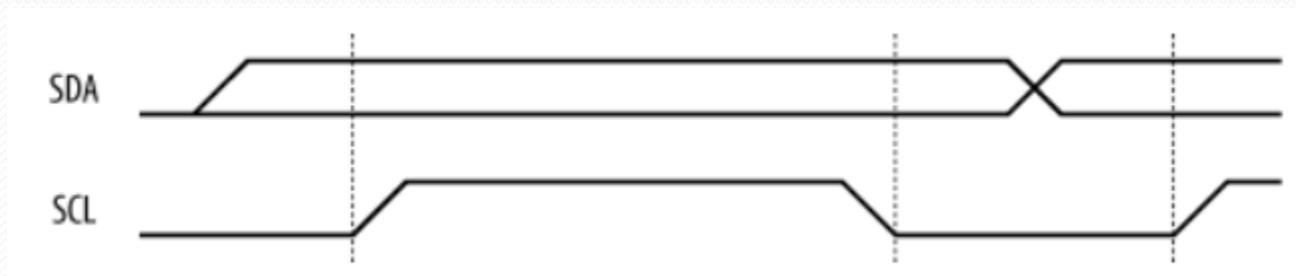
# Protocolos Seriais

- **I<sup>2</sup>C (Inter-IC):**

- Como é feita a comunicação via I<sup>2</sup>C?

- Cada bit a ser transmitido precisa ser colocado na linha SDA enquanto SCL está BAIXO. O bit é, então, amostrado na borda de subida do SCL e deve permanecer válido até que SCL vá para BAIXO novamente.

Transmissão de cada bit

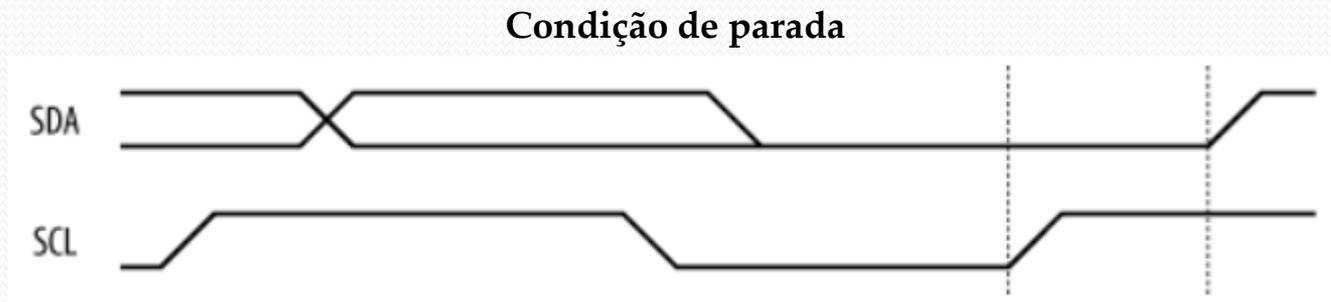


# Protocolos Seriais

- I<sup>2</sup>C (Inter-IC):

- Como é feita a comunicação via I<sup>2</sup>C?

- A transferência **termina** com uma transição de SDA de nível BAIXO para ALTO (borda de subida) enquanto SCL está em nível ALTO.



# Protocolos Seriais

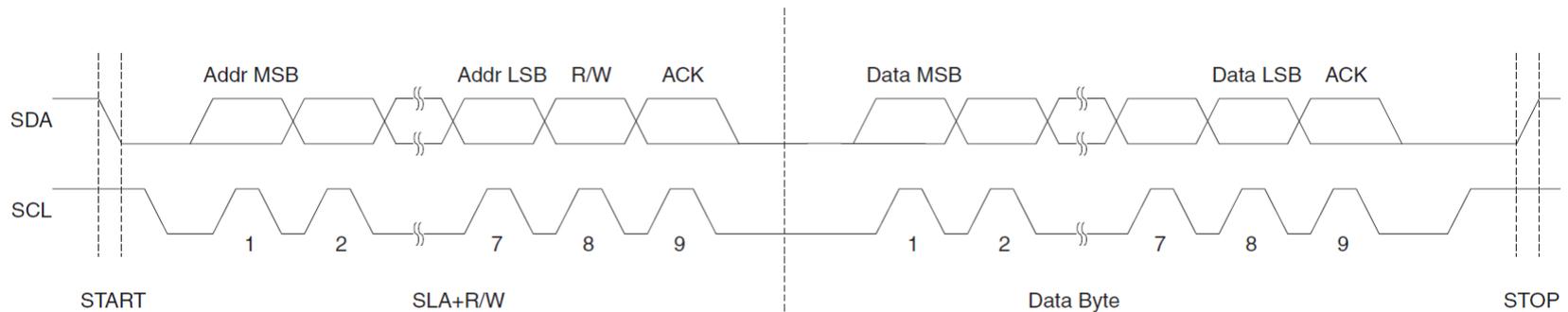
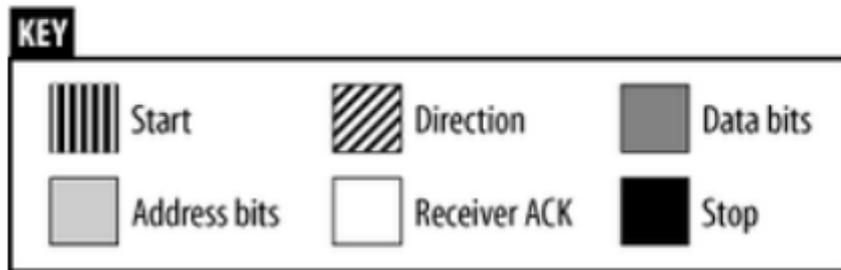
- I<sup>2</sup>C (Inter-IC):

- Como é feita a comunicação via I<sup>2</sup>C?

- Cada *byte* transmitido deve ser confirmado pelo receptor.
- Após a transmissão do oitavo bit de dado, o mestre libera a linha SDA e gera um pulso adicional de *clock* em SCL. Isto aciona o receptor, que, então, deve confirmar o recebimento do *byte* colocando a linha SDA em BAIXO.
- Se o receptor não faz isto, o mestre aborta a transmissão e toma medidas apropriadas de tratamento de erro.
- Qualquer número de *bytes* pode ser transmitido em um pacote I<sup>2</sup>C. Se o receptor está impossibilitado de receber mais *bytes*, ele aborta a transmissão segurando o *clock* em BAIXO. Isto força o transmissor a aguardar até que SCL seja liberado.

# Protocolos Seriais

- I<sup>2</sup>C (Inter-IC): Pacote de informações



R/W = 1 - Leitura  
R/W = 0 - Escrita

# Protocolos Seriais

- I<sup>2</sup>C (Inter-IC):

- Como lidar com múltiplos dispositivos e mestres?

- Dois mestres podem iniciar uma transmissão ao mesmo tempo: como SDA fica em ALTO naturalmente, o mestre que colocar um bit 1 (ALTO), mas perceber que a linha está no nível BAIXO, entenderá que há outro mestre usando o barramento e interromperá sua ação.
- Existem endereços associados a chamadas especiais de um mestre. Por exemplo, o endereço 0000000 com bit de direção 0 (escrita) indica que o mestre deseja transmitir o byte para todos os escravos conectados ao barramento (*broadcast*).

# Protocolos Seriais

- **I<sup>2</sup>C (Inter-IC):**

- Transferência de dados ocorre segundo uma taxa de 100 kHz e usando endereçamento com 7 bits no modo normal.
- 3,4 MHz e 10 bits de endereçamento no modo rápido.
- Dispositivos capazes de interfacear com o barramento I<sup>2</sup>C: EEPROMS, flash, algumas memórias RAM, relógios de tempo real, temporizadores *watchdog* e microcontroladores.

# Protocolos Seriais

- **USB** (*Universal Serial Bus*):

- Conexão simples entre PC e monitores, impressoras, alto-falantes digitais, *modems*, *scanners*, câmeras digitais, *joysticks*, equipamentos de jogos multimídia.
- Taxas de dados:
  - **USB 1.0 e 1.1:** 12 Mbps para dispositivos de grande largura de banda e 1,5 Mbps para dispositivos de baixa velocidade (*joysticks*, *game pads*). **USB 2.0:** até 480Mbps. **USB 3.0:** até 5Gbps.
- Topologia hierárquica em estrela:
  - Um dispositivo USB (*hub*) conectado ao PC.
    - *Hub* pode ser embutido em dispositivos como em monitores, impressoras, teclados ou pode ser utilizado em modo *standalone*.
  - Muitos dispositivos USB podem ser conectados ao *hub*.
  - Até 127 dispositivos podem ser conectados a uma porta do computador dessa maneira.
- Controlador *host* USB:
  - Gerencia largura de banda.
  - Aloca potência necessário para o funcionamento de cada dispositivo conectado.

# Protocolos Paralelos

- **Barramento PCI** (*Peripheral Component Interconnect*):
  - Barramento de alto desempenho desenvolvido pela Intel no começo dos anos 90.
  - Padrão adotado pela indústria e administrado pela PCISIG (*PCI Special Interest Group*).
  - Interconecta chips, placas de expansão, subsistemas de memória de processadores.
  - Transferência de dados na taxa de 127,2 a 508,6 Mbits/s e 32 bits de endereçamento.
    - Especificação mais recente: inclui 64 bits e mantém compatibilidade com esquemas em 32 bits.
  - Barramento síncrono.
  - Linhas de dados/endereços multiplexadas.

# Protocolos Sem Fio

- **Bluetooth:**

- Padrão mundial de conectividade sem fio.
- Baseado em tecnologia de rádio de baixo custo e pequeno alcance.
- Não há necessidade de campo de visão – por exemplo, conexão entre PC e impressora em cômodos distintos.
  - Classe 1: 100 mW (20 dBm) até 100 metros
  - Classe 2: 2,5 mW (4 dBm) até 10 metros
  - Classe 3: 1 mW (0 dBm) ~ 1 metro
- Taxas de transmissão: 1Mbps, 3 Mbps, 24Mbps.
- Dispositivos Bluetooth operam na faixa ISM (Industrial, Scientific, Medical) centrada em 2,45 GHz.

# Protocolos Sem Fio

- **IEEE 802.11:**

- Padrão proposto para LANs *wireless*.
- Parâmetros específicos para camadas PHY e MAC
  - Camada PHY
    - ✓ Camada física.
    - ✓ Manipulação dos dados transmitidos entre nós.
    - ✓ Transferência de dados em taxas da ordem de 1 ou 2 Mbps.
    - ✓ Opera em largura em banda de frequência na faixa compreendida entre 2,4 a 2,4835 GHz (RF), ou entre 300 e 428.000 GHz (Infra-vermelho).
  - Camada MAC (*Medium access control layer*).
    - ✓ Protocolo responsável pela manutenção da ordem no compartilhamento do meio físico.
    - ✓ Possui mecanismos de redução de colisões.