

# Transmissor de Dados Assíncrono

2022

## INTRODUÇÃO

Esta experiência visa projetar e implementar um circuito transmissor de dados assíncrono. Esse circuito possui em seus componentes um tipo de **circuito sequencial**, os **registradores**. Genericamente, um registrador contém **Flip-Flops (FF)**, pois estes armazenam os dados durante a operação do circuito digital, como ocorre com os contadores vistos anteriormente. Um circuito de transmissão de dados assíncrona é parte de uma **UART** (*Universal Asynchronous Receiver Transmitter*) é um circuito transmissor-receptor universal assíncrono usado por sistemas computacionais para comunicação de dados com outros dispositivos.

OBS: Veja o **ANEXO** sobre Conceitos de Comunicação Serial Assíncrona (seções 1.1.d, 1.2 e 2).

## OBJETIVO

O objetivo dessa experiência é projetar e implementar um circuito de transmissão de dados assíncrono usando como base um tipo de registrador chamado **registrador de deslocamento**.

Ao final da experiência, espera-se que o aluno tenha aprendido:

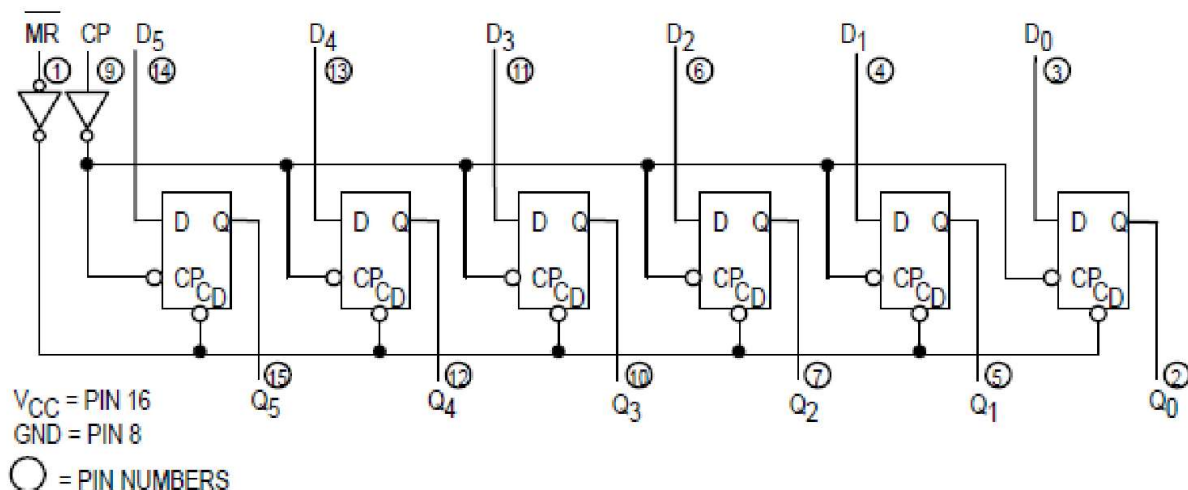
- Os conceitos de registradores de deslocamento.
- Os conceitos de uma UART.

## 1. PARTE EXPERIMENTAL

### a) Registradores

Muitos CIs são fabricados com *flip-flops* em conjunto para implementar **registradores de entrada/saída paralela**. Basicamente, estes são *flip-flops* em cascata usados para implementar registradores de vários comprimentos de bits. Por exemplo, o CI 74LS174 é um registrador de 6 bits, conforme pode ser visto na Figura 1.

Figura 1 – Diagrama Lógico do Registrador CI 74LS174.



Di – entradas de dados

Qi – saída de dados

MR (*Master Reset*) – força “0” nas saídas dos *flip-flops*

CP (*Clock Pulse*) – pulso do relógio (*Clock*)

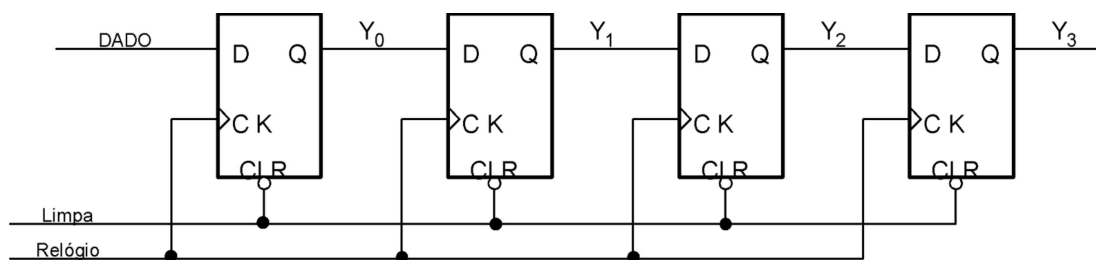
Pode-se implementar um circuito digital denominado **registrador de deslocamento** (*shift-register*) usando um ou mais **registradores de entrada/saída paralela**.

O **registrador de deslocamento** (*shift-register*) é um dos componentes mais utilizados em sistemas digitais. Tal fato pode ser constatado em muitas calculadoras portáteis, onde cada dígito de um número é deslocado para a esquerda, à medida que outro é inserido. Além da característica de deslocamento, tais componentes apresentam a característica de memorização, da mesma maneira que os biestáveis.

Eles são classificados como circuitos lógicos sequenciais, uma vez que são construídos a partir de biestáveis.

A Figura 2 apresenta um registrador de deslocamento de 4 bits, construído a partir de biestável tipo D. Neste exemplo, poder-se-ia usar o componente 74LS174, usando apenas 4 *flip-flops* contidos no CI.

**Figura 2 – Registrador de Deslocamento de 4 bits, com Deslocamento para a Direita e Carga em Série.**

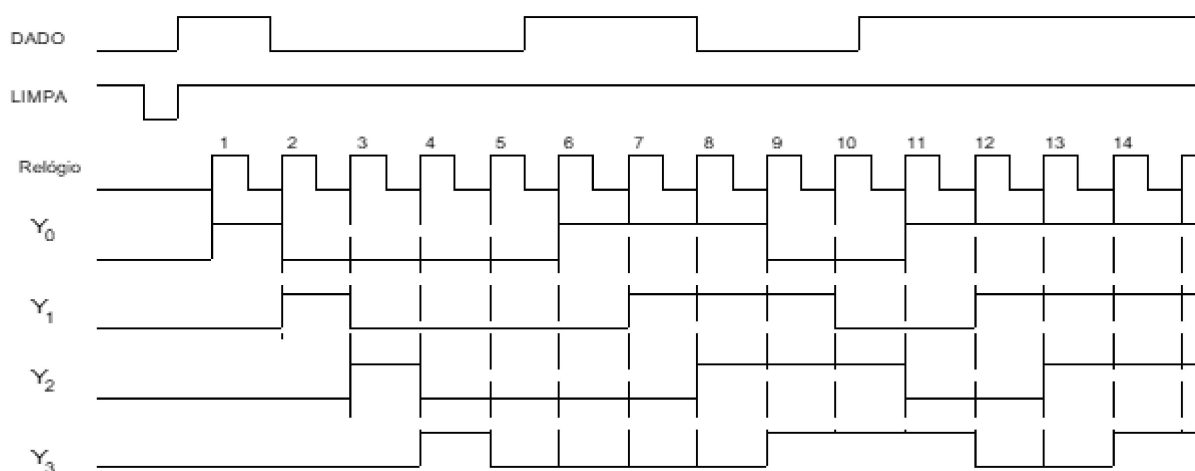


A carta de tempo com as formas de onda dos principais sinais é apresentada na Figura 3.

Como se pode observar, a cada pulso de relógio, o dado é deslocado para a direita. Este tipo de registrador de deslocamento é conhecido como **registrador de deslocamento à direita, com carga em série**.

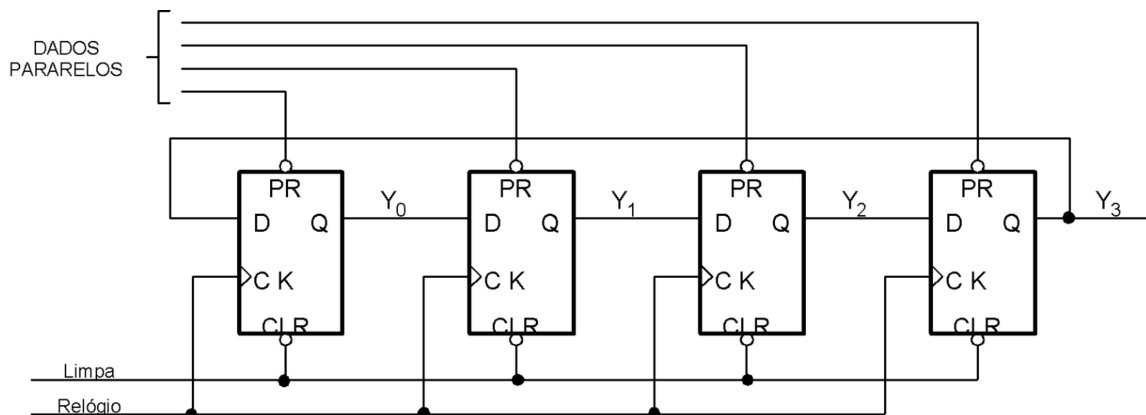
A desvantagem destes tipos de registrador é que são consumidos muitos pulsos de relógio (*Clock*) para se recarregar um estado inicial qualquer.

**Figura 3 – Carta de Tempos do Circuito da Figura 2.**



Pode-se construir, no entanto, **registradores de deslocamento com carga em paralelo**, como mostra a Figura 4.

**Figura 4 – Registrador de Deslocamento de 4 bits, com Deslocamento para a Direita e Carga Paralela.**



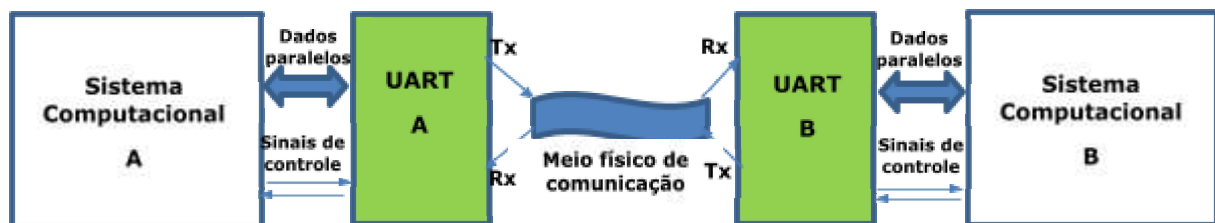
Os sinais de PRESET (PR) dos biestáveis são utilizados para a carga dos dados, sempre após o sinal "Limpa" (CLR) ter sido gerado.

Da mesma forma como ocorre com os contadores, pode-se encontrar registradores de deslocamento integrados, como por exemplo, os CI 7494, 74194, 74166, 74178, etc.

### 1.1. Projeto de um Transmissor de Dados Assíncrono

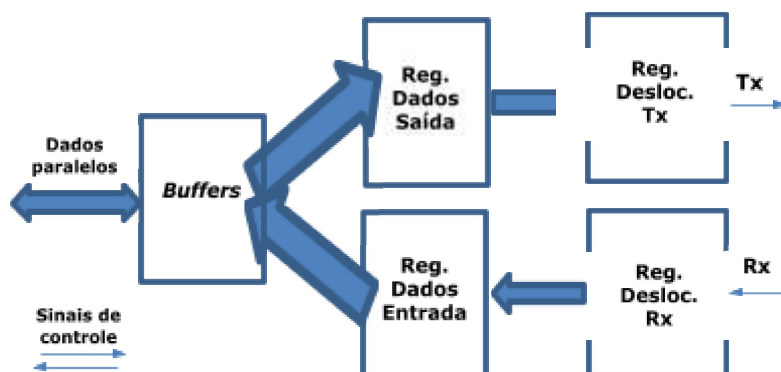
Um sistema computacional trabalha internamente com dados que trafegam em barramentos paralelos. Por razões tecnológicas, no momento em que o circuito vai comunicar os dados processados, esses dados são transmitidos serialmente para um dispositivo externo (ou outro sistema computacional) usando um meio físico de comunicação (pares de cabos telefônicos, cabo coaxial, fibra óptica, etc). A Figura 5 apresenta esse esquema de comunicação de dados com a UART.

**Figura 5 – Esquema de comunicação de dados com UART.**



Uma UART simplificada pode ser representada em blocos conforme é apresentada na Figura 6.

**Figura 6 – Diagrama de blocos de uma UART simplificada.**

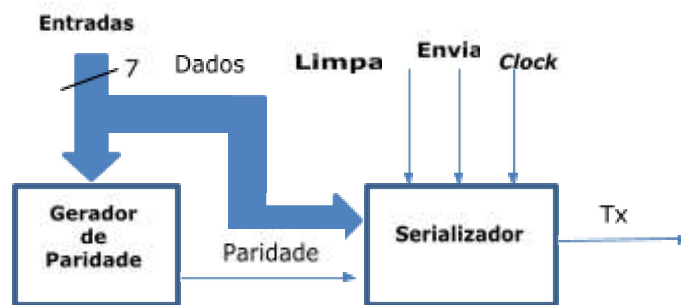


Fonte: adaptado de Floyd (2007).

Os registradores de deslocamento serializam os dados para transmissão (Tx) ou separam os bits durante a sua recepção (Rx). Os registradores de entrada e de saída armazenam temporariamente os dados recebidos ou a serem transmitidos, respectivamente. Os *buffers* são interfaces entre os registradores de dados e o barramento de dados do sistema computacional. Os sinais de controle permitem implementar um protocolo de comunicação estabelecido.

O diagrama de blocos de um Transmissor de Dados Assíncrono é mostrado na Figura 7. O bloco Gerador de Paridade gera um bit de paridade (par/ímpar definido previamente) e o bloco Serializador implementa o protocolo assíncrono mostrado no ANEXO (seção 1.2) e gera a sequência de 11 bits na saída TX. Os 11 bits são formados na seguinte sequência: 1 *start bit* (igual a 0), 7 bits de dados, 1 bit de paridade e 2 *stop bits* (iguais a 1). Os sinais Limpa, Envia e Clock controlam o funcionamento do Serializador.

**Figura 7 – Diagrama de Blocos do Transmissor de Dados Assíncrono.**



Projete um circuito um Transmissor de Dados Assíncrono conforme descrito anteriormente, de maneira que 7 *bits* de Dados e 1 *bit* de Paridade sejam transmitidos na saída TX (1 bit a cada pulso de **Clock**). Os bits devem ser transmitidos após o comando **Envia**. A transmissão deve parar a cada 11 bits transmitidos no total, incluindo-se 1 *start bit* e 2 *stop bits*. O comando **Limpa**, zera o circuito, deixando o circuito pronto para realizar o próximo envio de dados. A transmissão deve obedecer ao protocolo assíncrono mostrado no ANEXO (seção 1.2). Os dados transmitidos devem ser apresentados no terminal da bancada.

OBS: Não é necessário projetar os buffers e os registradores de dados da Figura 6.

- Elabore um Diagrama de Blocos do transmissor.
- Elabore um Diagrama Lógico do transmissor.
- Simule o transmissor na ferramenta Quartus.
- Elabore uma Tabela de Testes do transmissor, incluindo sinais intermediários.
- Insira o Diagrama de Blocos, Diagrama Lógico (não esqueça dos dois carimbos de identificação e de posicionamento do diagrama), a Carta de Tempos (simulação) e a Tabela de Testes no Planejamento.

## 1.2. Implementação dos circuitos

- Monte gradualmente o transmissor no Painel de Montagens.

*Dica: Integre cada componente testado isoladamente aos outros já testados.*

- Realize os testes do transmissor. Use a Tabela de Testes do Planejamento.  
OBS: Faça inicialmente os testes estáticos (manual) e, em seguida, os testes dinâmicos (c/ gerador de funções). Caso algum resultado obtido seja diferente do resultado esperado, proceda a depuração do circuito, usando os sinais intermediários do circuito. Caso tenha corrigido o defeito, repita os testes.
- Uma vez testado o transmissor, ligue a saída Tx em um terminal da bancada. Em seguida, transmita alguns dados e verifique se estão sendo apresentados corretamente no terminal.
- Anote e comente os resultados.

## 1.4. Desafio

O professor irá propor um desafio sobre esta experiência.

## 2. BIBLIOGRAFIA

- FREGNI, E.; SARAIVA, A. M. **Engenharia do Projeto Lógico Digital: Conceitos e Prática**. Editora Edgard Blücher, 1995.
- TEXAS INSTRUMENTS. **The TTL Logic Data Book**, 1994.
- TOCCI, R. J.; WIDMER, N.S.; MOSS, G.L. **Sistemas Digitais: Princípios e Aplicações**. Prentice-Hall, 11ª ed., 2011.
- WAKERLY, John F. **Digital Design Principles & Practices**. 4th edition, Prentice Hall, 2006.
- FLOYD, T. L. **Sistemas Digitais: Fundamentos e Aplicações**. trad. José Lucimar do Nascimento. 9ª. ed. Porto Alegre. RG. Brasil. Bookman. 2007.
- ALMEIDA, F. V.; SATO, L.; MIDORIKAWA, E. Tutorial Esquemático Quartus Prime 16.1, versão 1.0, 2017.

## 3. RECURSOS NECESSÁRIOS

- ✓ 1 Computador com MS-Windows
- ✓ 1 Painel de montagens de circuitos digitais
- ✓ 1 Fonte de 5 volts
- ✓ 1 Multímetro
- ✓ 1 Gerador de funções
- ✓ 1 ferramenta Quartus Prime 16.1 (ou versão mais recente)
- ✓ Portas básicas TTL
- ✓ Registradores de deslocamento (*Shift Registers*)
- ✓ Contadores
- ✓ Geradores de paridade
- ✓ Cabos, fios e jumpers

## ANEXO – Conceitos de Comunicação Serial Assíncrona (2019)

### Histórico de Revisões

Profs. Kechi Hirama, Glauber De Bona, Anarosa A F Brandão – versão 2021.

Profs. Kechi Hirama, Renan Cerqueira Afonso Alves - versão 2022.