

Transmissão Serial Assíncrona

Versão 2024

OBJETIVOS

Após a conclusão desta experiência, os seguintes tópicos devem ser conhecidos pelos alunos:

- Comunicação serial assíncrona (RS232C);
- Conversores de nível de tensão;
- Metodologia de Projeto de Circuitos Digitais com FPGA;
- Projeto de circuitos com Verilog;
- Depuração com ferramentas.

RESUMO

O objetivo desta experiência é projetar circuitos digitais para comunicação serial de dados (transmissão de dados) com um terminal de dados, utilizando a norma EIA-RS-232C e o código ASCII (*American Standard Code Information Interchange*). A parte experimental consiste no projeto e na implementação de um circuito digital que envia dados digitais (caracteres em código ASCII) para um terminal serial usando a linguagem de descrição de hardware Verilog e uma placa de desenvolvimento FPGA DE0-CV.

1. PARTE EXPERIMENTAL

A parte experimental envolve a realização de um conjunto de atividades visando o desenvolvimento de um circuito digital para a transmissão de dados para um terminal serial, usando a placa de desenvolvimento FPGA DE0-CV.

Esta experiência foi planejada para ser executada de forma presencial, usando os recursos disponíveis na bancada do Laboratório Digital inclui computador, uma placa FPGA DE0-CV, um Analog Discovery e outros componentes.

1.1. Atividade 1 – Estudo do Projeto Base do Circuito de Transmissão Serial

Esta primeira atividade envolve o estudo e a familiarização do projeto do **circuito base** de transmissão serial assíncrono. O arquivo `tx_serial_base.zip` fornecido contém os arquivos Verilog para o projeto com Intel Quartus Prime e simulação com ModelSim.

Especificação do Projeto Base: O circuito fornecido transmite um caractere ASCII de 8 *bits* especificado na entrada `dados_ascii` com o acionamento do sinal `partida`. Ao final da transmissão, o sinal `pronto` é acionado por um período de `clock`. A saída `saida_serial` deve ser usada para ser ligada ao terminal serial. A configuração da comunicação serial adotada no projeto é denominada 7N2, ou seja, 7 *bits* de dados, sem paridade e 2 *stop bits*, com uma taxa de comunicação de 9600 *bauds*. A figura 1 mostra a interface externa do circuito, com os sinais de entrada e saída.

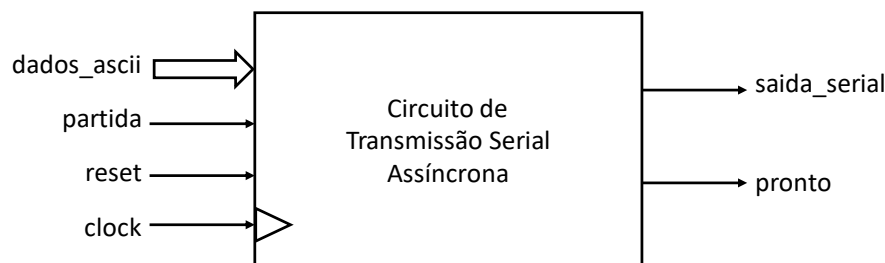


Figura 1 – Interface do Circuito Base de Transmissão Serial Assíncrona.

- a) Abrir o arquivo ZIP fornecido, crie um projeto com o Intel Quartus Prime e estudar o código Verilog do circuito.

Este estudo é muito importante para o aprendizado de técnicas de projeto e utilização eficiente de ferramentas de desenvolvimento. Por exemplo, a **técnica de superamostragem** aplicada no projeto do circuito de transmissão serial será também usada no projeto da experiência seguinte. Além disto, o ModelSim deverá ser usado não somente para a simulação e geração de formas de onda, como também para medidas e análise de parâmetros das saídas do circuito.

Alguns **aspectos essenciais** deste estudo incluem os seguintes pontos:

1. Identificar os componentes do fluxo de dados e a composição deles para formar o módulo fluxo de dados (entidade `tx_serial_7N2_fd`).
DICA: use a saída da ferramenta *RTL Viewer* para ilustrar este estudo.
2. Determinar como os *bits* de entrada são armazenados nos componentes internos do fluxo de dados.
3. Analisar a máquina de estados da unidade de controle (entidade `tx_serial_uc`) e estude os estados e as transições entre eles.
DICA: use a saída da ferramenta *State Machine Viewer* e edite a figura para incluir as condições de transição de estados e as saídas dos sinais de controle em cada estado.
4. Verificar como o sinal interno de *tick* é gerado e como este sinal é usado pela unidade de controle para determinar os instantes em que o sinal de saída serial é modificado.

Os diagramas gerados são importantes para as modificações a serem realizadas no projeto da Atividade 2 a seguir.

- b) Simular o funcionamento do projeto com ModelSim, e anexar as formas de onda obtidas no Planejamento. Use os casos de testes do *testbench* fornecido para verificar o correto funcionamento do circuito completo. Se julgar necessário, defina e acrescente no projeto alguns **sinais de depuração**.

DICA: um modelo de *testbench* (`tx_serial_7N2_tb.vhd`) é fornecido como base para o desenvolvimento das simulações com o ModelSim. Por exemplo, a figura 2 ilustra uma forma de onda para o caso de teste de transmissão do caractere 5 (código ASCII 35H), com acréscimo de sinais de depuração (sinal de *tick* e *estado* da unidade de controle). A partir da análise das formas de onda obtidas da simulação, podemos identificar o *start bit*, os bits de dados e os 2 *stop bits* e o final da transmissão.

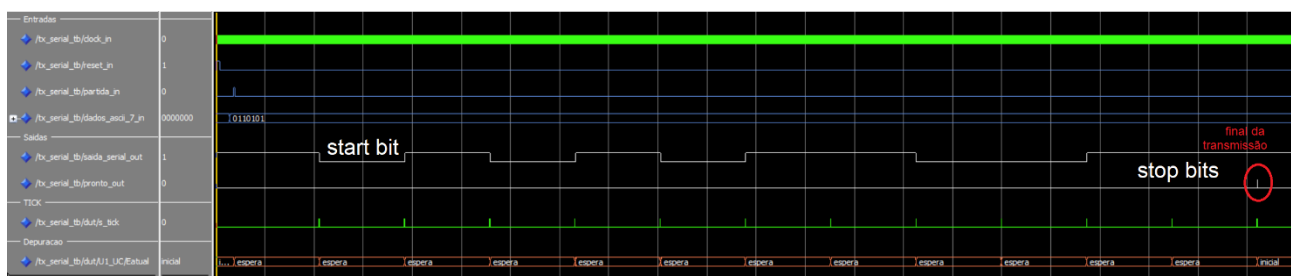


Figura 2 – Exemplo de forma de onda da simulação de um caso de teste.

Use as ferramentas disponíveis do software para verificar e analisar as formas de onda geradas (p.ex. largura de *bit*, conjunto de *bits* transmitidos, geração do sinal de *tick*, etc). A figura 3 mostra a medida da largura de 1 bit usando os cursores. Observe que, para uma taxa de 9600 *bauds*, a largura de 1 *bit* corresponde a $(1/9600)$ ou aproximadamente 104,17µs.

- c) Redigir um parágrafo descrevendo textualmente o funcionamento do circuito desde o acionamento do `reset`, espera do acionamento do sinal de `partida`, configuração dos dados de entrada, acionamento de `partida`, até o final da transmissão e a ativação do sinal `pronto`.
- d) Os resultados desta atividade devem ser incluídos no Planejamento.

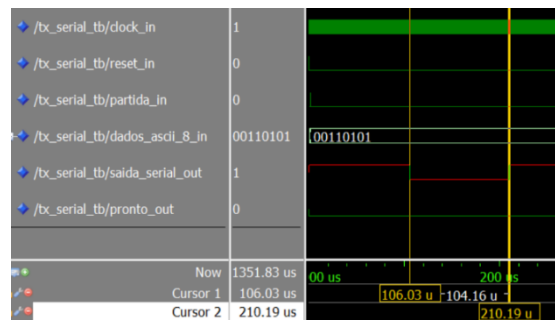


Figura 3 – Exemplo de medida de largura de *bit* usando cursores com ModelSim.

1.2. Atividade 2 – Modificação do Circuito de Transmissão Serial

Esta atividade envolve a modificação do projeto base para transmissão serial assíncrona estudado na Atividade 1 para usar a configuração de transmissão **701** (7 *bits* de dados, paridade ímpar, 1 *stop bit*) e uma taxa de comunicação de **115 200 bauds**. Nesta modificação deve-se criar um novo projeto no Intel Quartus Prime denominado `tx_serial_701`. Adote o seguinte módulo Verilog (figura 4). Repare que a entrada `dados_ascii` possui 7 *bits*.

```
module tx_serial_701 (
    input      clock      ,
    input      reset      ,
    input      partida    , // entradas
    input [6:0] dados_ascii ,
    output     saida_serial , // saidas
    output     pronto     ,
    output     db_clock    , // saidas de depuracao
    output     db_tick     ,
    output     db_partida  ,
    output     db_saida_serial ,
    output [6:0] db_estado
);
```

Figura 4 – Definição da entidade da Atividade 2.

e) Explique as **modificações necessárias** no circuito base para a implementação do circuito da atividade.

- Quais módulos do circuito necessitaram ser modificados e como?
- Quais foram as alterações para ajustar a configuração de comunicação de 7N2 para 701?
- Como a taxa de comunicação serial foi ajustada de 9600 *bauds* para 115 200 *bauds*?

Documente o projeto do grupo no Planejamento. **DICA:** anexe as saídas das ferramentas *RTL Viewer* e *State Machine Viewer* editadas para completar as informações de funcionamento.

- f) Use os casos de testes e sinais de depuração adotados no item b) para verificar e assegurar o correto funcionamento do circuito. Se julgar necessário, defina outros casos de teste e sinais de depuração.
- g) Simular o funcionamento do projeto com o ModelSim e anexar as formas de onda obtidas no Planejamento. Execute também medidas dos parâmetros das formas de onda e confirme sua corretude.
- h) O circuito de transmissão serial modificado deve usar preparado para síntese no Intel Quartus Prime com a seguinte designação (mínima) de pinos. Sinais podem ser adicionados pelo grupo.

sinal	pino	pino FPGA	Analog Discovery	
CLOCK	CLK_50	M9	-	-
RESET	SW0	U13	-	-
PARTIDA	SW1	V13	-	-
DADOS_ASCII[0]	SW2	T13	-	-
DADOS_ASCII[1..6]	SW3,SW4,SW5, SW6,SW7,SW8	T12,AA15,AB15, AA14,AA13,AB13	-	-
PRONTO	led LEDR9	L1	-	-
SAIDA_SERIAL	GPIO_0_D1	B16	Scope	CH1+

i) Submeter o arquivo QAR do projeto (`exp2_txserial_txbyy.qar`) junto com o Planejamento do grupo.

1.3. Atividade 3 – Implementação do Projeto na Placa FPGA DE0-CV

Neste item vamos implementar, no Laboratório Digital, o projeto do circuito de transmissão serial modificado na placa FPGA DE0-CV. Os sinais de entrada deste circuito serão acionados via chaves da placa DE0-CV e a saída serial será verificada com as ferramentas disponibilizadas pelo Analog Discovery.

- j) O sinal de saída do circuito `DB_SAIDA_SERIAL` deve inicialmente ser ligado no canal CH1+ da ferramenta *Scope* (osciloscópio) do Analog Discovery (designação conforme o item i).

DICA: não esquecer de ligar o sinal GND do Analog Discovery no pino GND da GPIO_0 da placa FPGA.

- k) Programar o circuito sintetizado na placa DE0-CV e execute testes de funcionamento do circuito, aplicando os casos de teste definidos no Planejamento.
- l) Use a **ferramenta *Scope*** do Analog Discovery para a visualização do sinal de saída serial para fins de depuração. Realize também medidas de parâmetros das formas de onda obtidas (largura de *bit*, tempo total de transmissão, etc). Adicione figuras com as saídas das ferramentas no Relatório. A figura 5 ilustra a medida da largura de 1 *bit*.



Figura 5 – Exemplo de uso dos recursos da ferramenta *Scope*.

- m) Se as formas de onda forem validadas, faça a ligação do sinal de saída serial para o **canal digital DIO7** do Analog Discovery. Mantenha o sinal GND conectado.

- n) A comunicação serial deve passar a ser analisada usando a **ferramenta *Protocol*** do Analog Discovery. Esta ferramenta implementa um analisador de protocolos e será usada para verificar a comunicação via protocolo RS-232C. A figura 6 ilustra o uso desta ferramenta.

DICA: mais informações sobre o uso desta ferramenta serão divulgadas na página da disciplina.

- o) Execute testes de funcionamento do circuito, aplicando mais uma vez os casos de teste definidos no Planejamento.

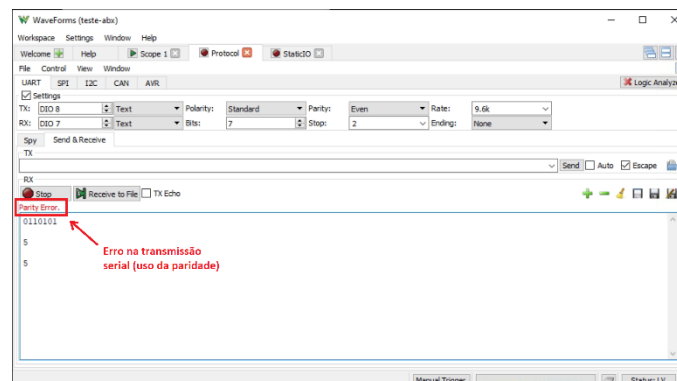


Figura 6 – Exemplo de uso dos recursos da ferramenta *Protocol*.

- p) Verifique o funcionamento do circuito, observando as saídas apresentadas na ferramenta *Protocol*.
- q) Colete imagens da tela da ferramenta e documente o funcionamento do projeto no Relatório.
- r) Submeter o arquivo QAR do projeto (`exp2_txserial701_txbyy.qar`) junto com o Relatório.

1.4. Atividade 4 –Desafio

Neste item deverão ser estudados e implementados alguns melhoramentos no circuito de transmissão serial na placa DE0-CV.

- s) Uma **modificação** no projeto do circuito de transmissão serial assíncrona será proposta pelo professor. Estude esta modificação e verifique qual parte do projeto deve ser alterado.
- t) Implemente o circuito no Intel Quartus Prime e sintetize-o na placa FPGA DE0-CV. Documente a designação de pinos adotada e alterações no projeto.
- u) Faça modificações também na montagem física do circuito e execute os testes de funcionamento. Documente os resultados obtidos nos testes experimentais realizados.
- v) Submeter o arquivo QAR do projeto do desafio (`exp2_desafio_txbyy.qar`) junto com o Relatório.

2. BIBLIOGRAFIA

- ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. **Tutorial para criação de circuitos digitais em Verilog no Quartus Prime 20.1**. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2024.
- ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. **Tutorial para criação de circuitos digitais hierárquicos em VHDL no Quartus Prime 16.1**. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2017.
- ALTERA / Intel. **DE0-CV User Manual**. 2015.
- ALTERA / Intel. **Quartus Prime Introduction Using Verilog Designs**. 2016.
- ALTERA / Intel. **Quartus Prime Introduction to Simulation of Verilog Designs**. 2016.
- CCITT - Fifth Plenary Assembly. Green Book. Vol. VIII, Geneve, December 1972.
- Electronic Industries Association. **Interface Between Data Terminal Equipment and Data Communication Equipment Employing Serial Date Interchange EIA-RS-232-C**, Washington, August 1969.
- HELD, G. **Understanding Data Communications**. 6th ed., New Riders, 1999.
- MIDORIKAWA, E.T. **Metodologia de Projeto com Dispositivos Programáveis**. Apostila de Laboratório Digital. PCS-EPUSP, 2016.
- PCS-EPUSP. **Conceitos de Comunicação Serial Assíncrona**. Apostila de Laboratório Digital. 2022.
- Ricardo Menotti, Ricardo dos Santos Ferreira. **Introdução à Lógica Digital com Verilog: uma abordagem prática**. Kindle. 2023
- WAKERLY, John F. **Digital Design Principles & Practices**. 5th edition, Prentice Hall, 2018.

3. MATERIAL DISPONÍVEL

- 1 placa com circuito integrado MAX3232 (conversor de nível de tensão)

4. EQUIPAMENTOS NECESSÁRIOS

- 1 computador com interface serial (ou 1 cabo USB serial) e software de comunicação.
- 1 computador com softwares Intel Quartus Prime e ModelSim.
- 1 dispositivo Analog Discovery da Digilent ou equivalente.
- 1 placa de desenvolvimento FPGA DE0-CV com o dispositivo Cyclone V 5CEBA4F23C7N.

Histórico de Revisões

E.S.G. e F.N.A./2001 – revisão
E.T.M./2004 – revisão
E.T.M./2005 – revisão
E.T.M./2008 – revisão
E.T.M./2011 – revisão
E.T.M./2012 – revisão
E.T.M./2013 – revisão da parte experimental
E.T.M./2014 – revisão
E.T.M./2015 – revisão do texto
E.T.M./2016 – revisão
E.T.M./2017 – revisão
E.T.M./2018 – revisão
E.T.M./2019 – revisão
E.T.M./2020 – revisão e reorganização da experiência para acesso remoto.
E.T.M./2021 – revisão
E.T.M./2022 – revisão e atualização para ensino presencial.
E.T.M./2023 – revisão.
E.T.M./2024 – revisão e adaptação para Verilog.