

UNIVERSIDADE DE SÃO PAULO
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E
SISTEMAS DIGITAIS
PCS3645 - LABORATÓRIO DIGITAL II



PLANEJAMENTO DA EXPERIÊNCIA 2

Felipe Luis Korbes - NUSP: 13682893

João Felipe de Souza Melo - NUSP: 13682913

João Felipe Pereira Carvalho - NUSP: 11808189

Turma: 5

Professor: Reginaldo Arakaki

São Paulo

2024

Sumário

1. Introdução e Objetivos.....	3
2. Planejamento.....	3
2.1 Atividade 1 - Estudo do Projeto Base do Circuito de Transmissão Serial.....	3
2.2 Atividade 2 – Modificação do Circuito de Transmissão Serial.....	5
2.3 Testes da modificação.....	7
2.3 Preparação da aula prática.....	8
2.4 Testes.....	8

1. Introdução e Objetivos

A experiência visa ensinar aos alunos sobre comunicação serial assíncrona (RS232C), conversores de nível de tensão, projeto de circuitos digitais com FPGA, uso da linguagem Verilog e técnicas de depuração. O objetivo é projetar e implementar um circuito digital que transmite dados em código ASCII para um terminal serial

2. Planejamento

2.1 Atividade 1 - Estudo do Projeto Base do Circuito de Transmissão Serial

O arquivo de transmissão serial disponibilizado contém vários módulos. Um deles é um contador responsável por contar a quantidade de bits deslocados. Além disso, há um deslocador que realiza o deslocamento dos bits do dado, e um detector de bordas que identifica o início de um sinal de entrada, retornando, com isso, apenas um pulso de clock. Esses são os componentes essenciais para compor o fluxo de dados na transmissão serial.

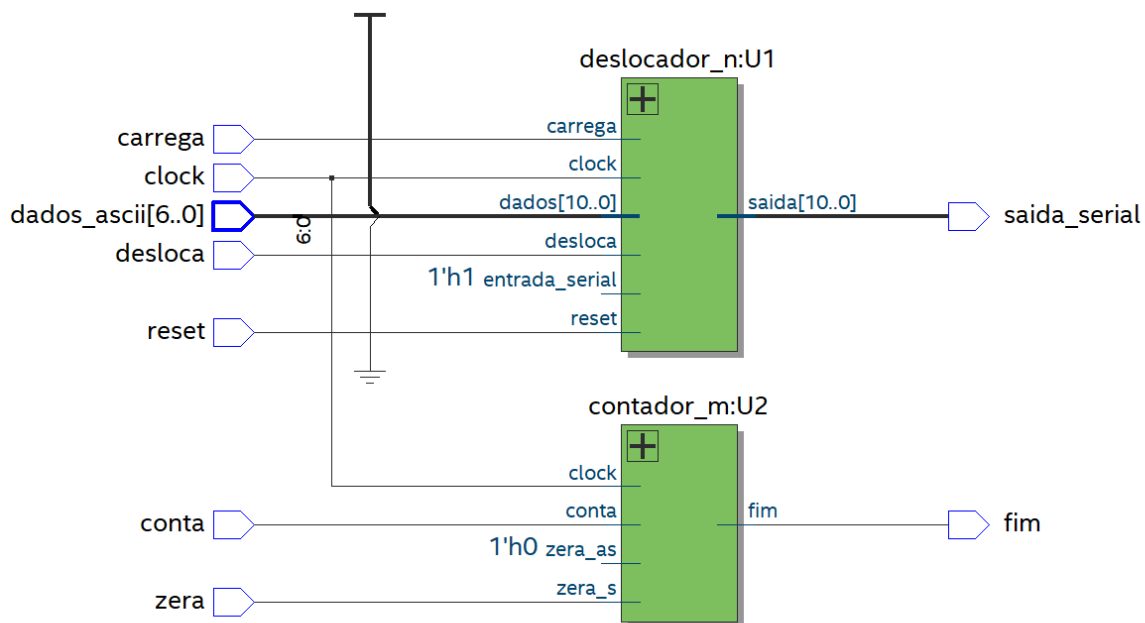


Figura 1 - Fluxo de dados do módulo de transmissão

Os bits de entradas são salvos no módulo *deslocador.v* no estado *preparacao*.

O módulo denominado *tx_serial_7N2* é onde o fluxo de dados, a unidade de controle e um contador de tick são instanciados. A unidade de controle tem os estados:

- *inicial*: o estado inicial
- *preparacao*: onde é carregado o dado a ser transmitido e resetado os contadores
- *espera*: é o estado de espera para o envio da saída serial.
- *transmissao*: é o estado onde ocorre o deslocamento dos bits do dado para a direita e ocorre a incrementação do contador de bits deslocados.
- *final_tx*: é o estado final onde o dado foi transmitido serialmente.

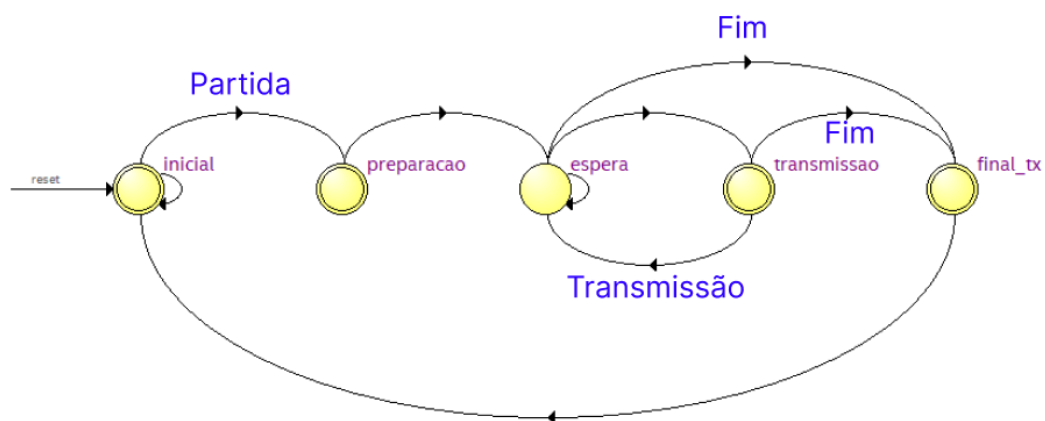


Figura 2 - Estados da unidade de controle

A unidade de controle começa no estado inicial e, após o acionamento da entrada "*partida*", muda-se para o estado de "*preparacao*". Nesse estado, os contadores são zerados e o dado é carregado no módulo *deslocador.v*. Após essa etapa, a unidade de controle passa automaticamente para o estado de "*espera*", onde nenhum sinal de controle é ativado, e a transição para outro estado só ocorre quando há um tick ou quando a transmissão é concluída. A indicação de que a transmissão terminou é dada pelo contador de bits deslocados: ao contar até 12, o sinal "*fim*" é ativado (HIGH), informando à unidade de controle que a transmissão foi finalizada, o que aciona a mudança para o estado "*final_tx*". Caso esteja no estado "*espera*" e a transmissão não tenha chegado ao final, após um tick pelo contador de ticks o próximo estado será "*espera*" novamente, aguardando o fim da transmissão serial.

The diagram shows a timing diagram for a serial communication system. The signals are as follows:

- clock_n**: A constant high signal (green bar).
- Entrada**: A constant low signal (0).
- reset**: A constant low signal (0).
- entrada_serial**: A constant high signal (1).
- dados_asci_7_in**: A signal that starts at 55, then drops to 0 after a red circle labeled "Início da transmissão". It remains at 0 until a blue circle labeled "Fim da transmissão".
- Sinais de saída**: A group of signals:
 - s_zera**: A constant low signal (0).
 - s_conta**: A constant low signal (0).
 - s_carrega**: A constant low signal (0).
 - s_desloca**: A constant low signal (0).
 - s_fin**: A constant high signal (1).
 - s_saida_serial**: A signal that starts at 1, then drops to 0 after a red circle labeled "Bit de início". It remains at 0 until a blue circle labeled "Fim da transmissão".
- s_saida**: A signal that starts at 111111111111, then drops to 0 after a red circle labeled "Bit de início". It remains at 0 until a blue circle labeled "Fim da transmissão".
- Sinal interno**: A signal that starts at 0, then drops to 1 after a red circle labeled "Bit de início". It remains at 1 until a blue circle labeled "Fim da transmissão".
- Estatul**: A signal that starts at 0, then drops to 1 after a red circle labeled "Bit de início". It remains at 1 until a blue circle labeled "Fim da transmissão".

Annotations and labels:

- Início da transmissão**: A red circle on the **dados_asci_7_in** signal.
- Fim da transmissão**: A blue circle on the **s_saida_serial** signal.
- Bit de início**: A red circle on the **s_saida_serial** signal.
- Bits de parada**: A blue circle on the **s_saida** signal.
- Estado de espera para transmissão**: A green circle on the **s_saida** signal.

Observa-se que o bit de início é 0 como esperado. Após isso é enviado serialmente o número 85_{10} (1010101_2). Após o envio desses 7 bits são enviados os dois bits de parada.



Para modificar o circuito de transmissão serial que tem configuração 7N1 para 7O1 basta modificar um bit de paridade e o valor do parâmetro no módulo contador de ticks para 434.

A configuração 7N1 tem dois bit de stop, portanto, basta mudar o primeiro deles (que é enviado) para o bit de paridade. A paridade ímpar indica que - no total de bits enviados - o número de bits sendo HIGH deve ser ímpar, portanto, o bit de paridade é dado pelo XNOR da entrada de dados.

Para modificar o módulo contador basta alterar o valor máximo que este contador conta para 434 e também modificar a quantidade de bits desse contador para 9 bits.

Abaixo está o fluxo de dados da alteração

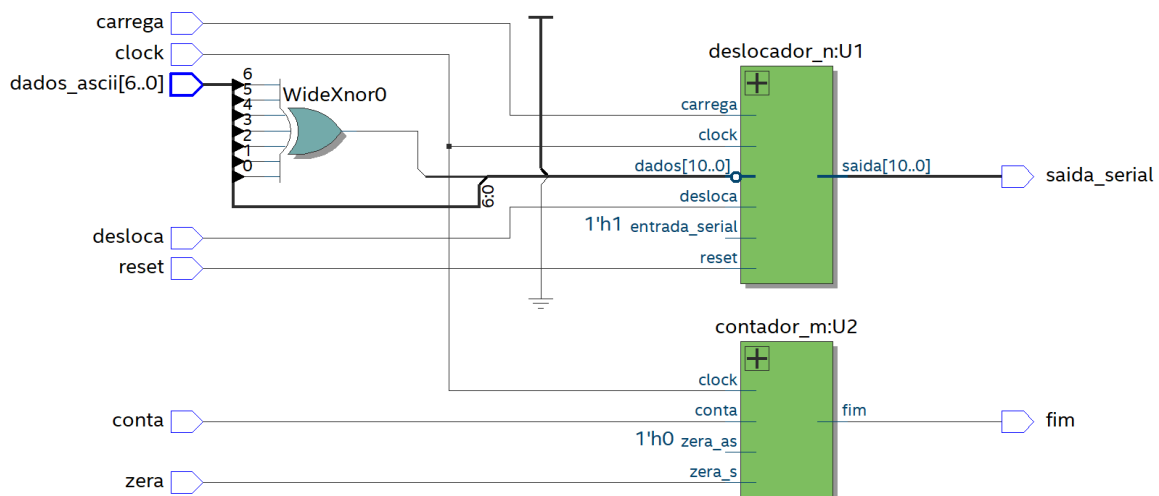


Figura 5 - Fluxo de dados após a alteração

E tem como estados da unidade de controle:

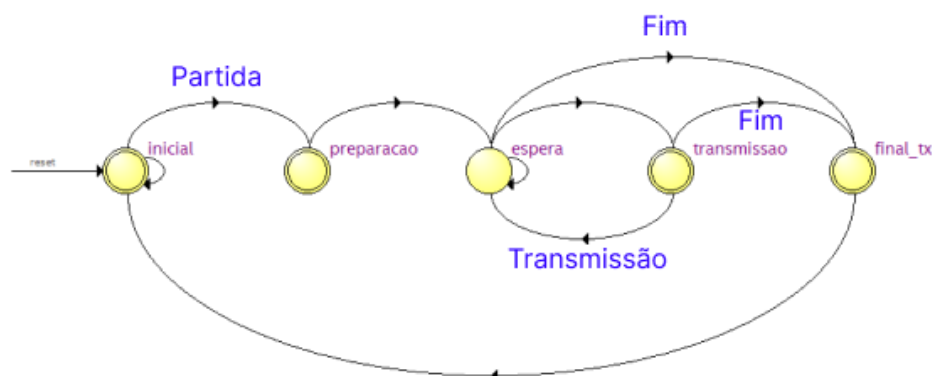
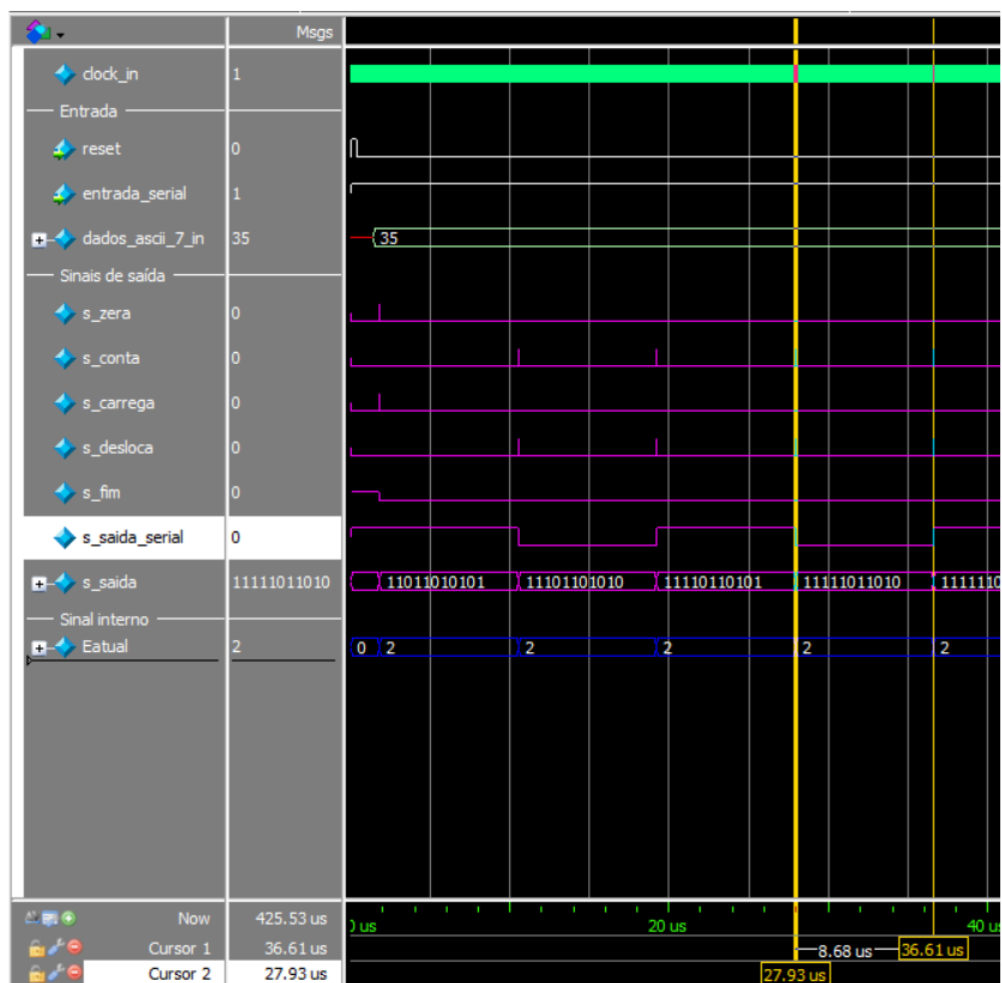


Figura 6 - Estados da unidade de controle após a alteração

2.3 Testes da modificação

Abaixo estão as imagens dos testes realizados. Vale ressaltar que, a largura do bit dessa vez foi de apenas 8 μ s, como visto na imagem abaixo.



É possível perceber que após o início da transmissão primeiro é enviado o bit de start e depois o dado. Também é enviado o bit de paridade ímpar e o bit de stop.

2.3 Preparação da aula prática

A pinagem a ser montada na aula prática está na tabela 1.

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+

Tabela 1 - Pinagem realizada na aula

Feito a montagem será realizada alguns testes para verificar a consistência da mudança e do circuito.

2.4 Testes

Os testes a serem realizados são:

Número	Esperado {stop, paridade, dado[6:0], start}	Resultado
$35_{10} = 0100011_2 = \#_{ASCII}$	1001000110	<input type="checkbox"/>
$63_{10} = 0111111_2 = ?_{ASCII}$	1101111110	<input type="checkbox"/>

$120_{10} = 1111000_2 = x_{\text{ASCII}}$	1111110000	<input type="checkbox"/>
$56_{10} = 0111000_2 = 8_{\text{ASCII}}$	1001110000	<input type="checkbox"/>

Tabela 2 - testes a serem realizados na aula teórica

Estes testes englobam tanto a verificação do bit de paridade como também o bound configurado. Este último será verificado pelo Analog Discovery.