

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



RELATÓRIO 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
3. Relatório.....	9
4. Conclusão.....	12

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.

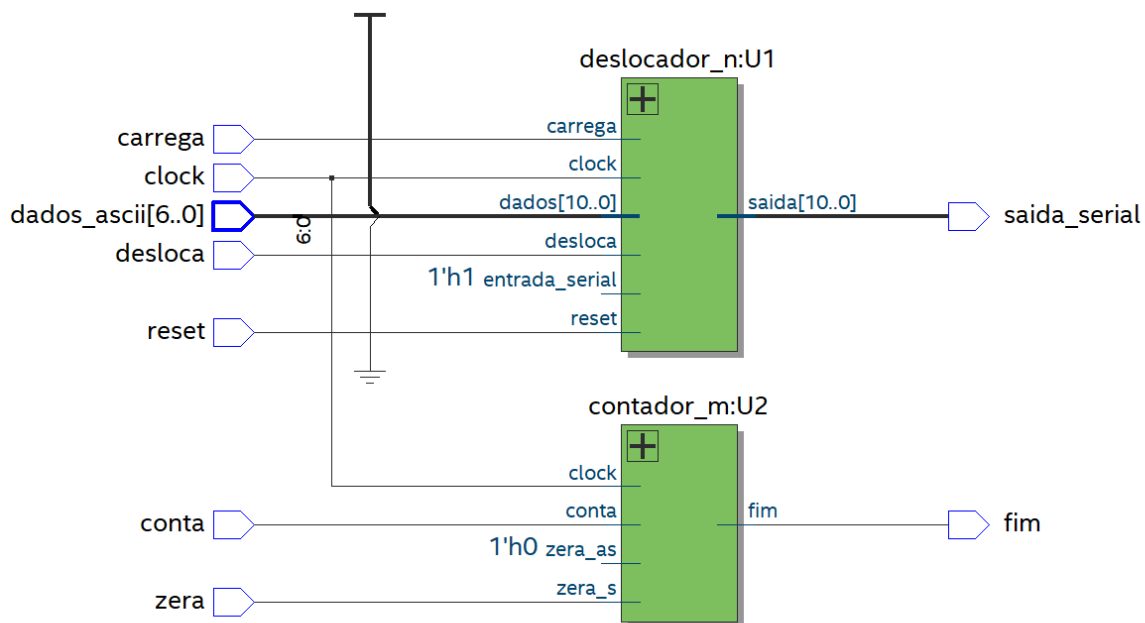


Figura 1 - Fluxo de dados do módulo de transmissão

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.

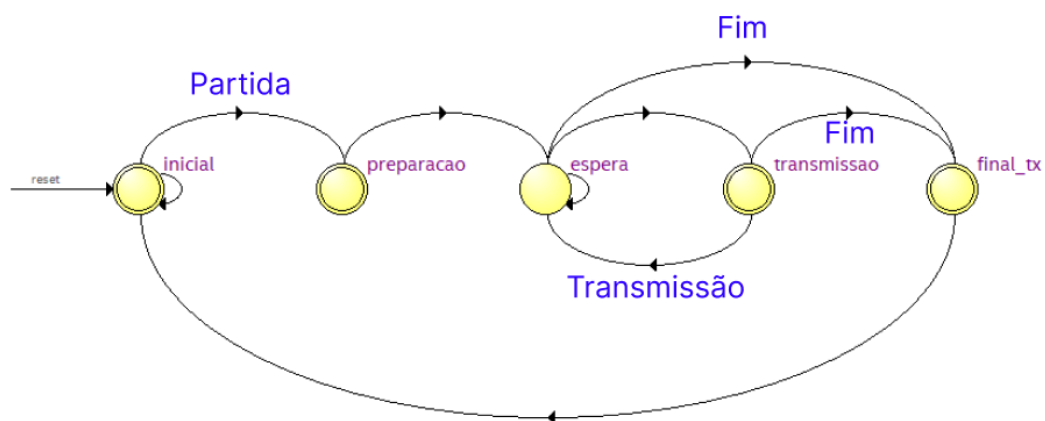


Figura 2 - Estados da unidade de controle

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.

Foi feita a análise das formas de onda do circuito com a test bench fornecida usando o ModelSim. Segue abaixo a imagem das formas de onda comentadas da test bench, assim como a largura do bit. Observa-se que a largura dos bits é de cerca de 104,14µs.

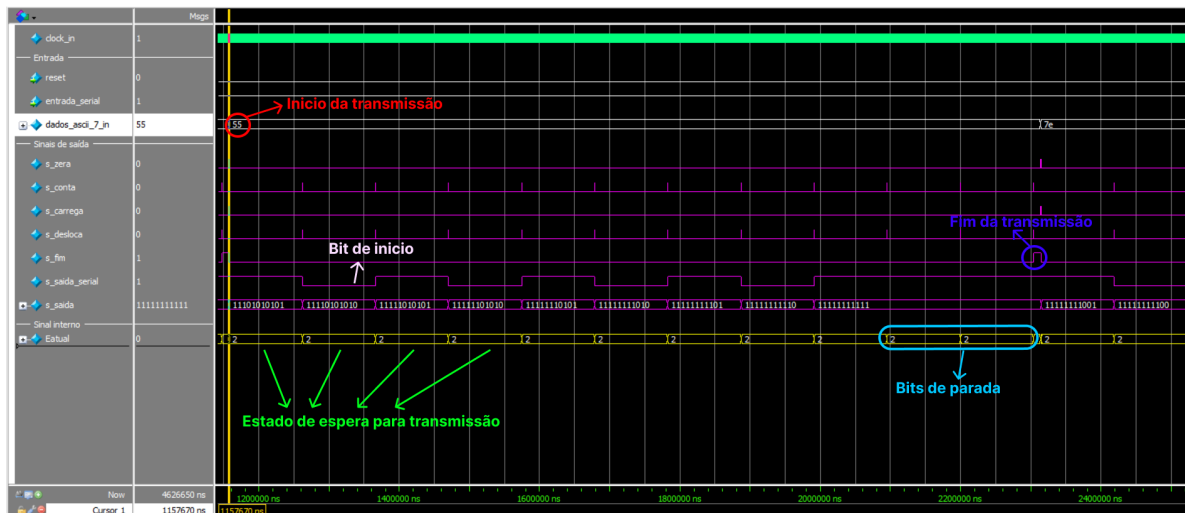


Figura 3 - Análise das formas de onda

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.

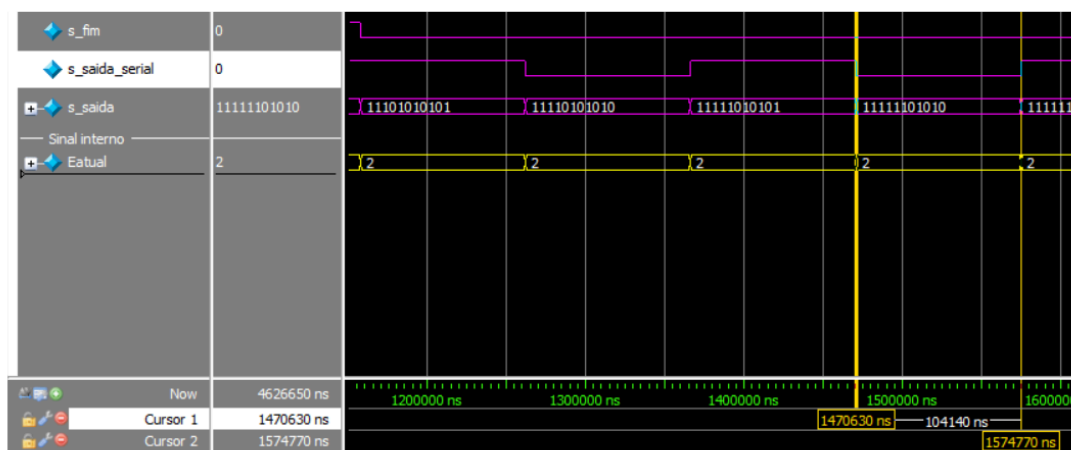


Figura 4 - Largura do bit

2.2 Atividade 2 – Modificação do Circuito de Transmissão Serial

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

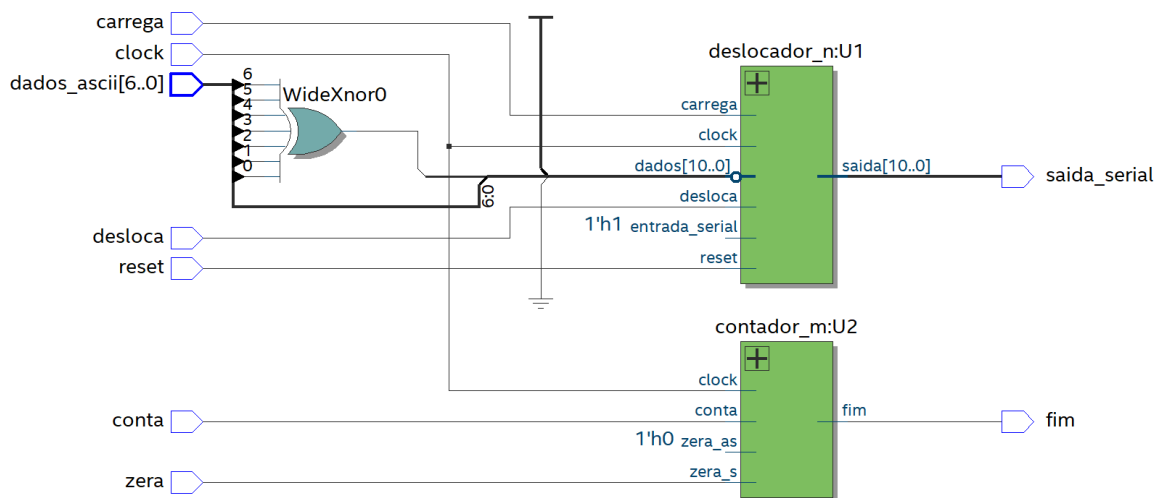


Figura 5 - Fluxo de dados após a alteração

E tem como estados da unidade de controle:

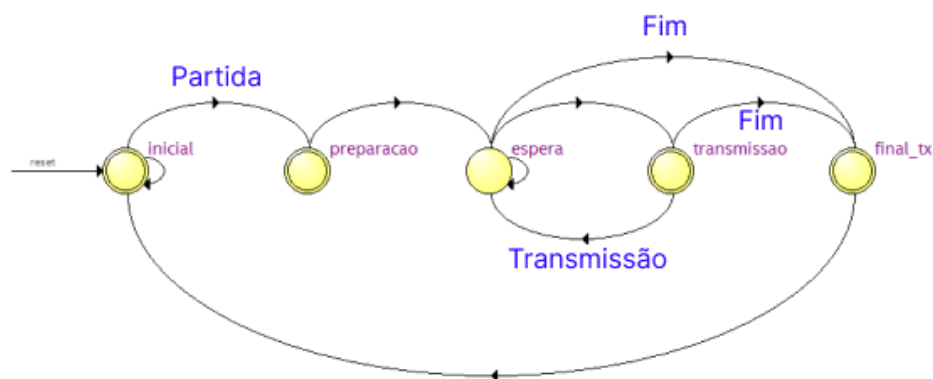
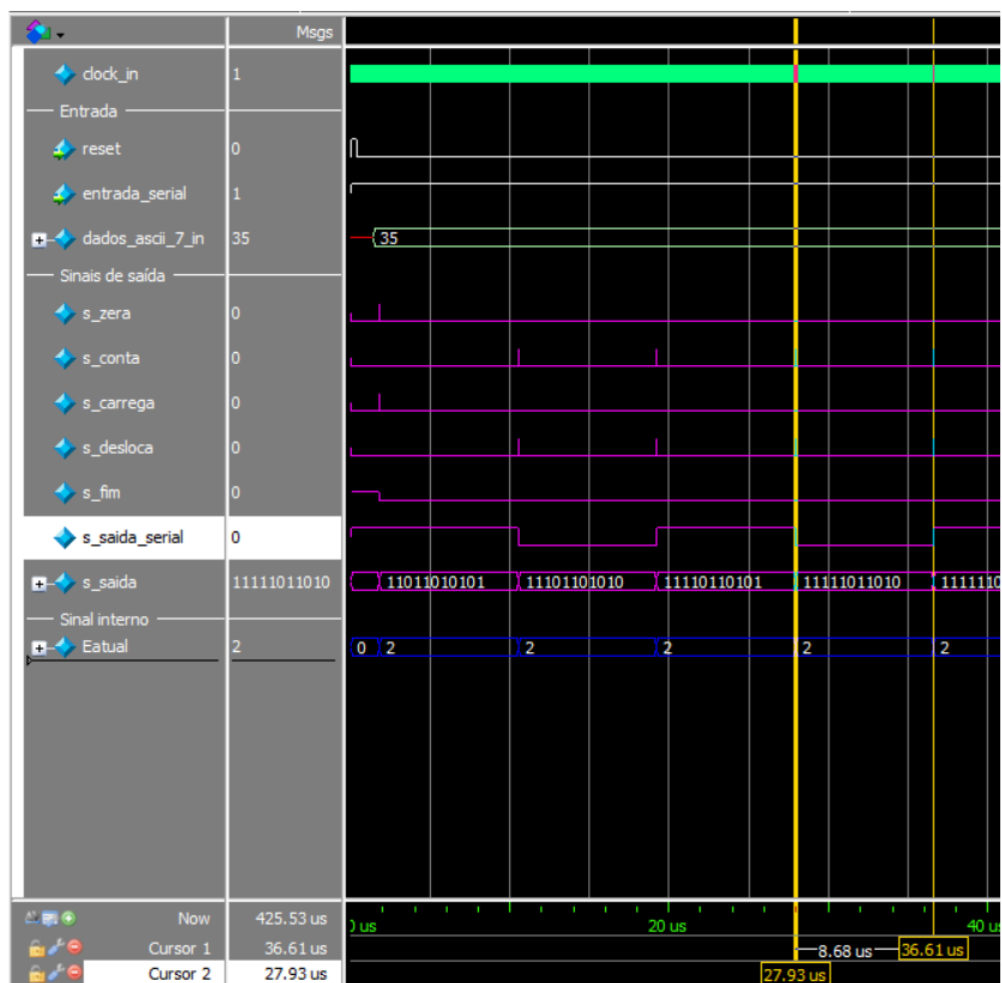


Figura 6 - Estados da unidade de controle após a alteração

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 checked="" type="checkbox"/>
$63_{10} = 0111111_2 = ?_{ASCII}$	1101111110	<input checked="" type="checkbox"/>

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

3. Relatório

Para iniciar o experimento, o grupo optou por, primeiramente, fazer os testes para verificar a conexão serial no Tera Term. Foi realizado um teste de loopback para verificar as partes analógica e digital do circuito. A imagem abaixo ilustra o loopback da parte analógica do circuito. Este procedimento permite identificar possíveis falhas no cabo ou problemas de mau contato.

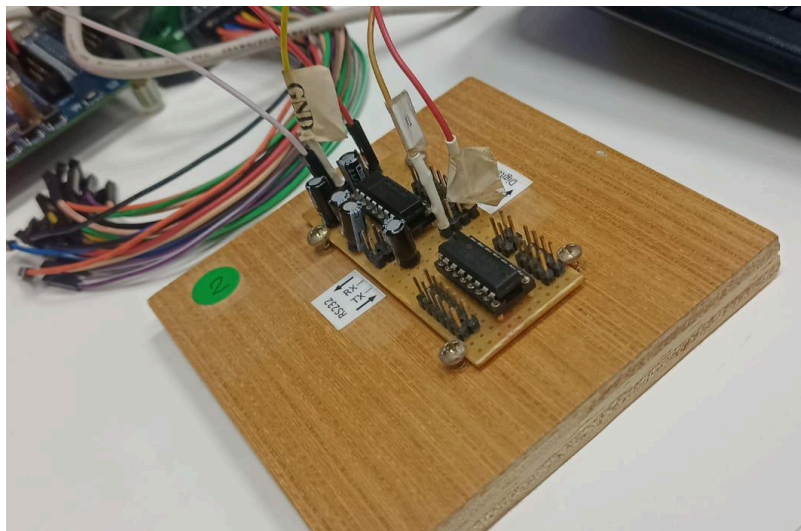


Figura 9 - LoopBack analógico do circuito

Abaixo está a figura que testa a parte digital do circuito, esse loopback para o circuito digital poderia ser feito e testaria, ao mesmo tempo, a parte digital e analógica do circuito. O grupo preferiu fazer os testes separados.

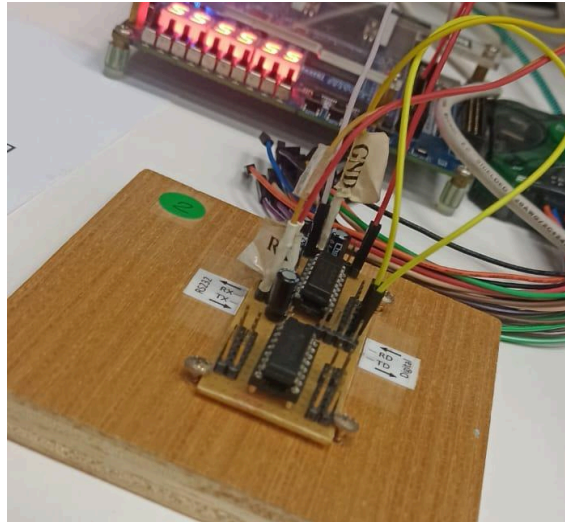


Figura 10 - LoopBack digital do circuito

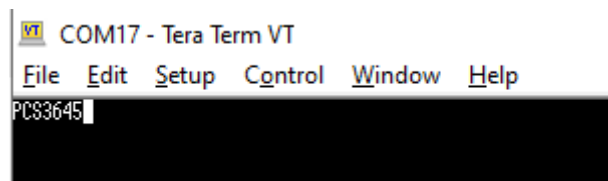


Figura 11 - LoopBack digital do circuito

Foi feito os testes propostos no planejamento. A Análise das ondas do sinal *db_saida_serial* estão nas figuras abaixo. Foi conectado o pino *GPO_0_1* da placa FPGA com o canal 1+ do Analog Discovery.

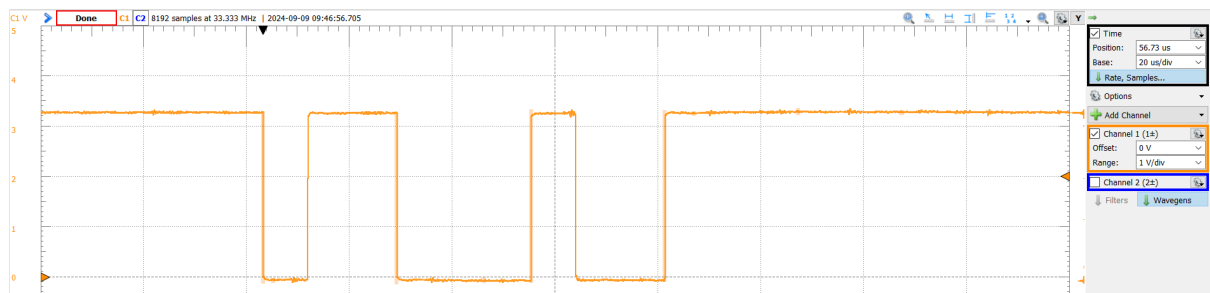


Figura 12 - Forma de ondas do caractere “#”

O dado a ser enviado é 0100011_2 , observe que inicialmente ocorre o bit de start e depois disso é enviado o dado do bit menos significativo para o mais significativo. Após esse envio é possível perceber o bit de stop.

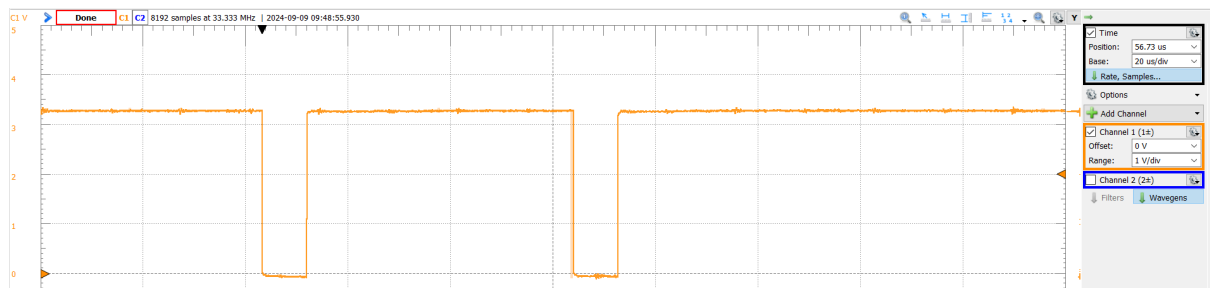


Figura 13 - Forma de ondas do caractere “?”

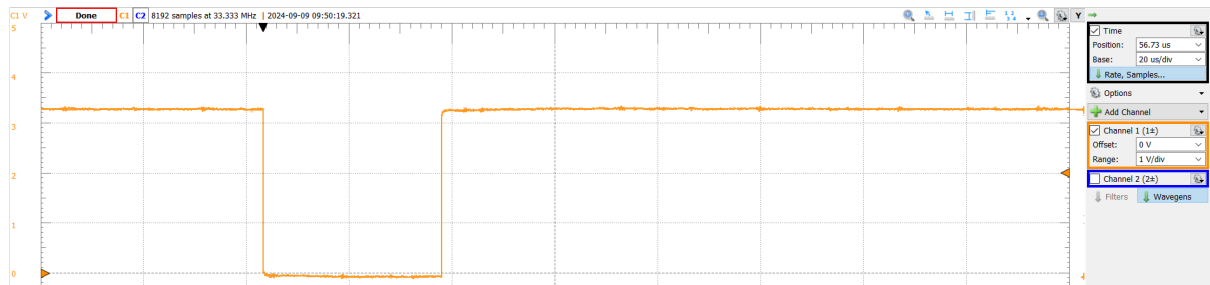


Figura 14 - Forma de ondas do caractere “x”

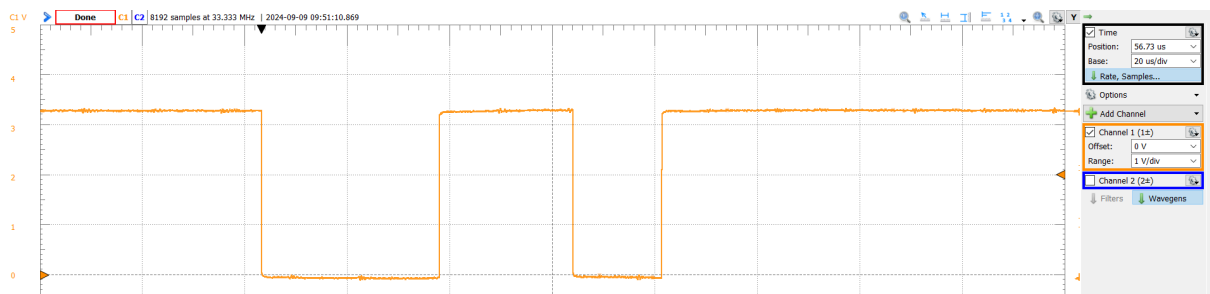


Figura 15 - Forma de ondas do caractere “8”

Abaixo está o terminal do Tera Term para os testes realizados.

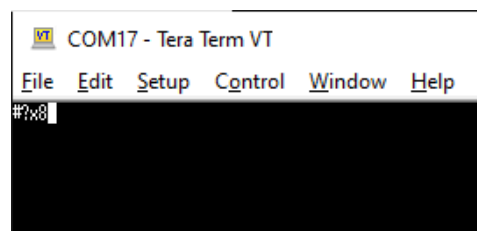


Figura 16 - Terminal do Tera Term

4. Conclusão

O experimento permitiu uma análise prática da comunicação serial assíncrona, confirmando o funcionamento dos circuitos analógicos e digitais através de testes de loopback. A avaliação das formas de onda no terminal Tera Term mostrou a correta transmissão dos dados em código ASCII, conforme o esperado. Os testes confirmaram a integridade do sinal e a correta implementação do circuito. A experiência atingiu seus objetivos ao ensinar sobre RS232C, conversores de nível de tensão, circuitos digitais com FPGA, e o uso de Verilog, consolidando o aprendizado teórico e prático dos alunos.