



TRANSMISSOR DE DADOS ASSÍNCRONO

Prova 1

PCS3335 - Laboratório Digital A

12/05/2022

Natanael Magalhães Cardoso, 8914122

Renato Naves Fleury, 11805269

Professor: Glauber de Bona

Turma: 10

Bancada: B3

Universidade de São Paulo

Escola Politécnica

Departamento de Eng. de
Computação e Sistemas Digitais



Transmissor de Dados Assíncrono

Natanael Magalhães Cardoso, Renato Naves Fleury

1. INTRODUÇÃO

A transmissão serial de dados de um sistema digital para outro é normalmente usada para reduzir o número de fios na linha de transmissão. Nesse tipo de comunicação é utilizada apenas dois fios, um de sinal GND e outro para a transmissão dos dados. Por exemplo, alguns periféricos do computador utilizam comunicação serial para se comunicarem com a placa mãe, tal qual o mouse e o teclado.

Nesse projeto será abordada apenas a comunicação serial assíncrona, na qual é transmitido um bit por vez, iniciando com um bit especial (bit Start) que tira a comunicação do repouso e indica a transmissão dos dados, e terminando com dois bits que indicam o fim da comunicação (bit Stop).

2. OBJETIVOS

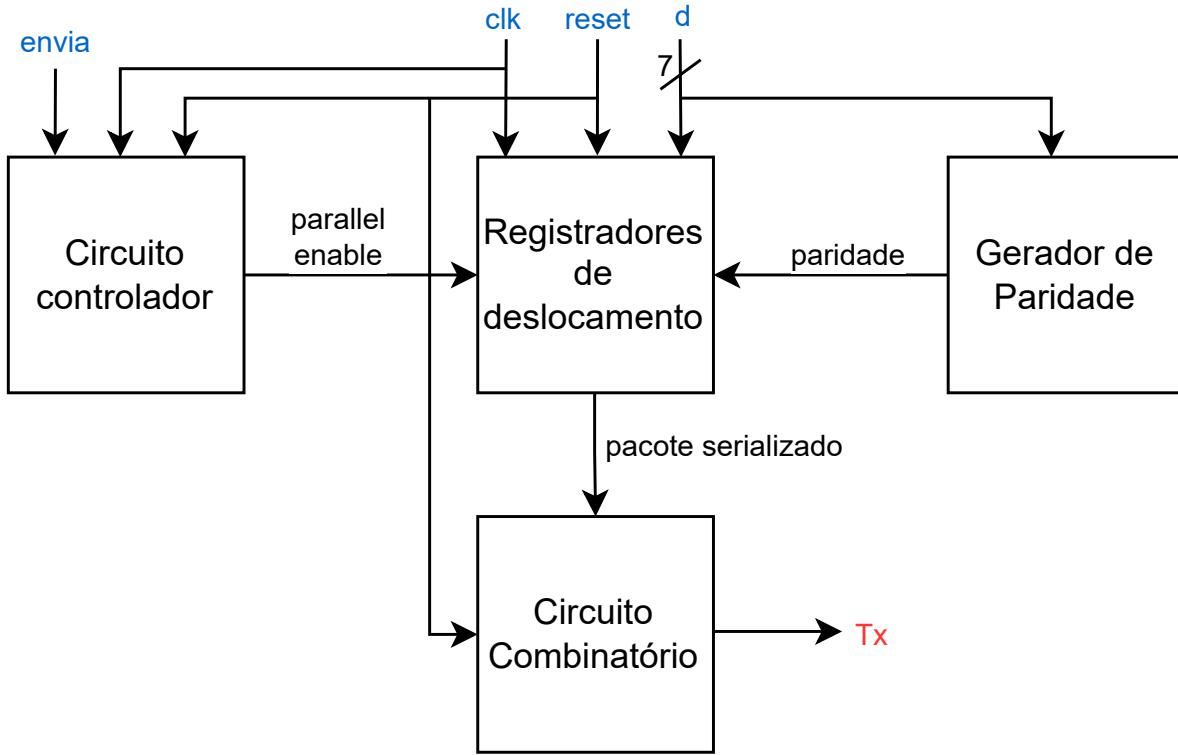
O objetivo deste experimento é projetar, simular, construir e testar um transmissor de dados assíncrono que transmita pacotes de dados de acordo com o protocolo RS232 e usando paridade ímpar.

3. PLANEJAMENTO

3.1. DIAGRAMA DE BLOCOS

A Fig. 1 mostra o diagrama de blocos do circuito transmissor de dados assíncrono. Este circuito recebe os dados em paralelo e realiza a serialização dos dados usando registradores de deslocamento. O circuito controlador implementa a lógica que liga/desliga o corte/ligamento/serialização dos registradores de deslocamento para que apenas um pacote seja enviado após o sinal envia. E, como os dados precisam ser transmitidos de acordo com o padrão RS232, há ainda a necessidade de calcular o bit de paridade e colocar o sinal Tx no nível lógico '1' (Mark) quando em repouso. Isso é feito pelo bloco gerador de paridade e circuito combinatório, respectivamente.

Importante: O sinal reset é equivalente ao sinal "limpa" do enunciado.



■ **Figura 1:** Diagrama de blocos do circuito com entradas em azul e saída em vermelho

3.2. SÍNTESE DO CIRCUITO

Na síntese do circuito transmissor de dados assíncrono, a primeira decisão tomada foi a escolha do CI que realizaria a conversão dos dados paralelos em dados seriais, para isso, utilizou-se o CI 74166 que já faz essa conversão internamente necessitando apenas de entradas de controle para configurá-lo. Como o circuito 74166 só possui 8 entradas paralelas optou-se por expandi-lo através da utilização de dois CIs desse tipo. A ligação para essa expansão consiste simplesmente em ligar a saída "QH" de um deles na entrada serial ("SER") do outro.

Para o controle desse "conversor paralelo serial" foi utilizado o contador módulo 11 projetado no experimento 4, porém com mais uma lógica, que no diagrama lógico tem o sinal "d_pe" como seu resultado. Sabe-se, pelo datasheet, que o 74166 só carrega as entradas paralelas, na próxima borda de subida do clock, quando a sua entrada de controle "STLD" está em 0. E, para esses dados serem enviados serialmente pela saída "QH" é preciso que a entrada "STLD" esteja em 1. Portanto, essa lógica adicional implementada no contador nada mais é do que um resultado zero somente quando o contador está em estado zero, ou seja, um OR de todas as saídas de estado do contador. Já o sinal "d_disable" é o resultado da lógica que transforma o contador 74163, que naturalmente é um contador módulo 16, em um contador módulo 11. Esse sinal foi obtido através da negação da equação algébrica resultante do mapa de Karnaugh da Fig.9 do apêndice.

Como a transmissão serial de dados assíncrona precisa seguir um protocolo de comunicação, precisamos colocar um bit que indica o início da transmissão (o bit "Start") e dois bits que indicam o

final (o bit "Stop"). Para isso, colocou-se a entrada "H" do primeiro 74166 em GND e as entradas "F" e "G" do segundo 74166 em VCC.

A fim de controlar os CIs 74166 e o contador para só iniciarem a transmissão dos dados quando o sinal "envia" for ativado, utilizou-se um flip-flop do CI 7474 com o próprio sinal envia na entrada de clock, e colocou-se a saída negada do flip-flop na entrada de controle "ENP" do contador, ou seja, o contador só começa a contagem e o 74166 só começa a transmissão dos dados quando o sinal envia é ativado.

Por fim, implementou-se a lógica que resulta no sinal de saída "Tx" pensando na padronização da comunicação para ficar em alta quando o transmissor está em repouso e, durante a transmissão, transmitir os dados sem interferência. Como pode ser visto no diagrama lógico da Fig 2, essa lógica consiste em uma porta lógica OR ligada à saída do 74166 com a saída de uma porta NAND que tem como entradas o sinal "d_pe" e o sinal "reset".

3.3. DIAGRAMA LÓGICO

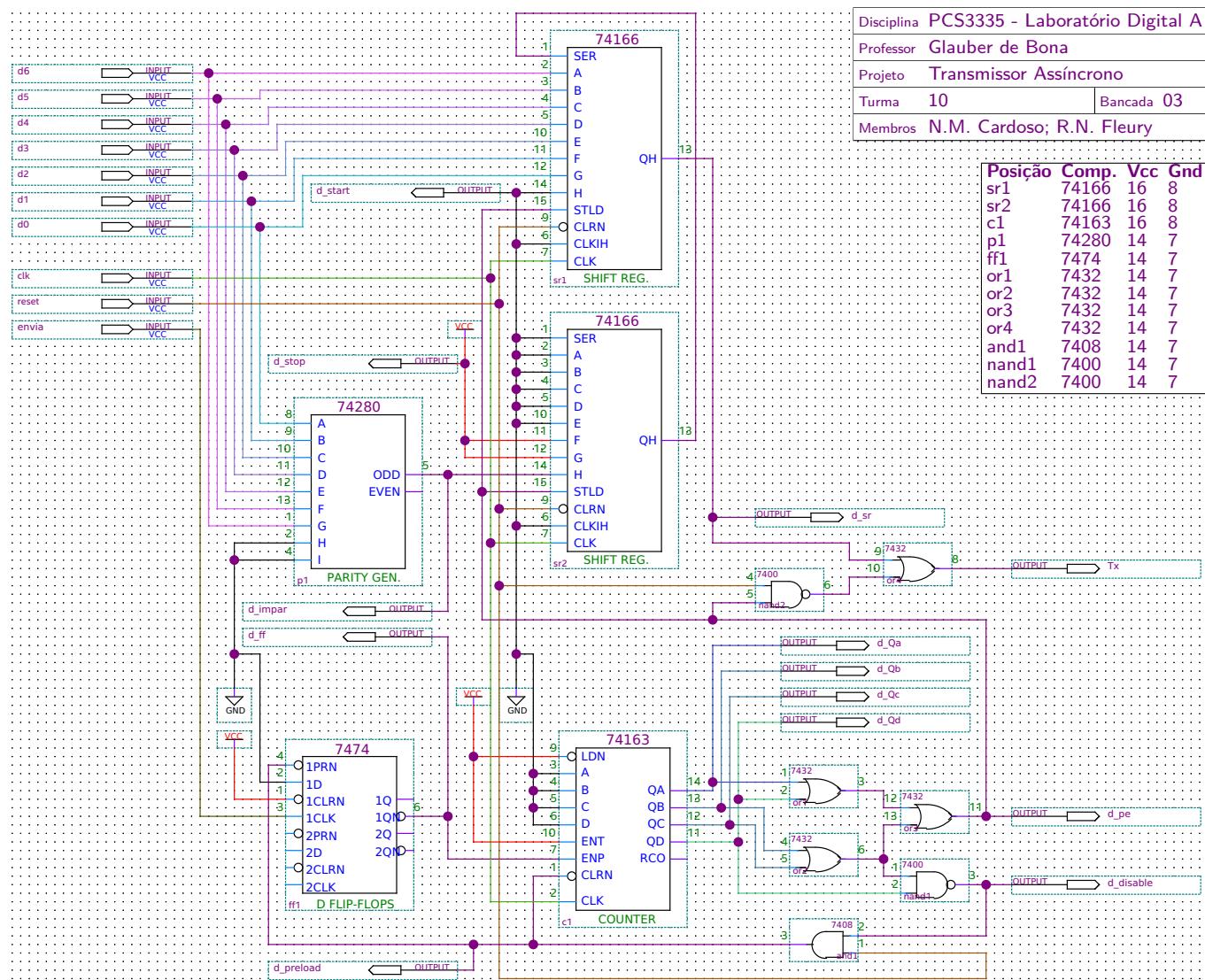


Figura 2: Diagrama lógico do transmissor assíncrono.

A Fig. 2 mostra o diagrama lógico criado com o método de síntese especificado na Seção 3.2. A disponibilidade dos componentes usados no diagrama foi conferida durante o OpenLab.

3.4. SIMULAÇÃO

Quatro casos de teste foram elaborados para validar a lógica do circuito projetado. Cada caso de teste possui uma descrição em alto nível de um comportamento esperado do sistema. Essas descrições são, então, convertidas em estímulos dos sinais de entrada para criação da carta dos tempos, uma para cada caso de teste. A inspeção dessas cartas dos tempos diz se a resposta do circuito está coerente com a esperada. Esses casos de teste estão detalhados nos Apêndices [A.1](#), [A.2](#), [A.3](#) e [A.4](#).

A análise de cada uma das cartas dos tempos (Figs. 5, 6, 7 e 8) mostrou que o circuito projetado passou em todos os casos de teste.

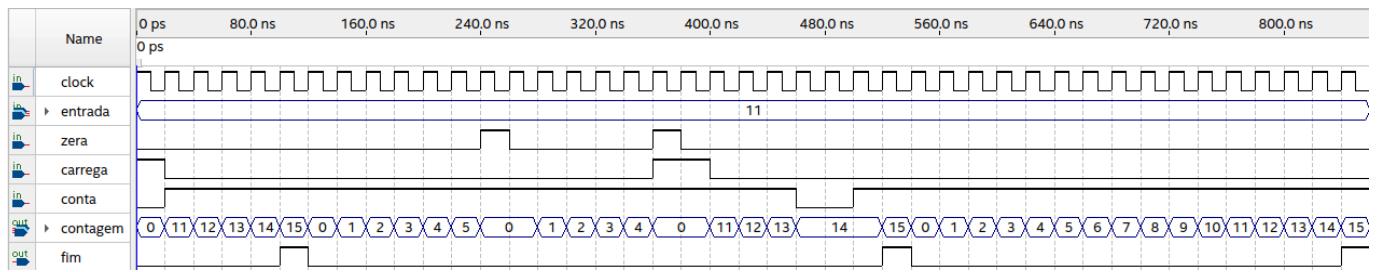


Figura 3: Carta dos tempos do circuito contador

3.5. LEVANTAMENTO DOS MATERIAIS NECESSÁRIOS

Tabela 1: Unidades requeridas para cada CI

Slot	Operação	CI	Un. Requeridas	Un. Disponíveis
1	Reg. Desl.	74166	1	1
2	Reg. Desl.	74166	1	1
3	Ger. Par.	74280	1	1
4	Flip-Flop	7474	1	2
5	Contador	7463	1	1
6	And-2	7408	1	4
7	Or-2	7432	4	4
8	Nand-2	7400	2	4

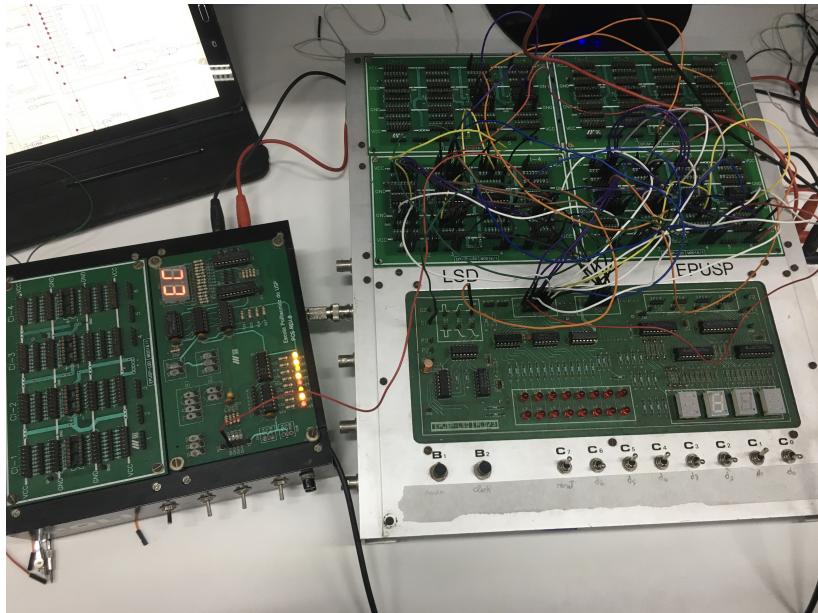
Para garantir que o circuito projetado respeite as restrições de montagem, fizemos um levantamento dos recursos necessários para este circuito mostrado na Tabela 1. Ela mostra a quantidade de unidades lógicas requeridas para cada CI utilizado. As especificações de cada CI foi obtido pelos respectivos *datasheets*.

3.6. METODOLOGIA DE MONTAGEM

A montagem será feita reduzindo o circuito em duas partes menores \mathcal{P}_1 e \mathcal{P}_2 . Separamos essas partes de forma a ser possível depurá-las separadamente usando as cartas dos tempos e a tabela de depuração nos Apêndices A e B, respectivamente. \mathcal{P}_1 é constituída pelos componentes c1, or1, or2, or3, nand1, and1 e ff1 do diagrama lógico (Fig. 2) e \mathcal{P}_2 é constituída pelos componentes p1, sr1, sr2, nand2 e or4. Para otimizar o tempo de montagem, \mathcal{P}_1 e \mathcal{P}_2 serão montadas em paralelo. As entradas de \mathcal{P}_1 são clk, reset e envia e os sinais de depuração são d_Qa, d_Qb, d_Qc, d_Qd, d_preload, d_ff, d_disable e d_pe, sendo esse último uma das entradas de \mathcal{P}_2 . Os sinais de entrada de \mathcal{P}_2 são os 7 bits de dados, clk e reset, com substituição do sinal d_pe por VCC para simular esse sinal durante a transmissão, e os sinais depuração são d_impar, d_sr e a saída do circuito Tx.

4. RESULTADOS

O circuito transmissor de dados foi montado conforme o diagrama lógico da Fig. 2 e testado conforme os casos de teste descritos nos Apêndices A e B. Durante a montagem (Fig. 4) foi notado um problema de Hazard no circuito causando um atraso no sinal $d_{preload}$, sendo esse sinal substituído por uma chave. Por isso, o pacote não pode ser transmitido pelo TeraTerm da forma esperada. O pacote era transmitido corretamente apenas ao ligar o circuito.



■ **Figura 4:** Montagem do circuito

5. CONCLUSÃO

Concluímos que, a partir do circuito lógico projetado, foi possível implementar um circuito na placa de montagem que cumpriu parcialmente com os requisitos. Houve a transmissão do caractere desejado, porém, para clocks de alta frequência, não era possível parar a transmissão manualmente.

APÊNDICE

A. CASOS DE TESTE

A.1 Primeiro Caso de Teste

Descrição do Caso de Teste: O sinal Tx deve estar inicialmente em estado de repouso (nível lógico '1'). Assim que o sistema receber o sinal envia, os dados paralelos devem ser carregados no próximo sinal de relógio. No clock de carregamento, o sinal Tx deve permanecer em estado de repouso e no sinal de relógio subsequente deve transmitir apenas um pacote de dados e voltar imediatamente para o estado de repouso. O pacote deve ser consituído por um bit start (nível lógico '0'), uma palavra com os 7 bits de dados, um bit de paridade ímpar e dois bits stop (nível lógico '1').

Esse é um caso de teste base e testa as características fundamentais do sistema.

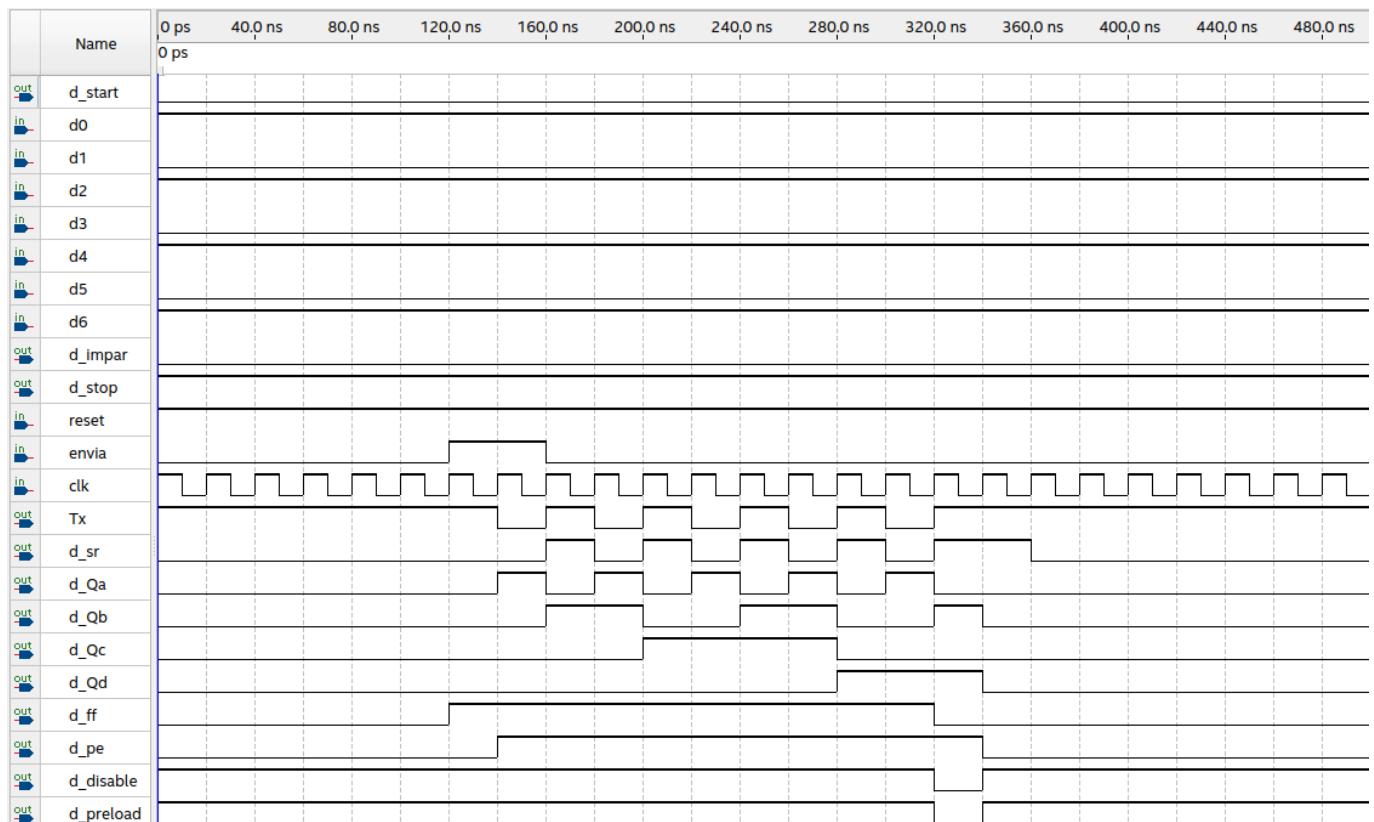


Figura 5: Carta dos Tempos do primeiro caso de teste

A.2 Segundo Caso de Teste

Descrição do Caso de Teste: Assim que o sistema receber o sinal reset no nível lógico '0' ele deve interromper imediatamente a transmissão e o sinal Tx deve voltar imediatamente para o estado de repouso. O interrompimento da transmissão não deve influenciar na transmissão seguinte. Assim, no próximo sinal envia, o sistema deve carregar novamente os dados e transmitir o pacote com os novos valores de dados.

Esse caso de teste tem finalidade de identificar possíveis efeitos colaterais (e.g. iniciar transmissão seguinte com contador diferente de zero) nos registradores de deslocamento e nos estados do contador causados pelo sinal reset.

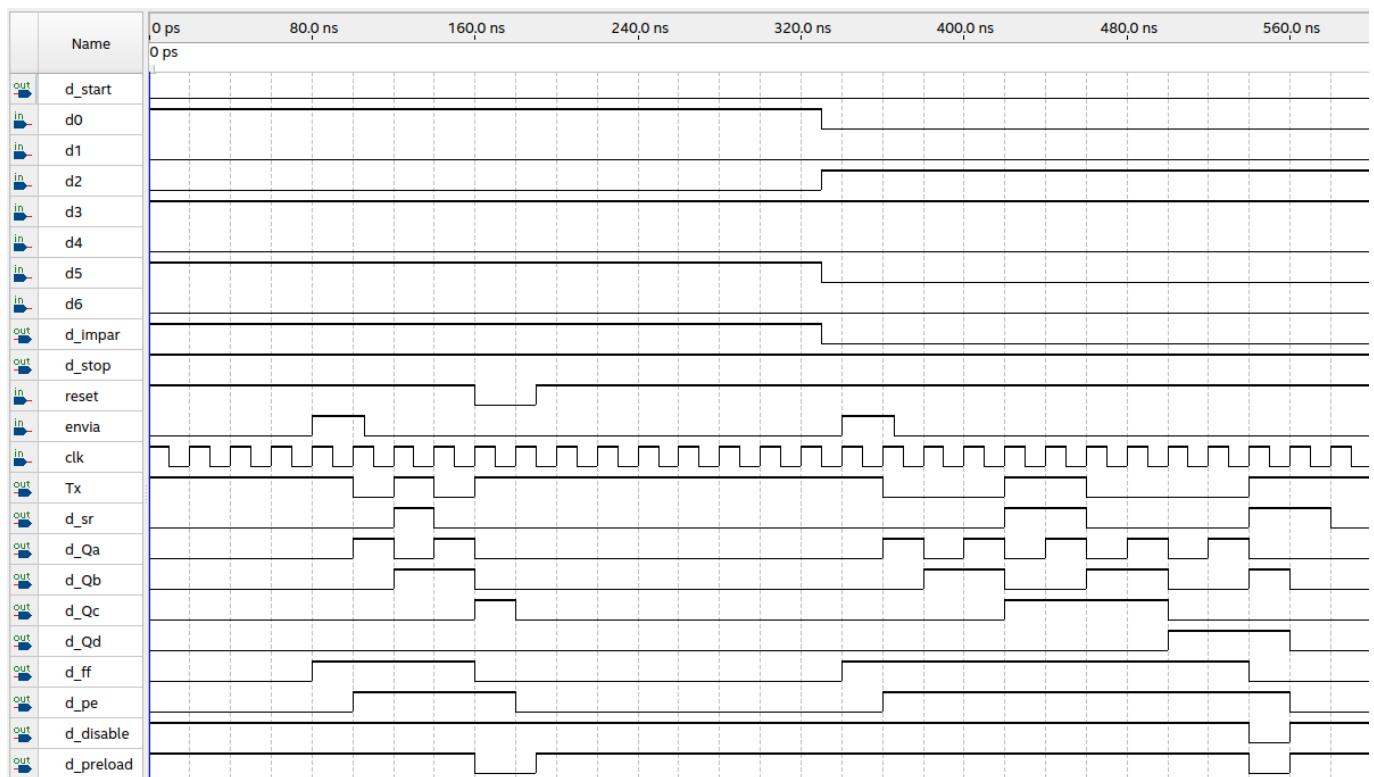


Figura 6: Carta dos Tempos do segundo caso de teste

A.3 Terceiro Caso de Teste

Descrição do Caso de Teste: O sistema deve ser sensível à borda de subida do sinal envia e deve enviar apenas um pacote de dados e voltar para o estado de repouso, mesmo que o sinal envia ainda esteja alto. O sistema só poderá enviar um novo pacote ao detectar uma nova borda de subida do sinal envia.

Esse caso de teste tem o objetivo de verificar se o sistema sairá do seu comportamento esperado caso o tempo de duração do sinal envia seja maior que o tempo de envio do pacote. Por exemplo, apertar o botão envia manualmente e receber um sinal de clock de um gerador de função com frequência da ordem de centenas de Hz.

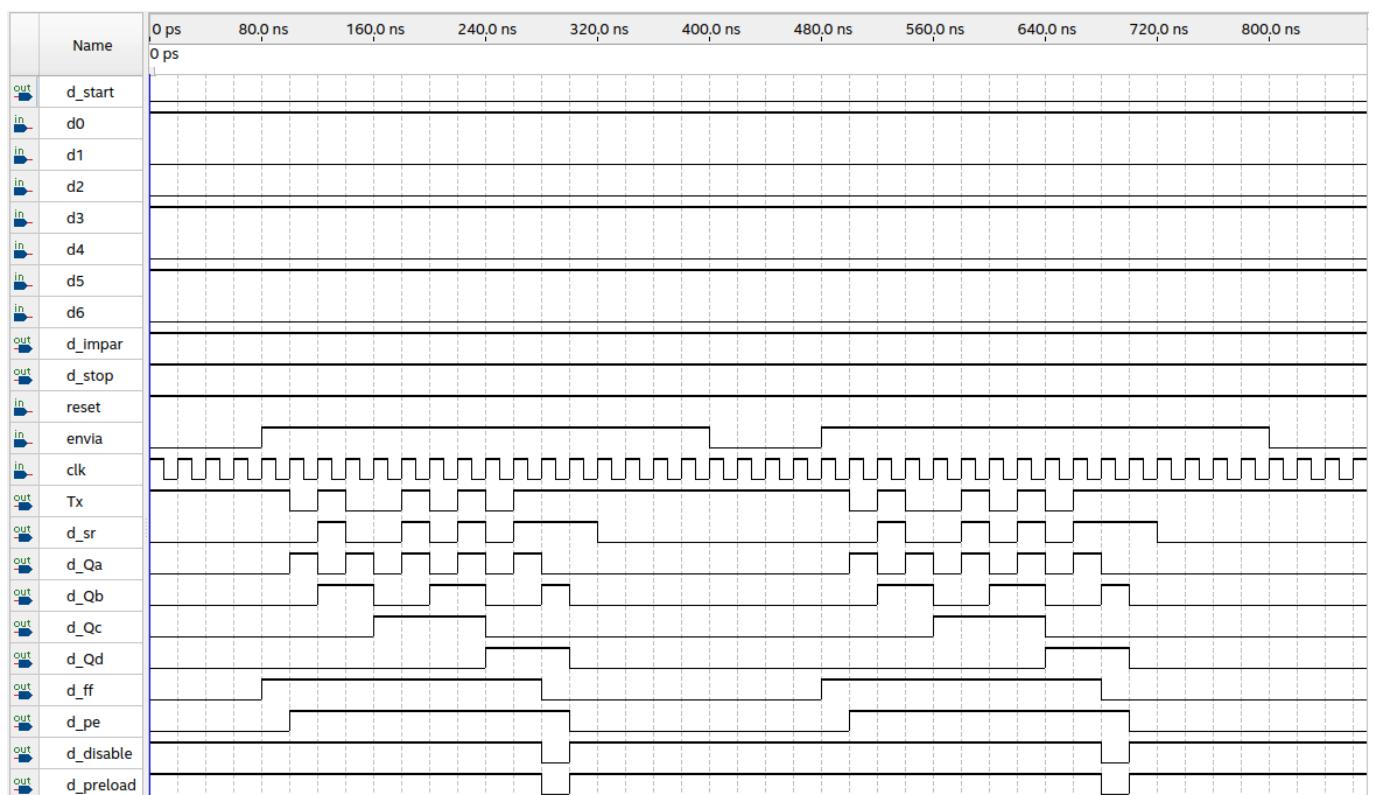


Figura 7: Carta dos Tempos do terceiro caso de teste

A.4 Quarto Caso de Teste

Descrição do Caso de Teste: O sistema deve apresentar paridade ímpar. Isto é, o bit de paridade deve ter o nível lógico '1' se o número de níveis lógicos iguais a '1' nos bits de dados forem ímpar e '0' caso contrário.

Esse é um caso de teste base e tem o objetivo de verificar se o bit de paridade é calculado corretamente.

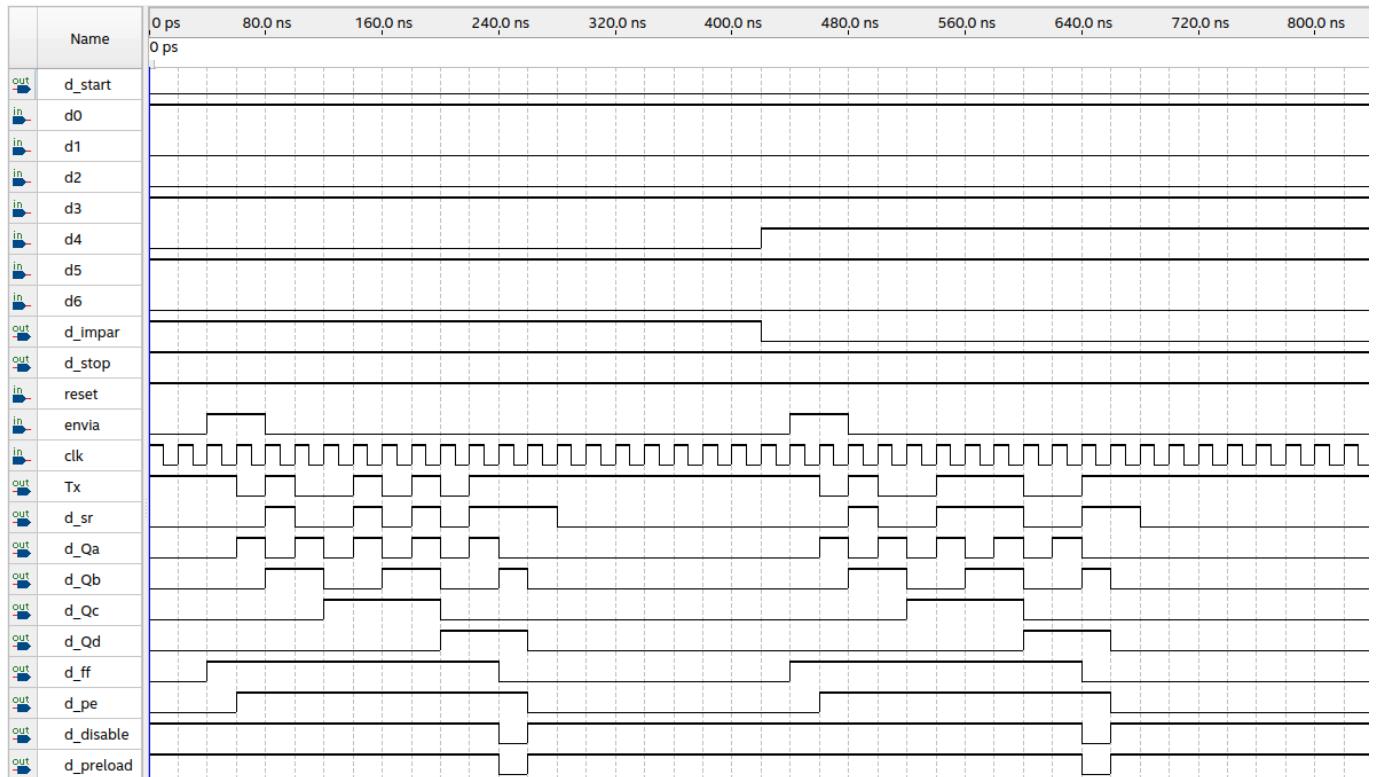


Figura 8: Carta dos Tempos do quarto caso de teste

B. TABELA DE DEPURAÇÃO

■ **Tabela 2:** Tabela de depuração do envio de uma palavra genérica $d_6d_5d_4d_3d_2d_1d_0$

Entradas			Sinais de Depuração									Saída	
clk	envia	reset	Qa	Qb	Qc	Qd	ff	pe	disable	preload	sr	Tx	*
↑	↑	1	0	0	0	0	1	0	1	1	0	1	1
↑	x	1	1	0	0	0	1	1	1	1	0	0	2
↑	x	1	0	1	0	0	1	1	1	1	d_0	d_0	3
↑	x	1	1	1	0	0	1	1	1	1	d_1	d_1	4
↑	x	1	0	0	1	0	1	1	1	1	d_2	d_2	5
↑	x	1	1	0	1	0	1	1	1	1	d_3	d_3	6
↑	x	1	0	1	1	0	1	1	1	1	d_4	d_4	7
↑	x	1	1	1	1	0	1	1	1	1	d_5	d_5	8
↑	x	1	0	0	0	1	1	1	1	1	d_6	d_6	9
↑	x	1	1	0	0	1	1	1	1	1	p	p	10
↑	x	1	0	1	0	1	0	1	0	0	1	1	11
↑	x	1	0	0	0	0	0	0	1	1	1	1	12
↑	x	1	0	0	0	0	0	0	1	1	1	1	13
x	x	0	x	x	x	x	0	x	1	0	0	1	14

¹ Detecção da borda de subida do sinal `envia`, carregamento dos dados paralelos (1 clock), sinal Tx continua em repouso; ff tem sinal '1' daqui até o clock 11 (11 clocks)

² Transmissão do `start bit`; pe tem sinal '1' daqui até o clock 12 (11 clocks)

³ Transmissão do bit d_0

⁴ Transmissão do bit d_1

⁵ Transmissão do bit d_2

⁶ Transmissão do bit d_3

⁷ Transmissão do bit d_4

⁸ Transmissão do bit d_5

⁹ Transmissão do bit d_6

¹⁰ Transmissão do bit de paridade

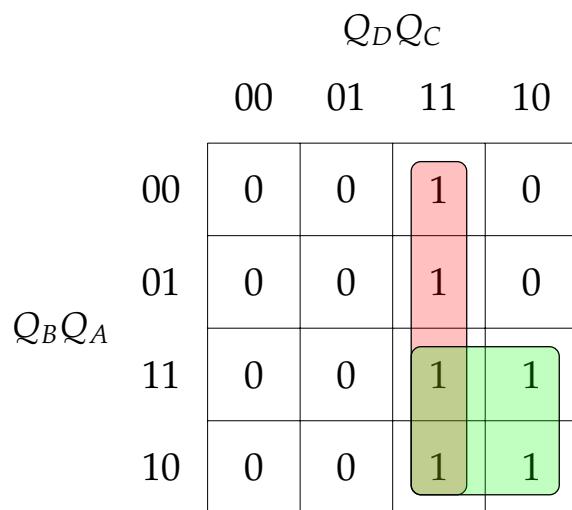
¹¹ Transmissão do primeiro `stop bit`; ff precisa ter sinal '0' a partir deste clock até a próxima borda de subida do sinal `envia`; disable e preload precisam ter sinal '0' apenas neste clock

¹² Transmissão do segundo `stop bit`; pe precisa ter sinal '0' a partir deste clock até o clock do próximo `start bit`

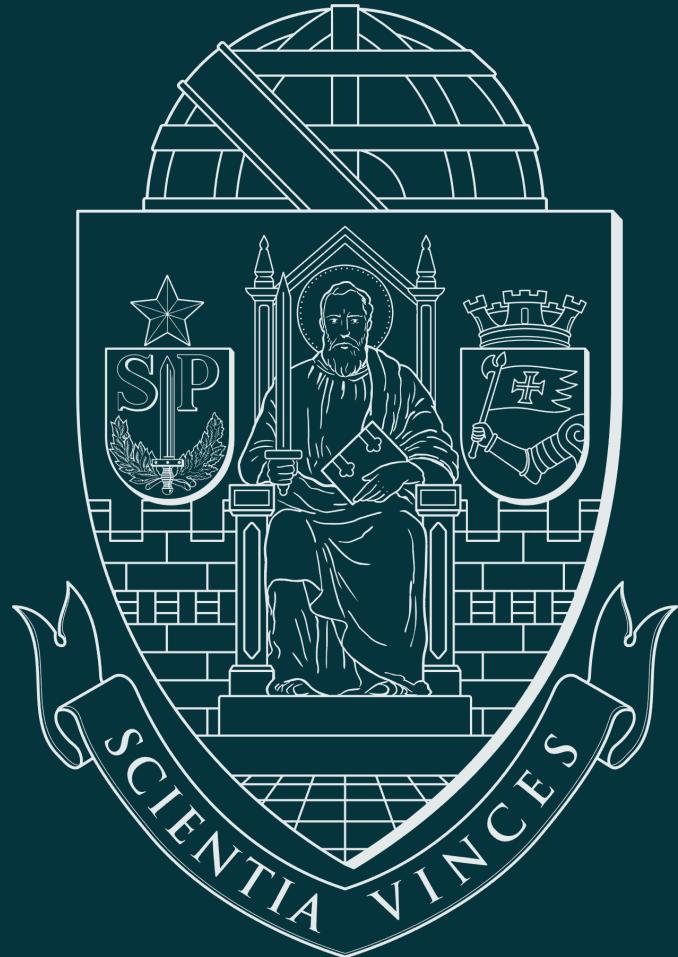
¹³ Estado de repouso

¹⁴ Reset

C. MAPA DE KARNAUGH DO CIRCUITO CONTROLADOR



■ **Figura 9:** Mapa de Karnaugh do Circuito Controlador do circuito com o contador 74LS163



Universidade de São Paulo
Escola Politécnica
Departamento de Eng. de Computação e Sistemas Digitais