

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO



PCS 3225 – SISTEMAS DIGITAIS II

ATIVIDADE FORMATIVA 5 – CONTADOR DE 1

Professor:

Dr. Marco Tulio Carvalho de Andrade

Grupo 10

NUSP

Guilherme Fortunato Miranda	13683786
João Pedro Dionizio Calazans	13673086
Lucas Pates Rodrigues	11545304
Matheus Tierro de Paula	11804021
Pedro Kooshi Ito Sartori	13683661

SÃO PAULO
2023

SUMÁRIO

1.	PROJETO LÓGICO	1
2.	DEFINIÇÃO DOS CASOS DE TESTE.....	2
3.	CODIFICAÇÃO DO CIRCUITO EM VHDL	3
4.	CODIFICAÇÃO DO TESTBENCH EM VHDL	3
5.	SIMULAÇÃO.....	3
6.	ANÁLISE DA VERIFICAÇÃO FUNCIONAL	4

1. PROJETO LÓGICO

O projeto de contador de 1 utilizou a conhecida estrutura de Unidade de Controle (UC) e Fluxo de dados (FD).

No fluxo de dados foram necessários dois componentes: um contador de 4 bits e um registrador deslocador de 15 bits.

O contador de 4 bits, a cada iteração do clock, recebe da UC o sinal de controle “zera” que comanda o recomeço ou não da contagem, e o sinal de controle “conta” que comanda o incremento ou não de 1 na contagem. Ademais, sua saída “Q” conecta-se a saída “outport” do contador de 1, para fornecer o número de 1’s ao final da contagem.

O registrador deslocador de 15 bits, a cada iteração do clock, recebe da UC o sinal de controle “carrega” a fim de armazenar ou não no registrador deslocador os bits da entrada “inport”, e o sinal de controle “desloca” a fim de deslocar ou não para a direita os dados armazenados no registrador deslocador. Ademais, sua saída “saida” fornece, por meio do FD, os sinais de condição para a UC, a saber, “zero” e “data0”, que informam se os dados armazenados no registrador deslocador são todos iguais a 0, e se o último bit é igual a 0, respectivamente.

Por fim, a UC determinou 6 possíveis estados: idle, load, go, count, shift, fins. Uma breve explicação do seu funcionamento se encontra a seguir.

- Idle ocorre quando o contador ainda não iniciou seu funcionamento (start = 0).
- Load é o estado imediatamente após o início do funcionamento do contador (start = 1). Apenas nele temos os sinais de controle “zera” e “carrega” iguais a 1.
- Go é um estado intermediário que ocorre após o estado Load, ou após o estado shift quando o sinal de condição “zero” for igual a 0.
- Count é o estado em que há incremento de 1 na contagem, por meio do sinal de controle “conta”. Ocorre após o estado Go, caso o sinal de condição “data0” for igual a 1, ou seja, último bit dos dados armazenados no registrador deslocador é 1.

- Shift é o estado em que há deslocamento para a direita dos dados armazenados no registrador deslocador, por meio do sinal de controle “desloca”. Ocorre sempre após o estado Count, ou após o estado Go, caso o sinal de condição “data0” for igual a 0.
- Fins é o estado que recebe “done” = 1, indicando o final da contagem. Ocorre quando o sinal de condição “zero” = 1, ou seja, todos os bits armazenados no registrador deslocador são iguais a 0.

2. DEFINIÇÃO DOS CASOS DE TESTE

A fim de verificar o correto funcionamento do circuito, o grupo decidiu por selecionar um subconjunto representativo de casos de teste. Na tabela 1 é possível verificar a heterogeneidade, dentro do possível, dos casos selecionados. Por exemplo, há casos apenas com 0's ou com 1's; que alternam entre 0's e 1's; que possuem apenas um bit 1; que possuem grandes sequências de 1's em seu corpo; com as duas extremidades iguais a 0; extremidades iguais a 1; extremidades mistas; que possuem quantidade de 0's e de 1's próximas; e combinações variadas dessas características.

É válido notar que a estratégia de teste exaustivo é impraticável para este projeto, dado que são 15 bits de entrada, o que leva a 32768 possibilidades.

Tabela 1: Casos de teste do testbench.

Inport	Outport
000000000000010	0001
110001011101101	1001
011000000111001	0110
000000000000000	0000
111111111111111	1111
001111111110010	1010
101010101010101	1000
010101010101010	0111
000001111100000	0101
110110101101001	1001

Fonte: De autoria própria.

3. CODIFICAÇÃO DO CIRCUITO EM VHDL

Enviado em arquivo ZIP.

4. CODIFICAÇÃO DO TESTBENCH EM VHDL

Enviado em arquivo ZIP. Foi utilizada leitura de arquivo para entrada dos casos testados.

5. SIMULAÇÃO

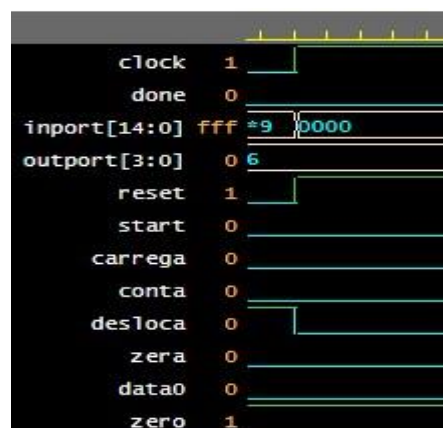
As figuras 1, 2 e 3 apresentam os resultados da simulação.

Figura 1: Mensagens de execução da simulação

```
testbench.vhd:60:7:@0ms:(assertion note): simulation start
testbench.vhd:80:9:@8500ps:(assertion note): Teste OK
testbench.vhd:80:9:@50500ps:(assertion note): Teste OK
testbench.vhd:80:9:@87500ps:(assertion note): Teste OK
testbench.vhd:80:9:@92500ps:(assertion note): Teste OK
testbench.vhd:80:9:@140500ps:(assertion note): Teste OK
testbench.vhd:80:9:@179500ps:(assertion note): Teste OK
testbench.vhd:80:9:@220500ps:(assertion note): Teste OK
testbench.vhd:80:9:@258500ps:(assertion note): Teste OK
testbench.vhd:80:9:@286500ps:(assertion note): Teste OK
testbench.vhd:80:9:@328500ps:(assertion note): Teste OK
testbench.vhd:84:7:@328500ps:(assertion note): simulation end
```

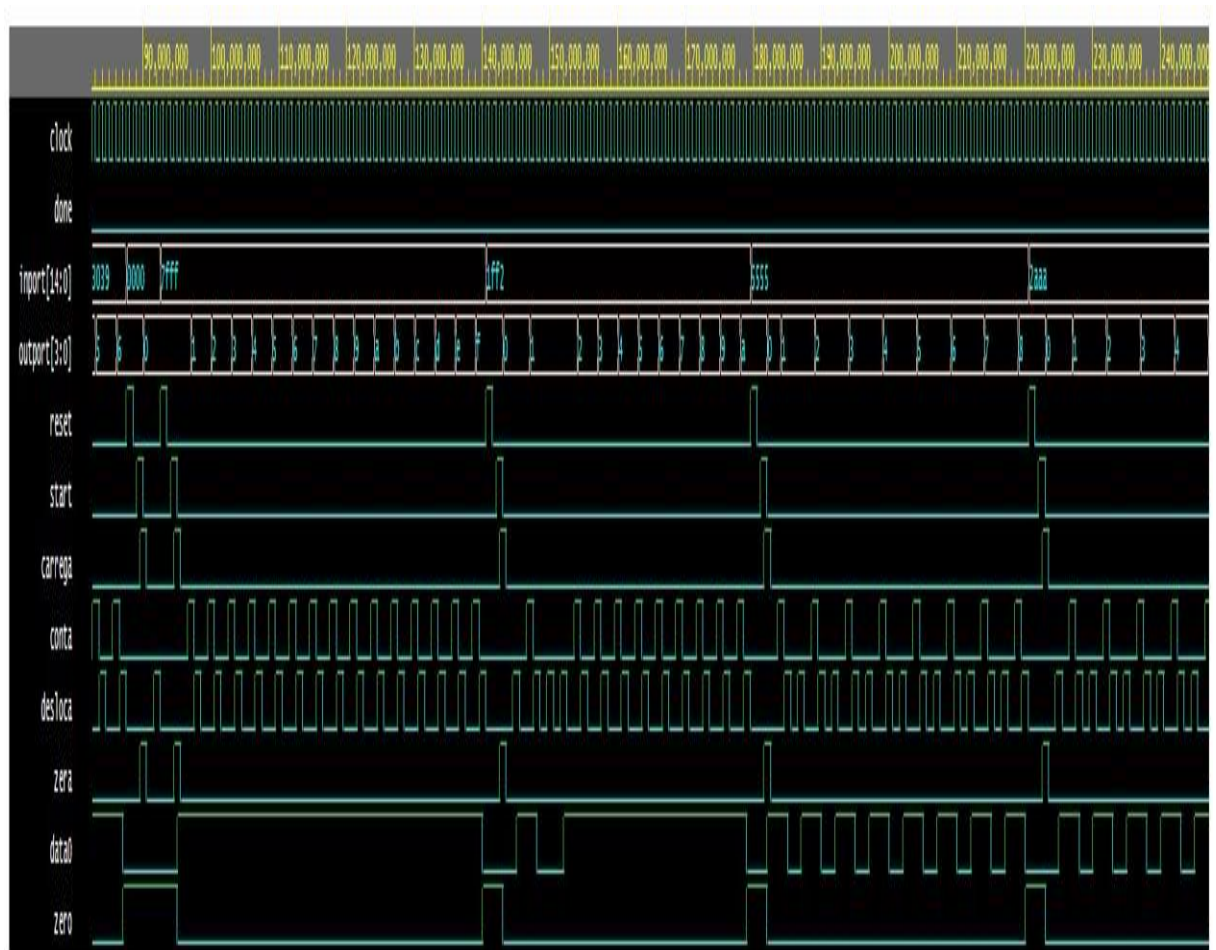
Fonte: De autoria própria via EDA Playground.

Figura 2: Sinais analisados na ferramenta EPWave.



Fonte: De autoria própria via EDA Playground.

Figura 3: Análise completa dos sinais na simulação.



Fonte: De autoria própria via EDA Playground.

6. ANÁLISE DA VERIFICAÇÃO FUNCIONAL

Tendo em vista que todos os casos de teste passaram na simulação, e que, conforme explicitado na seção 2, os casos foram criteriosamente selecionados a fim de alcançar, dentro do possível, a heterogeneidade da amostra, conclui-se que há alta confiabilidade no correto funcionamento do DUT (contador de 1).