



Departamento de Engenharia Elétrica
Universidade de Brasília
Disciplina: Laboratório de Sistemas Digitais (Turma: 04)
Professor: Guilherme Torres

Primeiro Semestre de 2024

Experimento 6

Datas da realização do experimento: 2 de maio de 2024

Aluno: Felipe Lopes Gibin Duarte

Matrícula: 231025207

1. INTRODUÇÃO

Neste experimento, vamos construir circuitos sequenciais, que utilizam elementos de memória, ou seja, o output não depende somente do input atual, mas também de inputs anteriores. Particularmente, construiremos um flip-flop e um registrador de deslocamento bidirecional, tudo isso usando a linguagem de descrição de hardware VHDL. Além disso, vamos ter um primeiro contato com a estrutura process da linguagem de descrição. Desse modo, todos os procedimentos serão desenvolvidos usando o software “ModelSim”. Para alcançar tais objetivos, o seguinte foi feito.

2. DESENVOLVIMENTO

Questão 1

Usando a estrutura “process”, implementar em VHDL e simular no ModelSim um flip-flop JK gatilhado pela borda de subida, com funcionamento descrito pela tabela verdade abaixo.









entradas					saída
<i>PR</i>	<i>CLR</i>	<i>CLK</i>	<i>J</i>	<i>K</i>	<i>Q</i>
1	x	x	x	x	1
0	1	x	x	x	0
0	0	┐	0	0	mantém
0	0	┐	0	1	0
0	0	┐	1	0	1
0	0	┐	1	1	inverte
0	0	outros	x	x	mantém

-O código documentado está no segundo arquivo

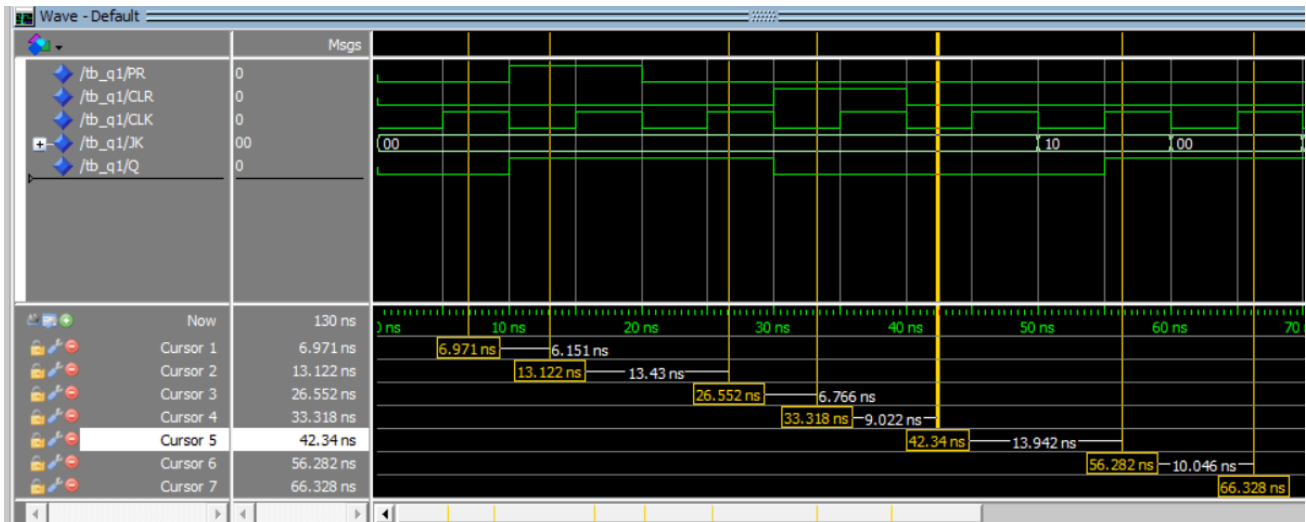
-Confirmação do código compilado:

```
# Compile of tb_Q1(6).vhd was successful with warnings.
# Compile of Q1(6).vhd was successful.
```

ModelSim>

	tb_Q1(6).vhd		VHDL	3	05/07/2024 08:32:04 ...
	Q2(6).vhd		VHDL	1	05/07/2024 07:02:42 ...
	tb_Q2(6).vhd		VHDL	2	05/07/2024 07:02:55 ...
	Q1(6).vhd		VHDL	0	05/07/2024 08:33:16 ...

-Simulação do código no ModelSim:



-OBS: Foi analisado somente os instantes em que o clock subiu (borda de subida é relevante nesse caso)

-Análise:

Cursor 1) Situação inicial, tudo zero

Cursor 2) PR= 1, CLR= 0, CLK= 0, JK= 00, Q= 1 (Preset, note que clk não importa)

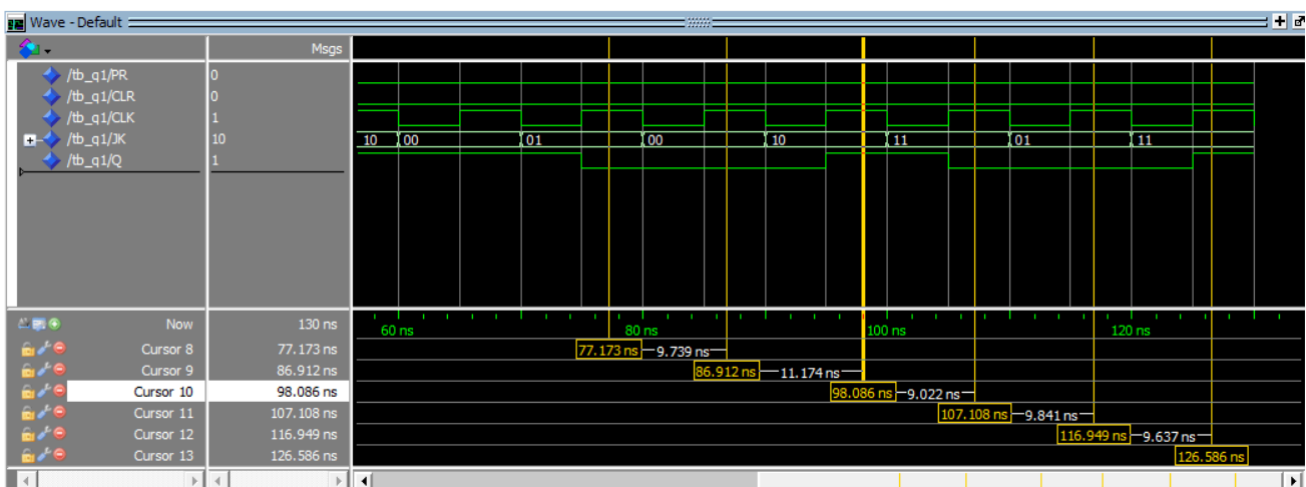
Cursor 3) PR= 0, CLR= 0, CLK= 1, JK= 00, Q= 1 (Mantém)

Cursor 4) PR= 0, CLR= 1, CLK= 0, JK= 00, Q= 0 (Clear, note que clk não importa)

Cursor 5) PR= 0, CLR= 0, CLK= 0, JK= 00, Q= 1 (Mantém)

Cursor 6) PR= 0, CLR= 0, CLK= 1, JK= 10, Q= 1 (Set)

Cursor 7) PR= 0, CLR= 0, CLK= 1, JK= 00, Q= 1 (Mantém)



Cursor 8) PR= 0, CLR= 0, CLK= 1, JK= 01, Q= 0 (Reset)

Cursor 9) PR= 0, CLR= 0, CLK= 1, JK= 00, Q= 0 (Mantém)

Cursor 10) PR= 0, CLR= 0, CLK= 1, JK= 10, Q= 1 (Set)

Cursor 11) PR= 0, CLR= 0, CLK= 1, JK= 11, Q= 0 (Inverte)

Cursor 12) PR= 0, CLR= 0, CLK= 1, JK= 01, Q= 0 (Reset)

Cursor 13) PR= 0, CLR= 0, CLK= 1, JK= 11, Q= 1 (Inverte)

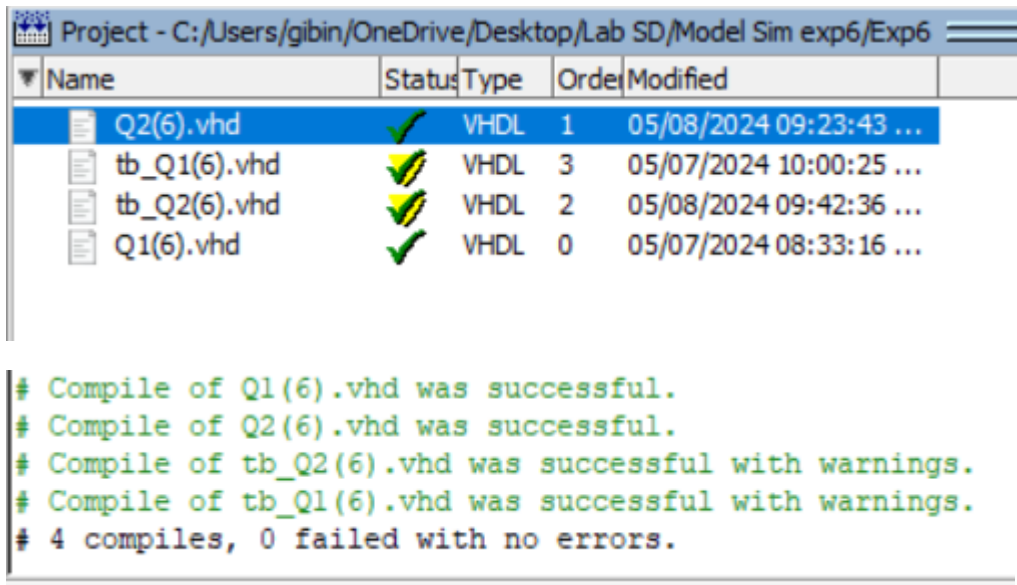
QUESTÃO 02.

Usando a estrutura “process”, implementar em VHDL e simular no ModelSim um registrador de deslocamento bidirecional de 4 bits, com funcionamento descrito pela tabela verdade abaixo. Respeite a ordem de significância dos bits dos vetores D e Q, de modo que os bits menos significativos (D0 e Q0) fiquem em posições à direita dos mais significativos.

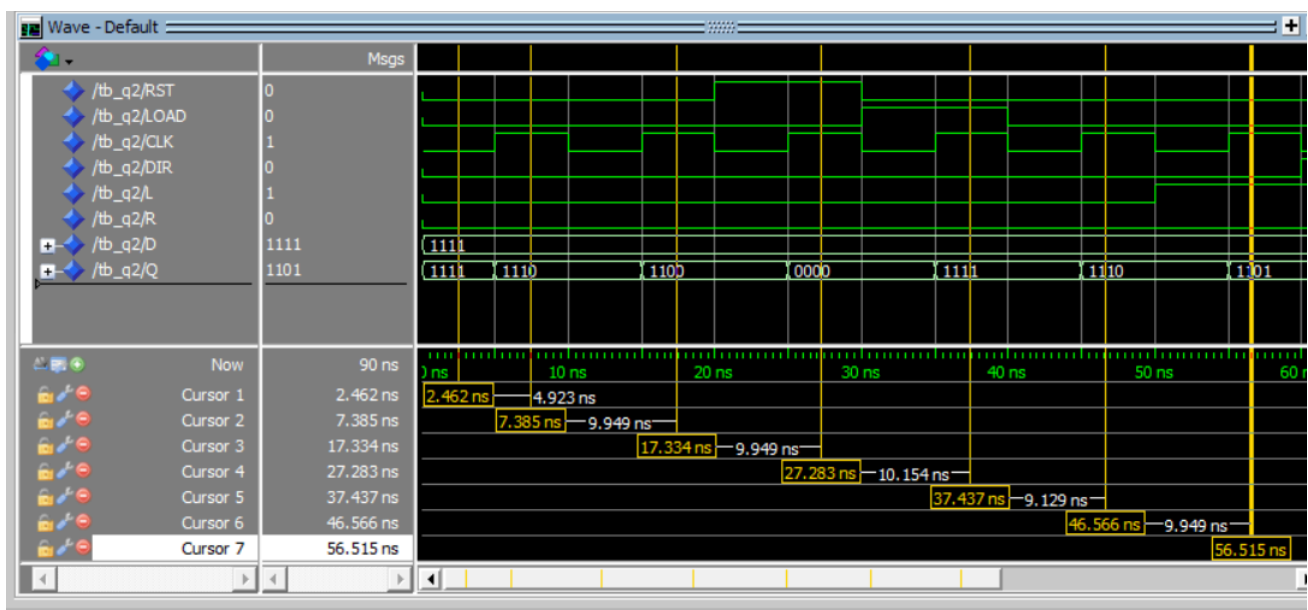
entradas							saída
CLK	RST	LOAD	D	DIR	L	R	Q
\downarrow	1	x	xxxx	x	x	x	0000
\downarrow	0	1	$D_3D_2D_1D_0$	x	x	x	$D_3D_2D_1D_0$
\downarrow	0	0	xxxx	0	0	x	$Q_2Q_1Q_00$
\downarrow	0	0	xxxx	0	1	x	$Q_2Q_1Q_01$
\downarrow	0	0	xxxx	1	x	0	$0Q_3Q_2Q_1$
\downarrow	0	0	xxxx	1	x	1	$1Q_3Q_2Q_1$
outros	x	x	xxxx	x	x	x	$Q_3Q_2Q_1Q_0$

-O código documentado está no segundo arquivo

-Confirmação do código compilado:



-Simulação do código no ModelSim:



-OBS: Foi analisado somente os instantes em que o clock subiu (borda de subida é relevante nesse caso)

-Análise:

Cursor 1) Situação inicial

Cursor 2) RST= 0, LOAD= 0, CLK= 1, DIR= 0, L=0, R= 0, D= 1111 Q= 1110 (Shift left com zero)

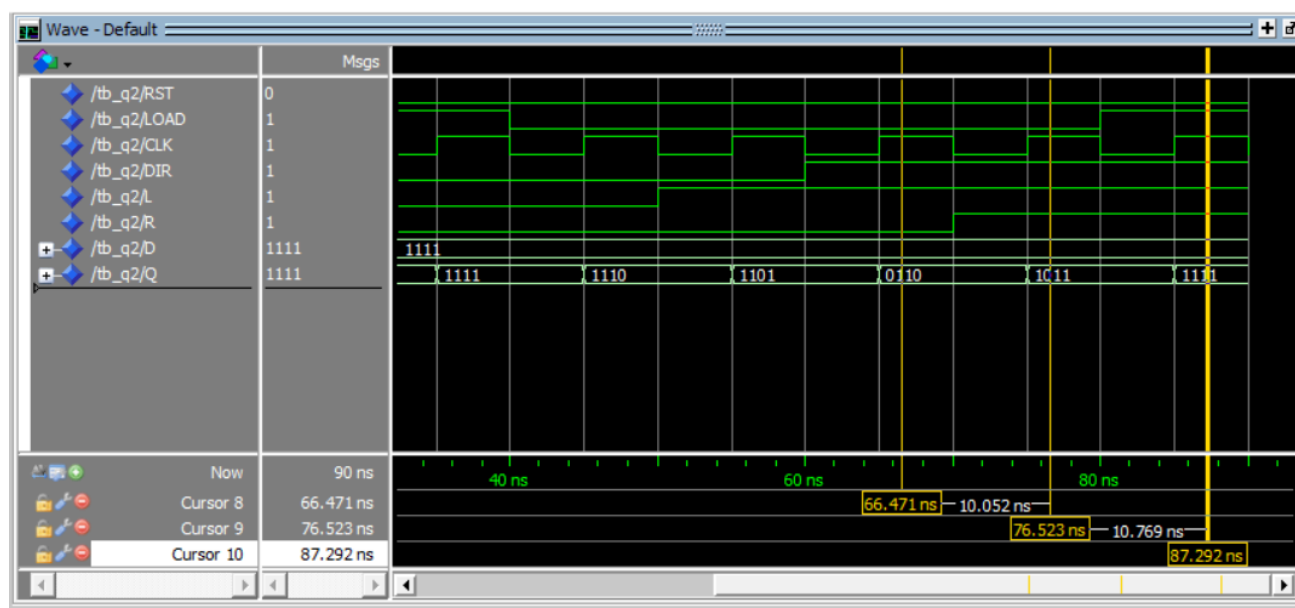
Cursor 3) RST= 0, LOAD= 0, CLK= 1, DIR= 0, L=0, R= 0, D= 1111 Q= 1100 (Shift left com zero)

Cursor 4) RST= 1, LOAD= 0, CLK= 1, DIR= 0, L=0, R= 0, D= 1111 Q= 0000 (Reset)

Cursor 5) RST= 0, LOAD= 1, CLK= 1, DIR= 0, L=0, R= 0, D= 1111 Q= 1111 (Load)

Cursor 6) RST= 0, LOAD= 0, CLK= 1, DIR= 0, L=0, R= 0, D= 1111 Q= 1110 (Shift left com zero)

Cursor 7) RST= 0, LOAD= 0, CLK= 1, DIR= 0, L=0, R= 0, D= 1111 Q= 1101 (Shift left com um)



Continuação da análise:

Cursor 8) RST= 0, LOAD= 0, CLK= 1, DIR= 1, L=1, R= 0, D= 1111 Q= 0110 (Shift right com zero)

Cursor 9) RST= 0, LOAD= 0, CLK= 1, DIR= 1, L=1, R= 1, D= 1111 Q= 1011 (Shift right com um)

Cursor 10) RST= 0, LOAD= 1, CLK= 1, DIR= 1, L=1, R= 1, D= 1111 Q= 1111 (Load)

3. CONCLUSÃO

Após todos os procedimentos, houve uma compreensão mais sólida acerca dos conceitos básicos de design digital usando VHDL. Nesse experimento em particular, construímos um flip flop JK e um registrador de deslocamento bidirecional de 4 bits, ambos engatilhados na borda de subida. Aprendemos como usar a construção process durante o experimento, de forma que o código dentro dela é analisado sequencialmente. Os maiores obstáculos encontrados no experimento foram erros de indefinição da saída Q, de forma que foi necessário declarar um valor padrão para a saída no começo. Por fim, a experiência foi proveitosa ao aprofundar conceitos do VHDL como o process e ao proporcionar um primeiro contato com flip flops e registradores.