

Laboratório de Sistemas Digitais

Experimento 06

OBJETIVOS:

- Implementar circuitos sequenciais (um flip-flop e um registrador de deslocamento bidirecional) em VHDL e simular no ModelSim usando a estrutura “process” da linguagem VHDL.

INSTRUÇÕES:

- O experimento deve ser realizado utilizando o ModelSim;
 - Cada experimento será avaliado por meio do relatório técnico e dos códigos submetidos pelo aluno, por meio da plataforma Aprender. Os códigos devem ser submetidos comprimidos em um único arquivo.
 - A sua simulação deve incluir o arquivo vhd contendo a entidade (*entity*), a arquitetura (*architecture*) do circuito e o arquivo vhd do *test bench* desenvolvido para simulá-los. Conforme descrito no guia de uso, o seu relatório deve conter os códigos, as telas de compilação e simulação do ModelSim e as formas de ondas obtidas com a simulação.
 - O relatório é individual e receberá uma nota de 0 a 10, considerando os seguintes aspectos:
 - Documentação do código, contida no relatório (pdf) e no código vhd - 20% da nota do projeto;
 - Compilação do código, apresentada no relatório do projeto e confirmado pelo código vhd - 10% da nota do projeto;
 - Simulação do código, apresentada no relatório do projeto e confirmado pelo código vhd - 70% da nota do projeto.
 - Os códigos VHDL das entidades e arquiteturas desenvolvidas neste experimento e no experimento anterior serão utilizadas em experimentos futuros para construir sistemas mais complexos.
-

QUESTÃO 01.

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

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
<i>CLK</i>	<i>RST</i>	<i>LOAD</i>	<i>D</i>	<i>DIR</i>	<i>L</i>	<i>R</i>	<i>Q</i>
↑	1	x	xxxx	x	x	x	0000
↑	0	1	$D_3D_2D_1D_0$	x	x	x	$D_3D_2D_1D_0$
↑	0	0	xxxx	0	0	x	$Q_2Q_1Q_00$
↑	0	0	xxxx	0	1	x	$Q_2Q_1Q_01$
↑	0	0	xxxx	1	x	0	$0Q_3Q_2Q_1$
↑	0	0	xxxx	1	x	1	$1Q_3Q_2Q_1$
outros	x	x	xxxx	x	x	x	$Q_3Q_2Q_1Q_0$