

Computação Reconfigurável

Aula prática 1 (versão 2019)

Resumo:

1. Introdução ao sistema Vivado (versão 2018.3).
2. Introdução à placa de prototipagem *Nexys-4/ZyBo*.
3. Projetos simples para FPGA.

Objetivos

Realização de um conjunto de exercícios práticos introdutórios para:

- Perceber o nível básico do sistema Vivado 2018.3;
- Perceber utilização da placa *Nexys-4/ZyBo*;
- Revisão da linguagem VHDL através da criação de projetos simples;
- Desenvolvimento, síntese, implementação e verificação de circuitos simples em FPGA.

Materiais adicionais úteis:

- Xilinx Design Constraints File XDC: [elearning <https://github.com/Digilent/digilent-xdc/blob/master/Nexys-4-Master.xdc>](https://github.com/Digilent/digilent-xdc/blob/master/Nexys-4-Master.xdc).
- Manual de utilização da placa *Nexys-4* : https://reference.digilentinc.com/_media/nexys:nexys4:nexys4_rm.pdf .

Problemas para resolver

Exercício 1.1. Mostrar os valores dos interruptores nos LEDs (verificar a instrução $\text{LED} \leq \text{SW}_i$). Mostrar os valores dos botões nos LED (verificar instruções $\text{LED}(0) \leq \text{BTNL}$; $\text{LED}(1) \leq \text{BTNC}$; $\text{LED}(2) \leq \text{BTNR}$; $\text{LED}(3) \leq \text{BTNU}$; $\text{LED}(4) \leq \text{BTND}$).

Exercício 1.2. Use os interruptores $\text{Sw}_0, \dots, \text{Sw}_3$ para escolher um botão (BTNU, BTNC, BTND, BTNL, BTNR) ou um interruptor $\text{Sw}_4, \dots, \text{Sw}_{15}$. Mostre o valor do interruptor ou do botão escolhido no LED₀. Exemplo: quando $\text{Sw}_0, \dots, \text{Sw}_3 = "000"$ mostrar o valor BTNU; quando $\text{Sw}_0, \dots, \text{Sw}_3 = "001"$ mostrar o valor BTNC; etc.

Exercício 1.3. Implementar um somador completo para dois operandos de um bit A (Sw_0) e B (Sw_1). Usar interruptor Sw_2 para sinal de entrada *carry in*. Mostrar o valor da soma no LED₀ e o valor do sinal *carry out* em LED₁.

Exercício 1.4. Fazer ligação com um divisor de frequência 100 MHz → 1 Hz. Mostrar a saída do divisor no LED₃.

Exercício 1.5. Ligar/desligar LED₃ com frequência aproximada de 1 Hz. Ligar/desligar LED₄ com frequência aproximada de 0.5 Hz. Ligar/desligar LED₅ com frequência aproximada de 0.25 Hz.

Exercício 1.6. Implemente um contador de 4 bits. Entradas são a) relógio com frequência 1 Hz (mais ou menos); b) *clock enable* (Sw_0) para permitir relógio na entrada; c) *reset* (BTNC) para fazer reset (i.e. todas as saídas são iguais a zero); d) *inc* ($\text{Sw}_1 = 0$) para incrementar o valor de saída com frequência de relógio; d) *dec* ($\text{Sw}_1 = 1$) para decrementar o valor de saída com frequência de relógio. Mostrar o conteúdo do contador nos LEDs (4 LEDs totalmente).

Exercício 1.7. Implemente uma função Booleana que permite verificar se o valor marcado nos interruptores SW_0, \dots, SW_{15} é par ou ímpar. Mostre o valor da função no LED₀.

Exercício 1.8. Mostre nos LEDs o valor $(Sw_0, \dots, Sw_7) + (Sw_8, \dots, Sw_{15})$ quando o botão BTNU está pressionado. Mostre nos LEDs o valor $(Sw_0, \dots, Sw_7) - (Sw_8, \dots, Sw_{15})$ quando o botão BTND está pressionado. Mostre nos LEDs o valor $(Sw_0, \dots, Sw_7) * (Sw_8, \dots, Sw_{15})$ quando o botão BTNL está pressionado. Mostre nos LEDs o valor $(Sw_0, \dots, Sw_7) / (Sw_8, \dots, Sw_{15})$ quando o botão BTNR está pressionado. Mostre nos LEDs o valor $(Sw_0, \dots, Sw_7) \% (Sw_8, \dots, Sw_{15})$ quando o botão BTNC está pressionado, onde % é o resto da divisão.

Exercício 1.9. Implemente um registo de deslocamento de 16 bits. Entradas são a) relógio com frequência 1 Hz (mais ou menos); b) *clock enable* para permitir relógio na entrada; c) *reset* para fazer reset (i.e. todas as saídas são iguais a zero); d) SW_0, \dots, SW_{15} entradas paralelas; e) *set* para gravar no registo entradas paralelas; f) *left* para fazer *shift* à esquerda com frequência de relógio; g) *right* para fazer *shift* à direita com frequência de relógio. Mostrar o conteúdo do registo nos LEDs (16 LEDs totalmente).

Exercício 1.10. Implemente um sistema de funções Booleanas para encontrar o *Hamming weight* de vetor binário com 3 bits: $(Sw_0, \dots, Sw_2) \rightarrow (LED_0, LED_1)$.

Exercício 1.11. Implemente um sistema de funções Booleanas para fazer conversão de códigos binários para valor hexadecimal nos segmentos dum display: $(Sw_0, \dots, Sw_3) \rightarrow (SEG_0, \dots, SEG_7)$. Usar várias possibilidades, nomeadamente a) definir uma constante; b) usar instrução **when ... else** dentro de uma arquitetura; c) usar instrução **case ... when** dentro de um processo; d) usar instrução **with ... select** dentro de uma arquitetura.