Elementos de Sistema - Prova 01 - Prática

Nome completo:		
Pontos de:	Н₩	SW
	/ 45	/ 15

Instruções:

- 1. A avaliação tem duração total de 120 minutos.
- 2. Você não pode consultar a internet, apenas seu repositório LOCAL
- **3.** Você deve editar esse documento.
- **4.** Assim como nos projetos, os códigos fontes estão em: /src/rtl/ e o arquivo de configuração dos testes em /test/config.txt

1. (7 pts HW) Circuito

a) (5 pts HW) VHDL

Implemente em VHDL o circuito a seguir:

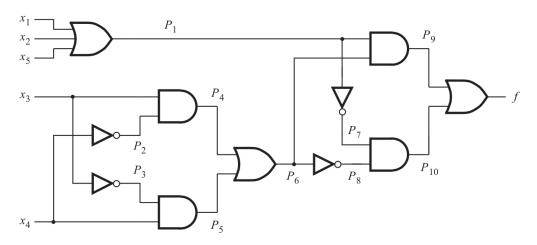


Figure 4.30 Circuit for Example 4.11.

Arquivo	/src/rtl/questao1.vhd
Deve passar no teste?	SIM

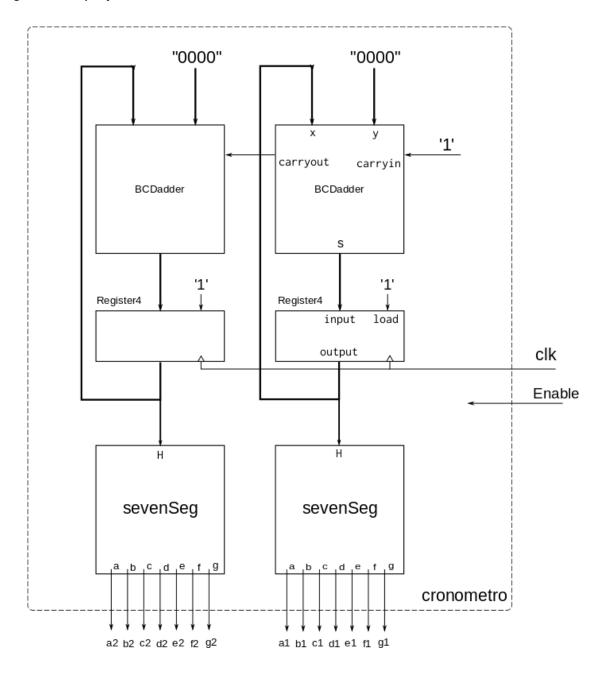
b) (2 pts HW) waveform

COLE AQUI A FORMA DE ONDA DO CIRCUITO

2. (30 pts HW) Cronometro

Você está desenvolvendo um hardware dedicado que será um cronometro de segundos, nesse equipamento você terá dois displays de sete segmentos, um para cada digito do seu cronômetro. O sinal de clock será um pulso de um segundo, fazendo com que o cronômetro conte "segundos".

Para desenvolver tal hardware, você irá utilizar o somador de BCD desenvolvido anteriormente. Em uma conversa com o seu professor, vocês chegaram no seguinte diagrama do projeto:



a) (7 pts HW) Você foi mostrar o diagrama para o seu grupo e eles não entenderam nada, explique como ele funciona:

ESCREVA AQUI SUA EXPLICAÇÃO

b) (5 pts HW) Utilizando os componentes **sevenSeg**, **Register4**, **BCDAdder**, implemente a entidade cronometro.

Implemente o componente Register4 que armazena um vetor de 4 bits.

Arquivo	/src/rtl/register4/Register4.vhd
Possui teste?	Sim, mas deve implementar todos os outros módulos antes,.

c) (2 pts HW) Gere o RTL da entidade cronometro

COLE AQUI O RTL

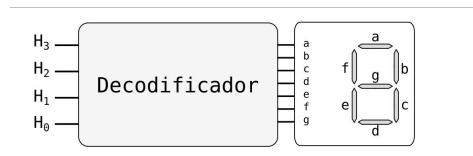
d) (3 pts HW) Implemente o componente Register4

Os componentes necessários para implementar o Register4 já estão prontos, são eles: **BinaryDigit, FlipFlopD** e **Mux2Way.**

Arquivo	/src/rtl/register4/Register4.vhd
Possui teste?	SIM

e) (5 pts HW) Implemente o componente sevenSeg (do dígito 0 até o dígito 9)

Você deve projetar um decodificador de números binários (H3, H2, ...) para um display de sete segmentos (g,f,e,...) que deve exibir os dígitos: 0,1,2,3,4,5,6,7,8,9 . Nesse projeto é utilizado um display de anodo comum, onde o segmento é aceso se acionado em nível baixo (0) e apagado em nível alto (1).



Arquivo	/src/rtl/sevenSeg.vhd
Deve passar no teste?	SIM

f) (5 pts HW) Melhoria

Repare que no diagrama do cronometro o sinal '**enable'** não está sendo utilizado. Esse sinal viria de um botão do equipamento que habilitaria a contagem de tempo ou não. A ideia é que se '**enable = 1**' o cronometro começa a contar os segundos e se '**enable = 0**' ele pararia a contagem, e só retornaria quando o sinal voltasse a ser '1'.

Desenhe no diagrama como utilizaria esse sinal para que essa lógica funcionasse

Gere um novo RTL do hardware com o sinal de enable implementado. Faça uma cópia do arquivo **cronometro.vhd** com o nome cronometroMelhoria.vhd para testar!

COLE AQUI O RTL

2. (3 HW, 15 SW)

Você deve escrever um programa em Assembly Z01.1 que faz com que os LEDs da FPGA acendam da maneira detalhada a seguir:

 LED[8]

 SW[8]
 !SW[7]
 OFF
 ON
 ON
 RAM[5]
 ON
 SW[0]
 OFF

- **!SW[3]**: Inverso que a terceira chave indica.

- **RAM[5][3]:** Bit 3 do que estiver na RAM[5]

ON : LED sempre ligadoOFF : LED sempre desligado

Arquivo	/src/nasm/SWeLED.nasm
Deve passar no teste?	SIM: testeNASM.py

Função	Pontos
X X X OFF ON ON X ON X OFF	0 HW / 2 SW
SW[8] X X X X X X X X	1 HW/ 2 SW
X !SW[7] X X X X X X	1 HW/ 3 SW
X X X X RAM[5][3] X X X	0 HW/ 4 SW
X X X X X X X SW[0] X	0 HW/ 4 SW
Grava no LED	1 HW/ 0 SW