

# *PCS3115 - Sistemas Digitais I*

## *Projeto 1 - Jokempô Melhor de Três*

*Glauber De Bona*

*04/05/2021*

O objetivo deste trabalho é exercitar a descrição de circuitos combinatórios em VHDL, bem como o uso de componentes, com a modelagem estrutural. Como consequência, este trabalho oferece um primeiro contato com a linguagem VHDL, a preparação de testbenchs e simulação.

### *Introdução*

Jokempô é um jogo onde cada um dos dois jogadores faz um gesto com a mão representando Pedra, Papel ou Tesoura. Caso usem o mesmo gesto, há um empate, caso contrário, o vencedor é definido da seguinte forma:

- Pedra vence Tesoura;
- Tesoura vence Papel;
- Papel vence Pedra.

Dois amigos iniciaram uma rivalidade neste jogo, que logo se tornou uma obsessão. Para agilizar os longos embates, formados por sequências de jogos, eles começaram a jogar com as duas mãos em paralelo, fazendo dois jogos de cada vez. Mas logo isso também se tornou lento e entediante, e os dois rivais agora tiveram a ideia de usar um circuito digital para fazer três jogos em paralelo.

Os gestos seriam codificados em 2 bits: Pedra (01), Papel (10) e Tesoura (11). Cada jogador controlaria 6 bits de entrada (com uma espécie de teclado), correspondendo aos gestos nos três jogos paralelos. A saída informaria quem venceu mais entre as três disputas, descontando os empates. O resultado da disputa (e de cada um dos 3 jogos) pode ser codificado usando também 2 bits: 10 para vitória do jogador A, 01 para a vitória do jogador B e 11 para empate. A saída 00 denota um estado de espera, quando um jogador ainda não incluiu um gesto (entrada 00).

Sua missão é projetar este sistema digital!

### *Atividades*

**T1A1** (3 pontos) Implemente um componente em VHDL que determine o vencedor de um jogo de Jokempô, com a seguinte entidade:

Trabalho 1, Atividade 1

```

entity jokempo is
port(
  a: in bit_vector(1 downto 0);  —! gesto do jogador A
  b: in bit_vector(1 downto 0);  —! gesto do jogador B
  y: out bit_vector(1 downto 0)   —! resultado do jogo
);
end jokempo;

```

As entradas a e b (2 bits cada) representam os gestos dos jogadores A e B (Pedra, Papel ou Tesoura), e a saída y (2 bits), codifica o resultado o jogo, conforme explicado acima. Note que, se  $a = 00$  ou  $b = 00$ , algum dos jogadores ainda não jogou, e a saída deve ser  $y = 00$ .

Por exemplo, se o Jogador A jogou pedra,  $a = 01$ .

**T1A2** (5 pontos) Implemente um componente em VHDL que determine o vencedor de uma disputa melhor de três, obedecendo a entidade abaixo:

Trabalho 1, Atividade 2

```

entity melhordetres is
port(
  resultado1: in bit_vector(1 downto 0);  —! resultado dos jogo 1
  resultado2: in bit_vector(1 downto 0);  —! resultado dos jogo 2
  resultado3: in bit_vector(1 downto 0);  —! resultado dos jogo 3
  z:          out bit_vector(1 downto 0)   —! resultado da disputa
);
end melhordetres;

```

As entradas resultado1, resultado2 e resultado3 (2 bits cada) representam o resultado de três jogos, e a saída z (2 bits), codifica o resultado da disputa melhor de três, conforme explicado acima.

Se algum resultado ainda está em espera (00), a saída deve ser  $z = 00$ .

**T1A3** (2 pontos) Implemente um componente em VHDL que determine o vencedor de uma disputa melhor de três no Jokempô, obedecendo a entidade abaixo:

Trabalho 1, Atividade 3

```

entity jokempotriplo is
port(
  a1, a2, a3: in bit_vector(1 downto 0);  —! gesto do jogador A para os 3 jogos
  b1, b2, b3: in bit_vector(1 downto 0);  —! gesto do jogador B para os 3 jogos
  z:          out bit_vector(1 downto 0)   —! resultado da disputa
);
end jokempotriplo;

```

As entradas a1, a2 e a3 (2 bits cada) correspondem aos gestos do jogador A em cada um dos 3 jogos (idem para o jogador B). A saída z mostra resultado do embate melhor de três. Utilize 3 componentes jokempo da atividade T1A1 e um componente melhordetres da atividade T1A2 na implementação de jokempotriplo.

### Instruções para Entrega

Você deve acessar o link específico para cada tarefa (T1A1, T1A2 e T1A3) dentro do tópico “Projetos” no e-Disciplinas, já logado com

seu usuário e senha, que levará à página apropriada do juiz eletrônico. Em cada atividade, você pode enviar apenas um único arquivo codificado em UTF-8. O nome do arquivo não importa, mas sim a descrição VHDL que está dentro. As entidades nas suas soluções devem ser idênticas àsquelas neste enunciado ou o juiz não irá processar seu arquivo.

Quando acessar o link no e-Disciplinas, o navegador abrirá uma janela para envio do arquivo. Selecione-o e envie para o juiz. Jamais recarregue a página de submissão pois seu navegador pode enviar o arquivo novamente, o que vai ser considerado pelo juiz como um novo envio e pode prejudicar sua nota final. Caso desista do envio, simplesmente feche a janela. Depois do envio, a página carregará automaticamente o resultado do juiz, quando você poderá fechar a janela. A nota dada pelo juiz é somente para a submissão que acabou de fazer.

O prazo para a submissão das soluções no Juiz é 19 de maio de 2021, quarta-feira, às 23:59. O Juiz aceitará até 5 submissões para cada atividade deste projeto. Sua submissão será corrigida imediatamente e sua nota será apresentada. A maior nota dentre as submissões será considerada. Neste trabalho, os problemas valem no máximo 10 pontos no juiz, porém a nota final deste trabalho será calculada com as ponderações indicadas em cada atividade neste enunciado, totalizando 10 para o trabalho todo. Como boa prática de engenharia, faça seus *test-benches* e utilize o GHDL ou o EDA Playgorund para validar suas soluções antes de postá-las no juiz.

**Atenção:** Para as três atividades do projeto, está proibido o uso de process ou das bibliotecas `std_logic_1164` e `textio`, ou de qualquer biblioteca não padronizada. Se seu arquivo mencionar essas bibliotecas, mesmo em um comentário, sua submissão nem será avaliada pelo juiz e ficará com nota 0 (zero).