

PCS3115 - Sistemas Digitais I

Projeto 4 - Jokempô PRO

Glauber De Bona

Prazo: 28/07/2021

O objetivo deste trabalho é a prática da metodologia de projeto via Fluxo de Dados e Unidade de Controle com diagramas ASM.

Introdução

Não satisfeitos com a limitação do JKP^3 a disputas do tipo melhor de três, os jogadores profissionais de Jokempô estão demandando um sistema que permita disputas com um número arbitrário de partidas. A empresa JKP^3 , onde você trabalha, está com planos ambiciosos para mudar de nome para JKP^n , lançando um sistema de mesmo nome que permitiria os jogadores escolher o número n de partidas a serem disputadas.

Veja o enunciado dos Projetos 1 e 2.

Para isso, seu chefe deseja um sistema onde os jogadores coloquem um gesto de cada vez, serialmente, com um placar mostrando quantas vitórias cada jogador já teve e quantas rodadas restam. No início, os jogadores entrariam o número n de rodadas desejadas, e o sistema seria capaz de avisar quando algum jogador ganhou o *match*, mesmo que ainda falem rodadas. Saídas extras mostrariam o resultado da última rodada e o resultado do match. O gesto de cada jogador seria representado por 2 bits, assim como o resultado de uma rodada e do match, utilizando a mesma codificação dos projetos anteriores.

Sua tarefa será desenvolver este novo sistema, o JKP^n , utilizando a metodologia de projeto estruturado em Unidade de Controle e Fluxo de Dados, partindo de um diagrama ASM.

Atividades

T4A1 (10 pontos) Implemente um componente em VHDL (com a entidade abaixo) que permita disputas de Jokempô compostas de n rodadas.

Trabalho 4, Atividade 1

```
entity jkpn is
  port (
    reset, clock:      in bit;
    inicia, carrega:   in bit;
    nJogos:             in bit_vector(3 downto 0);
    gestoA, gestoB:     in bit_vector(1 downto 0);
    zRodada, zMatch:    out bit_vector(1 downto 0);
    jogosRestantes:     out bit_vector(3 downto 0);
    placarA, placarB:   out bit_vector(3 downto 0)
  );
end jkpn;
```

A ativação da entrada *reset*, assíncrona, leva o sistema a um estado inicial, onde as saídas *PlacarA*, *PlacarB* e *zMatch* são ze-

radas. A entrada `nJogos` especifica em 4 bits o número (de 0 a 15) de rodadas desejadas. Para iniciar uma disputa (um match), a entrada `inicia` deve ser ativada com `nJogos` diferente de zero. Uma vez iniciado o match, entre os jogadores A e B, cada rodada é iniciada com a ativação da entrada `carrega`, desde que os gestos de ambos os jogadores (nas entradas `gestoA` e `gestoB`) sejam diferentes de “00” (espera). Iniciada uma rodada, o sistema deve decidir o seu vencedor (A, B ou empate), de acordo com os gestos na entrada, e escrever o resultado na saída `zRodada`. A cada rodada, o placar deve ser atualizado: as saídas `placarA` e `placarB` (4 bits cada) contam as rodadas vencidas pelos jogadores A e B, respectivamente (empates são ignorados); a saída `jogosRestantes` mostra quantas rodadas ainda restam no match, já descontando a atual; a saída `zMatch` mostra o resultado do match. Se o match ainda não terminou (isto é, são necessárias mais rodadas para decidir seu resultado), `zMatch` deve ser igual a “00”. Se o match terminou (não há mais rodadas restantes, ou algum jogador já ganhou por antecipação), `zMatch` deve mostrar o resultado final do match. Quando um match termina, as saídas (`jogosRestantes`, `placarA`, `placarB`, `zMatch` e `zRodada`) devem manter seus valores até que outro match se inicie, através da entrada `inicia`. O sistema deve ser síncrono, com as entradas (a menos de `reset`) fazendo efeito na borda de subida de `clock`. Cada rodada deve tomar no máximo 4 períodos de `clock`.

A entrada `inicia` não tem efeito se `nJogos` é igual a zero; e `carrega` não tem efeito se algum gesto é de espera.

Você pode usar o componente `jokempo` do Projeto 1.

`zRodada` e `zMatch` usam a codificação dos Projetos 1 e 2.

Por exemplo, se A lidera por 7 a 1, e faltam 5 rodadas, A já pode ser declarado vencedor.

Instruções para Entrega

Você deve acessar o link específico para a tarefa T4A1 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 àquelas 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 28 de julho 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. Como boa prática de engenharia, faça seus *test-benches* e utilize o GHDL ou o EDA Playground para validar suas soluções antes de postá-las no juiz.

Atenção: Está proibido o uso 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 o (zero).