

Departamento de Eletrónica, Telecomunicações e Informática da
Universidade de Aveiro



Laboratório de Sistemas Digitais

2014/2015 – 2º Semestre

Curso de Engenharia de Computadores e Telemática

Miniprojecto – FPGA

Autores:

Pedro Miguel André Coelho, Nº Mec. 68803

Tiago de Matos Ferreira Madeira, Nº Mec. 76321

Especificações do Sistema

- Clock
- 3 Divisores de Frequência
- Debouncer
- 5 Pseudo Random Generators
- 5 Registos (utilizaram 3bits)
- Somador
- Comparador (uma das entradas com o valor 21 – não presente na figura de arquitetura)
- Decoder BCD
- Máquina de estados (Mealy)
- 7 7BitDecoders (2 com EN)

Arquitetura do Sistema

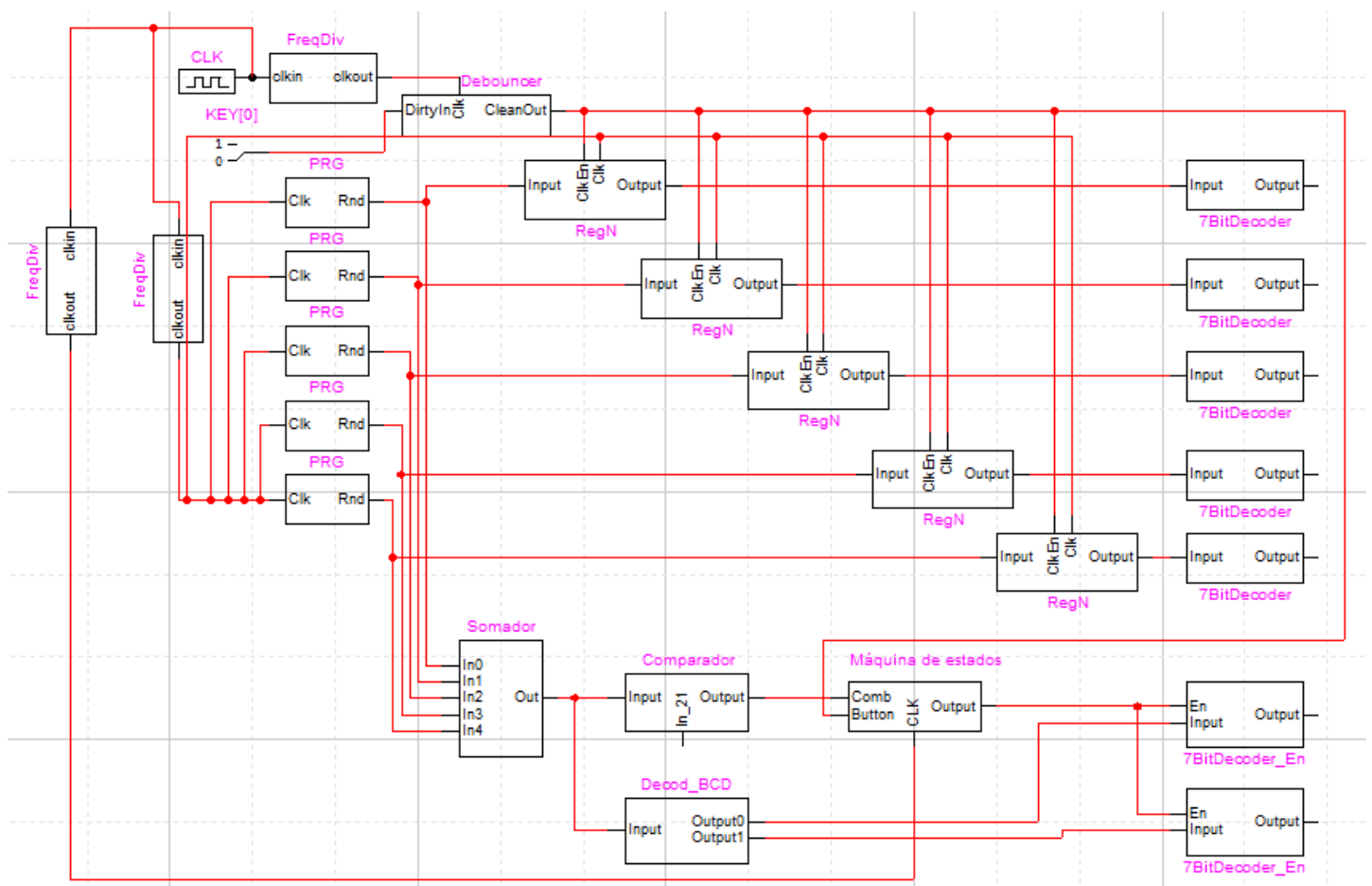


Figura 1.1 – Arquitetura do sistema

Abordagem Faseada de Desenvolvimento e Validação

O projeto será desenvolvido em duas fases principais. A primeira fase corresponde à criação do gerador de dados aleatórios e mostragem dos números. Será necessário um *clock* e um divisor de frequência ligados a um *debouncer*, este *debouncer* irá receber o sinal de um botão (KEY0), o sinal que sairá “limpo” do *debouncer* irá comandar o *clockEN* de 5 registos que estão a receber valores (correspondentes a cada dado) de 5 *pseudo random generators*, o objetivo é que enquanto a KEY[0] estiver premida os registos estejam ligados e mostrem a mudança de números pseudoaleatórios nos displays de 7 segmentos, passando por *7BitDecoders* e quando o botão é largado os *displays* congelam. Os *Pseudo Random Generators* também precisam de um *clock*, no qual será utilizado outro *FreqDividers*, este *clock* irá também funcionar como *input* do *clock* dos registos. A fase seguinte do projeto consiste em criar um circuito para somar os cinco números obtidos e mostra-los 1 segundo após a KEY[0] ter sido largada em dois *displays* adicionais, piscando caso a soma seja 21. Neste sentido é utilizado um somador e um comparador para comparar a soma ao número pretendido (21). Posteriormente, é utilizado um *decoderBCD* para representar o número obtido da soma em duas sequências de bits separadas, no sentido de mostrar cada algarismo num *display* diferente, passando cada um por um *7BitDecoder*. Existe uma máquina de estados que, dependendo das entradas KEY[0], ‘resultado da soma’ e do estado em que se encontra, mostra, ou não, o resultado, criando também o piscar caso o resultado da soma seja 21 (diagrama de estados - figura 1.2), esta tem um *clock* com um *FreqDivider* que põe o *clock* a 1Hz.

Em termos de verificações é necessário constatar se cada um dos comportamentos esperados se obtém, para a primeira fase, a geração e mostragem dos 5 números pseudo aleatórios e mostragem do “rolar dos dados” enquanto se mantém o botão premido; para a segunda fase, a soma correta dos números dos dados e apresentação do resultado 1 segundo após largar o botão, bem como o piscar do resultado caso a soma tenha resultado 21 e o *reset* do processo. Isto irá ser verificado através de uma *testbench* e posteriormente na FPGA.

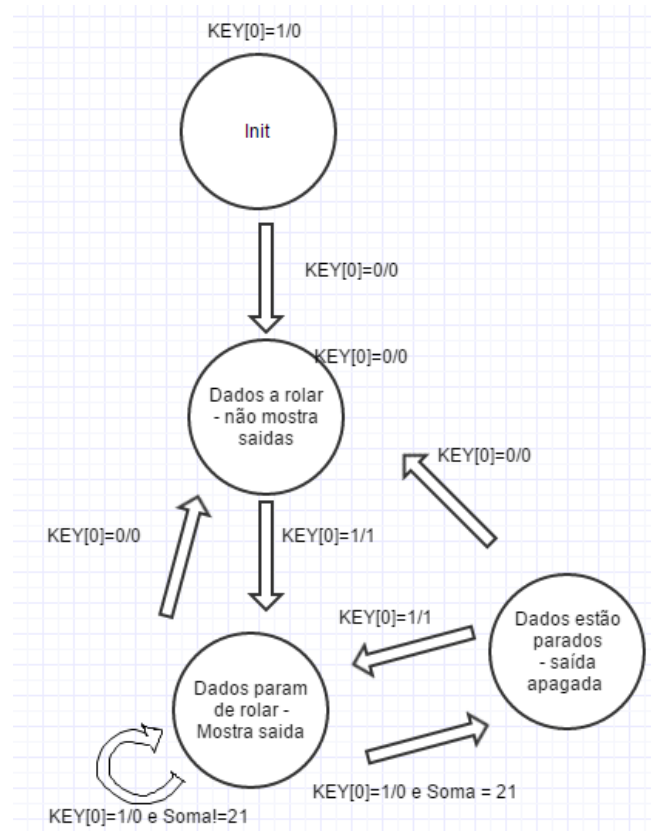


Figura 1.2 – Diagrama da máquina de estados

Divisão do Trabalho de Grupo

Tiago Madeira - Divisores de frequência

- Somador, Comparador, *DecoderBCD*
- Máquina de estados
- Verificação da fase 1 e implementação inicial da fase 2
- *TestBench*

Pedro Coelho - Divisores de frequência, *Debouncer*, *PRG*, *Registers* e *7BitDecoders*

- Verificação da implementação da fase 2
- Criação da *Shell*

Revisões finais e testes em conjunto.

Manual do Utilizador

Este projeto consiste num jogo de cinco dados independentes. Quando o utilizador pressiona o botão KEY[0] os valores dos dados valores são gerados aleatoriamente e apresentados em cinco *displays* de sete segmentos do *Kit DE2-115*, enquanto o botão estiver premido, os 5 *displays* mostram a mudança de valores, “o rolar dos dados” e os dois *displays* que irão apresentar o resultado ficam desligados. Assim que o utilizador larga o botão, os cinco valores são somados e o resultado é apresentado passado 1 segundo nos dois *displays* adicionais, estes irão piscar caso o resultado da soma seja 21. Se o utilizador pressionar novamente o mesmo botão, o processo repete-se.