

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

Relatório

Autores:

Pedro Miguel André Coelho, N.º Mec. 68803

Tiago de Matos Ferreira Madeira, N.º Mec. 76321

Introdução

Este relatório apresenta uma descrição do trabalho prático desenvolvido ao longo do semestre. O objetivo do trabalho foi o desenvolvimento de um programa que emulasse um jogo de dados utilizando programação VHDL em *FPGA's* (*kit DE2-115*). Serão descritas a arquitetura (descrição da estrutura conceptual do sistema), a implementação do projeto, com as respetivas representações máquinas de estado finitas e ligações a periféricos da *FPGA*, a validação (simulação e resultados finais), a conclusão do trabalho (discussão da forma como o trabalho realizado foi ao encontro dos objetivos definidos) e, finalmente, o manual do utilizador.

O projeto consiste na emulação de um jogo de cinco dados independentes, cujos valores são gerados pseudoaleatoriamente com o *clickar* de um botão. Os valores dos cinco dados e o valor da soma dos mesmos são mostrados em cinco mais dois *displays* de sete segmentos do *Kit DE2-115*, respetivamente. Quando a soma dos valores dos cinco dados é igual a “21”, os dois *displays* relativos à soma deverão piscar.

Arquitetura do Sistema

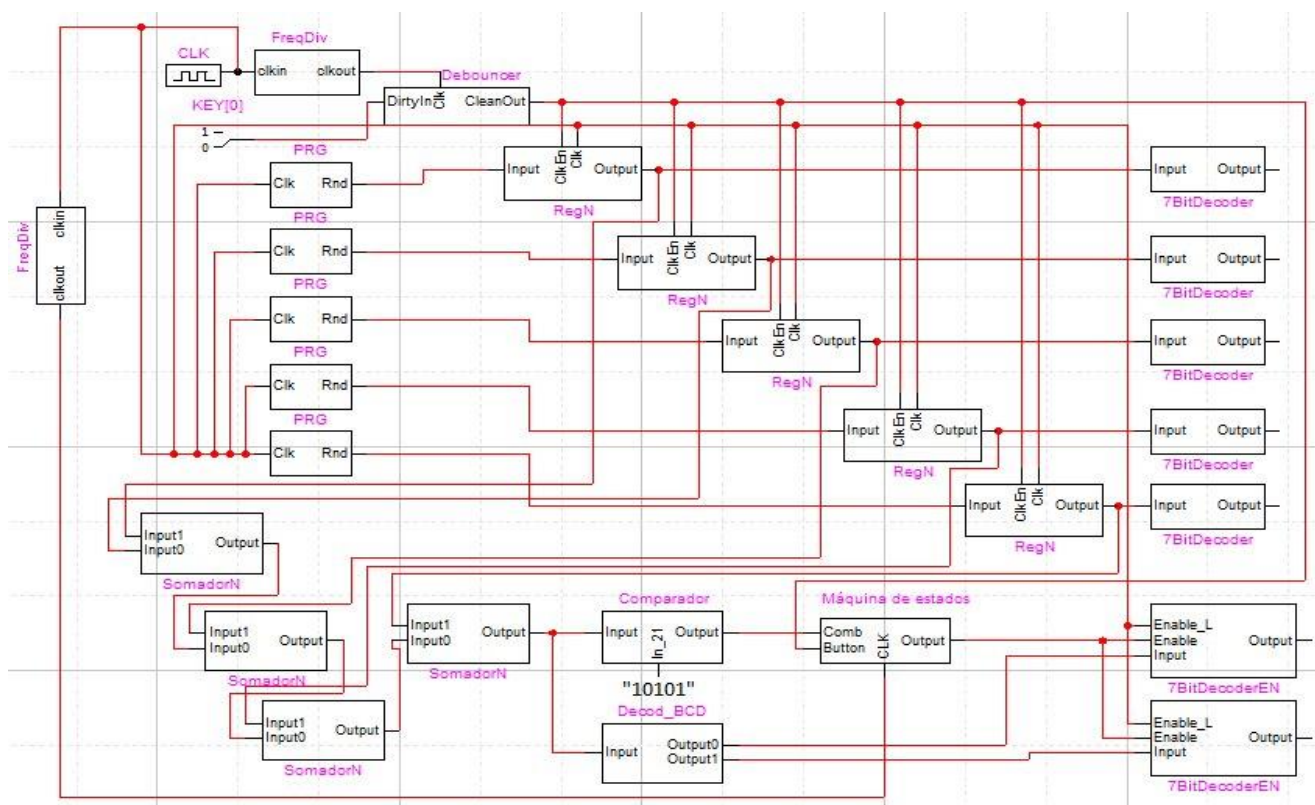


Figura 1.1 – Arquitetura do Sistema

Especificações do Sistema

- Clock
- 2 Divisores de Frequência
- Debouncer
- 5 Pseudo Random Generators
- 5 Registos
- 4 Somadores
- Comparador (uma das entradas com o valor 21)
- Decoder BCD
- Máquina de estados (Moore)
- 7 7BitDecoders (2 com 2 EN)

Implementação

O projeto foi desenvolvido em duas fases principais. Durante a primeira fase foram implementados cinco geradores pseudo aleatórios (*Pseudo Random Generators*) que funcionam sempre em segundo plano, gerando números a cada $\frac{1}{4}$ de segundo. Foram implementados também cinco registos e cinco *7BitDecoder* para a mostragem dos números, o botão KEY0 serve de *enable* para os registos, sendo que quando está pressionado os registos deixam passar os números aleatórios que estão a ser gerados e quando o botão é largado ficam guardados os valores. Para o sinal do botão ficar “limpo” foram utilizados um divisor de frequência ligado a um *debouncer*. Os *Pseudo Random Generators* fazem uso do CLOCK_50, tendo estes um mecanismo interno de contagem que atrasa a geração de um novo número $\frac{1}{4}$ de segundo.

A segunda fase do projeto consistiu em criar um circuito para somar os cinco números obtidos, fazendo uso de quatro somadores de N bits, e mostra-los em dois *displays*, os dois *7BitDecoders* correspondentes a esses *displays* também fazem uso do botão KEY[0] como *enable*, sendo que, enquanto o botão está premido (e os dados estão a ser gerados) não é mostrada a sua soma. Caso a soma dos dados seja igual a “21”, os dois *displays* deverão piscar, tal é comandado por uma máquina de estados (MEF), sendo feito uso de um comparador para comparar a soma obtida dos somadores ao número pretendido (“21”). É utilizado também um *decoderBCD* de modo a representar o número obtido da soma em duas sequências de bits separadas para mostrar cada algarismo decimal num *display* diferente, passando cada um por um *7BitDecoder*. A MEF é uma máquina de Moore, mudando o seu estado dependendo das entradas KEY[0] e ‘soma’, que provém do comparador. A saída da máquina está ligada ao *enable* dos *displays* que mostram o resultado da soma, 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 diminui a frequência do mesmo para 1Hz.

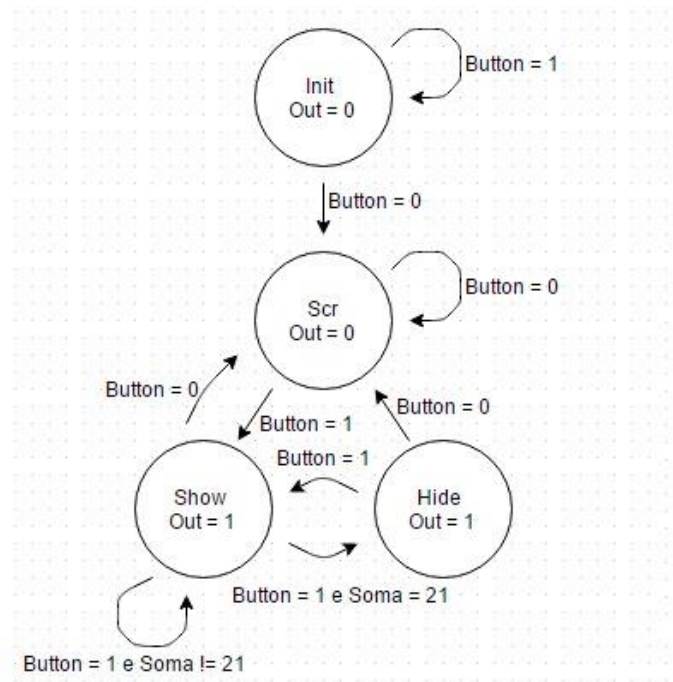


Figura 1.2 – Diagrama da máquina de estados

Ligação a periféricos do Kit DE2-115

- Clock ligado a PRG, Registos e Divisor de Frequência - CLOCK_50
- Enable dos registos e displays 6 e 7 - KEY[0]
- Valor do dado 1 em decimal - HEX0
- Valor do dado 2 em decimal - HEX1
- Valor do dado 3 em decimal - HEX2
- Valor do dado 4 em decimal - HEX3
- Valor do dado 5 em decimal - HEX4
- Valor da soma dos 5 dados - HEX6 (bit menos significativo)
- Valor da soma dos 5 dados - HEX7 (bit mais significativo)

Validação

Em termos de verificações foi constatado 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 após largar o botão, bem como o piscar do resultado caso a soma tenha resultado 21, bem como a repetição do processo, verificável a partir do correto funcionamento da máquina de estados (MEF), para a qual foi criada uma *testbench* (fig. 1.3). Testando no *Kit DE2-115*, concluímos que os resultados eram os esperados e a validação correu como pretendido.

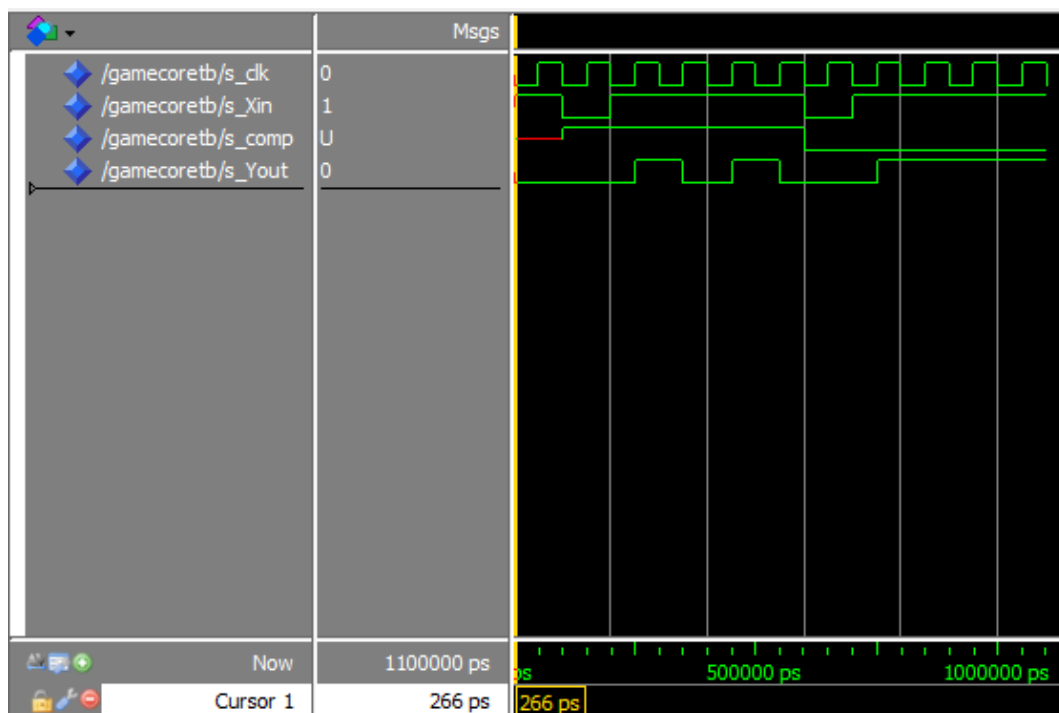


Figura 1.3 – Testbench da máquina de estados

Conclusão

Com o projeto concluído, consideramos que conseguimos alcançar os objetivos a que nos propusemos inicialmente. O projeto ficou funcional como pretendido, conseguimos obter números pseudoaleatórios, efetuar a soma deles, mostrar os números nos displays e que os displays da soma piscassem quando o resultado fosse 21, no fundo criando um jogo operacional, com as características pedidas no enunciado e com alguns retoques pessoais. Projetamos o jogo de maneira a, cumprir o pedido e ser mais divertido para o utilizador, mantendo uma visão dos números dos dados enquanto estes são gerados, aliando isto ao facto dos números serem gerados a cada $\frac{1}{4}$ de segundo, cria-se uma maior ilusão de controlo, apesar de ser virtualmente impossível o utilizador ler e somar 5 dígitos nesse tempo, torna-se uma experiencia mais emocionante do que se os dados fossem gerados números apenas com um *click*.

Manual do Utilizador

Este projeto consiste num jogo de cinco dados independentes. Inicialmente cinco *displays* correspondentes aos dados (cinco *displays* mais à direita) apresentam o valor “0”, representando cinco dados sem valor lançado ainda. Para “lançar” os dados o utilizador deve pressionar o botão KEY[0], enquanto este estiver premido os valores continuaram a ser gerados e mostrados, assim que o botão deixar de ser premido os valores irão “congelar” e irá aparecer a soma dos mesmos nos dois *displays* mais à esquerda, piscando se tiver ganho, ou seja, se a soma dos dados que “lançou” for igual a “21”. Para tentar novamente o utilizador deve apenas repetir o processo, pressionando novamente o mesmo botão.