

Teste Reação

Universidade de Aveiro

Rafael Marques, Tiago Pita



VERSAO 1

Teste Reação

Dept. de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro

Rafael Marques, Tiago Pita
(119927) rafaelfmarques@ua.pt, (120152) tiagomsp18@ua.pt

3 de junho de 2024

Conteúdo

1	Introdução	1
2	Manual de Utilização	2
2.1	Funcionamento	2
2.2	Esquema da Máquina	2
3	Arquitetura e Implementação	4
3.1	Debouncers	4
3.2	ReactionTimeFSM	4
3.3	RandomTimeGen	5
3.4	LightAndReactionTime	5
4	Validações	6
5	Conclusões e Contribuições	7
5.1	Conclusões	7
5.2	Contribuições dos autores	7

Lista de Tabelas

Lista de Figuras

2.1	Ilustração do Esquema da Máquina	2
3.1	Ilustração do Top-Level da Máquina	4
3.2	Esquema da Máquina de Estados	5

Capítulo 1

Introdução

Como projeto final da cadeira de Laboratório de Sistemas Digitais (LSD), foi-nos proposto o desenvolvimento de um teste de reação. Este projeto foi implementado utilizando VHSIC Hardware Description Language (VHDL) para simular o seu comportamento na placa FPGA Terasic DE2-115. O objetivo é medir o tempo de reação do utilizador a um estímulo luminoso.

O projeto contém:

1. Fase Configuração

Secção de Display: Uma secção que exhibe a palavra Config.

Secção de Display: Uma secção onde é possível ver o incremento ou diminuição da pontuação alvo.

Botões de Controle:

- **Botão Confirm:** Utilizado para confirmar a pontuação.
- **Botão de Aumentar:** Utilizado para aumentar a pontuação.
- **Botão de Diminuir:** Utilizado para diminuir a pontuação alvo.

2. Fase jogo

Secção de Display: Uma secção que exhibe o tempo de reação do utilizador ao estímulo luminoso.

Secção de Display: Duas secções que exibem os pontos de cada jogador.

Botões de Controle:

- **Botão Start:** Utilizado para iniciar o teste de reação.
- **Botão de Reset:** Utilizado para reiniciar o teste.
- **Botões Reação:** Utilizado por cada jogador.

3. Fase Conclusão

Secção display: Uma secção onde mostra o vencedor.

A implementação foi realizada em VHDL, e o comportamento foi simulado na placa FPGA Terasic DE2-115. O código foi desenvolvido para gerir a ativação das luzes, a contagem do tempo de reação, e a exibição do tempo medido no display. Além disso, a lógica inclui estados para aguardar a ativação do teste, iniciar a contagem do tempo de reação, e reiniciar o sistema quando necessário.

Capítulo 2

Manual de Utilização

2.1 Funcionamento

A máquina inicia pedindo ao utilizador que ajuste a pontuação alvo.

Quando o utilizador pressionar a tecla de confirmação, começa um jogo entre dois jogadores, para ver qual tem o menor tempo de reação perante um estímulo visual. O estímulo visual aparece de forma aleatório, devido a geração de um tempo aleatório, para que não exista fraude de resultados.

Caso um jogador tenha uma reação prematura, e reaja antes do estímulo visual, será penalizado.

2.2 Esquema da Máquina

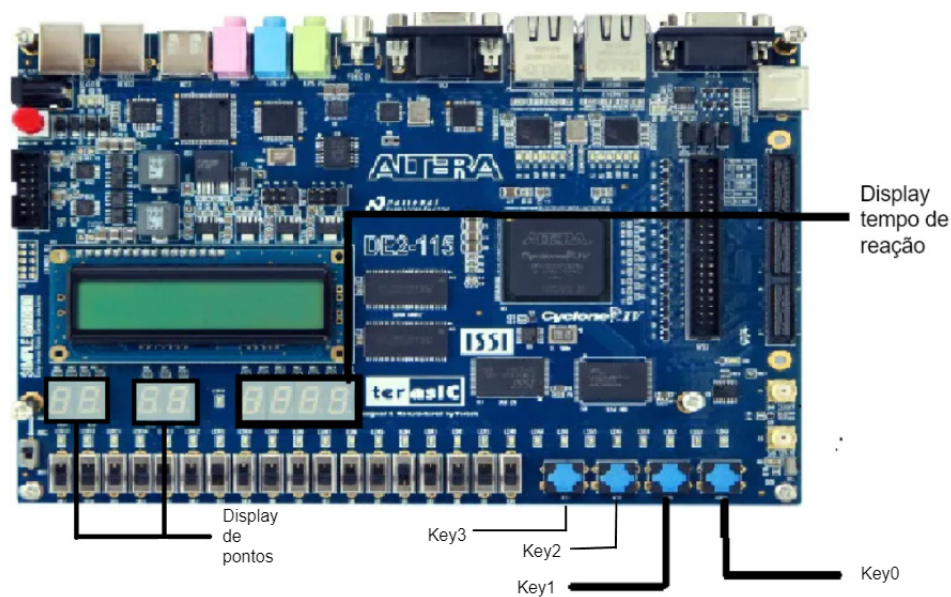


Figura 2.1: Ilustração do Esquema da Máquina

1. Key1: botão confirmar
2. **Início**
 - ↪ Key0: botão confirmar
 - ↪ Key2: Diminuir o máximo de pontos
 - ↪ Key3: Aumentar o máximo de pontos
3. **Durante o jogo**
 - ↪ Key0: Start
 - ↪ Key2: Player A
 - ↪ Key3: Player B

4. Fim do jogo

↪ Key0: Recomeçar o jogo

Capítulo 3

Arquitetura e Implementação

O Top-Level da máquina é composto por 3 componentes principais que depois se ramificam em subcomponentes mais pequenos.

A figura 3.1 representa uma ilustração gráfica do Top-Level da máquina implementado em VHDL.

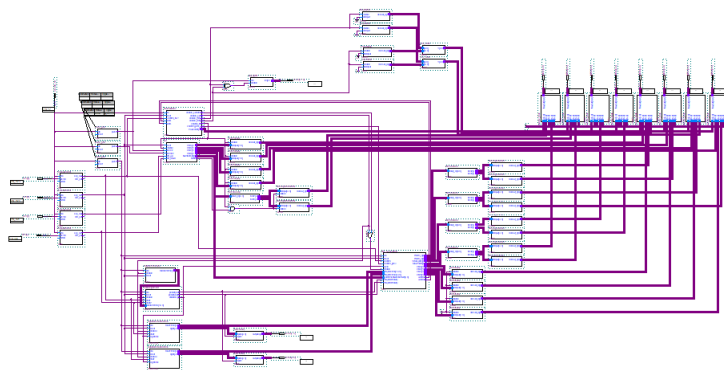


Figura 3.1: Ilustração do Top-Level da Máquina

3.1 Debouncers

Este componente é responsável por fazer o Debounce dos botões. Isto é necessário pois quando um botão é pressionado gera centenas de sinais, o que pode muitas vezes causar problemas.

O bloco recebe o valor do relógio geral da máquina bem como os valores dos botões a ser afetados.

Sai deste bloco os sinais dos botões já corrigidos para que seja emitido apenas 1 sinal positivo por clique.

3.2 ReactionTimeFSM

Este é o componente principal da máquina e é o responsável pelo processamento do funcionamento da mesma.

Entram neste componente o sinal do relógio da máquina, os sinais dos botões de 'Reiniciar', 'Start/Stop', os sinais dos botões correspondentes a cada jogador.

O funcionamento da máquina de estados será descrito com recurso à figura 3.2.

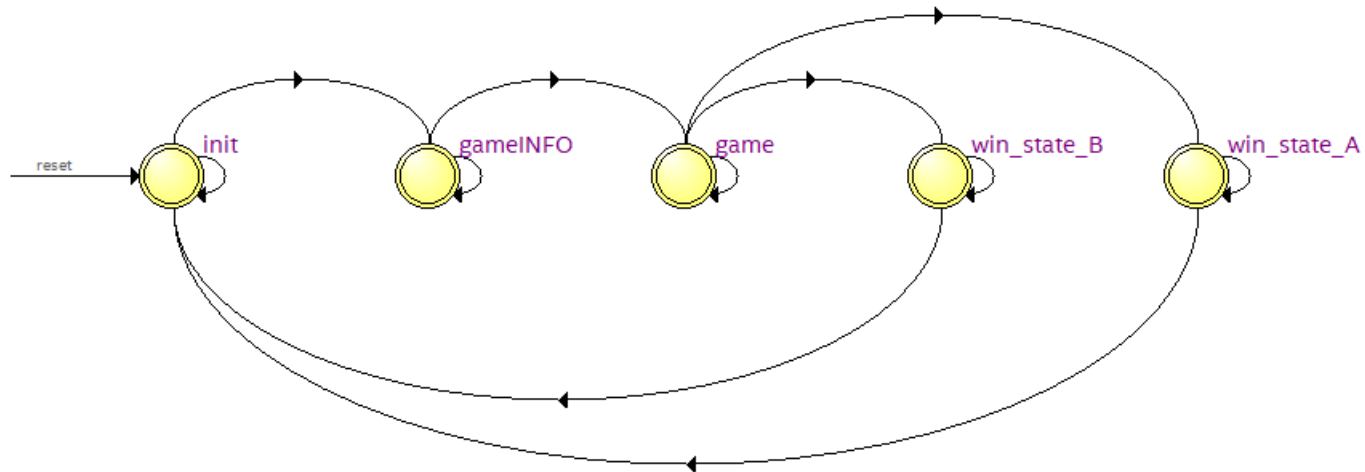


Figura 3.2: Esquema da Máquina de Estados

1. **Reset**
 \hookrightarrow A máquina volta sempre ao estado de Reset quando o botão ‘Reset’ é pressionado.
2. **(Init) \rightarrow gameINFO**
 \hookrightarrow Após o utilizador ajustar a pontuação e clicar no botão de confirmação, a máquina passa do estado init para gameINFO.
3. **gameINFO \rightarrow game**
 \hookrightarrow Após o utilizador pressionar no botão Start a máquina passa para o estado game.
4. **game \rightarrow win_state_B/win_state_A**
 \hookrightarrow Dependendo do utilizador que atinja primeiro a pontuação que foi definida no início, a máquina de estados transita para o estado win_state_B/win_state_A.
5. **Standby \rightarrow Delay**
 \hookrightarrow Quando o botão de ‘Start/Stop’ é pressionado, muda de estado para ‘Delay’ começando o timer do atraso inicial com o valor escolhido.
6. **win_state_B/win_state_A \rightarrow init**
 \hookrightarrow Se o utilizador pretender recomeçar o jogo basta pressionar a tecla.

3.3 RandomTimeGen

Este componente é uma implementação simples que ilustra o uso de um contador síncrono para gerar valores de tempo dentro de um intervalo especificado.

Embora o valor gerado não seja verdadeiramente aleatório, a variação no tempo é controlada pelo sinal de clock e pelo reset, permitindo uma sequência pseudo-aleatória de valores dentro do intervalo definido

3.4 LightAndReactionTime

Este implementa uma máquina de estados finitos para controlar uma matriz de luzes e medir o tempo de reação.

A MEF possui três estados principais que gerem a ativação das luzes, aguardam pelo início da medição do tempo de reação, e contam o tempo de reação até que o estímulo (start) seja detectado.

O tempo de reação é posteriormente convertido de binário para BCD para poder ser exibido no display de sete segmentos.

Capítulo 4

Validações

No decorrer do nosso projeto fomos confrontados com diversas adversidades no que toca a simulação e validação. Como tal, a principal maneira de verificação foi prática e feita com a placa, já que o trabalho funciona maioritariamente em segundos, um tempo difícil de se trabalhar tanto no simulador, como na testbench.

Capítulo 5

Conclusões e Contribuições

5.1 Conclusões

Após uma breve reflexão observámos que com este trabalho foram desenvolvidas novas capacidades em VHDL, lógica de estruturação (aplicada durante o planeamento das funções), otimização (de forma a simplificar o trabalho da melhor forma possível) e ainda capacidades a nível de trabalho em grupo. Vimos também algumas das capacidades da placa e o potencial da disciplina.

Autoavaliámos o nosso trabalho com 16 valores.

5.2 Contribuições dos autores

Neste projeto, uma vez que houve um maior empenho pela parte do elemento Tiago Pita , este mesmo elemento tem uma percentagem de 70% enquanto o elemento Rafael Marques , uma percentagem de 30%.

Acrónimos

LSD Laboratório de Sistemas Digitais

VHDL VHSIC Hardware Description Language