

Máquina de Pão

Universidade de Aveiro

Tiago Garcia, José Fernandes



VERSAO 1

Máquina de Pão

Dept. de Eletrónica, Telecomunicações e Informática

Universidade de Aveiro

Tiago Garcia, José Fernandes

(114184) tiago.garcia@ua.pt, (114472) jbfernandes@ua.pt

30 de maio de 2023

Conteúdo

1	Introdução	1
2	Manual de Utilização	2
2.1	Programas	2
2.2	Funcionamento	2
2.3	Esquema da Máquina	3
3	Arquitetura e Implementação	4
3.1	Debounceers	4
3.2	BreadMachineFSM	4
3.3	DisplaysManager	6
4	Validações	7
5	Conclusões e Contribuições	8
5.1	Conclusões	8
5.2	Contribuições dos autores	8

Lista de Tabelas

2.1	Programas de Pão	2
-----	----------------------------	---

Lista de Figuras

2.1	Ilustração do Esquema da Máquina	3
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 projeto da máquina de pão automática.

Para tal, usamos VHSIC Hardware Description Language (VHDL) para simular o seu comportamento na placa FPGA Terasic DE2-115. Haverão 3 secções de displays, uma para o tempo do programa, outra para o tempo extra e outra para o atraso até ao início da execução do programa. Haverá também um interruptor para seleccionar o programa a ser executado, 7 interruptores para escolher o atraso inicial (introduzido em binário) e ainda 3 botões, o 'Start/Stop', outro para reiniciar a máquina e o último para adicionar o possível tempo extra.

Capítulo 2

Manual de Utilização

2.1 Programas

Pão Caseiro		Pão Rústico	
Fase da Fabricação	Tempo (segundos)	Fase da Fabricação	Tempo (segundos)
Amassar	10	Amassar	15
Levedar	4	Levedar	8
Cozer	10	Cozer	10

Tabela 2.1: Programas de Pão

2.2 Funcionamento

A máquina inicializa permitindo o utilizador seleccionar o programa a ser executado. Para tal, deve colocar o interruptor ‘Selecionador Programa’ na posição do programa que deseja. O ‘Pão Caseiro’ será a posição inicial do interruptor e o ‘Pão Rústico’ será a outra posição. Sempre que a máquina é reiniciada estas posições são reatribuídas. Os programas, bem como as suas durações, são descritos em cima: Tabela 2.1.

O utilizador pode também escolher o atraso inicial a ser aplicado e, para isso, terá de usar os 7 interruptores do ‘Selecionador Atraso’ para indicar o número, este terá de ser introduzido em binário e a máquina irá converter para decimal. Este valor poderá variar entre 0 e 90 segundos e qualquer valor superior a isso será interpretado como 90.

Após a seleção do programa e do atraso inicial o utilizador poderá clicar no botão ‘Start/Stop’ para iniciar a máquina. O programa irá começar assim que o atraso inicial chegar a 0. Poderá ver a fase em que o programa se encontra com os ‘LEDS Display Fase’:

- 3 LEDS ligados → Amassar
- 2 LEDS ligados → Levedar
- 1 LED ligado → Cozer ou Tempo Extra
- 0 LEDS ligados → Standby, Atraso Inicial ou Espera da Confirmação do Tempo Extra/Reinicialização

O tempo, tanto do atraso como da programação normal poderá ser pausado e continuado em qualquer momento ao longo da sua execução.

Após o final da programação normal o utilizador terá a possibilidade de adicionar tempo extra utilizando o botão ‘Tempo Extra’. Para iniciar o mesmo, o utilizador deverá clicar no botão ‘Start/Stop’. Tal como o atraso inicial e o tempo do programa, o tempo extra também poderá ser pausado e continuado em qualquer momento. Se o botão de ‘Start/Stop’ for pressionado com o tempo extra a 0 então a máquina irá ser reiniciada. O utilizador pode usar o tempo extra o número de vezes que desejar.

O ‘LED Display Execução’ estará ligado durante a execução do programa normal e do tempo extra.

A máquina poderá também ser reiniciada em qualquer momento usando o botão ‘Reset’.

2.3 Esquema da Máquina

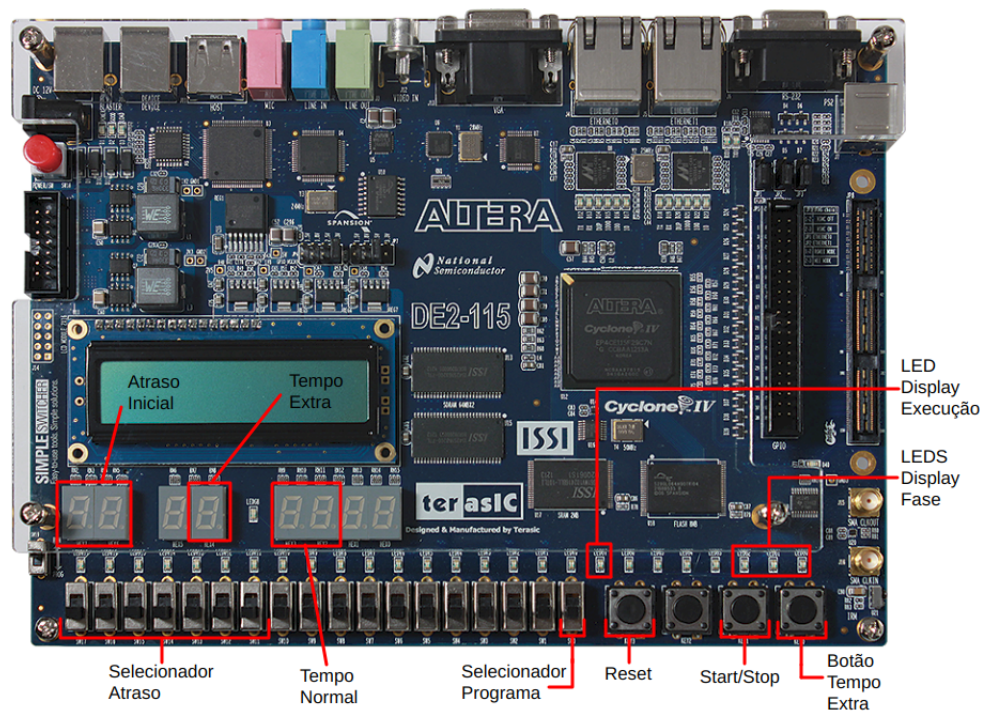


Figura 2.1: Ilustração do Esquema da Máquina

- **Display: Atraso Inicial** → Display para mostrar o tempo que irá decorrer antes do início da execução do programa.
- **Display: Tempo Extra** → Display para mostrar o tempo extra do programa.
- **Display: Tempo Programa** → Display para mostrar o tempo de execução do programa.
- **LED: Display Execução** → LED para indicar que o programa está a ser executado.
- **LEDS: Display Fases** → LEDS para indicar a fase do programa que está a ser executada.
- **Switch: Selecionador Atraso** → Série de interruptores para selecionar o tempo de atraso inicial.
- **Switch: Selecionador Programa** → Interruptor para selecionar o programa a ser executado.
- **Botão: Reset** → Botão para reiniciar a máquina.
- **Botão: 'Start/Stop'** → Botão para iniciar ou parar a execução do programa.
- **Botão: Tempo Extra** → Botão para adicionar tempo extra ao programa.

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.

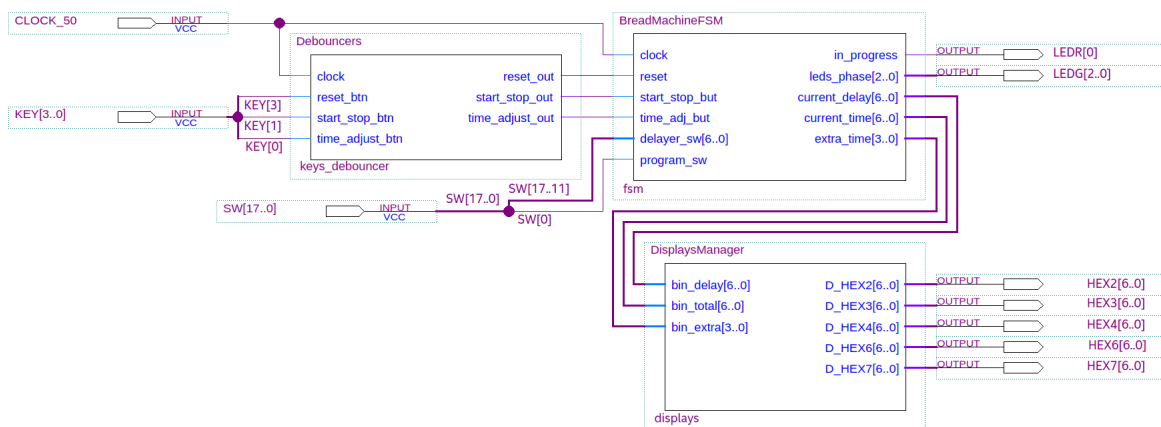


Figura 3.1: Ilustração do Top-Level da Máquina

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.

Dentro do bloco os valores dos botões são distribuídos entre 3 debouncers onde são processados para gerar os sinais pretendidos.

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

3.2 BreadMachineFSM

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' e 'Tempo Extra', recebe também o sinal do interruptor do 'Selecionador Programa' e, por último, o valor do vetor gerado pelos interruptores do 'Selecionador Atraso'.

Como saídas irá ter o sinal para indicar se a máquina se encontra no estado 'Progress' ou 'Extra'. Faz também parte das saídas o vetor de indicação da 'Fase da Fabricação' que está a decorrer no momento. Estas saídas são diretamente ligadas aos LEDs da máquina: 1 LED vermelho e 3 LEDs verdes, respetivamente.

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

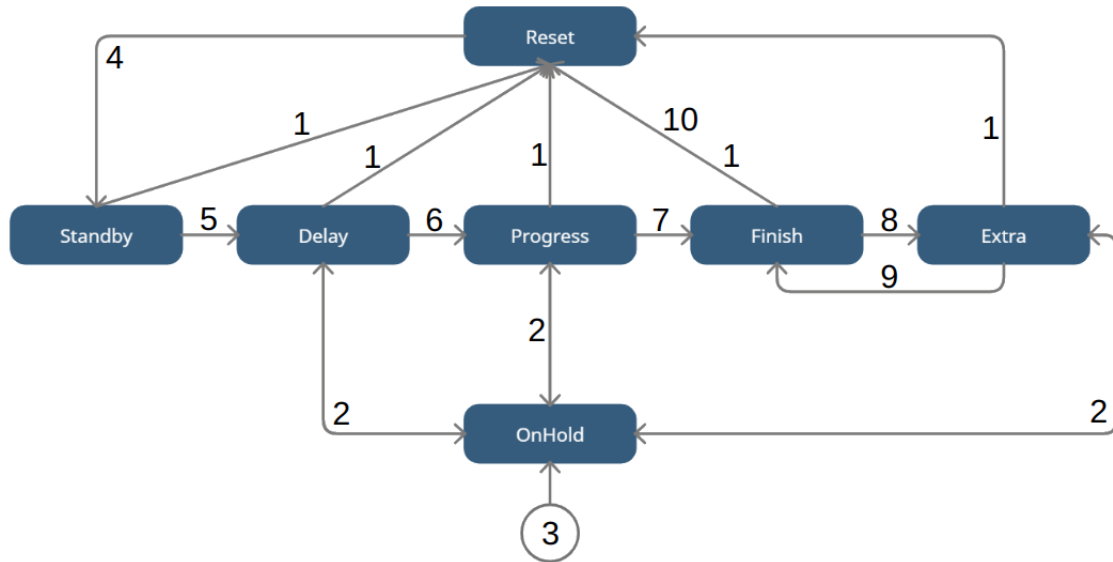


Figura 3.2: Esquema da Máquina de Estados

1. **Reset**

↔ A máquina volta sempre ao estado de Reset quando o botão ‘Reset’ é pressionado.

2. **(Delay/Progress/Extra) ↔ OnHold**

↔ Quando o botão de ‘Start/Stop’ é utilizado e o estado é ‘Delay’, ‘Progress’ ou ‘Extra’, muda o estado para ‘OnHold’ para pausar a máquina.

↔ Quando o botão de ‘Start/Stop’ é utilizado e o estado atual é ‘OnHold’, retoma o estado para o qual a máquina se previamente encontrava (‘Delay’, ‘Progress’ ou ‘Extra’).

3. **OnHold → OnHold**

↔ Atualiza o valor do start_stop negando o mesmo. Fica neste estado enquanto o valor do start_stop for ‘0’ e sai do estado quando passar para ‘1’.

4. **Reset → Standby**

↔ Assim que a máquina é reiniciada, o estado muda automaticamente para Standby após a reinicialização de todos os valores.

5. **Standby → Delay**

↔ Quando o botão de ‘Start/Stop’ é pressionado, muda de estado para ‘Delay’ começando o timer do atraso inicial com o valor escolhido.

6. **Delay → Progress**

↔ Assim que o tempo do atraso inicial chegar a 0, o estado passa automaticamente de ‘Delay’ para ‘Progress’.

7. **Progress → Finish**

↔ Assim que o tempo da programação chegar a 0, o estado passa automaticamente de ‘Progress’ para ‘Finish’.

↔ Neste estado, a máquina não se encontra com nenhuma mudança visual imediata, fica a aguardar por um dos botões. Usando o botão ‘Tempo Extra’ poderá ser definido o tempo extra a aplicar à máquina posteriormente no estado ‘Extra’.

8. **Finish → Extra**

↔ Esta mudança é realizada quando o botão ‘Start/Stop’ é pressionado e o tempo extra não se encontra a 0.

9. **Extra → Finish**

↔ Assim que o tempo do tempo extra chegar a 0, o estado passa automaticamente de ‘Extra’ para ‘Finish’.

10. **Finish → Reset**

↔ Esta mudança é realizada quando o botão ‘Start/Stop’ é pressionado e o tempo extra encontra-se a 0. Este passo reinicializa a máquina automaticamente.

Nota: Os passos 2, 3, 8 e 9 podem ocorrer um número de vezes indefinido durante uma execução completa da máquina.

3.3 DisplaysManager

Este último componente é o responsável pela gestão dos displays para que os números dos tempos da máquina sejam corretamente visualizados.

O bloco recebe separadamente os valores dos 3 tempos: 'Atraso Inicial', 'Tempo Normal' e 'Tempo Extra'.

O tempo extra é diretamente processado para a codificação dos displays, enquanto que o atraso inicial e o tempo normal têm de ser convertidos em numeração decimal (gerando 2 valores cada) para depois serem codificados. Este passo produz 5 valores diferentes.

Esses 5 valores são então distribuídos pelos 5 displays da máquina.

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, o que nos despertou interesse em saber mais e talvez desenvolver algo por conta própria. No geral, foram cumpridos todos os objetivos requeridos e ainda foram implementadas novas funcionalidades.

Autoavaliámos o nosso trabalho com 18 valores.

5.2 Contribuições dos autores

Neste projeto, ambos os elementos do grupo trabalharam de igual forma e com semelhante nível de empenho e por isso cada elemento do grupo tem uma percentagem de participação de 50%.

Acrónimos

LSD Laboratório de Sistemas Digitais

VHDL VHSIC Hardware Description Language