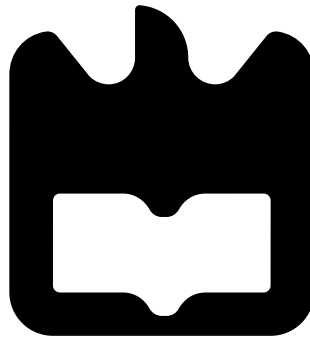




Gonçalo Cunha
108352
Anderson Lourenço
108579

Laboratórios de Sistemas Digitais
Projeto Final
Máquina de Lavar
Versão 3



Conteúdo

Conteúdo	1
1 Introdução	2
2 Arquitetura	3
3 Manual do Utilizador	4
4 Implementação	5
5 Conclusão	7

Capítulo 1

Introdução

Este projeto tem como objetivo a modelação em VHDL e a implementação na *FPGA* um sistema digital que simule uma máquina de lavagem de roupa com um botão de start/stop e outro de reset, tendo esta três programas possíveis que são compostos de 4 tarefas diferentes. Tem ainda uma porta que interrompe o funcionamento quando aberta e um modo diferido que conclui o programa selecionado ao fim de 60 segundos.

A máquina permite também observar ao longo do seu funcionamento se esta se encontra a meio de um programa, em que tarefa se encontra, e quanto tempo falta para esta acabar.

Capítulo 2

Arquitetura

O sistema é composto por vários blocos unidos através de uma *top-level entity*, sendo os blocos usados:

- Um Register Unit que sincroniza os inputs dos interruptores com o **CLOCK_50** e no qual é instanciado outro bloco, um debouncer, que "limpa" o sinal dos botões.
- Um Display Unit, que converte os valores recebidos de binário para BCD, e de BCD para sete segmentos, de forma a poderem ser mostrados nos displays da FPGA.
- Um Timer que recebe os valores do tempo de cada tarefa/programa e conta o tempo para este acabar, enviando o tempo atual para o display, e também envia um sinal, **timeExp** para a **FSM** que o tempo numa tarefa terminou.
- Um Pulse Generator que envia um pulso de 1Hz para o time para este so contar o tempo em segundos,
- Uma **FSM** cujo funcionamento se encontra elaborado no Capítulo 4

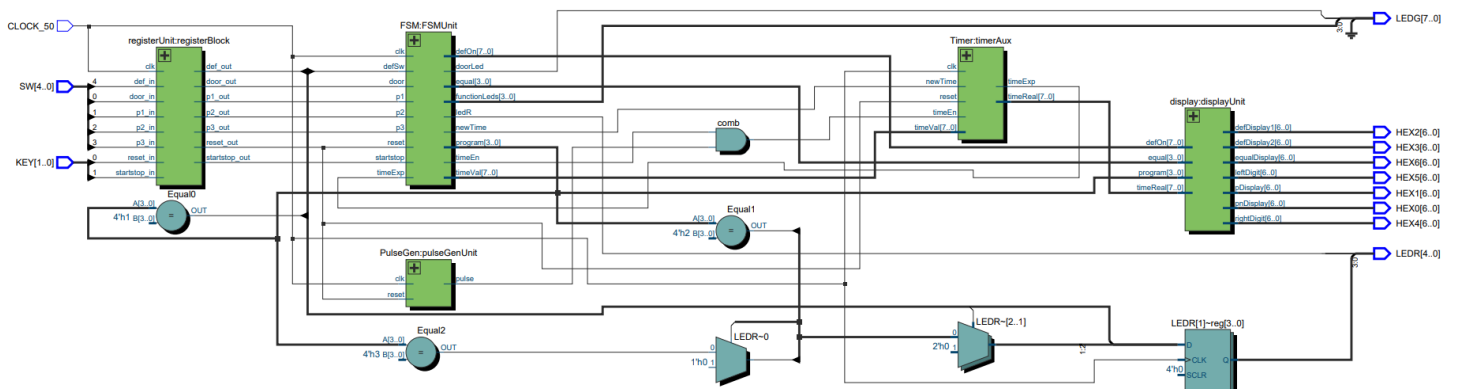


Figura 2.1: Diagrama de Blocos da Máquina de Lavar

Capítulo 3

Manual do Utilizador

Selecione o programa pretendido utilizando os interruptores **SW1**, **SW2** ou **SW3**.

SW1 Lavagem completa: Tarefas - meter água, enxaguar, remoção da água, meter água, enxaguar, remoção da água, spin e remoção da água.

SW2 Pré-lavagem: Tarefas - meter água, enxaguar, remoção da água, spin e remoção da água.

SW3 Extra-spin: Tarefas - spin e remoção da água.

O botão **KEY1** para começa/pausa o programa, dependendo se este está parado ou não.

O botão **KEY0** reinicia a máquina.

O interruptor **SW4** ativa o modo diferido,

O interruptor **SW0** funciona como porta da máquina.

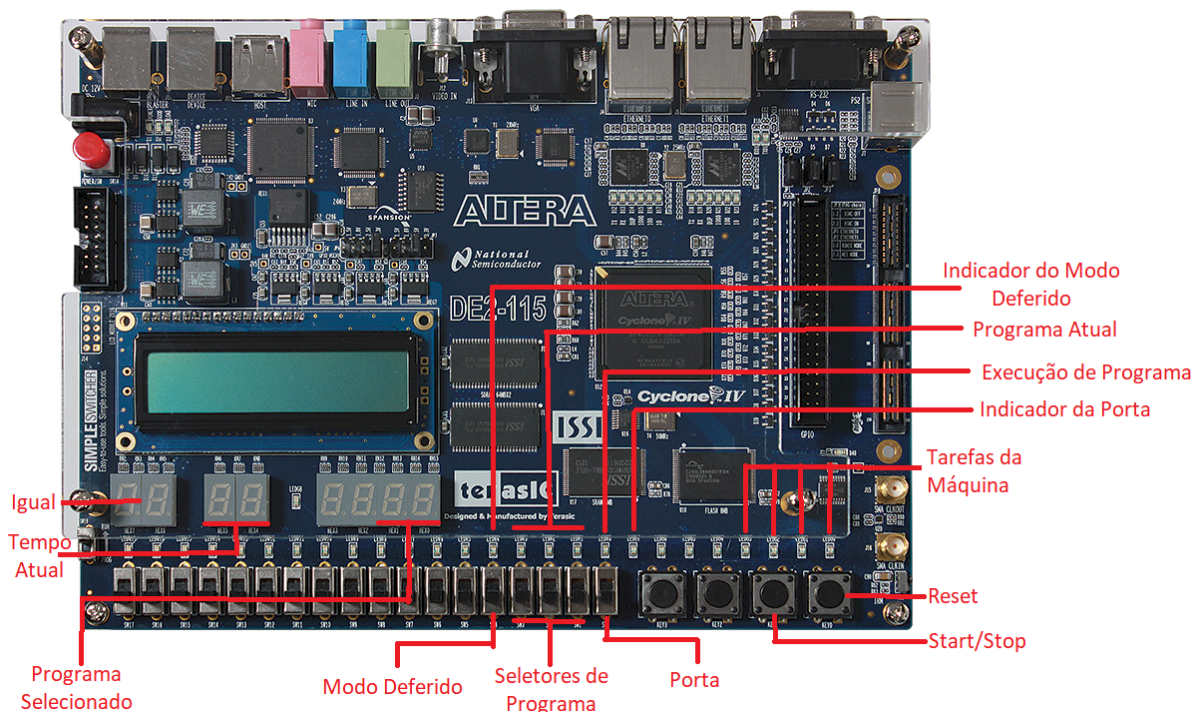


Figura 3.1: Controlos na *DE2-115 FPGA*

Capítulo 4

Implementação

O projeto funciona com base numa máquina de estados com sete estados: **idle**, o estado inicial, um estado para cada tarefa, **wIn** para meter água, **rns** para enxaguar, **wOut** para remover água, e **spn** para spin, o estado **tOff** para quando a máquina termina um programa, e o **def** que serve para implementar o modo diferido.

No estado **idle** só ao escolher um programa, mantendo a porta fechada e premindo o botão de start/stop é que se transiciona para um estado diferente, sendo este o **wIn** se o programa 1 ou 2 forem selecionados, o **spn** se o programa 3 for escolhido, e o **def** se o modo diferido estiver ativo, caso contrário, o estado seguinte será o **idle**.

Tanto o estado **wIn** e o estado **rns**, funcionam semelhantemente, se o **timeExp** recebido do Timer estiver ativo, passam ao estado **rns** e **wOut**, respetivamente, senão mantêm-se no mesmo estado.

No estado **wOut**, se o *timeExp* estiver ativo o próximo estado depende no valor dos sinais *cycle* e *programEnd*, sendo que o primeiro indica à máquina que tem de voltar ao estado **wIn**, e o último indica à máquina que o próximo estado é o **tOff**, tendo o programa terminado, se nenhum destes estiver ativo a máquina transiciona para o estado **spn**. Se o *timeExp* for inativo, o próximo estado será o mesmo.

No estado **spn** é ativado o sinal *programEnd*, seguindo para o estado **wOut** se o *timeExp* estiver ativo, e mantendo-se no mesmo estado caso contrário.

O estado **tOff** serve apenas para aguardar 2 segundos no fim de terminar o programa, sendo que passados este tempo recebe o sinal *timeExp* do Timer e passa ao estado **idle**.

O estado **def** serve para o programa selecionado terminar ao fim de 1 minuto, sendo que este estado se mantém por 60 segundos menos o tempo do programa selecionado. Quando este tempo acaba, semelhantemente ao **idle**, o próximo estado depende do programa selecionado, sendo **wIn** se o programa for 1 ou 2, e **spn** se for 3.

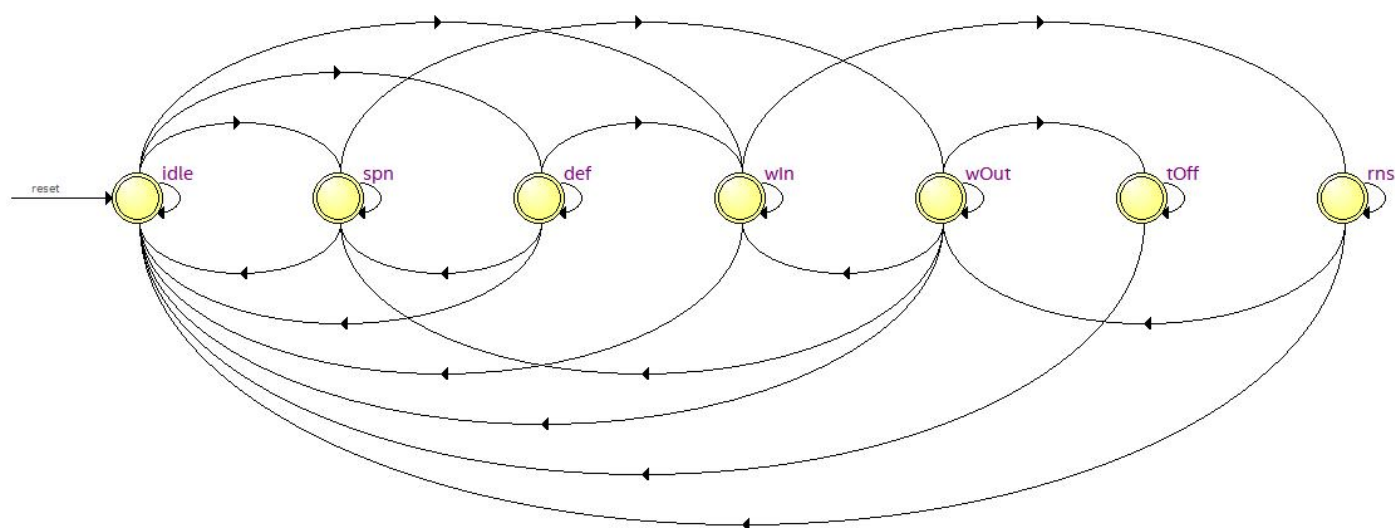


Figura 4.1: *Finite State Machine*

Capítulo 5

Conclusão

Este projeto foi aquele que nos chamou mais à atenção, sendo que nos pareceu o mais interessante, embora tenhamos encontrado algumas dificuldades na implementação de certas funções, como o modo diferido e também na criação de uma testbench para testar a máquina de estados, que não conseguimos criar, tendo todos os testes sido feitos na própria placa *FPGA*.

Contribuição

Gonçalo Cunha : 95%

Anderson Lourenço : 5%