



universidade de aveiro

Departamento de eletrónica, telecomunicações e informática

Curso 8204 - Mestrado Integrado e m Engenharia Eletrónica e Telecomunicações  
Disciplina 40333- Laboratório de Sistemas Digitais  
Ano letivo 2015/16

## Relatório Final

### *Fechadura Eletrónica*

Autores:

79970 Rui Filipe Santos Carapinha (Contribuição para o trabalho - 60%)  
79926 Pedro Francisco dos Santos Figueiredo (Contribuição para o trabalho – 40%)  
Turma P20 Grupo 1

Data 24/04/2016  
Docente José Luis Azevedo

Resumo: Este projeto teve como objetivo principal a realização em VHDL de uma fechadura eletrónica para ser possível podermos utilizar numa FPGA.

Neste relatório, explicamos pormenorizadamente as entidades utilizadas, como funcionam, para que servem e como é que se interligam umas com as outras. O resultado apesar de não ser o esperado teoricamente, foi satisfatório.

## Índice

---

- Introdução
- Arquitetura
- Implementação
- Conclusão

## Introdução

---

O objetivo deste projeto é o de programar a FPGA para que seja possível simular o funcionamento de uma fechadura eletrónica. O grupo responsável pela realização deste trabalho prático definiu também como um dos seus objetivos a aprendizagem mais aprofundada de conceitos outrora abordados.

No projeto em si, o utilizador deve introduzir uma palavra passe definida pelo supervisor, caso contrário, por cada tentativa errada aumenta o tempo até o sistema aceitar a introdução de uma nova palavra passe. Após, 10 tentativas erradas por parte do utilizador o sistema bloqueia até à reposição com um código de supervisão.

Para auxiliar o utilizador, a FPGA com recurso aos displays hexadecimais deverá mostrar o número de tentativas, a palavra passe introduzida e, caso o utilizador acerte na palavra passe, este deverá mostrar a palavra “GO”.

## Arquitetura

---

Neste ponto do relatório iremos explicar a função de cada entidade descrita no diagrama disponível em anexo.

O sistema tem como entradas 3 *switches* da FPGA, um dedicado a um *reset*, um dedicado ao fecho da fechadura e, por último, um dedicado à mudança da palavra passe do supervisor. O sistema também possui um relógio (*clock*) e as 4 *keys* para a introdução da palavra passe. Ligado a cada *key* está ligado um *debouncer*, de modo a originar um sinal com o comprimento de apenas 1 sinal de relógio.

A saída dos *debouncers* liga às entradas (*Enter*, *ToRight*, *Increase* e *Decrease*) da entidade *ReadCode* que vai receber esses sinais e vai atualizando a palavra passe do utilizador até ser pressionado a *key* destinada ao *Enter*. Esta entidade possui uma saída para cada dígito (*FirstDigit*, *EFirst*, *SecondDigit*, *ESecond*, *ThirdDigit*, *EThird*, *FourthDigit* e *EFourth*) que estão ligados, cada um deles, a uma entidade denominada *Bin7SegDecoder2* de modo a ser possível visualizar a palavra passe do utilizador nos displays hexadecimais, ao mesmo tempo que este a altera. A saída *valid* irá por sua vez fazer com que a máquina de estados, *LockerFSM*, transite do estado *BeginLocker* para o estado *CompareSuper* ou para o estado *CompareMaster*.

A máquina de estados denominada *LockerFSM*, possui 7 estados. Esta entidade é o bloco mais importante do projeto inteiro e possui na sua estrutura várias entradas e várias saídas. Esta entidade tem como entradas um *reset*, um relógio (*clock*), uma entrada *UserValid* (que interliga com a saída *valid* da entidade *ReadCode*), uma entrada *TimeExp* (que indica quando é

que a penalidade já passou), uma entrada *UserCode* (que interliga com a saída *Code* da entidade *ReadCode*), uma entrada *Close* (que faz fechar a fechadura), uma entrada *ChangePass* (que indica-nos quando é que o utilizador quer mudar a palavra passe de supervisor).

Quanto às saídas da máquina de estados, esta possui uma denominada *UserEnable* (que faz *enable* do *ReadCode*), uma denominada *TimeVal* (que inicia a máquina de estados auxiliar que conta a penalidade), uma denominada *Penalty* (que devolve à máquina de estados auxiliar quando tempo de penalidade esta deve contar), uma denominada *Counter* (que permite-nos ver nos displays hexadecimais o número de tentativas), uma denominada *Show* (que é ativada enquanto o penalidade ainda estiver em vigor) e uma denominada *OpenLock* (que permite-nos ver quando é que a fechadura abre).

Seguidamente, temos outra entidade denominada *TimerAuxFSM*. Esta é uma entidade que calcula a penalidade que o utilizador tem quando erra na palavra passe. Quando esta entidade recebe o sinal *TimeVal* (vindo do *LockerFSM*) ativo, então a entidade começa a contar até à penalidade estipulada (sinal mais uma vez vindo da entidade *LockerFSM*). Quando a penalidade acaba, esta máquina de estados envia um sinal *TimeExp* à máquina de estados *LockerFSM*.

Por último, as entidades que nos restam são apenas auxílios para ser possível visualizar a informação em *displays* hexadecimais.

## Implementação

---

Nesta secção, iremos explicar o que faz cada entidade, para que são necessárias e como funcionam. Em anexo, segue-se a representação gráfica da máquina de estados finitos, *LockerFSM*.

A entidade *ReadCode* é a entidade que permite-nos escrever a palavra passe a partir do uso das *keys*, de modo a ser mais fácil a introdução da palavra passe. Esta entidade também enviará um sinal *valid* para a máquina de estados quando a palavra passe estiver introduzida.

A entidade *ReadCode* o que faz é a cada *key* pressionada altera ou o dígito da palavra passe em que se encontra (incrementado ou decrementando) ou mover para o dígito seguinte ou então introduzir os dígitos na entidade *LockerFSM*.

A entidade *LockerFSM* é uma máquina de estados, com sete estados. No estado inicial, fazem-se atribuições essenciais para inicializar o diagrama de estados e os seus sinais. Depois, passamos para o segundo estado, onde a máquina de estados irá aguardar pelo código e por um sinal válido vindo da entidade *ReadCode*, depois de recebido um sinal válido a máquina de

estados passa para um próximo estado onde irá comparar os códigos. Caso o utilizador ainda não tenha chegado às 10 tentativas então a máquina de estados irá comparar com o código introduzida com o código do supervisor, caso contrário irá comparar com o código *Master*.

Ao comparar com o código do supervisor, caso acerte no código, a máquina de estados faz com que a fechadura se abra caso contrário a máquina de estados vai fechar a fechadura e aplicar uma penalidade, durante a qual o utilizador não pode introduzir nenhuma nova palavra passe. Quando a fechadura está aberta o utilizador pode escolher ou não alterar a palavra passe do supervisor e também pode escolher quando fechar a fechadura. Se o utilizador escolher mudar a palavra passe então a máquina de estados vai para um estado em que fica a aguardar um sinal válido vindo do *ReadCode*.

Quando as tentativas do utilizador foram maior que 10, então a máquina de estados vai comparar com o código *Master*. Caso o código *Master* seja correto então a máquina de estados reinicia as tentativas e a penalidade e volta ao estado inicial, caso contrário volta a aplicar a penalidade.

## **Conclusão**

---

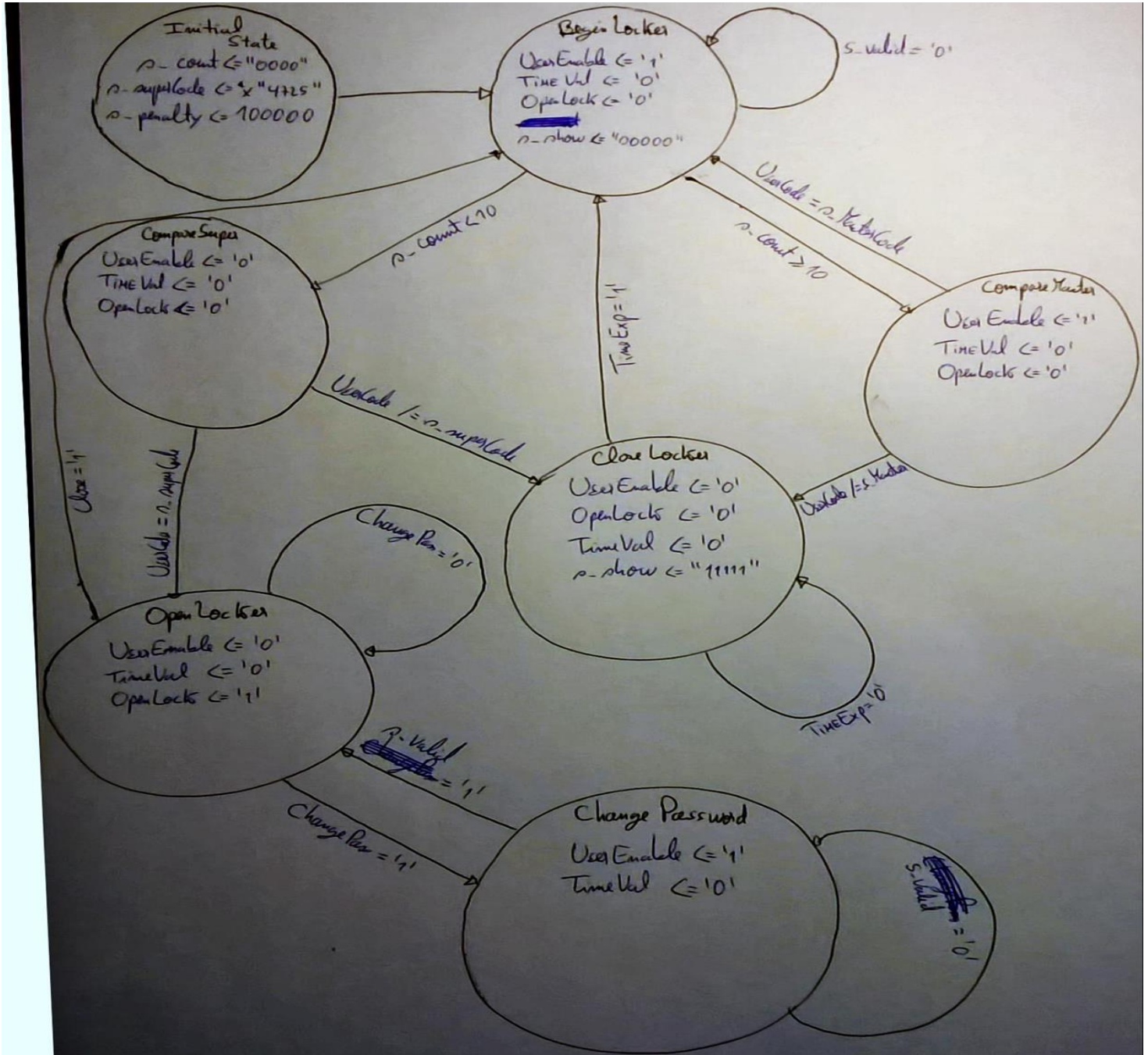
Após a conclusão deste projeto, sentimos que aprendemos muito mais sobre programação em *FPGA's*, foi possível ver algo realizado e funcional aplicando a teoria desenvolvida ao longo deste semestre. O plano inicialmente projetado para resolver este projeto provou estar bastante errado, sendo depois alterado. Foi um projeto desafiante e bastante esclarecedor sobre as potencialidades que os sistemas digitais possuem.

A maior dificuldade que atravessámos na realização deste projeto centrou-se com a construção da máquina de estados *LockerFSM*. Foi possível a realização desta máquina de estados após se ter adotado uma postura faseada em termos de programação.

Com isto retiramos, que foi um projeto aliciante com nível de dificuldade que esperaríamos encontrar.

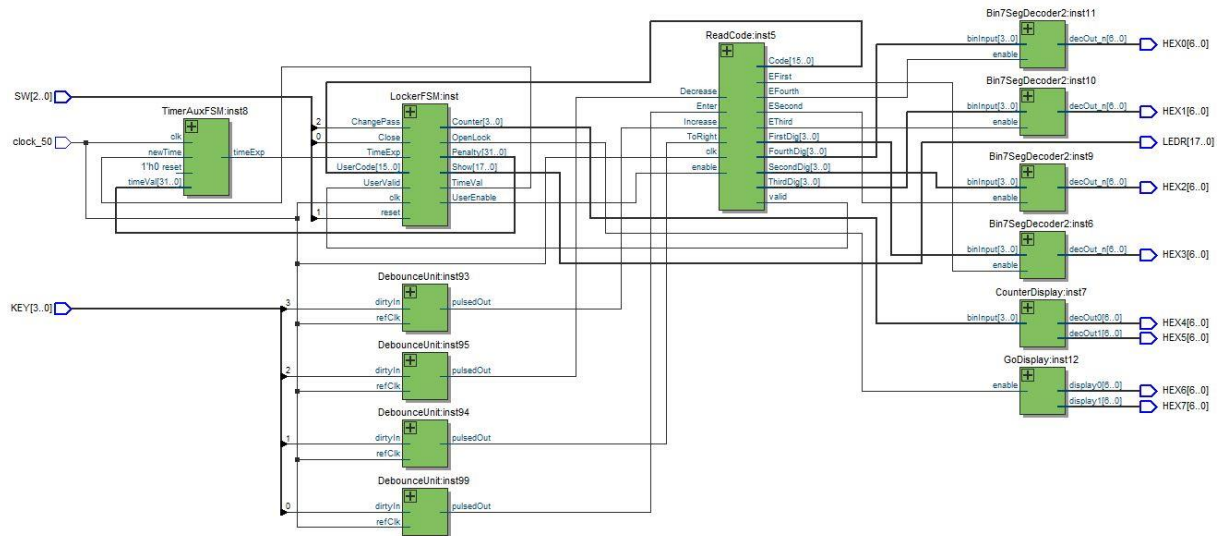
## Anexos

### ▪ Máquina de Estados Finita – LockerFSM

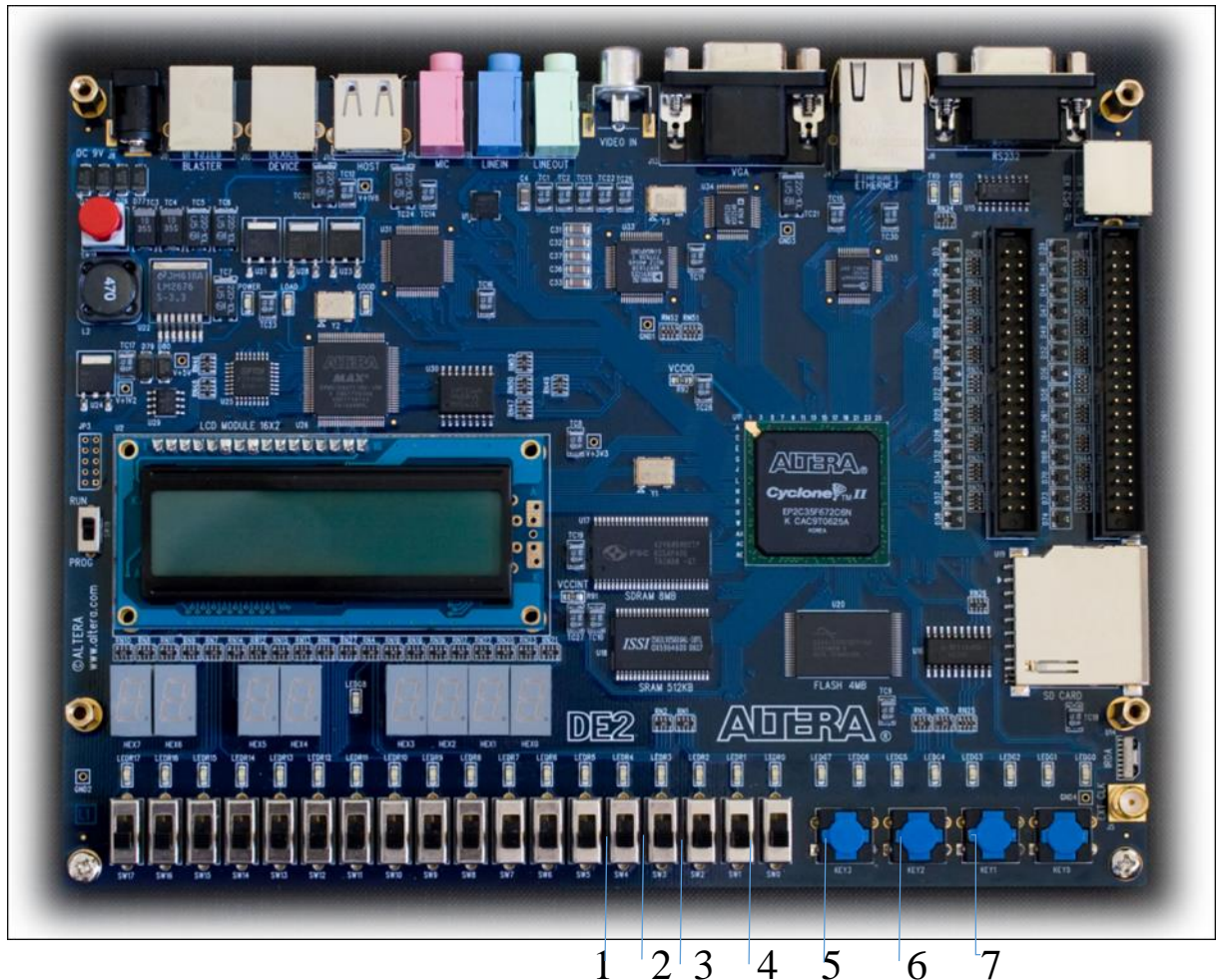




## ■ Diagrama de Blocos



## ■ Manual de Utilizador



- Switch SW2(1) - usado para selecionar a opção *ChangePass*
- Switch SW1(2) – usado para o reset da *LockerFSM*
- Switch SW0(3) – usado para fechar a fechadura
- Key KEY3(4) - usado para incrementar uma unidade no dígito a ser alterado na palavra passe
- Key KEY2(5) - usado para subtrair uma unidade no dígito a ser alterado na

- palavra passe
- Key KEY1(6) – usado para alterar dígito, neste caso, irá alterar o dígito para o dígito à direita
- Key KEY0(7) – usado como *enter*, para enviar a chave introduzida

Ao iniciar o programa, é necessário introduzir uma palavra passe. Para a introduzir, faz-se uso dos botões (4), (5), (6) e (7), sendo que as suas funções já foram explícitas no parágrafo anterior.

Caso o utilizador acerte na palavra passe irá entrar num estado onde, caso o (1) seja ativado é permitida a introdução de uma nova palavra passe, essa palavra passe acabada de introduzir será definida como a palavra passe a descobrir na fechadura. Caso o utilizador, não pretenda alterar a palavra passe então pode fechar a fechadura ativando o (3).

Caso algo corra mal, existe o interruptor (2) que faz o *reset* da máquina de estados devolvendo-a à sua posição inicial.