

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS BLUMENAU
DEPARTAMENTO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO – CAC
CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Adriano Pertile, Gabriel Linshalm Köhler

Projeto Final da disciplina BLU3048

Blumenau

2022

1 Introdução

O trabalho detalhado a seguir é o projeto final da disciplina de Tópicos Especiais em Mecatrônica I, ministrado pelo professor Doutor Marcos Vinicius Matsuo. O principal objetivo do trabalho é desenvolver um sistema de autenticação próxima ao apresentado na Figura 1.

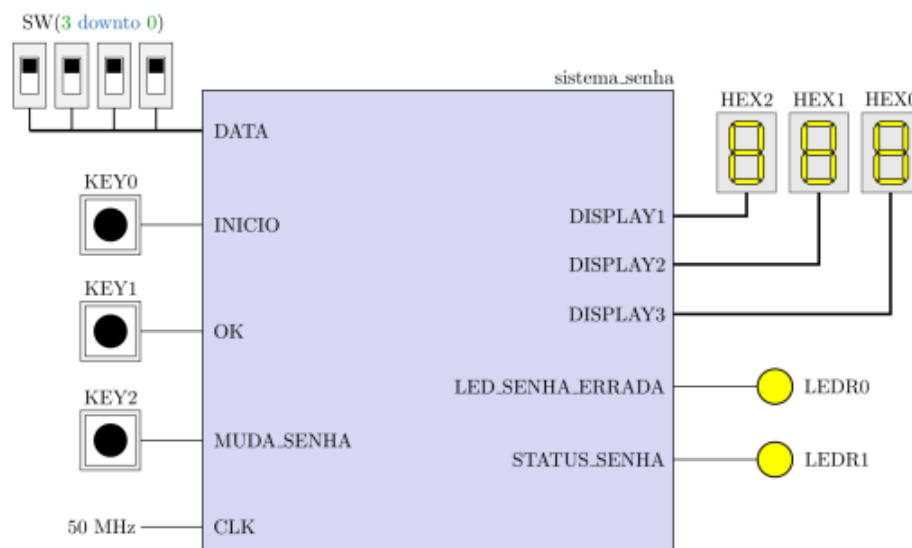


Figura 1 – Interface do sistema de comunicação. Fonte: Professor Doutor Marcos Vinicius Matsuo.

Como mostrado na Figura 1, temos 5 entradas e 5 saídas. A entrada DATA será utilizada para o fornecimento da senha pelo usuário, utilizando 4 bits; A entrada INICIO é responsável por inicializar o sistema; A entrada OK é utilizada sempre após o usuário escolher o valor da senha desejada na entrada DATA; A entrada MUDA_SENHA é responsável por dar a possibilidade de mudar a senha caso o usuário desejar; A entrada CLK é responsável pelo clock utilizado no sistema.

Para as saídas temos inicialmente 3 displays de 7 bits, utilizados para mostrar o número utilizado pelo usuário; A saída LED_SENHA_ERRADA é acionada sempre que a senha digitada pelo usuário for incorreta; A saída STATUS_SENHA será ativada sempre que a senha digitada pelo usuário for correta.

O projeto foi disponibilizado com um total de 3 níveis de dificuldade, onde conforme se aumenta o nível se aumenta o número de tarefas a serem realizadas. Neste projeto foi desenvolvido apenas até o nível 2.

Para o primeiro nível o sistema é responsável pela verificação da senha digitada pelo usuário e caso correta o LEDR1 deve ser acionado. Caso a senha seja incorreta o

LEDR0 deve ficar piscando. Para o segundo nível, o sistema deve realizar o primeiro nível e após isso dar a opção de mudar a senha armazenada nos registradores.

Para o trabalho, deve ser usado a abordagem RTL, onde temos o sistema dividido em bloco de controle e bloco operativo, como mostrado na Figura 2.

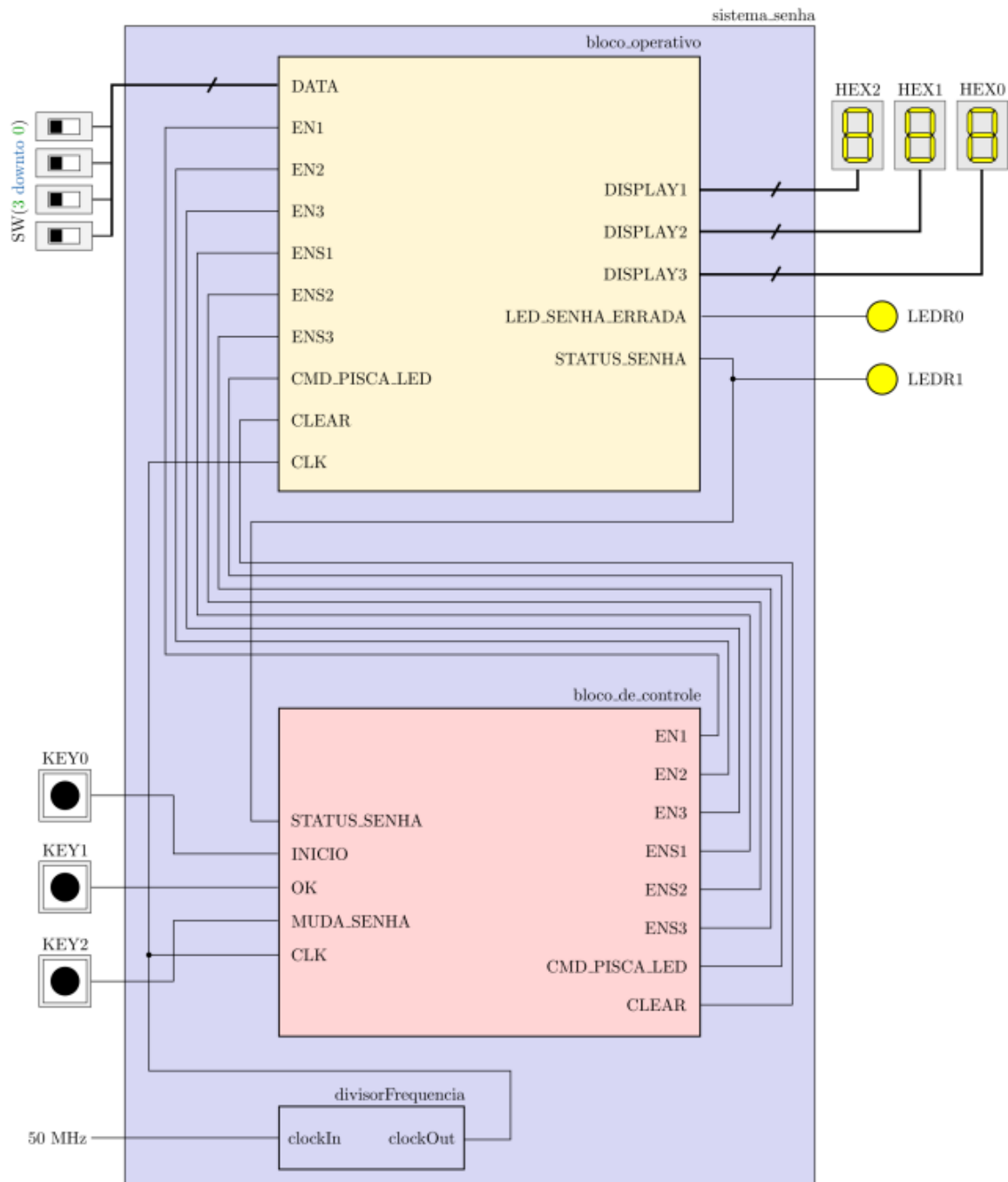


Figura 2 – Bloco operativo e bloco de controle do sistema de autenticação. Fonte: Professor Doutor Marcos Vinicius Matsuo.

2 Descrição do sistema desenvolvido

Esta seção apresenta a descrição com detalhes do bloco operativo e de controle implementados, mostrando ainda o diagrama de estados.

2.1 Bloco operativo

O bloco operativo foi disponibilizado juntamente com o enunciado do trabalho. Este bloco é uma entidade de "top-level" perante seus blocos internos. Isso é, o bloco operativo possui outros blocos dentro dele e faz com que eles se comuniquem entre si.

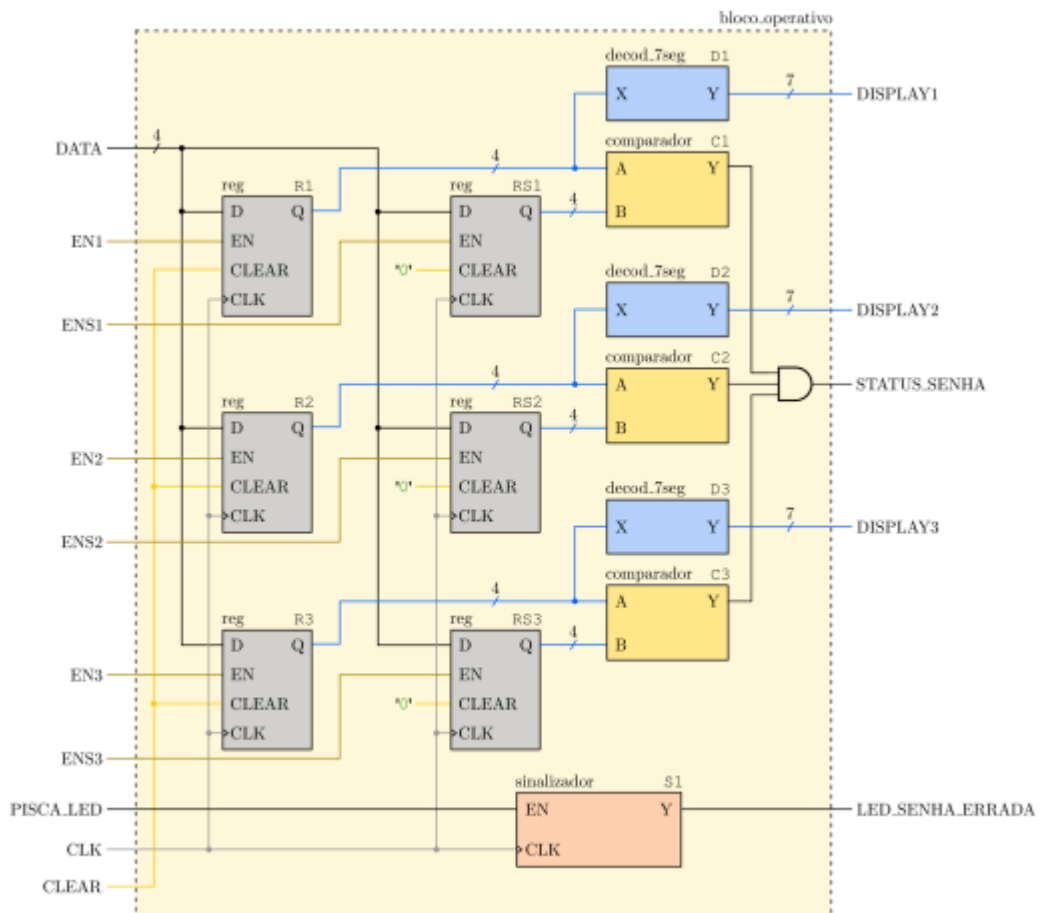


Figura 3 – Bloco Operativo. Fonte: Professor Doutor Marcos Vinicius Matsuo.

A Figura 3 demonstra como bloco operativo funciona e como ele se comunica internamente. O que facilita o entendimento de como cada estado deve atuar para garantir o funcionamento adequado do sistema de senha.

O bloco operativo é responsável pela inserção da senha, bem como a comparação dela, sua indicação em caso de senha correta e incorreta através de LED's e a visualização

dos dígitos da senha junto ao visor do tipo HEX, utilizado pela plataforma ALTERA Cyclone IV.

2.1.1 Registrador

Os registradores, identificados por "reg" na Figura 3, são responsáveis pelos armazenamentos das informações de entrada, nesse caso a entrada é dada por bits que correspondem algarismos de 0 a 9.

Esses registradores são um flip-flop do tipo D, ou seja, a informação que está na entrada será a informação da saída caso a entrada Enable tenha seu nível lógico alto. E o responsável por essa mudança de nível lógico é o bloco de controle. Isso ocorre para garantir que somente um registrador irá receber o valor por vez.

2.1.2 Comparador

Os comparadores, identificados por "comparador" na Figura 3, têm como função verificar se o dado da entrada A é igual ao dado da entrada B. Nesse caso o valor da entrada A será proveniente da saída do registrador de entrada. Já o valor da entrada B é proveniente da saída do registrador do dígito de senha gravado.

Caso os valores sejam iguais, o sinal de saída Y irá receber um bit igual a 1, ou seja verdadeiro. Caso os valores sejam distintos entre as entradas, o valor de saída Y será falso, recebendo um bit igual a 0.

2.1.3 Decodificador de 7 segmentos

O decodificador de 7 segmentos, identificados por "decod_7seg" na Figura 3, é responsável pela decodificação do dado recebido pelos registradores de entrada em uma saída de 7 bits. Cada um desses bits é responsável pela ativação de um segmento do display que contém 7 segmentos.

Com isso é possível transformar 4 bits em um número de 0 a 9 o qual o display será capaz de informar. Essa interface tem como objetivo informar ao usuário qual o valor inserido para a senha.

2.1.4 Sinalizador

O sinalizador, identificado por "sinalizador" na Figura 3, é usado para informar ao usuário caso a senha inserida esteja incorreta. Sua forma de sinalizar isso é fazendo com que um LED fique piscando intermitentemente.

Para isso, usa-se um divisor de frequência, que está incluso no bloco do sinalizador. Então caso a entrada PISCA_LED seja igual 1, o sinalizador é ativado e sua saída Y irá

ativar e desativar o LED fazendo com que ele fique piscando e informando ao usuário que a senha está incorreta.

2.2 Bloco de controle

Já o bloco de controle é um bloco cujo é responsável pelo funcionamento de forma sucessiva, isso é, seguindo uma ordem pré-estabelecidas de estados que o sistema deve assumir. Nesse bloco também fica presente a interface em que o usuário terá a maior interação, já que a condição de avanço entre cada estado depende do usuário interagir com as entradas do sistema.

A Figura 2 demonstra quais devem ser as entradas do bloco bem como suas saídas. Já quanto a parte lógica do bloco de controle pode ser ilustrada por um diagrama de blocos, conforme Figura 4, onde cada bloco representa um estado do sistema e cada flecha possui uma condicional.

Dessa forma entende-se que a cada estado as saídas do sistema devem mudar. Essas mesmas saídas são entradas do bloco operativo, ou seja, a cada novo estado implicará em mudanças na saída do bloco operativo que, como já mencionado, fica responsável pelas saídas do sistema como um todo até o usuário.

Para o bloco de controle implementado inicialmente é definido todos os estados presentes na aplicação, como mostrado na Figura 4. Para a resolução até o nível 2 foram necessários 16 estados (S0 até S15). Após definir todos os estados, a cada ciclo do clock é verificado o estado atual, de uma maneira sequencial, definindo todos os níveis lógicos das entradas e saídas mostrados na Figura 1.

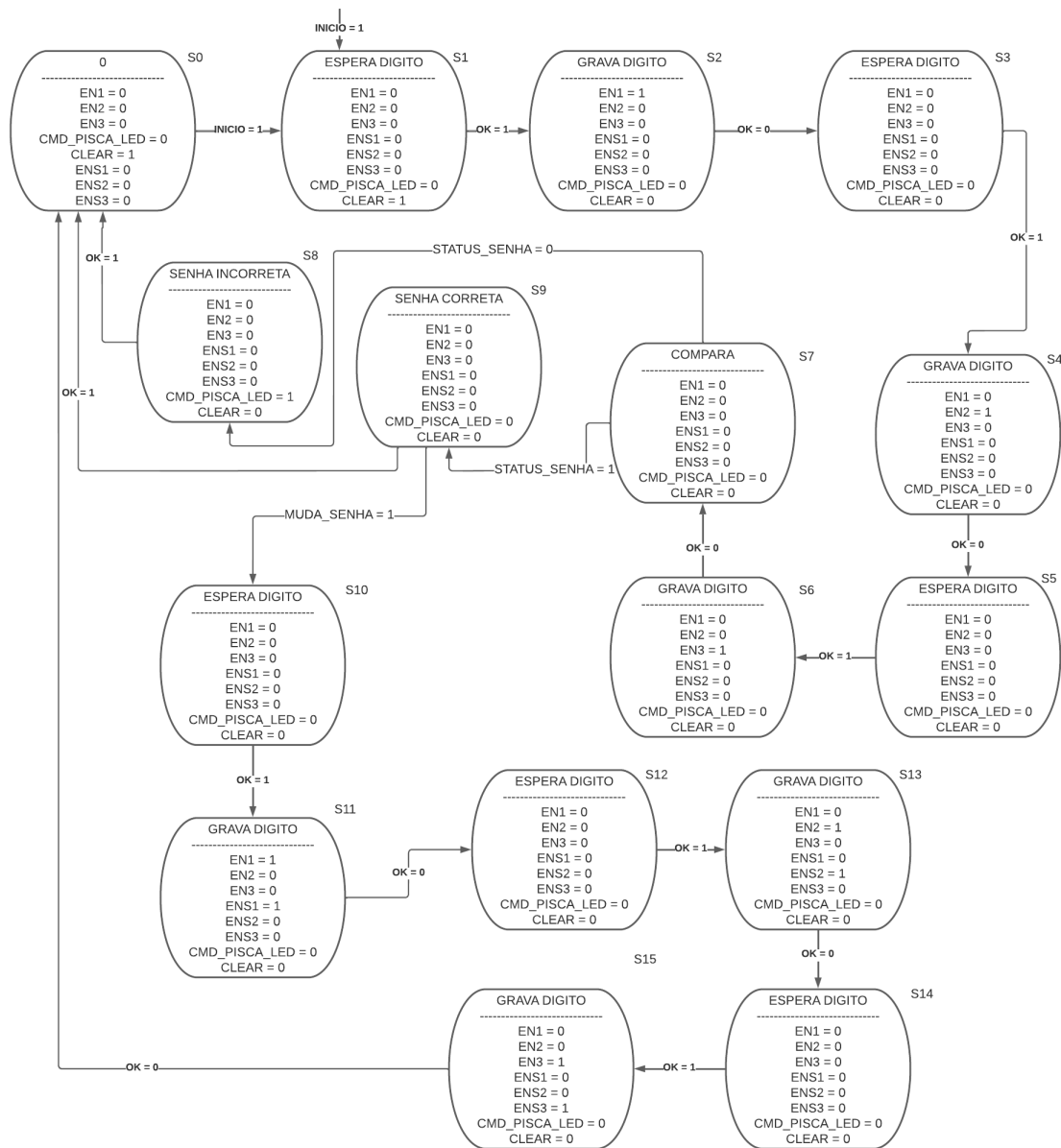


Figura 4 – Estados. Fonte: Os autores.

2.3 Bloco topo: Sistema Senha

Por fim, o último bloco contido no sistema é o bloco com a maior prioridade do sistema, isso é, de maior nível perante os outros. Esse bloco é responsável pela funcionamento do sistema como um todo, ou seja, pela comunicação entre os sub-blocos que o sistema contém.

Com esse bloco as comunicações de entrada e saída junto a placa ALTERA Cyclone IV são realizadas. Dessa forma as entradas usando KEY e SW são transferidas, através de sinais para os blocos de controle e operativo respectivamente. Já as saídas através dos LED's de status senha e pelo visor HEX também são transferidas através de sinais oriundos do bloco operativo para o topo que realiza a comunicação direta com a placa.

Como é requisitado no trabalho para ser usado a abordagem RTL, onde o comportamento do circuito é descrito através da transferência de dados entre os registradores presentes no hardware e as operações lógicas. A Figura 5 mostra essa comunicação para a transferência de dados entre os blocos operativos, de controle e as entradas e saídas do sistema.

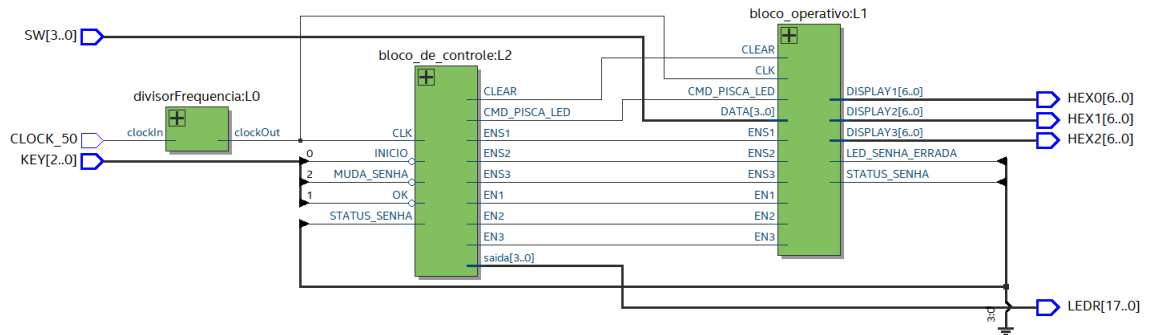


Figura 5 – RTL. Fonte: Os autores.

3 Considerações finais

O projeto teve como principal objetivo a implementação de um sistema de senha usando VHDL e a placa ALTERA Cyclone IV. Para essa implementação fez-se necessário o uso de diagramas e a separação do problema em estados.

Dessa maneira, a abordagem se deu analisando estado a estado, visando entender o que e como o hardware deveria se comportar a cada momento. Para essa implementação o sistema foi dividido em códigos distintos, sendo que, com uma entidade de prioridade superior, foi possível de realizar a comunicação entre eles de maneira sincronizada.

Com isso o problema explicitado pelo trabalho pôde ser resolvido atendendo aos níveis um e dois do enunciado com cerca de 16 estados. Os estados poderiam ter sido resumidos se não fossem levado em consideração os estados de transição, otimizando ainda mais o código do bloco de controle.