



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

CENTRO DE TECNOLOGIA - CT

CIRCUITOS DIGITAIS

COFRE DIGITAL

ELE2715 - Grupo 01 - Projeto - Problema 04

Albertho Síziney Costa

Igor Michael de Araujo de Macedo

Isaac de Lyra Junior

Sthefania Fernandes Silva

Vinicius Souza Fonseca

Natal, 14 de março de 2021

Albertho Síziney Costa
Igor Michael de Araujo de Macedo
Isaac de Lyra Junior
Sthefania Fernandes Silva
Vinicius Souza Fonseca

COFRE DIGITAL

Projeto da disciplina de Circuitos Digitais do
Departamento de Engenharia Elétrica da
Universidade do Rio Grande do Norte para relatório
das atividades.

Docente: Samaherni Morais Dias

Natal, 14 de março de 2021

RESUMO

Este relatório tem como objetivo desenvolver uma solução para o projeto de um circuito digital que simula um cofre. Este será capaz de controlar a abertura de uma trava eletrônica a partir de três senhas, de 6 bits cada, inseridas pelo usuário. No decorrer do texto, serão mencionadas todas as condições existentes para o funcionamento do cofre, de acordo com o que é pedido no problema, além de cada proposta de solução. A ideia é projetar uma resolução capaz de ser reproduzida, futuramente, em *softwares* de simulação de circuitos digitais, além de linguagens de descrição de *hardware*, como o VHDL. Serão abordados, assim, para o desenvolvimento, conceitos de circuitos sequenciais e máquinas de estados.

Palavras-chave: Circuitos Digitais. Circuitos Sequenciais. Máquinas de Estados. FSM. Cofre Digital. Trava Eletrônica.

SUMÁRIO

1 INTRODUÇÃO	5
2 DESENVOLVIMENTO	8
2.1 MÁQUINA DE ESTADOS	9
2.2 LÓGICA DO COFRE	11
2.3 DISPLAY	13
3 CONCLUSÃO	15
REFERÊNCIAS	16
ANEXOS	17

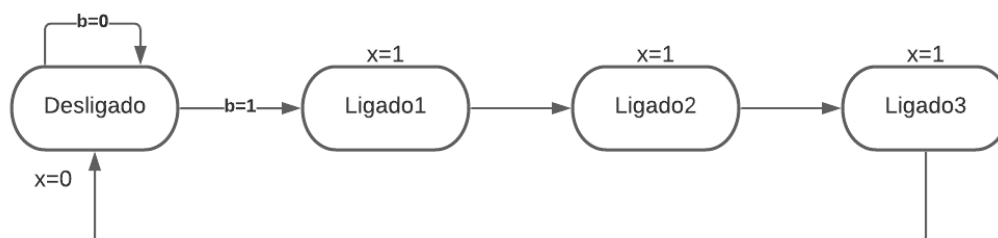
1 INTRODUÇÃO

Para projetar o comportamento de um circuito cujas saídas dependem apenas das entradas atuais, utiliza-se um conceito matemático nomeado equação booleana, o qual é convertido em circuitos combinacionais, caracterizados por não possuírem memória, ou seja, não há a possibilidade de armazenamento de bits para posterior utilização. Circuitos que possuem essa capacidade são os chamados sequenciais, que são compostos por estados, onde podem ser usados para projetar circuitos que tenham determinados comportamentos ao longo do tempo. Esses tipos de circuitos também são frequentemente denominados blocos de controle (VAHID, 2008).

Sabendo disso, circuitos que possuem diferentes estados que representam todos os modos possíveis de um sistema, podem ser denominados por máquinas de estado finitas (FSM - *finite state machine*). Na Figura 1 é possível ver um exemplo de uma máquina de estados para um temporizador que liga um laser durante três ciclos (VAHID, 2008).

O temporizador a laser deve permanecer desligado até que um botão seja pressionado, quando isso acontecer, um laser será ligado por 3 ciclos de relógio e depois será desligado novamente. Diante disso, pode-se construir um diagrama de estados composto por 4 estados: Desligado, Ligado1, Ligado2 e Ligado3. No estado Desligado, o laser deve estar desligado ($x=0$) e, enquanto o botão não for pressionado ($b=0$), permanece nesse estado. Para ligar o laser, o botão deve ser acionado ($b=1$) e um pulso de *clock* deve ocorrer, assim, será direcionado para o estado Ligado1. Após mais um ciclo de *clock*, é direcionado ao estado Ligado2 e, com mais um, para o estado Ligado3. No 4º pulso de *clock*, o sistema volta para o estado Desligado e, somente sairá dele, se o botão for pressionado novamente (VAHID, 2008).

Figura 1 - Diagrama de estados de um temporizador a laser.

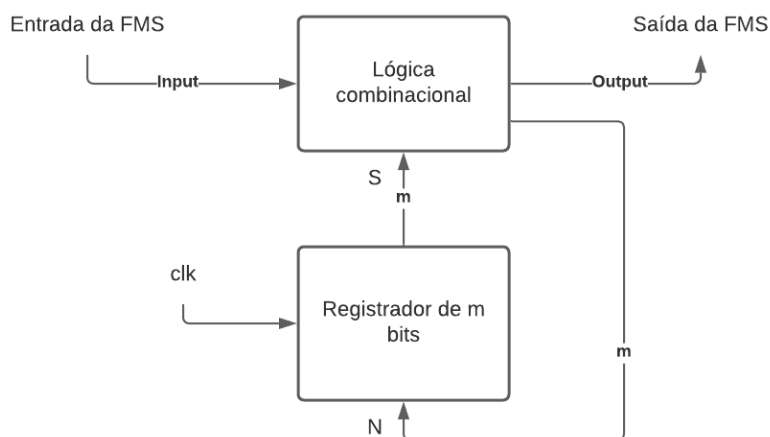


Fonte: Adaptado de VAHID(2008).

Com o conceito de FSM em mente, para descrever como se comportam circuitos com uso de *clock*, faz-se necessário um método que possa converter uma FSM em um circuito

sequencial. Para isso, é utilizada a arquitetura do bloco de controle, que consiste no uso de um registrador de estado e uma lógica combinacional (VAHID, 2008).

Figura 2 - Arquitetura de um bloco de controle padrão.



Fonte: Adaptado de VAHID(2008).

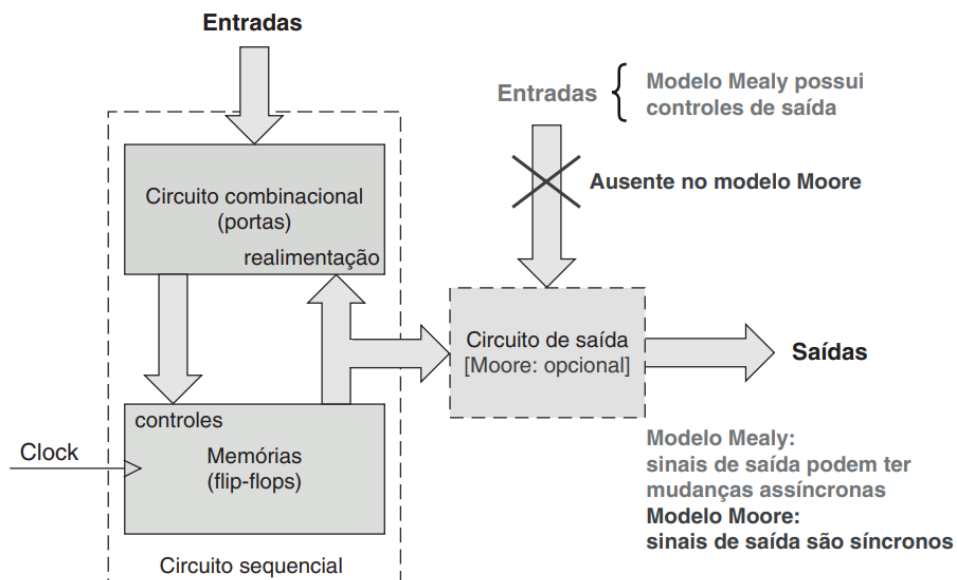
A Figura 2 mostra a arquitetura padrão de um bloco de controle. Nela existe um registrador de estado que contém um número binário para representar o estado atual do circuito. No que tange a lógica combinacional, há como entrada: as entradas da FSM (*input*) e as saídas do registrador; já para as saídas: a saída da FSM (*output*) e os bits que representam o próximo estado, os quais serão entradas do registrador de estados. Diante disso, o fator determinante para abranger as particularidades de cada circuito e determinar o seu comportamento será a lógica combinacional (VAHID, 2008).

Para determinar como esse circuito combinacional será construído é preciso utilizar uma tabela de estados. Essa tabela terá como entrada o estado atual do sistemas e suas entradas, como saída ela terá o próximo estado e as saídas do circuito. Tudo isso é feito em relação ao diagrama montado. Por fim, pode ser utilizado o mapa de Karnaugh e montar as expressões lógicas para definir o circuito lógico que será usado no problema (ALBERTINI, 2019).

A arquitetura mostrada na Figura 2 é o bloco de controle mais simples encontrado na literatura, podendo haver duas variações para adequação aos problemas: o modelo Mealy e Moore. No modelo Mealy, a saída é controlada por entradas adicionais, enquanto que, no Moore não há variáveis de controle externo nos sinais de saída. Logo, a saída do modelo Moore é função do estado atual do *flip-flop*. A consequência disso, é que as saídas de um circuito no modelo Moore serão síncronas, já as saídas do circuito Mealy podem ser alteradas

assincronamente (TOCCI, 2011). A Figura 3 sintetiza o que foi explanado sobre os modelos Mealy e Moore.

Figura 3 - Diagrama em bloco de uma máquina de estado.








Fonte: Tocci (2011).

Em suma, com base no referencial teórico apresentado, o relatório discutirá a elaboração do projeto de um cofre digital utilizando os conceitos vistos.

2 DESENVOLVIMENTO

O problema consiste em projetar um circuito que possa implementar um cofre digital. O sistema projetado deverá controlar a trava eletrônica do cofre, na qual somente deverá ser aberta quando as três senhas inseridas pelo usuário estiverem corretas. A Figura 4, mostra a interface visualizada pelo usuário e as configurações pré-definidas de cada elemento.

Figura 4 - Interface homem-máquina do cofre eletrônico.

Elemento	Descrição
	Display para exibição do valor a ser introduzido
	Led RGB: vermelho indica trava fechada, azul indica processando informação e verde indica trava aberta
	Botão para inicializar o processo de introdução da senha (<i>Pushbutton</i>)
	Botão para adicionar valor para a senha (<i>Pushbutton</i>)
	Chave para definição do valor da senha (esquerda: 0; direita: 1)

Fonte: Dados do problema.

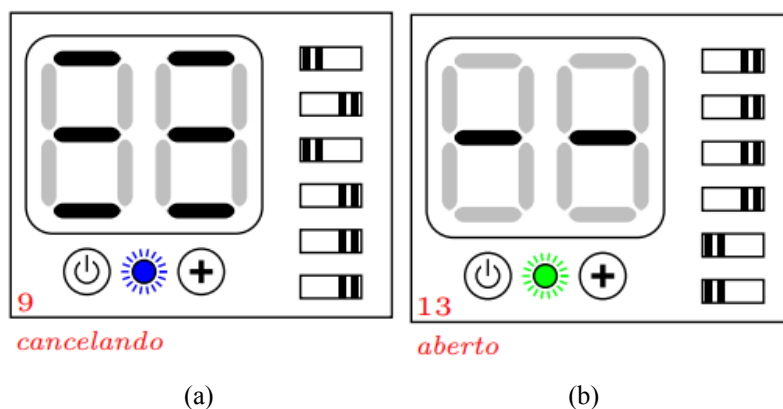
O sistema deve funcionar conforme descrito: Primeiramente, a senha, assim como o valor inserido pelo usuário, deve estar em um intervalo de 0 a 63. Quando a trava estiver fechada, sinalizada por um LED vermelho, o usuário deve apertar o botão de iniciar e a inicialização será feita. O processo de inicialização é representado por um LED azul aceso por 5 segundos. Até então o *display* deverá estar desligado. Após isso, a interface muda: o *display* é ligado e é exibido o valor das chaves de definição da senha. A partir de agora, pode ser inserida usando as chaves. Para adicionar a senha, o usuário deverá pressionar o botão de adicionar valor. Entre a introdução de cada senha, haverá um intervalo semelhante ao do processo de inicialização (LED azul ligado por 5 segundos).

Após o usuário inserir as 3 senhas, se a sequência e os valores estiverem corretos a trava é aberta por 20 segundos (a sinalização de abertura do cofre é feita por um LED verde e uma configuração específica do *display*); caso contrário, o cofre continua fechado e a interface muda para tela inicial (sinalizado por um LED vermelho e o *display* apagado).

Caso, durante a introdução da senha, o usuário queira cancelar a abertura do cofre, ele deve apertar o botão de iniciar. O processo de cancelamento será sinalizado através de um LED azul e uma configuração específica do *display*, e após 5 segundos ele volta à tela inicial.

As configurações do *display* para a abertura do cofre e cancelamento são mostradas na Figura 5.

Figura 5 - a) *Display* ajustado para o processo de cancelamento b) *Display* ajustado para a abertura do cofre.



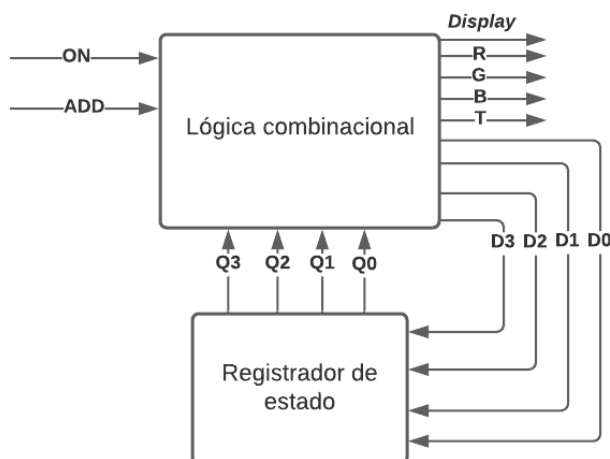
Fonte: Dados do problema.

2.1 MÁQUINA DE ESTADOS

Para projetar o bloco de controle do cofre digital, foi seguida a metodologia do VAHID (2008), a qual divide esse processo em 5 passos: 1º criar a FSM; 2º criar a arquitetura; 3º codificar os estados; 4º criar a tabela de estados e 5º implementar a lógica combinacional. Como o objetivo do relatório é somente projetar o circuito, o último passo não foi realizado.

Nesse sentido, primeiramente foi criada a FSM do cofre digital, a qual possui 10 estados: Fechado, Iniciando, Senha1, Processa1, Senha2, Processa2, Senha3, Cancelar, AnalisaSenha e Aberto. Para a transferência entre todos esses estados ocorrer, foi preciso considerar que, além das entradas botão iniciar e botão de adicionar (*ON* e *ADD*, respectivamente), outras entradas seriam necessárias. Logo, foram criadas as “entradas internas”: *Espera5*, *Espera20* e *Certo*. Essas entradas fazem parte de um circuito lógico externo à FMS, o qual será explicado no próximo tópico. Por fim, o diagrama de estados da FSM montada pode ser visualizado no Anexo B.

Feito isso, foi criada a arquitetura do circuito, onde pode ser vista na Figura 6. Como entrada da FSM, temos *ON* e *ADD* e, também, há como entrada os estados atuais do cofre (*Q0*, *Q1*, *Q2* e *Q3*). Na saída da FSM, temos *R*, *G* e *B*, que representam os LEDs vermelho, verde e azul, respectivamente, e a saída *T*, que é a trava que permite ou não a abertura do cofre. Além disso, temos como saída *D0*, *D1*, *D2* e *D3*, que são os estados futuros.

Figura 6 - Arquitetura do cofre digital.

Fonte: Autores.

Na terceira etapa foi feita a codificação dos estados, as quais estão exibidas abaixo.

Tabela 1 - Codificação dos estados.

ESTADOS	Bits do registrador de estados			
	Q3	Q2	Q1	Q0
Fechado	0	0	0	0
Iniciando	0	0	0	1
Senha1	0	0	1	0
Processa1	0	0	1	1
Senha2	0	1	0	0
Processa2	0	1	0	1
Senha3	0	1	1	0
Cancelar	0	1	1	1
AnalisaSenha	1	0	0	0
Aberto	1	0	0	1

Fonte: Autores.

Por fim, utilizando a FSM e a codificação de estados, foi montada a tabela de estados para o cofre digital. Utilizando o mapa de Karnaugh para simplificar as expressões que definem cada uma das saídas, será possível implementar a lógica combinacional mostrada na Figura 6. A tabela de estados pode ser conferida no Anexo C.

2.2 LÓGICA DO COFRE

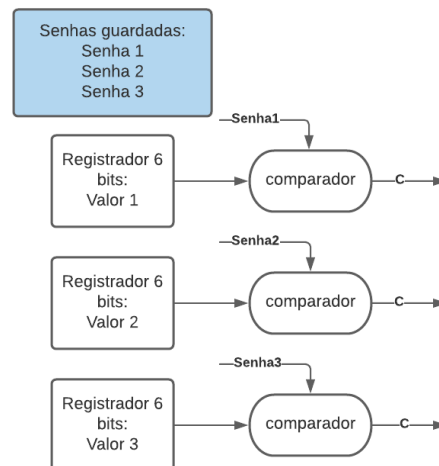
Como mostrado na seção anterior, a FSM do circuito vai possuir 10 estados e o detalhamento sobre cada um deles será feito a partir de agora.

O primeiro estado foi denominado “Fechado”, nele o *display* permanecerá desligado e o LED vermelho ligado. A mudança para o próximo estado ocorrerá quando somente o *pushbutton* “ON” for pressionado. No segundo estado, denominado “Iniciando”, o *display* permanecerá desligado e será emitido um pulso que permitirá o início da contagem do contador de 5 segundos. Durante essa contagem, o LED azul permanecerá aceso. A mudança para o próximo estado ocorrerá quando a contagem for terminada.

No estado “Senha1”, o *display* ligará e o usuário poderá escolher a primeira senha a ser inserida através das chaves de 6 bits. A mudança para o próximo estado ocorrerá quando somente o *pushbutton* “ADD” for pressionado. No estado seguinte, “Processa1”, o *display* permanecerá ativo com a senha que o usuário inseriu; será emitido um pulso para um registrador de 6 bits que armazenará a primeira senha inserida pelo usuário no estado anterior. Esse mesmo pulso é também enviado para o contador de 5 segundos que fará o LED azul ficar ligado e, apenas, quando o tempo acabar, será feita a mudança para o próximo estado.

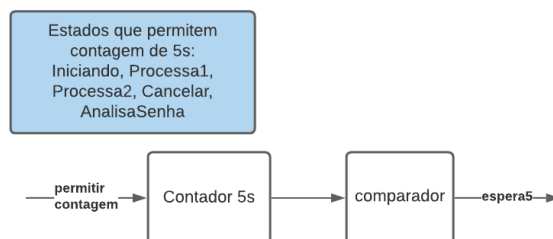
O quinto e sexto estado, denominados “Senha2” e “Processa2”, respectivamente, possuem o mesmo processo dos estados “Senha1” e “Processa1”. No entanto, a segunda senha inserida pelo usuário será armazenada em um registrador de 6 bits distinto do primeiro. No estado “Senha3”, o *display* permanecerá ligado, o led vermelho também e usuário colocará a terceira senha. A mudança de estado ocorrerá quando somente o *pushbutton* “ADD” for acionado. O estado “Cancelar” ocorrerá quando o usuário pressionar o *pushbutton* “ON”, enquanto estiver nos estados “Senha1”, “Senha2” ou “Senha3”. Nesse estado, o LED azul ficará ligado durante 5 segundos e após decorrido este tempo, o cofre voltará para o estado inicial “Fechado”.

O estado “AnalisaSenha” ocorrerá quando o usuário inserir a terceira senha. Nele, o LED azul ficará aceso por 5 segundos e durante esse período a terceira senha introduzida pelo usuário será armazenada em outro registrador de 6 bits. As três senhas armazenadas até então serão entradas para três comparadores que irão comparar os valores inseridos com as senhas do cofre. Caso a inserção do usuário tenha os valores e sequência correta a variável “Certo” receberá valor lógico alto e o cofre irá para o estado “Aberto”. Conforme mostrado na Figura 7.

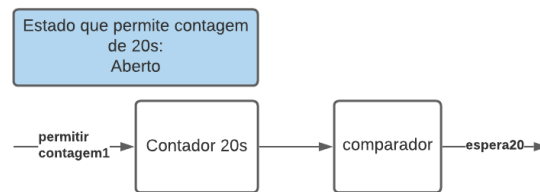
Figura 7 - Registrador e comparadores de senhas.**Fonte:** Autores.

No estado aberto, o LED verde ficará aceso durante 20 segundos e o *display* irá exibir o Padrão1 de segmentos, quando o tempo acabar o cofre voltará para o estado “Fechado”.

Note que, os estados Iniciando, Processa1, Processa2, Cancelar e AnalisaSenha têm uma particularidade em comum: os cinco possuem um contador de 5 segundos e um LED azul aceso. Diante disso, recomenda-se que seja criado um único contador para todos eles utilizando uma variável “CONTA5” que, de acordo com o estado, irá ativar o contador. A Figura 8, mostra o diagrama de blocos dessa proposta.

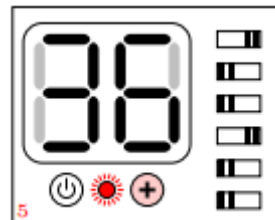
Figura 8 - Contador de 5 segundos.**Fonte:** Autores.

No que tange o contador de 20 segundos, recomenda-se uma lógica parecida. Mas dessa vez considera-se apenas o estado “Aberto”.

Figura 9 - Contador de 5 segundos.**Fonte:** Autores.

2.3 DISPLAY

Os *displays* exibirão os valores, em decimal, referentes às entradas selecionadas pelo usuário nas chaves, como está sendo mostrado na Figura 10.

Figura 10 - Chaves para definição do valor da senha.**Fonte:** Dados do problema.

Veja que, de acordo com a Figura 10, a chave para o lado direito indica nível lógico 1, e para o esquerdo, nível lógico 0. Além disso, temos que a chave mais acima é o bit mais significativo do número inserido. O *display* deve ser implementado para representar, em decimal, instantaneamente o valor das chaves, uma vez que a FSM esteja em um dos estados de senha, e também, representar duas outras combinações: situação de cancelamento e abertura do cofre (Figura 5).

Uma vez definida a lógica do funcionamento dos *display*, foi montada a tabela verdade das suas entradas e saídas, esta é mostrada na Figura 11.

Figura 11 - Tabela Verdade *Display* 7 Segmentos.

ENTRADAS				SAÍDAS							DÍGITO
<i>A SW</i> [3]	<i>B SW</i> [2]	<i>C SW</i> [1]	<i>D SW</i> [0]	<i>HEX</i> [6]	<i>HEX</i> [5]	<i>HEX</i> [4]	<i>HEX</i> [3]	<i>HEX</i> [2]	<i>HEX</i> [1]	<i>HEX</i> [0]	
0	0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	0	0	0	1	1	0	1
0	0	1	0	1	0	1	1	0	1	1	2
0	0	1	1	1	0	0	1	1	1	1	3
0	1	0	0	1	1	0	0	1	1	0	4
0	1	0	1	1	1	0	1	1	0	1	5
0	1	1	0	1	1	1	1	1	0	1	6
0	1	1	1	0	0	0	0	1	1	1	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	0	0	1	1	1	9
1	0	1	0	0	0	0	0	0	0	0	Desligado
1	0	1	1	1	0	0	1	0	0	1	Padrão 1
1	1	0	0	1	0	0	0	0	0	0	Padrão 2
1	1	0	1	0	0	0	0	0	0	0	Desligado
1	1	1	0	0	0	0	0	0	0	0	Desligado
1	1	1	1	0	0	0	0	0	0	0	Desligado

Fonte: Autores.

Como mostrado na Figura 11, cada *display* receberá as combinações de saídas, para os seus respectivos valores binários de 4 bits. Logo, faz-se necessário implementar um conversor binário-bcd que receba 6 bits e resulte em dois valores de 6 bits, sendo um direcionado para o *display* da dezena e outro para o *display* da unidade.

3 CONCLUSÃO

As máquinas de estado finito são um importante mecanismo para representar circuitos lógicos com elementos de memória. Esse método pode ser aplicado tanto em circuitos lógicos simples, como nos mais complexos. Nesse caso, as FSMs foram utilizadas para projetar um cofre digital.

O cofre em questão precisa ter sua interface alterada de acordo com a operação realizada pelo usuário. Para isso, há um padrão de acendimento dos LEDs RGB e dos *displays* de 7 segmentos. Para iniciar o cofre, cancelar o procedimento e adicionar as senhas, há os botões iniciar e adicionar. Caso o valor e a sequência introduzida estejam corretas, o cofre abre por 20 segundos, caso contrário, a interface retorna para o padrão de início. Sabendo disso, foi desenvolvido o projeto para realização de cada operação e espera-se que, se seguidos como sugerido, tenham resultados positivos.

Por fim, para projetar tal circuito fez-se necessário, além do conhecimento sobre máquinas de estado finito, ter um bom entendimento de circuitos combinacionais e sequenciais, como comparadores, *flip-flops* e registradores. Sendo assim, praticando ainda mais e fortalecendo tais conceitos tão utilizados em circuitos digitais.

REFERÊNCIAS

ALBERTINI, B.. **Máquinas de Estados Finitas**. 2019. Disponível em: https://balbertini.github.io/fsmbasics-pt_BR.html. Acesso em: 12 mar. 2021.

TOCCI, R.; WIDMER, N.; MOSS, G. **Digital Systems: Principles and Applications**. [S.l.]: Pearson Education Limited, 2011. ISBN 9780130387936.

VAHID, Frank. **Sistemas digitais: projeto, otimização e HDLS**. Rio Grande do Sul: Artmed Bookman, 2008. 558 p.

ANEXOS

ANEXO A - Relato semanal

Líder: Isaac de Lyra Junior

A.1 Equipe

Tabela 2 - Identificação da equipe.

Função do grupo:	Discente
Redator	Sthefania Fernandes Silva
Debatedor	Vinicius Souza Fonseca
Videomaker	Albertho Síziney Costa
Auxiliar	Igor Michael de Araujo de Macedo

A.2 Defina o problema

O problema consistia em projetar um circuito digital capaz de implementar um cofre digital. O circuito irá controlar a abertura da trava eletrônica de um cofre a partir de um pulso em nível alto na saída *Out*. A saída *Out* terá nível lógico alto quando a senha inserida pelo usuário for a correta, sendo esta composta por três números que devem ser introduzidos na sequência correta, caso o usuário erre a senha a saída *Out* terá nível lógico baixo e o cofre permanecerá fechado.

A.3 Registro do *brainstorming*

Na primeira reunião foi definido os cargos que cada membro do grupo ocuparia no projeto, além disso foi definido que todos do grupo teriam como obrigação ler o problema como um todo. A segunda reunião foi iniciada com todos os membros dando ideias de possíveis soluções para o problema em questão, ao final da segunda reunião foi feito um diagrama de blocos que reunia as ideias de todos os membros presentes para uma possível solução utilizando *flip-flops* para os temporizadores, registradores para armazenar as senhas e comparadores que deveriam comparar as senhas inseridas com as senhas armazenadas nos registradores, ficou como obrigação que todos os membros estudassem sobre Máquinas de

Estados Finitos (FSM). A terceira reunião foi iniciada com todos os membros mostrando o diagrama de estados que cada um pensou para o problema, foi discutido quantos estados o problema teria e quais seriam os próximos passos para a solução final, a reunião acabou com um consenso que o problema teria 8 estados. Na quarta reunião foi continuada a construção do diagrama de estados para o problema, porém, ao concluirmos a tabela de estados para os 8 estados pensados anteriormente, entramos em consenso que a solução que pensamos não poderia ser a correta, pois muitos dos estados que pensamos na verdade seriam processamento, a reunião foi terminada com um consenso que quatro dos oito estados pensados inicialmente não eram estados, dessa forma a solução pensada pelo grupo foi de 4 estados: Fechado, senha1, senha2, senha3. Na quinta reunião identificamos problemas ao utilizar apenas 4 estados, pois alguns processos como, por exemplo, o led azul ligando por 5 segundos antes de prosseguir para o próximo estado estava sendo difícil de imaginar a implementação, dessa forma, a reunião foi concluída com um consenso de que o projeto teria 10 estados distintos, são eles: Fechado, Iniciando, Senha1, Processa1, Senha2, Processa2, Senha3, Cancelar, AnalisaSenha e Aberto. A sexta e última reunião foi utilizada para fazer ajustes finais na solução do problema, conclusão do relatório e vídeo do projeto.

Figura 12 - Rotina de reuniões

Horário	Segunda	Terça	Quarta	Quinta	Sexta	Sábado	Domingo
M12							
M34							
M56							
T12						SEXTA REUNIÃO	
T34					QUINTA REUNIÃO		
T56							
N12	PRIMEIRA REUNIÃO (Definição dos cargos)	SEGUNDA REUNIÃO - PRIMEIRO BRAINSTORMING	TERCEIRA REUNIÃO - SEGUNDO BRAINSTORMING	QUARTA REUNIÃO			
N34							

Fonte: Autores

A.4 Pontos-Chaves

O ponto chave para a solução desse problema foi entender como uma máquina de estados finitos (FSM) funciona e definir quantos seriam os estados do problema, para que a

construção do diagrama de estados e tabela de transição de estados fosse feita de maneira correta.

A.5 Questões de pesquisa

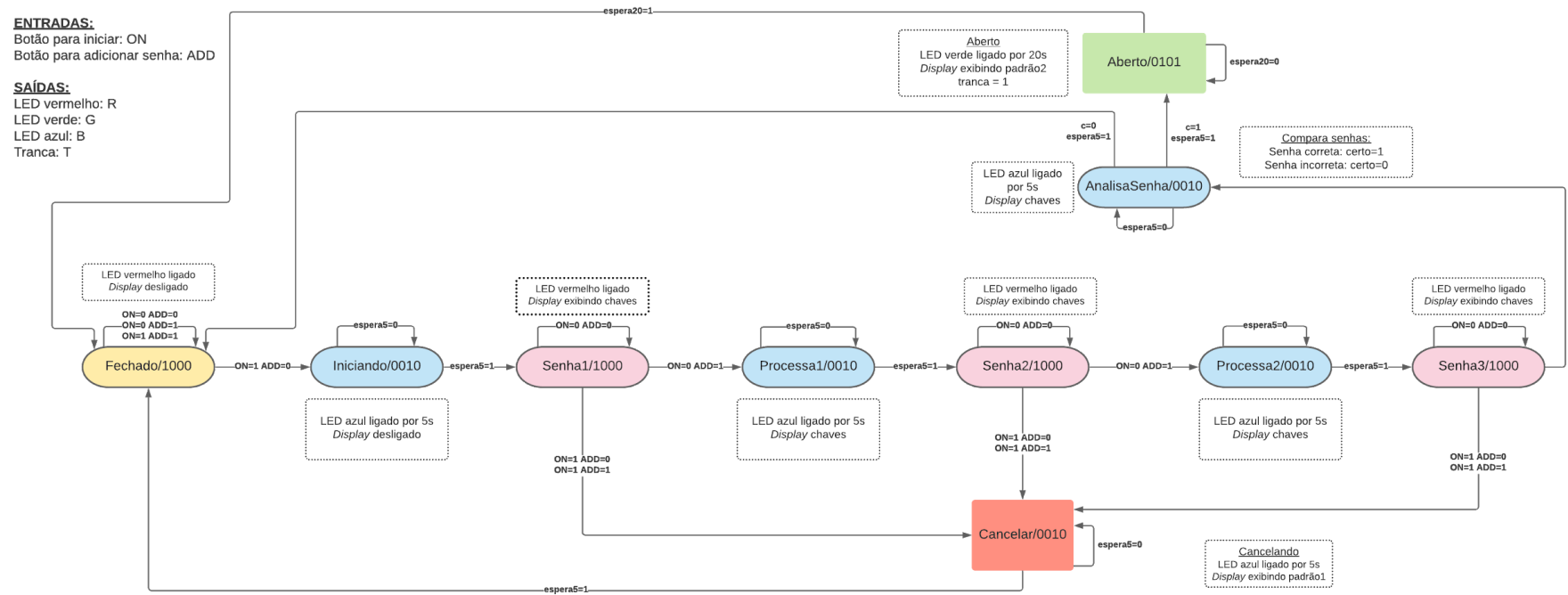
Para que a solução presente neste relatório fosse desenvolvida de maneira correta, foi necessário estudar os seguintes conceitos de circuitos digitais:

- Máquinas de estados finitos;
- Projeto de bloco de controle;
- Otimizações e tradeoff em lógica sequencial.

A.6 Planejamento da pesquisa

Para estudar os conceitos presentes no A.5 utilizamos como fonte o livro-texto “Sistemas digitais: projeto, otimização e HDLS” de Frank Vahid, como também o livro-texto “Sistemas digitais : princípios e aplicações ” de Ronald J. Tocci e também vídeo-aulas sobre os conceitos citados.

ANEXO B - Máquina de estado do cofre digital



ANEXO C - Tabela de estados do cofre digital

ESTADO	ESTADO ATUAL				ENTRADAS					ESTADO	PRÓXIMOS ESTADOS				SAÍDAS				
	Q3	Q2	Q1	Q0	ADD	ON	ESPERA5	ESPERA20	CERTO		D3	D2	D1	D0	R	G	B	T	Display
Fechado	0	0	0	0	x	0	x	x	x	Fechado	0	0	0	0	1	0	0	0	Desligado
	0	0	0	0	x	1	x	x	x	Iniciando	0	0	0	1	1	0	0	0	Desligado
Iniciando	0	0	0	1	x	x	0	x	x	Iniciando	0	0	0	1	0	0	1	0	Desligado
	0	0	0	1	x	x	1	x	x	Senha 1	0	0	1	0	0	0	1	0	Chaves
Senha 1	0	0	1	0	0	0	x	x	x	Senha 1	0	0	1	0	1	0	0	0	Chaves
	0	0	1	0	0	1	x	x	x	Cancelar	0	1	1	1	1	0	0	0	Padrão1
	0	0	1	0	1	0	x	x	x	Processa1	0	0	1	1	1	0	0	0	Chaves
Processa1	0	0	1	1	x	x	0	x	x	Processa1	0	0	1	1	0	0	1	0	Chaves
	0	0	1	1	x	x	1	x	x	Senha 2	0	1	0	0	0	0	1	0	Chaves
Senha 2	0	1	0	0	0	0	x	x	x	Senha 2	0	1	0	0	1	0	0	0	Chaves
	0	1	0	0	0	1	x	x	x	Cancelar	0	1	1	1	1	0	0	0	Padrão1
	0	1	0	0	1	0	x	x	x	Processa2	0	1	0	1	1	0	0	0	Chaves
Processa2	0	1	0	1	x	x	0	x	x	Processa2	0	1	0	1	0	0	1	0	Chaves
	0	1	0	1	x	x	1	x	x	Senha 3	0	1	1	0	0	0	1	0	Chaves
Senha 3	0	1	1	0	0	0	x	x	x	Senha 3	0	1	1	0	1	0	0	0	Chaves
	0	1	1	0	0	1	x	x	x	Cancelar	0	1	1	1	1	0	0	0	Padrão1
	0	1	1	0	1	0	x	x	x	AnalisaSenha	1	0	0	0	1	0	0	0	Chaves
Cancelar	0	1	1	1	x	x	0	x	x	Cancelar	0	1	1	1	0	0	1	0	Padrão1
	0	1	1	1	x	x	1	x	x	Fechado	0	0	0	0	0	0	1	0	Padrão1
AnalisaSenha	1	0	0	0	x	x	0	x	0	AnalisaSenha	1	0	0	0	0	0	1	0	Chaves
	1	0	0	0	x	x	1	x	1	Aberto	1	0	0	1	0	0	1	0	Chaves
	1	0	0	0	x	x	1	x	0	Fechado	0	0	0	0	0	0	1	0	Chaves
Aberto	1	0	0	1	x	x	x	0	x	Aberto	1	0	0	1	0	1	0	1	Padrão2
	1	0	0	1	x	x	x	1	x	Fechado	0	0	0	0	0	1	0	0	Padrão2