Departamento de Engenharia de Computação e Automação - DCA

Universidade Federal do Rio Grande do Norte - UFRN

Discentes:

Cicero Josean Mateus Nunes da Silva Matrícula: 20160001784

Jadson Araujo Bezerra Matrícula: 2016011250

Lucas de Azevedo Lima Matrícula: 2016014950

Docentes: Kennedy Reurison Lopes

Emanoel Raimundo Queiroz Chaves Junior

Turma: 02A

**Relatório de Circuitos Digitais**

**Cofre**

**NATAL, RN**

**2018**

**1 Introdução**

**2 Desenvolvimento**

**3 Conclusão**

**1 Introdução**

**VHDL** ou "**V**HSIC **H**ardware **D**escription **L**anguage" (Linguagem de descrição de hardware VHSIC "**V**ery **H**igh **S**peed **I**ntegrated **C**ircuits") é uma linguagem usada para facilitar o design (projeto/concepção) de circuitos digitais em CPLDs, FPGAs e ASICs.

Ela foi concebida pelo departamento de defesa dos Estados Unidos (DARPA) na década de 80 para a documentação dos circuitos vendidos às forças aéreas americanas. Em 1987 ela foi padronizada pela IEEE como uma linguagem de descrição de hardware, o fato de ser padronizada e de domínio público ampliou e muito a sua utilização.

Neste relatório será implementado um software simples de um cofre, que utiliza de conceitos de portas lógicas, flip-flops, comparadores dentre outros conhecimentos adquiridos em sala de aula na disciplina de circuitos digitais.

**2 Desenvolvimento**

De início, foi feito um estudo sobre a máquina de estados do sistema (FSM),

como seria cada estado da máquina e como as variáveis se comportam para transição de estado a estado. A partir daí montou-se uma tabela verdade do sistema assim possibilitando a implementação das expressões lógicas que seriam usadas para montagem do circuito na linguagem vhdl.

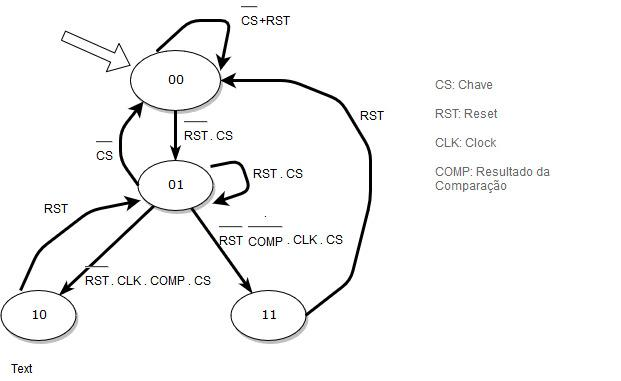
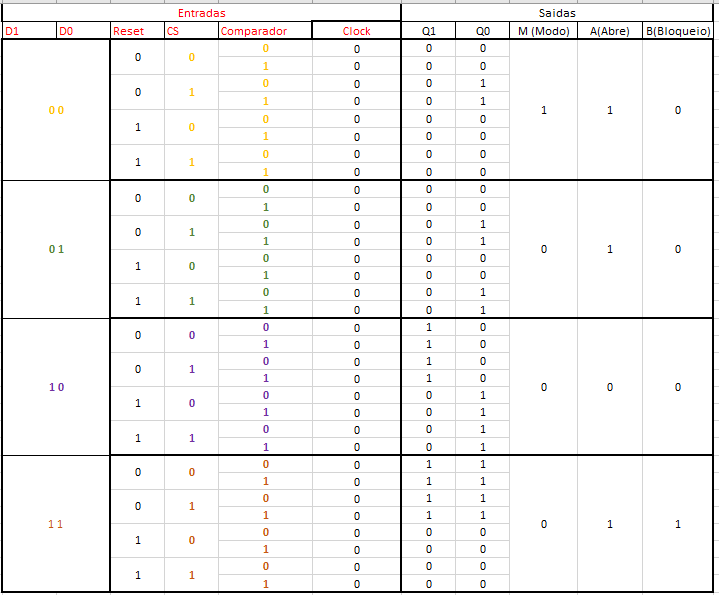


Fig. 01 – Diagrama de Estado

Os estados estão definidos como:

* 00: Estado inicial. Nessa configuração o sistema está pronto para cadastrar a senha. Após o cadastro da senha, deverá ser mudada a chave de configuração de CS para ativar o estado seguinte.
* 01: Nesse estado, o sistema fica esperando o usuário digitar a senha. Após a digitação da senha e apertar o botão clock, o sistema irá verificar se o usuário digitou a senha do cofre corretamente ou não.
* 10: Estado ativado quando se digitou a senha correta do cofre. Nesse caso o cofre se abre e o sistema espera o RST para voltar ao estado anterior.
* 11: Estado ativa quando o usuário digitou a senha incorreta. Esse é o estado de bloqueio, que só será mudado quando se apertar o botão RST.

Após a construção do diagrama de Estados, foi possível gerar a tabela verdade contendo todas as combinações possíveis das variáveis. O resultado foi o apresentado abaixo:

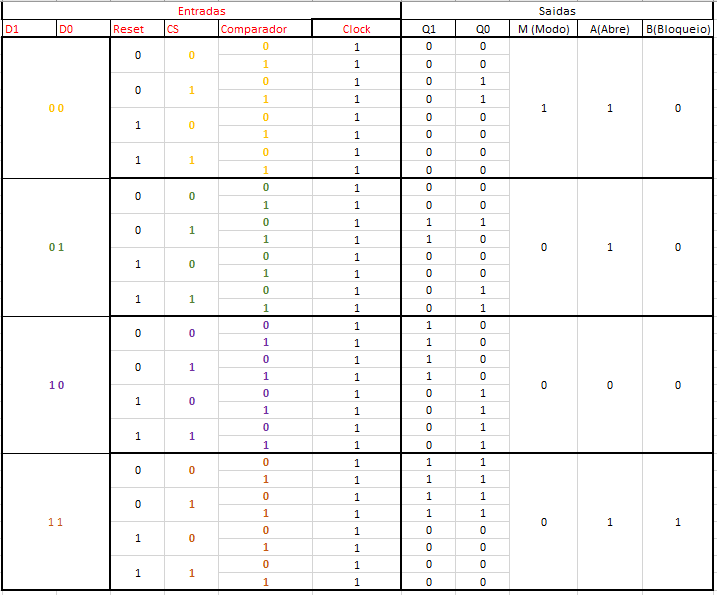


Fig. 02 – Tabela Verdade

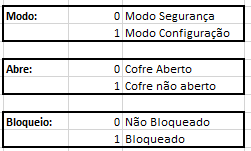


Fig. 03 – Saídas do Sistema

Para verificar se o funcionamento do circuito está correto, foram feitas simulações no programa Quartus. A primeira simulação foi feita com o usuário digitando a senha, e colocando a senha correta para abrir o cofre:

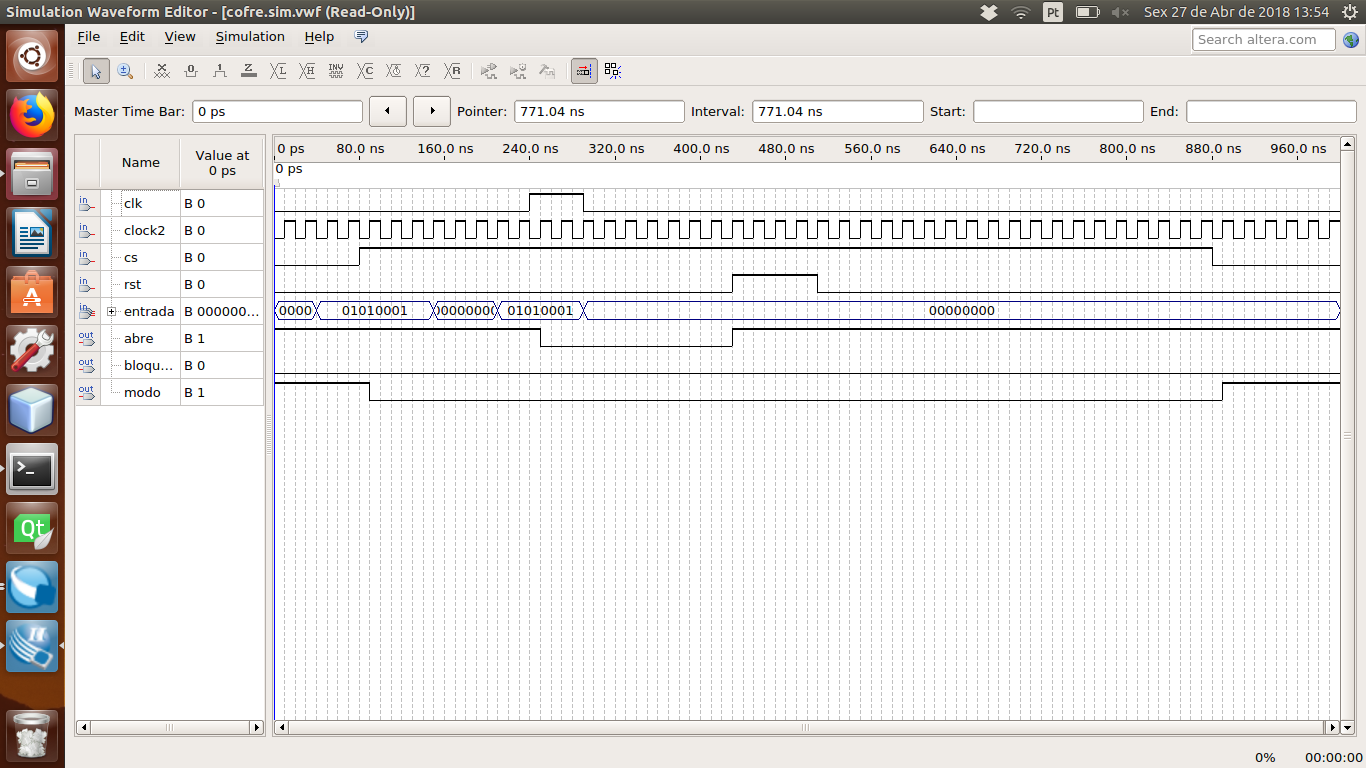


Fig. 04 – Simulação quando o usuário acerta a senha do cofre

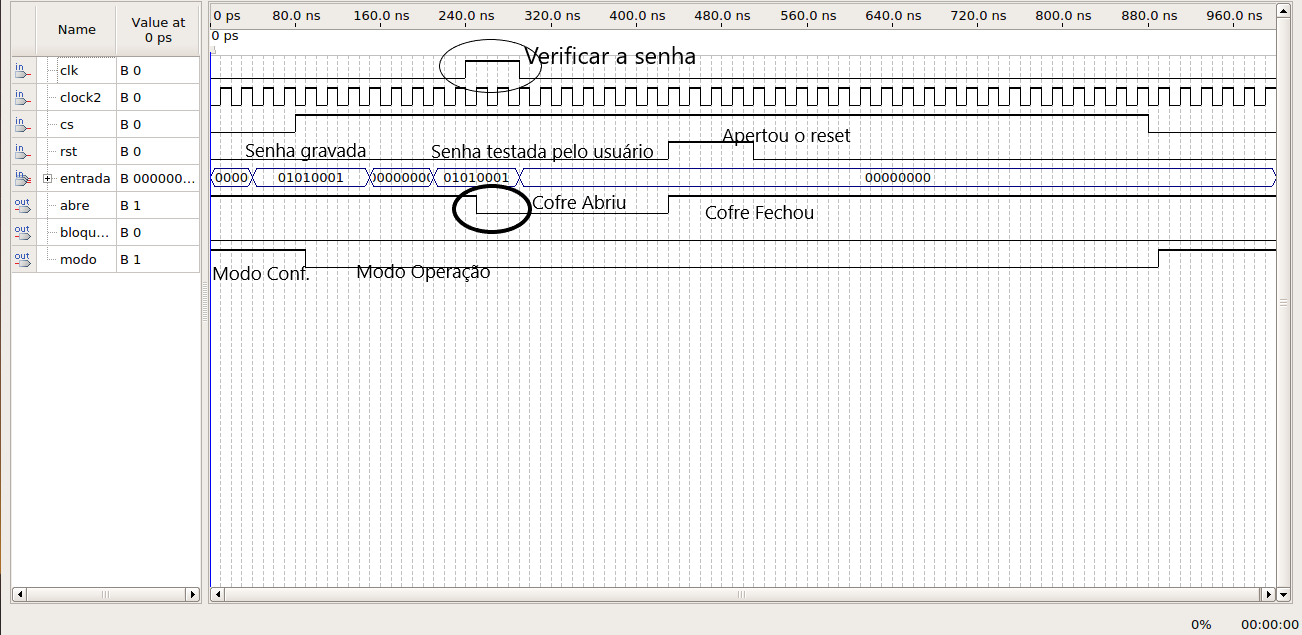


Fig. 05 – Interpretação quando o usuário acerta a senha do cofre

Posteriormente, foi cadastrada uma senha, e se tentou abrir o cofre com uma senha errada. Como era de se esperar, o cofre não abriu, e bloqueou o sistema. Nesse caso, é necessário apertar o reset para voltar ao estado inicial.

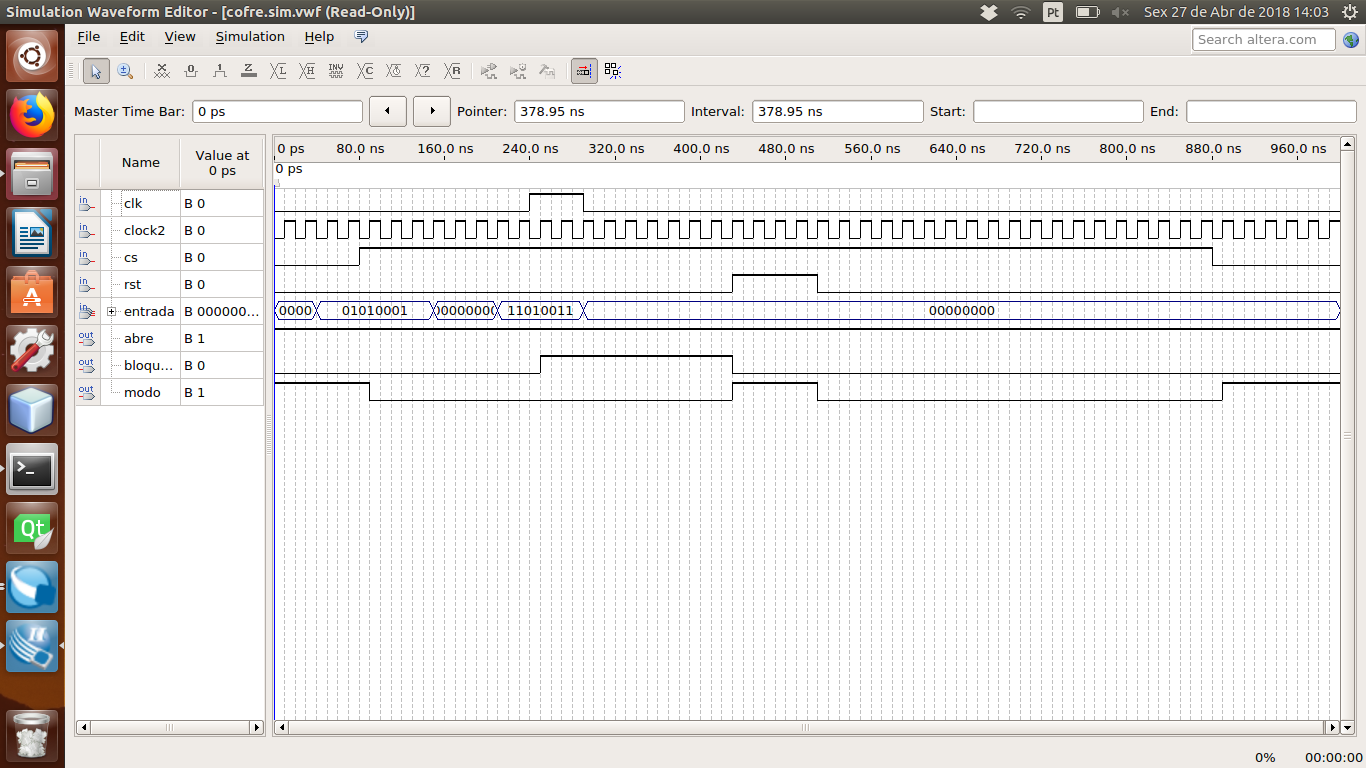


Fig. 06 – Simulação quando o usuário errou a senha do cofre

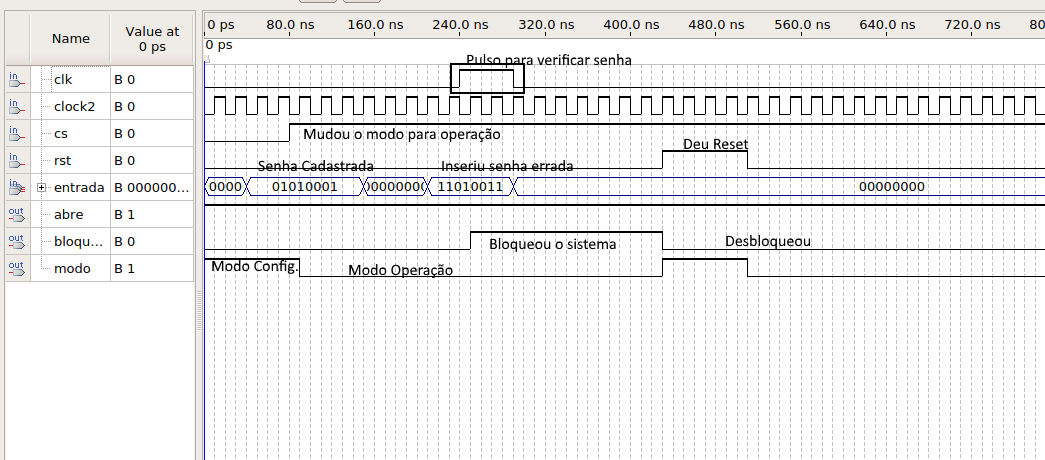


Fig. 07 – Interpretação quando o usuário errou a senha do cofre

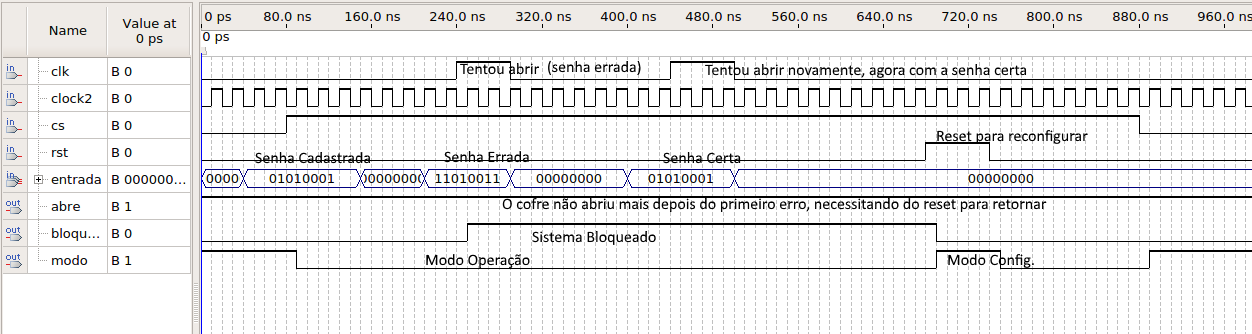


Fig. 08 – Interpretação quando o usuário errou a senha do cofre e tentou inserir novamente, agora com a senha certa.

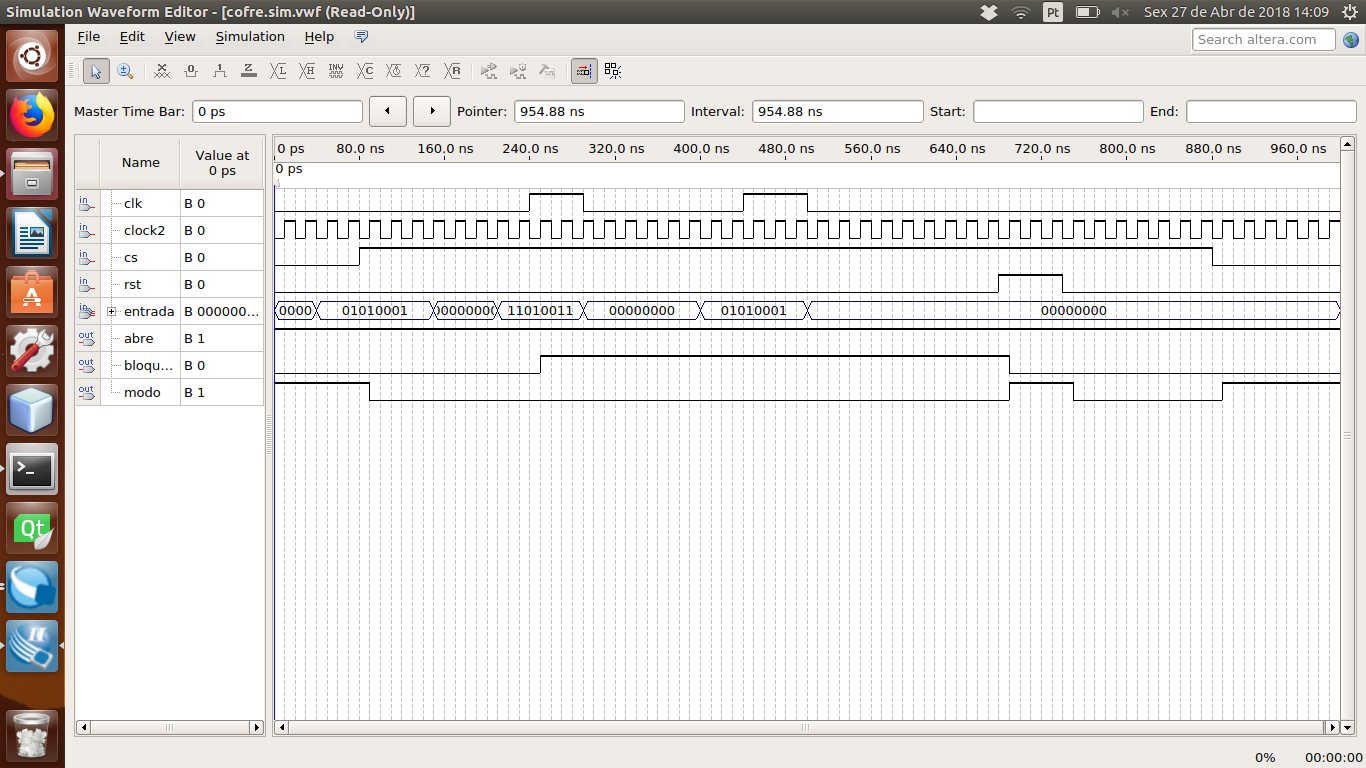


Fig. 09 – Simulação quando o usuário tenta colocar a senha correta após errar a primeira vez.

Em seguida verificamos, a partir do RTL viewer, o funcionamento do circuito combinacional gerado a partir das expressões lógicas usadas no código para a definição dos estados da máquina.

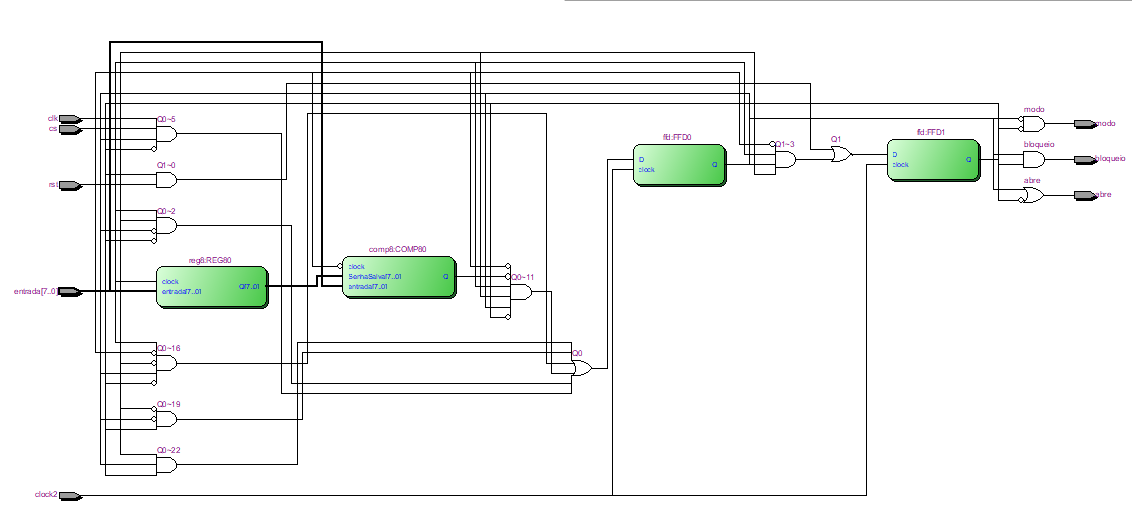


Fig. 10- Circuito no RTL viewer.

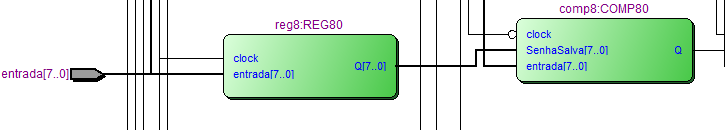


Fig. 11- Registrador e Comparador.

O registrador de 8 bits utilizado salva a entrada no sinal SenhaSalva no momento da borda de subida da entrada CS, uma vez que ela foi usada como clock para o registrador.

Já o comparador tem como entrada dois vetores de 8 bits, a entrada atual e o sinal SenhaSalva, ele retorna 1 se ambas as sequências de bits são idênticas e 0 caso contrário, a comparação ocorre em tempo real, mas ela só é passada a frente quando o clock e ativo. Isto sendo usado como parte da lógica que decide se o sistema vai ser bloqueado ou aberto.

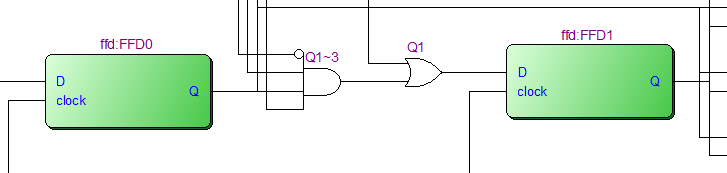


Fig. 12- Flip-Flops de registro de estado.

Os flip-flops acima são usados para realizar a mudança de estado seguindo as expressões a seguir :

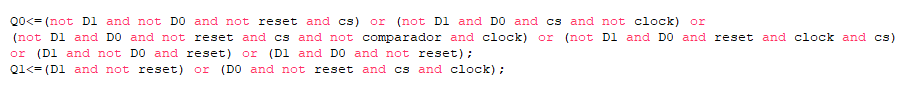


Fig 13- Expressões lógicas de estado.

O flip-flop FF0 registra a mudança de estado de Q0 em D0, utilizando um clock do sistema como seu clock, e o FF1 faz o mesmo para Q1 e D1.

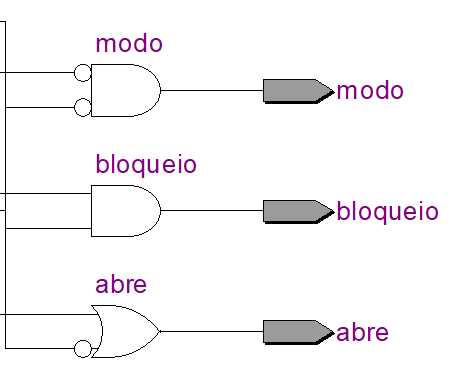


Fig 14- Saídas da máquina de estado.

Para que haja a possibilidade de haver várias tentativas de erro, poderia ser adicionado mais estados, no caso, a quantidade de estados seria o número de erros ou poderia ser implementado dentro do código RTL um sistema de contador interno que limitasse o número de tentativas. Também poderia ser colocado uma senha padrão de fábrica, no caso, o sistema teria, por exemplo, um sistema flip-flops que já tivessem essa senha armazenada, e o comparador teria que verificar tanto a senha atual quanto a senha inserida pelo usuário.

**3 Conclusão**

Concluímos que a partir dos conhecimentos adquiridos em sala de aula já é possível construir um sistema funcional, como o apresentado acima.