

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE GRADUAÇÃO

Relatório de Projeto Grau B:
Elevador de três pavimentos

Aline Nunes de Souza
Davi de Souza Leão Schmitz

São Leopoldo
2021

Aline Nunes de Souza
Davi de Souza Leão Schmitz

Relatório de Projeto Grau B:
Elevador de três pavimentos

Trabalho apresentado para a disciplina
Prototipação Digital, pelo Curso de
Engenharia da Computação da
Universidade do Vale do Rio dos Sinos –
UNISINOS, ministrada pela professora
Bruna Fernandes Flesch.

São Leopoldo
2021

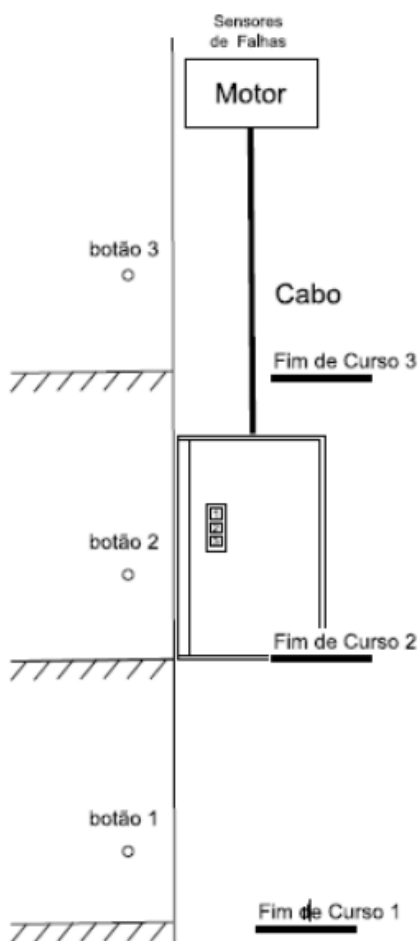
SUMÁRIO

1 INTRODUÇÃO	3
2 PREMISSAS DO PROJETO	4
3 METODOLOGIA	5
4 TEST BENCH	17
CONCLUSÃO	20
REFERÊNCIAS	21

INTRODUÇÃO

O presente trabalho descreve as etapas para o desenvolvimento e implementação, através do uso de máquina de estado de Moore, também chamada de *Finite State Machine* (FSM), e com uso de descrição de HW estrutural na linguagem *VHSIC Hardware Description Language* (VHDL) no *software ISE*, de um elevador de três pavimentos como o proposto na figura 1. Como pré-requisitos, foram previstos alarmes para situações não seguras, como os fim de curso estarem ativados ao mesmo tempo, ou em caso de falha mecânica. Além disso, há indicação de estados da FSM, sentido do motor, estado das portas e condições adicionais para o correto funcionamento do elevador. Para fins de análise e comprovação de funcionamento, há também a descrição de cada operação por simulação no *ISIM* e desenvolvimento de testes por *testbench*.

Figura 1 - Sistema de elevador com três andares



Fonte: FLESCHE, Bruna (2021).

PREMISSAS DO PROJETO

Para o desenvolvimento do projeto do elevador de três pavimentos foram seguidos os seguintes critérios:

- Devem ser previstos alarmes para situações não seguras, como os fim de curso estarem ativados ao mesmo tempo, ou em caso de falha mecânica;
- Devem haver LEDs indicando o estado em que se encontra a FSM;
- Deve haver indicação do sentido do motor (subindo/descendo);
- Deve haver indicação de porta dos andares e do elevador (fechadas/abertas);
- O motor do elevador só deve ligar se as portas dos pavimentos e do elevador estiverem fechadas e se for acionado por comandos localizados nos andares e na parte interna do elevador;
- A abertura de uma porta só deve ser liberada se o elevador estiver naquele pavimento;
- A abertura e o fechamento das portas deve ser manual;
- Necessário prever uso dos seguintes permissivos: um botão de emergência (ativo em nível baixo) e um alarme de incêndio (ativo em nível baixo). Para o funcionamento do sistema, nenhum dos permissíveis podem estar acionados.

METODOLOGIA

Para desenvolver o elevador, primeiramente foram listadas todas as entradas e saídas para em seguida definir os estados da FSM e as condições de transição para a máquina de estados, comportamento frente a situações de mal funcionamento do elevador e comportamento das saídas. A partir desses dados, foi possível desenvolver o diagrama da máquina de estados (FSM) utilizando o editor gráfico online *Draw.io* com o objetivo de auxiliar na descrição do VHDL. Após isso, foi feito o desenvolvimento em VHDL do arquivo TOP LEVEL e módulos auxiliares para controle.

Entradas

Clock, reset, sensor de porta, sensor de incêndio, botão de emergência, fins de curso de cada andar: FC1, FC2 e FC3, botões internos e externos para cada andar.

Saídas

Motor subindo, motor descendo, indicador de cada andar, indicador de subindo e descendo, alarme de emergência e travamento de porta.

Estados

A partir da listagem de entradas e saídas, foi definido que a máquina de estados iria possuir seis estados, sendo eles:

Andar1, Andar2, Andar3, Subindo, Descendo e Emergência.

Condições de transição da Máquina de Moore

Para identificar as transições da FSM, foi definida uma ordem para verificação posterior. A ordem dos estados dos bits para transição pode ser conferida na tabela 1, juntamente com a sua condição para nível lógico alto ou baixo.

Tabela 1 - Bits para transição de estados

Bit	Condição
Sensor de Porta	nível lógico alto se refere a porta travada, fechada
Sensor de incêndio	nível lógico alto quando ele não está ativo
Botão de emergência	nível lógico alto quando ele não está ativo
Fim de curso do andar 1	nível lógico alto quando estiver no primeiro andar
Fim de curso do andar 2	nível lógico alto quando estiver no segundo andar
Fim de curso do andar 3	nível lógico alto quando estiver no terceiro andar
Botão andar 1	nível lógico alto quando alguém do andar dois ou três pressionar o botão para chamar o elevador
Botão andar 2	nível lógico alto quando alguém do andar um ou três pressionar o botão para chamar o elevador
Botão andar 3	nível lógico alto quando alguém do andar um ou dois pressionar o botão para chamar o elevador

Fonte: os autores (2021).

Condição para trocar do Andar1 para o subindo

Para este caso, possuímos duas lógicas combinacionais para determinar o estado do elevador, pois a transição ocorre tanto se o botão do andar dois ou se o botão do andar três estiver com o nível lógico alto. Com isso, podemos representar a lógica combinacional da seguinte forma: **1,1,1,1,0,0,0,1,0 OR 1,1,1,1,0,0,0,0,1**.

- Sensor de Porta deve estar com nível lógico alto (travada, fechada)
- Sensor de incêndio deve estar com nível lógico alto (não ativo)
- Botão de emergência deve estar com nível lógico alto (não ativo)
- Fim de curso do andar 1 deve estar com nível lógico alto
- Fim de curso do andar 2 deve estar com nível lógico baixo

- Fim de curso do andar 3 deve estar com nível lógico baixo
- Botão andar 1 deve estar com nível lógico baixo
- Botão andar 2 deve estar com nível lógico alto OU Botão andar 3 deve estar com nível lógico alto

Condição para trocar do Subindo para o Andar2

Para este caso, também possuímos duas lógicas combinacionais para determinar o estado do elevador, pois a transição ocorre tanto se o botão do andar dois ou se o botão do andar três estiver com o nível lógico alto. Com isso, podemos representar a lógica combinacional da seguinte forma: **1,1,1,0,1,0,0,1,0 OR 1,1,1,0,1,0,0,0,1.**

- Sensor de Porta deve estar com nível lógico alto (travada, fechada)
- Sensor de incêndio deve estar com nível lógico alto (não ativo)
- Botão de emergência deve estar com nível lógico alto (não ativo)
- Fim de curso do andar 1 deve estar com nível lógico baixo
- Fim de curso do andar 2 deve estar com nível lógico alto
- Fim de curso do andar 3 deve estar com nível lógico baixo
- Botão andar 1 deve estar com nível lógico baixo
- Botão andar 2 deve estar com nível lógico alto OU Botão andar 3 deve estar com nível lógico alto

Condição para trocar do Andar2 para o subindo

Para este caso, possuímos uma lógica combinacional para determinar o estado do elevador: 1,1,1,0,1,0,0,0,1.

- Sensor de Porta deve estar com nível lógico alto (travada, fechada)
- Sensor de incêndio deve estar com nível lógico alto (não ativo)
- Botão de emergência deve estar com nível lógico alto (não ativo)
- Fim de curso do andar 1 deve estar com nível lógico baixo
- Fim de curso do andar 2 deve estar com nível lógico alto
- Fim de curso do andar 3 deve estar com nível lógico baixo
- Botão andar 1 deve estar com nível lógico baixo
- Botão andar 2 deve estar com nível lógico baixo
- Botão andar 3 deve estar com nível lógico alto

Condição para trocar do Subindo para o Andar3

Para este caso, possuímos uma lógica combinacional para determinar o estado do elevador: **1,1,1,0,0,1,0,0,1**.

- Sensor de Porta deve estar com nível lógico alto (travada, fechada)
- Sensor de incêndio deve estar com nível lógico alto (não ativo)
- Botão de emergência deve estar com nível lógico alto (não ativo)
- Fim de curso do andar 1 deve estar com nível lógico baixo
- Fim de curso do andar 2 deve estar com nível lógico baixo
- Fim de curso do andar 3 deve estar com nível lógico alto
- Botão andar 1 deve estar com nível lógico baixo
- Botão andar 2 deve estar com nível lógico baixo
- Botão andar 3 deve estar com nível lógico alto

Condição para trocar do Andar3 para o Descida

Para este caso, possuímos duas lógicas combinacionais para determinar o estado do elevador, pois a transição ocorre tanto se o botão do andar um ou se o botão do andar dois estiver com o nível lógico alto. Com isso, podemos representar a lógica combinacional da seguinte forma: **1,1,1,0,0,1,1,0,0 OR 1,1,1,0,0,1,0,1,0**.

- Sensor de Porta deve estar com nível lógico alto (travada, fechada)
- Sensor de incêndio deve estar com nível lógico alto (não ativo)
- Botão de emergência deve estar com nível lógico alto (não ativo)
- Fim de curso do andar 1 deve estar com nível lógico baixo
- Fim de curso do andar 2 deve estar com nível lógico baixo
- Fim de curso do andar 3 deve estar com nível lógico alto
- Botão andar 1 deve estar com nível lógico alto OU Botão andar 2 deve estar com nível lógico alto
- Botão andar 3 deve estar com nível lógico baixo

Condição para trocar do Descendo para o Andar2

- Sensor de Porta deve estar com nível lógico alto (travada, fechada)
- Sensor de incêndio deve estar com nível lógico alto (não ativo)
- Botão de emergência deve estar com nível lógico alto (não ativo)
- Fim de curso do andar 1 deve estar com nível lógico alto
- Fim de curso do andar 2 deve estar com nível lógico baixo

- Fim de curso do andar 3 deve estar com nível lógico baixo
- Botão andar 1 deve estar com nível lógico alto
- Botão andar 2 deve estar com nível lógico baixo
- Botão andar 3 deve estar com nível lógico baixo

Condição para trocar do Andar2 para o Descendo

Para este caso, possuímos uma lógica combinacional para determinar o estado do elevador: **1,1,1,0,1,0,1,0,0**.

- Sensor de Porta deve estar com nível lógico alto (travada, fechada)
- Sensor de incêndio deve estar com nível lógico alto (não ativo)
- Botão de emergência deve estar com nível lógico alto (não ativo)
- Fim de curso do andar 1 deve estar com nível lógico baixo
- Fim de curso do andar 2 deve estar com nível lógico alto
- Fim de curso do andar 3 deve estar com nível lógico baixo
- Botão andar 1 deve estar com nível lógico alto
- Botão andar 2 deve estar com nível lógico baixo
- Botão andar 3 deve estar com nível lógico baixo

Condição para trocar do Descendo para o primeiro andar

Para este caso, possuímos uma lógica combinacional para determinar o estado do elevador: **1,1,1,1,0,0,1,0,0**.

- Sensor de Porta deve estar com nível lógico alto (travada, fechada)
- Sensor de incêndio deve estar com nível lógico alto (não ativo)
- Botão de emergência deve estar com nível lógico alto (não ativo)
- Fim de curso do andar 1 deve estar com nível lógico alto
- Fim de curso do andar 2 deve estar com nível lógico baixo
- Fim de curso do andar 3 deve estar com nível lógico baixo
- Botão andar 1 deve estar com nível lógico alto
- Botão andar 2 deve estar com nível lógico baixo
- Botão andar 3 deve estar com nível lógico baixo

Situações de transição para emergência

- **Fins de curso estarem ativados ao mesmo tempo**

A partir de qualquer estado, se mais de um fim de curso estiver ativo simultaneamente, é feita a transição para o estado de emergência. Esta situação indica que o elevador está em uma posição não segura ou que há uma falha mecânica nos sensores de fim de curso.

- **Sensor de porta em nível baixo**

Para que ocorra qualquer troca de estado o sensor de porta deve estar em nível alto, indicando que esta está fechada. Caso contrário, ocorre a troca de estado para emergência.

- **Sensor de incêndio em nível baixo**

- **Botão de emergência em nível baixo**

Comportamento das saídas dos estados da FSM

A lógica combinacional das saídas da máquina de estado são dadas conforme a tabela 2.

Tabela 2 - Bits para as saídas dos estados da FSM

Bit	Condição
Motor subindo	nível lógico baixo quando ele não está ativo, desligado
Motor descendo	nível lógico baixo quando ele não está ativo, desligado
Indicador Andar 1	nível lógico alto quando estiver no primeiro andar
Indicador Andar 2	nível lógico alto quando estiver no segundo andar
Indicador Andar 3	nível lógico alto quando estiver no terceiro andar
Indicador Subindo	nível lógico alto quando o motor estiver subindo
Indicador Descendo	nível lógico alto quando o motor estiver descendo
Alarme de Emergência	nível lógico alto quando estiver em uma situação que comprometa a segurança
Trava Porta	nível lógico alto por default para garantir a segurança, nenhuma porta deve ser aberta enquanto o motor estiver se movendo

Fonte: os autores (2021).

- **Saída do primeiro andar:**

A saída do primeiro andar respeita a seguinte lógica combinacional: **0,0,1,0,0,0,0,0,0**. O elevador está parado, então os dois motores ficam desligados.

- Motor subindo desligado (nível lógico baixo)
- Motor descendo desligado (nível lógico baixo)
- LED do primeiro andar em nível lógico alto (indicando que está no andar)
- LED do segundo andar em nível lógico baixo
- LED do terceiro andar em nível lógico baixo

- LED indicativo de sentido do motor para subindo em nível lógico baixo (desligado)
- LED indicativo de sentido do motor para descendo em nível lógico baixo (desligado)
- Porta destravada (nível lógico baixo, as portas estão abertas)

- **Saída do segundo andar:**

A saída do segundo andar respeita a seguinte lógica combinacional:

0,0,0,1,0,0,0,0.

- Motor subindo desligado (nível lógico baixo)
- Motor descendo desligado (nível lógico baixo)
- LED do primeiro andar em nível lógico baixo
- LED do segundo andar em nível lógico alto (indicando que está no andar)
- LED do terceiro andar em nível lógico baixo
- LED indicativo de sentido do motor para subindo em nível lógico baixo (desligado)
- LED indicativo de sentido do motor para descendo em nível lógico baixo (desligado)
- Porta destravada (nível lógico baixo, as portas estão abertas)

- **Saída do terceiro andar:**

A saída do terceiro andar respeita a seguinte lógica combinacional:

0,0,0,0,1,0,0,0.

- Motor subindo desligado (nível lógico baixo)
- Motor descendo desligado (nível lógico baixo)
- LED do primeiro andar em nível lógico baixo
- LED do segundo andar em nível lógico baixo
- LED do terceiro andar em nível lógico alto
- LED indicativo de sentido do motor para subindo em nível lógico baixo (desligado)
- LED indicativo de sentido do motor para descendo em nível lógico baixo (desligado)
- Porta destravada (nível lógico baixo, as portas estão abertas)

- **Saída do estado Descendo**

A saída do estado descendo respeita a seguinte lógica combinacional:

0,1,0,0,0,0,1,0,1.

- Motor subindo desligado
- Motor descendo acionado (nível lógico alto indicando que o elevador está descendo)
- LED do primeiro andar em nível lógico baixo
- LED do segundo andar em nível lógico baixo
- LED do terceiro andar em nível lógico baixo
- LED indicativo de sentido do motor para subindo em nível lógico baixo (desligado)
- LED indicativo de sentido do motor para descendo em nível lógico alto (ligado, indicando que o elevador está descendo)
- Porta travada (nível lógico alto para garantir a segurança, nenhuma porta deve ser aberta enquanto o motor estiver se movendo)

- **Saída de Subindo**

A saída do estado subindo respeita a seguinte lógica combinacional:

1,0,0,0,0,1,0,0,1.

- Motor subindo acionado (nível lógico alto indicando que o elevador está subindo)
- Motor descendo desligado
- LED do primeiro andar em nível lógico baixo
- LED do segundo andar em nível lógico baixo
- LED do terceiro andar em nível lógico baixo
- LED indicativo de sentido do motor para subindo em nível lógico alto (ligado, indicando que o elevador está subindo))
- LED indicativo de sentido do motor para descendo em nível lógico alto (desligado)
- Porta travada (nível lógico alto para garantir a segurança, nenhuma porta deve ser aberta enquanto o motor estiver se movendo)

Descrição do VHDL

Como requisito de implementação, no arquivo TOP LEVEL não devem ter processos, mas podem ter instâncias. Logo, no nosso arquivo, nomeado de “elevador”, realizamos o instanciamento de dois módulos VHD criados para o controle do sistema e mapeamento das portas de entrada e saída.

No primeiro módulo, fizemos o controle de estados que decide se pode trocar de um estado para outro, e retornamos um vetor de três bits que representa o estado selecionado, conforme a tabela 2.

Tabela 2 - Saída do módulo “controle_estados”

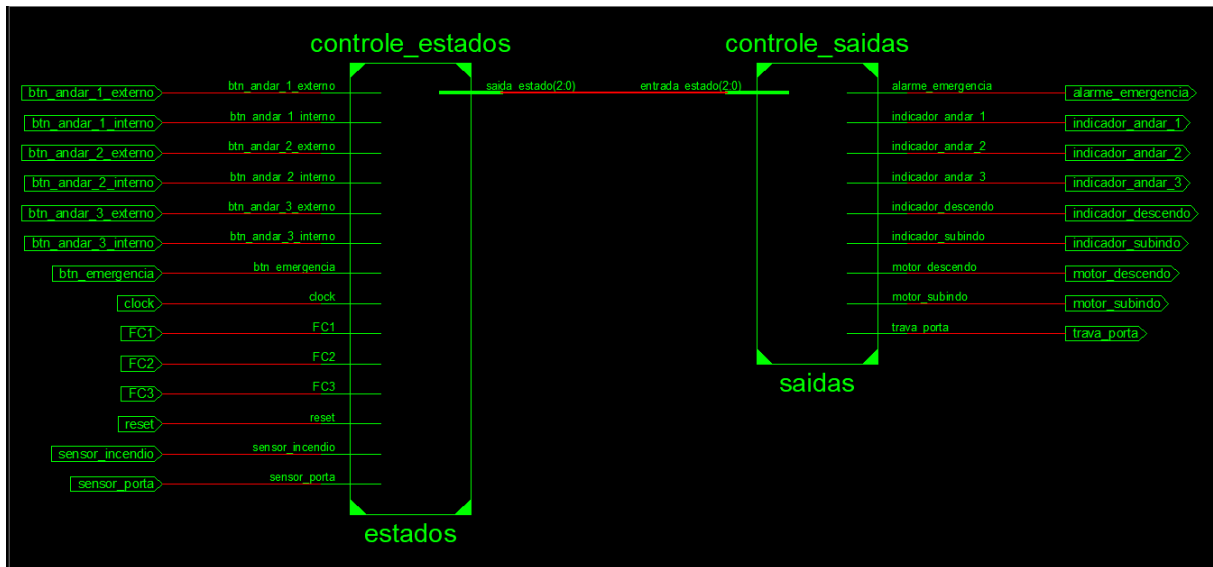
Estado	Valor na saída “saida_estado”
ANDAR1	000
ANDAR2	001
ANDAR3	010
SUBINDO	100
DESCENDO	101
EMERGENCIA	110

Fonte: os autores (2021).

No segundo módulo, fizemos o controle das saídas a partir do vetor de três bits de estados do primeiro módulo. Estas saídas são mapeadas com as saídas do *top level*.

Após a descrição de hardware, foi realizada a sintetização, conforme exibido na Figura 3. É possível visualizar os módulos de controle de estados e saídas, além do sinal utilizado para a comunicação entre módulos.

Figura 3 - Top level da Sintetização de hardware

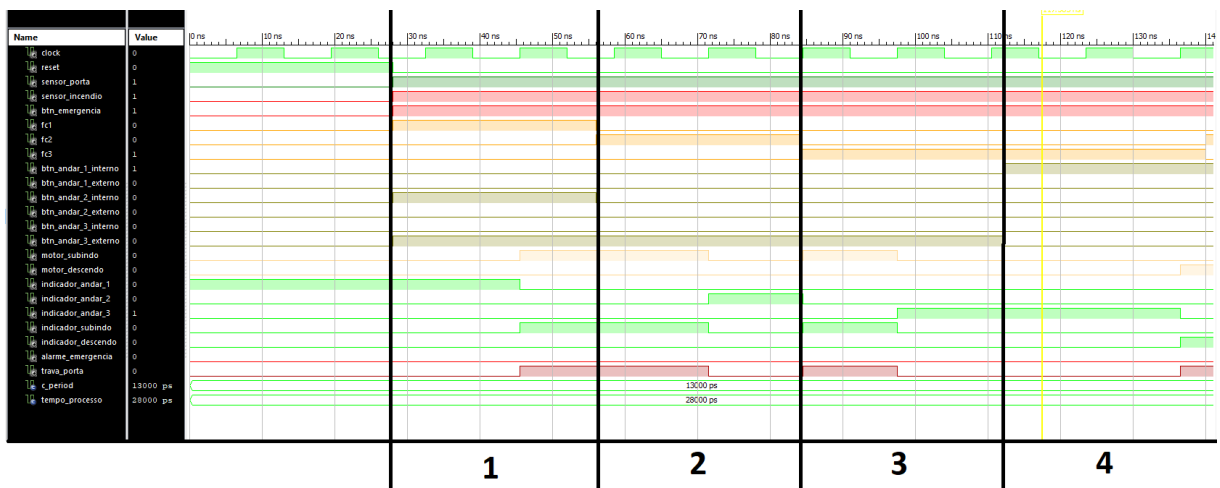


Fonte: os autores (2021).

TEST BENCH

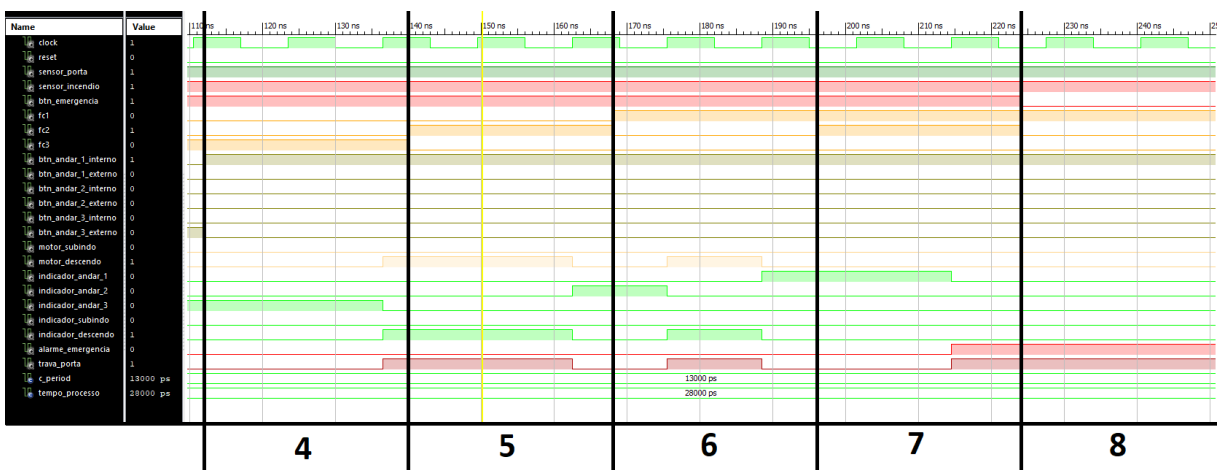
Para comprovar o funcionamento do projeto, foi desenvolvido um testbench abrangente para as funcionalidades do elevador, totalizando oito cenários que serão descritos abaixo. Definimos o período do clock para 13ns e a constante de tempo de espera para 28ns. Conforme indica a figura 3, podemos identificar um tempo de 28ns para entrar no primeiro cenário, pois ativamos o *reset* do sistema para garantir a execução a partir das condições iniciais de funcionamento. Após isso, entramos no primeiro cenário de teste. Após cada cenário, usamos o tempo de espera já citado e pode ser analisado nas figuras 3 e 4.

Figura 3 - Testbench para os cenários 1 até 4



Fonte: os autores (2021).

Figura 4 - Testbench para os cenários 4 até 8



Fonte: os autores (2021).

1) Elevador no primeiro andar, chamado no segundo e terceiro andar

Nesse cenário, conforme pode ser visto indicado pelo “1” em preto na figura 3, corresponde à lógica combinacional do estado 111100001000 OR 111100000001, em que o elevador está no primeiro andar e é chamado no segundo andar por um botão interno e no terceiro andar por um botão externo e com isso entra no estado de subindo pois atende todas as condições para o motor entrar no estado de subindo.

2) Elevador no segundo andar, chamado no terceiro andar

Nesse cenário, conforme pode ser visto indicado pelo “2” em preto na figura 3, corresponde à lógica combinacional do estado 11101000001, em que o elevador está no segundo andar e é chamado no terceiro andar e entra no estado de subindo pois atende todas as condições para o motor entrar no estado de subindo.

3) Elevador subindo para o terceiro andar

Nesse cenário, conforme pode ser visto indicado pelo “3” em preto na figura 3, corresponde à lógica combinacional do estado 111001000001, em que o elevador está subindo para o terceiro andar.

4) Elevador no terceiro andar chamado no primeiro andar

Nesse cenário, conforme pode ser visto indicado pelo “4” em preto na figura 4, corresponde à lógica combinacional do estado 111001100000, em que o elevador está no terceiro andar e é chamado no primeiro andar. Cumprindo todas as condições de segurança, o elevador entra em estado de descida.

5) Elevador no segundo andar chamado no primeiro andar

Nesse cenário, conforme pode ser visto indicado pelo “5” em preto na figura 4, corresponde à lógica combinacional do estado 111010100000, em que o elevador está no segundo andar e é chamado no primeiro andar. Cumprindo todas as condições de segurança, o elevador entra em estado de descida.

6) Elevador descendo para o primeiro andar

Nesse cenário, conforme pode ser visto indicado pelo “6” em preto na figura 4, corresponde à lógica combinacional do estado 111100100000, em que o elevador está descendo para o primeiro andar.

7) Emergência por 2 fins de curso ativos simultaneamente

Nesse cenário é simulado um problema, conforme pode ser visto indicado pelo “7” em preto na figura 4, corresponde à lógica combinacional do estado 111110100000, em que o elevador entra em emergência por ter dois fins de cursos ativos ao mesmo tempo, “FC1” e “FC2”.

8) Botão de emergência pressionado

Nesse cenário é simulado o pressionamento do botão de emergência, conforme pode ser visto indicado pelo “8” em preto na figura 4, corresponde à lógica combinacional do estado 110100100000. Como pode ser visto na simulação do *ISim*, o alarme de emergência aparece ativo e a porta travada.

CONCLUSÃO

Com a realização do presente trabalho, foi possível praticar intensamente o desenvolvimento em VHDL, com o uso de módulos, componentes, mapeamento de portas, utilização de TYPE para os estados da FSM e afins. Além disso, também exercitamos o desenvolvimento de projetos no geral, em que precisamos conversar com um cliente, no presente caso sendo a professora, para atribuir as premissas e objetivos do projeto. A partir das conversas e criação de diagramas, conseguimos realizar um desenvolvimento mais fluido da descrição do VHDL. Com a implementação de testbench e as simulações no *software ISIM*, podemos ver o funcionamento de cada operação com diferentes entradas e comprovar seu funcionamento.

REFERÊNCIAS

D'AMORE, Roberto. **VHDL: descrição e síntese de circuitos digitais**. 2. ed. Rio de Janeiro: LTC, 2012.