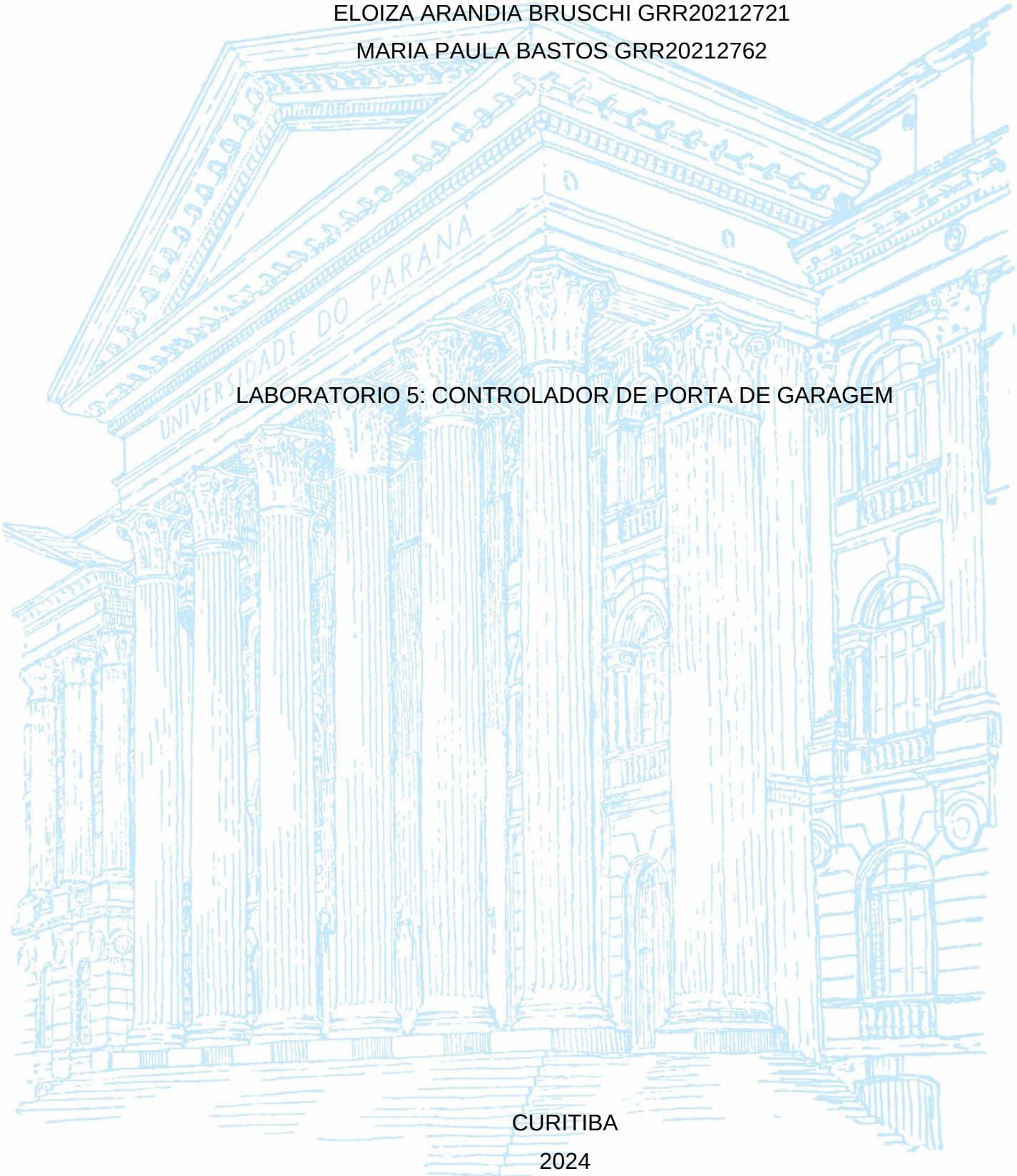
UNIVERSIDADE FEDERAL DO PARANÁ

ELOIZA ARANDIA BRUSCHI GRR20212721
MARIA PAULA BASTOS GRR20212762



LABORATORIO 5: CONTROLADOR DE PORTA DE GARAGEM

CURITIBA
2024

SUMÁRIO

1. INTRODUÇÃO

Este relatório apresenta o desenvolvimento de um controlador de porta de garagem utilizando a abordagem de máquina de estados finitos (FSM - *Finite State Machine*), projetado e implementado em VHDL.

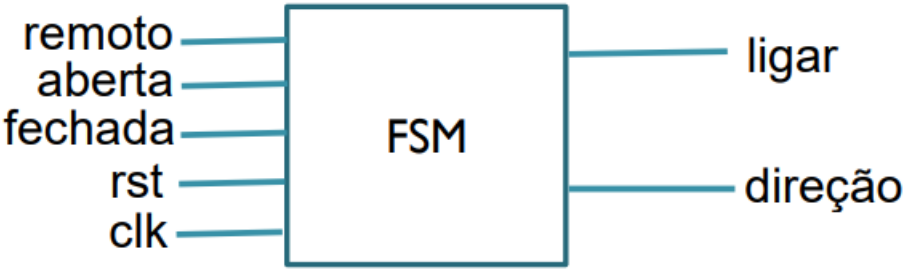
O projeto do controlador foi baseado em um sistema com três entradas principais: um controle remoto, que ativa a operação da porta; um sensor que detecta quando a porta está completamente aberta; e outro sensor que indica quando a porta está totalmente fechada. As saídas incluem um sinal para ligar o motor e outro para definir a direção de movimento, que pode ser de abertura ou fechamento da porta. O funcionamento do sistema foi configurado para obedecer a regras específicas de comportamento, incluindo abertura e fechamento controlados, parada em caso de acionamento do controle remoto durante o movimento, e um timer interno para evitar que a porta permaneça aberta por mais de 30 segundos.

Este relatório detalha o processo de desenvolvimento do controlador, desde o diagrama de transição de estados até a implementação prática no kit Nexys 2. Foram realizadas simulações para validar o comportamento do circuito em diferentes cenários, e o funcionamento foi testado em hardware.

2 PROJETO - Controlador de porta de Garagem

A Figura 1 mostra um controlador de porta de garagem. Este controlador tem, além do clock e reset, outras três entradas, denominadas remoto (= ‘1’ quando o controle remoto é acionado), aberta (= ‘1’ quando a porta estiver completamente aberta, fornecido por um sensor), fechada (= ‘1’ quando a porta estiver completamente fechada, também fornecido por um sensor). Na saída, os seguintes sinais são produzidos: ligar (quando ‘1’, liga o motor elétrico) e direção (quando ‘0’ gira na direção de abrir a porta; quando ‘1’ gira na direção de fechá-la).

FIGURA 1 – ESQUEMÁTICO CONCEITUAL DO PROJETO



FONTE: ‘LAB 5’, Sibilla França.

O sistema apresenta as seguintes características:

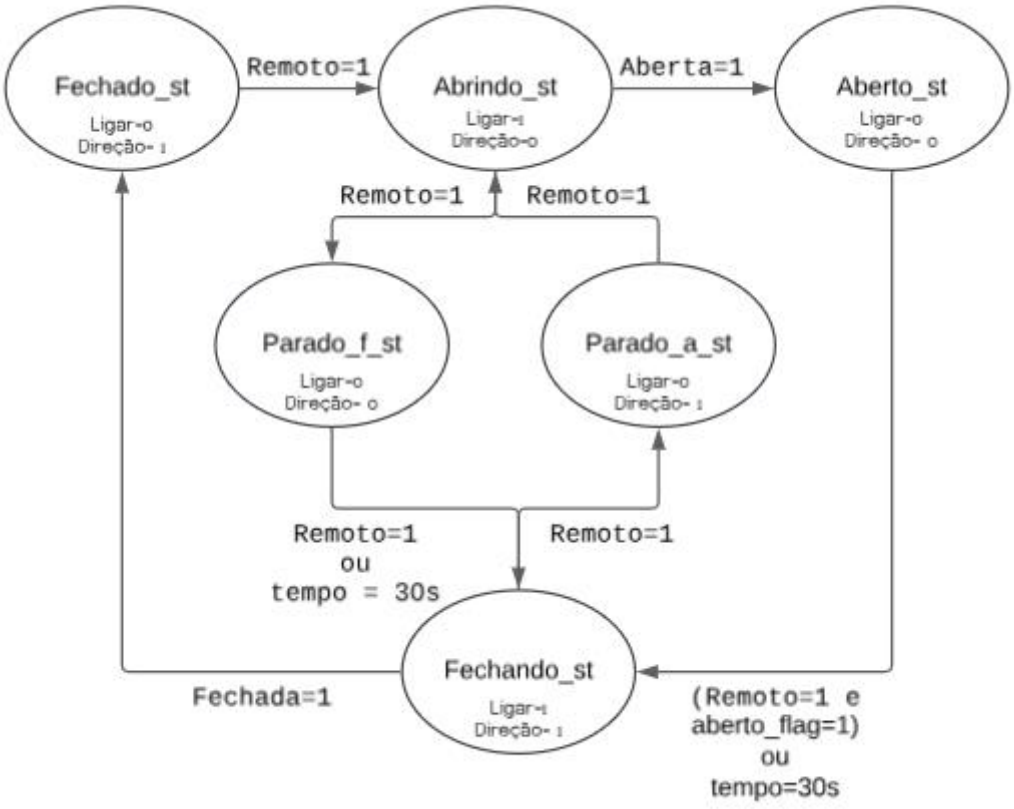
I. Se o controle remoto é acionado enquanto a porta estiver fechada, o motor é ligado imediatamente para abri-la;

II. Se o controle remoto é acionado enquanto a porta estiver aberta, o motor é ligado imediatamente para fechá-la;

III. Se o controle remoto é acionado enquanto a porta estiver abrindo ou fechando, a porta para imediatamente. Se o controle remoto é acionado novamente, a porta vai na direção oposta àquela em que estava indo;

IV. A porta não permanece aberta mais do que 30 segundos. Isto deve ser controlado por um timer interno (que deve assumir o valor igual a ‘1’ após 30 segundos de abertura completa da porta (aberta= ‘1’) ou 30 segundos após ter ocorrido uma parada (item III).

- **Diagrama de Transição de Estados**



Ele contém os seguintes estados:

- fechado_st: Porta totalmente fechada.
- abrindo_st: Porta em movimento de abertura.
- aberto_st: Porta completamente aberta.
- fechando_st: Porta em movimento de fechamento.
- parado_a_st: Porta parada durante a abertura.
- parado_f_st: Porta parada durante o fechamento.

Cada estado precisa ser representado de alguma forma interna para que o sistema possa saber em que condição está. Normalmente, usamos **Flip-Flops** para armazenar o estado atual da FSM.

Cálculo do número de bits necessários para representar os estados:

Para codificar **6 estados** diferentes no tipo de codificação binária sequencial, precisamos de uma quantidade mínima de bits n tal que: $2n \geq 6$. ou seja, precisamos de **3 bits** para representar 6 estados, pois $2\text{^}3 = 8$ estados possíveis (que é mais do que suficiente para cobrir os 6 estados definidos). A FSM será codificada

usando **3 bits**, e cada bit será armazenado em um **Flip-Flop (DFF)**. Portanto, serão necessários **3 DFFs** para armazenar o estado atual da FSM.

A Frequência do Clock

A frequência do clock, embora afete o tempo de operação do sistema (quanto mais alto o clock, mais rápido o circuito opera), **não afeta o número de DFFs** necessários. O número de Flip-Flops depende diretamente da quantidade de bits necessários para representar os estados da FSM e não da frequência do clock.

2. 1 SIMULAÇÕES

Neste projeto, a simulação foi desenvolvida utilizando um testbench em VHDL, que emula o comportamento das entradas e monitora as saídas do controlador em diversos cenários. O foco principal foi validar a lógica da máquina de estados finitos (FSM), incluindo o acionamento do motor, a definição da direção de movimento e a resposta aos sensores que indicam os estados da porta (aberta ou fechada). Além disso, a simulação abrangeu situações específicas, como interrupções no movimento da porta, acionamentos repetidos do controle remoto e a funcionalidade do timer interno que limita o tempo de permanência da porta aberta. No testbench foi modificado os valores das variáveis generics para fazer o timer e o o deboucing funcionar com poucos ciclos de clock As mudanças foram *fclk* para 1 (na placa real é 50000), *twindow* para 1 (real: 10, por cauas dos 10 ms do deboucing), *freq* para 1 (real: 50.000.000) e para *tempo_max_1* para 200 (real: 30).

Para tornar a simulação mais flexíve, foi implementada uma estrutura condicional `if` que permitiu configurar diferentes cenários de teste dentro de um único processo. Essa abordagem facilitou a análise do controlador em situações específicas e complexas, sem a necessidade de alterar o código principal repetidamente. Assim como para melhor visualização foram colocadas na visualização da simulação as variáveis ‘pr_state’ e ‘nx_state’ para melhor análise do comportamento da FSM.

Simulação 0: porta fechada e abertura inicial

O primeiro simulado foi o acionamento inicial do controle remoto para abrir a porta, ou seja, passando do estado fechado para o abrindo após a variável remoto ir para 1. E na Figura 3, inclui o funcionamento do timer interno e a transição para o

estado de fechamento, mesmo sem o remoto ser clicado e para o fechado, após ser clicado.

FIGURA 2 – SIMULAÇÃO DO PROJETO

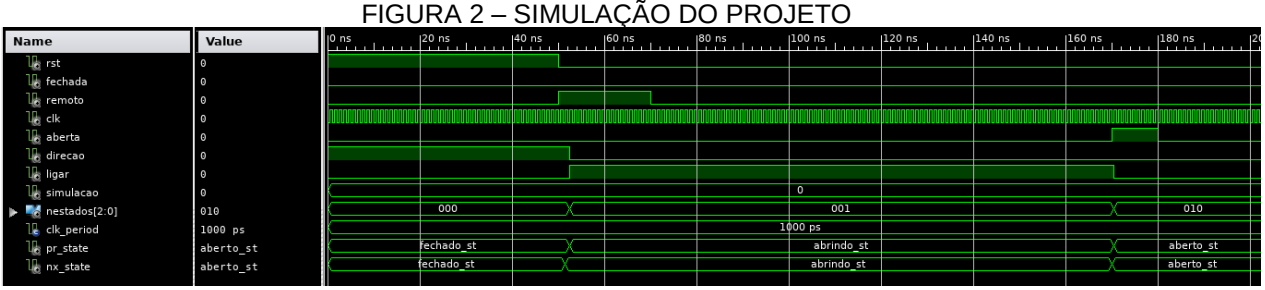
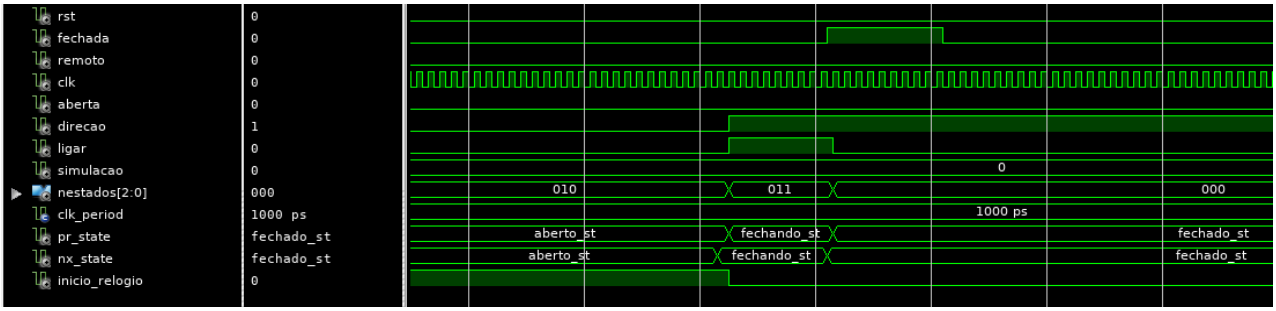


FIGURA 3 – SIMULAÇÃO DO PROJETO

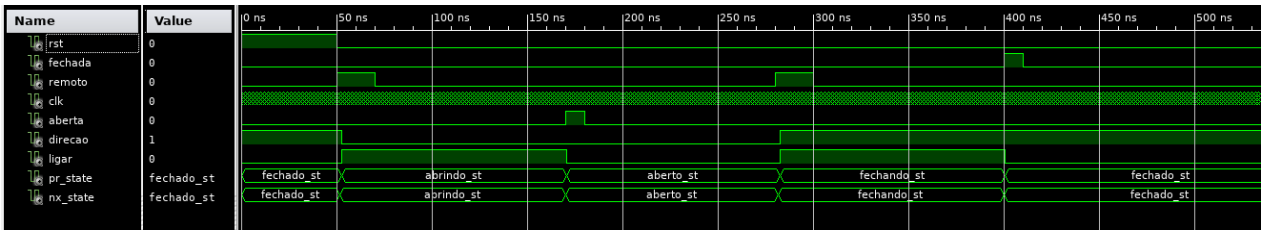


FONTE: As autoras (2024).

Simulação 1: Abertura com Interrupção e Fechamento

O segundo cenário avaliou a interrupção do movimento da porta devido ao acionamento do controle remoto durante a abertura, com posterior inversão de direção.

FIGURA 4 – SIMULAÇÃO 2 DO PROJETO



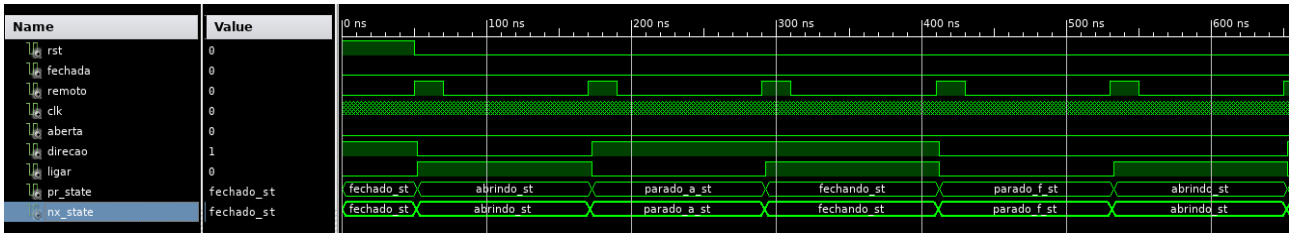
FONTE: As autoras (2024).

Simulação 2: Repetição de Comandos para Abertura e Fechamento

O terceiro teste verificou o comportamento do sistema sob múltiplos acionamentos consecutivos do controle remoto, validando a estabilidade da lógica de transição de estados, principalmente do ciclo abrindo, depois parado, depois

fechando, depois parado, sem dar tempo do temporizador atuar, ou qualquer outra variável do sistema.

FIGURA 5 – SIMULAÇÃO 2 DO PROJETO

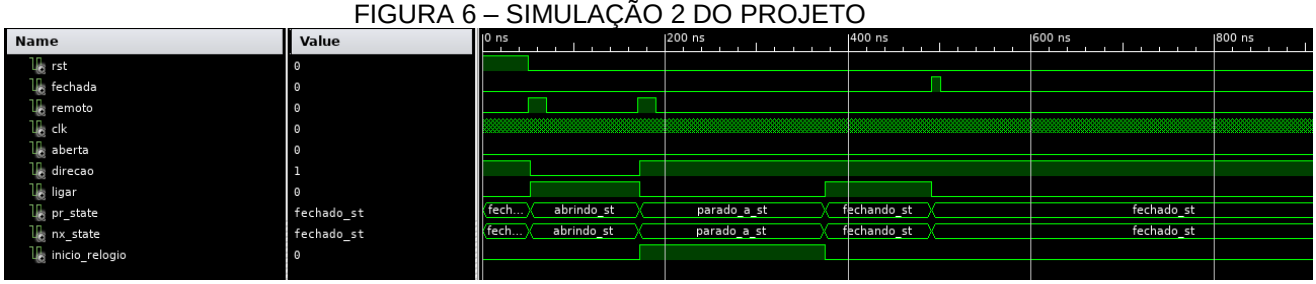


FONTE: As autoras (2024).

Simulação 3: Abertura com Tempo e Fechamento

O último cenário analisou a funcionalidade do temporizador interno ao manter a porta aberta por um período predeterminado antes de acionar o fechamento automático, ou seja, passou do estado '*parado_a_st'* para fechando apenas com o funcionamento do temporizador.

FIGURA 6 – SIMULAÇÃO 2 DO PROJETO



FONTE: As autoras (2024).

2.2 CÓDIGO COMENTADO

O código comentado é dividido em duas partes, no código de fato da placa física e no código para simular, com as mudanças que seriam feitas se fosse possível testar na placa.

2.2.1VHDL Module

O código VHDL Module, do projeto, está apresentado sem alterações no Anexo 1 deste presente trabalho, implementa um

2.2.2 VHDL Test Bench

O código VHDL Test Bench, do projeto, está apresentado sem alterações no Anexo 2 deste presente trabalho.

RECURSOS LÓGICOS

Os recursos lógicos resumem a utilização do dispositivo e os recursos do projeto estão apresentados abaixo.

FIGURA 7 – RECURSOS LÓGICOS PROJETO

Device Utilization Summary				[-]
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Total Number Slice Registers	57	9,312	1%	
Number used as Flip Flops	56			
Number used as Latches	1			
Number of 4 input LUTs	69	9,312	1%	
Number of occupied Slices	75	4,656	1%	
Number of Slices containing only related logic	75	75	100%	
Number of Slices containing unrelated logic	0	75	0%	
Total Number of 4 input LUTs	141	9,312	1%	
Number used as logic	69			
Number used as a route-thru	72			
Number of bonded IOBs	10	232	4%	
Number of BUFGMUXs	1	24	4%	
Average Fanout of Non-Clock Nets	2.20			

FONTE: As autoras (2024).

CONCLUSÃO

O desenvolvimento do controlador de porta de garagem, utilizando a abordagem de máquina de estados finitos (FSM) e implementado em VHDL. Foi possível explorar diversos aspectos teóricos e práticos do uso de FSMs em sistemas digitais, como a codificação de estados, a utilização de Flip-Flops para o armazenamento dos estados e a interação entre entradas e saídas do sistema.

O projeto foi desenvolvido com base em três entradas principais: o controle remoto, os sensores de abertura e fechamento da porta, e foi projetado para operar com saídas que controlam o motor da porta, incluindo a direção de movimento. Além disso, o sistema foi configurado com regras específicas para garantir o funcionamento correto da porta, como a limitação do tempo de abertura e a capacidade de interromper ou inverter o movimento da porta quando o controle remoto é acionado durante a operação.

As simulações realizadas foram essenciais para validar a lógica do controlador, cobrindo diversos cenários operacionais, como a abertura inicial, interrupções no movimento, acionamentos repetidos do controle remoto e o controle automático de tempo para evitar que a porta ficasse aberta por mais de 30 segundos. A implementação da estrutura condicional `if` no testbench permitiu explorar diferentes cenários de maneira eficiente, sem a necessidade de alterações constantes no código principal.

Por fim, o teste no kit Nexys 2 confirmou o funcionamento prático do sistema, e a análise dos recursos lógicos utilizados proporcionou uma visão clara do desempenho do dispositivo em termos de eficiência e otimização. Este trabalho, portanto, não apenas atingiu o objetivo de criar um sistema funcional para o controle de uma porta de garagem, mas também contribuiu para a compreensão e aplicação de conceitos importantes de sistemas digitais e máquinas de estados finitos em VHDL.

REFERÊNCIAS

PEDRONI, V. A. Circuit Design with VHDL. 2 ed. MIT Press, 2010.