



**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE**  
**DCA - CT | DCA0119 - SISTEMAS DIGITAIS**

**RELATÓRIO TÉCNICO:**  
**CONTROLE DE SISTEMA DE ILUMINAÇÃO RESIDENCIAL**

**Professor:** Jose Kleiton Ewerton Da Costa Martins

**Componente<sup>1</sup>:** Ernane Ferreira Rocha Junior - **Matrícula:** ..

**Componente<sup>2</sup>:** Quelita Miriam Nunes Ferraz - **Matrícula:** ..

**Componente<sup>3</sup>:** Thiago Lopes do Nascimento - **Matrícula:** ..

Natal/RN  
Dezembro de 2023

## **SUMÁRIO**

1. Introdução .....	3
2. Fundamentação teórica .....	3
3. Metodologia .....	
4. Resultado .....	13
5. Conclusão .....	14
6. Referências .....	16
7. Anexos .....	17

## **1. INTRODUÇÃO**

Este projeto visa à implementação de um sistema inovador de controle de iluminação residencial, destacando-se pela aplicação estratégica de Field-Programmable Gate Arrays (FPGAs). A proposta centraliza-se na oferta de um controle de iluminação preciso e eficiente, proporcionando aos usuários uma experiência personalizada e adaptável.

A utilização estratégica de FPGAs permite a integração de diversos modos de configuração, conferindo ao sistema uma notável flexibilidade e adaptabilidade. Ao simples toque de um botão, os usuários têm a capacidade de ajustar a intensidade da luz, criando uma atmosfera personalizada para diferentes atividades e preferências. Destaca-se ainda a inclusão de um modo noturno, ativado por uma chave seletora, que automaticamente suaviza a iluminação durante a noite, visando o conforto dos moradores e a eficiência energética.

O FPGA simplifica o controle individual das luzes em cada cômodo, permitindo ajustes precisos de iluminação em diferentes áreas da residência. A ativação de luzes por meio de botões ou switches específicos oferece conveniência, enquanto uma chave geral proporciona um controle centralizado para ligar ou desligar todas as luzes da residência.

A flexibilidade intrínseca dos FPGAs permite a reconfiguração fácil dessas funcionalidades, garantindo adaptabilidade a futuras modificações no layout da casa ou às mudanças nas preferências dos moradores. Este sistema digital, fundamentado em FPGAs, unifica eficiência energética, automação personalizada e facilidade de uso, visando criar um ambiente residencial inteligente e confortável. A escolha de Field-Programmable Gate Arrays (FPGAs) neste projeto de controle de iluminação residencial oferece uma solução flexível e eficiente.

## **2. FUNDAMENTAÇÃO TEÓRICA**

A fundamentação teórica deste projeto aborda de forma abrangente diversos elementos essenciais para o desenvolvimento bem-sucedido de um sistema de iluminação residencial com FPGA, utilizando o Quartus II Web Edition 13.0sp1 para simulação e implementação dos códigos<sup>1</sup>.

O Quartus II, uma suíte de software de design digital da Intel, desempenha um papel fundamental neste projeto, fornecendo uma plataforma abrangente para a concepção, simulação e implementação de lógica programável em FPGAs. Esta ferramenta oferece uma gama de recursos, incluindo ferramentas de design gráfico e suporte à linguagem de descrição de hardware VHDL para especificação do comportamento desejado do circuito digital<sup>2</sup>.

O VHDL (VHSIC Hardware Description Language) é utilizado como linguagem de descrição de hardware, sendo crucial para descrever o comportamento funcional e estrutural do sistema. Esta linguagem permite a modelagem do projeto de forma modular e hierárquica, facilitando a implementação eficiente de sistemas digitais complexos<sup>3</sup>.

O FPGA Cyclone II, da Altera, é o componente central neste projeto. A conectividade entre o Quartus II e o FPGA é crítica para a programação e configuração da lógica reconfigurável do FPGA com os códigos gerados. Essa integração é vital para o desempenho adequado do sistema de controle de iluminação residencial<sup>4</sup>.

Simulações desempenham um papel crucial na validação do projeto. O Quartus II Web Edition 13.0sp1 oferece ferramentas robustas para simulações, permitindo aos desenvolvedores testar e verificar o comportamento do sistema antes da implementação física. Essa fase é essencial para garantir a correção e eficácia do código VHDL desenvolvido<sup>5</sup>.

O manual do kit de desenvolvimento DE2, fornecido pela Terasic, desempenha um papel crucial na fundamentação teórica do projeto. Este manual fornece informações detalhadas sobre a arquitetura específica do FPGA Cyclone II, incluindo pinagens e interfaces disponíveis<sup>6</sup>. O conhecimento dessas informações é essencial para a correta alocação e roteamento das conexões entre o FPGA e os componentes externos.

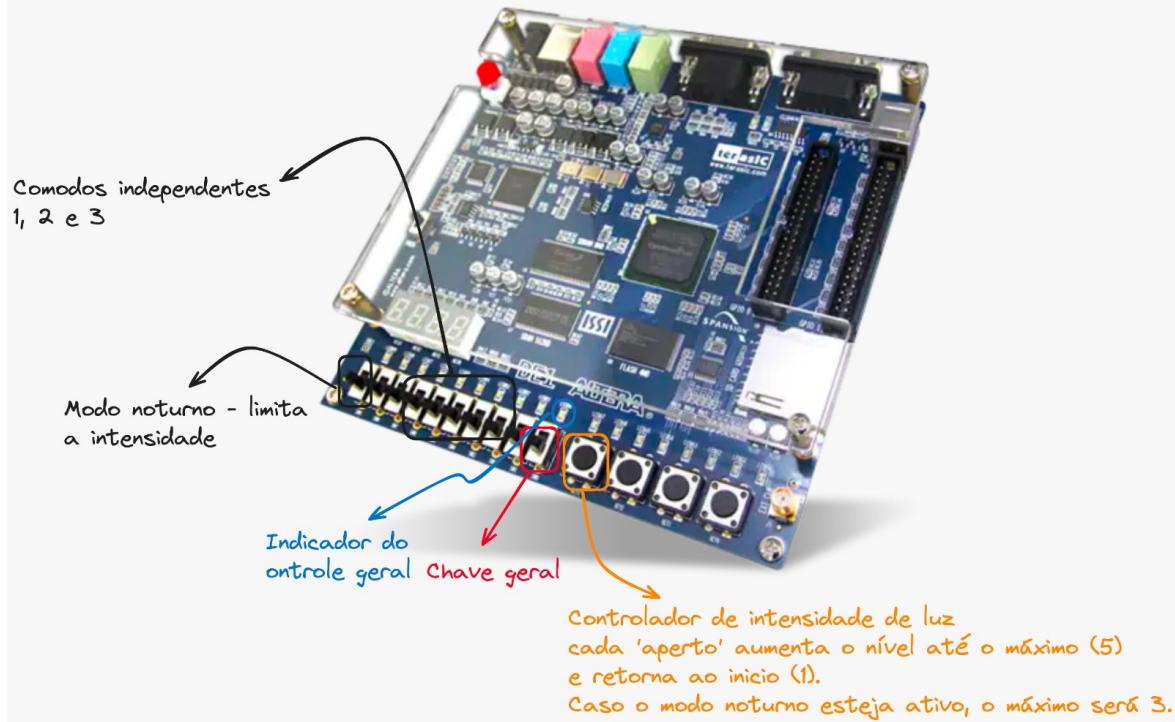
Ao abordar a construção de código VHDL, o uso do Quartus II, a importância do FPGA Cyclone II, e a realização de simulações, a fundamentação teórica proporciona uma base sólida para o desenvolvimento prático do sistema de controle de iluminação residencial. Este conhecimento é vital para assegurar uma integração precisa entre o FPGA e os periféricos associados, garantindo o sucesso do projeto.

### **3. METODOLOGIA**

O desenvolvimento do sistema de controle de iluminação residencial com FPGA é um processo complexo e estruturado que envolve a integração de hardware e software. A metodologia adotada abrange desde a concepção do projeto até a definição das funcionalidades específicas.

## Projeto de Sistemas Digitais - Sistema de Iluminação Residencial

1. Acender / Ligar luzes a partir de um botão/switch;
2. Controle de intensidade a partir de um push\_button;
3. Modo noturno por chave seletora;
4. Controle individual dos cômodos;
5. Chave geral da residência.



**Figure 1:** Representação das funcionalidades e utilização direta na FPGA

### 3.1. Estruturação do Projeto

#### 3.1.1. Definição de Requisitos

Identificação das funcionalidades desejadas, incluindo acionamento individual de luzes, controle de intensidade, modo noturno, controle por cômodo e chave geral da residência.

#### 3.1.2. Seleção de Ferramentas

Escolha do Quartus II Web Edition 13.0sp1 como ambiente de desenvolvimento, considerando suas capacidades abrangentes para design digital e suporte a FPGAs.

#### 3.1.3. Modelagem da Arquitetura

Estruturação do projeto com base em uma máquina de estados, proporcionando uma abordagem organizada e modular para a implementação do controle de iluminação.

### 3.2. Definição das Funcionalidades:

### **3.2.1. Acender / Ligar luzes a partir de um botão/switch**

Utilização de um switch para ligar e desligar a luz principal e switches individuais para cada cômodo.

### **3.2.2. Controle de intensidade a partir de um push\_button**

Implementação de um push\_button para controlar a intensidade da luz, permitindo ajustes progressivos.

### **3.2.3. Modo noturno por chave seletora**

Integração de uma chave seletora para ativar o modo noturno, ajustando automaticamente a iluminação para uma intensidade suave durante a noite.

### **3.2.4. Controle individual dos cômodos**

Configuração para permitir o controle independente da iluminação em cada cômodo da residência.

### **3.2.5. Chave geral da residência**

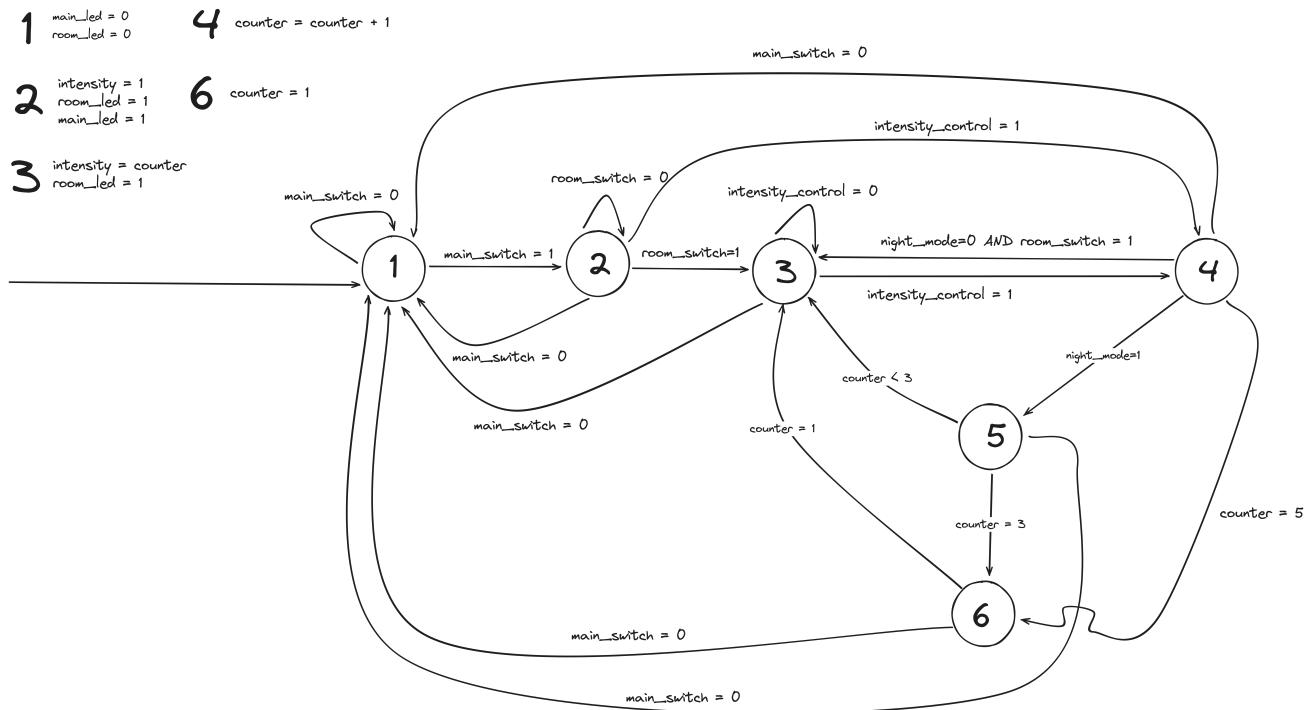
Implementação de uma chave geral para ligar e desligar todas as luzes da residência de forma centralizada.

### **3.2.6. Display de 7 segmentos**

A visualização da intensidade da luz é demonstrada por LEDs e por meio do display de 7 segmentos.

## **3.3. Estruturação do Projeto em Máquina de Estados:**

A metodologia adota uma abordagem baseada em máquina de estados para estruturar o projeto, proporcionando uma representação visual clara das transições de estados e das ações associadas a cada estado. A imagem abaixo ilustra a estrutura da máquina de estados:



**Figure 2:** Máquina de estados

A representação acima ilustra os estados da máquina e suas respectivas transições. Conforme evidenciado na imagem, no canto superior esquerdo, encontram-se os estados das variáveis empregadas na elaboração e controle da máquina. Imediatamente ao lado, apresenta-se sua representação gráfica, juntamente com as transições esperadas.

No primeiro estado, ocorre a inicialização da máquina, onde o controlador encontra-se desligado, não exercendo influência sobre o sistema de iluminação da residência. No segundo estado, dá-se início à utilização do controlador, revelando as possíveis direções a serem seguidas pela máquina: o controle individual dos cômodos e/ou a gestão da intensidade do sistema. No terceiro estado, efetivamente estamos regulando a intensidade do sistema e sua iluminação, enquanto no quarto estado, temos a estrutura literal para o controle do contador de intensidade.

Conforme observado, há a capacidade de contar até 3, com uma intensidade reduzida quando o modo noturno está ativo, e até 5, com sua intensidade máxima quando não está ativo. Em ambas as situações, o processo do contador ocorre em uma estrutura circular, reiniciando o estado do contador para sua configuração padrão (intensidade mínima de valor unitário) ao ultrapassar os limites, seja inferior ou superior.

A partir desse ponto, a máquina segue permitindo o controle não apenas da intensidade, mas também de cada cômodo de forma independente. Vale ressaltar a transição canônica para todos os estados, pois são diretamente dependentes da chave geral, denominada *main\_switch*. Assim, a máquina e suas funcionalidades retornam ao estado inicial ou prosseguem com suas transições apenas por dependência dessa variável controladora.

### 3.4. Construção do Código em VHDL:

O código VHDL a seguir representa a implementação do sistema de controle de iluminação residencial. O projeto é estruturado com base em sinais de entrada e saída que correspondem aos componentes do sistema, como LEDs, switches e push\_buttons. A lógica de controle é definida por meio de um processo, considerando os estados e transições da máquina de estados.

No código VHDL desenvolvido, as portas de entrada e saída foram associadas a pinagens específicas da FPGA, estabelecendo uma ligação direta entre o design lógico e os recursos físicos do dispositivo. Essas pinagens são essenciais para a integração, permitindo a entrada de sinais externos e a saída de sinais processados, pois facilitam a eficiente transferência de informações entre o código VHDL e o hardware real da FPGA. As tabelas detalhadas com as atribuições específicas das pinagens podem ser encontradas nos anexos, proporcionando uma referência clara para a configuração física das interfaces do circuito na FPGA.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- Definição da entidade
entity homeControll is
    Port (
        main_led : OUT STD_LOGIC; -- Luz da casa (LED)
        main_swich : IN STD_LOGIC; -- swich para a luz da casa
        (swich)

        room_1_led : OUT STD_LOGIC; -- Luz do quarto 1 (LED)
        room_1_swich : IN STD_LOGIC; -- swich para a luz do
        quarto 1 (swich)

        room_2_led : OUT STD_LOGIC; -- Luz do quarto 2 (LED)
        room_2_swich : IN STD_LOGIC; -- swich para a luz do
        quarto 2 (swich)

        room_3_led : OUT STD_LOGIC; -- Luz do quarto 3 (LED)
        room_3_swich : IN STD_LOGIC; -- swich para a luz do
        quarto 3 (swich)
```

```

        intensity_control : IN STD_LOGIC; -- controlador de
intensidade (push_button)
        intensity_1 : OUT STD_LOGIC; -- Marcador de intensidade 1
(LED)
        intensity_2 : OUT STD_LOGIC; -- Marcador de intensidade 2
(LED)
        intensity_3 : OUT STD_LOGIC; -- Marcador de intensidade 3
(LED)
        intensity_4 : OUT STD_LOGIC; -- Marcador de intensidade 4
(LED)
        intensity_5 : OUT STD_LOGIC; -- Marcador de intensidade 5
(LED)

        timer_switch : IN STD_LOGIC; -- switch para controle do
timer (swich)
        timer_led : OUT STD_LOGIC; -- LED de saída

night_mode : IN STD_LOGIC; -- Modo noturno (swich)

-- Visualização no display de 7 segmentos:
hex_0_a : OUT STD_LOGIC;
hex_0_b : OUT STD_LOGIC;
hex_0_c : OUT STD_LOGIC;
hex_0_d : OUT STD_LOGIC;
hex_0_e : OUT STD_LOGIC;
hex_0_f : OUT STD_LOGIC;
hex_0_g : OUT STD_LOGIC
);
end homeControll;

-- Arquitetura
architecture Behavioral of homeControll is
    signal counter : STD_LOGIC_VECTOR(2 downto 0) := "001"; --
Inicia o contador em 1
begin
    process(main_swich, room_1_swich, room_2_swich, room_3_swich,
intensity_control, timer_switch) is
    begin

```

```

        if main_swich = '1' then -- Switch ON (assumindo que '1'
é nível alto)
            main_led <= '1'; -- Acende o LED
            intensity_1 <= '1';

        if room_1_swich = '1' then
            room_1_led <= '1';
        else
            room_1_led <= '0';
        end if;

        if room_2_swich = '1' then
            room_2_led <= '1';
        else
            room_2_led <= '0';
        end if;

        if room_3_swich = '1' then
            room_3_led <= '1';
        else
            room_3_led <= '0';
        end if;

-- intensity
        if (intensity_control'EVENT and intensity_control = '0')
then -- Push button ON (assumindo que '1' é nível alto)
        counter <= counter + "001"; -- Incrementa o contador

        if night_mode = '1' AND counter = "011" then
            counter <= "001";
        end if;

        if counter = "001" then -- Se o contador for igual a 2
(assumindo que '010' é 2)
            intensity_1 <= '1'; -- Acende o LED
            intensity_2 <= '0'; -- Desliga o LED
            intensity_3 <= '0'; -- Desliga o LED
            intensity_4 <= '0'; -- Desliga o LED

```

```

intensity_5 <= '0'; -- Desliga o LED

-- Visualização no display da intensidade 1
hex_0_a <= '1';
hex_0_b <= '0';
hex_0_c <= '0';
hex_0_d <= '1';
hex_0_e <= '1';
hex_0_f <= '1';
hex_0_g <= '1';

elsif counter = "010" then -- Se o contador for igual a
2 (assumindo que '010' é 2)
    intensity_1 <= '1'; -- Acende o LED
    intensity_2 <= '1'; -- Acende o LED
    intensity_3 <= '0'; -- Desliga o LED
    intensity_4 <= '0'; -- Desliga o LED
    intensity_5 <= '0'; -- Desliga o LED

-- Visualização no display da intensidade 2
hex_0_a <= '0';
hex_0_b <= '0';
hex_0_c <= '1';
hex_0_d <= '0';
hex_0_e <= '0';
hex_0_f <= '1';
hex_0_g <= '0';

elsif counter = "011" then -- Se o contador for igual a
3 (assumindo que '011' é 3)
    intensity_1 <= '1'; -- Acende o LED
    intensity_2 <= '1'; -- Acende o LED
    intensity_3 <= '1'; -- Acende o LED
    intensity_4 <= '0'; -- Desliga o LED
    intensity_5 <= '0'; -- Desliga o LED

-- Visualização no display da intensidade 3
hex_0_a <= '0';
hex_0_b <= '0';
hex_0_c <= '0';

```

```

    hex_0_d <= '0';
    hex_0_e <= '1';
    hex_0_f <= '1';
    hex_0_g <= '0';
    elsif counter = "100" then -- Se o contador for igual a
4 (assumindo que '100' é 4)
    intensity_1 <= '1'; -- Acende o LED
    intensity_2 <= '1'; -- Acende o LED
    intensity_3 <= '1'; -- Acende o LED
    intensity_4 <= '1'; -- Acende o LED
    intensity_5 <= '0'; -- Desliga o LED

    -- Visualização no display da itensidade 4
    hex_0_a <= '1';
    hex_0_b <= '0';
    hex_0_c <= '0';
    hex_0_d <= '1';
    hex_0_e <= '1';
    hex_0_f <= '0';
    hex_0_g <= '0';
else -- Se o contador for igual a 5
    counter <= "001";
    intensity_1 <= '1'; -- Acende o LED
    intensity_2 <= '1'; -- Acende o LED
    intensity_3 <= '1'; -- Acende o LED
    intensity_4 <= '1'; -- Acende o LED
    intensity_5 <= '1'; -- Acende o LED

    -- Visualização no display da itensidade 5
    hex_0_a <= '0';
    hex_0_b <= '1';
    hex_0_c <= '0';
    hex_0_d <= '0';
    hex_0_e <= '1';
    hex_0_f <= '0';
    hex_0_g <= '0';
end if;
end if;

```

```

    else
        main_led <= '0'; -- Desliga o LED
        intensity_1 <= '0'; -- Desliga o LED
        intensity_2 <= '0'; -- Desliga o LED
        intensity_3 <= '0'; -- Desliga o LED
        intensity_4 <= '0'; -- Desliga o LED
        intensity_5 <= '0'; -- Desliga o LED

        room_1_led <= '0';
        room_2_led <= '0';
        room_3_led <= '0';
    end if;
end process;
end Behavioral;

```

### 3.5. Problemas Encontrados ao Longo do Caminho:

Durante o desenvolvimento, foram identificados desafios relacionados à utilização do software Quartus II nos laboratórios, bem como a presença de recursos limitados. Algumas dificuldades incluíram:

- **Limitações de Recursos:** Restrições de recursos computacionais nos laboratórios impactaram a eficiência da simulação e implementação do projeto.

## 4. RESULTADOS

O sistema é centralizado na chave geral da residência, a qual é orientada por uma chave seletora. Esta chave, juntamente com LEDs que representam os demais cômodos, unifica eficiência energética, automação personalizada e facilidade de uso. A chave geral controla o sistema de luminosidade da casa, impedindo que os demais cômodos estejam acionados caso a chave não esteja ligada, que contribui para a eficiência energética, permitindo que os usuários desliguem completamente os cômodos não utilizados.. Como demonstrado na foto 1.

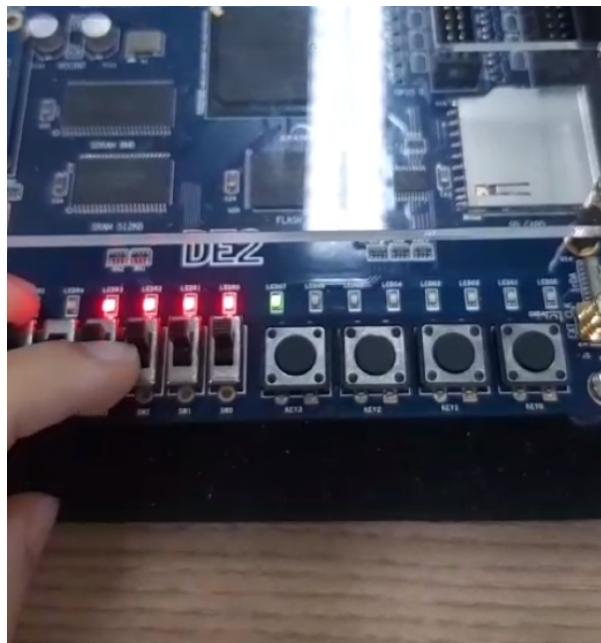


Foto 1: Acionamento da chave geral e dos cômodos

Para o controle de luminosidade, foi empregado um push\_button, acionado através da chave geral. Este botão permite a seleção de cinco níveis diferentes de intensidade luminosa, variando de 1 (mais fraco) a 5 (mais forte). Essa abordagem oferece flexibilidade ao usuário e automação personalizada para ajustar a luminosidade de toda a residência ou de cômodos específicos de acordo com suas preferências. Ademais, a interface simples, com a chave seletora e o push\_button, torna o sistema intuitivo e de fácil utilização, proporcionando uma experiência amigável aos usuários.

O sistema também incorpora um modo noturno, ativado por meio de uma chave seletora dedicada. Quando o modo noturno está ativo, o controle de luminosidade é restrito a apenas três níveis, proporcionando uma iluminação mais suave e aconchegante durante a noite. Essa funcionalidade visa melhorar a experiência dos usuários durante o período noturno, evitando níveis de luminosidade excessivos.

O projeto de controle de iluminação residencial utilizando FPGA demonstrou eficácia na implementação de um sistema que integra eficiência energética, automação personalizada e facilidade de uso. Os resultados obtidos refletem a capacidade da FPGA em lidar com as complexidades do controle de iluminação, proporcionando uma solução robusta e flexível para ambientes residenciais.

## 5. CONCLUSÃO

O presente projeto de controle de iluminação residencial com o uso de FPGA, implementado no ambiente Quartus II Web Edition 13.0sp1, representa uma integração bem-sucedida de hardware e software para criar um sistema versátil, inteligente e eficiente. A abordagem estruturada desde a definição de requisitos até a implementação do código VHDL reflete a aplicação de uma metodologia cuidadosa e sistemática.

Ao longo do desenvolvimento, foram estabelecidas funcionalidades específicas, incluindo o

acionamento individual de luzes por cômodo, controle de intensidade, modo noturno, e uma chave geral para ligar e desligar todas as luzes da residência. A estruturação do projeto em uma máquina de estados proporcionou uma visão clara das transições de estados e das ações associadas, contribuindo para a compreensão e manutenção do código VHDL.

A escolha do Quartus II como ambiente de desenvolvimento mostrou-se eficaz, embora tenhamos enfrentado desafios relacionados à limitação de recursos nos laboratórios. As dificuldades, como limitações computacionais e tempo de compilação, foram superadas por meio de otimizações no código VHDL e alocação eficiente de recursos.

A metodologia adotada permitiu uma abordagem modular e flexível, possibilitando a fácil adaptação do sistema a futuras modificações no layout da casa ou alterações nas preferências dos moradores. A implementação de um modo noturno automatizado, associado a uma chave seletora, destaca o comprometimento com a eficiência energética e o conforto dos usuários.

O código VHDL desenvolvido demonstra uma lógica de controle robusta, refletindo a complexidade do sistema de iluminação residencial. A utilização de uma máquina de estados como estrutura facilita a compreensão do fluxo de controle e a manutenção do código.

Em conclusão, o projeto atingiu seus objetivos, proporcionando um sistema de controle de iluminação residencial inteligente e adaptável. O aprendizado obtido ao longo do desenvolvimento contribuirá para futuros projetos e aprimoramentos, solidificando o conhecimento em design digital, FPGAs e sistemas embarcados. O sucesso deste projeto valida a viabilidade e eficácia da abordagem escolhida para o desenvolvimento de sistemas de automação residencial.

## **6. REFERÊNCIAS**

<sup>1</sup>Intel. (2013). Quartus II Web Edition 13.0sp1.

<sup>2</sup>Intel. (2013). Quartus II Software.

<sup>3</sup>Lippincott, B., & Lippincott, J. (1998). VHDL: A Logic Synthesis Approach. Springer.

<sup>4</sup>Altera. (2009). Cyclone II Device Handbook.

<sup>5</sup>Intel. (2013). Quartus II Handbook Volume 1: Design and Synthesis.

<sup>6</sup>Terasic. (2007). DE2 Development and Education Board User Manual.

## 7. ANEXOS

**FPGA Home Control.** Repositório oficial do projeto no Github. Disponível em: [Github.com/ErnaneJ/FPGA-home-control](https://github.com/ErnaneJ/FPGA-home-control) Acesso em 12/12/2023.

Signal Name	FPGA Pin No.	Description
SW[0]	PIN_N25	Toggle Switch[0]
SW[1]	PIN_N26	Toggle Switch[1]
SW[2]	PIN_P25	Toggle Switch[2]
SW[3]	PIN_AE14	Toggle Switch[3]
SW[4]	PIN_AF14	Toggle Switch[4]
SW[5]	PIN_AD13	Toggle Switch[5]
SW[6]	PIN_AC13	Toggle Switch[6]
SW[7]	PIN_C13	Toggle Switch[7]
SW[8]	PIN_B13	Toggle Switch[8]
SW[9]	PIN_A13	Toggle Switch[9]
SW[10]	PIN_N1	Toggle Switch[10]
SW[11]	PIN_P1	Toggle Switch[11]
SW[12]	PIN_P2	Toggle Switch[12]
SW[13]	PIN_T7	Toggle Switch[13]
SW[14]	PIN_U3	Toggle Switch[14]
SW[15]	PIN_U4	Toggle Switch[15]
SW[16]	PIN_V1	Toggle Switch[16]
SW[17]	PIN_V2	Toggle Switch[17]

Tabela 1: Pinagem dos toggle switches.

Signal Name	FPGA Pin No.	Description
KEY[0]	PIN_G26	Pushbutton[0]
KEY[1]	PIN_N23	Pushbutton[1]
KEY[2]	PIN_P23	Pushbutton[2]
KEY[3]	PIN_W26	Pushbutton[3]

Tabela 2: Pinagem dos push\_buttons.

Signal Name	FPGA Pin No.	Description
LEDR[0]	PIN_AE23	LED Red[0]
LEDR[1]	PIN_AF23	LED Red[1]
LEDR[2]	PIN_AB21	LED Red[2]
LEDR[3]	PIN_AC22	LED Red[3]
LEDR[4]	PIN_AD22	LED Red[4]
LEDR[5]	PIN_AD23	LED Red[5]
LEDR[6]	PIN_AD21	LED Red[6]
LEDR[7]	PIN_AC21	LED Red[7]
LEDR[8]	PIN_AA14	LED Red[8]
LEDR[9]	PIN_Y13	LED Red[9]
LEDR[10]	PIN_AA13	LED Red[10]
LEDR[11]	PIN_AC14	LED Red[11]
LEDR[12]	PIN_AD15	LED Red[12]
LEDR[13]	PIN_AE15	LED Red[13]
LEDR[14]	PIN_AF13	LED Red[14]
LEDR[15]	PIN_AE13	LED Red[15]
LEDR[16]	PIN_AE12	LED Red[16]
LEDR[17]	PIN_AD12	LED Red[17]
LEDG[0]	PIN_AE22	LED Green[0]
LEDG[1]	PIN_AF22	LED Green[1]
LEDG[2]	PIN_W19	LED Green[2]
LEDG[3]	PIN_V18	LED Green[3]
LEDG[4]	PIN_U18	LED Green[4]
LEDG[5]	PIN_U17	LED Green[5]
LEDG[6]	PIN_AA20	LED Green[6]
LEDG[7]	PIN_Y18	LED Green[7]
LEDG[8]	PIN_Y12	LED Green[8]

Tabela 3: Pinagem dos LEDs.

Signal Name	FPGA Pin No.	Description
HEX0[0]	PIN_AF10	Seven Segment Digit 0[0]
HEX0[1]	PIN_AB12	Seven Segment Digit 0[1]
HEX0[2]	PIN_AC12	Seven Segment Digit 0[2]
HEX0[3]	PIN_AD11	Seven Segment Digit 0[3]
HEX0[4]	PIN_AE11	Seven Segment Digit 0[4]
HEX0[5]	PIN_V14	Seven Segment Digit 0[5]
HEX0[6]	PIN_V13	Seven Segment Digit 0[6]
HEX1[0]	PIN_V20	Seven Segment Digit 1[0]
HEX1[1]	PIN_V21	Seven Segment Digit 1[1]
HEX1[2]	PIN_W21	Seven Segment Digit 1[2]
HEX1[3]	PIN_Y22	Seven Segment Digit 1[3]
HEX1[4]	PIN_AA24	Seven Segment Digit 1[4]
HEX1[5]	PIN_AA23	Seven Segment Digit 1[5]
HEX1[6]	PIN_AB24	Seven Segment Digit 1[6]

Tabela 4: Pinagem dos displays de 7 segmentos.

