

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE
SANTA CATARINA – CAMPUS FLORIANÓPOLIS
ENGENHARIA ELETRÔNICA

IDENTIFICADOR DE COR COM FPGA

LEONARDO SANTIAGO BENITEZ PEREIRA

**FLORIANÓPOLIS
DEZEMBRO DE 2017**

SUMÁRIO

INTRODUÇÃO	3
DESENVOLVIMENTO.....	4
IMPLEMENTAÇÃO	6
Sensoriamento	7
Conversão.....	7
Processamento.....	8
Interface	8
CONCLUSÃO	9
APÊNDICE A – MÁQUINA DE ESTADOS	11

INTRODUÇÃO

Este relatório tem por objetivo descrever, de forma breve, o processo de projeto e implementação de um sensor de cor utilizando o módulo didático DE2-115, que contém um FPGA (*Field Programmable Gate Array*) e diversos dispositivos periféricos.

Tal projeto tinha por escopo ser capaz de diferenciar um objeto verde de um objeto azul, quando este fosse colocado próximo ao sensor luminoso. O resultado dessa identificação deveria ser exibido no display LCD disponível no módulo.

Este projeto foi realizado no segundo semestre de 2017 como atividade para a disciplina de Eletrônica Digital II, do curso de Engenharia Eletrônica do Instituto Federal de Santa Catarina, ministrada pelo professor Marcos Matsuo.

DESENVOLVIMENTO

Utilizou-se a metodologia de dividir conceitualmente o projeto em um bloco de controle e um bloco operativo, porém optou-se por manter separado o bloco de comando do display LCD, melhorando assim a organização do projeto e o grau de abstração.

O bloco de controle foi responsável pelo sequenciamento e temporização das ações. A figura 1 contém um diagrama de tempo do funcionamento deste bloco, sendo *Light_blue* e *Light_green* sinais enviados para o periférico de sensoriamento e *Register_blue* e *Register_green* sinais enviados ao bloco operativo para habilitação dos registradores de armazenamento. Tal sequência era executada sempre que o usuário pressionar um botão presente na interface do módulo de desenvolvimento.

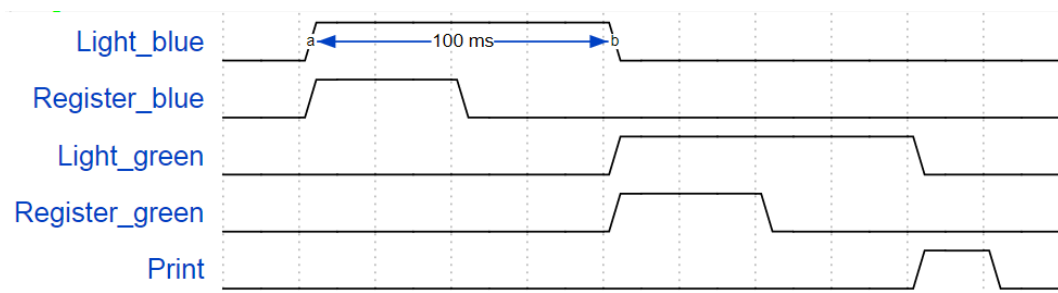


Figura 1 – Diagrama de tempo do bloco de controle.

Este funcionamento deveria ser descrito através de uma máquina de estados, cujo diagrama pode ser visto no apêndice A. Para evitar que o sensor permanecesse continuamente realizando leituras (caso o usuário pressionasse o botão de disparo por mais de 200ms), optou-se por manter a máquina de estados em “read blue” até o usuário soltar o botão.

O bloco operativo foi responsável pelo armazenamento e processamento das informações, podendo então ser implementado através de dois registradores paralelo-paralelo e um comparador de magnitude. A figura 2 contém um diagrama dos blocos constituintes do sistema como um todo.

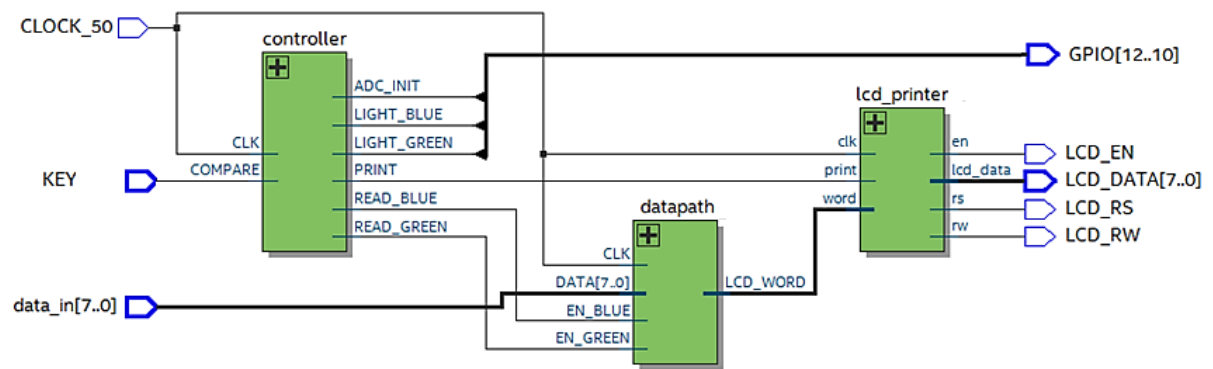


Figura 2 – Diagrama de blocos do sistema.

IMPLEMENTAÇÃO

Utilizou-se a linguagem VHDL para descrever o hardware e os *softwares* Notepad++ para editar o código, Quartus Prime 17 para compilar e a plataforma Github para realizar o controle de versão.

A figura 3 contém a Estrutura Analítica do Projeto, que subdivide o projeto em 4 entregas: sensoriamento, conversão, processamento e interface.

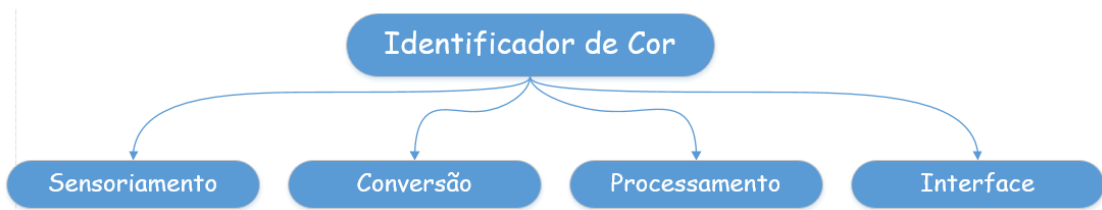


Figura 3 – Estrutura Analítica do Projeto

Ao longo de todo o ciclo de vida do projeto teve-se a preocupação de documentar, comentar e organizar os arquivos da melhor forma possível. A figura 4 contém um trecho de código com o cabeçalho – presente em todos os arquivos – e os devidos comentários explicando o código.

```
-- Project:    Color identifier with FPGA
-- File:      main.vhd
-- Author:    Leonardo Benitez
-- Endianness: little endian
-- Version:   1.0                                29/10/2017
--=====

library ieee;
use ieee.std_logic_1164.all;

-----

entity main is
    port(
        CLOCK_50: in    std_logic;
        KEY      : in    std_logic_vector (3 downto 0); -- Active in Low
        SW       : in    std_logic_vector (17 downto 0); -- (only for debug) Active in high
        LEDG     : out   std_logic_vector (8 downto 0);  -- (only for debug)
        LEDR     : out   std_logic_vector (17 downto 0); -- (only for debug)
        LCD_RW   : out   std_logic;                     -- LCD Read/Write
        LCD_RS   : out   std_logic;                     -- LCD Register Selection
        LCD_EN   : out   std_logic;                     -- LCD Enable
        LCD_DATA : out   std_logic_vector (7 downto 0); -- LCD data bus
        GPIO     : inout std_logic_vector (35 downto 0) -- General Purpose IO
    );
end main;
```

Figura 4 – Código documentado e comentado.

Sensoriamento

O sensor foi implementado com um *Light Emitter Diode RGB* (LED RGB) e um *Light Dependent Resistor* (LDR). As cores azul e verde do LED deveriam ser ativadas individualmente e a intensidade luminosa refletiva deveria ser transduzida para um nível de tensão pelo LDR.

O periférico de sensoriamento (figura 5) foi integralmente fornecido pelo professor da disciplina, cabendo aos alunos apenas a compreensão dos conceitos.

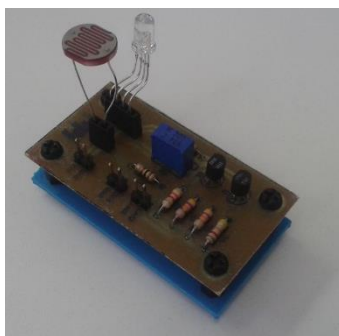


Figura 5 – Periférico de sensoriamento.

Conversão

O periférico de conversão (figura 6) foi implementado utilizando o conversor analógico-digital ADC0804 da *National Instruments* e, para evitar problemas por utilização incorreta do módulo didático, utilizou uma alimentação externa. O periférico de conversão também foi integralmente fornecido pelo professor da disciplina.

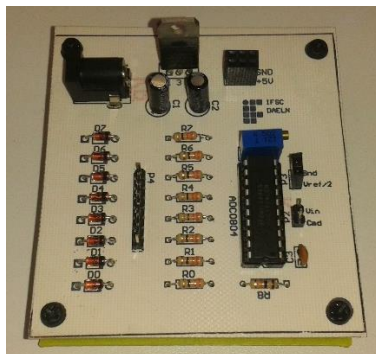


Figura 6 – Periférico de conversão.

Processamento

O processamento dos dados lidos no conversor foi implementado no FPGA com um comparador de magnitude, seguindo a estrutura de funcionamento estabelecida no início do projeto.

Interface

A comunicação com o usuário se deu através de um botão, na qual este poderia comandar o início da leitura, e de um display LCD presente no módulo didático, que indicava o resultado da identificação.

A figura 7 contém um trecho do código utilizado para imprimir uma palavra na tela. Tentou-se implementar também a exibição de uma mensagem inicial “welcome”, porém a equipe encontrou dificuldades técnicas para disparar eventos na inicialização do hardware. Como esta funcionalidade não pertencia ao escopo original do projeto, a implementação foi abandonada.

```
if (word="00") then
  case char is
    when 1      => lcd_enable <= '0'; lcd_clear <= '0';
    when 2      => lcd_enable <= '0'; lcd_clear <= '1';
    when 3      =>
    when 4      => lcd_bus <= W;
    when 5      => lcd_bus <= E;
    when 6      => lcd_bus <= L;
    when 7      => lcd_bus <= C;
    when 8      => lcd_bus <= O;
    when 9      => lcd_bus <= M;
    when 10     => lcd_bus <= E;
    when others => lcd_enable <= '0'; char := 0; printing := '1';
  end case;
end if;
```

Figura 7 – Código para imprimir uma palavra.

CONCLUSÃO

O projeto proposto como atividade final da disciplina foi realizado com sucesso, permitindo aos estudantes aplicar os conhecimentos obtidos ao longo do semestre e se familiarizar com o trabalho por projetos.

APÊNDICES

APÊNDICE A – MÁQUINA DE ESTADOS

