UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

CURSO DE CIÊNCIAS DA COMPUTAÇÃO

INE5406 - SISTEMAS DIGITAIS

PROJETO PRÁTICO DE SISTEMAS DIGITAIS:

VERIFICADOR PARA CÁLCULO DE POSIÇÃO DE UM PONTO EM UMA IMAGEM

Área:

Sistemas Digitais

Equipe:

Fabíola Maria Kretzer

Vinícius Schwinden Berkenbrock

Florianópolis

Novembro, 2016

Equipe:

Fabíola Maria Kretzer

Vinícius Schwinden Berkenbrock

PROJETO PRÁTICO DE SISTEMAS DIGITAIS:

VERIFICADOR PARA CÁLCULO DE POSIÇÃO DE UM PONTO EM UMA IMAGEM

Trabalho da disciplina “INE5406 - Sistemas Digitais” apresentado ao Curso

de Ciências da Computação do Departamento de Informática e Estatística da

Universidade Federal de Santa Catarina.

Professor: Rafael Luiz Cancian, Dr. Eng.

Florianópolis

Novembro, 2016

Sumário

[1. Introdução 4](#_Toc467638767)

[2. Projeto do Sistema 5](#_Toc467638768)

[2.1 Identificação das Entradas e Saídas 5](#_Toc467638769)

[Figura 2.1: Interface do sistema digital proposto 5](#_Toc467638770)

[2.2 Descrição e Captura do Comportamento 6](#_Toc467638771)

[Figura 2.2: FSM Projetada pelo software Quartus II 6](#_Toc467638772)

[2.3 Projeto do Bloco Operativo 6](#_Toc467638773)

[Figura 2.3: Diagrama do Topo do Projeto 6](#_Toc467638774)

[2.3.1 Projeto do cálculo da posição do ponto 7](#_Toc467638775)

[2.3.2 Projeto do Conversor Preto ou Branco 7](#_Toc467638776)

[2.4 Projeto do Bloco de Controle 7](#_Toc467638777)

[3. Desenvolmento 8](#_Toc467638778)

[4. Conclusões 8](#_Toc467638779)

[5. Bibliografia: 9](#_Toc467638780)

# 1. Introdução

A ideia desse projeto prático de sistemas digitais é programar um sistema digital síncrono que realize a identificação da posição de um ponto em uma imagem que será gerada pela câmera de modelo ov7670. Para a descrição do comportamento será utilizado à ferramenta Quartus II. Nela será utilizado código VHDL para passar uma imagem colorida obtida através de barramento serial com a câmera OV7670 para a forma ‘preto ou branco’, e por fim indicar o endereço em que possui o primeiro ponto preto que for varrido.

Depois deverá ser prototipado em FPGA da Altera.

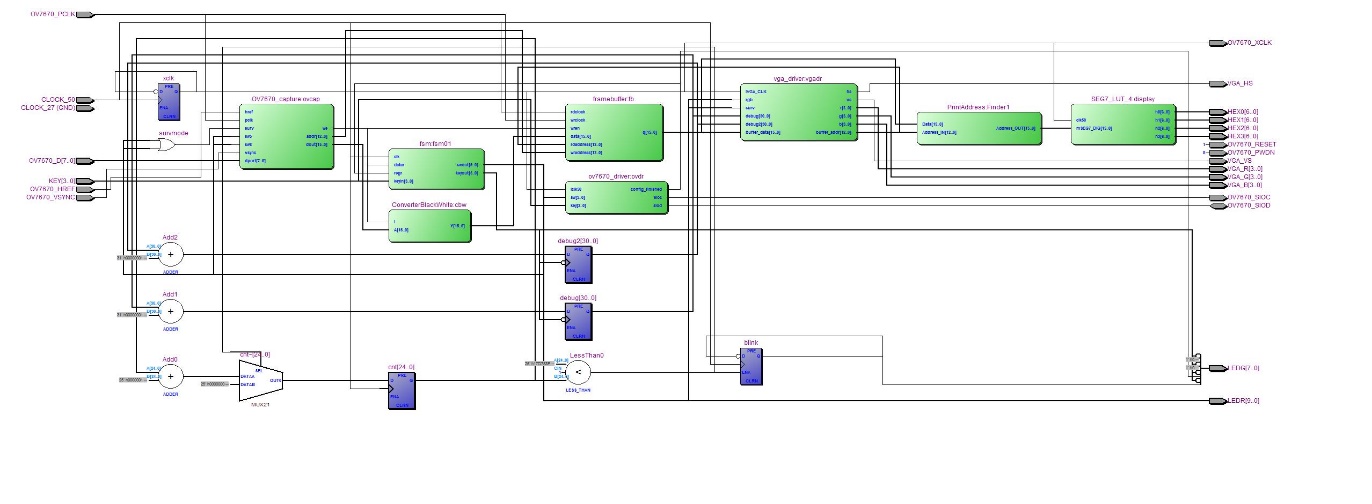
Para o sistema digital funcionar, basta o usuário apontar a câmera para uma folha com um ponto preto bem definido, usando a saída VGA para se guiar caso preciso e ativar a porta SW(6).

# 2. Projeto do Sistema

Nas seções seguintes será realizado a descrição e o comportamento do sistema digital.

## 2.1 Identificação das Entradas e Saídas

O sistema digital possui a interface apresentada na figura 2.1.



### Figura 2.1: Interface do sistema digital proposto

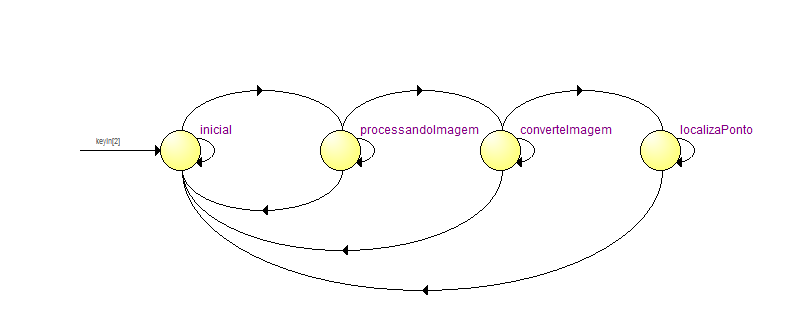
Fonte: Própria

#### Entradas e Saídas:

* Entrada para câmera OV7670 serial.
* Entrada para Clock\_50.
* Entrada para Keys e SW.
* Saída para VGA (para mirar).
* Saída para Leitores Hexadecimal.
* Saídas para Leds Verdes e Vermelhos para conferir FSM funcionando.

O reset assíncrono desse sistema é KEY(2);

## 2.2 Descrição e Captura do Comportamento



### Figura 2.2: FSM Projetada pelo software Quartus II

O sistema começa no estado inicial, e logo após quando os componentes estão prontos para importar, ele segue ao próximo estado (captura e armazena imagem no buffer).

Em seguida ele converte a Imagem para Preto ou Branco (Imagens claras para branco, escuras para preto). Por fim ele lê os dados armazenados finais e marca o ponto que for preto no leitor em Hexadecimal.

## 2.3 Projeto do Bloco Operativo

C:\Users\owner\AppData\Local\Microsoft\Windows\INetCacheContent.Word\Untitled Diagram(1).png

### Figura 2.3: Diagrama do Topo do Projeto

Utilizamos um projeto de captura de imagem pronto e a partir dele modificamos o comportamento base para adaptar à nossa necessidade.

Foram inclusos 2 arquivos VHDL nesse bloco sendo eles o conversor para Preto ou Branco e o indicador de endereço.

### 2.3.1 Projeto do cálculo da posição do ponto

O sistema não funciona por cálculo em si, mas sim por varredura de memória sendo que mostra e mantem o primeiro pixel preto à mostra.

### 2.3.2 Projeto do Conversor Preto ou Branco

Forçamos a saída ser x”FFFF” quando a média de cor é maior que x”000F” (força o branco), caso contrário força a saída ser x”0000”;

## 2.4 Projeto do Bloco de Controle

O Bloco de Controle funciona da seguinte forma:

Possui de entradas: Clock, SW, KEY, Registradores Prontos, Data Pronta.

Para saída possui: SW e KEY, sendo que todos os outros módulos do projeto são controlados por essas chaves.

Para passar do estado Processando\_Imagem para Converte\_Imagem ativa-se a chave SW(6).

Para dar um reset Assíncrono usa-se o botão Key(2).

# 3. Desenvolmento

O Desenvolvimento do projeto deu-se inicio tentando implementar primeiramente apenas com as funções do QSyS para que pude-se ser feito o uso do processador NIOS II.

Contudo essa investida não foi favorável pois dificultou muito a implementação de operações por componentes (por exemplo a conversão preto branco) se não fosse diretamente por programação em linguagem C diretamente.

O trabalho foi refeito 8 vezes até o estado atual, e essa foi a melhor configuração que dispomos.

Por fim resolvemos voltar e usar apenas VHDL para implementação do projeto tendo em vista que facilitou muito para o término do trabalho.

# 4. Conclusões

Quando começamos a desenvolver o projeto, achamos que para fazer um sistema digital era só aplicar todos os métodos de descrição de hardware (VHDL) e interligar com o software, mesmo que não esteja apresentado no trabalho pois como foi dito refizemos o trabalho diversas vezes, além de outras metodologias ensinadas durante a aula prática. Mas ao iniciar, percebemos que iria ser necessária muita concentração, planejamento e esforço para buscar as soluções e resolver todos os problemas que apareciam. Todas as dificuldades e problemas que enfrentamos durante a elaboração do projeto nos permitiram esclarecer muitas dúvidas que surgiram nesse caminho e não tínhamos percebido ao longo do semestre.

Mas com muito esforço e determinação conseguimos aprender muitas coisas que nos ajudaram para terminar o projeto.

# 5. Bibliografia:

<https://github.com/ShiGGie/FPGA-OV7670-cam>

<http://hamsterworks.co.nz/mediawiki/index.php/OV7670_camera>

<http://ac.els-cdn.com/S1877705811054853/1-s2.0-S1877705811054853-main.pdf?_tid=581b450c-b13d-11e6-8ea7-00000aab0f02&acdnat=1479878951_5545a6e0e3248bdc455369f7677ab324>

<https://moodle.ufsc.br/mod/resource/view.php?id=1056248>

<https://moodle.ufsc.br/mod/resource/view.php?id=1056249>

<https://moodle.ufsc.br/mod/resource/view.php?id=1056250>

<http://www.voti.nl/docs/OV7670.pdf>

<https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/ug_vip.pdf>

<http://www.alterawiki.com/wiki/Altera's_Video_and_Image_Processing_Suite_Demo_on_the_NEEK>

<http://www.lcis.com.tw/paper_store/paper_store/ALTERA%20FPGA%20in%20IMAGE-20147129104695-2014124234631.pdf>