

INSTITUTO FEDERAL DE SANTA CATARINA

VINÍCIUS ANTÔNIO HAMES

**Interface para Captura e Processamento de
Imagens em FPGA**

São José - SC

dezembro/2017

INTERFACE PARA CAPTURA E PROCESSAMENTO DE IMAGENS EM FPGA

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina para a obtenção do diploma de Tecnólogo em Sistemas de Telecomunicações.

Orientador: Professor Me. Diego da Silva de Medeiros

São José - SC

dezembro/2017

Vinícius Antônio Hames

Interface para Captura e Processamento de Imagens em FPGA/ Vinícius Antônio Hames. – São José - SC, dezembro/2017-

49 p. : il. (algumas color.) ; 30 cm.

Orientador: Professor Me. Diego da Silva de Medeiros

Monografia (Graduação) – Instituto Federal de Santa Catarina – IFSC
Campus São José

Tecnólogo em Sistemas de Telecomunicações, dezembro/2017.

1. FPGA 2. VGA 3. VHDL 4. Memória 5. Vídeo Analógico I. Orientador Prof.Me. Diego da Silva de Medeiros. II. Instituto Federal de Santa Catarina. III. Campus São José. IV. Interface para Captura e Processamento de Imagens em FPGA

VINÍCIUS ANTÔNIO HAMES

INTERFACE PARA CAPTURA E PROCESSAMENTO DE IMAGENS EM FPGA

Este trabalho foi julgado adequado para obtenção do título de Tecnólogo em Sistemas de Telecomunicações, pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, e aprovado na sua forma final pela comissão avaliadora abaixo indicada.

São José - SC, 15 de outubro de 2015:

**Professor Me. Diego da Silva de
Medeiros**
Orientador
Instituto Federal de Santa Catarina

Professor Fernando Rodrigues Santos
Instituto Federal de Santa Catarina

Professor Juliano Souza
Instituto Federal de Santa Catarina

*Este trabalho é dedicado às crianças adultas que,
quando pequenas, sonharam em se tornar cientistas.*

AGRADECIMENTOS

Primeiramente, agradeço à Deus, por me proporcionar inspiração e coragem na busca de meus objetivos.

À minha mãe, Janete, e meu pai, Valcir, por todo amor e apoio durante toda minha vida escolar e acadêmica, que incentivaram e permitiram todas as minhas conquistas.

À minha companheira, Ágata, que sempre me incentivou, acreditando em meu potencial.

Ao Prof. Me. Diego da Silva de Medeiros, por ter aceitado ser meu orientador, e me dar todo o suporte necessário durante o desenvolvimento deste trabalho.

Aos valiosos e eternos amigos do Instituto Federal de Santa Catarina do Campus São José, por todo o companheirismo e carisma, tanto nas horas de trabalho quanto nos momentos de descontração. Não seria possível citar todos, mas à alguns, um agradecimento especial: Thiago J. Silveira, Michel A. Zanetti e Guilherme Flores.

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

RESUMO

Este trabalho propõe um sistema baseado em componentes de hardware programável do tipo FPGA que será administrado através da linguagem VHDL. O sistema irá realizar a captura de imagens à partir de uma câmera de vigilância analógica, armazenando em memória RAM, processando e reproduzindo através de uma interface VGA. Será abordado um breve histórico das tecnologias e padrões que virão a ser utilizados para a implementação do sistema e funcionamento do mesmo.

Palavras-chave: FPGA. VGA. VHDL. Memória. Vídeo Analógico.

ABSTRACT

This document proposes a system based on programmable hardware components of the FPGA will be administered through the VHDL language fit. The system will perform the capture of images starting from an analog surveillance camera, storing in RAM, processing and reproducing through a VGA interface. a brief history of the technologies and standards that will be utilized for the implementation of the system and operation thereof will be discussed.

Keywords: FPGA. VGA. VHDL. Memory. Analog Video.

LISTA DE ILUSTRAÇÕES

Figura 1 – Kit de Desenvolvimento DE2-115	24
Figura 2 – Grafico Vetorial RGB	25
Figura 3 – Entrelaçamento de Quadro	26
Figura 4 – Tabela Evolutiva VGA	27
Figura 5 – Conector D-SUB e pinagem padrão	27
Figura 6 – Sentido da Varredura do Quadro	28
Figura 7 – Sinal de sincronismo Horizontal	29
Figura 8 – Sinal de sincronismo Vertical	29
Figura 9 – Estrutura básica de um CPLD	30
Figura 10 – Estrutura Básica de um código em VHDL	32
Figura 11 – Diagrama de Blocos dos componentes físicos	33
Figura 12 – Diagrama de Blocos lógico	34
Figura 13 – Bloco Lógico ITU_656_Decoder	34
Figura 14 – Bloco Lógico Reset_Delay	35
Figura 15 – Bloco Lógico SDRAM_Control_4Port	36
Figura 16 – Bloco Lógico VGA_Control	37
Figura 17 – Ocupação dos FPGA's	39
Figura 18 – FPGA Cyclone IV código adaptado	40
Figura 19 – Exibição das imagens Obtidas	42
Figura 20 – CT 840 VDF da fabricante CS	49
Figura 21 – HMC-BU106-30B da fabricante HDL	49
Figura 22 – CD 1030 da fabricante JFL	49

LISTA DE ABREVIATURAS E SIGLAS

ASICs - Application Specific Integrated Circuits

CBS - Columbia Broadcasting System

CI - Circuito Integrado

CPLD - Complex PLD

DARPA - Departamento de Defesa dos Estados Unidos

EAV - End of Active Vídeo

EEPROM - Electrically Erasable Programmable Read Only Memory

EUA - Estados Unidos da América

FPGA - Field-Programmable Gate Array

FPS - Frames por segundo

GAL - Generic Array Logic

IEEE - Institute of Electrical and Electronic Engineers

ISSI - Integrated Silicon Solution Inc

NTSC - National Television System Committee

OLMC - output logic macrocell

PAL - Programmable Array Logic

PLA - Programmable Logic Array

PLD - Dispositivos Lógicos Programáveis

PLL - Phase Locked Loop

PSM - Programmable Switch Matrix

RGB - Red, Green, Blue

SAV - Start of Active Vídeo

SDRAM - Synchronous dynamic random access memory

SPLD - Simple PLD

SVGA - Super VGA

SXGA - Graphics Array Super Estendida

VBI - Vertical Blanking Interval

VGA - Video Gate Array

VHDL - VHSIC Hardware Description Language

XGA - Extended Graphics Array

SUMÁRIO

1	INTRODUÇÃO	21
1.1	Motivação	21
1.2	Objetivos	22
1.3	Organização de texto	22
2	FUNDAMENTAÇÃO TEÓRICA	23
2.1	Kit de Desenvolvimento	23
2.2	Sistema NTSC	24
2.2.1	Padrão de Cores	24
2.2.2	Características de Reprodução	26
2.3	Padrão VGA	26
2.3.1	Modelo de exibição do quadro	28
2.4	Dispositivos Lógicos Programáveis	29
2.5	Linguagem de Descrição de Hardware	31
3	DESENVOLVIMENTO	33
3.1	Aquisição do Vídeo	34
3.2	Buffers de linha e Delay do Sistema	35
3.3	Memória de Vídeo	36
3.4	Exibição do vídeo	37
4	ADAPTAÇÕES E RESULTADOS OBTIDOS	39
4.1	Adaptações	39
4.2	Testes	41
5	CONSIDERAÇÕES FINAIS	43
5.1	Trabalhos Futuros	43
	REFERÊNCIAS	45
	APÊNDICES	47
	APÊNDICE A – CÂMERAS	49

1 INTRODUÇÃO

Com o passar dos anos, houveram diversos avanços na área de eletrônica, dos quais podemos citar alguns essenciais: a invenção da válvula, os diodos e transistores, os circuitos integrados (CI) e microprocessadores. Estes avanços nos permitiram chegar em uma revolução no mundo dos circuitos eletrônicos em meados dos anos 70: os Dispositivos Lógicos Programáveis (PLD). O desenvolvimento dos Dispositivos Lógicos Programáveis destacou-se frente aos microprocessadores devido a programação em nível de hardware, permitindo que estes pudessem ser configurados de acordo com a necessidade do usuário, facilidade que não era possível em tecnologias anteriores (popularmente conhecidos como "hardware fixo"). Os primeiros exemplares de PLDs utilizavam apenas portas lógicas convencionais, dispensavam o uso de flip-flops, e foram denominados Programmable Array Logic (PAL) ou Programmable Logic Array (PLA).

A primeira evolução dos PLDs permitiu o uso de funções sequenciais por meio de flipflops, ainda em meados dos anos 70. No início dos anos 80 foi lançada uma nova estrutura de PLD, os Generic Array Logic (GAL), que conferiu grande melhoria em relação aos primeiros PLD. Os GAL utilizam circuitos lógicos adicionais nas saídas de cada PLD, contendo além de flip-flops, multiplexadores e portas lógicas. Em meados dos anos 80, surgiu um novo dispositivo com diversas diferenças em relação as anteriores como custo, desempenho, arquitetura, tecnologia, entre outros. O Field-Programmable Gate Array (FPGA) é baseado em uma matriz de portas lógicas - ao invés de uma "pilha" (Pedroni, 2010) - e possui um número maior de blocos, porém, individualmente são menores e mais sofisticados. Atualmente, os chips FPGA estão presentes em muitos dispositivos de uso diário, como smartphones, notebooks, e outros. Devido a facilidade de possuir hardware mutável, permitindo que seus arranjos físicos sejam "reprogramados" (fundamento dos PLDs), também vem se destacando na indústria e na área acadêmica para projetos e prototipação.

1.1 Motivação

Atualmente, no ambiente corporativo ou residencial, há a possibilidade de implementação de diversos sistemas e equipamentos para auxiliar nas necessidades básicas do dia-a-dia ou, ainda, prover algum tipo serviço. Um exemplo para o ambiente corporativo seria o uso de um PABX para gerenciar a telefonia da organização, ou o uso de DVRs para gerenciar e gravar imagens de um circuito fechado de TV (CFTV) visando a monitoração de ambientes. No âmbito residencial, uma central de alarmes pode ser utilizada para fins de segurança do patrimônio. Entretanto, para que todas essas funções sejam realizadas,

é necessário o uso de equipamentos específicos para desempenhar tais atribuições, e muitas vezes estes equipamentos não possuem qualquer forma de comunicação, inibindo a possibilidade de integração dos serviços prestados e maior eficiência do sistema como um todo. Desta forma, visando prover esta comunicação, o FPGA foi o objeto de estudo por ser um dispositivo que consegue promover esta interação.

A proposta inicial seria utilizar o kit DE2-115 - que já possui o FPGA integrado a sua estrutura - como base para centralizar as funcionalidades encontradas no PABX, DVR e Central de Alarme, em um único hardware. Entretanto, devido a inviabilidade de aprofundar as pesquisas e desenvolvimentos neste nível de complexidade, optou-se por focar apenas na interface para captura e exibição de imagens com a possibilidade de implementação outras funções sem a necessidade de hardware adicional.

1.2 Objetivos

Este trabalho propõe uma interface implementada em FPGA para realizar a captura de imagens à partir de câmeras de vigilância analógicas, processando-as e reproduzindo-as através de uma interface Vídeo Gate Array (VGA). Para controlar os componentes deste sistema, foi utilizada uma linguagem de descrição de hardware. Serão abordados os conceitos fundamentais utilizados e as funções de cada bloco lógico do sistema serão esclarecidas.

1.3 Organização de texto

Para uma melhor estruturação do projeto, o presente documento se apresenta dividido em capítulos: No capítulo 1, presente capítulo, foi apresentada a motivação da realização deste trabalho, assim como uma introdução do que será desenvolvido. No Capítulo 2 será apresentada a fundamentação teórica e breve explicação sobre as tecnologias e padrões utilizados. Detalharemos a estrutura do KIT que é o hardware base para implementação da Interface. No Capítulo 3 explicaremos os módulos criados para implementação do projeto e descreveremos o funcionamento e lógica empregada em cada um deles. No Capítulo 4 serão evidenciados os testes e seus resultados e por fim no Capítulo 5 será abordaremos a conclusão do projeto.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os padrões e tecnologias que foram empregados durante o desenvolvimento do trabalho. Dentre estes, apresentaremos o Kit utilizado como base do projeto, o dispositivo FPGA, a linguagem de descrição de hardware, o sistema de vídeo NTSC, o padrão VGA e as memórias.

2.1 Kit de Desenvolvimento

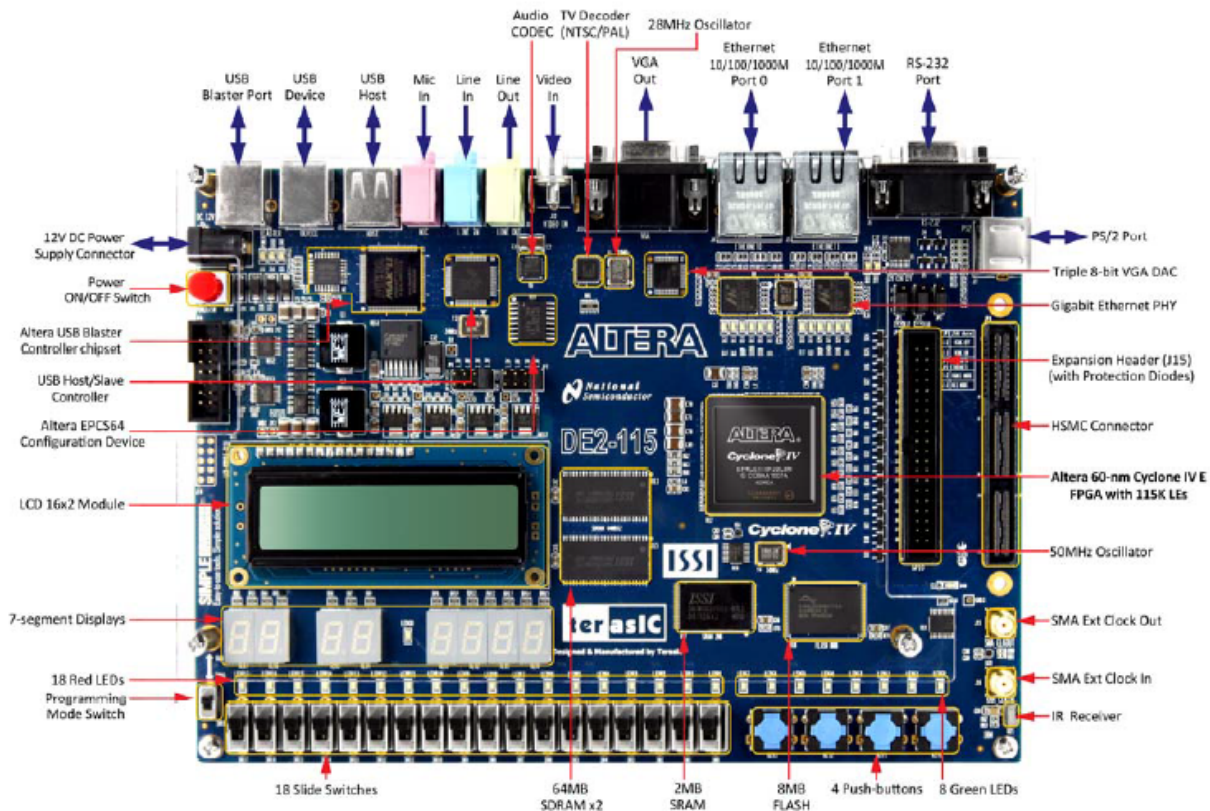
Ao longo dos anos, com a evolução das tecnologias empregadas para construção de hardware, o mercado conseguiu disponibilizar diversas versões de kits de desenvolvimento com variedades que circulam entre preço e funcionalidade. Esta variedade permite, cada vez mais, que universidades e empresas que atuam no ramo de projetos para área tecnológica invistam em estruturas desta linha. O uso destes kit's nas universidades permite que discentes possam transformar os conhecimentos adquiridos nas aulas teóricas, em prática. Ao passo que nas empresas auxiliam nos testes de hardware, fazendo prototipagem e ajustes antes de encaminhar um item na sua versão final para linha de fabricação e venda.

Uma das variedades que diferem uns kits dos outros são as interfaces como, por exemplo, portas USB, Serial, Ethernet e VGA. Essas interfaces que podem aparecer todas no mesmo kit, ou ainda, apenas um tipo por item, possibilitam os mais diversos projetos e experimentos, como, por exemplo, transmissão de dados. Embora estes hardwares sejam mais acessíveis e conhecidos quando comparados com décadas atrás, quanto maior a variedade de portas apresentada pelo hardware, maior a probabilidade de o item possuir um valor agregado maior (Altera Corporation, 2010).

Para o desenvolvimento do projeto apresentado neste trabalho, foi utilizado o modelo kit DE2-115, fabricado pela Terasic Technologies e Altera Corporation.

- Características;
 - FPGA Altera Cyclone IV EP4CE115F29C7;
 - 2 MB de SRAM;
 - 64 MB de SDRAM;
 - 8 MB de Flash;
 - Switch;
 - Leds;
 - Módulo LCD.

Figura 1 – Kit de Desenvolvimento DE2-115



Fonte: Terasic,2010.

Este kit foi selecionado para ser a base da interface de processamento de imagem por contemplar todos os componentes necessários para a execução desta demanda, como, por exemplo, entrada de vídeo analógica, componentes de memória SDRAM, uma FPGA da família Cyclone IV EP4CE115F29C7 e o interface VGA.

2.2 Sistema NTSC

O padrão NTSC (National Television System Committee) foi desenvolvido em 1953 nos Estados Unidos da América (EUA) como sistema padrão para TV analógica a cores no país. Suas principais características são a simplicidade e compatibilidade com sistemas preto e branco, como, por exemplo, o antigo Columbia Broadcasting System, conhecido como CBS.

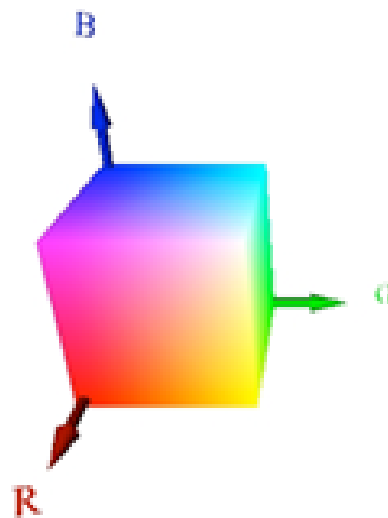
2.2.1 Padrão de Cores

Para definição de cores no NTSC, usou-se inicialmente o modelo YUV. Mais tarde, devido a questões históricas, foi iniciada a utilização do sistema YIQ, que se baseia na separação dos sinais das cores primárias vermelho, verde e azul. Este padrão foi denominado RGB devido às iniciais das cores utilizadas na língua inglesa (Red, Green, Blue). O modelo

RGB considera que todas as cores geradas na transmissão são o resultado da adição das três cores primárias citadas anteriormente.

Todas as cores primárias tem intensidade que pode variar entre 0 e 255, sendo 0 valor corresponde a intensidade mínima e 255 valor de intensidade máxima. A variação entre 0 e 255 determinará o tom da cor obtida. A cor branca representa a soma das três cores primárias em sua intensidade máxima (255, 255, 255), enquanto o preto representa a soma em sua intensidade mínima (0, 0, 0). Com este padrão é possível obter até 16 milhões de cores se considerarmos a soma todas as cores primárias com todos os seus níveis de intensidade (255^3).

Figura 2 – Grafico Vetorial RGB



Fonte: J. M. Brisson Lopes IST, 2013.

Com base no conhecimento sobre o modelo RGB, o padrão YIQ permitiu que a transmissão a cores fosse realizada mantendo uma compatibilidade com os sistemas preto e branco. Isto era possível porque o padrão foi desenvolvido separando a informação de luminância da informação de cromaticidade. Enquanto aparelhos preto e branco recebiam apenas o sinal de luminância (Y), aparelhos a cores recebem, além do sinal de luminância, os sinais de matiz ($I = R - Y$) e quadratura ($Q = B - Y$) (J. M. Brisson Lopes, 2013).

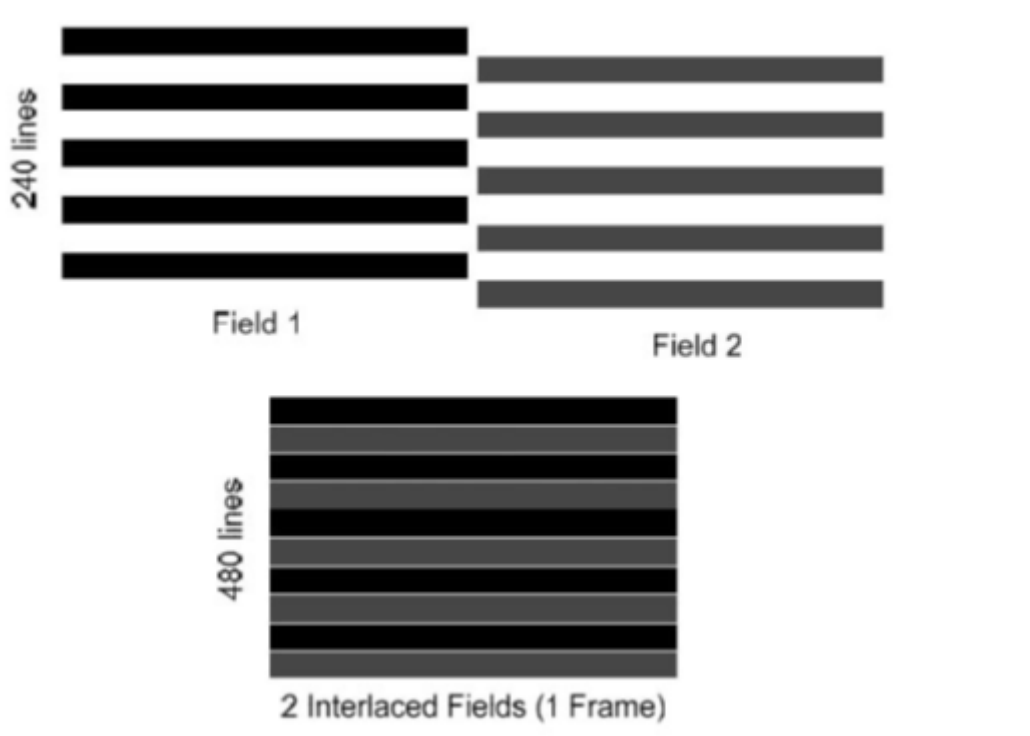
Entende-se por sinal de luminância a quantidade de luz, ou, ainda, luminosidade, emitida por uma superfície em determinada direção, ou seja, o que conhecemos como “brilho”. Por sua vez, matiz é o estado puro da cor, portanto, não possui informação de branco nem preto. Pode ser definido como um atributo de cor que nos permite distinguir cores diferentes, como o azul do vermelho, ou distinguir os diferentes tons de uma mesma cor como, por exemplo, vermelho sangue e vermelho rubi. Já a quadratura ou mais conhecida como saturação é um conceito que define a intensidade com que a cor se apresenta com

base em uma tendência de aproximar-se ou afastar-se do cinza. Quanto mais saturada, mais “viva” e menos próxima do cinza. Quanto menos saturada, mais opaca e mais próxima do cinza.

2.2.2 Características de Reprodução

O sistema NTSC apresenta uma resolução de 525 linhas, das quais 45 são VBI (Vertical Blanking Interval) e as 480 restantes formam a imagem visível. Este formato de vídeo é transmitido em 60 fps (frames por segundo) devido a frequência da rede elétrica (60Hz) nos países em que é utilizado (Unifore, 2014). A transmissão é realizada de forma entrelaçada, ou seja, são transmitidas todas as linhas ímpares primeiro e depois todas as linhas pares, formando o quadro por completo conforme [Figura 3](#).

Figura 3 – Entrelaçamento de Quadro



Fonte: Unifore, 2014

2.3 Padrão VGA

O sistema de vídeo VGA (Video Graphics Array) foi elaborado pela International Business Machines, mais conhecida como IBM, em 1987. O VGA consiste em um padrão de disposição gráfica de vídeo que utiliza sinais analógicos para gerar cores (GARCIA, 2015). A disposição gráfica destas cores, permite que sejam geradas imagens de duas resoluções: 640x480 e 320x240. Como no VGA cada componente RGB é representado por 6 bits, a gama de cores alcança uma variedade de 262.144 cores diferentes. De acordo com a

resolução escolhida, uma pequena parte destas 262.144 cores é selecionada, disponibilizando 16 cores na resolução 640x480 e 256 cores na resolução 320x240(Fernando Deluno Garcia, 2015).

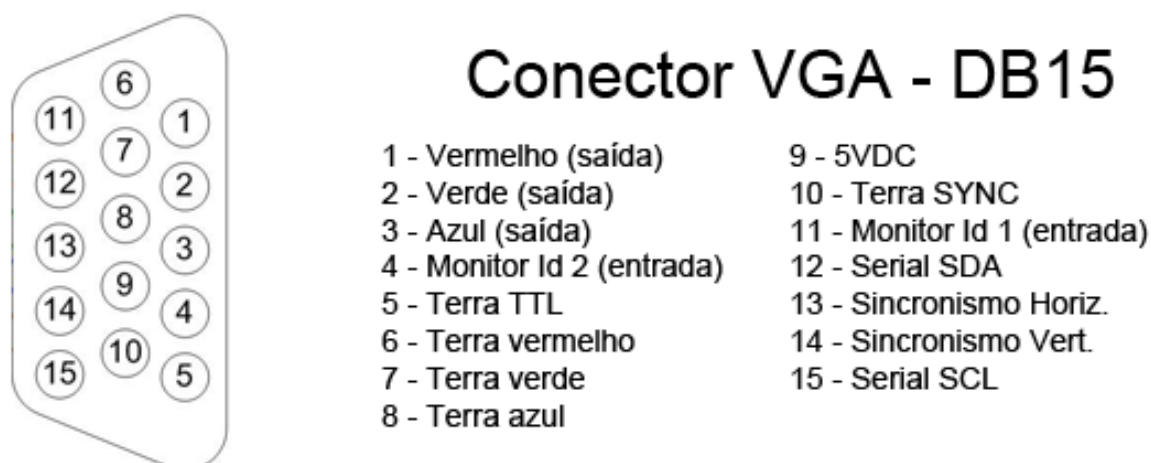
Figura 4 – Tabela Evolutiva VGA

Encontro	Padrão	Descrição	Típico resolução	Número de cores
1981	CGA	Color Graphics Adapter	640x200 ou 160x200	16
1984	EGA	Enhanced Graphics Adapter	640x350	64
1987	VGA	Video Graphics Array	640x480 ou 320x200	262K
1987	SVGA	Super Video Graphics Array	800x600	256~16,7M
1987	8514 / A	IBM interlaced standard	1024x768	262K
1990	XGA	Extend Graphics Array	1024x768	16,7M
1990	SXGA	Super Extend Graphics Array	1280x1024	16,7M
1990	UXGA	Ultra Extend Graphics Array	1600x1200	16,7M
1990	WXGA	Wide Extend Graphics Array	1366x768	16,7M

Fonte: Info Cellar adaptado pelo Autor

Após o desenvolvimento do VGA, diversas variações deste padrão foram criadas, como, por exemplo, o SVGA (Super VGA), XGA (Extended Graphics Array) e o SXGA (Graphics Array Super Estendida). Estes novos padrões foram lançados com intuito de aumentar a resolução e número de cores existentes no VGA. O conector utilizado no padrão VGA é o D-SUB, conhecido também como DB15, e leva este nome por possui o formato da letra D e ter 15 pinos de conexão. Cada pino possui uma especificação de utilização descrita pelo padrão, conforme figura [Figura 5](#).

Figura 5 – Conector D-SUB e pinagem padrão

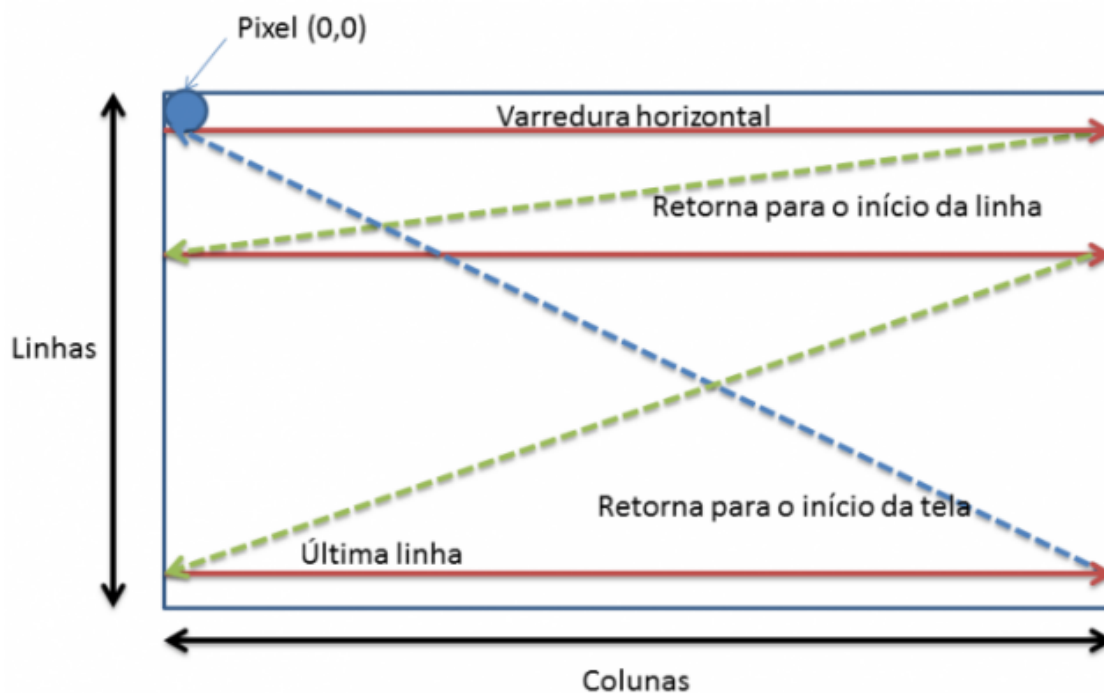


Fonte: Daniel Omar Basconcello Filho, 2016.

2.3.1 Modelo de exibição do quadro

A imagem disposta na tela através do padrão VGA é composta por pixels, que por sua vez, estão organizados em posições baseadas em uma matriz com N linhas e M colunas (NxM). Em cada posição da matriz temos um pixel com suas componentes de cores RGB e sua exibição sempre ocorrerá da esquerda para direita e de cima para baixo. Quando a exibição de uma linha é finalizada, o processo é iniciado novamente, conforme na figura [Figura 6](#).

Figura 6 – Sentido da Varredura do Quadro



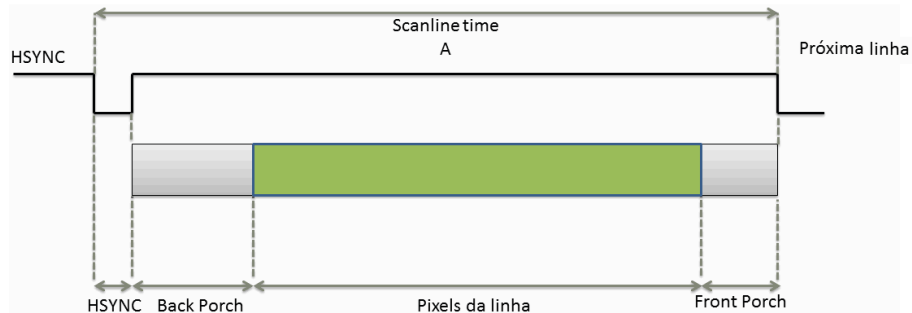
Fonte: Embarcados, 2015

Quando todos os pixels estiverem expostos na tela, teremos um frame (quadro). Para a perfeita exibição do frame é necessário que existam sinais de sincronização alinhados com a utilização de um clock de 25MHz. Estes sinais de sincronização são chamados de Horizontal Sync (H_SYNC) e Vertical Sync (V_SYNC). Estes sinais compõem o que é chamado de *Blanking Time*, parte do frame não visível e à partir deles são determinados os valores de Back Porch e Front Porch (Fernando D. Garcia, 2015).

O sinal H_SYNC é responsável por determinar o sincronismo da varredura horizontal (colunas), conforme veremos na [Figura 7](#). Este sinal deve permanecer, inicialmente, em nível lógico baixo por um período de 96 pulsos de clock (3,84us). Após o período de Sync Pulse, o sinal permanecerá em nível alto por um período de 704 pulsos de clock (28,16us), dos quais, os primeiros 16 pulsos (0,64us) são referentes ao Back Porch. Em seguida, inicia-se a região ativa da linha (pixels correspondentes a linha). As linhas serão exibidas durante 640 pulsos de clock (25,6us), e por fim, haverá mais 48 pulsos de clock

(1,92us), chamados de Front Porch. Após a finalização deste processo, o sinal H_SYNC retorna ao nível lógico baixo, dando início a nova linha (Fernando D. Garcia, 2015).

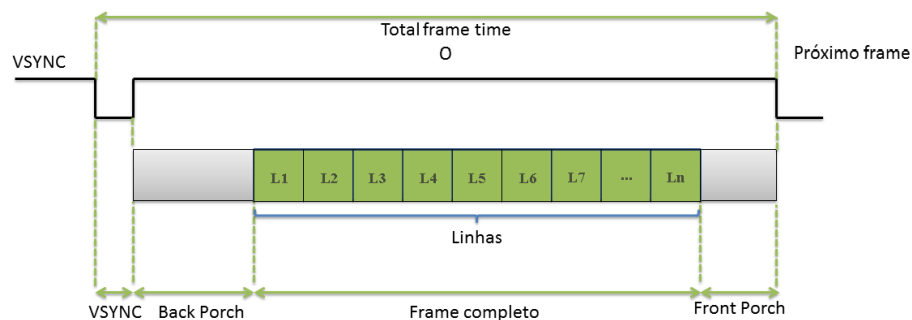
Figura 7 – Sinal de sincronismo Horizontal



Fonte: AGE TECHNOLOGY, 2016

Por sua vez, o sinal V_SYNC é responsável por determinar o sincronismo de varredura vertical (linhas), conforme veremos na Figura 8. Este sinal deve permanecer, inicialmente, em nível lógico baixo por um período de 2 linhas (0,064ms). Após o período de Sync Pulse, o sinal permanece em nível lógico alto por um período de 523 linhas (16,716ms), das quais: as primeiras 12 linhas (0,45ms) são referentes ao Back Porch, as 480 próximas linhas (15,25ms) são referentes a área visível, e por fim, as últimas 31 linhas (1,02ms) são de Front Porch. Após isto, o sinal V_SYNC retorna ao nível lógico baixo dando início a um novo quadro.

Figura 8 – Sinal de sincronismo Vertical



Fonte: AGE TECHNOLOGY, 2016

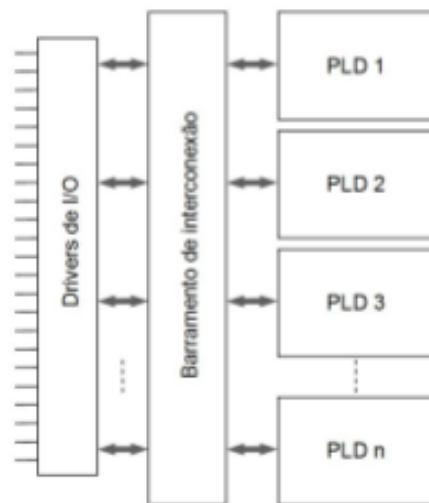
2.4 Dispositivos Lógicos Programáveis

Durante muito tempo, os projetos de sistemas digitais foram realizados utilizando lógica física através de circuitos integrados (CI) que implementavam portas lógicas. Durante os anos 70, surgiram os Programmable Logic Devices (PLD), que consistem em chips com hardware programável. Com o lançamento dos PLDs, os sistemas digitais puderam ser projetados a partir de CI's genéricos, permitindo assim um menor tempo de desenvolvimento e maior flexibilidade no uso.

Com este modelo de arquitetura e tecnologia empregada, os PLDs foram classificados em três categorias: os SPLDs (Simple PLD), CPLDs (Complex PLD) e os FPGA's (Field programmable gate arrays). Os SPLDs são formados por três dispositivos, são eles: os PALs (Programmable Array Logic), que consistem em um conjunto de portas AND programáveis e um conjunto de portas OR fixas. Os PLAs (Programmable Logic Array), que permitem que tanto as portas AND quanto OR sejam configuráveis. E por fim, os GALs (Generic Array Logic), que possuem uma célula de saída chamada OLMC (output logic macrocell), composta por: flip-flops, portas XOR, multiplexadores, um sinal de retorno e memória EEPROM (Electrically Erasable Programmable Read Only Memory) para armazenar as configurações dos circuitos.

Os CPLDs são o estágio intermediário das PLDs, segundo (PEDRONI, 2010). Os CPLDs são chips que agregam vários blocos de SPLDs, conectados através de uma matriz de conexões programáveis. Esta matriz programável é responsável também pela interconexão dos drivers e pinos de I/O, conforme [Figura 9](#).

Figura 9 – Estrutura básica de um CPLD



Fonte: PEDRONI, 2010b.

Os CLBs são responsáveis pelas funções lógicas, implementadas à partir de suas tabelas verdade nos elementos lógicos usando uma memória volátil chamada de Lookup Table (LUT). Geralmente, utiliza-se uma pequena memória FLASH EEPROM, cuja função é carregar automaticamente as células de armazenamento toda vez que o FPGA for energizado. Com isso, um bloco de configuração lógica também pode ser usado como uma memória em diferentes configurações, dependendo dos multiplexadores.

Os IOB são utilizados para prover as interfaces de entrada e saída do FPGA. Consistem em um buffer de entrada e um de saída, ambos com flip-flops, permitindo que saídas com clocks sejam disponibilizadas sem encontrar atrasos significantes e reduzindo a necessidade de manter sinais de entrada por muito tempo.

A chaves de interconexão configuráveis são responsáveis interligar vários CLBs através de caminhos unidirecionais ou em distâncias maiores pelos blocos PSM (Programmable Switch Matrix), os quais consistem em uma matriz programável de comutação. Estes blocos PSM acabam que introduzindo atrasos de propagação do sinal dentro do chip FPGA.

Ainda segundo (PEDRONI,2010), o alto desempenho justifica a necessidade de controle de clock através de circuitos conhecidos como PLL (Phase Locked Loop). Estes circuitos manipulam os sinais de clock que entram no FPGA, podendo até mesmo, quando necessário, deslocar, multiplicar ou dividir o sinal. Ou seja, estes circuitos são capazes de manipular e gerenciar o clock sem que cause distorções no projeto realizado.

2.5 Linguagem de Descrição de Hardware

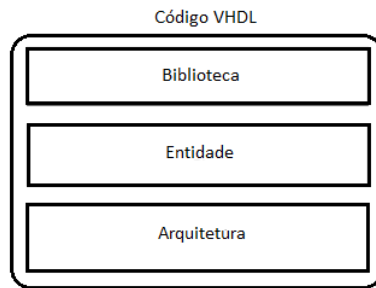
Em sistemas digitais uma linguagem de descrição de hardware a especificação ou modelagem de circuitos digitais em CPLDs, FPGAs e ASICs, sendo assim pode descrever o funcionamento do circuito, a sua concepção, organização e realizar simulações da sua operação. Existem muitos tipos de linguagens para descrever hardware, as mais utilizadas atualmente são Verilog e VHDL (VHSIC Hardware Description Language). A linguagem VHDL surgiu na década de 1980, e foi desenvolvida pela DARPA (Departamento de Defesa dos Estados Unidos) para evidenciar o comportamentos de ASICs que eram utilizados nos equipamentos vendidos as Forças Armadas do EUA. Antes desta linguagem o funcionamento dos ASICs era descrito através de longos e complexos manuais. Após o sucesso inicial do VHDL, sua definição foi posta em dominio público, levando a ser padronizada pelo IEEE (Institute of Electrical and Electronic Engineers) ainda na década de 80. Com o passar dos anos esta linguagem passou por diversas modificações e adaptações em um processo natural de aprimoramento, a versão mais recente foi aprovada em 2008.

A principal característica é execução concorrente das linhas de códigos, diferentemente de processadores e CPU dos computadores, onde os códigos são executados linha a linha, em VHDL as linhas de código são executadas todas no mesmo instante de tempo, ou seja dando um efeito de execução simultânea, outra característica importante é que o código não é case sensitive. O VHDL possui uma estrutura básica relativamente simples, dividida em em três partes, que são Package (Pacotes e Bibliotecas), Entity (Entidade) e Architecture (Arquitetura), conforme podemos observar na [Figura 10](#).

O Package é onde são declaradas as bibliotecas e pacotes que irão ser utilizados, as bibliotecas STD e WORK são disponibilizadas automaticamente, um pacote que é muito utilizado é o STDLOGIC1164 que faz parte da biblioteca IEEE.

A Entity é parte onde declaramos as configurações de PORT (portas) de entrada e saída do nosso projeto, estas portas podem ser de entrada (IN), de saída (OUT)

Figura 10 – Estrutura Básica de um código em VHDL



Fonte: Elaborada pelo Autor.

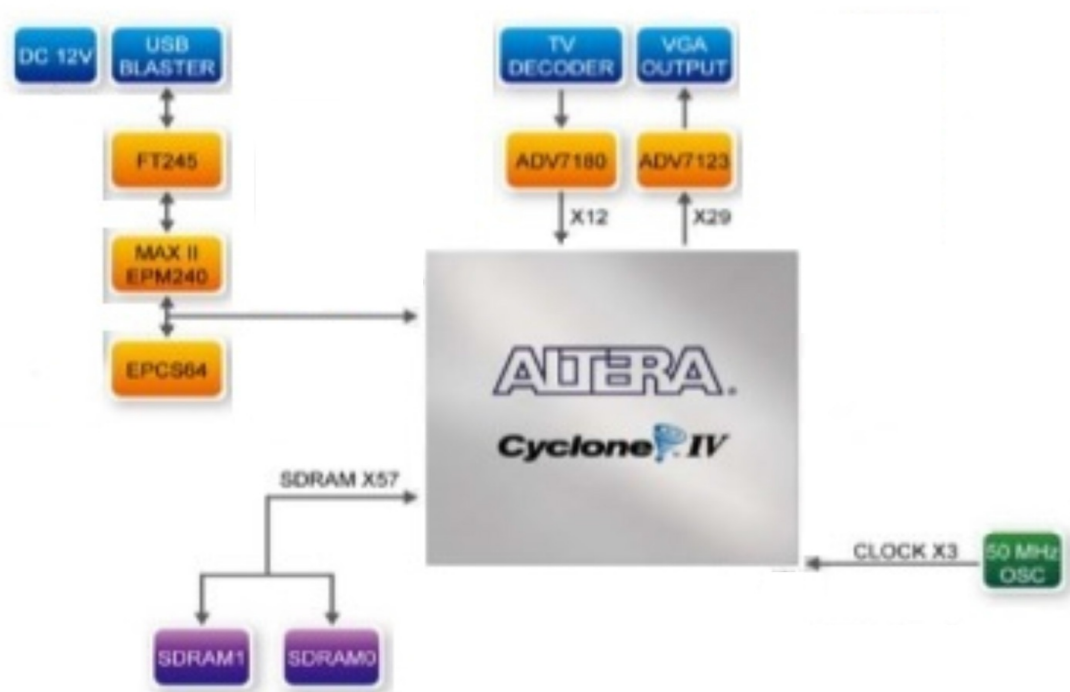
e bidirecionais (INOUT). E para informações globais do sistema é utilizado o campo **GENERIC**.

A **Architecture** é o corpo do projeto propriamente dito, é onde iremos declarar as atribuições, operações, comparações que serão realizadas no projeto.

3 DESENVOLVIMENTO

Neste capítulo será apresentado o desenvolvimento do projeto de forma detalhada, expondo a função que cada bloco exerce no sistema. A interface para reprodução de imagens foi elaborada com base na segunda versão do projeto DE2_115_TV (Terasic, 2012), conforme diagrama de blocos a seguir. Estes componentes encontram-se espalhados pelo kit de desenvolvimento e foram agrupados de acordo com a finalidade pretendida.

Figura 11 – Diagrama de Blocos dos componentes físicos



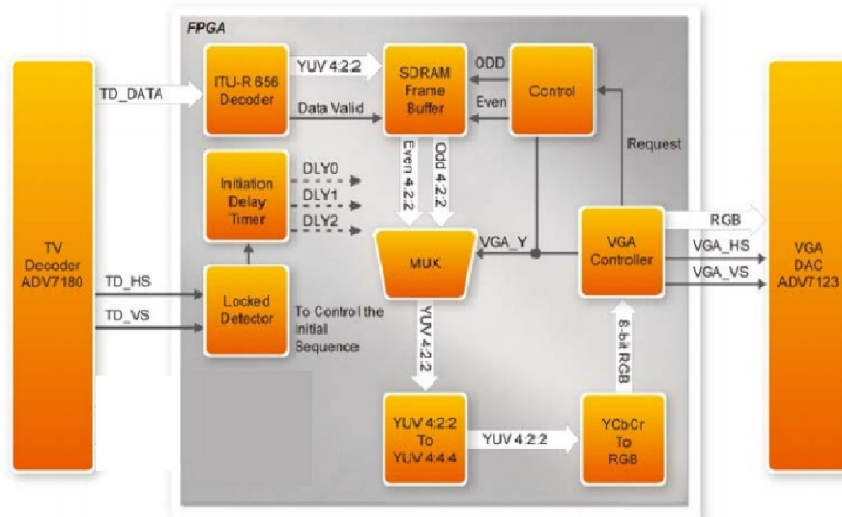
Fonte: Terasic Techonologies Inc., 2003-2010.

O diagrama evidencia todos os componentes citados no capítulo 2.3 e sua interligação, de forma que o sinal capturado pela câmera analógica seja encaminhado para o decodificador de TV (TV decoder) e repassado para o ADV7180 (que faz a conversão analógico – digital). Depois, segue para o FPGA através de um barramento de 12 elementos, onde ocorrerá todo o tratamento do sinal, como armazenamento temporário e adequação para que possa ser encaminhado para o ADV7123, através de um barramento de 29 elementos. No ADV7123 o sinal volta a ser analógico e é encaminhado para a saída VGA (VGA Output) para ser exibido no monitor.

Assim como os componentes físicos, o sistema contempla componentes lógicos que

serão descritos no diagrama de blocos abaixo.

Figura 12 – Diagrama de Blocos lógico

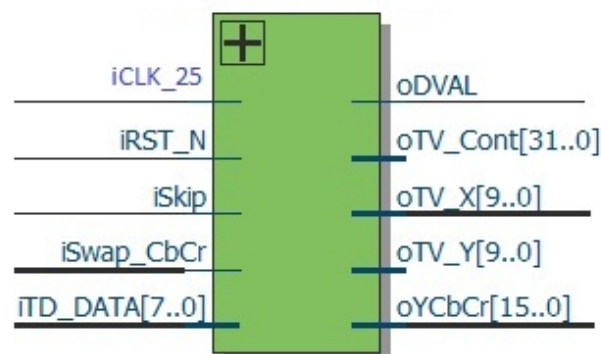


Fonte: Terasic, 2010.

3.1 Aquisição do Vídeo

Conforme citado anteriormente, a placa DE-2 possui um conversor A-D, o ADV 7180. Este componente consiste em um decodificador de vídeo integrado que detecta e converte o sinal NTSC de banda base de televisão em sinal de vídeo digital. O conversor ADV 7180 foi usado no projeto de forma que a câmera entrega o sinal NTSC na entrada iTD_DATA e ele devolve o sinal decodificado no formato 4:2:2 ou 16 bits bits na saída oYCbCr, como podemos verificar no bloco ITU_656_Decoder.

Figura 13 – Bloco Lógico ITU_656_Decoder



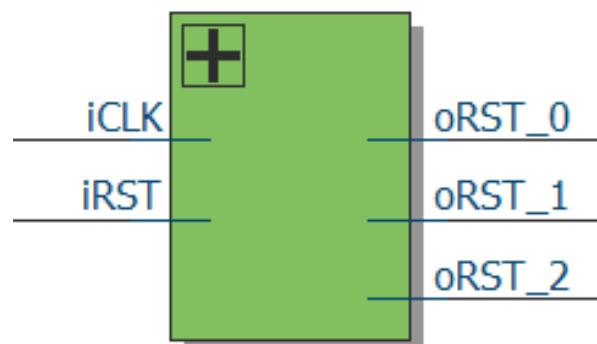
Fonte: Gerado através do software Quartus II

Após o decodificador ser configurado, ele está pronto para receber os pixels do vídeo digitalizado. Na porta de saída do ADV7180 o sinal disponibilizado estará no formato YCbCr (4:2:2), de acordo com a norma ITU-T BT.656. É interessante salientar que o vídeo disponibilizado será na forma entrelaçada, portanto, serão encaminhadas primeiro as linhas ímpares e em seguida as linhas pares. O início de cada linha será sinalizado pelo código SAV (Start of Active Vídeo) e finalizado quando o código EAV (End of Active Vídeo) for enviado. Após o vídeo ser obtido será encaminhado para a memória SDRAM para ser armazenado.

3.2 Buffers de linha e Delay do Sistema

Devido à diferença nas frequências de operação de entrada e saída do sistema é necessário criar alguns atrasos derivados do sinal original. O módulo Reset_Delay é responsável por gerar esses atrasos oRST_0, oRST_1 e oRST_2 que serão usados como sinais de clock por outros blocos do sistema.

Figura 14 – Bloco Lógico Reset_Delay



Fonte: Gerado através do software Quartus II

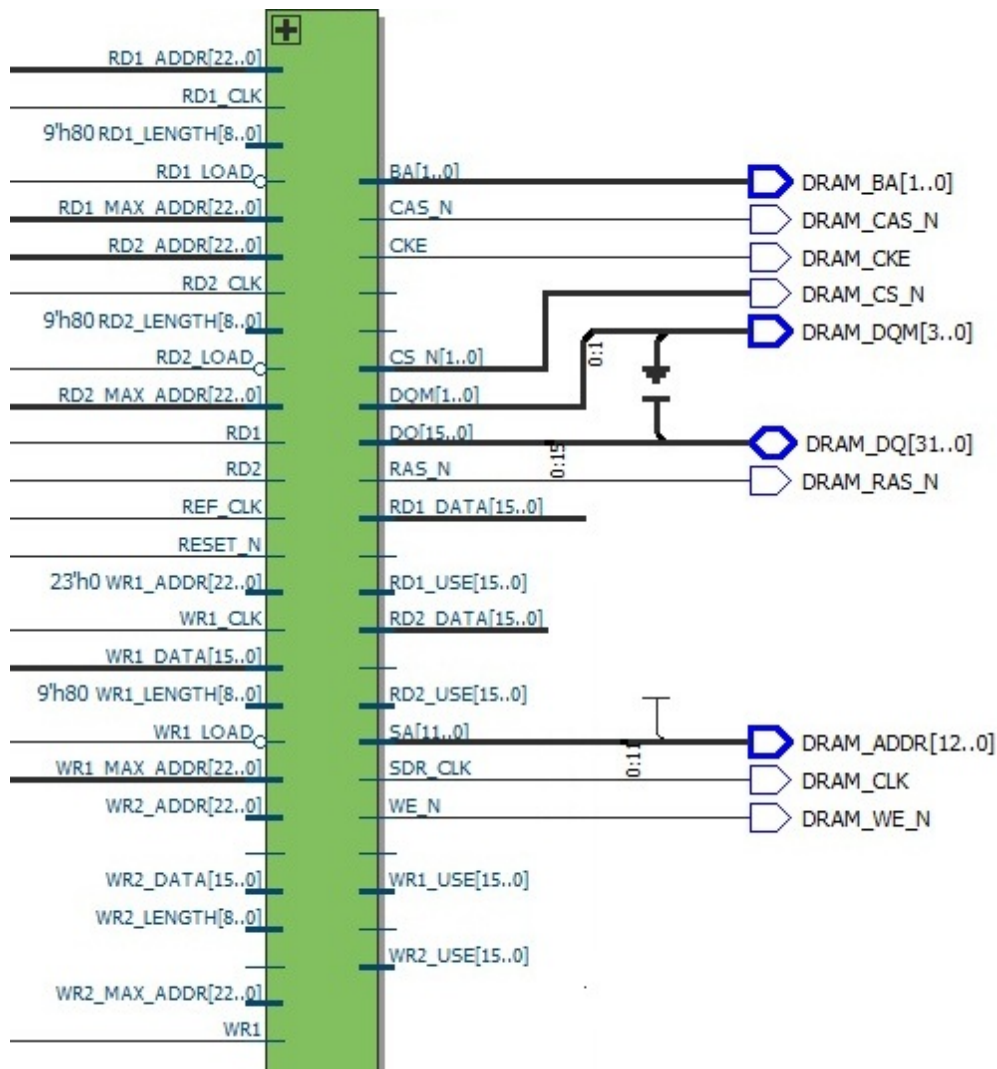
O oRST_0 provê sinais de clock utilizados pelo bloco de memória do sistema (Sdram_Control_4Port), o oRST_1 serve de sinal de reset para o bloco de decodificação de vídeo (ITU_656_Decoder) e o oRST_2 serve de sinal de reset para três blocos do sistema que irão compor o módulo de exibição de vídeo (YCbCr2RGB, Algoritmo e VGA_Ctrl).

As linhas de Buffers estão entre os blocos de entrada e saída de vídeo, sendo que entrada tem sua operação em 13,5MHz e a saída em 25MHz, tendo em vista que os padrões de comunicação da câmera e do monitor usam estas frequências. Estes módulos recebem os sinais de sincronismo TD_HS e TD_VS que são obtidos a partir do ADV7180, com a manipulação desses sinais é possível fazer com que o sistema opere em harmonia, não havendo problemas de sincronismo entre entrada e saída.

3.3 Memória de Vídeo

Devido à diferença de frequências citada acima, após gerar os atrasos necessários utilizamos a memória para armazenar cada frame do vídeo antes do mesmo ser exibido, desta maneira evitamos que o vídeo seja corrompido na hora de exibição. No KIT DE2-115 existe alguns tipos de memórias disponíveis, para realizar o armazenamento temporário das imagens foi utilizado a memória SDRAM (Synchronous Dynamic RAM) este tipo de memória foi escolhido devido a sua capacidade de maior velocidade nas tarefas de leitura e escrita de dados, e possui uma capacidade máxima de 8Mbytes de armazenamento. Podemos ver a disposição de portas de entrada e saída no bloco abaixo.

Figura 15 – Bloco Lógico SDRAM_Control_4Port



Fonte: Gerado através do software Quartus II

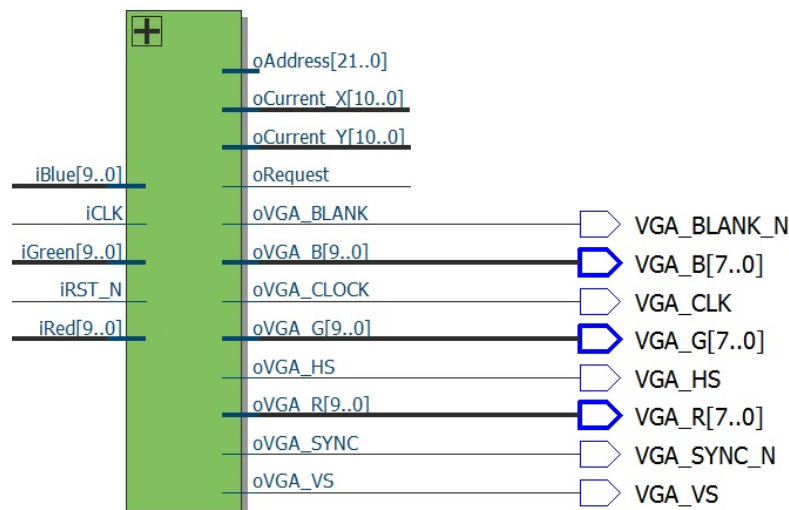
Como podemos ver no diagrama acima a memória é uma parte extremamente complexa do sistema, sendo assim o estudo de construção da memória SDRAM foge ao escopo do trabalho, porém, para mais informações sobre este assunto, pode ser consultado o datasheet no site da Altera Corporation ou do fabricante da memória que neste caso é Integrated

Silicon Solution Inc (ISSI, Dezembro2009) <<http://www.issi.com/WW/pdf/42S16320B-86400B.pdf>>.

3.4 Exibição do vídeo

A saída VGA possui alguns padrões que precisam ser respeitados, relógio (oVGA_CLOCK), sincronismo vertical e horizontal (oVGA_VS e oVGA_HS respectivamente), bloqueio da saída VGA(oVGA_BLANK). Todos os sinais da saída VGA são analógicos e os sinais gerados pelo FPGA são digitais. O kit utilizado tem um conversor D/A composto pelo CI ADV 7123, podemos ver o bloco utilizado a seguir:

Figura 16 – Bloco Lógico VGA_Control



Fonte: Gerado através do software Quartus II

Todos os sinais de sincronismos são baseados no clock de entrada da interface e o valor deste clock é 25 MHz. Para gerar os sincronismos, utilizamos contadores, um para controlar o sincronismo horizontal e um para o sincronismo vertical.

O sincronismo Horizontal é baseado na borda de subida do clock de entrada, esta borda se repete a cada 40 ns. Obedecendo aos padrões descritos na seção 2.2, o sincronismo horizontal fica em estado “1”, alto, durante 660 bordas do clock de entrada, destes, 640 é o período onde são enviados os sinais representando os pixels de uma linha, os outros 20 ciclos são especificados no padrão VGA. Após este período, este sinal de sincronismo passa para estado baixo “0”, por 96 ciclos e retorna ao nível alto por mais 44 ciclos, totalizando assim 800 bordas de subida necessárias.

O sincronismo vertical permanece em nível alto por 494 ciclos, sendo 480 ciclos correspondentes as linhas, os outros 14 ciclos são especificados no padrão VGA como tempo de guarda para o traço. Após este período o sinal de sincronismo passa para

estado baixo, por 2 ciclos e torna a nível alto por mais 29 ciclos, totalizando assim 525 sinais de sincronismo horizontal.

Estes dois sincronismos são de polaridade negativa, então quando o sincronismo horizontal vai pra nível “0”, baixo, significa que acabou uma linha. E quando o sincronismo vertical vai para nível baixo, significa que acabou um quadro, a representação de uma tela.

Este bloco possui 11 portas de saída e 5 portas de entrada, os sinais RGB ocupam 3 portas de saída e três de entrada com 10 bits cada uma. As portas de saída são interligadas com o conversor D/A da saída VGA, os de entrada são ligados no bloco seleção de memória, por elas são recebidos os valores correspondentes ao RGB dos pixels que estão gravados na memória.

4 ADAPTAÇÕES E RESULTADOS OBTIDOS

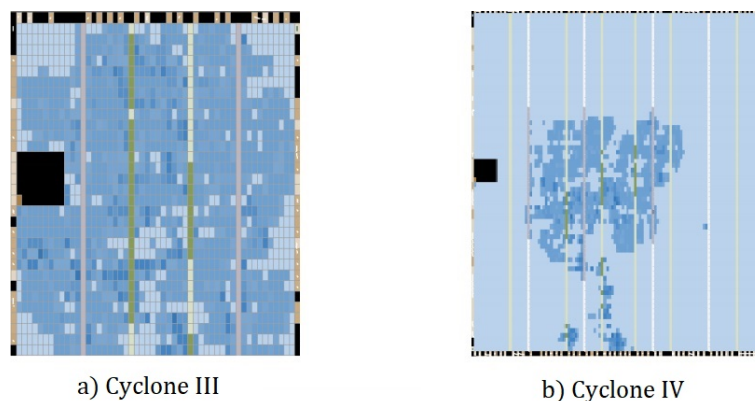
Neste capítulo será apresentada a forma como o código do projeto DE2_115_TV versão 2 foi adaptada em relação a sua versão inicial e os resultados obtidos através do uso da versão adaptada. Além disto, foram realizadas comparações que exemplificam o quanto a adaptação reduziu a lógica empregada.

4.1 Adaptações

O código utilizado como base, conforme citado anteriormente, foi o DE2_115_TV, que foi adaptado por possuir diversos blocos e funcionalidades que não se aplicam neste projeto, como, por exemplo, um barramento I2C, memória Eeprom e codec de áudio. Desta forma, algumas destas funcionalidades foram removidas e outras foram configuradas para ficarem inativas, embora permaneçam no kit.

Uma diferença bastante relevante é que o código I.P.(propriedade intelectual) da Altera Inc. utiliza um FPGA da família Cyclone III EP3C25Q240C8, que é inferior em termos de capacidade de processamento, pois possui quase 5 vezes menos elementos lógicos que o Cyclone IV, modelo utilizado no projeto. Podemos ver um comparativo entre os FPGA's citados na imagem a seguir.

Figura 17 – Ocupação dos FPGA's



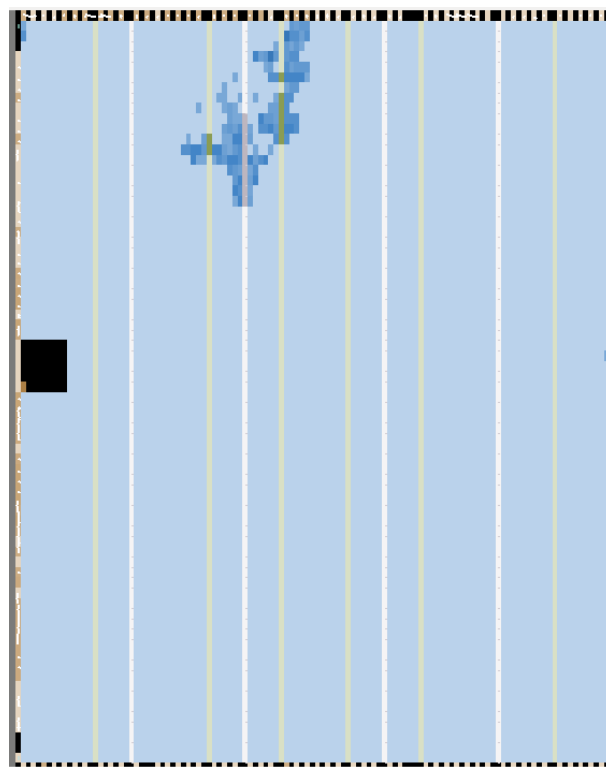
Fonte: Gerado através do software Quartus II

Na imagem da esquerda é apresentado o FPGA da família Cyclone III EP3C25Q240C8, onde fica visível a grande proporção de ocupação dos elementos lógicos, visto que os LUT's em tons de azul claro representam elementos lógicos não utilizados. Já na imagem da direita é apresentado o FPGA da família Cyclone IV EP4CE115F29C7, onde fica evidente

que este modelo possui muito mais elementos lógicos quando comparado ao modelo anterior diante da proporção de ocupação, possuindo aproximadamente 114 mil elementos.

É possível comparar ainda, o FPGA Cyclone IV com o código original e com código adaptado, conforme figura [Figura 18](#). O código original utilizou 13.803 elementos lógicos, cerca de 12% da capacidade total do FPGA. Após algumas adaptações, a ocupação foi reduzida para menos de 1%. O número de bits de memória também obteve uma redução significativa, passando de 7% para exatos 1%.

Figura 18 – FPGA Cyclone IV código adaptado



Fonte: Gerado através do software Quartus II

Para uma melhor observação dos resultados, foi gerada a tabela a seguir:

Tabela 1 – Comparativo de Resultados

Códigos	Original	Adaptado
Elementos Lógicos Totais	13.803/114.480	1.490/114.480
Funções Combinacionais Totais	13.315/114.480	1.189/114.480
Registradores Totais	2.042	978
Pinos Totais	107	101
Total de bits de memória	267.232/3.981K	45.024/3.981K

4.2 Testes

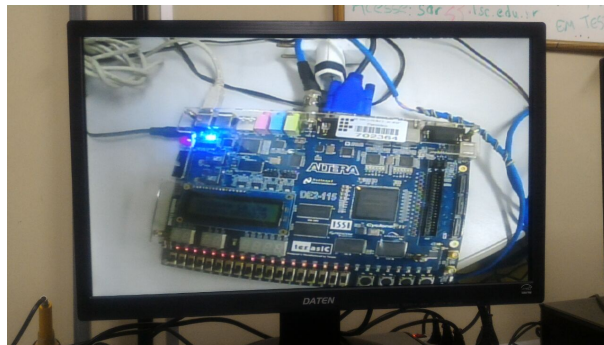
Para comprovar o funcionamento do sistema, o mesmo foi submetido a alguns testes, onde testamos o mesmo código com cameras e monitores de diferentes fabricantes afim de verificar se haveria alterações ou falhas que tenham passadas despercebidas no decorrer do desenvolvimento do sistema. A seguir podemos ver as câmeras (para ver imagens das câmeras consultar apêndice A) e monitores utilizados.

- Câmeras
 - CT 840 VDF da fabricante CS;
 - HMC-BU106-30B da fabricante HDL;
 - CD 1030 da fabricante JFL;

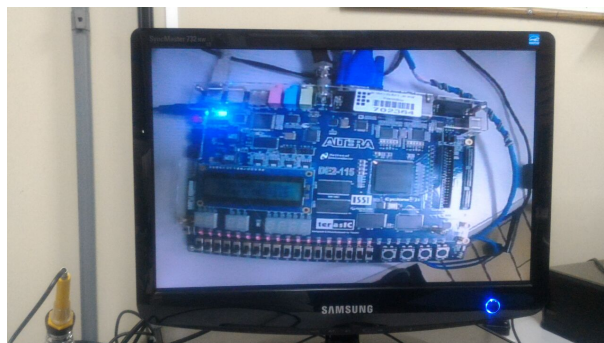
- Monitores
 - 22MP55PQ - 8 da fabricante Daten;
 - 732NW da fabricante Samsung;
 - F19L da fabricante AOC;

Nos testes realizados foi percebido algumas alterações de cores nas imagens variando de acordo com o modelo do monitor e câmera utilizada, porém todas as imagens exibidas estavam de acordo com o esperado. Nas imagens a seguir podemos ver fotos elaboradas dos monitores com sistema operando.

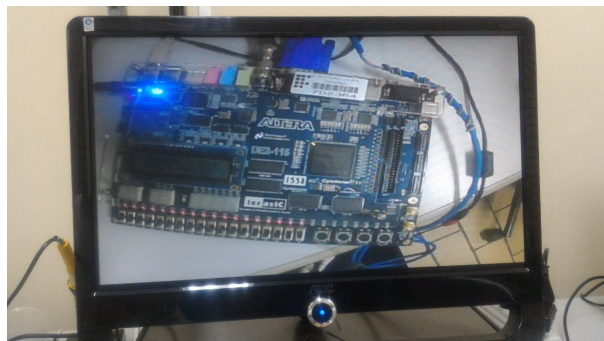
Figura 19 – Exibição das imagens Obtidas
[Daten]



Fonte: Imagem elaborada pelo autor.
[Samsung]



Fonte: Imagem elaborada pelo autor.
[AOC]



Fonte: Imagem elaborada pelo autor.

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi apresentado um sistema de captação de imagem analógica, descrita em linguagens de descrição de hardware para o uso em componentes reprogramáveis do tipo FPGA.

Após o término do projeto podemos concluir que o objetivo foi alcançado de forma satisfatória, pois foi cumprida a proposta inicial. O sistema foi testado com câmeras e monitores de diferentes fabricantes já citados anteriormente.

Os blocos foram implementados usando linguagem VHDL e Verilog, esquemático de blocos e IPs da Altera®. A maior dificuldade encontrada foi fazer a interação do módulo de aquisição com o de exibição de vídeo, devido aos mesmos trabalharem com frequências diferentes. Para solucionar este problema utilizamos uma memória de alto grau de complexidade. Com a adaptação brusca no código, foram identificadas diferenças entre as imagens exibidas de acordo com o modelo e marca dos fabricantes das câmeras e monitores. Entretanto, essas pequenas variações mantiveram-se dentro do esperado e são irrelevantes em relação a usabilidade do sistema, de forma que não interferem de maneira negativa na visualização.

5.1 Trabalhos Futuros

Este trabalho de conclusão de curso não configura somente uma pesquisa, visto que seu objetivo foi, principalmente, a implementação de melhorias a sistemas já existentes. Entretanto, o conhecimento reunido torna-se uma plataforma para captura e processamento de imagens mostrados através de uma interface VGA que poderá ser usada amplamente nos cursos da área de telecomunicações e afins no IFSC.

A seguir serão listados alguns possíveis trabalhos:

- Realizar um sistema de armazenamento das imagens para consulta posterior, tendo em vista que o trabalho só exibe imagens em tempo real;
- Prover um sistema para detecção de objetos nas imagens obtidas;
- Prover mais entradas de vídeo, podendo assim alternar entre as imagens exibidas por cada entrada;
- Implementar uma entrada de áudio, para uso de câmeras com microfones imbutidos.

REFERÊNCIAS

[1] BOCHETTI, M. R. Mecanismo de reconfiguração dinâmica aplicados ao projeto de um processador de imagens reconfiguráveis 2004.

[2] Glaucio Bertelli Peres. Sistema de análise de sinais: Módulo de interface VGA em FPGA São José, SC, Março 2012.

[3] João Donato da S. Petroni. Interface para processamento de imagens implementada em FPGA São Carlos, SP, Fevereiro 2015.

[4] PEDRONI, V. A. Eletrônica Digital Moderna e VHDL: Princípios Digitais, Eletrônica Digital, Projeto Digital, Microeletrônica e VHDL. 2010.

[5] Newton C. Braga. TV Analógica (TEL074). 2003.

[6] Rafael C. González and Richard E. Woods. Digital Image Processing Addison-Wesley. 1993.

[7] João Paulo S. dos Santos. Sistema Embarcado para Rastreamento de Imagens. Brasília, DF, Junho 2009.

[8] Altera Corporation. DE2-115 Manual, Disponível em: <https://www.ee.ryerson.ca/courses/coe608/labs/DE2_115_User_Manual.pdf> Acesso em: Dezembro de 2016.

[9] SERVIDOR LAPLACE. Ambiente de Testes para Qualidade de Experiência em Redes para Transmissão de Televisão sobre IP (IPTV), Disponível em: <<http://laplace.eletrica.ufpr.br/>> Acesso em: Março de 2017

[10] Telecommunication Standardization Sector (ITU-T). BT.656, Disponível em: <<https://www.itu.int/rec/R-REC-BT.656-5-200712-I/en>> Acesso em: Março de 2017

[11] REIS, Felipe. Detecção de cortes de cena, Disponível em: <<https://web.fe.up.pt/ee99035/re>> Acesso em: Agosto de 2017

[12] POYNTON, Charles. Frequently Asked Questions about Color, Disponível em: <http://www.compression.ru/download/articles/color_space/colorfaq.pdf> Acesso em: Agosto de 2017

[13] UNIVERSITY of TEXAS at DALLAS. Verilog HDL Operands, Disponível em: <https://www.utdallas.edu/~akshay.sridharan/index_files/Page4933.htm> Acesso em: Agosto de 2017

[14] Terasic, Altera DE2-115 Development and Education Board, Disponível em: <<http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=EnglishCategoryNo=139No=502>>

-
- [15] João Manuel Brisson Lopes, Cor e Luz, Disponível em: <<http://disciplinas.ist.utl.pt/leiccg/textos/livro/Cor.pdf>> Dezembro 2008.
- [16] Unifore, Disponível em: <<https://www.hkvstar.com/technology-news/cctv-tips-tv-lines-analog-vs-pixels-digital-resolution.html>>
- [17] Info Cellar, Disponível em: <<http://www.infocellar.com/hardware/ga.htm>>
- [18] Daniel Omar Basconcello Filho, Robotizando Tecnologia, Disponível em: <http://www.robotizando.com.br/pinagem_vga.php>
- [19] Fernando D. Garcia, Controlador VGA - Parte 1, Disponível em: <<https://www.embarcados.com.br/controlador-vga-parte-1/>> 2015.
- [20] Alexandre Magalhães, VGA (Video Graphics Array) – Parte I, Disponível em: <<https://agetechology.wordpress.com/2016/03/23/vga-video-graphics-array-parte-i/>> 2016.

Apêndices

APÊNDICE A – CÂMERAS

Figura 20 – CT 840 VDF da fabricante CS



Fonte: Elaborada pelo Autor.

Figura 21 – HMC-BU106-30B da fabricante HDL



Fonte: Elaborada pelo Autor.

Figura 22 – CD 1030 da fabricante JFL



Fonte: Elaborada pelo Autor.