

# Editor de Imagens com Basys3

## Relatório do Trabalho Final

# SEDR

Julho, 03.07.2019

---

### Disciplina

Sistemas Eletrônicos Digitais Reconfiguráveis

Professor Jardel Silveira

### Alunos

Michelly Karen Diógenes Pereira 385211

Rodrigo Silva Lima 385216

2019

Fortaleza - Ceará

Universidade Federal do Ceará

---

## Sumário

I.	Introdução.....	3
II.	Material Utilizado.....	5
III.	Metodologia.....	5
	A. Filtragem Digital	
	B. Módulo BRAM	
	C. Filtros	
	D. Módulo VGA	
IV.	Considerações Finais.....	15
	A. Relatórios de Utilização na FPGA.....	15
	B. Considerações Finais.....	16

## I. Introdução

Na era digital em que vivemos, são poucos os profissionais da fotografia e quase nenhuma agência de publicidade, jornalismo, e marketing que usam imagens sem edição final. Isto acontece porque os programas de edição melhoram a qualidade da imagem ou até mesmo podem alterá-la retirando ou acrescentando efeitos, personagens e objetos da foto, o que enriquece a imagem que geralmente é vendida para revistas, agências e meios de comunicação, todas as ferramentas computacionais que trabalham com edição de imagem utilizam algoritmos de processamento digital de imagens, um exemplo pode ser visto da Imagem 1.0.

Processamento de imagem é qualquer forma de processamento de dados no qual a entrada e saída são imagens tais como fotografias ou quadros. A maioria das técnicas envolve o tratamento da imagem como um sinal bidimensional, no qual são aplicados padrões de processamento de sinal. O resultado deste processamento não precisa ser necessariamente uma imagem, mas pode ser uma instância de certas características desta imagem.



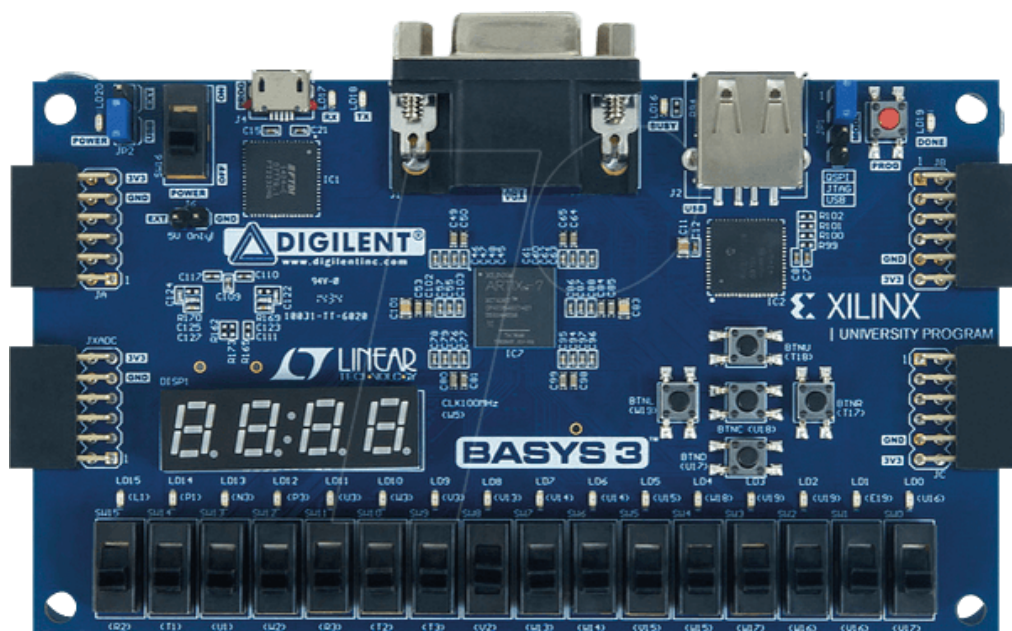
**Imagem 1.0** - Famosa foto de Lena sendo usada para processamento digital de imagens com diferentes efeitos

Esse trabalho desenvolvido dentro da disciplina de Sistemas Eletrônicos Digitais Reconfiguráveis apresenta um sistema de processamento digital de imagens baseado em hardware reconfigurável implementado em uma FPGA, que é um dispositivo semiconductor largamente utilizado para o processamento de informações digitais e pode ser programado de acordo com as aplicações do usuário.

Para definir o comportamento da FPGA o usuário (programador) pode optar por fazer utilizando uma linguagem de descrição de hardware (HDL), como VHDL ou Verilog, um desenho esquemático ou em linguagem de alto nível, como C ou C++. Então usando uma ferramenta como o Vivado ou Vivado HLS esta lógica é aplicada à arquitetura da FPGA. O sistema proposto envolve um computador para a leitura e pré-processamento da imagem. Esta imagem é enviada através de uma interface paralela à FPGA que aplicará operações, filtros e transformadas sobre ela, de acordo com o que o usuário desejar através de um conector Video Graphics Array (VGA) presente na Basys 3 que é a FPGA utilizada no projeto.

A Basys 3 vista na Imagem 1.1, utilizada no projeto, é uma placa de desenvolvimento FPGA de nível básico projetada exclusivamente para o Vivado Design

Suite, com arquitetura FPGA Artix-7 da Xilinx. A Basys 3 é a mais nova adição à popular linha Basys de placas de desenvolvimento FPGA, e é perfeitamente adequado para estudantes ou iniciantes apenas começando com a tecnologia FPGA. O Basys 3 inclui os recursos padrão encontrados em todas as placas Basys: hardware completo pronto para uso, uma grande coleção de dispositivos de I/O integradas, todos os circuitos de suporte FPGA necessários, uma versão gratuita de ferramentas de desenvolvimento e um aluno ponto de preço de nível.



**Imagem 1.1 - Basys 3.**

Fonte: <https://www.xilinx.com/products/boards-and-kits/1-54wqge.html>

## II. Material Utilizado

- A. Computador Pessoal
- B. Cabo VGA
- C. Basys3
- D. Software Vivado HLX  
(disponível em <https://www.xilinx.com/support/download.html>)

## III. Metodologia

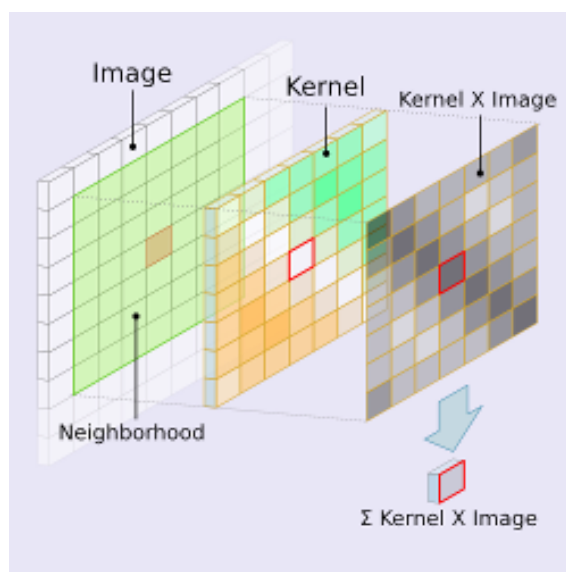
Primeiramente foi realizada uma pesquisa com o intuito de identificar quais algoritmos de processamento de imagens seriam utilizados, após isso iniciamos a implementação dos mesmos, em paralelo, estudamos as práticas realizadas em sala de aula para podermos implementar a possibilidade de visualização da imagem processada através do cabo VGA pelo computador.

A seleção dos efeitos se dá pelos botões de nível presentes na placa de acordo com a tabela abaixo.

Efeito	Seleção (Três botões menos significativos)	Nível (Nove botões mais significativos)
Original	0000	Não se aplica
Aumentar Brilho	0001	de 000000000 à 111111111
Diminuir Brilho	0010	de 000000000 à 111111111
Escala de Cinza	0011	Não se aplica
Desfoque Médio	0100	Não se aplica
Deteção Borda Sobel	0101	Não se aplica
Deteção de Borda	0110	Não se aplica
Desfoque em X	0111	Não se aplica
Relevo	1000	Não se aplica
Afinamento	1001	Não se aplica

### a. FILTRAGEM DIGITAL

Usamos técnicas de Filtragem Digital, que é um conjunto de técnicas destinadas a corrigir e realçar uma imagem. A correção é a remoção de características indesejáveis, e a melhoria/realce é a acentuação de características desejáveis. As formas de Implementação da Filtragem são por Convolução no domínio espacial e por Análise de Fourier no domínio da frequência. Convolução é o processo de calcular a intensidade de um determinado pixel em função da intensidade de seus vizinhos. Para alguns efeitos como Afinamento, Contorno e Relevô, foram utilizados Kernels. Podemos ver o funcionamento de um processamento de imagem com um Kernel na Imagem 3.0.



**Imagem 3.0 - Kernel x Imagem**

### b. MÓDULO BRAM

Vale ressaltar que no projeto foi instanciado o módulo BRAM no Vivado HLX. A BRAM é um bloco de módulo de memória que tem uma quantidade de endereços igual ao número de linhas da imagem e 24 bits de dados que representa a altura, esse módulo é instanciado como um IP CORE, usando o *IP Catalog*. Essa BRAM é utilizada para armazenar a imagem de entrada do processamento.

### c. FILTROS

Os filtros utilizados foram:

- Imagem Original
- Imagem em Escala de Cinza
- Ajustes de Brilho
- Detecção de Borda de Sobel
- Desfoque de Imagem
- Desfoque de Movimento
- Afinamento de Imagem
- Contorno
- Relevô

#### 1. Imagem Original

Nessa opção a imagem salva na BRAM instanciada na FPGA é reproduzida na saída do processamento sem nenhuma alteração.

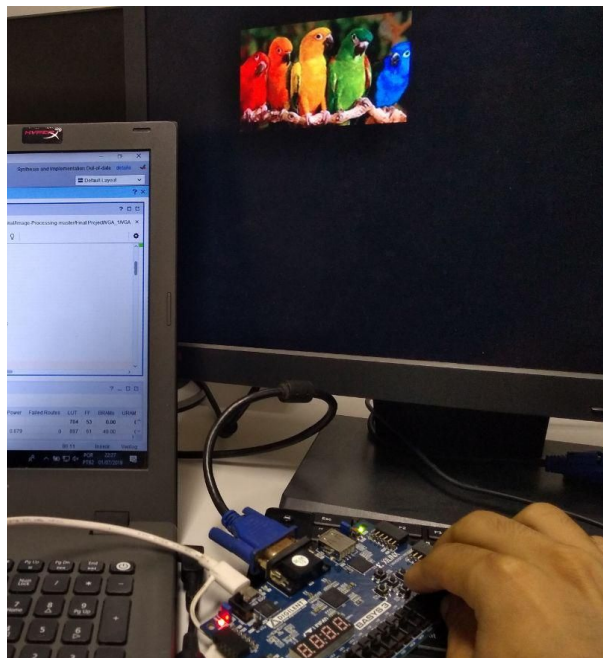


Imagem de teste Original

## 2. Imagem em Escala de Cinza

Converte os valores dos pixels verde, vermelho e azul (RGB) em um valor correspondente na escala de cinza (0 à 255).

$$EscalaCinza = (Red * 0.3) + (Green * 0.4) + (Blue * 0.2)$$

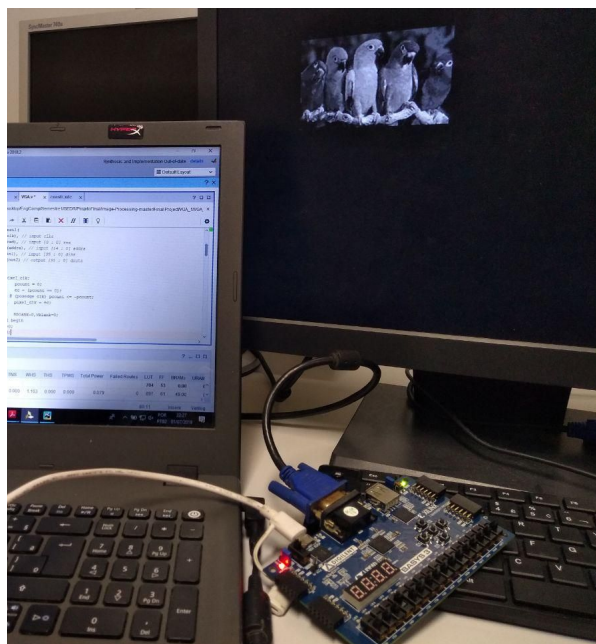


Imagem de teste em Escala de Cinza

## 3. Ajustes de Brilho

Para realizar o ajuste de brilho somamos ou subtraímos um certo valor, que pode ser ajustado na FPJA, dos valores dos pixels RGB da imagem de entrada. Implementamos ao total 512 níveis de intensidades de brilho diferentes, em que 0 é o nível mais baixo e 511 o nível mais alto. Para selecionar o nível do filtro usamos 9 botões de estado (alto ou baixo), tornando possíveis as 512 opções, que devem ser usadas na base binária.

$$CanalAzul = (Blue + / - valor)$$

$$CanalVermelho = (Red + / - valor)$$

$$CanalVerde = (Green + / - valor)$$



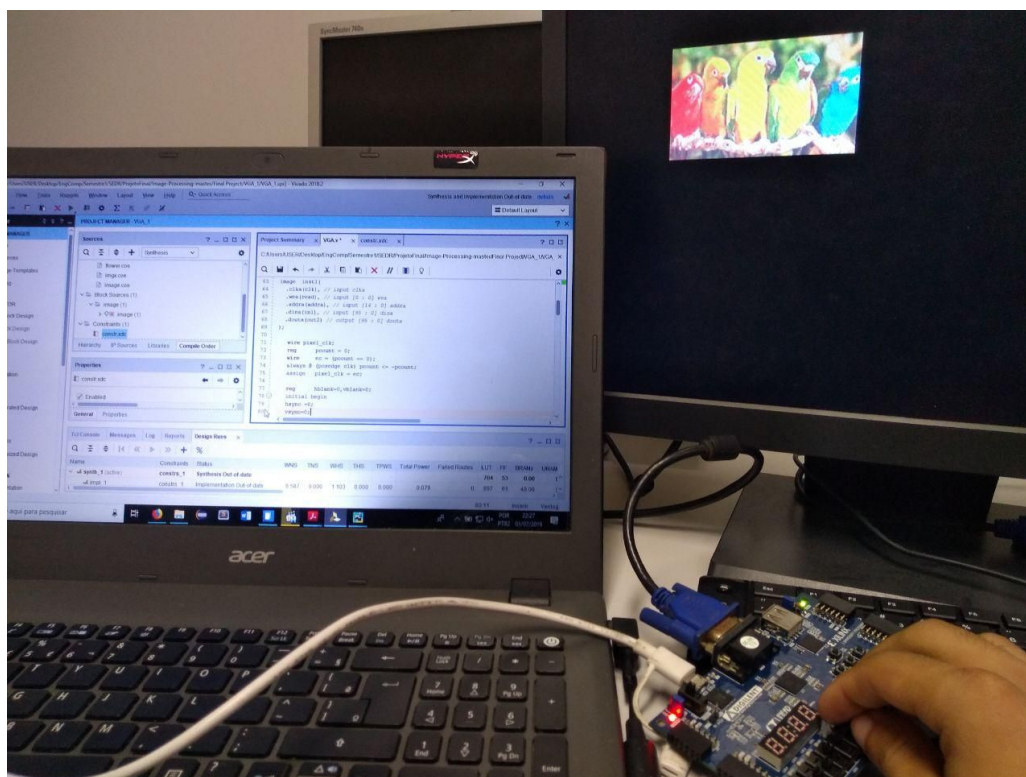


Imagem de teste com aumento de brilho

#### 4. Detecção de Borda de Sobel

Se baseado na implementação desse efeito realizada nos laboratórios, ele faz um destaque de contorno nas bordas da imagem original, o cálculo é baseado em ponderação, isto é, utilizam-se pesos diferentes para pixels vizinhos diferentes. Isso é feito aplicando um Kernel a cada um dos pixels.

-1	0	1
-2	0	2
-1	0	1

Horizontal

-1	-2	-1
0	0	0
-1	-2	-1

Vertical

Imagem 3.1 - Kernel de Sobel

#### 5. Desfoque de Imagem

Nesse efeito é realizado uma média dos valores de todos os pixels adjacentes a cada pixel na imagem.

#### 6. Desfoque de Movimento

Esse efeito realiza o desfoque gerando uma sensação de movimento em um dos eixos, horizontal ou vertical. Assim como o citado anteriormente, ele calcula a média dos pixels adjacentes, mas em apenas uma das direções ao pixel a ser modificado, para criar o desfoque com movimento vertical são usados os valores acima e abaixo do pixel, ao contrário, para criar um desfoque com movimento horizontal são usados os valores de um lado e do outro ao pixel correspondente.

### 7. Afinamento de Imagem

É usado um Kernel para tornar a imagem mais nítida. O Kernel pode ser visto na imagem abaixo.

0	-1	0
-1	5	-1
0	-1	0

**Imagem 3.2** - Kernel para afinamento de imagem

### 8. Contorno

É usado um Kernel para gerar um contorno da imagem. O Kernel pode ser visto na imagem abaixo.

-1	-1	-1
-1	8	-1
-1	-1	-1

**Imagem 3.3** - Kernel para contorno da imagem

### 9. Relevô

É usado um Kernel para criar um efeito de sombras 3D na imagem, dando a sensação de relevo. O Kernel pode ser visto na imagem abaixo.

-2	-1	0
-1	1	1
0	1	2

**Imagem 3.4** - Kernel usado para o efeito de relevo

A imagem de entrada deve ser de 160x115 px, usando a saída VGA da Basys 3 conseguimos ver o resultado final depois do processamento da imagem. O resultado é mostrado em um display de 480x640 px com uma taxa de atualização de 60Hz, para cada atualização cada pixel é atualizado um após o outro

## D. MÓDULO VGA

O módulo VGA usado no projeto foi implementado em Verilog, a sigla VGA vem de Video Graphics Array, um padrão de disposição gráfica para vídeo que utiliza sinais analógicos para gerar as cores.

Esse padrão foi criado pela IBM em 1987 e possibilita que imagens sejam geradas com resolução de 640x480 pixels ou 320x240 pixels. No VGA são utilizados 6 bits para representar cada componente RGB, logo a paleta de cores possui 262.144 cores. Conforme a resolução utilizada apenas uma pequena quantidade de cores da paleta pode ser selecionada, neste caso, na primeira resolução o sistema trabalha com 16 cores, já na segunda com até 256 cores.



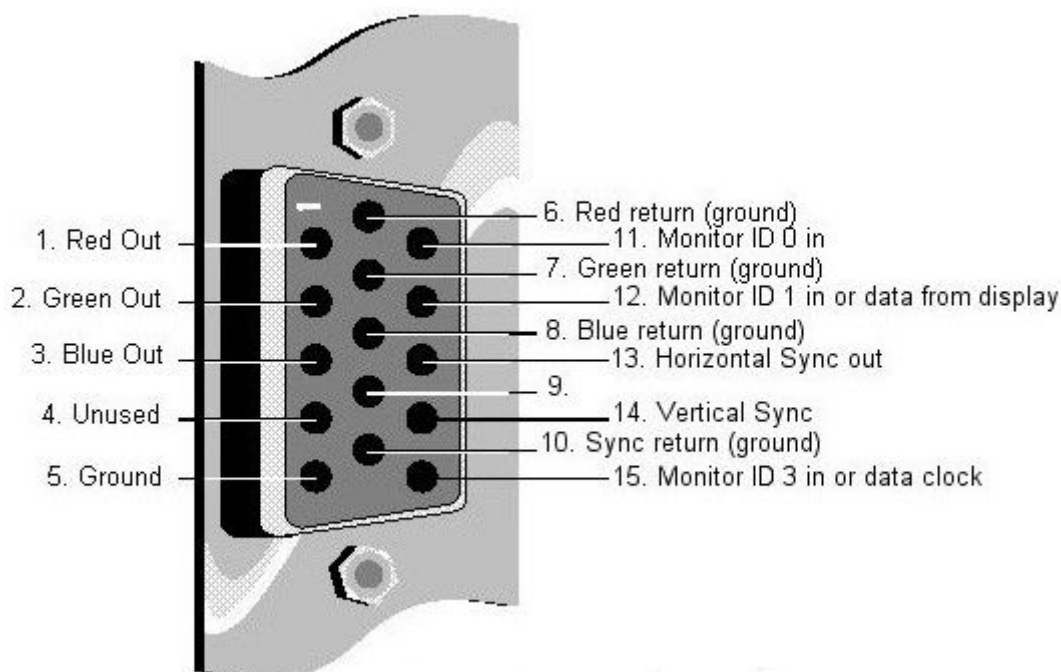
**Imagem 3.5** - Paleta de 256 cores

Logo após o lançamento do padrão algumas variações surgiram com objetivo de aumentar a resolução do vídeo e o número de cores. Entre esses padrões destacam-se o Super VGA (SVGA) com resolução de 800x600 pixels e o eXtended Graphics Array (XGA) com resolução de 1024x768 pixels. Embora vários padrões tenham sido lançados, o nome permaneceu VGA.

No padrão VGA são utilizados 5 sinais de controle. Esses sinais estão presentes na maioria dos sistemas na porta VGA, isto é, no conector VGA.

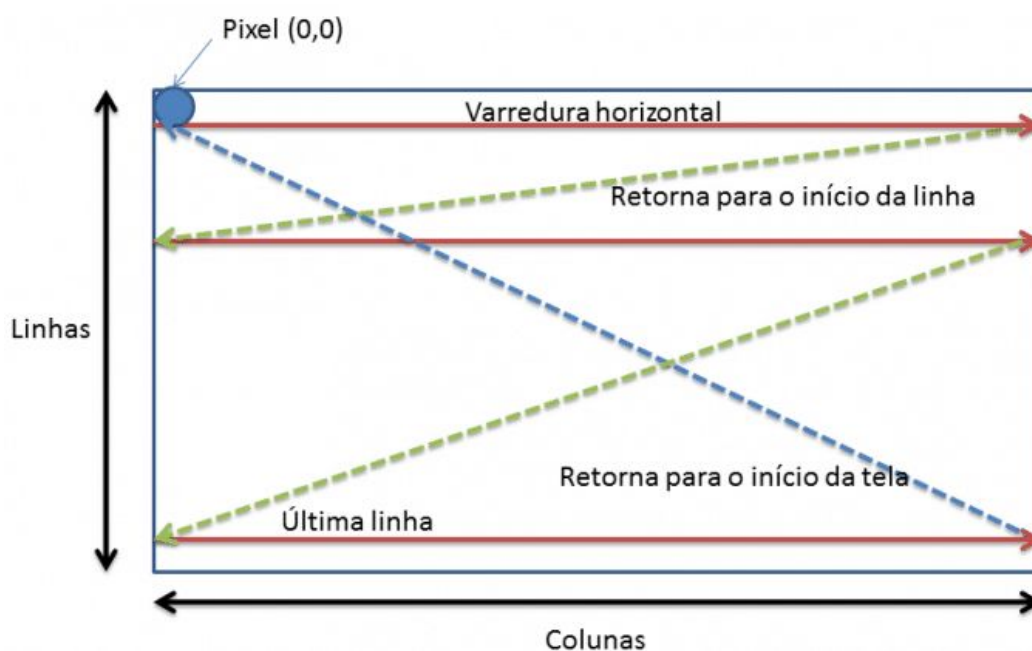


**Imagem 3.6** - Conector VGA



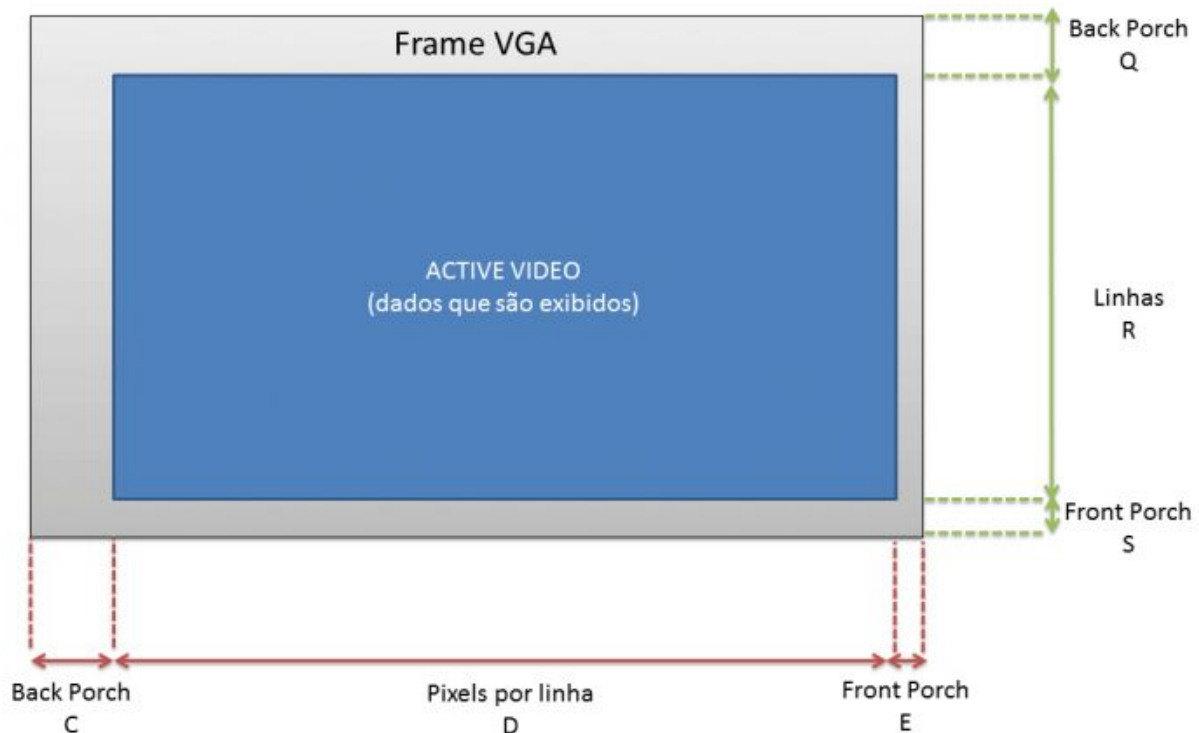
**Imagem 3.7** - Pinagem do conector VGA

Dos pinos mostrados na Figura 3 somente cinco são utilizados no controle de vídeo. São os pinos: 1, 2, 3, 13 e 14. Os pinos 1, 2 e 3 são utilizados para determinar a cor (RGB) de um determinado pixel, já os pinos 13 e 14 são utilizados na sincronização do vídeo. Cabe frisar que R, G e B são sinais analógicos, isto é, a cor varia conforme a intensidade do sinal. Na imagem 3.8 podemos ver o funcionamento da varredura do quadro VGA.



**Imagem 3.8** - Controlador VGA, varredura do quadro VGA

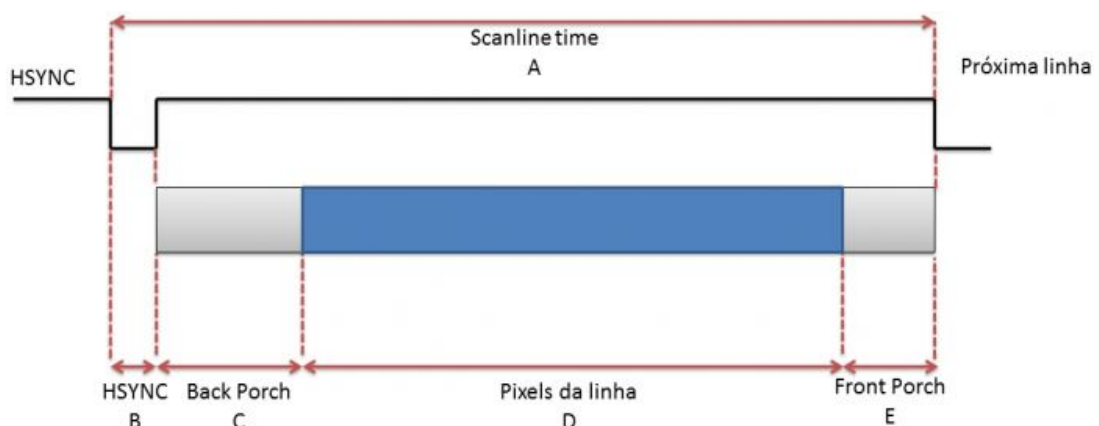
Um frame VGA define um conjunto de parâmetros que devem ser seguidos para sincronizar o vídeo com o mecanismo de varredura da tela. A Imagem 3.9 exibe a região ativa de vídeo que corresponde à tela exibida.



**Imagem 3.10 - Frame VGA**

Convém observar que além da região ativa de vídeo outros parâmetros de tempo devem ser respeitados. Tanto na varredura horizontal quanto na vertical são estabelecidos alguns parâmetros como: Back Porph e Front Porph. Esses parâmetros estão relacionados com os sinais Horizontal Sync (**H\_SYNC**) e Vertical Sync (**V\_SYNC**) que são utilizados para sincronizar a varredura da tela.

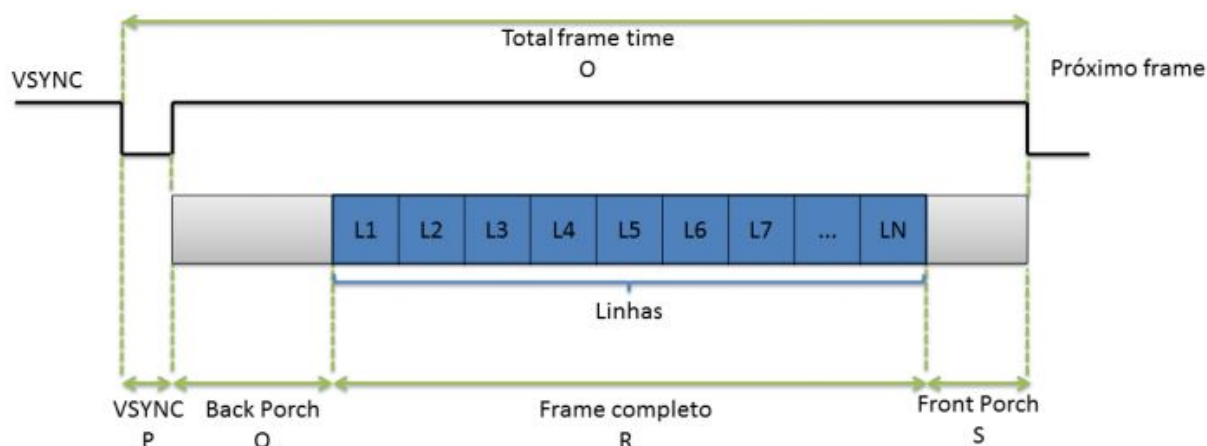
O sincronismo de varredura horizontal é determinado pelo sinal **H\_SYNC**. O sinal **H\_SYNC** é ativado em nível lógico 0 e indica que uma linha foi finalizada e que próxima linha será iniciada. O funcionamento da sincronização horizontal do frame VGA pode ser vista na Imagem 3.11.



**Imagem 3.11 - Sincronização horizontal do frame VGA**



O sincronismo de varredura vertical é determinado pelo sinal **V\_SYNC**. O sinal **V\_SYNC** é ativado em nível lógico 0 e indica que um quadro foi finalizado e que o próximo será iniciado. Um quadro é um conjunto N de linhas, sendo que cada linha possui M colunas.



**Imagem 3.12** - Sincronização vertical do frame VGA

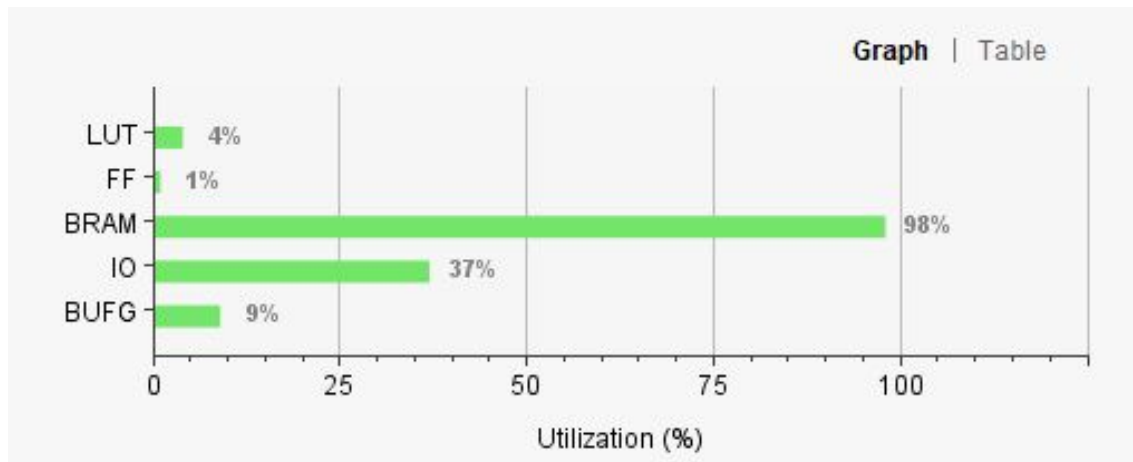
Considerando que um sistema para controlar um dispositivo VGA tenha uma base de tempo precisa, pode-se utilizar a tabela abaixo para determinar o período dos parâmetros de controle. A **Tabela 1** exibe a relação de pulsos de clock necessários para cada parâmetro (B, C, D, E, P, Q, R e S) conforme a resolução da tela e a taxa de atualização. Como usamos o formato 640x480 a taxa de atualização determinada é de 60Hz.

Format	Pixel Clock (MHz)	Horizontal (in Pixels)				Vertical (in Lines)			
		Active Video	Front Porch	Sync Pulse	Back Porch	Active Video	Front Porch	Sync Pulse	Back Porch
640x480, 60Hz	25.175	640	16	96	48	480	11	2	31
640x480, 72Hz	31.500	640	24	40	128	480	9	3	28
640x480, 75Hz	31.500	640	16	96	48	480	11	2	32
640x480, 85Hz	36.000	640	32	48	112	480	1	3	25
800x600, 56Hz	38.100	800	32	128	128	600	1	4	14
800x600, 60Hz	40.000	800	40	128	88	600	1	4	23
800x600, 72Hz	50.000	800	56	120	64	600	37	6	23
800x600, 75Hz	49.500	800	16	80	160	600	1	2	21
800x600, 85Hz	56.250	800	32	64	152	600	1	3	27
1024x768, 60Hz	65.000	1024	24	136	160	768	3	6	29
1024x768, 70Hz	75.000	1024	24	136	144	768	3	6	29
1024x768, 75Hz	78.750	1024	16	96	176	768	1	3	28
1024x768, 85Hz	94.500	1024	48	96	208	768	1	3	36

**Tabela 1** - Temporização VGA.

## IV. Conclusão

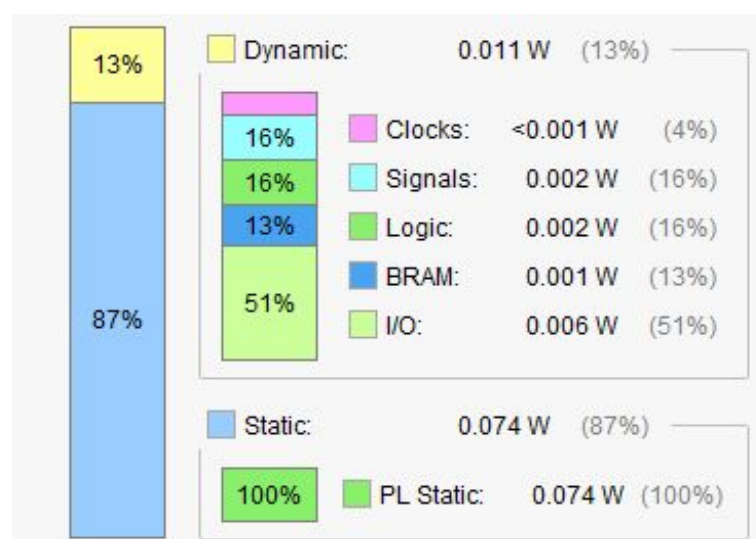
### A. Relatório de Utilização



**Gráfico 1 - Utilização dos Recursos**

<b>Total On-Chip Power:</b>	<b>0.085 W</b>
<b>Junction Temperature:</b>	<b>25,4 °C</b>
Thermal Margin:	59,6 °C (11,8 W)
Effective $\theta_{JA}$ :	5,0 °C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low
<a href="#">Implemented Power Report</a>	

**Imagem 4.0 - Características da Implementação**

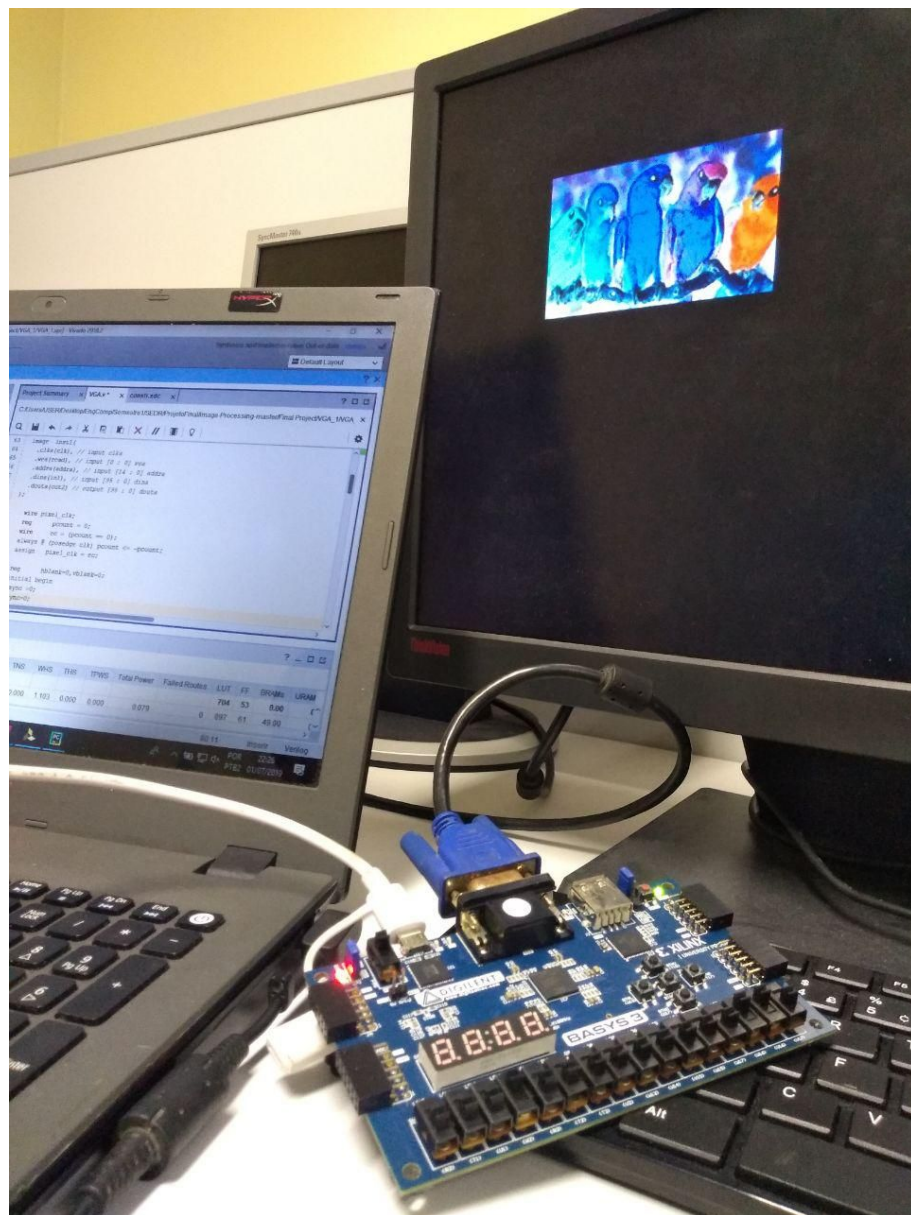


**Imagem 4.1 - Detalhamento da potência utilizada**

## B. Considerações Finais

Concluímos que o processamento digital de imagem requer um conhecimento sobre processamento de sinais, esse conhecimento foi uma das principais dificuldades que encontramos, por outro lado conseguimos grande apoio na literatura, com Kernels já consagrados e populares, o que facilitou bastante o desenvolvimento do projeto.

Por fim, conseguimos alcançar o nosso objetivo fazendo o processamento da imagem com os efeitos desejados e reproduzindo-as na tela de um computador através do conector VGA, o controle e seleção de efeitos são realizados exclusivamente pelos botões de estado presentes na Basys 3.



**Imagem 4.2** - Exemplo de testes de filtros no projeto