UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA DE SÃO CARLOS
DEPARTAMENTO DE FÍSICA E INFORMÁTICA
GRUPO DE FÍSICA COMPUTACIONAL E INSTRUMENTAÇÃO APLICADA

Mario Alexandre Gazziro

*Desenvolvimento e implementação de
instrumentação eletrônica para criação de
estímulos visuais para experimentos com o
duto óptico da mosca*

São Carlos, SP

Julho de 2009

Mario Alexandre Gazziro

*Desenvolvimento e implementação de
instrumentação eletrônica para criação de
estímulos visuais para experimentos com o
duto óptico da mosca*

Tese apresentada ao Programa de Pós-Graduação
em Física do Instituto de Física de São Carlos
da Universidade de São Paulo, como parte dos
requisitos para obtenção do título de Doutor em
Ciências.

Área: Física Aplicada - Computacional.

Orientador: Prof. Dr. Jan F.W. Slaets

São Carlos, SP

Julho de 2009

FOLHA DE APROVAÇÃO

Mario Alexandre Gazziro

Tese apresentada ao Instituto de Física de São Carlos da Universidade de São Paulo para obtenção do título de Doutor em Ciências. Área de Concentração: Física Aplicada – Opção: Física Computacional.

Aprovado em: 23/09/2009

Comissão Julgadora

Prof. Dr. Jan Frans Willem Slaets

Instituição: UFSCar

Assinatura

Prof. Dr. Sergio Sheiji Fukusima

Instituição: FFCLRP/USP

Assinatura

Prof. Dr. Valentin Obac Roda

Instituição: EESC/USP

Assinatura

Prof. Dr. Jose Hiroki Saito

Instituição: UFSCar

Assinatura

Dr. Mateus Jose Martins

Instituição: IFSC/USP

Assinatura

*Aos meus pais,
Mario e Bene*

Agradecimentos

Ao meu orientador, Prof. Jan Frans Willem Slaets, pela paciência, por seu apoio e direcionamento preciso durante a execução do presente trabalho, sem o qual o mesmo não seria possível. Ao Prof. Roland Köberle, por sua natureza científica impecável, sendo uma inspiração à todos que desejem encontrar algo novo em meio ao desconhecido. Aos professores Eduardo Marques, José Saito, Valentin Roda, Guilherme Sipahi, pelo apoio e acompanhamento durante minha carreira acadêmica na última década. Embora mais recente, devo agradecer também ao Prof. Reynaldo Daniel, pois sua empolgação pela neurociência serve como inspiração à todos que atuam na área. Em especial à Lirio Almeida, dinossauro do Rock'n'Roll e guru da eletrônica, sendo que o presente trabalho é fruto da continuidade dos grandes avanços obtidos por ele no campo da eletrônica instrumental de apoio à neurociência da visão da mosca. Aos amigos do laboratório Dipteralab: Nelson Fernandes, Ingrid Miranda, Paulo Matias, Nataly Horner, Rafael Viegas, Ivanilda Stéfani e Ailton Alves, os quais trabalham com extrema dedicação nesta área de pesquisa a qual permeia as fronteiras da ciência conhecida. A todos os funcionários do IFSC, em especial à Sílvia Athayde e Victor Barioto, pelo profissionalismo e pronto atendimento. Aos professores do Instituto Superior Técnico de Lisboa, João Cardoso e Pedro Diniz, pelo apoio durante meu estágio no exterior. Aos amigos recentes Caio Assunção, Carlos Eduardo, Nicole Novaes, Dalmir Rapelli, Paula Neves, Adriano Sanches, Luis Uebel e Tatiana de Alencar e aos antigos, Raul Habesch, Alexandre Medeiros, Richard Lingner e Augusto Carbol e Elias Neto. À Izis Cavalcanti, por seu trabalho e dedicação na criação de nossos filhos, Caio e Alice, os quais me proporcionam imensuráveis momentos de felicidade. Agradeço por fim às agências financiadoras FAPESP, CAPES e CNPq, respectivamente pelo apoio à compra dos materiais utilizados, pela realização do meu estágio no exterior e pelo meu treinamento em projeto de circuitos integrados digitais.

*"He who controls the past
commands the future."
George Orwell*

Resumo

GAZZIRO, M. **Desenvolvimento e implementação de instrumentação eletrônica para criação de estímulos visuais para experimentos com o duto óptico da mosca.** 2009. 87p. Tese (Doutorado) – Instituto de Física de São Carlos – Universidade de São Paulo, São Carlos, 2009.

O presente trabalho descreve o desenvolvimento de geradores de estímulos visuais para serem utilizados em experimentos de neurociência com invertebrados, tais como moscas. O experimento consiste na visualização de uma imagem fixa que é movida horizontalmente de acordo com os dados de estímulo recebidos. O sistema é capaz de exibir 640x480 pixels com 256 níveis intensidade a 200 frames por segundo em monitores de varredura convencional. É baseado em hardware reconfigurável (FPGA), incluindo a lógica para gerar as temporizações do vídeo, dos sinais de sincronismo, assim como da memória de vídeo. Uma lógica de controle especial foi incluída para atualizar o deslocamento horizontal da imagem, de acordo com os estímulos desejados, a uma taxa de 200 quadros por segundo. Em um dos geradores desenvolvidos, a fim de duplicar a resolução de posicionamento horizontal, ‘passos artificiais entre-pixels’ foram implementados usando dois frame buffers de vídeo, contendo respectivamente os pixels ímpares e pares da imagem original a ser exibida. Esta implementação gerou um efeito visual capaz de dobrar a capacidade de posicionamento horizontal deste gerador.

Palavras-chave: Neurobiofísica. Instrumentação eletrônica. FPGA. Sistema visual da mosca. Processamento de video em tempo real.

Abstract

GAZZIRO, M. **High-performance visual stimulation system for use in neuroscience experiments with the blowfly**. 2009. 87p. PhD Thesis – Instituto de Física de São Carlos – Universidade de São Paulo, São Carlos, 2009.

This thesis describes the development of many visual stimulus generators to be used in neuroscience experiments with invertebrates such as flies. The experiment consists in the visualization of a fixed image which is moved horizontally according to the received stimulus data. The system is capable to display 640x480 pixels with 256 intensity levels at 200 frames per second on conventional raster monitors. It's based on reconfigurable hardware (FPGA), includes the logic to generate video timings and synchronization signals as well as the video memory. Special control logic was included to update the horizontal image offsets according to the desired stimulus data at 200 fps. In one of the developed generators, with the intent to double the horizontal positioning resolution, 'artificial inter-pixel' steps are implemented using two video frame buffer containing respectively the odd and the even pixels of the original image to be displayed. This implementation generates a visual effect capable to double the horizontal positioning capabilities of the generator.

Keywords: Neurobiophysics. Electronic instrumentation. FPGA. Fly's visual system. Real-time video processing.

Lista de Figuras

Figura 1 – Olho composto da mosca	16
Figura 2 – Como o olho da mosca difere dos demais artrópodes	17
Figura 3 – Tempo de resposta dos foto-receptores	18
Figura 4 – Corte de seção do olho composto	20
Figura 5 – Sensitividade binocular dos neurônios H1	21
Figura 6 – Estrutura interna do monitor CRT com deflexão eletromagnética . .	22
Figura 7 – Tempo de ativação do fósforo em uma tríade do monitor RGB	23
Figura 8 – Gerador de estímulos visuais Picasso	24
Figura 9 – Padrão gerado pelo Picasso	24
Figura 10 – Gerador de estímulos visuais ViSaGe	25
Figura 11 – Gerador de estímulos visuais FliMax	26
Figura 12 – Gerador de estímulos visuais VSIImG	26
Figura 13 – Diagrama em bloco do experimento completo.	27
Figura 14 – Mosca imobilizada para o experimento	28
Figura 15 – Cabeça da mosca exposta	29
Figura 16 – Inserção do micro-eletrodo extracelular	30
Figura 17 – Tela do osciloscópio: sinal captado e discriminado	30
Figura 18 – As cinco principais etapas na aquisição do sinal	31

Figura 19 – Posicionamento ideal do micro-eletrodo extracelular	31
Figura 20 – Estímulo: velocidades horizontais	34
Figura 21 – Kit de desenvolvimento DE2 da Terasic	36
Figura 22 – Kit de desenvolvimento DE3 da Terasic	37
Figura 23 – Kit de desenvolvimento ML505 da Xilinx	38
Figura 24 – Gerador de estímulos visuais operando em monitor XY	39
Figura 25 – Arquitetura do gerador de estímulos utilizando uClinux	40
Figura 26 – Relação entre os espaços do usuário e do kernel	41
Figura 27 – Arquitetura do gerador de estímulos baseado em FIFO	42
Figura 28 – Arquitetura da função PULSO	44
Figura 29 – Gerador de estímulos baseado em FIFO em funcionamento	44
Figura 30 – Arquitetura do gerador de dois monitores	45
Figura 31 – Gerador de estímulos de dois monitores em funcionamento	46
Figura 32 – Arquitetura de comunicação e verificação de erros	47
Figura 33 – Controle do cursor em funcionamento	48
Figura 34 – Estrutura dos omatídeos no olho da mosca	49
Figura 35 – Geração das imagens com diferentes amostragens	50
Figura 36 – Detalhe do funcionamento do efeito ‘entre-pixel’	51
Figura 37 – A arquitetura ‘Passos Artificiais Entre-Pixels’	52
Figura 38 – Cena 3D naturalística renderizada pela plataforma Ogre 3D	53
Figura 39 – Arquitetura da integração entre FPGA e GPU	54
Figura 40 – Sistema integrado entre FPGA e GPU em funcionamento	55
Figura 41 – Geração do mesmo estímulo com duas amostragens	58

Figura 42 – Experimento realizado com gerador baseado em FIFO	59
Figura 43 – Raster do experimento realizado pelo gerador baseado em FIFO . . .	60
Figura 44 – Comparação entre os PSTH dois sistemas	61
Figura 45 – Sistema em funcionamento apresentando padrão naturalístico	61
Figura 46 – Histograma de intervalos	62
Figura 47 – Cabo do gerador de estímulos visuais.	79
Figura 48 – RTL do gerador de estímulos baseado em FIFO	81
Figura 49 – RTL da geração do clock de vídeo	81
Figura 50 – RTL do gerador de dois monitores: Parte do kit DE3	82
Figura 51 – RTL do gerador de dois monitores: Parte do kit DE2	83

Sumário

1	Introdução	15
1.1	Objetivos	15
1.2	O olho composto	16
1.3	O sistema visual da mosca	19
1.4	Uso de tubos de raios catódicos em neurociências	21
1.5	Técnicas atuais de estimulação visual de insetos	24
2	O Experimento	27
2.1	Preparação da mosca	28
2.2	Aquisição de dados neurais	30
2.3	O estímulo	34
3	Metodologia	35
3.1	Kits de desenvolvimento utilizados	35
3.2	Primeira abordagem: controle do monitor XY	38
3.3	Segunda abordagem: utilização do NIOS e uClinux	39
3.4	Terceira abordagem: deslocamento da imagem baseado em FIFO	42
3.5	Quarta abordagem: controle de dois monitores	44
3.6	Quinta abordagem: verificação de erros e controle do cursor	46

3.7	Sexta abordagem: ampliação da capacidade horizontal	48
3.7.1	A técnica ‘Passos Artificiais Entre-Pixels’	50
3.7.2	A arquitetura para geração dos ‘Passos Artificiais Entre-Pixels’ . .	51
3.8	Sétima abordagem: integração entre FPGA e GPU	52
4	Resultados e Discussão	57
4.1	‘Deslocamento da imagem baseado em FIFO’	57
4.2	‘Ampliação da capacidade horizontal’	60
5	Conclusões	63
	Referências	65
A	Listagem dos códigos fonte em Matlab	69
B	Listagem dos principais códigos fonte em Verilog	73
C	Cabo de ligação entre FPGA e porta paralela	79
D	Diagramas RTL	81
E	Processo de Ornstein-Uhlenbeck	85
	Lista de abreviaturas e siglas	87

1 Introdução

1.1 Objetivos

Esse trabalho tem por objetivo desenvolver equipamentos geradores de estímulos visuais para utilização em experimentos de neurociência.

Nossa proposta consiste na utilização de monitores convencionais e no uso de tecnologia de hardware reconfigurável. Tal tecnologia deve possibilitar atender os requisitos desejados, listados abaixo:

- Possibilitar que padrões visuais naturalísticos sejam exibidos com taxa de atualização de quadros de 200 Hz;
- Permitir um controle preciso para o deslocamento do estímulo visual, da ordem de mil pontos.

Ambos os requisitos devem ser atingidos com a criação de uma arquitetura gráfica específica para geração de vídeo em tempo-real, contendo um registrador especial para controle do deslocamento. Serão implementados ‘passos artificiais entre-pixels’ usando dois frame buffers de vídeo, contendo respectivamente os pixels ímpares e pares da imagem original a ser exibida. Espera-se que essa implementação produza um efeito visual capaz de dobrar a capacidade de posicionamento horizontal do gerador de estímulos. Posteriormente será realizada uma análise entre o estímulo e os sinais neurais.

O equipamento desenvolvido será validado junto ao laboratório Dipteralab do IFSC-USP, tendo seus resultados comparados com os equipamentos tradicionais utilizados nesse laboratório.

1.2 O olho composto

A partir de fósseis, na maioria trilobitas, nós sabemos que os artrópodes utilizam olhos compostos desde a era cambriana, há cerca de 540 milhões de anos atrás. Olhos compostos produzem uma visão em mosaico do ambiente, devido ao fato de que cada uma das facetas (ou omatídeos) do olho é direcionada para um ponto ligeiramente diferente no espaço (Figura 1). A resolução da imagem fica, então, limitada pelo número de omatídeos e pelo ângulo entre eles.

Figura 1: Olho composto da mosca (1).

Existem ainda condições diferentes para visão durante o dia e a noite. Insetos que devem enxergar bem durante o dia possuem sete ou oito células foto-receptoras agrupadas embaixo de cada lente dos omatídeos. Essas células estão dispostas de forma que suas superfícies coletoras de luz (os rabdômeros) são fundidas no centro de cada omatídeo, criando uma estrutura chamada Rabdoma, semelhante a um bastonete (Figura 2a).

Todo omatídeo é encapsulado em um tubo de células de pigmento, de forma que raios de luz que incidem em ângulos diferentes da orientação do tubo são removidos (Figura 2b). Dessa forma muitos fótons são perdidos, mas a resolução é melhorada.

Figura 2: Como o olho da mosca (d)(e) difere dos olhos dos demais artrópodes (a)(b)(c) (1).

Insetos que precisam enxergar bem durante a noite apresentam menor quantidade de pigmentos no encapsulamento dos Rabdomas, permitindo que a luz proveniente de um campo maior consiga atingir seus foto-receptores (Figura 2c). Isso aumenta a sensibilidade, porém sacrifica a resolução.

Muitos insetos conseguem deslocar as células de pigmento, configurando o olho composto para trabalhar da melhor forma possível, tanto de dia como à noite (1).

A cerca de 100 milhões de anos atrás, a ordem Diptera (mosquitos e moscas) desenvolveu um truque para conseguir maior resolução de seus olhos compostos: eles separaram os rabdômeros foto-receptores em cada omatídeo, de forma que, ao invés de ver apenas um ponto por omatídeo, eles agora veriam sete (Figura 2d). Isso é chamado de olho

de superposição neural, o que também envolveu algumas ligações dentro do cérebro da mosca (Figura 2e)(2).

Outro aspecto no olho composto da mosca é a resposta dos foto-receptores em função da temperatura. Em 34°C, a velocidade de resposta dos foto-receptores de uma mosca *Calliphora vicina* ultrapassa o dobro da velocidade registrada aos 19°C de temperatura (3), conforme apresentado na Figura 3.

Figura 3: Tempo de resposta dos foto-receptores da mosca após incidência de flash em função da temperatura ambiente (3).

Um conceito importante relacionado ao tempo de resposta de qualquer sistema visual biológico é a Fusão de Cintilação (*Flicker Fusion*), também conhecido como CFFT (*Critical Flicker Fusion Threshold*). Ele determina (em Hz) o ponto limite no qual um animal passa a ver uma luz estroboscópica como se fosse um feixe contínuo. O valor da Fusão de Cintilação para moscas foi avaliado em 200 Hz (4), porém, Tatler et al determinaram que esse valor pode estar entre 400 e 500 Hz para temperaturas ambientes extremas de 34°C (3).

1.3 O sistema visual da mosca

As moscas povoam a Terra há mais de 300 milhões de anos, tendo desde então se ramificado em mais de 125.000 espécies. Isso faz com que as moscas sejam um dos grupos animais mais bem-sucedidos em termos evolutivos no nosso planeta (5).

Assim como $3/4$ das espécies animais na face da Terra, as moscas possuem um intrincado sistema de olhos compostos. Numa mosca existem cerca de 50.000 omatídeos distribuídos em ambos os olhos. Ao contrário do que ocorre no cérebro de vertebrados, um único neurônio pode ser localizado e identificado várias vezes, e em diferentes moscas. Isso permite a aquisição de mais dados sobre neurônios em localizações idênticas, porém em moscas distintas, proporcionando uma caracterização quantitativa mais confiável em termos estatísticos (6).

Cada omatídeo da mosca é dotado de oito células foto-receptoras, denominadas de R1 a R8. As células R1 a R6 de um mesmo omatídeo possuem diferentes alinhamentos ópticos. Isso permite que a luz que sai de um ponto espacial incida de forma que as células foto-receptoras R1 a R6 dos omatídeos vizinhos recebam a luz do mesmo ponto do espaço do qual os foto-receptores R7 e R8 do omatídeo central recebem (Figura 4). Os foto-receptores contêm organelas pigmentadas responsáveis pela captação da luz incidente e sua conversão em sinal elétrico por meio de reações químicas. Esse sinal elétrico é propagado pelos axônios de seu respectivo foto-receptor.

Conforme se avança para centros superiores de processamento visual, os mecanismos e funções exatas de suas estruturas tornam-se menos compreendidos, sendo alvos de intensa pesquisa. Há evidências por meio de observações da medula que sugerem que ela esteja ligada à computação e extração inicial de parâmetros visuais tais como velocidade, direção, orientação e contraste do estímulo fornecido (7).

O sistema visual da mosca possui dois neurônios H1, um em cada hemisfério cerebral,

Figura 4: Corte de seção do olho composto (2).

que conectam as duas placas lobulares diretamente. Com uma árvore dendrítica de $\approx 250 \mu m$ e um axônio medindo $\approx 1.200 \mu m$ de comprimento e $\approx 5 \mu m$ de diâmetro, as respostas do sistema visual ocorrem na forma de potenciais de ação detectáveis extracelularmente ao longo do axônio e da árvore dendrítica que se estendem pelo hemisfério oposto ao do neurônio H1 e seu soma. Portanto podemos dizer que esse tipo de aquisição é sensível a movimentos ipsilaterais regressivos. Isso quer dizer que obtemos resposta máxima do neurônio ao estimular os olhos compostos do lado oposto ao hemisfério onde se dá a aquisição do sinal, com movimentos horizontais de trás para frente em relação à cabeça da mosca, e obtemos uma resposta nula, com a inibição do sinal, para movimentos na direção contrária (Figura 5).

Os neurônios H1 apresentam um período refratário de $2 ms$. Isso quer dizer que após efetuar um disparo, o neurônio fica inativo durante $2 ms$, tempo esse necessário para sua repolarização (8).

Muitas pesquisas já foram feitas em torno do H1 e, hoje em dia, há um modelo desse tipo de neurônio consolidado em neurociência, não apenas pela já mencionada facilidade de identificação e pela resposta em *spikes* (disparos dos neurônios), mas também por sustentar respostas a estímulos por longos períodos de tempo e pelo fato da placa lobular ser reconhecidamente um centro de computação neural dos movimentos no entorno do

Figura 5: Sensitividade binocular dos neurônios H1 (lóbulo direito e esquerdo, respectivamente). Seta cheia representa a direção preferencial e seta vazia a direção de inibição (8).

animal, mantendo respostas estereotipadas e relativamente robustas (8). Apesar disso, é importante lembrar que, por mais estereotipada que sejam as respostas do H1, não se observa uma repetibilidade, *spike* por *spike*, entre duas respostas do mesmo H1 para dois estímulos idênticos. Ou seja, o H1 é apenas uma das várias partes do sistema complexo que ele constitui, sendo que cada resposta que se mede e analisa nele é resultado de muitos outros processamentos complexos realizados nas estruturas descritas anteriormente.

Uma das características mais notáveis do sistema visual da mosca, e típica desse tipo de sistemas, é seu alto grau de flexibilidade, sendo capaz de adaptar sua dinâmica conforme a estatística do estímulo fornecido, de forma a otimizar a transmissão de informações (9).

1.4 Uso de tubos de raios catódicos em neurociências

Tubos de raios catódicos (CRT)(Figura 6) têm se tornado uma condição *sine qua non* para a pesquisa experimental em visão. Porém, largura de banda, ajuste de alta voltagem, persistência do fósforo e taxas de atualização constituem limitações que afetam

a aparência das imagens apresentadas nos CRTs. O efeito conjunto de todos esses fatores pode resultar em uma imagem na tela significativamente diferente da imagem nominal que se deseja apresentar (10).

Figura 6: Estrutura interna do monitor CRT com deflexão eletromagnética: 1: Três canhões de elétrons (para os pontos de fósforo vermelho, verde e azul); 2: Feixes de elétrons; 3: Bobinas de foco; 4: Bobinas de deflexão; 5: Conexão do ânodo; 6: Máscara de separação dos feixes vermelho, verde e azul; 7: Camada de fósforo com regiões vermelhas, verdes e azuis; 8: Visão ampliada do lado interno da tela, coberto com camada de fósforo (11).

Experimentos com visão em neurociências frequentemente forçam os CRT's até o limite, ao necessitarem, por exemplo, de altas taxas de exibição de quadros por segundo. Um problema decorrente do uso de altas taxas de atualização é, no caso do tempo de duração do quadro ser menor que o tempo de decaimento do fósforo, a fusão da imagem apresentada com a luminância residual do quadro anterior (10). A fim de evitar esse problema, deve-se ter em mente que não é possível apresentar estímulos cujos períodos sejam menores do que o tempo de decaimento do pigmento de cor do monitor. A Tabela 1 apresenta os tempos de decaimento dos fósforos mais comuns nos monitores coloridos, em relação percentual ao valor atingido durante o pico de luminosidade.

Zele et al. (13) demonstraram que, para experimentos em neurofisiologia, tubos de raios catódicos podem produzir artefatos sobre o sinal que se deseja transmitir. Em suas conclusões, apontam que o uso de taxa de atualização de quadros com frequências maiores do que 100 Hz contribuem para a diminuição da geração de ruído. Testando

Tabela 1: Tempo de decaimento dos fósforos mais comuns (10)(12).

Fósforo	10%	1%	0.1%
P22r	1,5 <i>ms</i>	-	-
P22g	6,0 <i>ms</i>	-	-
P22b	4,8 <i>ms</i>	-	-
P4	0,060 <i>ms</i>	0,22 <i>ms</i>	1,25 <i>ms</i>
P104	0,060 <i>ms</i>	0,65 <i>ms</i>	3,50 <i>ms</i>
P45	1,5 <i>ms</i>	2,8 <i>ms</i>	4,25 <i>ms</i>

monitores RGB convencionais, eles também avaliaram o tempo de decaimento de uma tríade (conjunto formado pelos três pigmentos de fósforo colorido em um pixel), ativada de forma acromática (branca) e com luminância uniforme de 60 cd.m^{-2} , utilizando um espectroradiômetro, com diferentes aberturas. Os resultados são apresentados na (Figura 7).

Figura 7: Tempo de ativação do fósforo em uma tríade acromática do monitor RGB: linha contínua representa abertura de 2' de arco no detector e a linha pontilhada representa abertura de 1° (13).

1.5 Técnicas atuais de estimulação visual de insetos

1. Picasso

O gerador de estímulos Picasso, apresentado na Figura 8 não foi especificamente desenvolvido para esta finalidade. Trata-se na verdade de um sintetizador de imagens muito versátil, que serviu bem aos fins da comunidade científica.

Figura 8: Gerador de estímulos visuais Picasso (14).

A Figura 9 apresenta um padrão criado por Reinagel et al (15) utilizando o sintetizador Picasso, com uma taxa de 200 quadros por segundo.

Figura 9: Padrão gerado pelo Picasso (15).

Borst (16), Flanagan (14) e Warzecha (17) também utilizam o Picasso para geração de estímulos visuais para moscas, com taxas de repetição de quadros entre de 180 e

200 quadros por segundo, utilizando monitores XY (de deflexão eletrostática).

2. ViSaGe

O ViSaGe é um sistema que utiliza uma mistura de tecnologias de prateleira (como placas de vídeo com tecnologia *PCI-Express*) com hardwares dedicados, como controle de sinais de entrada e saída e conversores A/D e D/A. Este sistema, apresentado na Figura 10, opera apenas com monitores de tubo convencionais (18).

Figura 10: Gerador de estímulos visuais ViSaGe (19).

Ele apresenta resoluções da ordem de 512x480 pixels, com taxa de apresentação de quadros da ordem de 180 Hz. É muito utilizado em experimentos de psicologia com humanos ou em experimentos de monitoração de atividade no córtex de pequenos mamíferos (6). Possui um *Dot Clock* de 160 MHz e uma resolução de saída de 14 bits, com uma memória para sobreposição de quadros de 16 MB.

3. FliMax

FliMax (Figura 11) é um sistema de geração de estímulo visual da Universidade de Groningen, que possui um display com 7000 LEDs. Este sistema utiliza circuitos *sample-and-hold* para converter o sinal de uma placa VGA que controla o painel de LEDs. A vantagem do FliMax é que, ao contrário dos monitores XY, a apresentação de uma imagem constante não depende da taxa de *refresh*, e a taxa de atualização da imagem apresentada chega a 370Hz, sendo que a imagem gerada abrange um campo visual maior (20).

Figura 11: Gerador de estímulos visuais FliMax (21).

4. VSIImG

Desenvolvido por Almeida (22), o VSGImG foi uma parte de sua pesquisa de mestrado, em conjunto com outros módulos para obtenção e processamento dos sinais neurais captados do duto óptico da mosca.

O VSGImG é apresentado na Figura 12, no qual vemos a mosca em frente a um monitor XY (de deflexão eletrostática), sendo apresentado a ela um padrão de código de barras a uma taxa de 500 quadros por segundo.

Figura 12: Gerador de estímulos visuais VSIImG (22).

Este padrão, que possui resolução de 256x256 pixels e 256 níveis de intensidade, é deslocado horizontalmente (a fim de gerar os estímulos desejados) feito de forma analógica, através da aplicação de uma tensão de entrada junto ao canal de varredura horizontal, o que permite um deslocamento muito suave da imagem.

2 *O Experimento*

O experimento consiste na apresentação de um estímulo visual controlado com uma distribuição de velocidades horizontais conhecida, em um monitor CRT de deflexão eletromagnética. O neurônio H1 responderá a esse tipo de movimento com a geração de *spikes* (23), que serão captados extracelularmente por um micro-eletrodo. Após a amplificação, filtragem e discriminação adequada do sinal captado, os instantes t_i de cada *spike* serão armazenados no computador, para posterior análise, conforme apresentado no diagrama em bloco da Figura 13.

Figura 13: Diagrama em bloco do experimento completo.

2.1 Preparação da mosca

As diferenças anatômicas e estruturais entre o sistema visual de moscas machos e fêmeas são conhecidas (24), embora ainda não se saiba exatamente como isso afeta o processamento da informação visual e quão diferenciado ele resulta. Por isso, todos os experimentos foram realizados em moscas fêmeas, por possuírem um H1 de localização mais fácil.

Primeiramente, a mosca é preparada para uma microcirurgia através de sua imobilização com cera, posicionamento da mosca no suporte e posicionamento adequado da cabeça. Para manter a mosca viva por um longo período de tempo - que pode chegar a uma semana, em condições ambientais favoráveis - é inserido um tubo de alimentação no suporte que fica ao alcance da boca da mosca, fornecendo-lhe água com açúcar quando necessário (Figura 14).

Figura 14: Mosca imobilizada para o experimento (25).

Com um microscópio e um micro-bisturi, num exercício de habilidade motora são feitas incisões seguidas pela remoção do exoesqueleto protetor na parte posterior da cabeça da mosca. O interior da cabeça fica então exposto, e com um micro-gancho rompe-se um músculo transparente de difícil visualização, localizado ao longo do canto interno

do hemisfério exposto. Se isso não for feito adequadamente, a aquisição de dados é comprometida, pois esse músculo é uma grande fonte de ruídos devido a suas contrações periódicas. Ainda com o gancho, remove-se cuidadosamente o tecido protetor que recobre as camadas do cérebro. É posicionado então o fio de referência no canto inferior externo da cabeça da mosca (Figura 15).

Figura 15: Cabeça da mosca exposta. Detalhe no zoom: traquéias respiratórias e posição provável do sinal ótimo do H1 (invisível a olho nu) (25).

A próxima etapa consiste na localização do sinal do neurônio H1 com micro-eletrodo extracelular. Essa tarefa pode durar até horas e exige experiência e intuição por parte do experimentador, uma vez que sua localização é feita “às cegas” dentro de uma região de maior probabilidade de encontrá-lo, com a ajuda da saída de áudio amplificada do sinal elétrico captado (Figura 16). Uma vez localizado o sinal do neurônio, são realizados micro-ajustes do micro-eletrodo, a fim de amplificar o sinal e reduzir o ruído proveniente de outros neurônios e de traquéias respiratórias que se encontram na região (Figura 17).

Figura 16: Inserção do micro-eletrodo extracelular no cérebro da mosca. Esq: Mosca posicionada para localização do sinal. Dir: micro-eletrodo inserido com sinal detectado (25).

Figura 17: Tela do osciloscópio: sinal captado e discriminado em tempo-real (26).

2.2 Aquisição de dados neurais

Forneceremos a seguir uma visão geral da instrumentação utilizada no sistema de aquisição de dados neurais, que vem sendo desenvolvida e aperfeiçoada no DipteraLab (27) desde 1998. Toda a instrumentação de aquisição foi desenvolvida por nosso laboratório de instrumentação aplicada, exclusivamente para a realização desse tipo de experimento (22).

A aquisição do sinal pode ser dividida em cinco etapas principais, que apresentaremos na Figura 18 a seguir.

Figura 18: As cinco principais etapas na aquisição do sinal: 1: Captação; 2: Amplificação; 3: Filtragem; 4: Discriminação; 5: Registro dos tempos de ocorrência (22).

1. **Captação:** Como em nossa pesquisa estamos interessados apenas nos *spikes* disparados pelo neurônio H1 em resposta ao estímulo fornecido, foi feita a opção de aquisição extracelular do sinal, onde o micro-eletrodo é posicionado numa região externa porém próxima ao neurônio de interesse. Essa técnica, sendo menos invasiva, permite a aquisição de sinais neuronais mais próximos possíveis das condições naturais de funcionamento do H1, sem comprometer seriamente a dinâmica interna da célula e permitindo um nível de detalhe adequado ao que nos interessa.

Figura 19: Posicionamento ideal do micro-eletrodo extracelular em relação ao neurônio H1. Note que se trata de uma montagem, uma vez que em condições naturais o H1 não é visível a olho nu (25).

No meio extrafisiológico, a intensidade do sinal propagado em volta do axônio do neurônio H1 devido ao *spike* é da ordem de 20 - 100 μV . A captação desse sinal foi feita por meio de um micro-eletrodo de tungstênio marca FHC, próprio para esse tipo de experimento. Esse micro-eletrodo tem seu corpo revestido por verniz isolante de epoxi, sendo que apenas sua ponta fica exposta. Teremos uma maior ou menor seletividade do sinal pelo micro-eletrodo conforme a área exposta da ponta. Dado o grande número de neurônios que disparam em resposta aos estímulos visuais, é importante ter um alto grau de seletividade de sinal, a fim de podermos captar essencialmente o sinal de interesse. Assim, no nosso caso procuramos usar micro-eletrodos com áreas expostas menores, que garantem uma maior seletividade dos sinais porém com uma maior resistência elétrica em contrapartida - que com esses níveis de sinais neurais não contribuirá significativamente com ruídos térmicos. As impedâncias dos micro-eletrodos utilizados variaram entre 2,7 - 5 $M\Omega$.

2. **Amplificação:** A amplificação se dará em duas etapas:

- **Amplificador Diferencial:** Foi utilizado um amplificador de instrumentação que se encontra localizado imediatamente após o micro-eletrodo. Isso é feito para evitar a interferência de mais ruídos ao longo da transmissão de um sinal sem amplificação. As duas entradas desse amplificador serão dadas pelo micro-eletrodo, posicionado numa região de máxima proximidade ao neurônio H1, e pelo fio de referência, também posicionado dentro da cabeça da mosca de forma simétrica ao micro-eletrodo - em geral, no canto inferior da cabeça aberta (Figura 19). Essa configuração permitirá a eliminação de ruídos comuns a essas duas entradas. O amplificador também atua como um filtro passa alta, para frequências acima de 300 Hz, tendo conseqüentemente como saída um sinal AC. Essa saída terá um ganho de voltagem de 100 vezes.
- **Amplificador Inversor:** Nessa etapa, foi utilizado um amplificador operacional de baixo ruído, que amplificará o sinal também em 100 vezes e cuja saída será direcionada para o rack principal.

Ao fim da etapa de amplificação, o sinal terá sido amplificado cerca de 10.000 vezes, atingindo então a ordem de 1 volt, e já tendo sido pré-filtrado.

3. **Filtragem:** Para esse propósito, foram usados filtros ativos passa alta (frequência de corte = 300 Hz) e passa baixa (frequência de corte = 7 kHz), conformando um filtro passa-banda que refina mais a banda de frequências desejadas, eliminando componentes de altas e de baixas frequências.
4. **Discriminação:** O sinal antes de ser efetivamente discriminado passa por um monitor de áudio que permite um monitoramento sonoro do sinal, que será de grande valia na preparação do experimento, durante a localização do H1. A discriminação se dá por meio de um discriminador de janela de amplitude ajustável, com um limiar inferior (LI) e superior (LS) (Figura 18). Se o sinal estiver fora desse intervalo (tanto para mais como para menos), nada acontece. Se o sinal estiver dentro dessa janela de amplitude, isso elicia um pulso digital com largura fixa de $10 \mu s$. Temos assim a digitalização do sinal captado por discriminação, onde se atribue o valor 1 para cada pulso gerado pelo discriminador e um valor 0 quando nada é detectado.
5. **Registro dos tempos de ocorrência:** Nessa etapa, para cada *spike* ($= 1$) será atribuído seu respectivo instante temporal t_i . No sistema de aquisição utilizado em nossos experimentos, isso foi feito com o auxílio do 'Registro de tempo de ocorrência' (Figura 13), cujo relógio tem resolução de $1 \mu s$. O instante de cada disparo será registrado num arquivo de dados, no computador hospedeiro.

2.3 O estímulo

O estímulo visual é realizado movendo-se uma imagem fixa de maneira horizontal, com velocidade $v(t)$. Nós discretizamos o tempo em intervalos denominados *bins*, com 5 *ms*, valor próximo ao período refratário dos neurônios H1 (2 *ms*). A mosca então vê um novo quadro no monitor a cada $\delta t = 5$ *ms*, o qual muda sua posição δx de acordo com $\delta x(t) = v(t)\delta t$.

A velocidade $v(t)$ é gerada por um processo *Ornstein-Uhlenbeck* com tempos de correlação $\tau_c = 20$ *ms*. Esse processo é detalhado no Apêndice E. O estímulo é tomado a partir de uma distribuição gaussiana com função de correlação $C(t) = e^{-t/\tau_c}$. O experimento dura 20 minutos, consistindo de trechos repetidos com duração de 10 segundos. Em cada trecho, nos primeiros 5 segundos o mesmo estímulo é apresentado, sendo que nos próximos 5 segundos a mosca vê um estímulo diferente, assim como apresentado na Figura 20.

Figura 20: Estímulo: velocidades horizontais para o deslocamento de uma imagem fixa. Os primeiros 5 segundos constituem o estímulo a ser repetido, e os demais 5 segundos constituem um padrão aleatório, para evitar a adaptação do sistema visual da mosca (23).

3 *Metodologia*

Nesse capítulo são detalhados os materiais utilizados, e são apresentadas as diversas abordagens utilizadas para tratar o problema da geração de estímulos visuais para o estudo de neurociência com insetos. Durante todo o projeto, a linguagem de desenvolvimento utilizada foi o Verilog (28).

3.1 Kits de desenvolvimento utilizados

A maior parte da eletrônica utilizada foi baseada em kits de desenvolvimento de FPGA. Nas diferentes abordagens, foram utilizados chips dos dois principais fabricantes mundiais: as empresas Altera e Xilinx. Os softwares de desenvolvimento utilizados foram o Quartus 9.0 e o ISE Design Suite 10.1. A Figura 21 apresenta o kit DE2, com chip FPGA da família Cyclone II da Altera. Trata-se do kit mais popular no meio acadêmico, além de possuir custo acessível. Suas características mais relevantes para as finalidades do projeto são listadas a seguir.

- FPGA Altera Cyclone II EP2C35F672C6 com 33.216 elementos lógicos, 483.840 bits de memória, 4 PLLs e 475 pinos;
- Memória SRAM com 512 Kbytes (tempo de acesso de 10 ns);

Figura 21: Kit de desenvolvimento DE2 (29).

- Memória SDRAM com 8 Mbytes;
- DAC de vídeo com 10 bits por canal e com banda de 140 MHz (modelo ADV7123);
- Interfaces Ethernet e USB em hardware;
- Conectores de expansão IDC-40, mapeados como General Purpose I/O (GPIO).

Na linha das FPGAs mais avançadas atualmente, encontra-se a família Stratix III da Altera. A Figura 22 apresenta o kit DE3, do fabricante Terasic. Seu custo é consideravelmente elevado, e suas características principais são listadas a seguir.

- FPGA Altera Stratix III EP3SL150F1152C2 com 170.400 elementos lógicos, 5.630.976 bits de memória, 8 PLLs e 744 pinos;
- Conectores de expansão IDC-40, mapeados como General Purpose I/O (GPIO);
- Conectores HSTC de alta velocidade.

Além de possuir uma grande quantidade de elementos lógicos e bits de memória, o kit DE3 possui PLLs são mais avançadas, permitindo o ajuste de frequências intermediárias

com maior precisão e estabilidade. Os conectores HSTC apresentam uma grande quantidade de pinos, sendo que cada um dos quatro conectores encontrados no kit pode se ligar a um adaptador com três conectores IDC-40 na modalidade GPIO. Além das conexões para aterramentos e alimentações, temos a possibilidade de ligar cerca de 400 pinos de entrada e saída à esse kit, ao passo que, o modelo DE2 permite a conexão de pouco mais de 60 pinos.

Figura 22: Kit de desenvolvimento DE3 com adaptador HSTC para GPIO (29).

A Figura 23 apresenta o kit de desenvolvimento ML505 da Xilinx. Esse kit possui FPGA da família Virtex-5 e também faz parte da linha de modelos mais avançados da atualidade, além de possuir conexão para barramento PCI-Express.

- FPGA Xilinx Virtex-5 XC5VLX50T;
- Conector PCI Express Edge Connector (x1 Endpoint);
- Saída VGA/DVI;
- Gerador de *clock* de sistema programável.

O uso do barramento PCI-Express como duto para troca de dados entre o computador hospedeiro e a instrumentação desenvolvida representa o maior atrativo desse kit de desenvolvimento. Seu custo é intermediário dentre os custos dos kits descritos anteriormente.

Figura 23: Kit de desenvolvimento ML505 da Xilinx (30).

3.2 Primeira abordagem: controle do monitor XY

No início do desenvolvimento, estipulou-se a necessidade de averiguar como as FPGAs iriam se comportar frente à tarefa de geração de sinais de vídeo com altas taxas de amostragem. Características inerentes às FPGAs, como atrasos de *clock* (*clock skew*), *glitches* durante cruzamentos de domínios de clock, metaestabilidade, *jitter*, dentre outras, colocavam em dúvida se as mesmas seriam capazes de realizar essa tarefa de forma adequada e sem a introdução de artefatos visuais.

Uma possibilidade seria tentar gerar um padrão de imagem nos monitores XY veto-

riais atualmente utilizados em nosso laboratório, com taxa de 500 quadros por segundo. Foi projetado então um processador gráfico simplificado, e um padrão de barras foi armazenado em uma LUT implementada utilizando os bits de memória internos da FPGA.

Para esse experimento, foi utilizado o kit DE2. Como a saída do DAC de vídeo desse kit atua na faixa analógica do sinal de vídeo, entre 0 e 1 volt, circuitos eletrônicos foram construídos externamente ao kit com a finalidade de gerar as amplitudes da ordem de 0 a 10 volts, necessárias ao controle do monitor XY. A Figura 24 apresenta esse experimento em funcionamento.

Figura 24: Gerador de estímulos visuais operando em monitor XY.

3.3 Segunda abordagem: utilização do NIOS e uClinux

Uma vez comprovada a capacidade das FPGAs em gerar e manter os sinais de sincronismo da varredura do vídeo, mesmo com altas taxas de amostragem (como foi comprovado na seção anterior), a próxima etapa no desenvolvimento do projeto consistia em

estabelecer uma forma de enviar informações a partir de um computador hospedeiro para o sistema, de forma estável e robusta.

Dada a necessidade de enviar o padrão de imagem a ser exibido, doravante denominado simplesmente por *monet*, assim como as informações sobre o deslocamento a ser realizado na imagem, foi utilizada uma abordagem muito atual, na qual um processador baseado em síntese lógica (softcore) foi utilizado e, sobre o mesmo, foi executado o sistema operacional uClinux (31). A memória externa SDRAM (8 Mbytes) foi utilizada pelo sistema operacional embarcado, e a imagem do sistema de arquivos foi carregada diretamente, por meio da interface JTAG.

A arquitetura do processador gráfico, desenvolvida anteriormente, foi adaptada para uso em monitores CRT de deflexão eletromagnética (com geração de sincronismos horizontal e vertical), operando inicialmente em 150 Hz. Foram acrescentadas lógicas adicionais para realização do deslocamento da imagem, com base em registradores de entrada específicos para essa finalidade. Também foi adicionada lógica para geração de uma saída com a finalidade de indicar a mudança de quadros na geração do vídeo (sincronismo vertical).

Figura 25: Arquitetura do gerador de estímulos utilizando uClinux.

Nessa arquitetura, o *monet* é carregado diretamente por intermédio de uma lógica de controle, operando por meio da porta USB. O processador softcore NIOS II fica, então, encarregado de receber os estímulos pela interface Ethernet e repassá-los para o processador gráfico (utilizando um porto de entrada e saída em seu barramento, padrão Avalon),

cada vez que esse último gera uma interrupção indicando requisição de novo quadro.

O sistema operacional uClinux fica encarregado do gerenciamento da rede TCP/IP, em execução sobre a interface Ethernet, e também pelo gerenciamento das memórias FIFO, dentro do espaço do kernel, as quais mantêm as informações dos estímulos até que um pedido de interrupção seja realizado.

Os arquivos de estímulo são então acessados diretamente pela rede, utilizando mapeamento de disco por NFS. Um aplicativo rodando no espaço do usuário realiza as leituras dos estímulos, repassando os dados das mesmas a um módulo do kernel especialmente desenvolvido para essa aplicação, conforme ilustrado na Figura 26.

Figura 26: Relação entre os espaços do usuário e do kernel: acesso aos registradores

O código fonte completo do processador gráfico é apresentado no Apêndice B. Um estudo completo e detalhado sobre a utilização do sistema operacional uClinux sobre o processador NIOS II é detalhado pelo autor em (32).

3.4 Terceira abordagem: deslocamento da imagem baseado em FIFO

Na seção anterior foi explorada uma das formas de realizar a interface entre um sistema utilizando FPGA com um computador hospedeiro. Outra maneira de implementar essa interface se faz com a elaboração um tipo de circuito eletrônico customizado, também chamado de “glue logic”, no qual a FPGA se torna uma ponte entre o computador hospedeiro e os periféricos que se pretende controlar. Nessa abordagem, é utilizada a própria FPGA como ponte entre o computador hospedeiro e o processador gráfico.

A Figura 27 apresenta a arquitetura dessa nova abordagem. Assim como na abordagem anterior, o *monet* continua sendo carregado pela interface USB, em uma memória RAM de duas portas, cuja a segunda porta é controlada diretamente pelo processador gráfico, o qual realiza a requisição do valor do pixel correspondente ao instante da varredura.

O processador *softcore* foi substituído por uma memória FIFO de *clock* duplo, criada por meio da Megafunção DC-FIFO (33), um *IP Core* fornecido pela Altera. De um lado da FIFO, o computador hospedeiro, utilizando a porta paralela em modo padrão (SPP), envia os estímulos. O *clock* de escrita é determinado pelo computador, o qual também monitora um flag que indica quando a FIFO está preenchida pela metade.

Figura 27: Arquitetura do gerador de estímulos baseado em FIFO.

Do outro lado da FIFO, o *clock* de leitura é determinado pelo *dot-clock* do processador gráfico, o que evita problemas de *clock-skew*, dada a mudança de domínios de *clock*. Embora o *dot-clock* seja alto (84,164 MHz), os dados de estímulo que informam o deslocamento do vídeo são liberados apenas após o recebimento do sinal de mudança de quadro (sincronismo vertical) com taxa de 200 Hz, gerado a partir de uma saída registrada no processador gráfico.

Código 1 Saída registrada do sinal de sincronismo vertical.

```
output reg VERT_out;
reg VERT_in;
always@(posedge iCLK)
begin
    VERT_out<=VERT_in;
end
```

Como o sinal de sincronismo vertical (originalmente desenvolvido para atender as exigências da temporização do vídeo) era muito longo, a FIFO apresentava problemas em seu funcionamento, esporadicamente liberando mais de um dado e causando flickers no sistema gerador de estímulos.

Uma solução encontrada foi a utilização da função PULSO, a qual gera um pulso curto quando uma borda de subida é detectada. Seu código é descrito a seguir e sua arquitetura é apresentada na Figura 28.

Código 2 Função PULSO para gerar um pulso curto quando uma borda de subida é detectada.

```
module pulso(output reg pulse_out, input trigger_in, input clk);
reg delay;
always @ (posedge clk)
begin
    if (trigger_in && !delay) pulse_out <= 1'b1;
    else pulse_out <= 1'b0;
    delay <= trigger_in;
end
endmodule
```

Nessa abordagem foram utilizados kits DE2 e DE3, conforme mostrado pela Figura 29. Porém, toda a arquitetura foi implementada utilizando-se apenas o kit DE2. O kit DE3

tação de dois estímulos visuais independentes (34). A Figura 30 apresenta a arquitetura utilizada em uma abordagem inicial para tratar dessa necessidade.

Figura 30: Arquitetura do gerador de dois monitores: utilização híbrida dos kits DE2 e DE3.

Nessa arquitetura foram utilizados dois kits DE2 e um kit DE3. Dois processadores gráficos independentes foram implementados na FPGA Stratix, do kit DE3, compartilhando do mesmo *dot-clock*. O acesso dos processadores gráficos às memórias de vídeo e aos DACs foi realizado por meio da interface GPIO. Como o objetivo dessa abordagem era simplesmente averiguar o controle de dois monitores simultaneamente, não foram inseridas as FIFOs de estímulo na arquitetura. Na Figura 31, que mostra o sistema em funcionamento, é possível observar todos os kits conectados por cabos IDC de 40 vias (mesmo padrão dos cabos de *hard disks* IDE).

Figura 31: Gerador de estímulos de dois monitores em funcionamento.

3.6 Quinta abordagem: verificação de erros e controle do cursor

Nas abordagens anteriores, não houve preocupação com a verificação da validade dos dados do estímulo. Também não era possível realizar o envio de outros tipos de estímulo, além da informação de deslocamento. A fim de tratar dessas questões, foi desenvolvida uma arquitetura para um controlador de comunicação por meio do porto paralelo, permitindo validação dos dados e recebimento de outros tipos de estímulos (Figura 32).

Essa arquitetura foi desenvolvida para responder automaticamente às requisições do porto paralelo do computador hospedeiro, quando configurado em modo EPP (35). Essa arquitetura mantém um registrador especial para armazenar o valor do *checksum* (soma de todos os *bytes*) dos valores recebidos. Esse registrador é automaticamente recalculado na FPGA, sendo fornecido ao computador de controle, sempre que requerido. O computador, por sua vez, mantém seu próprio controle de *checksum*, a fim de verificar se os valores são

Figura 32: Arquitetura de comunicação e verificação de erros.

coincidentes.

Os códigos das funções em Matlab para comunicação com essa interface estão listados no Apêndice A, e o código fonte em Verilog do controlador EPP encontra-se listado no Apêndice B.

Uma vez que o controlador EPP realiza o tratamento dos dados de chegada, pode-se decidir o destino dos mesmos dentro do sistema. Nessa abordagem, optou-se por implementar um cursor junto à imagem do vídeo. A posição desse cursor é atualizada pelo computador de controle, o qual especifica que tipo de dado está sendo enviado, quer seja um deslocamento horizontal ou as coordenadas do cursor.

A apresentação de estímulos visuais mistos em diferentes camadas tem sido objeto de discussão recente nas pesquisas em neurociência da visão. A Figura 33 apresenta esse sistema em funcionamento.

Figura 33: Controle do cursor em funcionamento.

3.7 Sexta abordagem: ampliação da capacidade horizontal

Nas abordagens realizadas até agora, o único aspecto relacionado ao sistema visual da mosca levado em consideração havia sido a taxa de amostragem, da ordem de 200 quadros por segundo. Essa seção visa analisar a forma como a resolução espacial da imagem apresentada é percebida pelos foto-receptores nos omatídeos da mosca.

Inicialmente é necessário determinar o *dot-pitch* efetivo do monitor CRT utilizado. Trata-se de um modelo Studioworks 221U da marca LG, com diâmetro de tela de 21 polegadas, banda de 250 MHz, resolução máxima de 1600x1200 pixels e *dot-pitch* nominal de 0,266mm (36). Para atingir a taxa de 200 quadros por segundo, é necessário reduzirmos a resolução do monitor, a fim de não ultrapassar a banda de vídeo suportada. Nesse sistema utilizou-se então a resolução de 640x480 pixels.

Com essa resolução, em um monitor com tela de 21 polegadas, obtemos um *dot-pitch* efetivo de 0,666mm, de acordo com a equação 3.1, extraída de Compton (12).

$$DOT_PITCH_EFETIVO = \left(\frac{\left(\frac{horizontal}{\left(\frac{\sqrt{horizontal^2 + vertical^2}}{(diagonal * 25.4)} \right)} \right)}{horizontal} \right) \quad (3.1)$$

Se considerarmos a estrutura de distribuição dos omatídeos da mosca, veremos que existe um alinhamento de um grau entre cada omatídeo. Com a mosca posicionada a uma distância de $10cm$ ($R = 100mm$) da tela do monitor (Figura 34), e de acordo com a equação 3.2, obtemos $x = 1,745mm$.

Figura 34: Estrutura dos omatídeos no olho da mosca com alinhamento de um grau (1).

$$x = \frac{R \cdot \pi}{180} \quad (3.2)$$

Em uma análise inicial, o sistema é satisfatório, pois o valor mínimo do deslocamento da imagem ($0,666mm$) é menor do que o valor mínimo detectado pelos omatídeos à distância de 10 cm ($1,745mm$). Porém, levando em conta o que foi discutido na seção 1.2, o olho composto das moscas possui rabdomas abertos, o que contribui para o aumento efetivo da resolução espacial percebida pelo inseto.

Como não existem estudos conclusivos sobre o quão efetiva é a capacidade de percepção da resolução espacial das moscas, o desafio reside em se criar um sistema visual que apresente um deslocamento horizontal maior do que a própria resolução utilizada no monitor.

3.7.1 A técnica ‘Passos Artificiais Entre-Pixels’

A fim de atingir uma precisão de posicionamento maior do que a própria resolução da varredura horizontal do monitor, foi desenvolvida a técnica em questão. Seu princípio de funcionamento consiste no fato de que, dada uma imagem original, se amostrarmos a mesma tomando conjuntos de pixels vizinhos, e separando esses conjuntos em imagens intermediárias menores (Figura 35), a apresentação sequencial dessas imagens no tempo gera um efeito visual de deslocamento na direção do estímulo.

Figura 35: Geração das imagens com diferentes amostragens utilizando a técnica ‘Passos Artificiais Entre-Pixels’: a: imagem original; b: colunas pares e ímpares; c: imagem par; d: imagem ímpar.

O código a seguir apresenta como é realizada a geração das imagens amostradas para conjuntos de dois grupos de pixels vizinhos, utilizando o software Matlab. A Figura 36 tenta ilustrar o efeito causado, com base na ampliação exata de uma mesma região nas duas imagens geradas. Percebe-se nitidamente a impressão visual de deslocamento, embora o efeito seja mais simples de ser visualizado com a imagem em movimento.

Código 3 Script em Matlab para amostrar duas imagens a partir da original.

```

I = imread('imagem.bmp');           % Carrega imagem original (1280x960)
I=rgb2gray(I);                     % Converte para tons de cinza
A=I(1:2:end,1:2:end);               % Extrai pixels (primeira amostragem)
B=I(1:2:end,2:2:end);               % Extrai pixels (segunda amostragem)
imwrite(A,'amostragem1.bmp','BMP') % Armazena imagem A (640x480)
imwrite(B,'amostragem2.bmp','BMP') % Armazena imagem B (640x480)

```

Figura 36: Detalhe do funcionamento do efeito ‘entre-pixel’: as imagens ampliadas em destaque correspondem a ampliação exata da mesma região nas duas imagens amostradas.

3.7.2 A arquitetura para geração dos ‘Passos Artificiais Entre-Pixels’

A fim de gerar o efeito desejado, a arquitetura deve inicialmente comportar uma quantidade de frame-buffers de vídeo igual ao número de imagens amostradas. A Figura 37 apresenta a arquitetura desenvolvida para duas imagens.

Agora, os estímulos são mapeados utilizando-se 10 bits, sendo o bit menos significativo

Figura 37: A arquitetura para geração dos ‘Passos Artificiais Entre-Pixels’.

responsável pela seleção de qual frame-buffer estará ativo no momento da varredura do vídeo. Essa arquitetura passa a utilizar a interface EPP desenvolvida, para envio dos dados à FIFO de estímulo e também para carregamento dos *monets* aos frame-buffers.

Para ampliar ainda mais a resolução no efeito de posicionamento horizontal, é necessário tomar um conjunto maior de grupos de pixels durante a amostragem, ampliar a quantidade de frame-buffers de vídeo, e realizar a multiplexação utilizando a quantidade correspondente de bits menos significativos do estímulo.

3.8 Sétima abordagem: integração entre FPGA e GPU

A geração de estímulos visuais mais aprimorados, como cenas em 3D, não é efetivamente explorada experimentalmente devido à complexidade em modelar o sistema visual da mosca para esse tipo de estímulo. Porém, uma proposta de abordagem, pensando na necessidade futura desse tipo de experimento, é tratada aqui.

A idéia principal é fazer uso das ferramentas e códigos livres disponíveis, que são

abundantes na área, a adaptá-los às necessidades de nossos experimentos.

A primeira ferramenta necessária é um *engine* 3D. O mais indicado para nossa utilização é o Ogre3D (37), devido a seu tipo de licença, o qual permite o re-desenvolvimento e alteração dos últimos estágios do *pipeline* 3D em seu código fonte. A Figura 38 apresenta uma cena 3D naturalística renderizada pelo *engine* Ogre, com 6.500 triângulos, calculada a uma taxa de 33 quadros por segundo.

Figura 38: Cena 3D naturalística renderizada pela plataforma Ogre 3D.

Pretende-se então utilizar o *engine* Ogre para carregamento e definição dos elementos da cena, como objetos e luzes, e redirecionar o *pipeline* de renderização 3D para a arquitetura proposta, formada pela integração de FPGA e GPU. Basicamente, podemos definir os estágios desse *pipeline* da seguinte maneira:

- Cálculo de perspectiva;
- Geração de vértices;
- Enquadramento (Clipping);
- Eliminação por profundidade (Z-Buffer);
- Exibição (Saída de vídeo).

Os dois primeiros itens, cálculo de perspectiva e geração de vértices, demandam muito processamento em ponto flutuante, o que sabemos consumir muitos recursos quando implementados em FPGA. Já os demais itens podem ser implementados com relativa simplicidade e expectativa de ganho de performance.

Rollins (38) escreveu um algoritmo completo de traçado de raios para a plataforma CUDA(39) da NVidia, utilizando as vantagens do paralelismo desse sistema. A partir de seu trabalho, foi então desenvolvida a arquitetura apresentada na Figura 39.

Figura 39: Arquitetura da integração entre FPGA e GPU.

O cálculo de perspectiva e a geração de vértices foram mantidos dentro da arquitetura CUDA, devido à utilização de recursos em ponto flutuante. Já o cálculo de clipping (que determina quais objetos estão enquadrados na visão da câmera) e determinação de z-buffer (que descarta os objetos que estão obstruídos por outros no plano da câmera), assim como a saída final do frame buffer de vídeo, foram implementados na FPGA, utilizando o kit de desenvolvimento ML505 da Xilinx.

Para integrar as duas arquiteturas foi utilizado o barramento PCI-Express. Para lidar com os complicados níveis de sinais LVDS e as sincronizações críticas desse barramento foi utilizado um IP Core do ambiente de programação ISE, intitulado LogiCore ‘*Endpoint Block Plus Wrapper for PCI Express*’, o qual provê uma interface direta para a memória SDRAM no kit.

Os resultados desse sistema são apresentados na Figura 40, onde em destaque é possível visualizar o kit ML505 e a placa de vídeo NVidia 8600 no computador hospedeiro, e ao lado, uma cena em 3D gerada no monitor de apoio, ligado ao kit da FPGA.

Figura 40: Sistema integrado entre FPGA e GPU em funcionamento.

4 *Resultados e Discussão*

Várias das abordagens apresentadas no capítulo 3 tiveram seus resultados apresentados ao final de sua descrição, com a apresentação do equipamento em funcionamento.

Nesse capítulo, vamos nos ater aos resultados obtidos com experimentos utilizando o sistema visual da mosca propriamente dito, utilizando aquisições neurais do cérebro do animal. As duas abordagens que foram testadas em laboratório com o sistema visual da mosca foram:

- Deslocamento da imagem baseado em FIFO (seção 3.4);
- Ampliação da capacidade horizontal (seção 3.7.)

4.1 ‘Deslocamento da imagem baseado em FIFO’

Inicialmente pretende-se validar o sistema, comparando-o com o atual equipamento utilizado no laboratório, o VSImG, apresentado na seção 1.5. Uma vez que o sistema proposto opera em 200 Hz, e o VSImG opera em 500 Hz, foi necessário gerar um estímulo com mil pontos, sendo que a cada 5 pontos foi tomada uma amostra para gerar o estímulo

de 200 Hz, e a cada 2 pontos, foi tomada outra amostra, agora para formar o estímulo de 500 Hz, conforme apresentado na Figura 41. Esse procedimento garantiu que ambos os estímulos visuais se comportassem da mesma forma, com as mesmas velocidades e a mesma duração.

Figura 41: Geração do mesmo estímulo com duas amostragens: Losangos: amostragem de 200 Hz; Pontos: amostragem de 500 Hz.

Preparada a mosca, foi realizado o experimento utilizando o VSImG, com duração de 20 minutos. A mosca foi posicionada à uma distância de 10 cm da tela, com brilho ajustado para gerar fluxo luminoso de 5 lux. O deslocamento máximo da tela foi ajustado em 7 cm.

Após o término do experimento padrão, e utilizando-se a mesma mosca, um novo experimento foi realizado, com o gerador baseado em FIFO, conforme mostra a Figura 42.

As características principais do experimento foram mantidas, com a mosca posicionada a 10 cm do centro da tela, fluxo luminoso ajustado em 5 lux, e o deslocamento máximo da tela ajustado em 7 cm. A temperatura ambiente se manteve em 24°C em ambos os experimentos. O resultado pode ser visualizado na Figura 43, que apresenta o diagrama Raster do experimento. Um diagrama Raster é gerado a partir de uma dada lista de *spikes*

Figura 42: Experimento realizado com gerador baseado em FIFO.

gerada devido à execução repetitiva do mesmo ciclo de estímulos.

Outra forma de visualizar os resultados da aquisição neural é utilizando o chamado PSTH (*Peristimulus Time Histogram*), ou simplesmente gráfico das taxas de disparo, conforme apresentado na Figura 44. Nessa figura são apresentadas ambas as taxas de disparo registradas nos dois experimentos, sendo que as barras em azul correspondem ao VSI_{MG}, e as barras em vermelho, ao gerador baseado em FIFO. Dado que as barras são transparentes, é possível perceber que ambos os sistemas obtiveram taxas de disparo significativamente similares, validando então nosso sistema.

Figura 43: Raster do experimento realizado pelo gerador baseado em FIFO.

4.2 ‘Ampliação da capacidade horizontal’

O sistema de ampliação da capacidade horizontal foi utilizado em um experimento com o sistema visual da mosca, conforme apresentado na Figura 45. O experimento foi realizado duas vezes, com a mesma mosca, sendo que na primeira vez o efeito ‘entre-pixel’ foi desabilitado, para atuar como experimento de controle.

Para saber se o efeito ‘entre-pixel’ produziu os resultados desejados, deve-se fazer uso do histograma ISI (*Inter Spike Interval*), o qual apresenta os intervalos entre os disparos. Esperamos que, se a precisão do deslocamento horizontal for melhorada, o número de intervalos curtos entre os spikes deva diminuir.

Os histogramas ISI da Figura 46 mostram claramente uma diminuição do número de intervalos nos experimentos, sendo que na Figura 46-c podemos visualizar a ocorrência da diminuição dos intervalos curtos, entre 1 e 2 bins, validando então o sistema proposto.

Figura 44: Comparação entre os PSTH dos dois sistemas: Azul: VSImG; Vermelho: Gerador baseado em FIFO.

Figura 45: Sistema em funcionamento apresentando padrão naturalístico.

Figura 46: Histograma de intervalos: a: experimento padrão; b: experimento com efeito ‘entre-pixel’; c: diferença entre o padrão e o efeito ‘entre-pixel’, apresentando diminuição para os intervalos curtos, entre 1 e 2 bins.

5 *Conclusões*

Foram desenvolvidos diversos geradores de estímulos visuais para realização de experimentos em neurociência envolvendo insetos, especificamente moscas. A maioria dos geradores desenvolvidos utiliza monitores de vídeo convencionais, de prateleira.

Implementou-se geradores de estímulos visuais duplos, envolvendo dois monitores, tendo sido também acrescentado um estímulo de sobreposição baseado na geração de um cursor. Dois geradores desenvolvidos foram validados em laboratório, em experimentos envolvendo o sistema visual da mosca e a aquisição de dados neurais.

Para um dos geradores foi desenvolvida uma nova arquitetura, a fim de melhorar a capacidade de deslocamento horizontal do sistema. Embora originalmente projetada para estimular a visão de insetos da ordem Diptera, essa nova arquitetura pode ser utilizada em experimentos visuais de neurociência envolvendo quaisquer espécies de animais.

Referências

- 1 MOSES, K. Fly eyes get the whole picture. *Nature*, v. 443, n. 12, p. 638–639, 2006.
- 2 ZELHOF, A. et al. Transforming the architecture of compound eyes. *Nature*, v. 443, n. 12, p. 696–699, 2006.
- 3 TATLER, B.; O’CARROL, D.; LAUGHLIN, S. Temperature and the temporal resolving power of fly photoreceptors. *Journal of Comparative Physiology A*, v. 186, n. 4, p. 399–407, 2000.
- 4 MCCANN, G.; MACGINITIE, G. Optomotor response studies of insect vision. *Proceedings of the Royal Society B: biological sciences*, v. 163, n. 992, p. 369–401, 1965.
- 5 BORST, A.; HAAG, J. Neural networks in the cockpit of the fly. *Journal of Comparative Physiology A*, v. 188, n. 6, p. 419–437, 2002.
- 6 BURKHARDT, D. Autrum’s impact on compound eye research in insects. In: STAVENGA, D. G.; HARDIE, R. C. (Ed.). *Facets of vision*. Heidelberg: Springer-Verlag Berlin, 1989.
- 7 LAUGHLIN. The roles of the parallel channels in the early visual processing by the arthropod compound eye. In: ALI, M. (Ed.). *Photoreception and vision in invertebrates*. New York: Plenum Press, 1982.
- 8 HAUSEN, K. The lobula-complex of the fly: structure, function and significance in visual behaviour. In: ALI, M. (Ed.). *Photoreception and vision in invertebrates*. New York: Plenum Press, 1982. v. 74.
- 9 FAIRHALL, A. et al. Efficiency and ambiguity in an adaptive neural code. *Nature*, v. 412, n. 6849, p. 787–792, Aug. 2001.
- 10 GARCÍA-PÉREZ, M.; PELI, E. Luminance artifacts of cathode-ray tube displays for vision research. *Spatial Vision*, v. 14, n. 2, p. 201–215, 2001.
- 11 WIKIPEDIA’S definition for CRT Monitor. Disponível em: <http://en.wikipedia.org/wiki/Cathode_ray_tube>. Acesso em: Jul. 2009.

- 12 COMPTON, K. *Image performance in CRT displays*. [S.l.]: SPIE Publications, 2003.
- 13 ZELE, A.; VINGRYS, A. Cathode-ray-tube monitor artefacts in neurophysiology. *Journal of Neuroscience Methods*, v. 141, n. 1, p. 1–7, 2005.
- 14 FLANAGIN, V. *Dynamic adaptation in fly motion vision. 2006*. 113 p. (Masters Thesis) - Ludwig-Maximilians Universität, München, 2006.
- 15 REINAGEL, P. et al. Encoding of visual information by lgn bursts. *The American Physiological Society*, v. 81, n. 5, p. 2558–2569, 1999.
- 16 BORST, A. Noise, not stimulus entropy, determines neural information rate. *Journal of Computational Neuroscience*, v. 14, p. 23–31, 2003.
- 17 WARZECHA, A.; EGELHAAF, M. On the performance of biological movement detectors and ideal velocity sensors in the context of optomotor course stabilization. *Visual Neuroscience*, v. 15, n. 1, p. 113–122, 1998.
- 18 NERI, P. Spatial integration of optic flow signals in fly motion-sensitive neurons. *Journal of Neurophysiology*, v. 95, n. 12, p. 1608–1619, Dec 2006.
- 19 REN, J. *Cambridge research systems*. Disponível em: <http://www.crs-ltd.com/catalog/visage/overview.html>. Acesso em: 22 julho 2009.
- 20 ESTEVES, I. *Sistemas ópticos para geração de estímulos visuais para a mosca. 2007*. 42 p. Monografia (Projeto de Graduação) - Escola de Engenharia de São Carlos, Universidade de São Paulo. 2007.
- 21 LINDEMANN, J. et al. Flimax, a novel stimulus device for panoramic and highspeed presentation of behaviourally generated optic flow. *Vision Research*, v. 43, n. 7, p. 779–791, Mar. 2003.
- 22 ALMEIDA, L. O. B. *Desenvolvimento de Instrumentação eletrônica para estudos de codificações neurais no duto óptico em moscas. 2006*. 86 p. Dissertação (Mestrado) — Instituto de Física de São Carlos, Universidade de São Paulo, 2006.
- 23 STEVENINCK, R. et al. Reproducibility and variability in neural spike trains. *Science*, v. 275, n. 5307, p. 1805–1808, 1997.
- 24 FRANCESCHINI, N. Retinal mosaic of the fly compound eye. In: ALI, M. (Ed.). *Photoreception and vision in invertebrates*. New York: Plenum Press, 1982.

- 25 HORNER, N. *Multifractalidade no código neural da mosca. 2008.* 70 p. Dissertação (Mestrado) — Instituto de Física de São Carlos, Universidade de São Paulo, 2008.
- 26 GAZZIRO, M. A. *Instrumentação para neurobiofísica. 2002.* 21 p. Monografia (Projeto de Graduação) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. 2002.
- 27 KÖBERLE, R.; SLAETS, J. *Website do Laboratório DipteraLab do IFSC-USP.* Disponível em: <<http://www.dipteralab.ifsc.usp.br/novo>>. Acesso em: 15 de julho 2009.
- 28 IEEE-SA. *Standard Verilog Hardware Description Language.* [S.l.]: IEEE, 2001.
- 29 Terasic. *Terasic FPGA Systems.* Disponível em: <<http://www.terasic.com.tw>>. Acesso em: 23 julho 2009.
- 30 XILINX. *Xilinx INC.* Disponível em: <<http://www.xilinx.com>>. Acesso em: 23 julho 2009.
- 31 UCLINUX. *Embedded linux project.* Disponível em: <<http://www.uclinux.org/>>. Acesso em: 29 julho 2009.
- 32 GAZZIRO, M. A. *Tutorial de uClinux sobre NIOS II.* Disponível em: <<http://gpu-sc.googlecode.com/files/tutorialUclinux.pdf>>. Acesso em: 30 julho 2009.
- 33 ALTERA. *Single and Dual-Clock FIFO Megafunction: user guide.* 23 p. Disponível em: <http://www.altera.com/literature/ug/ug_fifo.pdf>. Acesso em: 28 julho 2009.
- 34 FERNANDES, N. et al. Recording from two neurons: second order stimulus reconstruction from spike trains. Neural computation, Cambridge, USA. In press. Disponível em: <<http://arxiv.org/abs/0906.3023>>. Acesso em: julho 2009.
- 35 PEACOCK, C. *Interfacing the parallel port.* Disponível em: <<http://www.beyondlogic.org/epp/epp.htm>>. Acesso em: 20 julho 2009.
- 36 LG. *Studioworks 221U Repairing manual.* 1999. 30 p. Technical manual.
- 37 STREETING, S.; JOHNSTONE, B. *OGRE - Open source 3D graphics engine.* Disponível em: <<http://www.ogre3d.org>>. Acesso em: 17 de julho 2009.
- 38 ROLLINS, E. *Real-time ray tracing with NVIDIA CUDA GPGPU.* Disponível em: <http://eric_rollins.home.mindspring.com/ray/cuda.html>. Acesso em: 17 de julho 2009.

- 39 NVIDIA. *CUDA Zone – The resource for CUDA developers*. Disponível em: <http://www.nvidia.com/object/cuda_home.htm>. Acesso em: 23 de julho 2009.
- 40 COVER, T. M.; THOMAS, J. A. *Elements of information theory*. 2nd. ed. Hoboken, N.J.: Wiley-Interscience, 2006.
- 41 FERNANDES, N. M. *Redução dimensional no espaço de estímulos no sistema visual da mosca*. 2009. 110 p. Tese (Doutorado) — Instituto de Física de São Carlos, Universidade de São Paulo, (Defesa prevista para out. 2009).

A Listagem dos códigos fonte em Matlab

Código 4 Função para ajuste do deslocamento da imagem (10 bits) em 2 ciclos.

```
function ajusta_desloc(ioObj, porta, dado, hi_low)
```

```
    io32(ioObj, porta+2, 1);
    io32(ioObj, porta, dado);

    if (hi_low>0) io32(ioObj, porta+2, 9);
        else io32(ioObj, porta+2, 3);
    end

    io32(ioObj, porta+2, 1);
```

Código 5 Função para ajuste das coordenadas do cursor (8 bits).

```
function ajusta_cursor(ioObj, porta, dado, x_ou_y)
```

```
    io32(ioObj, porta+2, 5);
    io32(ioObj, porta, dado);

    if (x_ou_y>0) io32(ioObj, porta+2, 13);
        else io32(ioObj, porta+2, 7);
    end

    io32(ioObj, porta+2, 5);
```

Código 6 Função para ler sinal de ‘meio-vazio’ enviado pela FIFO.

```
function [dado_int]=FIFO_meio_vazio(ioObj, porta)
```

```
    dado=io32(ioObj, porta+1);
    dado_int=(bitand(uint8(dado), uint8(64))/64);
```

Código 7 Software para envio do estímulo.

```

clear all
% Inicializa a porta paralela
ioObj = io32();
status = io32(ioObj);
porta = hex2dec('378');
io32(ioObj, address, 0);
% Carrega arquivos de estimulo (deslocamento e coordenadas do cursor)
importfile('estimulo.dat');
[TAM B]=size(estimulo);
importfile('cursor_x.dat');
importfile('cursor_y.dat');
chksum=0;
old_chksum=0;
n=10; % largura do valor de deslocamento, em bits
for C = 1:TAM % LACO PRINCIPAL
    % Aguarda FIFO de estímulos estar pela metade
    while (FIFO_meio_vazio(ioObj, porta)),
        end
    % Envia byte mais significativo do deslocamento
    if ((bitget(estimulo(C), 8)==0) && (bitget(estimulo(C), 9)==0))
        ajusta_desloc(ioObj, porta, 0, 1);
    elseif ((bitget(estimulo(C), 8)==1) && (bitget(estimulo(C), 9)==0))
        ajusta_desloc(ioObj, porta, 1, 1);
    elseif ((bitget(estimulo(C), 8)==0) && (bitget(estimulo(C), 9)==1))
        ajusta_desloc(ioObj, porta, 2, 1);
    elseif ((bitget(estimulo(C), 8)==1) && (bitget(estimulo(C), 9)==1))
        ajusta_desloc(ioObj, porta, 3, 1);
    end;
    % Envia byte menos significativo do deslocamento
    ajusta_desloc(ioObj, porta, estimulo(C), 0);
    % Ajusta posição do cursor
    ajusta_cursor(ioObj, porta, cursor_x(C), 1);
    ajusta_cursor(ioObj, porta, cursor_y(C), 0);
    % Calcula Checksum no MATLAB
    dado=estimulo(C);
    old_chksum=chksum;
    chksum=dado+old_chksum;
    if ((dado+old_chksum)>=(2^n)) chksum=chksum-(2^n); end
    % Recupera Checksum calculado pela FPGA
    low=le_chksum(ioObj, porta, 0)
    high=le_chksum(ioObj, porta, 1)
    chk_fpga=(high*256)+low;
    % Se diferente, aborta a execucao do experimento
    if (chk_fpga~=chksum) disp('Erro de Checksum!!'); break; end
end

```

Código 8 Função para ler checksum calculado pela FPGA (10 bits) em 2 ciclos.

```
function [dado]=le_chksum(ioObj, porta, hi_low)
```

```
    io32(ioObj, porta+2, 36);
```

```
    if (hi_low>0) io32(ioObj, porta+2, 44);
```

```
        else io32(ioObj, porta+2, 38);
```

```
    end
```

```
    dado=io32(ioObj, porta);
```

```
    io32(ioObj, porta+2, 36);
```

```
    io32(ioObj, porta+2, 0);
```

B Listagem dos principais códigos fonte em Verilog

Parâmetros do Processador Gráfico

```
// Arquivo PARAMET.H

// Para DOT CLOCK de 84,164 MHz
// Resolução ajustada em 640x480 pixels
// Frequencia horizontal ajustada em 104,7 kHz
// Frequencia vertical ajustada 200 Hz

// Horizontal (Pixel)
parameter H_SYNC_CYC = 96;
parameter H_SYNC_BACK = 48;
parameter H_SYNC_ACT = 640;
parameter H_SYNC_FRONT= 16;
parameter H_SYNC_TOTAL= 800;

// Vertical (Linha)
parameter V_SYNC_CYC = 2;
parameter V_SYNC_BACK = 32;
parameter V_SYNC_ACT = 480;
parameter V_SYNC_FRONT= 11;
parameter V_SYNC_TOTAL= 525;

// Offset de inicio
parameter X_START = H_SYNC_CYC+H_SYNC_BACK+4;
parameter Y_START = V_SYNC_CYC+V_SYNC_BACK;
```

Processador Gráfico

```
module Processador_Grafico(
    iCursor_RGB_EN,    // habilita cores do cursor
    iCursor_X,         // coordenada horizontal do cursor
    iCursor_Y,         // coordenada vertical do cursor
    iCursor_R,         // valor da cor vermelha do cursor
    iCursor_G,         // valor da cor verde do cursor
    iCursor_B,         // valor da cor azul do cursor
    iRed,
    iGreen,
    iBlue,
    oAddress,
    oCoord_X,
    oCoord_Y,
    oVGA_R,
    oVGA_G,
    oVGA_B,
    oVGA_H_SYNC,
    oVGA_V_SYNC,
    oVGA_SYNC,
    oVGA_BLANK,
    oVGA_CLOCK,
    SHIFT_out,
    VERT_out,
    iCLK,
```

```

    iRST_N
);

`include "PARAMET.H"

// Host Side
output reg [19:0] oAddress;
output reg [9:0] oCoord_X;
output reg [9:0] oCoord_Y;
input [3:0] iCursor_RGB_EN;
input [9:0] iCursor_X;
input [9:0] iCursor_Y;
input [9:0] iCursor_R;
input [9:0] iCursor_G;
input [9:0] iCursor_B;
input [9:0] iRed;
input [9:0] iGreen;
input [9:0] iBlue;
output [9:0] oVGA_R;
output [9:0] oVGA_G;
output [9:0] oVGA_B;
output reg oVGA_H_SYNC;
output reg oVGA_V_SYNC;
output oVGA_SYNC;
output oVGA_BLANK;
output oVGA_CLOCK;
input [8:0] SHIFT_out;
output reg VERT_out;
input iCLK;
input iRST_N;

reg [9:0] H_Cont;
reg [9:0] V_Cont;
reg [9:0] Cur_Color_R;
reg [9:0] Cur_Color_G;
reg [9:0] Cur_Color_B;
wire mCursor_EN;
wire mRed_EN;
wire mGreen_EN;
wire mBlue_EN;

assign oVGA_BLANK = oVGA_H_SYNC & oVGA_V_SYNC;
assign oVGA_SYNC = 1'b0;
assign oVGA_CLOCK = iCLK;
assign mCursor_EN = iCursor_RGB_EN[3];
assign mRed_EN = iCursor_RGB_EN[2];
assign mGreen_EN = iCursor_RGB_EN[1];
assign mBlue_EN = iCursor_RGB_EN[0];

reg VERT_in;

always@(posedge iCLK)
begin
    VERT_out<=VERT_in;
end

// Plota pixel de acordo com a verredura e o conteudo da memoria
always@(posedge iCLK or negedge iRST_N)
begin
    if(!iRST_N)
    begin
        oCoord_X <= 0;
        oCoord_Y <= 0;
        oAddress <= 0;
    end
    else
    begin
        if( H_Cont>=X_START && H_Cont<X_START+H_SYNC_ACT &&
            V_Cont>=Y_START && V_Cont<Y_START+V_SYNC_ACT )
        begin
            oCoord_X <= H_Cont-X_START+SHIFT_out;
            oCoord_Y <= V_Cont-Y_START;
            oAddress <= oCoord_Y*H_SYNC_ACT+oCoord_X-3;
        end
    end
end

```



```

                (V_Cont==Y_START + iCursor_Y+10) ||
                (V_Cont==Y_START + iCursor_Y-10) ||
                (V_Cont==Y_START + iCursor_Y+11) ||
                (V_Cont==Y_START + iCursor_Y-11) ||
                (V_Cont==Y_START + iCursor_Y+12) ||
                (V_Cont==Y_START + iCursor_Y-12) ||
                (V_Cont==Y_START + iCursor_Y+13) ||
                (V_Cont==Y_START + iCursor_Y-13) ||
                (V_Cont==Y_START + iCursor_Y+14) ||
                (V_Cont==Y_START + iCursor_Y-14) ||
                (V_Cont==Y_START + iCursor_Y+15) ||
                (V_Cont==Y_START + iCursor_Y-15)
            )
        )
        && mCursor_EN
    )
    begin
        Cur_Color_R <= iCursor_R;
        Cur_Color_G <= iCursor_G;
        Cur_Color_B <= iCursor_B;
    end
    else
    begin
        Cur_Color_R <= iRed;
        Cur_Color_G <= iGreen;
        Cur_Color_B <= iBlue;
    end
end
end
end

// Gerador de sincronismo horizontal
always@(posedge iCLK or negedge iRST_N)
begin
    if(!iRST_N)
    begin
        H_Cont      <= 0;
        oVGA_H_SYNC <= 0;
    end
    else
    begin
        // Contador do sincronismo horizontal
        if( H_Cont < H_SYNC_TOTAL )
            H_Cont <= H_Cont+1;
        else
            H_Cont <= 0;
        // Gera o sinal de sincronismo horizontal
        if( H_Cont < H_SYNC_CYC )
            oVGA_H_SYNC <= 0;
        else
            oVGA_H_SYNC <= 1;
    end
end

// Gerador de sincronismo vertical
always@(posedge iCLK or negedge iRST_N)
begin
    if(!iRST_N)
    begin
        V_Cont      <= 0;
        oVGA_V_SYNC <= 0;
        VERT_in <= 0;
    end
    else
    begin
        // Se ocorrer sincronismo horizontal, reinicia
        if(H_Cont==0)

```



```

begin
    // Contador do sincronismo vertical
    if( V_Cont < V_SYNC_TOTAL )
    begin
        V_Cont <= V_Cont+1;
        VERT_in <= 0;
    end
    else
    begin
        V_Cont <= 0;
        VERT_in <= 1;
    end
    // Gera o sinal de sincronismo vertical
    if( V_Cont < V_SYNC_CYC )
        oVGA_V_SYNC <= 0;
    else
        oVGA_V_SYNC <= 1;
    end
end
end

// Registradores internos que mantem o valor dos canais de cor

reg [9:0] R_R;
reg [9:0] G_G;
reg [9:0] B_B;

assign oVGA_R = R_R;
assign oVGA_G = G_G;
assign oVGA_B = B_B;

always@(posedge iCLK)
begin
    R_R = (H_Cont>=X_START+9 && H_Cont<X_START+H_SYNC_ACT+9 &&
        V_Cont>=Y_START && V_Cont<Y_START+V_SYNC_ACT )
        ? (mRed_EN ? Cur_Color_R : 0) : 0;
    G_G = (H_Cont>=X_START+9 && H_Cont<X_START+H_SYNC_ACT+9 &&
        V_Cont>=Y_START && V_Cont<Y_START+V_SYNC_ACT )
        ? (mGreen_EN ? Cur_Color_G : 0) : 0;
    B_B = (H_Cont>=X_START+9 && H_Cont<X_START+H_SYNC_ACT+9 &&
        V_Cont>=Y_START && V_Cont<Y_START+V_SYNC_ACT )
        ? (mBlue_EN ? Cur_Color_B : 0) : 0;
end

endmodule

```

Interface EPP com controle de CHECKSUM

```

// Atribui pinagem da FPGA e cria registradores de comunicacao
wire EPP_WRITE = ~GPIO_1[7];
wire [7:0] EPP_DATAIN={GPIO_1[21],GPIO_1[19],GPIO_1[17],GPIO_1[15],GPIO_1[14],GPIO_1[13],GPIO_1[11],GPIO_1[9]};
wire [7:0] EPP_DATAOUT;
assign {GPIO_1[21],GPIO_1[19],GPIO_1[17],GPIO_1[15],GPIO_1[14],GPIO_1[13],GPIO_1[11],GPIO_1[9]}=EPP_DATAOUT;
wire EPP_INTERRUPT;
assign GPIO_1[23] = EPP_INTERRUPT;
wire EPP_WAIT;
assign GPIO_1[31] = EPP_WAIT;
wire EPP_DATA_STROBE = ~GPIO_1[25];
wire EPP_RESET = GPIO_1[27];
wire EPP_ADDR_STROBE = ~GPIO_1[29];

//Devemos sincronizar os sinais de strobe com nosso clock
// Utilizacao de registrador de deslocamento de 3 pontos para sincronizar clock
wire EPP_STROBE = EPP_DATA_STROBE | EPP_ADDR_STROBE; // apenas um eh ativo por vez
reg [2:0] EPP_STROBE_REG;
always @(posedge OSC_50) EPP_STROBE_REG <= {EPP_STROBE_REG[1:0], EPP_STROBE};

// Detecta as bordas de strobe
wire EPP_STROBE_EDGE1 = (EPP_STROBE_REG[2:1]==2'b01);
wire EPP_STROBE_EDGE2 = (EPP_STROBE_REG[2:1]==2'b10);

// Responde imediatamente a uma transacao
assign EPP_WAIT = EPP_STROBE_REG[1];

// Recebe escrita da porta EPP e calcula CHECKSUM
reg [7:0] addr_reg, data_reg;
reg [7:0] cursor_y, cursor_x;
reg [9:0] chksum=0, old_chksum=0;

always @(posedge OSC_50)
begin
    if(EPP_STROBE_EDGE1 & EPP_WRITE)
    begin
        if (EPP_ADDR_STROBE & ~EPP_RESET)
        begin
            addr_reg = EPP_DATAIN;
            old_chksum = chksum;
            chksum = ((addr_reg<<8)+ old_chksum);
        end
        else
        if(EPP_DATA_STROBE & ~EPP_RESET)
        begin
            data_reg = EPP_DATAIN;
            old_chksum = chksum;
            chksum = (data_reg + old_chksum);
        end
        // Recebe posição do Cursor pela sinalizacao do bit EPP_RESET
        else if(EPP_ADDR_STROBE & EPP_RESET) cursor_y <= EPP_DATAIN;
        else if(EPP_DATA_STROBE & EPP_RESET) cursor_x <= EPP_DATAIN;
    end
end

// Atende a um comando de leitura da porta EPP para informar o CHECKSUM
wire EPP_READ = ~EPP_WRITE;
wire [7:0] EPP_DATA_MUX = EPP_ADDR_STROBE ? chksum[9:8] : chksum[7:0];
assign EPP_DATAOUT = (EPP_READ & EPP_WAIT) ? EPP_DATA_MUX : 8'hZZ;

```

C Cabo de ligação entre FPGA e porta paralela

Figura 47: Cabo do gerador de estímulos visuais.

Tabela 2: Atribuição dos pinos da porta paralela no modo EPP (35).

Pino	Sinal	Direção	Função	Invertido
1	EPP_WRITE	Saída	Nível baixo indica escrita e alto leitura	Sim
2-9	EPP_D0-D7	Bi-dir	Barramento de dados 8 bits	Não
10	EPP_INTERRUPT	Entrada	Interrupção na borda de subida	Não
11	EPP_WAIT	Entrada	Usado para handshaking	Sim
12	RESERVADO	Entrada	Reservado	Não
13	RESERVADO	Entrada	Reservado	Não
14	EPP_DATA_STROBE	Saída	Indica transferência de DADO	Sim
15	RESERVADO	Entrada	Reservado	Não
16	EPP_RESET	Saída	Reset, ativo em nível baixo	Não
17	EPP_ADDR_STROBE	Saída	Indica transferência de ENDEREÇO	Sim
18-25	GROUND	GND	Aterramento	—

D Diagramas RTL

Figura 48: RTL do gerador de estímulos baseado em FIFO.

Figura 49: RTL da geração do clock de vídeo no kit DE3

Figura 51: RTL do gerador de dois monitores: Parte do kit DE2

E Processo de Ornstein-Uhlenbeck

A posição ocupada pela imagem a cada $5ms$ é determinada por um arquivo de posição gerado no computador. Este conjunto de posições é obtido ao integrarmos uma série de velocidades gerada por um processo estatístico denominado processo de *Ornstein-Uhlenbeck* (40)(41). Esta série temporal de velocidades V_t é obtida da seguinte maneira,

$$V_{t+\Delta} = c + \alpha \times V_t + \Phi_t, |\alpha| < 1. \quad (E.1)$$

Onde os valores Φ_t são obtidos de uma distribuição Gaussiana de média zero e desvio σ_Φ^2 . O termo α representa quanto o próximo valor $V_{t+\Delta}$ está correlacionado com o valor anterior V_t . Veremos como a escolha de α determina a escala temporal de correlação entre os valores de V_t . O termo c representa uma velocidade constante.

Para determinarmos o valor médio desta série temporal, efetuamos a média sobre *ensembles* da eq. E.1,

$$\langle V_{t+\Delta} \rangle = c + \alpha \times \langle V_t \rangle + \langle \Phi_t \rangle. \quad (E.2)$$

Como temos $\langle \Phi_t \rangle = 0$,

$$\langle V \rangle = c + \alpha \times \langle V \rangle \quad (E.3)$$

$$\langle V \rangle = \frac{c}{1 - \alpha}. \quad (E.4)$$

Em geral teremos, $c = 0$, ou seja, $\langle V \rangle = 0$.

Para obtermos o desvio da distribuição V_t de média zero, precisamos obter a média de seus valores ao quadrado.

$$V_{t+\Delta}^2 = [\alpha \times V_t + \Phi_t]^2 \quad (E.5)$$

$$\langle V^2 \rangle = \alpha^2 \langle V^2 \rangle + \langle \Phi^2 \rangle + 2\alpha \langle V \times \Phi \rangle. \quad (E.6)$$

O produto de duas distribuições independentes simétricas de média zero resulta em uma distribuição simétrica de média zero. Ou seja,

$$\langle V \times \Phi \rangle = 0. \quad (E.7)$$

Reescrevendo, $\langle \Phi^2 \rangle \equiv \sigma_\Phi^2$ temos,

$$\sigma_V^2 = \langle V^2 \rangle - \langle V \rangle^2 = \frac{\sigma_\Phi^2}{1 - \alpha^2}. \quad (\text{E.8})$$

Vemos que para $\alpha \rightarrow 0$, reobtemos a distribuição Gaussiana inicial, $V \rightarrow \Phi$.

Podemos calcular a influência da correlação introduzida por α na determinação dos próximos valores desta série temporal. Escreveremos $V_{t+N} \equiv V_N$, ou seja, $V_t \equiv V_0$. Desta forma, iterativamente,

$$V_1 = \alpha V_0 + \Phi_0 \quad (\text{E.9})$$

$$V_2 = \alpha V_1 + \Phi_1 = \alpha(\alpha V_0 + \Phi_0) + \Phi_1 = \alpha^2 V_0 + \alpha \Phi_0 + \Phi_1 \quad (\text{E.10})$$

$$V_3 = \alpha V_2 + \Phi_2 = \alpha(\alpha^2 V_0 + \alpha \Phi_0 + \Phi_1) + \Phi_2 = \alpha^3 V_0 + \alpha^2 \Phi_0 + \alpha \Phi_1 + \Phi_2. \quad (\text{E.11})$$

Consequentemente,

$$V_N = \alpha^N V_0 + \sum_{i=0}^{N-1} \alpha^i \Phi_{N-1-i}. \quad (\text{E.12})$$

Ao multiplicarmos a expressão acima por V_0 e efetuarmos novamente a média sobre *ensembles*, obteremos a correlação entre a velocidade no tempo t e $t + \Delta$.

$$\langle V_N V_0 \rangle = \alpha^N \langle V_0^2 \rangle + \langle V_0 \sum_{i=0}^{N-1} \alpha^i \Phi_{N-1-i} \rangle. \quad (\text{E.13})$$

Mais uma vez utilizando o fato da multiplicação de duas distribuições independentes de média zero ter média zero, $\sum_{i=0}^{N-1} \alpha^i \langle V_0 \Phi_{N-1-i} \rangle = 0$ e utilizando eq. E.8,

$$\langle V_N V_0 \rangle = \alpha^N \sigma_V^2 = \alpha^N \frac{\sigma_\Phi^2}{1 - \alpha^2} \quad (\text{E.14})$$

Podemos reescrever esta expressão a fim de obtermos a relação entre α e o tempo de correlação do estímulo τ ,

$$\langle V_{t+N} V_t \rangle = \frac{\sigma_\Phi^2}{1 - \alpha^2} \times e^{-N/\tau} \quad (\text{E.15})$$

$$\tau = -\frac{1}{\ln |\alpha|} \quad (\text{E.16})$$

Ou seja, a partir do processo de *Ornstein-Uhlenbeck* (eq. E.1, $c = 0$) geramos uma série temporal Gaussiana de média zero (eq. E.4), desvio σ_V (eq. E.8) e correlação temporal τ (E.16).

Lista de abreviaturas e siglas

CFFT	Critical Flicker Fusion Threshold,	p. 20
IDC	Insulated Displacement Connector,	p. 38
GPIO	General Purpose Input/Output,	p. 38
HSTC	High-Speed Terasic Connector,	p. 38
PCI	Peripheral Component Interconnect,	p. 39
VGA	Video Graphics Array,	p. 39
DVI	Digital Visual Interface,	p. 39
LUT	Look Up Table,	p. 40
JTAG	Joint Test Action Group,	p. 41
FIFO	First In First Out,	p. 42
NFS	Network File System,	p. 42
CRT	Cathode Ray Tube,	p. 23
IP	Intellectual Property,	p. 43
SPP	Standard Parallel Port,	p. 43
RTL	Register Transfer Level,	p. 45
IDE	Integrated Drive Electronics,	p. 46
EPP	Enhanced Parallel Port,	p. 47
GPU	Graphics Processing Unit,	p. 52
CUDA	Compute Unified Device Architecture,	p. 54
LVDS	Low-Voltage Differential Signaling,	p. 54
PSTH	Peristimulus Time Histogram,	p. 58
ISI	Inter Spike Interval,	p. 60
RGB	Red Green Blue,	p. 24
FPGA	Field Programmable Gate Array,	p. 37
PLL	Phase-Locked Loop,	p. 37
SRAM	Static Random Access Memory,	p. 37
SDRAM	Synchronous Dynamic Random Access Memory,	p. 37
DAC	Digital-to-analog converter,	p. 37
USB	Universal Serial Bus,	p. 37