



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 nº 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

Projeto h3dge

Relatório de andamento

Autores: <i>Jefferson Chaves Ferreira</i> <i>João Paulo Condé Oliveira prado</i>	Data de emissão: 13/09/2011
--	--------------------------------



ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Avenida Professor Luciano Gualberto, travessa 3 n° 158 CEP 05508-900 São Paulo SP
Telefone: (11) 3091-5583 Fax (11) 3091-5294

Departamento de Engenharia de Computação e Sistemas Digitais

SUMÁRIO

1. Evolução do projeto	3
1.1. Correção de Bugs da implementação em SystemC	3
1.2. Implementação em Verilog.....	3
1.3. Refinamento da arquitetura do hardware	4
1.3.1. Processador	5
1.3.2. Coprocessador.....	6
1.4. Software embarcado	7
1.5. Ferramentas de software.....	7
2. Cronograma	8
Bibliografia	8



PROJETO h3dge

Relatório de andamento

1. EVOLUÇÃO DO PROJETO

1.1. CORREÇÃO DE BUGS DA IMPLEMENTAÇÃO EM SYSTEMC

O código em SystemC foi estudado; contudo, apesar de serem encontradas algumas inconsistências no processo de construção da árvore, os principais problemas da implementação não foram resolvidos. A descrição em SystemC ainda continua tendo o problema do buraco no meio da figura e o fato de alguns triângulos não serem exibidos. A Figura 1 demonstra os problemas.



FIGURA 1. BUGS NA IMPLEMENTAÇÃO SYSTEMC DO RAYTRACER.

1.2. IMPLEMENTAÇÃO EM VERILOG

Para evitar um grande atraso no projeto, a implementação em hardware foi iniciada. A linguagem de descrição de hardware escolhida é o Verilog e a plataforma de desenvolvimento que será utilizada é a Xilinx Design Suite 13.1 a qual acompanha a FPGA do projeto. Cabe ressaltar que esta plataforma não suporta System Verilog (uma extensão de Verilog que facilita a concepção de projetos mais complexos), logo o processo de implementação será mais trabalhoso que o esperado. O módulo que testa a interseção de um raio com um triângulo já



foi começado e espera-se concluí-lo até o fim desta semana juntamente com sua bancada de teste.

Este módulo recebe um raio de luz e um triângulo, inicia os cálculos para a verificação da intersecção e retorna um vetor no qual os dois primeiros bits (da esquerda pra direita) indicam se houve intersecção (00 – não houve, 01 – houve) e os 32 restantes indicam o valor t que deve ser somado à coordenada inicial da reta de equação $X = \text{start} + t * \text{direction}$ para atingir o triângulo. As principais dificuldades no processo de implementação estão ligadas às diferenças de abstração entre o desenvolvimento em hardware e o desenvolvimento em software.

1.3. REFINAMENTO DA ARQUITETURA DO HARDWARE

Conforme definido em (Chaves Ferreira & Condé Oliveira Prado, 2011), a arquitetura geral a ser adotada no projeto do hardware é a exposta na Figura 2.

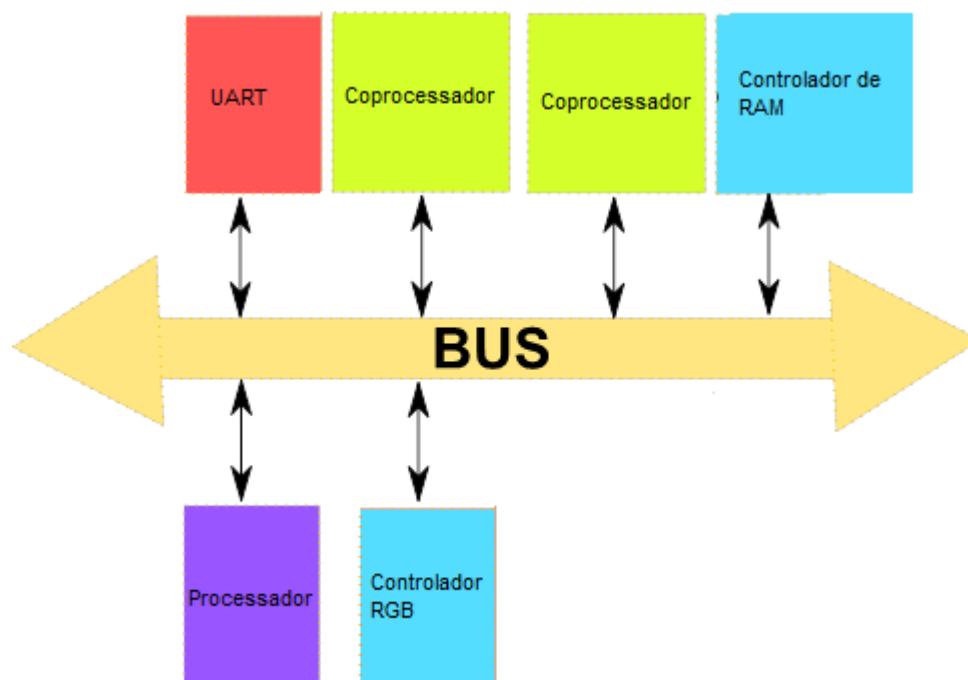


FIGURA 2. ARQUITETURA GERAL DO SOC *H3DGE*

Durante esta primeira do projeto, estudamos os recursos oferecidos pela placa de desenvolvimento ML507, da Xilinx, para a implementação dessa arquitetura. Essa placa é



Departamento de Engenharia de Computação e Sistemas Digitais

construída ao redor da FPGA Virtex-5 XC5VFX70T e possui uma série de periféricos, dentre os quais destacamos (Xilinx, Inc, 2011):

- Memória DDR2 SODIMM de 256 MB;
- Memória flash de 32 MB;
- SRAM de 9 MB com bus de dados de 32 bits e 4 bits de paridade;
- Saída de vídeo DVI, e adaptador para VGA incluso;
- Conector DB9 para porta serial RS232;
- Interface JTAG;
- LEDs, botões e switches para debug.

A Xilinx também fornece as ferramentas de software para utilização da placa, notadamente o Embedded Development Kit, parte da suíte de desenvolvimento ISE, que contém diversos IPs para a utilização dos periféricos disponíveis.

1.3.1. PROCESSADOR

Inicialmente, foi proposta a utilização do processador PicoBlaze para comandar as atividades do SoC. O PicoBlaze é um *soft core* altamente otimizado para as FPGAs da Xilinx, que permite sintetizar em um pequeno número de slices uma unidade de processamento funcional e flexível. Suas principais características são (Chu, 2008):

- Palavra de dados de 8 bits;
- 16 registradores de 8 bits; (Git - Fast Version Control System)
- Memória de dados de 64 bytes;
- Instruções de 18 bits de largura;
- Endereços de instrução de 10 bits de largura, suportando um programa de até 1024 instruções;
- 256 portas de entrada e 256 portas de saída;
- 2 ciclos de clock por instrução;
- 5 ciclos de clock para tratamento de uma interrupção.

O PicoBlaze é uma boa alternativa para executar programas que fazem essencialmente E/S. Entretanto, após estudo aprofundado de suas características, é possível perceber que ele não é apropriado às necessidades de processamento do *h3dgc*, pelas seguintes razões:

1. O algoritmo de criação da k -d tree será executado em software, visto que essa etapa do processo não pode ser paralelizada. Visto que o PicoBlaze só pode ser programado em linguagem assembly, a escrita do algoritmo seria dificultada;



2. Mesmo se a decisão de implementar o algoritmo em assembly fosse tomada, a realização seria trabalhosa, visto o tamanho dos registradores e da ULA – uma operação de adição de dois números de 32 bits necessitaria da metade dos registradores;
3. Mesmo se o algoritmo fosse implementado com sucesso, ele dificilmente caberia na memória de programa de 1024 instruções;
4. O PicoBlaze não tem um controlador de memória externa embutido. Assim, seria necessário adicioná-lo como uma IP, e escrever o driver para utilizá-lo – driver este que, por sua vez, ocuparia espaço na já escassa memória de programa.

Tendo em vista essas limitações, optou-se pelo uso do core PowerPC 440 embutido na FPGA. Tal processador possui funcionalidades muito mais favoráveis às necessidades deste projeto, tais como (Xilinx, Inc, 2010):

- Arquitetura RISC 32 bits;
- 32 registradores de 32 bits;
- Caches de instrução e de dados de 32 KB cada;
- Bus PLB (Processor Local Bus) para conexão de processadores adicionais;
- Interface JTAG;
- Suporte à programação em C via gcc.

Além das funcionalidades superiores, um outro argumento a favor da utilização desse core é seu “custo zero”, visto que o bloco está presente em silício (sua utilização não necessita de slices adicionais).

1.3.2. COPROCESSADOR

A arquitetura do coprocessador também foi refinada, tendo em vista a implementação da interface deste com o bus PLB. Foi adotado um modelo de registradores de controle mapeados em memória, a saber:

Registrador	Função
CTRL	Registrador de controle. Inclui um bit START para iniciar a atividade do módulo, e flags que indicam o estado da execução
TREE_ROOT	Endereço da raiz da árvore k -d
IMG_WIDTH	Largura da imagem



IMG_HEIGHT	Altura da imagem
IMG_X	Coordenada x do canto superior esquerdo da imagem
IMG_Y	Coordenada y do canto superior esquerdo da imagem
IMG_ADDR	Endereço de memória a partir do qual deverá ser gravada a imagem de saída
LSRC_NB	Número de fontes de luz
LSRC_ADDR	Endereço do vetor de fontes de luz
MAT_NB	Número de materiais da cena
MAT_ADDR	Endereço do vetor de materiais

TABELA 1. REGISTRADORES DO COPROCESSADOR DE RAY TRACING

1.4. SOFTWARE EMBARCADO

Com a definição do processador a ser utilizado, foi possível iniciar também as considerações sobre o software embarcado. Visto que o processador a ser utilizado possui recursos suficientes, será utilizado um sistema operacional embarcado em tempo real – o FreeRTOS. Esse sistema possui um *port* para a arquitetura PowerPC 440 e exemplos de aplicações na placa ML507, o que deve facilitar o desenvolvimento; no mais, ele ainda possui as seguintes qualidades:

- *Overhead* mínimo de RAM, ROM e processamento. Tipicamente o kernel ocupa entre 4 e 9 Kbytes na memória;
- Simplicidade: o núcleo do SO está contido em somente 3 arquivos C;
- Licença livre para utilização em projetos comerciais ou não;
- Fácil utilização, amplo suporte de uma comunidade bastante ativa;
- Documentação ampla e detalhada. (Chu, 2008)

1.5. FERRAMENTAS DE SOFTWARE

Como ferramenta de controle de versão, foi escolhido o sistema Git (Xilinx, Inc, 2010), que utiliza versionamento distribuído. Git funciona em Linux, Mac OS X e Windows – as três plataformas sendo utilizadas simultaneamente para o desenvolvimento do projeto –, e é reconhecido pela sua rapidez, eficiência e flexibilidade.



Em ambiente Windows, utiliza-se a IDE Eclipse para o desenvolvimento do código em SystemC, e a suíte ISE para o desenvolvimento em Verilog. Em Mac OS X e Linux utilizam-se Makefiles e editores de texto para SystemC, e a versão Linux do ISE.

2. CRONOGRAMA

	Setembro			Outubro				Novembro				
	1	2	3	4	5	6	7	8	9	10	11	12
Testes com a FPGA												
Implementação estruturas matemáticas												
Implementação árvore kd												
Implementação árvore de raios de luz												
implementação interfaces entrada saída												
Testes de integração												
Monografia												

BIBLIOGRAFIA

Chaves Ferreira, J., & Condé Oliveira Prado, J. (2011). *h3dge: um gerador de imagens 3D em hardware*.

Chu, P. P. (2008). *FPGA Prototyping by VHDL Examples*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Git - Fast Version Control System. (s.d.). Acesso em 12 de Setembro de 2011, disponível em <http://git-scm.com/>

Xilinx, Inc. (24 de Fevereiro de 2010). Embedded Processor Block in Virtex-5 FPGAs.

Xilinx, Inc. (16 de Maio de 2011). ML505/ML506/ML507 Evaluation Platform User Guide.