



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

h3dge

A **h**ardware **3D** image **g**enerator

Relatório de andamento II

Autores: <i>Jefferson Chaves Ferreira</i> <i>João Paulo Condé Oliveira Prado</i>	Data de emissão: 01/11/2011
--	--------------------------------



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. Evolução da arquitetura de hardware	3
1.2. Implementação em SystemVerilog.....	4
1.3. Particionamento software/hardware	4
1.4. Software embarcado	5
2. Cronograma	5
Bibliografia	6



PROJETO h3dge

Relatório de andamento II

1. EVOLUÇÃO DO PROJETO

1.1. EVOLUÇÃO DA ARQUITETURA DE HARDWARE

Conforme definido em (Chaves Ferreira & Condé Oliveira Prado, Projeto h3dge — Relatório de andamento I, 2011), buscou-se a implementação na FPGA Virtex-5 utilizada neste projeto de um System-on-Chip baseado no processador embarcado PowerPC 440.

Para programar a FPGA, utiliza-se um cartão Compact Flash e o módulo SystemACE presente na placa ML-507, que permite a configuração do hardware por meio de um arquivo .ace. Tal arquivo, por sua vez, é gerado por um script TCL a partir de:

- Um arquivo .bit contendo o bitmap do circuito a ser gravado na FPGA, gerado pelo programa Xilinx Platform Studio;
- Um ou vários arquivos .elf contendo programas a serem carregados na memória, gerados pelo Xilinx Software Development Kit;
- E um ou vários arquivos .dat contendo dados binários a serem carregados na memória.

Os primeiros testes foram realizados a partir dos arquivos de exemplo fornecidos juntamente com o kit de desenvolvimento. Esses testes se mostraram conclusivos, visto que foi possível verificar o funcionamento de praticamente todas as funções que a placa oferece: LEDs, botões, chaves, buzzer, porta serial, interface Ethernet, saída DVI. Como validação final, foi testada uma distribuição BlueCat Linux executando o servidor web Apache, que mostrou-se completamente funcional.

Após essa primeira etapa, procedeu-se à criação do sistema definitivo, contendo somente os componentes necessários à realização do projeto, a saber:

- Processador PPC440;
- Controlador de memória DDR2;
- Controlador de porta serial;



- Controlador de vídeo (saída DVI);
- Portas de entrada/saída de propósito geral (GPIOs);
- Bus interno PLB v4.6.

Tal sistema-base foi testado a partir de programas em C simples, objetivando verificar a correta integração entre os módulos. Até o presente momento, somente foi possível validar o correto funcionamento da portas de entrada/saída e da porta serial; foram encontradas severas dificuldades na integração do controlador de vídeo Xilinx TFT ao sistema utilizando a versão 13.1 da suíte Xilinx ISE. Atualmente, estuda-se o emprego de uma versão anterior da ferramenta (12.1) para proceder à síntese do circuito, para evitar problemas que possivelmente teriam origem na versão mais recente.

1.2. IMPLEMENTAÇÃO EM SYSTEMVERILOG

É importante ressaltar que houve uma grande dificuldade para conseguir uma combinação de ferramentas que possibilitasse uma implementação coerente em Verilog e SystemVerilog. Após várias tentativas, conseguiu-se a combinação do ActiveHDL (para a implementação e simulação), Synplify (para a síntese inicial) e ISE (para o mapeamento e roteamento) do código sintetizado. Após a resolução destas dificuldades, prosseguiu-se com o processo de especificação dos diversos elementos que envolvem a dinâmica do algoritmo de *ray tracing* como os triângulos, os vetores e os raios de luz.

Embora todos estes elementos já encontrem-se implementados, tal implementação não foi executada da maneira planejada. É importante ressaltar que houve problemas de limitações causados pela versão disponível do SystemVerilog para a implementação.

O primeiro módulo (teste de intersecção de um raio com um triângulo) foi implementado e testado, contudo ainda faltam alguns elementos relacionados com a sua estrutura como o operador divisão que encontra-se em teste. Após a realização do primeiro módulo, espera-se que os outros módulos sejam implementados com maior velocidade.

1.3. PARTICIONAMENTO SOFTWARE/HARDWARE

O ambiente de desenvolvimento composto por diversas ferramentas somado com o processo não trivial de uso do kit de desenvolvimento da ML507 levou a proposta de um particionamento da implementação do raytracer em duas partes: uma composta por hardware e outra composta por software. Tal particionamento será executado da seguinte forma:

- Entrada e construção da árvore de indexação espacial – **software**;



- Intersecção, reflexões e refrações – **hardware**;
- Cálculo da intensidade luminosa – **hardware**;
- Saída dos dados – **hardware** (utilizando recursos disponíveis na plataforma embarcada)

Para cada imagem gerada, é necessária a construção de uma árvore de indexação espacial antes da execução do algoritmo de *ray tracing*. Dessa forma não haverá sobrecarga de nenhum barramento presente o que viabiliza tal divisão. A divisão permitirá uma implementação mais facilitada sem prejudicar as vantagens de se possuir um hardware dedicado para a execução do algoritmo de *ray tracing*.

1.4. SOFTWARE EMBARCADO

Uma vez o sistema-base definido, foi possível iniciar a implementação do software embarcado. Tal implementação iniciou-se com a adaptação do *port* do sistema operacional FreeRTOS para a versão 13.1 da ferramenta Xilinx Software Development Kit, que modificava em alguns pontos a API utilizada. Após modificações nas camadas de abstração do hardware, o sistema pôde ser executado com sucesso na FPGA.

Em seguida, tendo em vista o novo particionamento software/hardware, foi começada a modificação do software em SystemC para a execução no sistema embarcado. Visto que não foi possível compilar toda a biblioteca SystemC para a plataforma PPC440, atualmente o trabalho em software está focalizado na confecção de uma biblioteca C++ que implemente os tipos de dados necessários ao correto funcionamento do software existente (notadamente os números em ponto flutuante).

2. CRONOGRAMA

	novembro				
	1	2	3	4	5
Integração controlador de vídeo					
Software embarcado em C++					
Implementação estruturas matemáticas					
Implementação árvore kd					
Implementação árvore de raios de luz					
Integração dos módulos de hardware					
Testes de integração					
Monografia					



BIBLIOGRAFIA

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

Chaves Ferreira, J., & Condé Oliveira Prado, J. (2011). *Projeto h3dge — Relatório de andamento I*. Escola Politécnica da Universidade de São Paulo, São Paulo.

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

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.