Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

Programa de Licenciatura en Ingeniería en Computadores

Curso: CE-4302 Arquitectura de Computadores II

Especificación Proyecto I: Diseño e Implementación de una Arquitectura Vectorial/SIMD para la generación de gráficos 2-D por Computador

Profesor:

Ronald García Fernández

Fecha de entrega: 11 de octubre, 2023

Semestre: II 2023

Objetivo general

Mediante el desarrollo de este proyecto, el estudiante aplicará los conceptos de arquitectura de computadores II en el diseño e implementación de una Arquitectura Vectorial para la generación de gráficos 2-D por computador. Atributos relacionados: Herramientas de Ingeniería (HI3), Diseño (DI2).

Motivación

Los gráficos por computador son un área de la informática visual, donde se emplean computadoras tanto para generar imágenes visuales sintéticamente como integrar o cambiar la información visual y espacial del mundo "real".

A pesar de que existen procesadores especializados para realizar su generación (GPU) para efectos prácticos de este proyecto se pretende diseñar e implementar una arquitectura vectorial o SIMD que permita la generación de gráficos 2-D por computador con el fin de crear una imagen a partir de estructuras geométricas simples (círculos, líneas, elipses, triángulos)

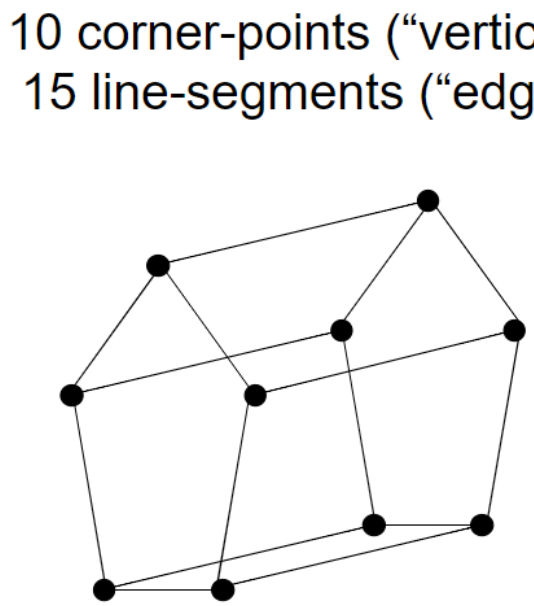

Figura 1. Figura geométrica generada mediante un computador.

Características generales del sistema que implementa la Arquitectura

1- Las imágenes salida deben ser almacenadas en memoria, y desplegadas de alguna forma.
2- El tamaño de la imagen no debe ser muy grande (recomendación [640x480])
3- El formato de las imágenes debe ser RGB con pixeles con valores entre [0, 255]
4- El sistema debe permitir al usuario cargar el código de asm con la implementación de las figuras a dibujar.
5- El sistema debe permitir la interacción del usuario observar el flujo del programa ensamblador.

El método de generación de las figuras es completamente libre (usando CORDIC, Bresenham, etc) pero debe ser forma "programática" **vectorial** (sugerencia http://math.hws.edu/graphicsbook/c2/s2.html).

Requisitos de Arquitectura (ISA)

1- Debe diseñar un conjunto de instrucciones y arquitectura vectorial/SIMD que permita solucionar el problema planteado, considerando detalles como:
   a- Modos de direccionamiento.
   b- Tamaño y tipo de datos.
   c- Tipo y sintaxis de las instrucciones.
   d- Registros disponibles y sus nombres
   e- Codificación y descripción funcional de las instrucciones

   Tome en cuenta que estos detalles deben ser justificados desde el punto de vista de diseño (complejidad, costo, área, recursos disponibles).

2- Las instrucciones a desarrollar son libres así como el tipo de datos, **sin embargo se deben crear instrucciones especializadas vectoriales/SIMD para la generación de los gráficos**.

Los productos finales de esta etapa son el *instruction reference sheet* y un programa ('compilador') que permita traducir las instrucciones del formato establecido a lenguaje de máquina, con la finalidad de cargarlo en memoria.

Puede reutilizar la implementaciones "escalares" de otros cursos, o inclusive referirse a extensiones de *ISAs* existentes sin embargo **es obligatorio que se diseñe instrucciones vectoriales para generar las imágenes.**

Requisitos de Micro-arquitectura

1- La implementación diseñada debe ser correcta respecto a las reglas definidas por la arquitectura, esto quiere decir que el procesador debe ser capaz de ejecutar todas las instrucciones definidas y su especificación respecto a errores y excepciones.

2- El diseño e implementación debe emplear ***pipelining***. Tenga en cuenta las implicaciones respecto a riesgos de dicha técnica, el uso de registros y unidades de ejecución.

3- Debe ser implementado usando *SystemVerilog**, NO se permite la entrega empleando *Verilog* únicamente.

4- **No es permitido el uso de 3rd Party IPs que hagan de VPU** (Vector Processing Unit), sin embargo puede reutilizar bloques funcionales como multiplicadores, sumadores como bloques básicos.

5- El procesador debe tener capacidad de segmentación de memoria en datos e instrucciones además debe ser capaz de acceder los dispositivos de entrada y salida del sistema (display, volcado de memoria, *switches*, etc).

6- Cada unidad funcional del sistema debe ser debidamente probada en simulación, para verificar su funcionamiento correcto (*unit tests*) y además debe incluir pruebas de integración y sistema, si no entrega simulaciones no será evaluado el proyecto de forma funcional.

Los resultados finales de esta etapa son:

a- El código fuente (*SystemVerilog**) y el *bitstream** para programar la tarjeta de desarrollo.

b- Un diagrama de bloques de micro-arquitectura y descripción de las interacciones entre ellos.

c- Simulaciones de las pruebas unitarias y de integración.

d- Reporte de consumo de recursos del FPGA*,**

Requisitos de software

1- Crear un modelo de referencia que permita validar la correctitud de la implementación de HW.
2- Crear una aplicación (software) empleando la arquitectura diseñada, con el fin de crear una imagen basada en figuras geométricas simples (se premiará al grupo con la imagen más "compleja" y original mediante votación)

Notas

El proceso de diseño debe incluir propuestas y comparación de viabilidad de las mismas.

Evaluación

El proyecto se desarrollara de forma grupal con máximo de 5 integrantes, note que solo hay 8 tarjetas disponibles

La evaluación del proyecto se da bajos los siguientes rubros:

- Presentación Funcional 60% (incluye las simulaciones, presentación corta)
- Artículo científico/comercial con la descripción del proceso de diseño, simulaciones y resultados 20%
- Documentación del proceso de diseño, pruebas* y *reference sheet* 20%

Para que este proyecto sea evaluado es necesario cumplir con la entrega de los siguientes avances:

- Prototipo en lenguaje de "alto" nivel para generar gráficos usando vectorización/*loop unrolling* y diseño inicial del set de instrucciones, indicando que instrucciones específicas serán desarrolladas y en cual ISA está basado, y por qué serán implementadas **22 de octubre 12mn 2023**.
- Diagrama de bloques de la micro-arquitectura, justificando fuentes y consideraciones **01 de noviembre 2023**.
- Simulación de las instrucciones que se encuentren funcionalmente correctas junto el consumo recursos del FPGA*,** **12 noviembre 2023.**
- **Entrega final del código fuente del compilador, algoritmo en asm, código en *SystemVerilog* y artículo 17 de noviembre 2023 (revisión 18 de noviembre 2023)**

El formato de entrega seria comunicado el 16 de Octubre 2023.

Este proyecto tiene la oportunidad de obtener hasta un máximo de 40% por criterio de puntaje extra, en función de optimización en el uso de recursos del FPGA**, creatividad en el diseño.

Por ejemplo:

1- Diseño e implementación de hardware específico para la mejora de los gráficos generados (antialiasing, dithering, entre otros) 15%
2- Plan de pruebas unitarias e integración y cobertura de funcionalidades 10%
3- Comparación de desempeño de ejecución vectorial sobre no-vectorial 15%

*Este proyecto puede ser modelado en SystemC si es así la demostración puede ser basada únicamente simulaciones, ya sea modelando usando *TLM* o generando *SystemC* sintetizable, si desea explorar esta idea puede consultar los enlaces:

https://learnsystemc.com/

https://github.com/intel/systemc-compiler

**El FPGA disponible es el DE2i-180 Se puede pedir al profesor Luis Alberto Chavarría, de haber otro disponible considere la compatibilidad de versiones de herramientas.

Referencias

[1] http://math.hws.edu/graphicsbook/ visitada 07-10-23

[2] https://www.geeksforgeeks.org/bresenhams-circle-drawing-algorithm/ visitada 05-10-23

[3] https://hackernoon.com/computer-graphics-scan-line-polygon-fill-algorithm-3cb47283df6 visitada 11-10-23

[4] https://github.com/accellera-official/systemc visitada 11-10-23