

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

Programa de Licenciatura en Ingeniería en Computadores

Curso: CE-4301 Arquitectura de Computadores I



Especificación Proyecto I

Profesor:

Ronald García Fernández

Fecha de entrega: 06 de Abril, 2018

Semestre: I, 2018

Objetivo general

Mediante el desarrollo de este proyecto, el estudiante aplicará los conceptos de arquitectura de computadores en el diseño e implementación en hardware de una arquitectura del set de instrucciones (ISA), específico y propio para una aplicación de procesamiento digital de imágenes. Atributos relacionados: Herramientas de Ingeniería (HI), Diseño (DI).

Motivación

El procesamiento y análisis digital de imágenes consiste en el uso de sistemas computarizados para la manipulación de imágenes digitales, ya sea para mejorar su calidad (procesamiento) o para la extracción de información (análisis).

Una de las técnicas más comunes en el procesamiento de imágenes es el filtrado, el cual puede ser logrado mediante el uso de la convolución entre la imagen (I) y un elemento de tamaño variable denominado máscara o kernel (H), mediante el cual se obtiene una nueva imagen I' como se describe a continuación.

$$I' = I * H$$

$$I'(y, x) = \sum_{(j, i) \in R_H} I(y - j, x - i) H(j, i)$$

A continuación se muestra un ejemplo de convolución 2D

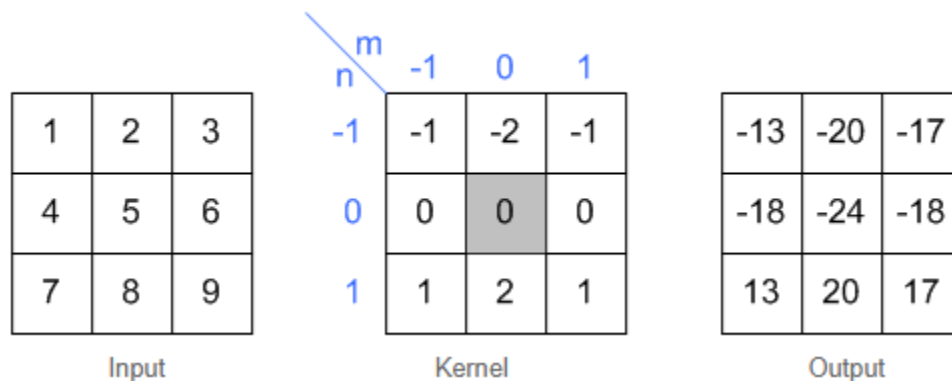


Figura 2. Ejemplo de convolución 2D.

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

$$\begin{aligned}
 y[0,0] &= x[-1,-1] \cdot h[1,1] + x[0,-1] \cdot h[0,1] + x[1,-1] \cdot h[-1,1] \\
 &\quad + x[-1,0] \cdot h[1,0] + x[0,0] \cdot h[0,0] + x[1,0] \cdot h[-1,0] \\
 &\quad + x[-1,1] \cdot h[1,-1] + x[0,1] \cdot h[0,-1] + x[1,1] \cdot h[-1,-1] \\
 &= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 + 0 \cdot 0 + 1 \cdot 0 + 2 \cdot 0 + 0 \cdot (-1) + 4 \cdot (-2) + 5 \cdot (-1) = -13
 \end{aligned}$$

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

$$\begin{aligned}
 y[1,0] &= x[0,-1] \cdot h[1,1] + x[1,-1] \cdot h[0,1] + x[2,-1] \cdot h[-1,1] \\
 &\quad + x[0,0] \cdot h[1,0] + x[1,0] \cdot h[0,0] + x[2,0] \cdot h[-1,0] \\
 &\quad + x[0,1] \cdot h[1,-1] + x[1,1] \cdot h[0,-1] + x[2,1] \cdot h[-1,-1] \\
 &= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 + 1 \cdot 0 + 2 \cdot 0 + 3 \cdot 0 + 4 \cdot (-1) + 5 \cdot (-2) + 6 \cdot (-1) = -20
 \end{aligned}$$

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

$$\begin{aligned}
 y[2,0] &= x[1,-1] \cdot h[1,1] + x[2,-1] \cdot h[0,1] + x[3,-1] \cdot h[-1,1] \\
 &\quad + x[1,0] \cdot h[1,0] + x[2,0] \cdot h[0,0] + x[3,0] \cdot h[-1,0] \\
 &\quad + x[1,1] \cdot h[1,-1] + x[2,1] \cdot h[0,-1] + x[3,1] \cdot h[-1,-1] \\
 &= 0 \cdot 1 + 0 \cdot 2 + 0 \cdot 1 + 2 \cdot 0 + 3 \cdot 0 + 0 \cdot 0 + 5 \cdot (-1) + 6 \cdot (-2) + 0 \cdot (-1) = -17
 \end{aligned}$$

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

$$\begin{aligned}
 y[0,1] &= x[-1,0] \cdot h[1,1] + x[0,0] \cdot h[0,1] + x[1,0] \cdot h[-1,1] \\
 &\quad + x[-1,1] \cdot h[1,0] + x[0,1] \cdot h[0,0] + x[1,1] \cdot h[-1,0] \\
 &\quad + x[-1,2] \cdot h[1,-1] + x[0,2] \cdot h[0,-1] + x[1,2] \cdot h[-1,-1] \\
 &= 0 \cdot 1 + 1 \cdot 2 + 2 \cdot 1 + 0 \cdot 0 + 4 \cdot 0 + 5 \cdot 0 + 0 \cdot (-1) + 7 \cdot (-2) + 8 \cdot (-1) = -18
 \end{aligned}$$

1 1	2 2	1 3
0 4	0 5	0 6
-1 7	-2 8	-1 9

$$\begin{aligned}
 y[1,1] &= x[0,0] \cdot h[1,1] + x[1,0] \cdot h[0,1] + x[2,0] \cdot h[-1,1] \\
 &\quad + x[0,1] \cdot h[1,0] + x[1,1] \cdot h[0,0] + x[2,1] \cdot h[-1,0] \\
 &\quad + x[0,2] \cdot h[1,-1] + x[1,2] \cdot h[0,-1] + x[2,2] \cdot h[-1,-1] \\
 &= 1 \cdot 1 + 2 \cdot 2 + 3 \cdot 1 + 4 \cdot 0 + 5 \cdot 0 + 6 \cdot 0 + 7 \cdot (-1) + 8 \cdot (-2) + 9 \cdot (-1) = -24
 \end{aligned}$$

1	2	3	1
4	5	6	0
7	8	9	-1

$$\begin{aligned}
 y[2,1] &= x[1,0] \cdot h[1,1] + x[2,0] \cdot h[0,1] + x[3,0] \cdot h[-1,1] \\
 &\quad + x[1,1] \cdot h[1,0] + x[2,1] \cdot h[0,0] + x[3,1] \cdot h[-1,0] \\
 &\quad + x[1,2] \cdot h[1,-1] + x[2,2] \cdot h[0,-1] + x[3,2] \cdot h[-1,-1] \\
 &= 2 \cdot 1 + 3 \cdot 2 + 0 \cdot 1 + 5 \cdot 0 + 6 \cdot 0 + 0 \cdot 0 + 8 \cdot (-1) + 9 \cdot (-2) + 0 \cdot (-1) = -18
 \end{aligned}$$

	1	2	3
1	2	1	
0	0	0	
-1	-2	-1	

$$\begin{aligned}
 y[0,2] &= x[-1,1] \cdot h[1,1] + x[0,1] \cdot h[0,1] + x[1,1] \cdot h[-1,1] \\
 &\quad + x[-1,2] \cdot h[1,0] + x[0,2] \cdot h[0,0] + x[1,2] \cdot h[-1,0] \\
 &\quad + x[-1,3] \cdot h[1,-1] + x[0,3] \cdot h[0,-1] + x[1,3] \cdot h[-1,-1] \\
 &= 0 \cdot 1 + 4 \cdot 2 + 5 \cdot 1 + 0 \cdot 0 + 7 \cdot 0 + 8 \cdot 0 + 0 \cdot (-1) + 0 \cdot (-2) + 0 \cdot (-1) = 13
 \end{aligned}$$

	1	2	3
1	4	5	6
0	7	8	9
-1	-2	-1	

$$\begin{aligned}
 y[1,2] &= x[0,1] \cdot h[1,1] + x[1,1] \cdot h[0,1] + x[2,1] \cdot h[-1,1] \\
 &\quad + x[0,2] \cdot h[1,0] + x[1,2] \cdot h[0,0] + x[2,2] \cdot h[-1,0] \\
 &\quad + x[0,3] \cdot h[1,-1] + x[1,3] \cdot h[0,-1] + x[2,3] \cdot h[-1,-1] \\
 &= 4 \cdot 1 + 5 \cdot 2 + 6 \cdot 1 + 7 \cdot 0 + 8 \cdot 0 + 9 \cdot 0 + 0 \cdot (-1) + 0 \cdot (-2) + 0 \cdot (-1) = 20
 \end{aligned}$$

	1	2	3
4	5	6	1
7	8	9	0
	-1	-2	-1

$$\begin{aligned}
 y[2,2] &= x[1,1] \cdot h[1,1] + x[2,1] \cdot h[0,1] + x[3,1] \cdot h[-1,1] \\
 &\quad + x[1,2] \cdot h[1,0] + x[2,2] \cdot h[0,0] + x[3,2] \cdot h[-1,0] \\
 &\quad + x[1,3] \cdot h[1,-1] + x[2,3] \cdot h[0,-1] + x[3,3] \cdot h[-1,-1] \\
 &= 5 \cdot 1 + 6 \cdot 2 + 0 \cdot 1 + 8 \cdot 0 + 9 \cdot 0 + 0 \cdot 0 + 0 \cdot (-1) + 0 \cdot (-2) + 0 \cdot (-1) = 17
 \end{aligned}$$

Dependiendo de los valores de la máscara se pueden obtener diferente efectos en la imagen como se ilustra en la figura 1.






Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	

Figura 1. Filtrado de imagen mediante convolución usando diferentes tipos de kernel.

Este proceso de convolución puede ser optimizado bajo la propiedad de un kernel separable en dos vectores un vector fila r , y un vector columna c , de forma que:

$$H = c \times r$$

Esto permite que se pueda realizar la convolución 2D en dos etapas una de convolución de filas con el vector r generando una imagen temporal I' a la cual se le aplica la convolución de columnas con el vector $c[1]$.

Se le solicita desarrollar una arquitectura y un sistema que permita realizar filtrado de imágenes mediante convolución con kernels de 3x3, bajo los siguientes requisitos generales de funcionalidad:

- 1- Las imágenes de entrada y salida deben ser almacenadas en memoria.
- 2- El tamaño de la imagen no debe ser muy grande (recomendación [640x480,1024x768])
- 3- El formato de las imágenes será en escala de grises con pixeles con valores entre [0, 255]
- 4- El sistema debe permitir la visualización de la imagen, en caso de no ser posible debe proveer un mecanismo para hacer volcado de memoria y obtener la imagen para visualizarla por aparte.
- 5- El sistema debe permitir la interacción del usuario para ejecutar el filtrado.

Requisitos de Arquitectura ISA

- 1- Debe diseñar un conjunto de instrucciones y arquitectura 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, puede implementar instrucciones de aritmética vectorial, aunque no hay un límite en la cantidad de instrucciones es importante que provea al menos instrucciones para control de flujo, operaciones aritméticas-lógicas, acceso a memoria.
- 3- 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.

Requisitos de Microarquitectura

- 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 procesador diseñado 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.
- 4- 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).
- 5- 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 su grupo es de 4 integrantes se le pide un plan de pruebas donde especifique los objetivos y descripción de las pruebas junto con sus resultados.

6- 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 la microarquitectura y descripción de las interacciones entre ellos.
- c- Simulaciones de las pruebas unitarias y de integración.
- d- Plan de pruebas* (n==4).

7- Reporte de consumo de recursos del FPGA.

Requisitos de software

- 1- Crear una aplicación (software) empleando la arquitectura diseñada, con el fin de hacer filtrado con diferentes kernels de 3x3, (*blur, sharpen, edge detection*).

Notas

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

Evaluación

La evaluación del proyecto se da bajo los siguientes rubros:

Presentación Funcional 60% (incluye las simulaciones)

Artículo científico con la descripción del proceso de diseño, simulaciones y resultados 20%

Documentación del proceso de diseño, pruebas* y *reference sheet* 20%

Este proyecto tiene la oportunidad de obtener hasta un máximo de 25% por criterio de puntaje extra, en función de optimización en el uso de recursos del FPGA, creatividad en el diseño, y plan de pruebas.

Por ejemplo:

- 1- Agregar lógica de prueba o medición (performance counters) para evaluar el desempeño.
- 2- Uso de kernels de tamaño variable.
- 3- Uso de imágenes en color y representación punto flotante.
- 4- Etapa de visualización no VGA.
- 5- Medición de kernel separable vs kernel no separable.
- 6- Uso de 'cache' para procesamiento de datos.

Referencias

[1] Karas Pavel, Svoboda David, and Gustavo Ruiz, "Algorithms for Efficient Computation of Convolution," in Design and Architecture for Digital Signal Processing, INTECH, 2013, pp. 179–208.

[2] [https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))