Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computadores

Programa de Licenciatura en Ingeniería en Computadores

Curso: CE-4302 Arquitectura de Computadores II

Especificación Proyecto 02: Diseño e Implementación de un Arreglo Sistólico para Unidad de Procesamiento Neural (NPU)

Profesores:

Luis Alonso Barboza Artavia

Ronald García Fernández

Fecha de entrega: 08-09 de Mayo, 2025

Semestre: I 2025

Objetivo general

Diseñar, implementar y verificar un arreglo sistólico [2] que funcione como unidad funcional de un Neural Processing Unit (NPU), acelerando operaciones fundamentales de redes neuronales (multiplicación de matrices y funciones de activación no lineales), con control, e interfaz con memoria externa.

Atributos relacionados: Diseño (DI).

Diseña soluciones creativas para problemas de ingeniería complejos y diseña sistemas, componentes o procesos para satisfacer las necesidades identificadas con la consideración adecuada para la salud y la seguridad públicas, el costo total de la vida, el carbono neto cero, así como las consideraciones de recursos, culturales, sociales y ambientales según sea necesario.

DI1- Identifica las necesidades y los requerimientos de un problema complejo de ingeniería considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.
DI2- Valora alternativas de solución para un problema complejo de ingeniería que cumplan con necesidades específicas, considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.
DI3- Diseña de forma creativa, la alternativa seleccionada que cumpla con las necesidades específicas para resolver el problema complejo de ingeniería, considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.
DI4- Valida el diseño final de acuerdo con los requerimientos, la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.

Motivación

Los NPUs modernos utilizan arreglos sistólicos para acelerar cargas de trabajo de inteligencia artificial mediante operaciones matriciales paralelas. Este proyecto permite aplicar conceptos avanzados de arquitectura, gestión eficiente de memoria externa y estrategias de flujo de datos realistas.

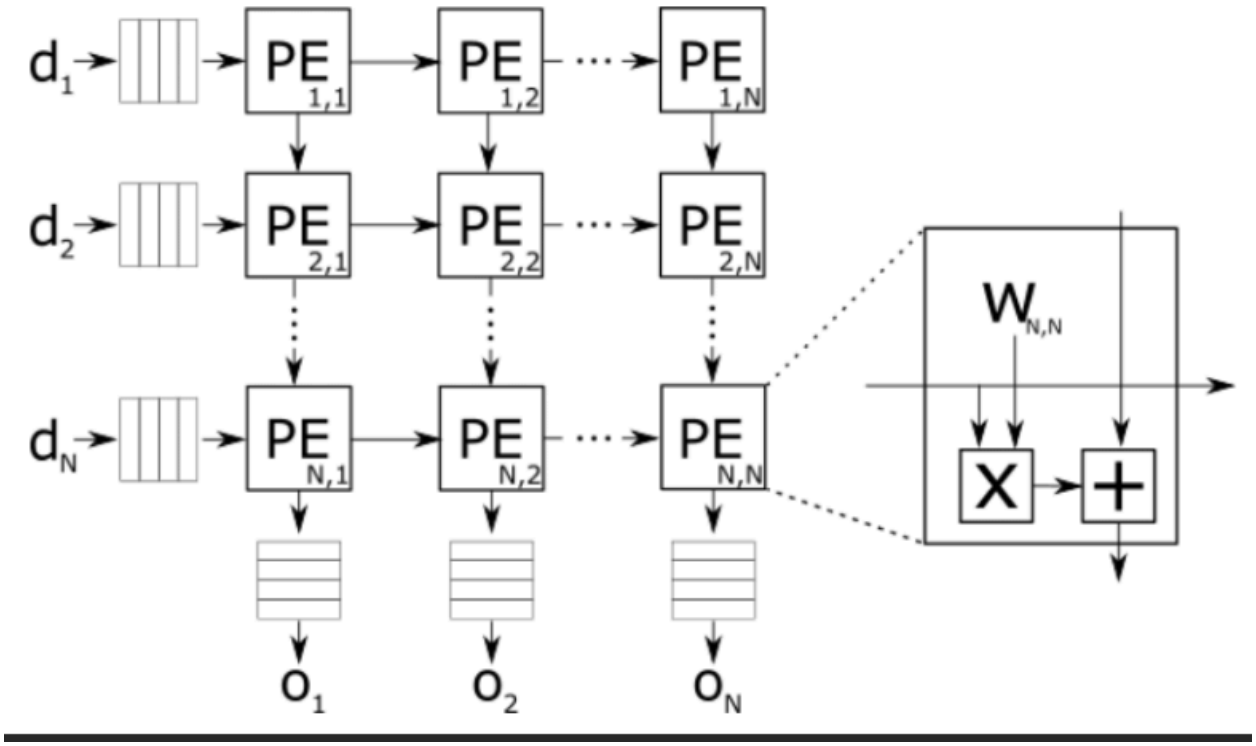

Figura 1. Ejemplo de una arquitectura tipo arreglo sistólico[3].

Mediante el diseño, implementación y verificación de una arquitectura basada en un arreglo sistólico se pretende la aplicación de conocimientos típicos de procesamiento vectorial e introducir de forma práctica-teórica al tema de arquitecturas de dominio específico [1] (*domain specific architectures*).

Características generales del proyecto

1- Implementar operaciones de multiplicación y acumulación en un arreglo sistólico escalable (mínimo 4x4 PEs).
2- La arquitectura debe soportar funciones de activación no lineales (ejemplo ReLU).
3- Debe implementar interfaz con memoria externa SDRAM de 64 MB (32Mx16) en la tarjeta DE1-SoC-MTL2 para almacenamiento de datos y resultados.
4- Permitir control y comunicación con PC vía IP JTAG-Over-Protocol (JOP) o UART.
5- Implementar mecanismo de ejecución por pasos (*stepping*) para depuración.
6- Utilizar enteros con signo para datos y coeficientes, con tamaño (por ejemplo, 16 o 32 bits) definido y justificado según precisión, rango dinámico, recursos y rendimiento.
7- Proveer registros de control y estado para configuración, monitoreo y manejo de excepciones.
8- Diseñar una arquitectura modular y *pipelined* [5], sintetizable para Cyclone V y DE1-SoC MTL2.
9- Desarrollar un modelo de referencia en C/C++ que implemente las mismas operaciones con el mismo formato numérico para validar la funcionalidad hardware.

10- Definir y ejecutar un plan de validación con simulaciones unitarias, de integración y pruebas en hardware.
11- Implementar *performance counters* para medir operaciones aritméticas y accesos a memoria, permitiendo estimar la intensidad aritmética aproximada.
12- Implementar y documentar un flujo de datos realista para el arreglo sistólico, seleccionando y justificando uno de los esquemas comunes en NPUs actuales: Weight Stationary (WS), Input Stationary (IS) o Output Stationary (OS) [4].
13- No se permite el uso de multiplicadores de matrices preconstruidos ni IPs que realicen la multiplicación matricial completa. Todos los bloques funcionales deben ser diseñados por el grupo de trabajo.

Requisitos de arquitectura:

1- Se debe definir un conjunto de registros para control y acceso a datos.
2- Se debe definir un flujo de datos a soportar (por ejemplo Weight Stationary (WS), Input Stationary (IS) o Output Stationary (OS))
3- Los datos y coeficientes deben representarse con enteros con signo.
4- El tamaño de palabra debe ser definido y justificado considerando precisión, rango dinámico, uso eficiente de recursos y rendimiento.
5- El formato debe ser consistente en toda la implementación, incluyendo el modelo de referencia en C/C++ y el hardware.

Requisitos de Hardware:

1- Plataforma: Tarjeta DE1-SoC MTL2 con FPGA Cyclone V.
2- Arreglo sistólico con PEs que realicen multiplicación-acumulación y activación no lineal.
3- Interfaz con memoria SDRAM de 64 MB para almacenamiento y acceso de datos.
4- Debe soportar comunicación FPGA-PC mediante IP JTAG-Over-Protocol [6] o UART para control, carga/lectura de datos y stepping.
5- Mecanismo de ejecución por pasos (stepping) controlable y documentado.
6- Debe implementar los registros de control, estado y performance counters.
7- Debe soportar el tamaño y formatos de enteros con signo elegido.
8- Debe ser capaz de correr el flujo de datos realista [4]*
9- Se debe implementar un diseño modular, *pipelined* y sintetizable en SystemVerilog.
10- Se prohíbe el uso de multiplicadores de matrices preconstruidos o IPs que realicen la multiplicación matricial completa.

Requisitos de Software

1- Modelo de referencia en C/C++ con el mismo formato numérico.
2- Se debe implementar una aplicación simple para control, stepping y validación mediante la PC.
3- Implementar 2 Testbenches para pruebas unitarias y de integración
4- Cada prueba debe claramente decir si pasa o falla de forma automática (no es permitido que sea realizado por inspección visual)

Requisitos de documentación

Se debe entregar los siguientes documentos:

Artículo científico conteniendo como mínimo los siguientes aspectos:

1- Motivación y arquitectura.
2- Identificación de las necesidades y los requerimientos considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.
3- Valora alternativas de solución para un problema complejo de ingeniería que cumplan con necesidades específicas, considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario.
4- Diseña de forma creativa, la alternativa seleccionada que cumpla con las necesidades específicas para resolver el problema complejo de ingeniería, considerando la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario. Como mínimo se espera:
    a. Proceso de diseño y estrategia de verificación.
    b. Justificación y descripción del flujo de datos.
    c. Justificación del tamaño y tipo de datos.
    d. Consideración de la salud y la seguridad pública, el costo total de la vida, el carbono neto cero, así como aspectos relacionados con recursos, culturales, sociales y ambientales según sea necesario
5- Análisis de resultados y *performance counters*.
6- Conclusiones y recomendaciones.
7- Referencias bibliográficas

Plan de verificación que incluya:

1- Descripción de las pruebas a realizar y su justificación.
2- Reporte Simulaciones unitarias y de integración.

Entregables

Se debe entregar los siguientes archivos:

1- Código SystemVerilog sintetizable.
2- Modelo de referencia en C/C++ y aplicación para comunicación con la PC.
3- Código de los testbenches para simulaciones unitarias y de integración.
4- Plan de verificación.
5- Artículo científico.
6- Bitstream para DE1-SoC MTL2.

Notas:

1- Durante el proceso de diseño se deben evaluar diferentes propuestas, entiéndase como proceso de diseño a los pasos, ideas y discusiones necesarias para obtener una solución dada.
2- Es obligatoria la entrega del plan de pruebas para la revisión funcional, de no ser así no se revisará el proyecto.
3- Aunque no es obligatorio se recomienda realizar reuniones con el profesor para obtener guía al respecto.

Evaluación

El proyecto se desarrollará en grupos de 3 integrantes como máximo.

La evaluación del proyecto se da bajos los siguientes rubros:

1- Presentación Funcional (**60%**) todos los miembros del grupo deben estar presentes en una sesión demostrativa (previa cita con el profesor) la funcionalidad del sistema, en la cual se realizarán preguntas sobre cualquier etapa del sistema, según la rúbrica correspondiente.

2- Artículo científico (**20%**) con la descripción del proceso de diseño del sistema y análisis de resultados, según los requisitos de documentación.

3- Plan de verificación (**20%**) con la descripción de pruebas y resultados según los requisitos de documentación.

Si tiene dudas puede contactar al profesor por medio de correo electrónico, la entrega debe ser realizada mediante el Tec Digital en la pestaña de evaluaciones no se aceptarán entregas después de las 11:59PM del 17 junio (grupo 2) y 18 junio (grupo 1) 2025.

Las defensas serán 17 junio (grupo 2) y 18 junio (grupo 1).

Referencias

[1] John L Hennessy y David A Patterson. Computer Architecture: A Quantitative Approach. Elsevier, 2017

[2] https://en.wikipedia.org/wiki/Systolic_array visitada el 01-05-2025.

[3] Neggaz, Mohamed. Hardware Accelerators for Machine Learning Applications. Case Study : Autonomous Vehicles. 2020

[4] T. Raja, Systolic Array Data Flows for Efficient Matrix Multiplication in Deep Neural Networks, arXiv preprint arXiv:2410.22595v1, Oct. 2024.

[5] https://zipcpu.com/blog/2017/08/14/strategies-for-pipelining.html visitada el 08-05-2025

[6] https://www.intel.com/content/www/us/en/docs/programmable/728673/21-3/jtag-over-protocol-overview.html visitada el 07-05-2025