TEC | Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica
Área Académica de Ingeniería en Computadores
CE 4302 — Arquitectura de Computadores II

Proyecto Grupal 1:
Diseño e Implementación de un ASIP vectorial para encriptar imágenes

Fecha de asignación:	12 de abril 2023	Fecha de entrega:	17 de mayo 2023
Grupos:	4 o 5 personas	Profesor:	Luis Barboza Artavia

Mediante el desarrollo de este proyecto, el estudiante aplicará los conceptos de paralelismo a nivel de datos, específicamente procesadores *Single Instruction Multiple Data* (SIMD) del tipo vectorial. Se realizará un *Application Specific Instruction Set Processor* (ASIP) para el tratamiento de imágenes en algoritmos de encriptación.

1. Atributos relacionados

A continuación se describen los atributos del graduado que se pretenden abordar con el desarrollo del proyecto.

1.1. 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.

El atributo de diseño será evaluado tanto formativamente (reuniones de seguimiento con el profesor) como sumativamente, en especial en la sección de documentación de diseño de los entregables.

2. Descripción General

El paralelismo en múltiples niveles ha sido la motivación del diseño de computadoras con el fin de mejorar el rendimiento términos de energía y costo. Un tipo básico de paralelismo es a nivel de datos donde surge porque hay muchos datos que pueden ser operados al mismo tiempo. Las arquitecturas vectoriales son los referentes en este aspecto porque utilizan el concepto de SIMD para explicar el paralelismo a nivel de datos aplicando una única instrucción a una colección de datos de manera paralela. Los diseñadores han encontrado que SIMD trae ventajas puesto que es potencialmente eficiente en energía que otras técnicas como MIMD.

Para este proyecto se aplicarán los conceptos de arquitectura de computadoras para el diseño e implementación de un procesador vectorial. La arquitectura del set de instrucciones (ISA) será propuesta por cada grupo según las necesidades de la aplicación para diversos algoritmos de encriptación de imágenes.

En el proyecto se desarrollará un acercamiento práctico al diseño de un set de instrucciones propio y específico, la realización en hardware de un modelo funcional de un procesador vectorial que implemente el set propuesto, y programación de sistemas computacionales en general.

2.1. Algoritmos de encriptación

Los algoritmos de encriptación serán los siguientes:

- **XOR con clave privada**: Este tipo de encriptación es uno de los más utilizados como base de algoritmos criptográficos más complejos, como AES, por ejemplo. Para este algoritmo, a de cada pixel (i,j) deberá aplicársele una operación XOR con un dato de 8 bits, denominado clave privada. Para desencriptar una imagen encriptada con este algoritmo, debe aplicarse el mismo proceso.
- **Desplazamiento circular**: Este algoritmo deberá aplicar un desplazamiento circular de una cantidad entre 0-255 a cada pixel, lo que implica que los datos que serán desplazados no se perderán, sino que pasan del bit más significativo al menos significativo, y viceversa. Para desencriptar, se deberá aplicar el desplazamiento circular en la dirección contraria a la encriptación, para la misma cantidad de bits desplazados.
- **Algoritmo de encriptación propio**: cada grupo deberá investigar sobre algoritmos de encriptación de imágenes utilizados en seguridad informática. Luego de dicha investigación, deberá seleccionar e implementar uno para la arquitectura propia. **Nota: los grupos de 5 personas deben implementar dos algoritmos investigados**.

3. Especificación

Se le solicita desarrollar una **arquitectura** y una **microarquitectura** que realice la ejecución de los algoritmos de encriptación en imágenes. Se usará una imagen en escala de grises de entrada libre con dimensión mínima de 256×256. En la salida se mostrará la imagen final luego de realizado el algoritmo.

Se deben seguir los siguientes requisitos generales de funcionalidad:

1. El diseño completo debe poder ser sintetizable en una tarjeta de desarrollo Terasic DE1-SoC (debe caber todo ahí, inclusive la imagen de entrada y el producto de la encriptación).
2. Las imágenes de entrada y salida deben ser almacenadas en memoria (se recomienda usar el bloque IP de la biblioteca de Quartus).
3. El sistema debe permitir la interacción con el usuario, para poder escoger el algoritmo de encriptación o desencriptación mediante algún periférico (e.g., botones, switches).

4. El ISA debe ser eficiente y congruente, con criterios de diseño definidos. Es importante hacer reuniones con el profesor para guía.

5. Las imágenes estarán en escalas de grises.

6. El tamaño del vector deberá ser de al menos 6 bytes, es decir, cada vector deberá tener, al menos, 6 datos de 8 bits cada uno.

7. La imagen resultante debe ser mostrada por medio de VGA[1]. Se sugiere una resolución de 640x480 pixeles.

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. Se deberán incluir instrucciones con operaciones vector-vector y vector-escalar.

3. Los productos finales de esta etapa son el *instruction reference sheet* o *green sheet*.

4. El ISA debe ser personalizado y realizado por los estudiantes, no se aceptarán ISAs ya diseñados (e.g., ARM, x86, RISC-V, otros). Debe justificar cada característica del mismo.

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. **No se revisará si no tiene pipeline.**

[1]Se recomienda ver el Capítulo 9 del libro Harris & Harris

3. Debe ser implementado usando SystemVerilog.

4. No se permite realizar módulos especializados de hardware. Es un curso de Arquitectura de Computadores, no de Diseño de Sistemas Digitales.

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 (GPIO, volcado de memoria, switches, etc).

6. Cada unidad funcional del sistema debe ser debidamente probada en simulación, para verificar su funcionamiento correcto (unit tests). Además debe incluir pruebas de integración y sistema. Se le solicita un plan de pruebas donde especifique los objetivos y descripción de las pruebas junto con sus resultados.

7. El procesador debe poseer al menos 4 *lanes* para ejecutar de manera paralela las operaciones en los vectores.

8. 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) Reporte de consumo de recursos del FPGA para el modelo.

Requisitos de Software:

1. Crear una aplicación (software) empleando la arquitectura diseñada, con el fin de implementar la aplicación descrita.

2. Debe realizar un programa ('compilador') que permita traducir las instrucciones del ISA a binario, con la finalidad de ejecutarlo en el procesador. No es necesario que realice análisis léxico, sintáctico y semántico (este curso no es de Compiladores).

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

4. Evaluación y entregables

La defensa será el mismo día de la entrega y todos los archivos (incluyendo código fuente) serán entregados a las 11:59 pm ese mismo día (realícenlo progresivamente y no lo deje para el final). La evaluación del proyecto se da bajos los siguientes rubros contra rúbrica correspondiente:

- Presentación proyecto 100 % funcional (65 %): cada grupo deberá demostrar en una sesión (previa cita con el profesor) de 30 minutos los diferentes componentes del proyecto. El profesor evaluará las pruebas según rúbrica correspondiente. En la sesión se harán preguntas relacionadas sobre cualquier etapa del sistema. Se habilitará un espacio en el tec digital para colocar un enlace del repositorio con el código fuente del proyecto (**siguiendo la metodología de trabajo descrita en la especificación del proyecto 1**). Se revisará el commit antes del 17 de mayo 2023 a las 17:00. En la defensa se debe presentar lo siguiente:

  1. Todo el diseño debe ser sintetizable en una tarjeta: Terasic DE1-SoC.
  2. Debe reservar los espacios de memoria para la imagen de salida.

  Los entregables adicionales que se revisarán en la defensa son los siguientes:

  1. Arquitectura:
     a) *Instruction reference sheet* o *green sheet*.
  2. Microarquitectura:
     a) Diagrama de bloques de la microarquitectura.
     b) Reporte de consumo de recursos del FPGA.
  3. Software
     a) Programa de software.
     b) Compilador usado.
  4. Plan de pruebas: debe incluir las simulaciones de las pruebas unitarias de los diferentes módulos.

- Documentación de diseño (20 %): La documentación del diseño deberá contener las siguientes secciones:

  1. Listado de requerimientos del sistema: Cada grupo deberá identificar 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.

2. Elaboración de opciones de solución al problema: Para el problema planteado deberán documentarse al menos dos opciones de solución. Cada solución deberá ser acompañada de algún tipo de diagrama. **Estas opciones de solución no deben ser fácilmente descartables y deben llevar un análisis objetivo con base en criterios técnicos o teóricos.**
3. Valoración de opciones de solución: Se deberán valorar 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. Selección de la propuesta final: Se deberá seleccionar una propuesta final de las opciones de solución, de acuerdo con los criterios de comparación.
5. Diseño de la alternativa seleccionada: Se deberá documentar completamente el diseño final seleccionado 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. Para el caso de este proyecto esto incluye: descripción de arquitectura del set de instrucciones (ISA), diagrama de bloques del modelo del procesador, diagrama de bloques del computador (procesador + interfaz con aplicación), diagramas propios de diseño de sofware aplicables (de flujo, clases, composición, UML, patrones de diseño, etc) y descripción de algoritmo propuesto.
6. Validación del diseño: se deberá validar 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.

- Presentación del *paper* y resultados (15 %): Cada grupo deberá grabar y entregar un video de **4:30 minutos a 5:30 minutos**, presentando la idea de su solución (puede utilizar diapositivas o algún otro medio de referencia). Debe considerar que el público meta de su presentación no tiene necesariamente el *background* técnico, por lo que deberá exponer de forma clara lo que se ha realizado, así como los resultados más importantes de su diseño y conclusiones. Puede utilizar el canal de Youtube "Two Minute Papers", como referencia. Por motivos de acreditación del programa, se recomienda vehemente que el enlace proporcionado esté disponible por cualquier persona que tenga acceso a él, hasta un año después de entregada esta evaluación.
- Puntos extra (10 %): El grupo realiza el video (grabación y contenido) del entregable anterior en inglés.