

Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores
(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores
(Licentiate Degree Program in Computer Engineering)

Curso: CE-4302 Arquitectura de Computadores II
(Course: CE-4302 Computer Architecture II)



Especificación Proyecto I
(Project I specification)

Profesor:
(Professor)

Ing.Jeferson González Gómez, M.Sc

Fecha de entrega: 18 de Abril de 2018
(Due Date: April 18th, 2018)

Proyecto I. Computador con sistema multiprocesador heterogéneo

1. Objetivo

Mediante el desarrollo de este proyecto el estudiante aplicará los conceptos de sistemas multiprocesadores, esquemas de memoria y procesamiento paralelo (TLP + DLP), en el diseño de un computador con multiprocesador heterogéneo, que incluya la interacción entre un procesador (hard processor) Arm Cortex A-9 y un co-procesador vectorial a diseñar, en un tarjeta de desarrollo TerasIC DE1-SoC.

Atributos relacionados: Análisis de problemas (AP), Diseño (DI).

2. Descripción general

Los sistemas multiprocesador heterogéneos son la base funcional de los computadores modernos. Desde los Sistemas en Chip (SoC), para sistemas embebidos de consumo como dispositivos móviles, hasta los *clusters* de computación de alto desempeño y procesamiento masivo paralelo, la interacción entre múltiples procesadores así como la capacidad para compartir tareas y recursos han sido un problema a resolver para los diseñadores de computadores actuales. Las interfaces de memoria compartida, en tales escenarios, representan además uno de los desafíos más grandes.

Para este proyecto, se deberán aplicar los conceptos de arquitectura de computadores, vista como una combinación de elementos de software y hardware, en el diseño e implementación de un computador basado en un sistemas multiprocesador en los que se aplica paralelismo a nivel de hilos, así como paralelismo a nivel de datos, con la debida interfaz de memoria compartida y sincronización entre los procesadores. Este sistema realizará una aplicación de procesamiento digital de imágenes, en la que cantidad de datos a procesar crea la necesidad de un sistema multiprocesador, para lograr un mayor desempeño.

En el proyecto se desarrollará un acercamiento práctico al diseño de un procesador vectorial, así como interfaces de memoria compartida y sincronización, desde el punto de vista de diseño, integración y programación de sistemas computacionales en general.

3. Especificación

Para este proyecto, se deberá diseñar e implementar un computador basado en una arquitectura multiprocesador, con memoria compartida, en la que se pueda distribuir trabajo computacional entre un procesador comercial y un co-procesador, cuya arquitectura deberá ser diseñada específicamente para la aplicación que se describirá más adelante.

Una vez diseñado e integrado el sistema a nivel de hardware, se deberá diseñar una aplicación que aplique métodos de procesamiento digital de imágenes, a describir posteriormente, de manera paralela entre un procesador maestro y un procesador esclavo. Para el caso de este proyecto, el procesador esclavo será descrito a partir de una arquitectura vectorial, propuesta por cada grupo de trabajo. A continuación se describe a mayor detalle la especificación del proyecto:

ISA vectorial

Para el desarrollo del proyecto, deberá plantearse como punto inicial, la arquitectura del set de instrucciones (ISA) que utilizará como base para el diseño en hardware del co-procesador, así como la programación sobre el mismo. El set de instrucciones deberá poseer la documentación adecuada sobre todos los elementos del mismo. Será importante detallar cada una de las instrucciones en cuanto a funcionalidad, sintáxis, modos de direccionamiento, formato, tipo de datos, encodificación, etc. En este punto, debe tenerse en cuenta además la cantidad y tipo de registros de propósito general, y la interfaz con memoria (esquema Von Neumann, Hardward). Cada decisión tomada en el set de instrucciones deberá ser justificada, con base a la aplicación específica (que se detalla más adelante), y aspectos de eficiencia, tomando en cuenta recursos (costo, área, potencia, etc). Como primero producto del proyecto, deberá generarse un documento con la descripción del set, así como un hoja de referencia rápida al set y los aspectos más importantes del mismo. En general, los sets diseñados deberán contar al menos con los siguientes tipos de instrucciones **vectoriales**: operaciones aritméticas, operaciones lógicas, carga, almacenamiento, desplazamientos regulares en ambas direcciones y desplazamientos circulares en ambas direcciones. El tipo de dato del set serán vectores de números enteros de 8 bits (el signo, o no, quedará a criterio de cada grupo con la debida justificación del caso). El tamaño del vector deberá ser de al menos 4 bytes, es decir cada vector deberá tener, al menos, 4 datos de 8 bits cada uno.

Se deberán incluir al menos 12 instrucciones, que deberán incluir operaciones vector-vector y vector-escalar.

Organización del sistema

Para el desarrollo de la plataforma de hardware del sistema multiprocesador se debe utilizar una tarjeta TerasIC DE1-SoC, la cual posee un procesador fijo Arm Cortex A-9, así como una FGPA Altera Cyclone V SoC 5CSEMA5F31C.

Desde punto de vista de organización, el sistema deberá utilizar el procesador Cortex A-9, así como el co-procesador vectorial, que deberá ser instanciado en el FPGA mediante la herramienta Qsys. Adicionalmente, se deberá incluir en la FPGA los módulos de memoria (para el co-procesador), controlador de PCI/AXI/otro (para la comunicación con el procesador Cortex A-9) y la lógica de sincronización para uso de memoria compartida. En la Fig. 1 se muestra un diagrama de bloques de la organización del sistema a desarrollar.

Para el caso del módulo de memoria y sincronización, adicional a cualquier otra lógica que desee agregarse, se recomienda utilizar la memoria en chip de la tarjeta FPGA. Esto con el fin de brindar una mayor eficiencia. La eficiencia será evaluada en la rúbrica correspondiente.

Para el procesador vectorial, el set deberá utilizar la técnica de segmentación (*pipeline*), para aumentar el desempeño. Para esto, deberá implementarse (de ser el caso) un lógica de detección de riesgos, que puede realizarse por software o por hardware. Adicionalmente, el procesador debe poseer al menos 2 *lanes* para la ejecución paralela de las operaciones en los vectores.

Aplicación

Desde el punto de vista de software, se deberá diseñar una aplicación que tome una imagen, la convierta a escala de grises y posteriormente aplique diferentes algoritmos de encriptación de imágenes. Este proceso debe ser compartido por los procesadores, de forma que se aproveche

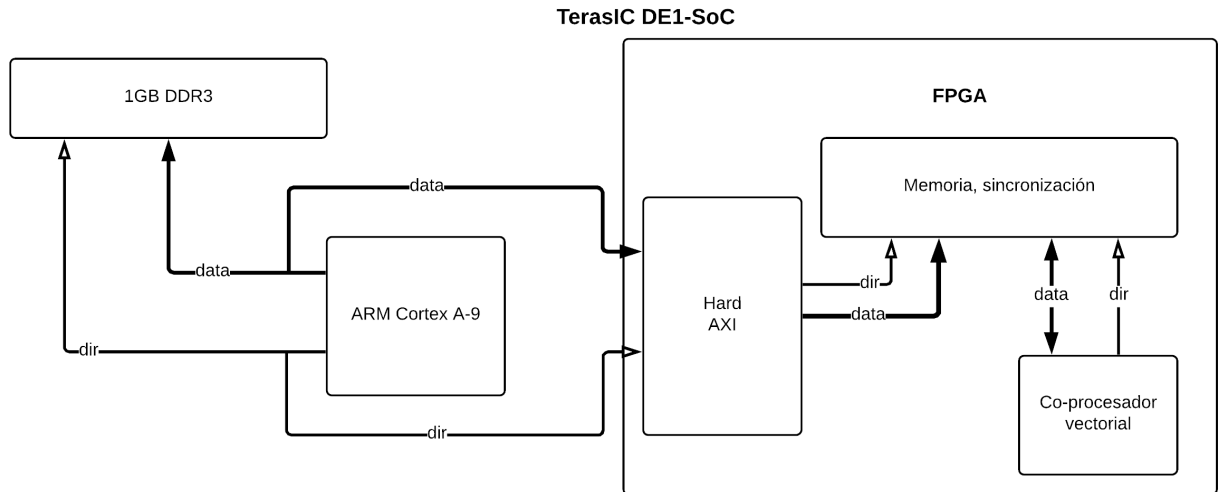


Figura 1: Diagrama de bloques de la organización del sistema a desarrollar

al máximo la capacidad de paralelismo a nivel de tareas/hilos entre ambos. Dentro del co-procesador vectorial se deberá realizar paralelismo a nivel de datos, aplicando las operaciones a vectores de píxeles.

La tarea principal (o maestra) deberá ejecutarse en el procesador Cortex A-9, bajo una distribución de Linux (CentOS, Ubuntu, personalizada, etc). La tarea principal deberá realizar todo el proceso de inicialización y conversión de la imagen a escala de grises, así como distribuir la carga computacional en los procesadores. Una vez distribuida la carga, el procesador Cortex A-9 (maestro) deberá alertar al procesador esclavo (vectorial) sobre el inicio del proceso, utilizando algún mecanismo de comunicación a través del hardware. Posteriormente, el procesador maestro debe comenzar con su segmento de procesamiento. De manera simultánea, el procesador esclavo deberá realizar su segmento de procesamiento, y alertar el procesador esclavo al finalizar la tarea, para que le sea asignado un nuevo segmento de procesamiento. Este proceso deberá repetirse hasta que el algoritmo de filtrado haya sido realizado completamente. Una vez finalizado el algoritmo, deberá mostrarse tanto la imagen original como la imagen filtrada.

Adicionalmente, el procesador maestro (Cortex A-9) deberá llevar un control del tiempo que tarda ejecutar la aplicación, así como, habilitar o inhabilitar estática o dinámicamente (por medio de una bandera o macro) el uso del procesador esclavo. En modo inhabilitado, el procesamiento deberá realizarse solamente en el procesador maestro. De igual manera, la fracción de computación a asignar al procesador esclavo deberá ser modificada estáticamente. Al finalizar este proyecto, se espera que se obtengan curvas del desempeño de la aplicación, para diferentes fracciones de computación (desde 0 hasta 100 %) y diferentes tamaños de imágenes. La comunicación entre los procesadores deberá realizarse por medio de algún protocolo adecuado (PCI, AXI, etc). Para esto, **deberá** crear o modificar un driver de bajo nivel, en linux, que represente el sistema de hardware diseñado.

Finalmente, la imagen deberá mostrarse tanto la imagen filtrada como la imagen original, en el sistema del procesador maestro.

A continuación se describe de una mejor manera los algoritmos a implementar.

Desde el punto de vista de software, se deberá diseñar una aplicación que a partir de una imagen en escala de grises (directamente pre-cargada en memoria, o cargada dinámicamente por algún método) aplique cada uno de los algoritmos de encriptación de imágenes que se describirán adelante, y muestre la imagen original, la imagen encriptada y la imagen desencriptada, en el sistema del procesador maestro.

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 al color (grises) 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 simple**: En este algoritmo, se deberá aplicar un desplazamiento de una cantidad definida de bits entre 1 y 7, hacia cualquier dirección, al color de cada pixel. Para desencriptar, se deberá aplicar un desplazamiento simple en la dirección contraria, con la misma cantidad de bits. Este algoritmo generará pérdidas en la información, a la hora de desencriptar.
- **Desplazamiento circular**: Este algoritmo será similar al anterior, pero el desplazamiento será circular, 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.
- **Suma simple**: En este algoritmo, al color de cada pixel dentro de un vector, deberá sumársele un valor determinado diferente, dentro de otro vector (clave). Así, para un vector de 4 pixeles [30,60,1,1], el vector clave a sumar (para toda la imagen) podrá ser, por ejemplo, [12, 5, 100, 10], y el resultado de color, para este primer vector de pixeles debe ser entonces [42,65,101,11]. Deberá considerarse asuntos de desbordamiento. Para desencriptar, deberá restarse cada vector de pixeles en la imagen con respecto al mismo vector clave, definido previamente.

Notas adicionales

- El desarrollo de este proyecto se dará en grupos de 4 personas.
- Para el uso de imágenes se recomienda la biblioteca OpenCV.

4. Entregables

Como entregables en este proyecto se evaluará lo siguiente:

- Presentación funcional completa (65 %). Se evaluará según rúbrica correspondiente.

- Paper (máximo 4 páginas) 15 %
 - Abstract - 1 %
 - Introducción - 2 %
 - Sistema desarrollado - 3 %
 - Resultados - 4 %
 - Conclusiones - 3 %
 - Referencias - 2 %
- Documentación de diseño (20 %)
 - Documento de descripción de arquitectura del set de instrucciones (ISA) del coprocesador. Deberá incluir la descripción completa del set, así como la hoja de referencia rápida del mismo.(10 %)
 - Metodología de diseño de sistema: Deberá especificar y detallar la metodología de diseño utilizada en el proyecto que involucre el análisis del problema, investigación respectiva, propuestas de diseño, comparación y evaluación de propuestas. (10 %)