

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería Electrónica



Diseño de una herramienta de generación automática de código para plataformas MPSoC

RWTH Aachen University

Chair for Software for Systems on Silicon, SSS.

Institute for Communication Technologies and Embedded Systems, ICE

Anteproyecto de Graduación

Ronny Jiménez Araya

200811178

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Ronny Jiménez Araya

Cartago, 16 de diciembre de 2013

Cédula: 1-1373-0768

Índice general

Índice de figuras	ii
Índice de tablas	iii
1 Entorno del proyecto	1
2 Definición del problema	4
2.1 Generalidades	4
2.2 Síntesis del problema	4
3 Enfoque de la solución	5
4 Resultados y Análisis	6
Bibliografía	7
A Demostración del teorema de Nyquist	8

Índice de figuras

Índice de tablas

Capítulo 1

Entorno del proyecto

Hoy en día los sistemas embebidos o empotrados se utilizan en muchos de los dispositivos y equipos con los cuales el ser humano interacciona o tiene relación alguna. Desde un dispositivos móvil como un teléfono celular, pasando por televisores de alta definición, equipos de seguridad en medios de transporte y hasta en la industria aeroespacial la cual requiere de los elementos mas robustos y de soporte critico, son ejemplos claros de la utilización de los sistemas embebidos.

En general, un sistema empotrado se considera como aquel conjunto de elementos de carácter electrónico y/o computacional-informático que se encuentran relacionados entre si y que tienen como función una aplicación especifica dada.[3] [9]

Hasta hace algunos años la arquitectura de estos dispositivos generalmente fue la utilización de un único procesador, con la tendencia del "escalado en frecuencia" [5] para aumentar las capacidades de los mismos. Sin embargo las tendencias han variado y han surgido los Sistemas en Chip de Múltiples Procesadores (*MPSoC, Multi-Processor Systems on Chip*), los cuales integran una gran variedad de dispositivos entre los cuales se encuentran elementos con múltiples núcleos de procesamiento.

Actualmente estos Sistemas en Chip (*SoC, Systems on Chip*) incorporan en sus unidades de procesamiento central múltiples núcleos de diferente naturaleza entre los cuales se encuentran los Procesadores de Propósito General (*GPP, Global Purpose Processors*) como la gama de procesadores ARM, Procesadores Digitales de Señales (*DSP, Digital Signal Processors*) que se encuentran de diversos tipos como los especializados para imágenes (*DIP, Digital Image Processor*), así también como los dedicados al tratamiento de gráficos llamados GPU (*GPU, Graphics Processing Units*) [10]. Todos los anteriores encapsulados en un mismo circuito integrado o bien incorporados en el SoC que los alberga.

Este cambio en el paradigma de la computación de pasar de sistemas de un solo núcleo de procesamiento a los MPSoC trae consigo retos que antes no se conocían, tanto para los proveedores como para los desarrolladores de las plataformas. Los modelos de programación cambian de un estándar secuencial a uno paralelo que justamente aproveche el paralelismo inherente a los múltiples procesadores, lo que conlleva nuevos métodos de programación, compilación, y ejecución que deben de cumplir y satisfacer los requerimientos característicos de los sistemas embebidos como operación en tiempo real [4], eficiencia energética y procesamiento limitado.

La Universidad de Aquisgran (*RWTH Aachen University*) es uno de los mejores centros académicos y de investigación en diversas ramas de la ingeniería. En la actualidad es reconocida mundialmente por los trabajos realizados en las áreas de ingeniería mecánica, electrónica y ciencias de la computación [1]. Dentro de su sistema académico, la universidad cuenta con diversos centros de investigación (institutos) que realizan aportes de punta con nuevas tecnologías y avances, concretando estos mediante publicaciones en revistas internacionales de alto renombre. Estos centros de investigación son dirigidos mediante Juntas (*Chairs*) que tienen a cargo diversos proyectos tanto en áreas específicas como en áreas multidisciplinarias.

Uno de los institutos mas reconocidos por sus aportes a la ciencia es el Instituto de Tecnologías de Comunicaciones y Sistemas Embebidos (*ICE, Institute for Communications Technologies and Embedded Systems*) [2], el cual integra tres de las juntas de mayor importancia en el área de la electrónica y las ciencias de la computación (*SSS, Software for Systems on Silicon; ISS, Integrated Signal Processing; Research Group MPSoC Architectures UMIC Research Cluster*); estos "Chairs" colaboran entre si en la generación de conocimiento, y cuentan con tres áreas vitales de las comunicaciones eléctricas y los sistemas embebidos.

- Algoritmos y esquemas de transmision y recepcion en los sistemas de comunicaciones inalambricas (Grupo Algoritmos)
- Hardware y arquitecturas de los chips para sistemas embebidos, en particular para los MPSoC (Grupo Hardware)
- Diseno de herramientas a nivel de sistema (*ESL, Electronic System-Level*) para sistemas y circuitos integrados complejos (Grupo Herramientas)

En la actualidad el Instituto cuenta con una aplicación llamada MAPS (*MAPS, MPSoC Application Programming Studio*) [6], la cual consiste en un sistema de herramientas que asisten a los programadores en la paralelización de código secuencial C. La interfaz del usuario esta basada en las herramientas de C/C++ del ambiente de desarrollo Eclipse.

A pesar de la gran cantidad de tiempo que se ha invertido en el desarrollo de la aplicación MAPS, esta no se encuentra concluida y solo cuenta con soporte para determinadas plataformas de desarrollo, entre ellas un soporte limitado para la KeyStone I [8], la cual es una plataforma de múltiples unidades de procesamiento (*PE, Processing Elements*) homogéneas, en este caso DSP's. Parte del soporte que del todo no se encuentra es para las arquitecturas de KeyStone II [7], cuya estructura consta de unidades de procesamiento pero heterogéneas, es decir, procesadores de propósito general ARM mas DSP's.

Capítulo 2

Definicion del problema

2.1 Generalidades

Como generalidad del entorno de los sistemas embebidos se encuentra la dificultad para los desarrolladores de trasladares de un paradigma de programación secuencial a uno que implique la utilización de múltiples unidades de procesamiento en forma paralela de las nuevas generaciones de plataformas multi núcleo, reutilizando para ello código ya existente (*Legacy Code*).

En términos generales existen herramientas para la plataforma KeyStone I, sin embargo estas no se encuentran optimizadas para un mejor consumo de los recursos y de la energía, por lo que la optimización de las mismas es un aspecto a considerar.

De forma mas especifica, para la plataforma KeyStone II no existe ninguna herramienta que logre programar sus múltiples núcleos heterogéneos de manera automática, generando código paralelo a partir de código secuencial en C.

2.2 Síntesis del problema

Inexistencia y falta de optimización de una herramienta de generación automática de código paralelo a partir de código secuencial C para las plataformas MPSoC KeyStone de Texas Instruments.

Capítulo 3

Enfoque de la solución

El enfoque que se pretende dar a la solución se basa en la utilización de un lenguaje de programación de alto nivel como lo es C, para el diseño e implementación de un algoritmo que permita la programación de diferentes unidades de procesamiento en plataformas MPSoC. El desarrollo va a ser realizado en un ambiente Linux que facilite el uso de los recursos de la plataforma así como la depuración y evaluación de los diferentes componentes de hardware que se emplearán.

Además del lenguaje a utilizar, se pretende manejar otros elementos a nivel de software como lo son simuladores, depuradores y las bibliotecas para compilación cruzada, los cuales estarán integrados dentro de un ambiente de desarrollo (*IDE, Integrated Development Environment*) de Texas Instruments que podrá ser descargado y utilizado sin algún costo. Sin embargo, deberá desarrollarse interfaces específicas (*API's, Application Programming Interface*) que puedan integrarse al IDE de Texas Instruments para la correcta validación de lo realizado.

Agregado a lo anterior se manipularán plataformas de hardware destinadas a ser los elementos objetivo de la programación, específicamente KeyStone I y II. La primera tiene como unidades de procesamiento 8 DSP's TMS320C66x, mientras que la segunda implementa en su arquitectura la misma cantidad y tipo de DSP's más 4 procesadores de propósito general ARM Cortex-A15.

También será necesario utilizar diversos instrumentos de medición como analizadores lógicos y osciloscopios que permitan la depuración mediante señales eléctricas de los comportamientos no deseados en el sistema en general.

Con lo anterior establecido entonces se podrá realizar la programación del algoritmo y su posterior depuración.

Capítulo 4

Resultados y Análisis

En tesis formales en este capítulo se exponen los diseños experimentales realizados para comprobar el funcionamiento correcto del sistema. Por ejemplo, si se realiza algún sistema con reconocimiento de patrones, usualmente esta sección involucra las llamadas *matrices de confusión* donde se compactan las estadísticas de reconocimiento alcanzadas. En circuitos de hardware, experimentos para determinar variaciones contra ruido, etc. También pueden ilustrarse algunos resultados concretos como ejemplo del funcionamiento de los algoritmos. Puede mostrar por medio de experimentos ventajas, desventajas, desempeño de su algoritmo, o comparaciones con otros algoritmos.

Recuerde que debe minimizar los “saltos” que el lector deba hacer en su documento. Por tanto, usualmente el análisis se coloca junto a tablas y figuras presentadas, y debe tener un orden de tal modo que se observe cómo los objetivos específicos y el objetivo general del proyecto se han cumplido.

Bibliografía

- [1] Excellence initiative, Noviembre 2013. URL http://www.rwth-aachen.de/cms/root/Die_RWTH/~emq/Exzellenzinitiative/lidx/1/.
- [2] Institute for communications technologies and embedded systems, Noviembre 2013. URL <http://www.ice.rwth-aachen.de/institute/about-us/>.
- [3] Doug Abbot. *Linux for Embedded and Real-time Applications*. Newnes, 2003.
- [4] Doug Abbot. *Linux for Embedded and Real-time Applications*, chapter 1, pages 2,3. Newnes, 2003.
- [5] Jack Ganssle. *The Art of Designing Embedded Systems*. Newnes, 2008.
- [6] W. Sheng H. Scharwächter R. Leupers G. Ascheid H. Meyr T. Isshiki H. Kunieda J. Cheng, J. Castrillon. Maps: An integrated framework for mp soc application parallelization. In *45th Design Automation Conference (DAC '08)*, pages 754–759, June 2008.
- [7] Texas Instruments. *KeyStone II User's Guide*, December 2013.
- [8] Texas Instruments. *TMS320C6678 Multicore Fixed and Floating-Point Digital Signal Processor*, Abril 2013.
- [9] Marilyn Wolf. *Computers as components: Principles of embedded computing design*. Morgan Kaufmann, 2007.
- [10] Marilyn Wolf. *High-Performance Embedded Computing*. Morgan Kaufmann, 2007.

Apéndice A

Demostración del teorema de Nyquist

El título anterior es solo un ejemplo ilustrativo. Éste teorema no ameritaría un apéndice pues es parte normal del currículum de Electrónica, pero apéndices usualmente involucran aspectos de esta índole, que se salen de la línea de la tesis, pero que es conveniente incluir por completitud.

Los anexos contienen toda información adicional que se considere pertinente agregar, como manuales de usuario, demostraciones matemáticas que se salen de la línea principal de la tesis, pero que pueden considerarse parte de los resultados del trabajo.