NTNU – Trondheim
Norwegian University of
Science and Technology

Energy Efficiency Experiments on Mali Powered Exynos 5 using OmpSs

Rune Holmgren

December 2014

PILOT PROJECT FOR MASTER THESIS

Department of Computer and Information Science

Norwegian University of Science and Technology

Supervisor 1: Professor Lasse Natvig

Co supervisor: Antonio Garcia Guirado

Problem statement

Here is the problem statement.

Acknowledgements

Here are the acknowledgements.

Abstract

This is the abstract.

Contents

Chapter 1

Introduction

1.1 Motivation

Increase of performance and power efficiency are the main goal of processor designers. Unfortunatly we are currently reaching the limits of the current strategies for further development. For some time, our processors have been strugeling to achive increased performance. Heat stops us from driving the clock frequency higher, while memory is lagging more and more behind. A solution to enable continued performance growth is multicore processors, and for the last decade this has been the focus. Unfortunatly adding cores will not be a sustainable solution forever. As the amount of cores grow, they are still competing for the same system resources and may have to wait for eachother to complete calculations on data with dependencies.

A promissing solution to this issue is heterogenous multi-processor systems. Heterognous multi-processor systems utilize multiple different processor cores in the same system. This allow different parts of a program to be executed on a suitable processor. By using a suitable core for each part of the program it is possible to achive better performance than homogenous multi-processor systems.

1.2 Project Scope and Goal

This pilot projects main goal is to do preliminary research and experiments on the energy efficiency of the Exynos 5 processor, with the intent to use the results next spring in my master

thesis. The goal of this research is to explore the potential of the task based programming model heterogenous multi-processor systems.

1.3 Problem Statement Interpretation and Approach

- Task 1: Implement or adapt suitable experiment applications for testing energy efficiency.
- Task 2: Implement some energy efficiency measurement application for both Arendale duo and Odroid-xu3.
- Task 3: Optimize experiment applications for both platforms.
- Task 4: Gather performance and energy efficiency results from the experiment applications on both platforms.
- Task 5: Analyze and evaluate experiment results.

TODO: Introduce how these tasks were solved.

1.4 Outline

TODO: This section need to be completed after the outline of the report is done.

Chapter 2

Related work

The heterogen property of this

Chapter 3

Background

3.1 Energy measurement

3.2 NEON

NEON is a general-purpose single input multiple data (SIMD) technology implemented in the ARM Cortex A series of processors. It is able to run SIMD instructions on 128bit registers. By utilizing the NEON unit of the ARM processors, it is possible to achive paralellism in each seperate core. This will often open for great performance boost on problems like the ones explored in this paper. Each register may be filled with single precission floating point numbers ranging from 8 to 64 bit each. In future generations of the ARM ISA there will be support for other data types as well. Different implementations of NEON exist in the Cortex A cores, and while the even the simple implementations in smaller cores like the A7 can give great performance boost, the implementations present in the newest cores are performing even better. The A15 offer two NEON units, and the instruction pipeline to start the cores are shorter than in simpler implementations.

3.3 Task based programming

Task based programming allow a programmer to work with parallel programs, with an abstraction from the paralellization itself. When programming with this model, the program can be

split into tasks which can run in parallel. When the program run, it will run a task manager as part of the program. This task manager can dynamically assign tasks to the processors, and the programmer does not have to handle all the time consuming tasks related to manual parallelisation. As long as the programmer correctly handle dependencies in the paralellized code, it will be possible to write this kind of code as if it was serial.

The task based programming model also allow simpler development of portable programs. When the program is running tasks on available CPUs, it is not a problem to allow it to run on larger or smaller numbers of processors, and even clusters can support the program. This model even allow the tasks to run on different types of processors in a hetrogenous enviroment.

3.4 OpenMP Super scalar

OpenMP Super scalar (OmpSs) is a extention of the OpenMP API to integrate features from the StarSs programming model. It is currently under development at the Barcelona Supercomputing Center. The goal of OmpSs is to extend the programming model to support a wide range og processors. The OmpSs programming model will run on a wide variety of different systems, such as traditional personal computers, clusters, shared memory systems and hetrogenous processors. While the software is not yet comlpeted or fully tested, there have been several reports exploring it's potentilal. The results have proven OmpSs as an efficient solution on both clusters and hetrogenous systems utilizing OpenCL and CUDA.

3.5 Heterognous multi-processor

Heterognous multi-processor systems have multiple different processors, opposed to traditional multi-processor systems. A typical modern processor have several processors, and a program can run effectivly by having threads running parts of theis work on each of them. This work is often of such a nature that it can run better on a different processor. Sometimes it can run just as well on multiple simple processor, while using less die space and energy. In other instances, an advanced processor with some special capabilities, like vector instructions, can be more efficient.

This kind of processors have a potential to help us overcome the challenges that are emerging in processor development. Unfortunatly they also introduce several new challenges.

3.6 Experiment platforms

3.6.1 Arendale Board

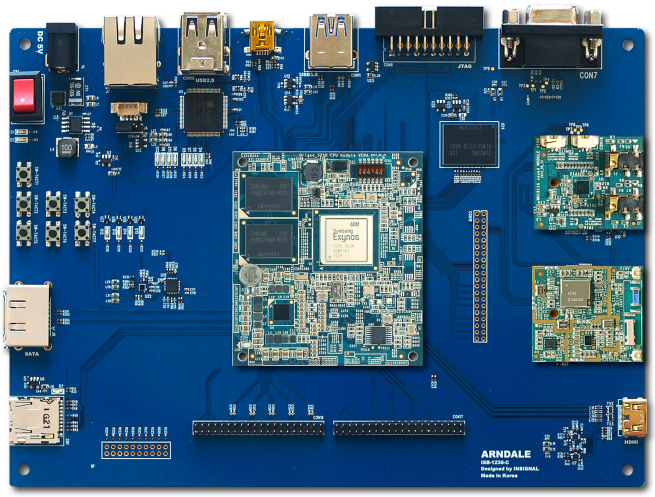

Figure 3.1: Arendale Duo

The Arendale Duo is a computing system mounted on a single board. It is fitted with an Exynos 5250 SoC, which contain a dualcore Arm Cortex-A15 , as well as an ARM Mali T-604 GPU. This computer offer a range of supported linux distributions, as well as the OmpSs programming model. The computer was used in the 2014 master thesis "Acceleration with OmpSs and Neon/OpenCL on ARM Processor" by Trond Inge Lillesand. The thesis lay alot of the ground for this pilot project and planned master thesis.

3.6.2 ODROID-XU3

The ODROID-XU3 is a new single-board computing system, offering interesting properties for these experiments. The system has an Exynos 5422 heterogenous Soc. Exynos 5422 has a quadcore ARM Cortex-A15 CPU and a ARM Mali T-628 GPU, but also a smaller quadcore ARM Cortex-

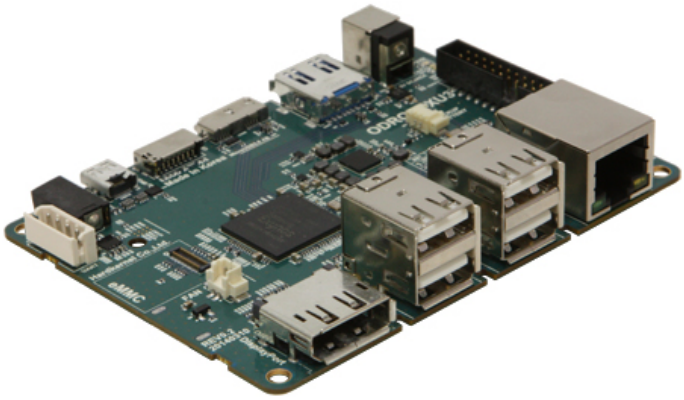

Figure 3.2: ODROID-XU3

A7 coprocessor. These 3 different processing units can be used simultaniously to solve problems. In this paper, and the planned master thesis following it, the potency of this kind of heterogenous processor will be explored.

3.6.3 ARM Cortex-A15

Performance	1.0 GHz to 2.5GHz
L1 Cache	64KB
L2 Cache	4 MB
L3 Cache	None in core, may be implemented shared in multicore system.
Architecture	ARMv7-A
Supported features	ARM Thumb-2
	TrustZone® security technology
	NEON™ Advanced SIMD
	DSP & SIMD extensions
	VFPv4 Floating point
	Hardware virtualization support
	Integer Divide
	Fused MAC
	Hypervisor debug instructions
Memory management	40-bit ARMv7 Memory Management Unit

3.6.4 ARM Cortex-A7

The ARM Cortex-A7 is designed to be a low power alternative to the ARM Cortex-A15 and ARM Cortex-A17, with the same supported ISA and features. This enable the ARM Cortex to be paired with it's largers relatives in a ARM big.LITTLE configuration.

Performance	1.2 GHz to 1.6GHz
L1 Cache	8-64KB
L2 Cache	up to 1 MB
L3 Cache	None in core, may be implemented shared in multicore system.
Architecture	ARMv7-A
Supported features	ARM Thumb-2
	TrustZone® security technology
	NEON™ Advanced SIMD
	DSP & SIMD extensions
	VFPv4 Floating point
	Hardware virtualization support
	Integer Divide
	Fused MAC
	Hypervisor debug instructions
Memory management	40-bit ARMv7 Memory Management Unit

3.6.5 ARM Mali T604

Performance	533 MHz
	17 GFLOPS
Multicore support	1-4 cores
API Support	OpenGL 1.1, 2.0, 3.0 and 3.1
	OpenCL 1.1
	DirectX 11
	RenderScript
Anti-Aliasing	4xFSAA with minimal performance drop
	16xFSAA
Cache	32-256KB L2 cache

3.6.6 ARM Mali T628

Performance	533/695 MHz
	17/23.7 GFLOPS
Multicore support	1-8 cores
API Support	OpenGL 1.1, 2.0, 3.0 and 3.1
	OpenCL 1.1
	DirectX 11
	RenderScript
Anti-Aliasing	4xFSAA with minimal performance drop
	16xFSAA
Cache	32-256KB L2 cache

3.7 Algorithms

Here I will write about the algorithms used in the experiments.

Chapter 4

Setup and Methodology

Chapter 5

Implementation

Chapter 6

Result and Discussion

Chapter 7

Conclution

Chapter 8

Future Work

These are some suggestions for future work that may build uppon the work in this thesis.

8.1 Experiment with heterogenousity

In this thesis, there have been done experiments with the Exynos 5, which support ARM big.LITTLE. The heterogen properties of this processor was outside of the scope of this pilot project. The same applications can be adapted and optimized to explore the potential of this processor architecture. This is planned for the master thesis following this pilot project.

8.2 OmpSs with OpenCL kernels

A new feature of OmpSs is it's ability to manage OpenCL kernels as tasks. It is possible to issue OpenCL kernels as OmpSs tasks, and have the task manager assign them to GPUs and CPUs. This allow for portable code that can run effectivly on a range of different system. It would be interesting to examine the potency of this way of utilizing the GPU, as it save the programmer from the job of manually tuning the loadbalance between GPU and CPU.

8.3 ARMv8-A 64-bit processors

ARM have created the next generation ARM processors. They run a new instruction set, with support for both 32- and 64-bit instructions. Running similar experiments on such a processor would be interesting.

Appendix A

Implementation

A.1 Introduction

A.1.1 Program 1

Bibliography

Rausand, M. and Høyland, A. (2004). *System Reliability Theory: Models, Statistical Methods, and Applications*. Wiley, Hoboken, NJ, 2nd edition.

Curriculum Vitae

RUNE HOLMGREN

DATATEKNIKKSTUDENT OG SYSTEMUTVIKLER

Utdanning

08.2010 - d.d. **Norges teknisk-naturvitenskaplige universitet (NTNU).**
Sivilingeniør, Datateknikk. Ferdig juni 2015.

08.2007 - 06.2010 **Heggen videregående skole.**
Studiespessialisering med realfag.

Erfaring som systemutvikler

06.2014 - 08.2014 **Webutvikler for Visma Consulting**
Sommerjobb på fulltid som utvikler og scrum master med angularJS og node.js med Leger uten grenser som kunde.

06.2013 - 09.2013 **Systemutvikler hos Connome AS.**
Sommerjobb på fulltid og deltid etter sommeren som androidutvikler mot hjemmeautomasjon.

09.2012 - d.d. **Webutvikler ved Studentersamfundet.**
Webutvikling mot samfundet.no i Ruby on Rails. Ca. 12 timer i uken.

2011 - 2012 **Androidutvikler i sammarbeid med Norsk Friidrett.**
Hobbyprosjekt som ble lansert i sammarbeid med Norsk Friidrett.

Øvrig arbeidserfaring

01.2013 - d.d. **Undervisningsassistent ved NTNU.**
Deltidsjobb som assistent i faget energieffektive datasystemer.

08.2012 - 11.2012 **Studentassistent ved NTNU.**
Deltidsjobb som assistent i faget algoritmer og datastrukturer.

06.2012 - 07.2012 **Drift og vedlikehold i Harstad Kommune.**
Sommerjobb.

01.2012 - 03.2012 **Studentassistent ved NTNU.**
Deltidsjobb på fysikkløypa, et tiltak for å fremme realfag i barneskolen.

07.2011 - 08.2011 **Driftslabratoriet hos Tine Meierier.**
Sommerjobb.

06.2010 - 08.2010 **Driftslabratoriet hos Tine Meierier.**
Sommerjobb.

04.2010 - 08.2010 **Plantasjen Harstad.**
Deltidsjobb parallelt med videregående.

06.2009 - 08.2009 **Privat ansatt i byggeprosjekt hos Byggmestrene Nilsen & Haukland.**
Sommerjobb.

06.2008 - 08.2008 **Byggmestrene Nilsen & Haukland.**
Sommerjobb.

06.2007 - 08.2007 **Byggmestrene Nilsen & Haukland.**
Sommerjobb.

06.2006 - 07.2006 **Aktiv byggpartner.**
Sommerjobb.

Kontaktinformasjon

Telefon: 99 49 26 99

E-mail: runholm@stud.ntnu.no

Addresse: Abels Gate 20A
7030 Trondheim

Kompetanseområder

Java	Javascript	Linux
Python	Ruby on Rails	WebGL
C	Android	VHDL
HTML	Photoshop	Norsk
CSS	InDesign	Engelsk
SQL	Vim	Tysk

Idrettserfaring

Aktiv i trener og funksjonærroller.

Aktivitetslederkurs, friidrett . (en helg)

Trener 1 kurs rettet, friidrett . (to helger)

Kretsdommerkurs i friidrett. (en helg)

Kurslærerkur, friidrett.

Ung:leder utdanning, friidrett (fire langhelger, samt faglig oppfølgning av mentor)

Sertifikater

Førerkort Kl. B

Referanser

Tilgjengelig ved forespørsel