COMPUTING SCIENCE

Experiments with Odroid-XU3 board

Rem Gensh, Ali Aalsaud, Ashur Rafiev, Fei Xia, Alexei Iliasov, Alexander Romanovsky and Alex Yakovlev

TECHNICAL REPORT SERIES

No. CS-TR-1471 May 2015

TECHNICAL REPORT SERIES

No. CS-TR-1471 May, 2015

Experiments with Odroid-XU3 board

Rem Gensh, Ali Aalsaud, Ashur Rafiev, Fei Xia, Alexei Iliasov, Alexander Romanovsky, Alex Yakovlev

Abstract

Power and energy consumption is a crucial factor for modern computer systems. The optimal operation of the system could be attained only if the interplay between performance and energy consumption was considered during the design. We performed several experiments with the Ordoid-XU3 board. The results of these experiments will help to model runtime for high performance and energy efficient system operations.

Printed and published by Newcastle University,
Computing Science, Claremont Tower, Claremont Road,
Newcastle upon Tyne, NE1 7RU, England.

Bibliographical details

GENSH, R; AALSAUD, A; RAFIEV, A; XIA, F; ILIASOV, A; ROMANOVSKY, A; YAKOVLEV, A;
Experiments with Odroid-XU3 board
[By] R. Gensh, A. Aalsaud, A. Rafiev, F. Xia, A. Iliasov, A. Romanovsky, A. Yakovlev
Newcastle upon Tyne: Newcastle University: Computing Science, 2015.

(Newcastle University, Computing Science, Technical Report Series, No. CS-TR-1471)

Added entries

NEWCASTLE UNIVERSITY
Computing Science. Technical Report Series. CS-TR-1471

Abstract

Power and energy consumption is a crucial factor for modern computer systems. The optimal operation of the system could be attained only if the interplay between performance and energy consumption was considered during the design. We performed several experiments with the Ordoid-XU3 board. The results of these experiments will help to model runtime for high performance and energy efficient system operations.

About the authors

Rem Gensh is a PhD student and a Research Technician at SRS group in the School of Computing Science of Newcastle University, Newcastle-upon-Tyne, UK. He graduated Kyrgyz-Russian Slavic University, Kyrgyzstan, with honors in 2008. After a 6-years extensive industrial experience as a software developer and a team leader, he reallocated to research area and started his PhD in 2014. He is involved in EPSRC/UK PRiME project. His research interests include fault tolerance, energy-efficient software design and many-core architectures.

Ali Majeed Aalsaud is a PhD candidate in Electronic engineering at Newcastle University.Ali completed his BSc in electronic and communication engineering in 1997 and then his MSc in electronic engineering, University of Technology in Baghdad- Iraq in 2000. Between 2000 and 2013, Ali was a Lecturer in computer and software engineering department at Al-Mustansiriya University in Baghdad.

Ashur Rafiev is an RA on the EPSRC PRiME Program Grant. In this project he leads the development of the ArchOn modelling environment.

Fei Xia is a Senior RA on the EPSRC PRiME Program Grant. Fei is with the EE School. His research interests include Asynchronous Data Communication. Asynchrnous System Design. Systems and Networks on Chip. Energy and Power in Computing.

Alexei Iliasov is a Researcher Associate at the School of Computing Science of Newcastle University, Newcastle-upon-Tyne,UK. He got his PhD in Computer Science in 2008 in the area of modelling artefacts reuse in formal developments. His research interests include agent systems, formal methods for software engineering and tools and environments supporting modelling and proof.

Alexander (Sascha) Romanovsky is a Professor in the Centre for Software and Reliability, Newcastle University. His main research interests are system dependability, fault tolerance, software architectures, exception handling, error recovery, system structuring and verification of fault tolerance. He received a PhD degree in Computer Science from St. Petersburg State Technical University and has worked as a visiting researcher at ABB Ltd Computer Architecture Lab Research Center, Switzerland and at Istituto di Elaborazione della Informazione, CNR, Pisa, Italy. In 1993 he became a postdoctoral fellow in Newcastle University, and worked on the ESPRIT projects on Predictable Dependable Computing Systems (PDCS), Design for Validation (DeVa) and on UK-funded projects on the Diversity, both in Safety Critical Software using Off-the-Shelf components. He was a member of the executive board of EU Dependable Systems of Systems (DSoS) Project, and between 2004 and 2012 headed projects on the development of a Rigorous Open Development Environment for Complex Systems (RODIN), and latterly was coordinator of the major FP7 Integrated Project on Industrial Deployment of System Engineering Methods Providing High Dependability and Productivity (DEPLOY). He now leads work on fault tolerance in Systems of Systems within the COMPASS project and is Principal Investigator of Newcastle's Platform Grant on Trustworthy Ambient Systems.

Alex Yakovlev received D.Sc. from Newcastle University in 2006, and M.Sc. and Ph.D. from St. Petersburg Electrical Engineering Institute in 1979 and 1982. Since 1991 he has been at the Newcastle University, where he worked as a lecturer, reader and professor at the Computing Science department until 2002, and is now heading the Microelectronic Systems Design research group (http://async.org.uk) at the School of Electrical, Electronic and Computer Engineering. His current interests and publications are in the field of modeling and design of asynchronous, concurrent, real-time and dependable systems on a chip. He has published four monographs and more than 200 papers in academic journals and conferences, has managed over 25 research contracts.

Suggested keywords

PERFORMANCE
POWER CONSUMPTION
ENERGY CONSUMPTION
VOLTAGE-FREQUENCY SCALING

Experiments with Odroid-XU3 board

R. Gensh, A. Aalsaud, A. Rafiev, F. Xia, A. Iliasov, A. Romanovsky, A. Yakovlev

Newcastle University
Newcastle upon Tyne, UK

{r.gensh, a.m.m.aalsaud, ashur.rafiev, fei.xia, alexei.iliasov,
alexander.romanovsky, alex.yakovlev}@ncl.ac.uk

Abstract. Power and energy consumption is a crucial factor for modern computer systems. The optimal operation of the system could be attained only if the interplay between performance and energy consumption was considered during the design. We performed several experiments with the Ordoid-XU3 board. The results of these experiments will help to model runtime for high performance and energy efficient system operations.

Keywords: performance, power consumption, energy consumption, voltage-frequency scaling.

1 Introduction

Nowadays computer technologies are developing very fast, and the architectures are becoming increasingly more complex. The development of multi-core and many-core systems is also fast. The popularity of heterogeneous architectures, containing two or more types of different CPUs is growing [1]. These systems offer better performance and concurrency, however it is necessary to ensure optimal power and energy consumption. The Odroid-XU3 board [2] allows us to better understand the nature of multi-core heterogeneous systems. The board provides the possibilities to apply techniques like voltage frequency scaling, affinity, and core disabling, which are used to optimize the system operation in terms of performance and energy consumption.

In the presented work various experiments were carried out in order to find the correlation between frequency, power consumption and performance in the heterogeneous system. The results of these experiments will provide ideas on how to reduce power and energy consumption without significant performance deteriorations. In addition, these results provide support for the initial validation of the parametric significance-driven modelling approach [3].

This paper is organised as follows. Section 2 provides experimental platform description. Section 3 gives a thorough description of the experiments. Section 4 gives the conclusions. Data obtained during the experiments is provided in the tables in the Appendix.

2 Platform description

The Odroid-XU3 board [2] is a small eight-core computing device implemented on energy-efficient hardware. The board can run Ubuntu 14.04 or Android 4.4 operating systems. The main component of Odroid-XU3 is the 28nm Application Processor Exynos 5422. The architecture of the processor is shown in Figure 1. This System-on-Chip is based on the ARM big.LITTLE [4] heterogeneous architecture and consists of a high performance Cortext-A15 quad core processor block, a low power Cortex-A7 quad core block, a Mali-T628 GPU and 2GB DRAM LPDDR3.

The board contains four real time current sensors that give the possibility of the measurement of power consumption on the four separate power domains: big (A15) CPUs, little (A7) CPUs, GPU and DRAM. In addition, there are also four temperature sensors for the each of A15 CPUs and one sensor for the GPU.

On the Odroid-XU3, for each power domain, the supply voltage (Vdd) and clock frequency can be tuned through a number of pre-set pairs of values. The performance-oriented Cortex-A15 block has a range of frequencies between 200MHz and 2000MHz with a 100MHz step, whilst the low-power Cortex-A7 quad core block can scale its frequencies between 200MHz and 1400MHz with a 100MHz step. Dynamic frequency scaling (DFS) is applied for A15 when its frequency ranges between 200MHz and 700MHz (the Vdd stays constant in this region) or for A7 when its frequency ranges between 200MHz and 500MHz. Dynamic voltage and frequency scaling (DVFS) is used when the frequency is 800MHz and above for A15 or 600MHz and above for A7.

The Cortex-A15 block is a high performance 32-bit quad core mobile processor using ARMv7-A instruction set. It has 32 KB instruction and 32 KB data caches. In addition, 2 MB Level 2 Cache is provided. Each A15 core has integrated floating point unit VFPv4.

Cortex-A7 has the same architecture and feature set as Cortex-A15, however Cortex-A7 microarchitecture provides optimum energy efficiency. It has 512 KB Level 2 Cache. The LITTLE Cortex-A7 processor is more suitable for performimg low power tasks like texting, background processes and audio.

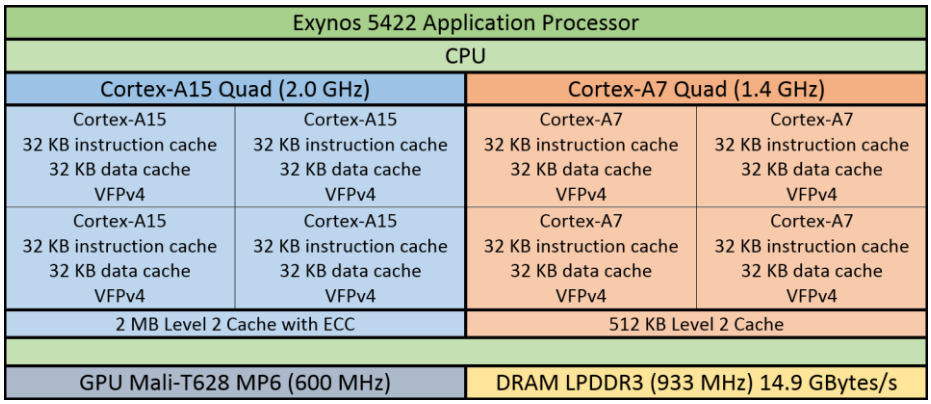

Fig. 1. Exynos 5422 diagram

3 Experiments

Experiments with the Odroid-XU3 platform were carried out in order to examine the power consumption under different operation frequencies and voltages. The frequency of each block can be changed independently using special utility programs and the system scales the operating voltage of the block to fit the chosen frequency. Eight cores in the board are numerated as follows: core 0, core 1, core 2 and core 3 belong to the A7 processor block, core 4, core 5, core 6 and core 7 belong to the A15 processor block.

Three types of experiments were carried out:

- Dynamic frequency scaling
- Duty cycle with idle-wait state
- Controlling the number of active cores

3.1 Dynamic frequency scaling

In the first part of this experiment, voltage, current and power were measured on A7 and A15 power domains without any additional workload, with only Ubuntu 14.04 OS running. Experimental data can be seen in Table 1 and Table 2. Figure 2 and Figure 3 represent the voltage-frequency characteristics of A7 and A15 power domains in this experiment. It should be noted that below some frequency, voltage remains the same, but above this point, the voltage linearly increases. As an example, A7 has a voltage of 0.913V at frequencies 200MHz – 500MHz, meanwhile A15 has a voltage of 0.9125V at frequencies 200MHz – 700MHz. This experiment clarifies the voltage-frequency dependencies for A7 and A15 cores in Odroid-XU3 board.

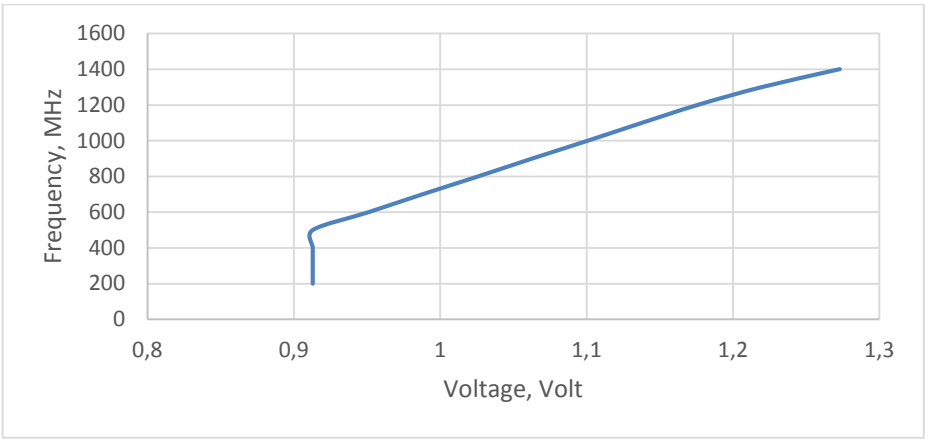

Fig. 2. Cortex-A7 voltage-frequency characteristic

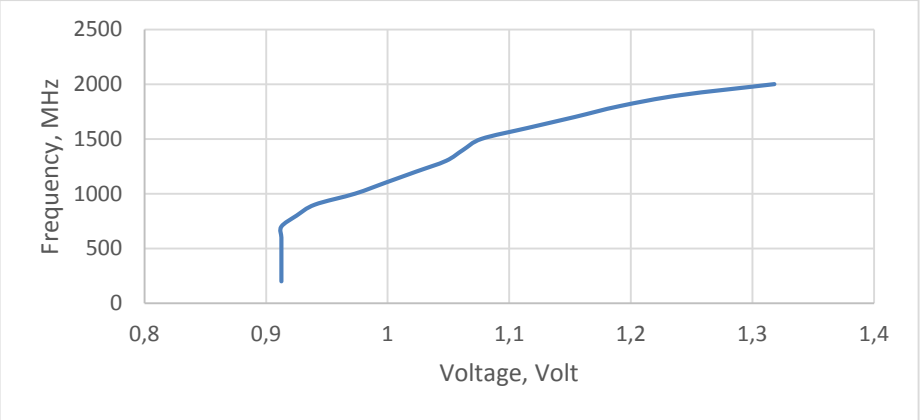

Fig. 3. Cortex-A15 voltage-frequency characteristic

cpufreq Linux governor provides the utility *cpufrec-set*, which was used to change the frequency of all four cores in the domains of either A7 or A15. For example, *cpufrec-set –u 1200MHz –c 7* sets maximum frequency 1200MHz for CPU core 7. Since it is possible to change the frequency only for all cores of the processor at the same time, all four cores of A15 (4, 5, 6 and 7) will get the same frequency 1200MHz.

In the second part of the experiment, the same parameters were measured for each core with 100% loading. The workload was created by a custom stress test program, which has been written in C language. The program executes 50 million square-root operations. Without artificial delays in the code (like *usleep* function), this program creates 100% workload for a CPU core.

Thread affinity was applied in order to execute the program on the specified CPU core. To bind the task to the CPU core *taskset* Linux command was used.

taskset SqrtStress –c 0

SqrtStress program will be executed on CPU core 0 (the first core of Cortex-A7 processor).

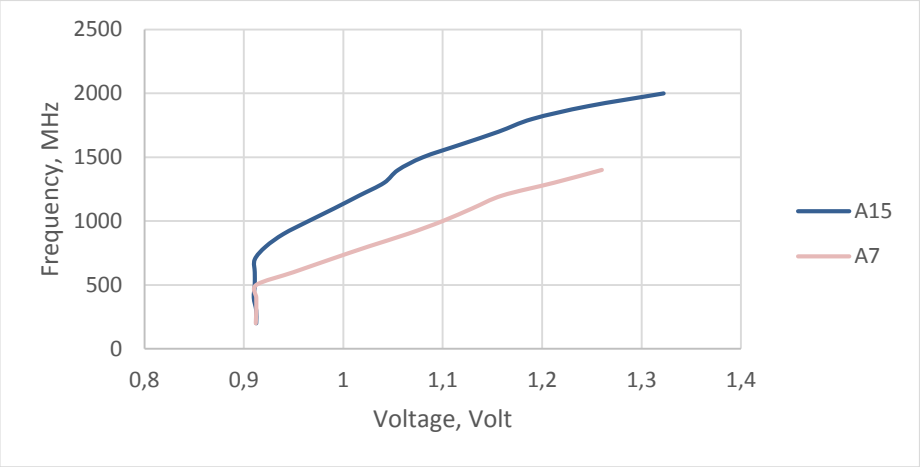

Fig. 4. Cortex-A7 and Cortex-A15 voltage-frequency characteristic under 100% workload

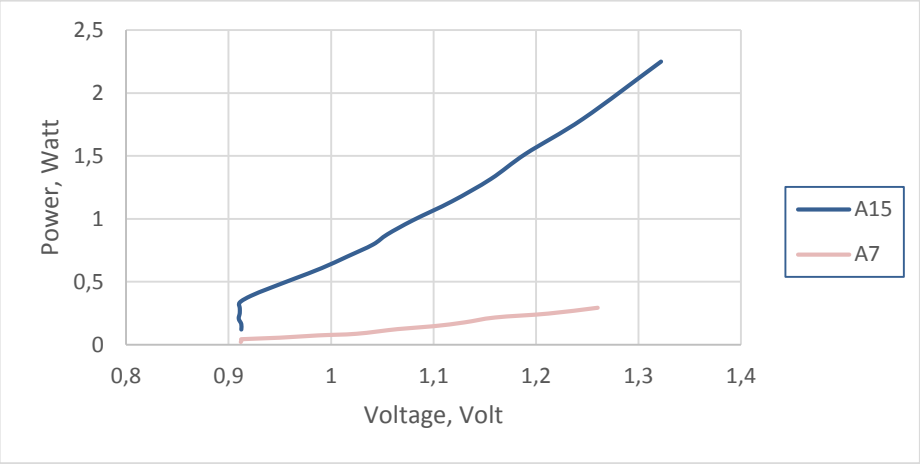

Fig. 5. Cortex-A7 and Cortex-A15 voltage-power characteristic under 100% workload

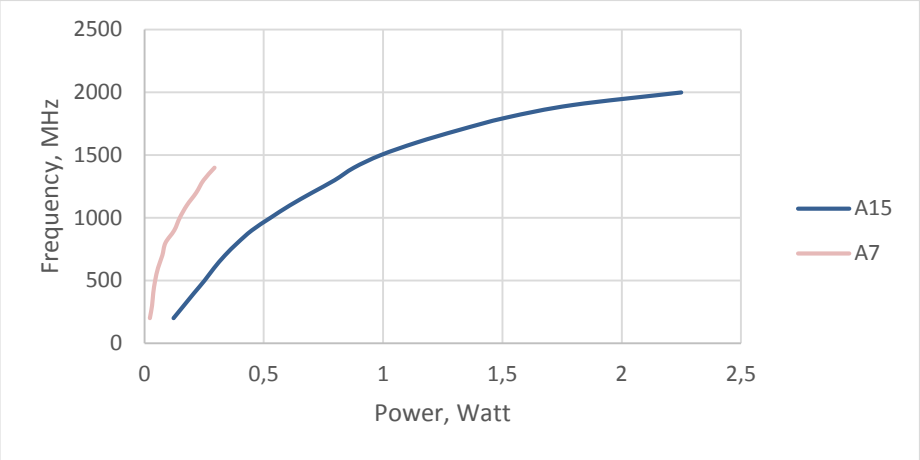

Fig. 6. Cortex-A7 and Cortex-A15 power-frequency characteristic under 100% workload

Experiments with the execution time of logarithm, addition, subtraction, multiplication and division operations gave the anticipated result. A15 was more than twice faster than A7 at the same frequency and almost three times faster at the maximum frequency. Unexpected results were received during experiments with the execution time of square-root operation. At the maximum frequency (2.0GHz) Cortex-A15 was just 1.2 times more productive than Cortex-A7 at the maximum frequency (1.4GHz), 10.9 seconds and 13.2 seconds for 50 million operations correspondingly. However, when the execution time was calculated at the same frequencies, A7 was faster than A15, for example at 1.0GHz frequency A15 core finished the task at 21.9 seconds, whereas A7 core required only 18.5 seconds and consumes a quarter of the power. The same trend was observed with sine and cosine functions.

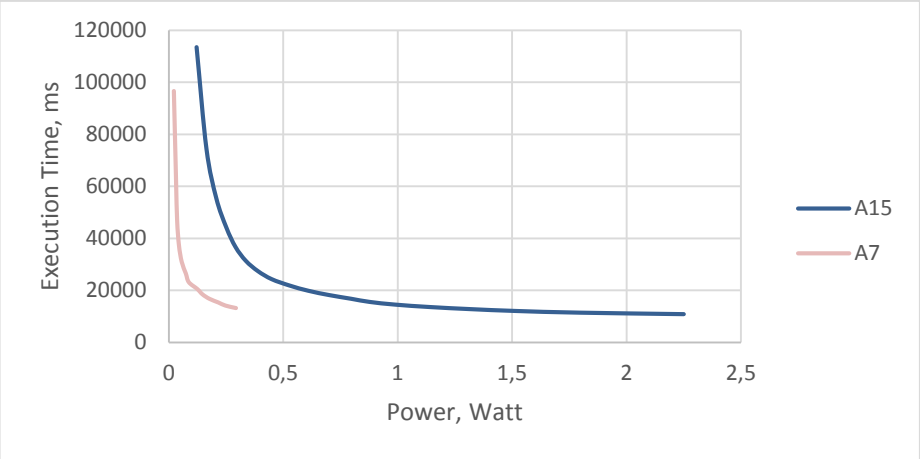

Fig. 7. Cortex-A7 and Cortex-A15 power-execution time characteristic under 100% workload

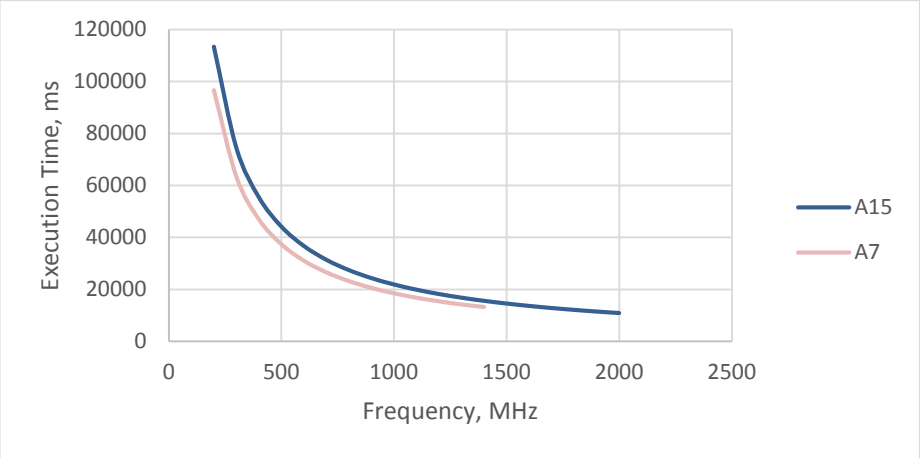

Fig. 8. Cortex-A7 and Cortex-A15 frequency-execution time characteristic under 100% workload

Diagrams on Figure 4, Figure 5 and Figure 6 represent common trends for A7 and A15 processors independently of the calculation task. Diagrams on Figure 7 and Figure 8 depend on type of the operation. The line on diagram, which relates to the processor with better performance (for the predefined operation) will be situated below. In case of square-rooting calculations, A15 shows worse performance than A7 running at the same frequency, that's why A15 line is above than A7 line.

Data related to the second part of the experiment can be found in Table 3 and Table 4.

3.2 Duty cycling with idle-wait state

Table 5 and Table 6 show the experimental results for different CPU loadings. These results represent the power consumption and execution time of 50 million square root operations (Figure 9 and Figure 10). The *usleep* function (C language) was used after every 100000 operations to put the thread into sleep state. We can create necessary CPU loading from about 0% to 100% by passing different arguments in the *usleep* function. As seen from the results duty cycling with idle-wait state is highly energy inefficient. The energy consumption required for calculations remains roughly the same, whereas total energy consumption increases when CPU loading is decreased. It is more efficient to execute the task as fast as possible than using this method of duty cycling.

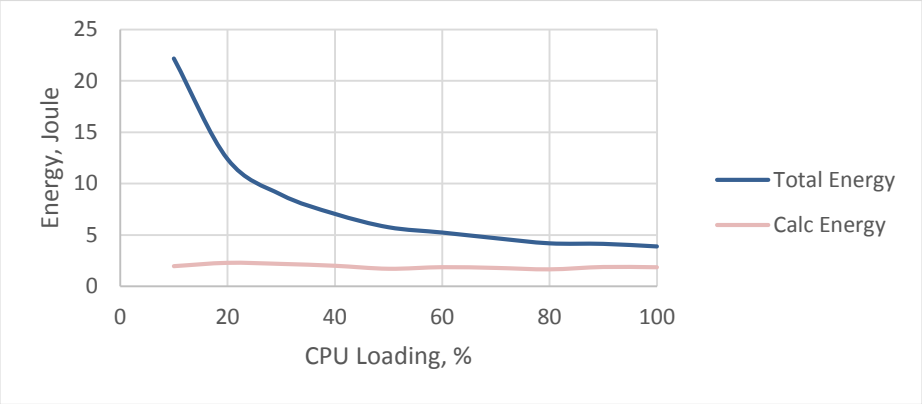

Fig. 9. Dependence of total energy and calculation energy on A7 CPU loading

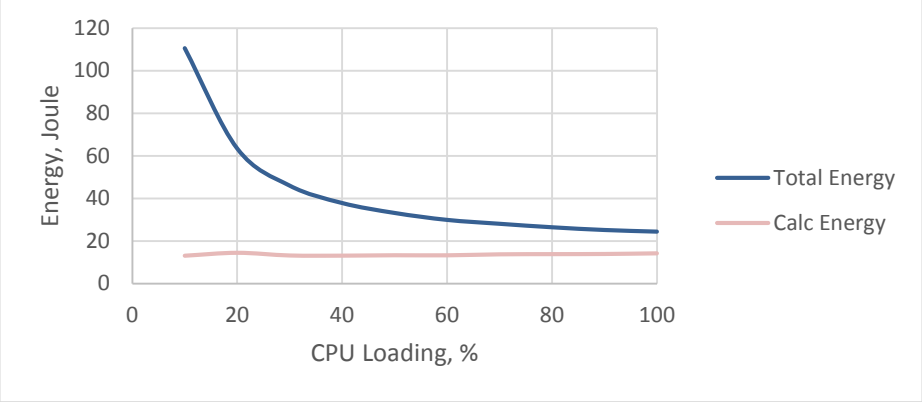

Fig. 10. Dependence of total energy and calculation energy on A15 CPU loading

3.3 Controlling the number of active cores

This experiment measures the same parameters while some of the cores in each block are disabled. It was carried out in order to investigate possible power and energy savings when the workload is not very high. Up to four A15 cores and up to three A7 cores can be disabled on Odroid-XU3. At least one A7 core must be running for the OS to be alive.

The following Linux command is used to disable a core:

echo 0 | sudo tee /sys/devices/system/cpu/cpu1/online

and to re-enable it again:

echo 1 | sudo tee /sys/devices/system/cpu/cpu1/online

The results show that disabling of one, two or three cores does not give big benefits for power saving (Figure 11 and Figure 12 or Table 7 and Table 8). For example, when there is no additional workload and only OS is running and all four A15 cores are enabled, A15 domain consumes 0.921W of power at 2000MHz frequency. If one A15 core

is disabled, A15 domain consumes 0.888W. If three A15 cores are disabled, A15 domain consumes 0.833W. However, when all four cores of A15 are disabled, the power consumption plummeted to 0.119W. Moreover, it is possible to reduce this value significantly by decreasing the A15 frequency from 2000MHz to 200MHz before disabling all four cores. In this case, A15 domain will consume only 0.021W. It is possible to use this technique to save power and energy.

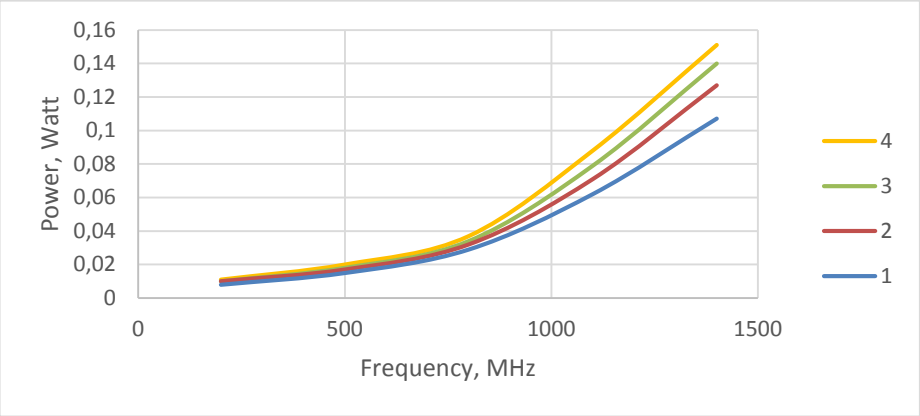

Fig. 11. Power consumption of A7 domain with different number of active cores

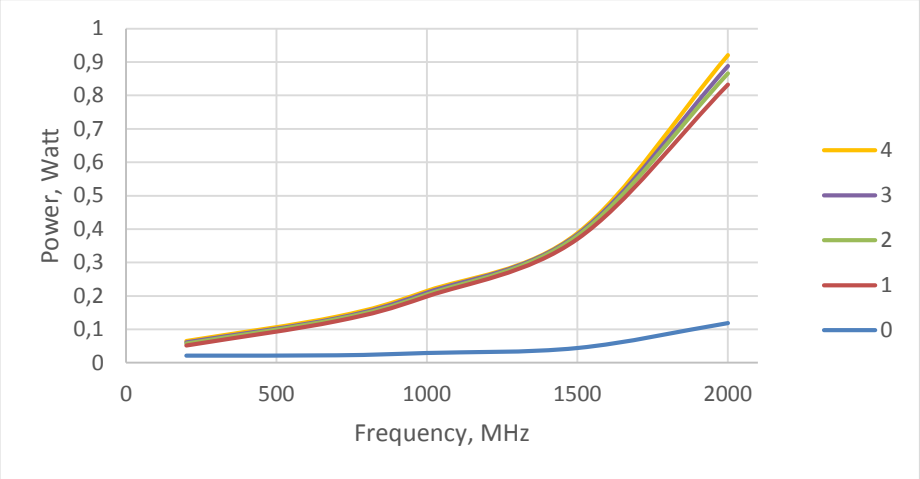

Fig. 12. Power consumption of A15 domain with different number of active cores

4 Conclusion

Several experiments were carried out in order to find out power, frequency and performance interplays on Odroid-XU3 board. Dynamic-frequency scaling is a very useful technique that can be applied for the adjustment to the system loading. The idle-wait state is very inefficient and it should be avoided whenever possible and not used to duty cycle operations in order to save power. Core disabling provides the possibility for substantial power and energy savings when the loading is low. These experiments give a deeper understanding of the benefits of heterogeneous architectures. The trade-offs between performance and energy-consumption obtained during the experiments are very useful for the run-time modelling in order to achieve optimal system operation.

5 Acknowledgments

This report is supported by the EPSRC/UK PRiME project. Ali Aalsaud is supported by the scholarship from Computer and Software Engineering Department, Al-Mustansiriya University.

6 References

1. Shekhar Borkar, “Thousand Core Chips—A Technology Perspective,” in *Proceedings of the 44th annual Design Automation Conference (DAC)*, 2007.
2. Hardkernel, “ODROID-XU3,” [Online]. Available: http://www.hardkernel.com/main/products/prdt_info.php?g_code=G140448267127.
3. Rafiev A., Xia F., Iliasov A., Gensh R., Aalsaud A., Romanovsky A., Yakovlev A., “Power-proportional modelling fidelity,” 2015.
4. ARM, “big.LITTLE Technology,” [Online]. Available: http://www.arm.com/products/processors/technologies/biglittleprocessing.php.

A Appendix

Table 1. Cortex-A7 frequency scaling

Frequency, MHz	Voltage, Volt	Current, Ampere	Power, Watt
200	0,913	0,018	0,0164
300	0,913	0,025	0,0228
400	0,913	0,028	0,0245
500	0,913	0,03	0,029
600	0,9512	0,034	0,035
700	0,988	0,045	0,039
800	1,026	0,049	0,048
900	1,063	0,069	0,0722
1000	1,101	0,074	0,081
1100	1,138	0,088	0,11
1200	1,176	0,109	0,132
1300	1,22	0,122	0,149
1400	1,273	0,143	0,185

Table 2. Cortex-A15 frequency scaling

Frequency, MHz	Voltage, Volt	Current, Ampere	Power, Watt
200	0,9125	0,0775	0,0639
300	0,9125	0,085	0,0775
400	0,9125	0,12	0,1039
500	0,9125	0,168	0,144
600	0,9125	0,191	0,174
700	0,9125	0,212	0,197
800	0,925	0,171	0,156
900	0,94	0,195	0,184
1000	0,973	0,223	0,215
1100	0,998	0,248	0,248
1200	1,023	0,279	0,285
1300	1,048	0,315	0,3225
1400	1,062	0,3359	0,357
1500	1,077	0,367	0,396
1600	1,115	0,414	0,46
1700	1,15375	0,459	0,529
1800	1,191	0,509	0,605
1900	1,241	0,592	0,7414
2000	1,318	0,683	0,9129

Table 3. Cortex-A7 frequency scaling with CPU loading

Frequency, MHz	Voltage, Volt	Current, Ampere	Power, Watt	Exec. time, ms
200	0,912	0,025	0,0228	96617
300	0,9125	0,035	0,032	63341
400	0,9125	0,042	0,037	46920
500	0,9125	0,049	0,045	37331
600	0,95	0,062	0,057	30990
700	0,987	0,079	0,075	26509
800	1,025	0,087	0,0881	23152
900	1,065	0,113	0,125	20537
1000	1,1	0,136	0,148	18464
1100	1,13	0,157	0,178	16768
1200	1,16	0,177	0,217	15359
1300	1,212	0,209	0,248	14166
1400	1,26	0,234	0,294	13205

Table 4. Cortex-A15 frequency scaling with CPU loading

Frequency, MHz	Voltage, Volt	Current, Ampere	Power, Watt	Exec. time, ms
200	0,9125	0,133	0,122	113443
300	0,9125	0,182	0,165	74452
400	0,91	0,229	0,208	55457
500	0,911	0,277	0,252	44169
600	0,911	0,322	0,292	36709
700	0,911	0,37	0,337	31393
800	0,922	0,422	0,39	27429
900	0,94	0,481	0,45	24351
1000	0,965	0,546	0,528	21906
1100	0,991	0,617	0,61	19896
1200	1,016	0,689	0,702	18226
1300	1,041	0,766	0,797	16818
1400	1,055	0,832	0,878	15610
1500	1,08	0,915	0,99	14570
1600	1,118	1,022	1,14	13649
1700	1,156	1,14	1,318	12843
1800	1,19	1,274	1,518	12130
1900	1,246	1,444	1,796	11483
2000	1,322	1,72	2,25	10912

Table 5. Duty cycle with idle-wait state for Cortex-A7

CPU Loading, %	Power, Watt	Time, seconds	Total Energy, Joule	Calc. Energy, Joule
100	0,295	13,205	3,895475	1,8487
90	0,283	14,655	4,147365	1,87584
80	0,255	16,463	4,198065	1,6463
70	0,25	18,781	4,69525	1,784195
60	0,24	21,864	5,24736	1,85844
50	0,22	26,194	5,76268	1,70261
40	0,216	32,705	7,06428	1,995005
30	0,205	43,545	8,926725	2,17725
20	0,19	65,231	12,39389	2,283085
10	0,17	130,44	22,17446	1,95657

Power that is consumed by A7 domain without additional CPU loading is 0.155 Watt.

Table 6. Duty cycle with idle-wait state for Cortex-A15

CPU Loading, %	Power, Watt	Time, seconds	Total Energy, Joule	Calc. Energy, Joule
100	2,23	10,95	24,4185	14,28975
90	2,08	12,11	25,1888	13,98705
80	1,95	13,573	26,46735	13,912325
70	1,82	15,444	28,10808	13,82238
60	1,67	17,925	29,93475	13,354125
50	1,55	21,415	33,19325	13,384375
40	1,42	26,66	37,8572	13,1967
30	1,3	35,406	46,0278	13,27725
20	1,2	52,885	63,462	14,543375
10	1,05	105,32	110,5902	13,1655

Power that is consumed by A15 domain without additional CPU loading 0.925 Watt.

Table 7. Control the number of active cores for Cortex-A7

Frequency, MHz	Voltage, Volt	Current, Ampere	Power, Watt	Number of active cores
200	0.9	0.009	0.008	1
		0.012	0.010	2
		0.013	0.010	3
		0.014	0.011	4
500	0.9	0.017	0.015	1
		0.019	0.017	2
		0.020	0.018	3
		0.022	0.020	4
800	1.0	0.028	0.029	1
		0.032	0.032	2
		0.033	0.034	3
		0.036	0.037	4
1100	1.1	0.055	0.062	1
		0.062	0.071	2
		0.070	0.079	3
		0.079	0.088	4
1400	1.2	0.086	0.107	1
		0.100	0.127	2
		0.110	0.140	3
		0.121	0.151	4

Table 8. Control the number of active cores for Cortex-A15

Frequency, MHz	Voltage, Volt	Current, Ampere	Power, Watt	Number of active cores
200	0.9	0.024	0.021	0
		0.058	0.052	1
		0.063	0.057	2
		0.067	0.061	3
		0.072	0.065	4
700	0.9	0.025	0.022	0
		0.138	0.125	1
		0.143	0.130	2
		0.147	0.133	3
		0.152	0.138	4
1000	0.9	0.03	0.029	0
		0.205	0.199	1
		0.210	0.204	2
		0.215	0.210	3
		0.221	0.215	4
1500	1.0	0.04	0.044	0
		0.342	0.371	1
		0.350	0.380	2
		0.356	0.384	3
		0.358	0.388	4
2000	1.3	0.09	0.119	0
		0.634	0.833	1
		0.658	0.866	2
		0.676	0.888	3
		0.702	0.921	4