Microprocessor Optimization for the IoT
(May 2017)

Adolfo Segura

Slippery Rock University

***Abstract—* It is no secret that the Internet of Things is on track to be one of the most popular technologies ever created. The idea of having an interconnected world with every device being able to communicate with each other is now very possible. From everyday life to the business world, the Internet of Things is taking over by bringing everything closer together. According to Business insider, by 2020 the Internet of Things devices that are connected to the Internet will have more than tripled going from 10 billion to 34 billion costing around 6 trillion dollars. This enormous market opportunity has influenced a lot of research emphasizing the software layer and how it interacts with devices; however, research that should also be considered are their microprocessors, edge nodes hardware components and processing capabilities.**

Right away the Internet of Things is challenged by many obstacles such as: microprocessors keeping up with the fast growing computing speed and memory requirements, maintaining connectivity, and all while maintaining a low cost, low energy budget. All these obstacles are driving more and more researchers to focus on optimizing the architecture of microprocessors. The heart of the Internet of Things is made up of many types of low-powered sensors (edge nodes) that receive and send the data to high-powered servers (head nodes) that take care of all the high-powered computation. But with the Internet of Things is only getting bigger by the second. As we look closer at the IoT architecture another challenge arises and it has to do with bandwidth and latency; however, can be diminished by edge computing. Rather than sending, processing and interpreting data in cloud or a remote data center, all the computations are performed at edge nodes. Edge computing minimizes how much data is actually sent ergo, improving bandwidth, latency, and the amount of energy that is consumed. But it is not as easy as it sounds, the sensors (edge nodes) computing power has to be able to execute and maintain the required computations, while meeting the servers (head node) constraints. To make sure that the architecture of a Internet of Things microprocessor is capable to meet all these requirements, it is important to compare the characteristics of application and microarchitecture to determine design tradeoffs. But with the amount of Internet of Things applications, as well as the amount of different types of architectures, finding the right architecture is not easy.

In this paper, I will look into many types of architectures for edge computing based on the executing applications and the characteristics of applications execution. I will also discuss the characteristics of micro-architecture of different types of low-powered processors based around the results of other researchers using the GEM5 architecture simulations. Lastly, I will evaluate where these processors stand in edge computing regarding the Internet of Things and discuss what tradeoffs should be expected in micro-architectural optimizations that will enable the design of a good Internet of Things microprocessor architecture.

I. INTRODUCTION

IoT technology is directly affecting many aspects of the world. It offers computing potential for many application domains, such as, healthcare, smart technology and etc. However, there is currently very little research that characterizes these applications with respect to their execution characteristics.

To better understand the execution characteristics of certain Internet of Things devices, a classification of 5 categories was made to help with the classification of these devices. The classification includes sensing, image processing, communication, compression, security and fault tolerance. Many may argue that a larger variety of classifications would better classify the IoT devices, which may be true; however, the five categories were enough for a minor study.

If your paper is intended for a conference, please contact your conference editor concerning acceptable word processor formats for your particular conference.

A. Sensing

The first category that was looked into, and one of the most important, is sensing. Sensing has to do mainly with the gathering and processing of different kinds of data (EEG, EMG etc.). This study not only focuses on the fact that IoT devices are able to collect real time data, but also that more and more IoT devices are able to analyze real time data. By doing so, the devices reduce how much data is actually sent. This then leads into a very interesting topic that can be a paper of its own, Sensor fusion algorithms. Sensor fusion algorithms are definitely a topic to include in future research.

B. *Communications*

The second category discussed in this paper is Communications. One of the most common characteristics of any IoT device is its internal structure and how its data moves from point A to point B by traversing many connected nodes. In other words, most IoT devices are most likely using Bluetooth/Wi-Fi, combined with TCP/IP; however, we highlight a new upcoming called Software Defined Radio (SDR) [Lee 2005]. SDR is a flexible upcoming communication system where components such as, filters and modems, that were normally implemented in hardware are being implemented in the software. SDR is a very important topic that can be a paper of its own. For simplicity I don't go into further detail about SDR. SDR is great topic to include for further research.

C. *Image Processing*

The third and pretty straightforward category is image processing. Image processing is much like sensing; however, instead of small data signals it involves extracting and converting real time image/video. With the fast growing world of IoT, things like face detection, automatic license plate recognition etc. is becoming more and more common.

D. *Compression*

With the increase in data and bandwidth-limited systems, compression can reduce communication requirements to ensure that data is quickly retrieved, transmitted, and/or analyzed. Additionally, since most IoT devices are resource-constrained, compression also reduces storage requirements when storage on the edge node is required. There are two types of classification techniques when dealing with compression, they are known as lossy and lossless. Jpeg is a perfect example of lossy compression. This technique reduces the storage size needed and like described, mainly used for multimedia files. On the other hand, Lossless compression does not degrade the data. Therefore, is mainly used for text and data files.

E. *Security*

Last but definitely not least, security is also a category worth focusing on. With the rapid increase of IoT devices being connected to the Internet, the likelihood of a cyber attack also increases. That's why it is very important that many of the new upcoming IoT devices have a good security system. Data encryption is a very popular technique to help protect anything valuable from getting stolen. It uses an encryption algorithm to transform the original data into encrypted data that can only be decrypted by a key. This plays an important role in edge computing because encrypting/decrypting the data requires computational and memory access.

F. *Fault tolerance*

Fault tolerance refers to a system's ability to operate properly in the event of a failure of some of the system's components. Fault tolerant applications are especially Microprocessor Optimizations for the Internet of Things 7 vital since IoT devices may be deployed in harsh and unattended environments, where QoS must be maintained in potentially adverse conditions, such as cryogenic to extremely high temperatures, shock, vibration, etc. In some emerging IoT devices, such as implantable medical devices, fault tolerance could be the single most critical requirement, since faults can be potentially fatal.

II. IoT Microprocessor Configurations

There are a few microprocessor characteristics that this study focuses on. These characteristics include number of cores, cache, off-chip memory support, power consumption, number of pipeline stages and a few more that are touched briefly but will be included in future research.

With all the information gathered, I shall discuss the results and the set of high level microarchitecture configurations for IoT edge computing that was developed by well-known researchers. We see how there are still big technological advances to be made to implement future Internet of Things microprocessors. This study specifically focuses on central processing units and intends to evaluate other kinds of microprocessors for future studies.

In this study, four different types of IoT microprocessors, all with different configurations, were chosen to evaluate. These configurations can be found in table 1 below.

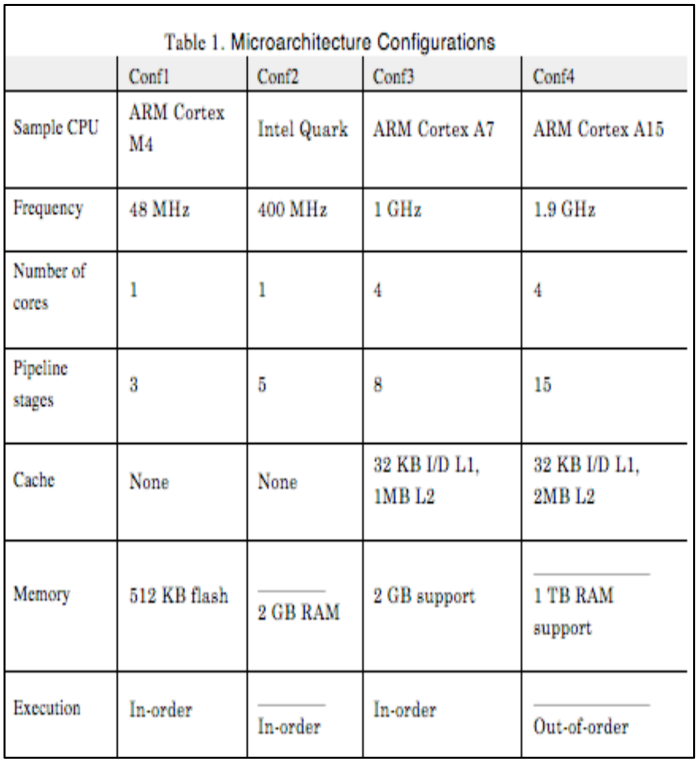


Table 1. Microarchitecture Configurations

	Conf1	Conf2	Conf3	Conf4
Sample CPU	ARM Cortex M4	Intel Quark	ARM Cortex A7	ARM Cortex A15
Frequency	48 MHz	400 MHz	1 GHz	1.9 GHz
Number of cores	1	1	4	4
Pipeline stages	3	5	8	15
Cache	None	None	32 KB I/D L1, 1MB L2	32 KB I/D L1, 2MB L2
Memory	512 KB flash	2 GB RAM	2 GB support	1 TB RAM support
Execution	In-order	In-order	In-order	Out-of-order

III. METHODS

I will talk about the classification created by other researchers that consists of 7 kernels that closely model to structure of mew upcoming Internet of Things edge computing devices. These 7 kernels help us establish a solid foundation of the functions of modern IoT devices. If you take a look at Table 2 depicts our application functions, each application function's representative benchmarks, and the benchmarks' descriptions

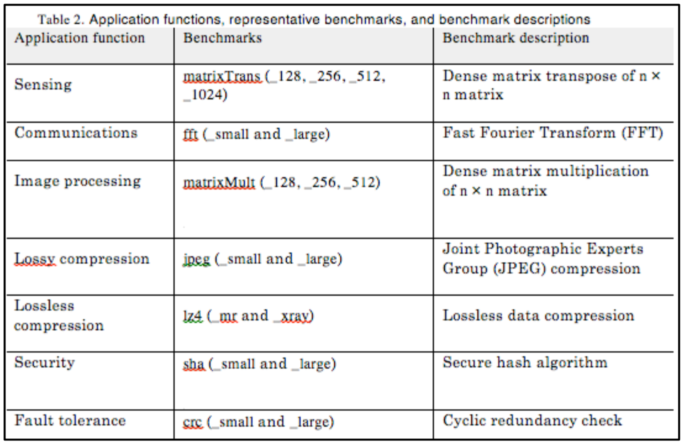
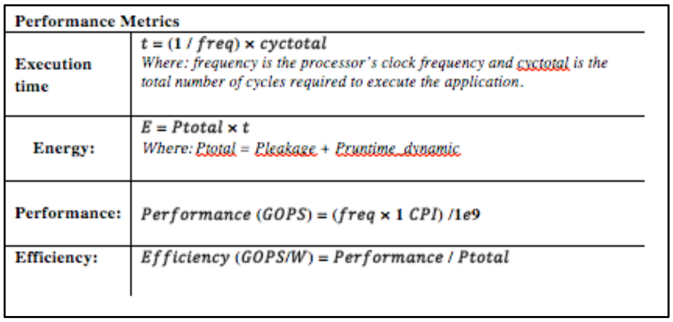


Table 2. Application functions, representative benchmarks, and benchmark descriptions

Application function	Benchmarks	Benchmark description
Sensing	matrixTrans (_128, _256, _512, _1024)	Dense matrix transpose of n × n matrix
Communications	fft (_small and _large)	Fast Fourier Transform (FFT)
Image processing	matrixMult (_128, _256, _512)	Dense matrix multiplication of n × n matrix
Lossy compression	jpeg (_small and _large)	Joint Photographic Experts Group (JPEG) compression
Lossless compression	lz4 (_mr and _xray)	Lossless data compression
Security	sha (_small and _large)	Secure hash algorithm
Fault tolerance	crc (_small and _large)	Cyclic redundancy check



Performance Metrics	
Execution time	$t = (1 / freq) \times cyctotal$ *Where: frequency is the processor's clock frequency and cyctotal is the total number of cycles required to execute the application.*
Energy:	$E = Ptotal \times t$ *Where: Ptotal = Pleakage + Pruntime_dynamic*
Performance:	$Performance\ (GOPS) = (freq \times 1\ CPI)\ /1e9$
Efficiency:	$Efficiency\ (GOPS/W) = Performance\ /\ Ptotal$

Execution time: The execution time, t is the time required to execute an application from start to finish.

Energy: The energy, E is the product of the power consumed by the processor, Ptotal and the application's execution time.

Performance: We define the performance in terms of the number of giga (billion) operations per second (GOPS).

Efficiency: We define the efficiency in terms of the performance per watt (GOPS/W), i.e., the attainable performance while taking into account the power consumed.

A. *Simulators*

Two simulators were used to test the IoT device configurations: the GEM5 and the McPAT. GEM5 simulator was used to generate execution statistics while running several benchmarks on the configurations as shown in Table 1 [Binkert 2011]. The McPAT simulator [Li 2009] was used to generate leakage, dynamic power, and area values for the different configurations, and used Perl scripts to drive our simulations.

IV. RESULTS

Simulation results for execution time, energy, performance, and performance per watt on the microarchitecture configurations listed in Table 1. We also perform sensitivity analysis with respect to varying application data sizes, various micro architectural characteristics, and evaluate the impacts of idle energy and leakage power reduction.

A. *Execution Characteristics and sensitivity to data sizes*

To evaluate the execution characteristics of the different benchmarks, we used the percentage of memory references per instruction (MPI) and the instructions per cycle (IPC) to provide insight into the benchmarks' memory and compute intensities, respectively. Figure 1 shows the MPI and IPC for all of the configurations with different input data sizes. Figure 1 (a) we can see that the memory intensity for the different configurations did not change with different data sizes

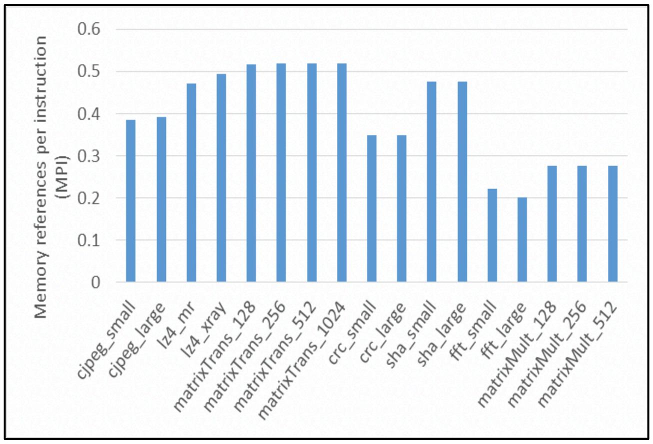


Figure 1 (a) Memory references per instruction (MPI)

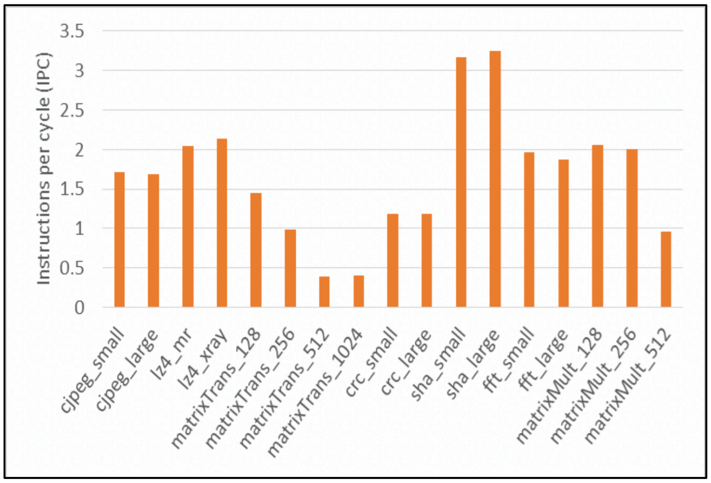


Figure 1. (b) Instructions per cycle (IPC) of the benchmarks with different data sizes

B. *Execution time, energy, performance and efficiency*

Figure 1(a) shows that conf1 increased by 202 times, conf2 by 23 times, and conf3 by over 9 times. In conclusion, when we focus on latency, the fourth configuration excels.

Taking a look at Figure 1 (b), we can see that when energy consumption is taken into consideration conf1 rose by 35 times its original amount, conf2 rose by over 4.6 times, and conf3 rose over 4.7 times is original energy amount. We notice that the smaller configurations lead to the time it took to execute to extend a great amount. In the experiment, the first configuration modeled the Micro-controller unit currently in the Internet of Things device. This means that the Micro-controller units are not able to fulfill all the edge computing needs.

Switching over to Figure 5 (a) we focus on the overall efficiency of the 4 configurations in our experiment. The greater the configurations, the better the improvements that are visible. Configuration four turns out to be the best when trying to minimalize the latency.

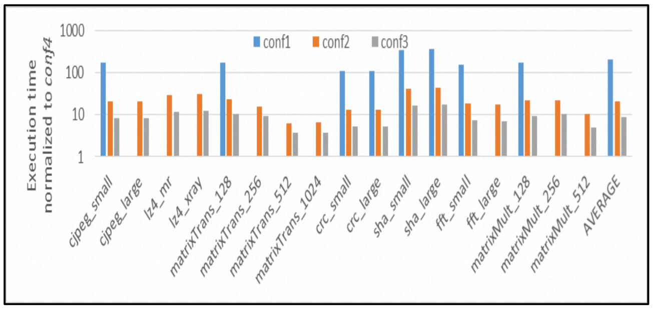


Figure 1. (a) Execution Time

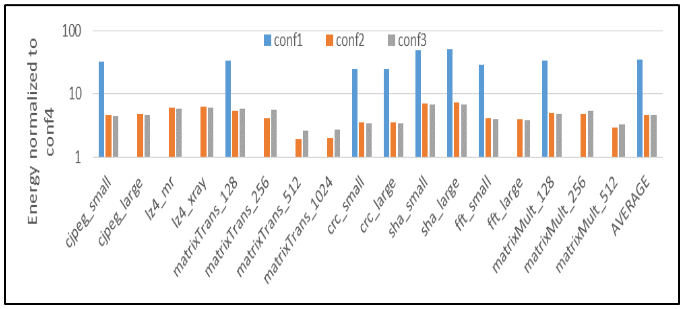


Figure 1. (b) Energy

C. Sensitivity to various Microarchitecture characteristics

To identify the most impactful microarchitecture characteristics on system execution time, energy, performance, and efficiency, we evaluated conf4 with a 1 GHz clock frequency, in-order execution, and a 16 KB cache size.

Taking a look at the results from Figure 6 (a) we can see how much effect the clock frequency has on the time it takes to execute and how much energy is consumed. The median of all configurations when the frequency of the clock was amplified to 1 GHz shows a seventy-five percent increase in execution time and forty-one percent increase in energy consumption.

Taking a look at Figure 6 (b) we can see a decrease of forty percent and twenty-seven percent in the median efficiency when the clock frequency was decreased. Thus we can see how important the frequency is when talking about Internet of Things edge computing.

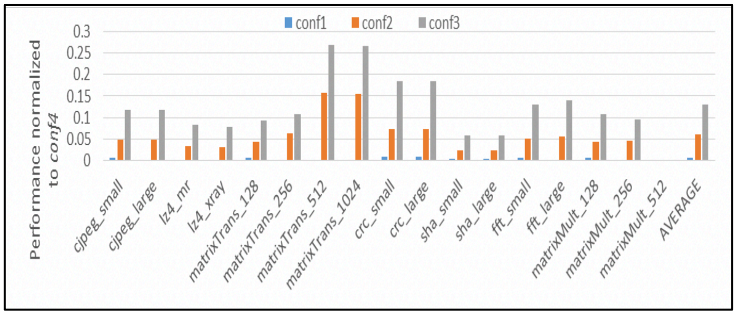


Figure 5. (a) Performance

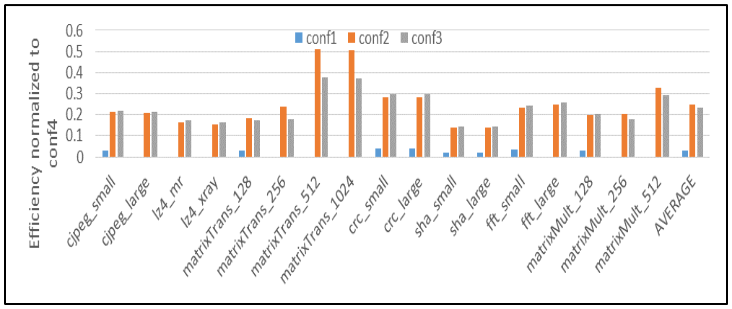


Figure 5. (b) Efficiency

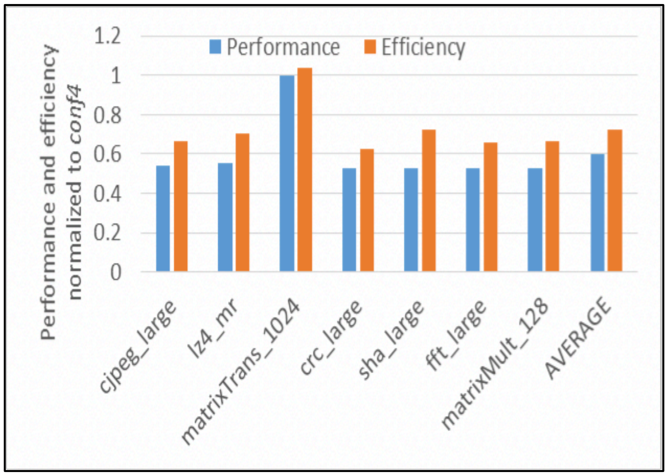


Figure 6. (b) Performance & Efficiency

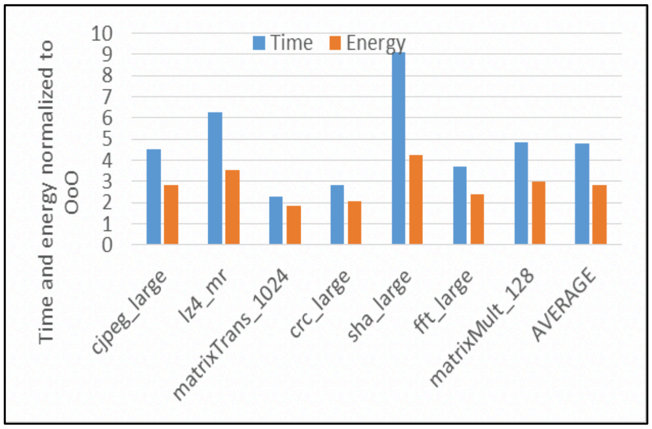


Figure 7. (a) Execution time and Energy

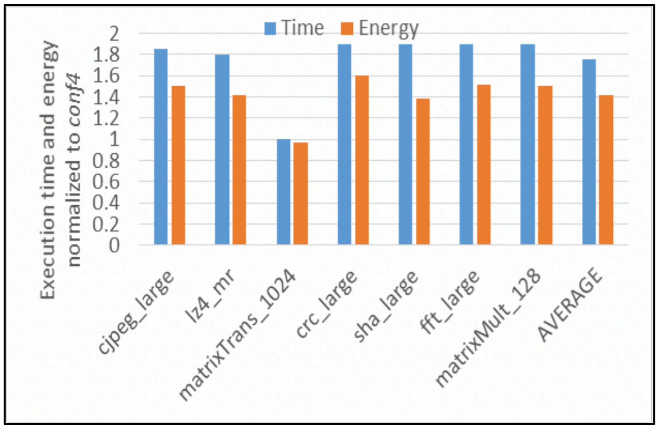


Figure 6. (a) Execution time and Energy

In-order execution is also something to consider as it caused the median execution time to increase by almost five times and the energy by almost three times. This shows us that out of order execution tends to outperform In-order. However, taking a look at Figure 7 (b), we can see a decrease in median performance time by seventy-five percent.

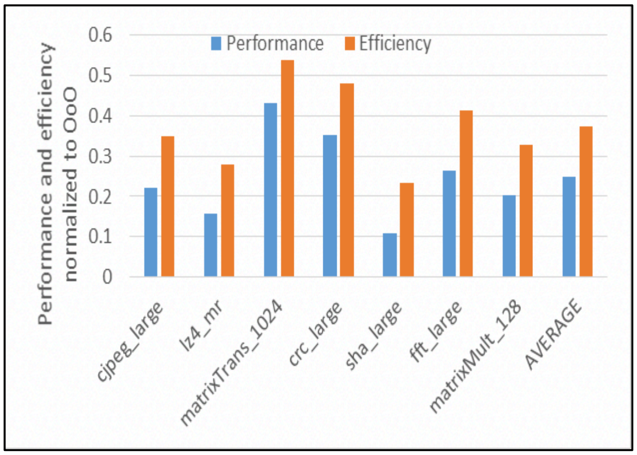


Figure 7. (b) Performance and efficiency of in-order

On the other hand, we tried reducing the cache to see the effects on energy, performance and efficiency; however, did not see a big difference in the overall results. (See figure 8)

D. *Execution time, energy, performance and efficiency*

As we gather all the results from our study a few optimization techniques for upcoming IoT microprocessors standout. The main mission from the start of this study was to get the maximum energy for execution without completely monopolizing the system and negatively affecting all other goals. We soon figure out what to expect when different microprocessor optimizations come into play and how to achieve the best optimization for every design.

V. Discussion

A. *Dynamic Voltage and Frequency Scaling (DVFS)*

Dynamic voltage and frequency scaling (DVFS) [Firouzi 2010; Schmitz 2002; Shin 2000] can be used to optimize using a variable voltage and/or frequency levels during a device execution. We know that when we talk about dynamic power we are basically dealing with the volts/frequency. The total power that is consumed decreases if frequency is decreased since that also reduces the required volts to function normally. "The major challenges of DVFS are the potential overheads from incorporating DVFS in a microprocessor and the performance degradation from reducing the operating frequency/voltage in order to save power/energy"
[Adegbija, Patel, Ross]. Since trying to implement Dynamic voltage and frequency scaling brings in certain overheads like power from voltage regulators; the ideal characteristics of Internet of Things microprocessor must be set in order to reduce the overheads.

B. *Configurable and Adaptable Caches*

"Since emerging IoT applications will increase in memory and compute intensity, IoT microprocessors must be equipped with more advanced memory hierarchies to take advantage of the spatial and temporal locality of the IoT applications. Due to the memory hierarchy's large impact on system performance and energy consumption, much emphasis must be placed on efficient caching techniques for IoT microprocessors."

Throughout this study I came to realize that many over-provisioned caches use a large amount of energy and do not provide any performance benefits. I have also come to realize that by simply reducing the cache size, the amount of energy that is used will greatly decrease. This means that a perfect optimization for microprocessors is to simply change the cache size at run time; however, three major challenges arise when trying to do this. The three challenges are "augmenting caches for configurability, cache tuning algorithms/heuristics, and cache tuners." In order to maximize the benefits of configurable caches, the required hardware optimizations to enable configurability must accrue minimal overhead.

This eventually leads us into a very important topic that can be a research paper of its own: cache tuning. A lot of previous work (e.g., [Adegbija 2014; Gordon-Ross 2005; Zhang 2003]) has been done in this topic and many algorithms/heuristics for cache tuning to minimize the potential of overhead and optimize the cache tuning benefits have been proposed. I wont go into this topic and will save it for future research.

The only thing that is worth mentioning is the cache tuning eventually leads to power, area, and/or performance overheads [Adegbija 2014].

C. Heterogeneous Architectures

It is important to briefly touch on Heterogeneous Architectures; however, this to is a separate topic that deserves its own research study. "One of the huge advantages of heterogeneous architectures for IoT microprocessors, from a design perspective, is that existing cores (e.g., CPUs, DSPs, GPUs, etc.) can be reused in the implementation of heterogeneous microprocessors, and this allows previous design and verification efforts to be amortized." A few quick downsides that come to mind when we talk about heterogeneous architectures for the IoT are the number/choice of cores, and scheduling certain applications to the appropriate cores. This would require a great amount of time spent on trying to figure out the best core/configuration to implement into the IoT microprocessor.

VI. Conclusion

Throughout this paper I discussed in fine detail many characteristics of Microprocessor Optimization that helps the growth of the Internet of Things (IoT) industry. From everyday life to the business world, the Internet of Things is taking over by bringing everything closer together. With the Internet of Things only getting bigger by the second, one thing is clear, the devices that are connected to the Internet will more than triple by 2020 and that all these devices will produce endless amount of data causing bandwidth, latency and energy challenges. Edge computing; however, provides certain devices with specific microprocessors that minimize how much data is actually sent by processing, interpreting and performing computations at edge nodes.

With all these new Internet of Things applications it is important to establish a base that supports the design of a base that supports the design of Internet of Things microprocessors. The comprehensive all-inclusive microprocessor design approach that I discussed takes into account the basic functionality of the device, as well as different microarchitectures that fulfill the functionality requirements.

References

Adegbija, T., Gordon-Ross, A., and Munir, A. 2014. Phase distance mapping: a phase-based cache tuning methodology for embedded systems. Design Automation for Embedded Systems. Vol. 18, Issue 3, pp. 251-278.

Adegbija, T., Gordon-Ross, A., and Rawlins, M. 2014. Analysis of cache tuner architectural layouts for multicore embedded systems. IEEE International Performance Computing and Communications Conference.

Atzori, L., Iera, A., and Morabito, G. 2010 The internet of things: a survey. International Journal of Computer and Telecommunications Networking.

Bash, C., Patel, C., and Sharma, R. 2006. Dynamic thermal management of air cooled data centers. Thermal and Thermomechanical Phenomena in Electronics Systems.

Binkert, M., et al. 2011. The gem5 simulator. Computer Architecure News.

Bonomi, F., Milito, R., Zhu, J., and Addepalli, S. 2012. Fog computing and its role in the internet of things. Workshop on Mobile Cloud Computing.

Chen, M., Gonzalez, S., Vasilakos, A., Cao, H., and Leung, V. 2011. Body area networks: a survey. Mobile networks and applications.

Choi, K., Soma, R., and Pedram, M. 2004. Dynamic voltage and frequency scaling based on workload decomposition. International Symposium on Low Power Electronics and Design.

Giusto, D., Iera, A., Morabito, G., Atzori, L. 2010. The internet of things. Tyrrhenian Workshop on Digital Communications.

Gordon-Ross, A., Vahid, F., and Dutt, N. 2005. Fast configurable-cache tuning with a unified second level cache. International Symposium on Low Power Electronics and Design.

Gubbi, J., Buyya, R., Marusic. S., and Palaniswami, M. 2013. Internet of things (IoT): A vision, architectural elements, and future directions. Future Generation Computer Systems 29, pp. 1645-1660.

Hu. Z., Buyuktosunoglu. A., Srinivasan, V., Zyuban. V., Jacobson, H., and Bose, P. 2004. Microarchitectural techniques for power gating of execution units. International Symposium on Low Power Electronics and Design (ISLPED).

Hua, S. and Qu, G. 2003. Approaching the maximum energy saving on embedded systems with multiple voltages. International Conference on Computer-aided Design.

Kwok, T. and Kwok, T. 2006. Computation and energy efficient image processing in wireless sensor networks based on reconfigurable computing. Proc. Of the International Conference on Parallel Processing Workshops (ICPPW), Columbus, Ohio.

Lee, H. et al.. 2005. Software defined radio – a high performance embedded challenge. High Performance Embedded Architectures and Compilers, Springer.

Lehr, W. and McKnight, L. 2003. Wireless internet access: 3G vs. WiFi. Elsevier Telecommunications Policy, vol. 27, issues 5 – 6, pp. 351 – 370.

Lovely, T. et al.. 2014. A framework to analyze processor architectures for next-generation on-board space computing. IEEE Aerospace Conference.

Luo, J., Tang, K., Chen, Y., and Luo, J.. 2015. Remote monitoring information system and its applications based on the internet of things. International Conference on Future BioMedical Information Engineering, 2009.

McKinsey Global Institute. 2015. Disruptive technologies: advances that will transform life, business, and the global economy. www.mckinsey.com

Miorandi, D., Sicari, S., Pellegrini, F., and Chlamtac, I. 2012 Internet of things: vision, application and research challenges. Ad Hoc Networks, Vol 10, Issue 7.

Nakamura, E., Loureiro, A., and Frery, A. 2007. Information fusion for wireless sensor networks: methods, models, and classifications. ACM Computing Surveys, Vol. 39, Issue 3.

Patel, C., Bash, C., Sharma, R., Beitelmal, M., and Friedrich, R. 2003. Smart cooling of data centers. International Electronic Packaging Technical Conference and Exhibition.

Sanchez-Sinencio, E. 2014. Smart nodes of Internet of Things (IoT): a hardware perspective view and implementation. ACM Great Lakes Symposium on VLSI (GLVLSI).

Schmitz, M., Al-Hashimi, B., and Eles, P. 2002. Energy-efficient mapping and scheduling for DVS enabled distributed embedded systems. Design, Automation and Test in Europe Conference.

Shin, Y., Choi, K., and Sakurai, T. 2000. Power optimization of real-time embedded systems on variable speed processors. International Conference on Computer Aided Design.

Tosiron, Adegbija, Anita Rogacs, and Ann Gordon-ross. Microprocessor Optimizations for the Internet of Things (n.d.): n. pag. Web. 2 May 2017.

Zhou, L. and Chao, N. 2011. Multimedia traffic security architecture for the internet of things. IEEE Network, Vol 25, Issue 3.

Zhang, C., Vahid, F., and Najjar, W. 2003. A highly configurable cache architecture for embedded systems. International Symposium on Computer Architecture.