Yue Zequn (A0129884M)

Liu Zijian ()

A Survey on Energy Efficient Multicore and Multiprocessor Systems in IOT: Architecture, Thread Scheduling and Communication

EE5902 CA Report

*Abstract –* In this project, we explored six papers which aim to achieve energy efficiency in multicore and multiprocessor systems in the field of IoT applications. In **“A Hierarchical Reconfigurable Micro-coded Multi-core Processor for IoT Applications**[1]”, the author proposed a simplified logic and shallow pipelined reconfigurable multi-core architecture which utilize long microinstructions for better energy efficiency. In “**ECAP: Energy Efficient CAching for Prefetch Blocks in Tiled Chip MultiProcessors**[2]”, the author explore the technique of using nearby chip free cache set as virtual cache and Confidence-Aware Replacement policy (CARP) to avoid extra energy consumption for unnecessary memory fetching. In “**Energy-Efficient Hardware-Accelerated Synchronization for Shared-L1-Memory Multiprocessor Clusters**[3]”, the author introduced a light-weight hardware-supported synchronization solution to reduce the synchronization overhead in terms of cycles and energy.

# INTRODUCTION

Over the last few decades, the progressive semiconductor manufacturing industry is slowly bringing us to the edge of the technological barrier in terms of silicon-based transistor size, the drastic computational improvements in compute power are gradually slowing down and the technical solutions started to branch into parallelism of processors/cores to improve/maintain the processing power trajectory. We are seeing more and more multicore and multiprocessor systems ranging from high-end processors to simple device controllers. With more cores, multiple processors and increasing die size, the power consumption is exponentially increasing as well. For small devices, especially in mobile and compact IoT applications which run on battery, there lies the strong need for an energy-efficient solution with appropriate computational power.

There are several different ways to achieve energy efficiency in a multicore or multiprocessor system. The direct way is to utilize near-threshold computing (NTC) by simply scaling down the supply voltage of part of the system to the optimal energy point (OEP) while maintaining an appropriate processing capability through parallel processing and proper thread scheduling and task mapping across multi-units in the system. More complicated ways of achieving energy efficiency include special software reconfigurable architectures, unique cache fetching strategies and communication techniques among cores or processors. In the field of IoT, there can be special ways of energy saving like synchronization schemes between sensors, master and slave distributed computing approaches. All the above mentioned perspectives will be thoroughly discusses in this survey on the sic chosen papers.

# PATHS TO ENERGY EFFICIENCY

In this section, the technique used to achieve energy efficiency will be discussed briefly.

In “A Hierarchical Reconfigurable Micro-coded Multi-core Processor for IoT Applications[1]”, the author mainly use 3 ways to save energy. Firstly, the overall control logic is simplified and deep pipelining is avoided. In order to still keep the same functionality, a more complex instruction set, namely long microinstructions is implemented. Here software complexity is used to bring simplicity to hardware, thus reducing intrinsic energy consumption for all operations. The author also explore ways to combine C and Java Instruction Set Architectures (ISAs) for better programmability. Secondly, with a multicore system, each core can be reconfigured to different working modes in order to achieve the best energy efficiency. A core can be reconfigured to function as a accelerator like an application specific processor and take specific task from the general cores. Thirdly, the system has integrated router implemented to facilitate short-ranged off-chip network for multi-processor systems. The most efficient data routing algorithm can be constructed with the integrated router. Data routing and transfer account for a very large portion of the total energy consumption for IoT application chips.

In “ECAP: Energy Efficient CAching for Prefetch Blocks in Tiled Chip MultiProcessors[2]”, the author specifically dive into a better data prefetch technique in tiled chip multiprocessors. Each processor in the package has a private L1 cache and a shared L2 cache. Through utilizing the L1 cache of nearby less used processors which are running less data intensive applications as virtual L1 cache, we can avoid data prefetched onto critical data blocks which generates more L1 cache miss and increase the frequency of L2 cache accesses. This essentially save large amount of energy in terms of data movement over high-bandwidth buses.

In “Energy-Efficient Hardware-Accelerated Synchronization for Shared-L1-Memory Multiprocessor Clusters[3]”, the author explore similar domain like the previous article: it is on the basis of a shared cache processing element (PE) cluster. The author focus on establishing a hardware-accelerated synchronization and communication unit (SCU) to facilitate synchronization between all the PEs and perform power management for idling PEs at a lower granularity level.

# DETAIL IMPLEMENTATIONS

In the field of architecture restructuring, “A Hierarchical Reconfigurable Micro-coded Multi-core Processor for IoT Applications[1]” propose a reconfigurable multicore system which can adapt to the specific applications at run time. Assuming a system with dual identical core, each core consist of arithmetic logic unit (ALU), multiplier, shifter, accumulator etc. which support all normal operation functionalities. The two cores share an on-chip memory where all the program are stored. There is a separated shared reconfigurable memory where all long micro codes are stored and used by the cores to decode the long micro coded instructions. Some of the instructions can be specifically created according to need to execute more complex and application specific workflows with simplified data and logic path. The long micro instruction has a more complex functionality than conventional simplified instruction sets like Reduced Instruction Set Computer (RISC). Unlike RISC instructions which are general in purpose and very short in nature, the proposed long microinstruction has multiple functionalities, including: controlling all function elements like ALU and multiplication and accumulation unit (MAC); I/O and memory access; control logic to steer data path inside the functional units. For conventional RISC instructions, normally each instruction only handle one operation which means all functional units beside the one that is active at this clock cycle are all idle, this intrinsically increase the total program execution time thus increase the static power drawn. One might argue that this could be improved by introduce deep pipelining stage to increase the utilization of all the functional units at each cycle. However, deep pipelining will come with additional hardware overhead and extra stages of data steering, this will increase the overall die size and additional dynamic power consumption which defects the purpose of energy efficiency in IoT applications. Owing to the use of micro coding, very complex operations can be done in one instruction[1]. This bring 2 main perks: with more predictable operational combination inside one long instruction, less branching is needed and all parallelism is fully utilized amongst all the functional units; it also reduce the frequency of memory access and data block prefetching will be less necessary as data loading will be done in parallel with the computations. As a result, the dynamic power for branching and energy heavy memory access in high-speed buses, and the static power of idle functional units is saved, achieving significant energy efficiency improvement.

Another notable benefit of having a core with reconfigurable instruction set is the flexibility of the functionality of a specific core can be altered in the whole system. All cores can be either general purpose processors or application specific accelerators based on the need for different IoT usage scenarios. This is not achievable by conventional architectures as normal RISC instruction set cores cannot run an application specific accelerator flow with acceptable speed and efficiency.

In the domain of efficient caching, author of “ECAP: Energy Efficient CAching for Prefetch Blocks in Tiled Chip MultiProcessors[2]” described a new idea to place the prefetched data block inside a Tiled Chip Multiprocessor (TCMP) with an underlying Network on Chip (NoC). Prefetching is a commonly used technique to avoid logic units idling while the needed data is being accessed through the slow bus to

# IMPLEMENTATION METHOD COMPARISONS

## Architecture

## Network communication

## Memory caching, distribution, and access

## Task mapping

## Security and reliability

# PERFORMANCE QUANTIFICATIONS AND RESULTS COMPARISONS

# UNIQUENESS IN THE IMPLEMENTATIONS

[1] N. Ma, Z. Zou, Z. Lu, L. Zheng, and S. Blixt, “A hierarchical reconfigurable micro-coded multi-core processor for IoT applications,” in *2014 9th International Symposium on Reconfigurable and Communication-Centric Systems-on-Chip, ReCoSoC 2014*, 2014, doi: 10.1109/ReCoSoC.2014.6861360.

[2] D. Deb, J. Jose, and M. Palesi, “ECAP: Energy-efficient caching for prefetch blocks in tiled chip multiprocessors,” *IET Comput. Digit. Tech.*, vol. 13, no. 6, 2019, doi: 10.1049/iet-cdt.2019.0035.

[3] F. Glaser, G. Tagliavini, D. Rossi, G. Haugou, Q. Huang, and L. Benini, “Energy-Efficient Hardware-Accelerated Synchronization for Shared-L1-Memory Multiprocessor Clusters,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 3, 2021, doi: 10.1109/TPDS.2020.3028691.