International Journal on Recent and Innovation Trends in Computing and Communication
Volume: 3 Issue: 5

ISSN: 2321-8169
150 – 152

Design and FPGA Implementation of Resizable Cache Memory for Low Power Computing

Prof. Mayuri Chawla
Electronics and Telecommunication Dept.
Jhulelal Institute of Technology
Nagpur, India
chawlamayuri23@gmail.com

Dr. Sanjay Asutkar
Electronics Department
Manoharbai Institute of Engg. & Technology
Gondia, India
asutkarsanjay@yahoo.com

Dr. Vijay chourasia
Electronics and Communictaion Department
Manoharbai Institute of Engg. & Technology
Gondia, India
chourasiav@gmail.com

Abstract— This paper is proposed on resizable cache architecture that is able to configure itself to the running application. In our day to day life, we rely upon microprocessors and microcontrollers having effective caching technique the size of the cache greatly affect performance, or the time needed to execute programs. The optimal high-performance, low power cache will minimize energy consumption, or the product of power and execution time. This research shall design and evaluate a new caching technique that dynamically shuts down part of a processor's cache in order to reduce overall energy consumption.

Keywords- low power, cache, reconfiguration.

*****

I. INTRODUCTION

Our society relies more heavily on computers and microprocessors with each passing year. Building a microprocessor requires organizing a large number of transistors onto a complex integrated circuit (IC). Currently, the high transistor density on modern microprocessors forces computer architects to consider both power consumption and performance. Shutting down parts of the microprocessor serves as the easiest, most effective mechanism to conserve power. Many general-purpose processors utilize this technique [1, 2]. Because the caching structures on microprocessors use a large percentage (up to 80%) of the transistors, shutting down parts of the cache would save a considerable amount of power [3]. However, the size of the cache greatly affects performance, or the time needed to execute programs. The optimal high-performance, low power cache will minimize energy consumption, or the product of power and execution time. This research shall design and evaluate a new caching technique that dynamically shuts down part of a processor's cache in order to reduce overall energy consumption. We also target a cache controller that is dynamically reconfigurable according to the running application and adjusts itself to provide highest possible performance while using minimum cache.

For the past 37 years, Moore's law has accurately predicted that the number of transistors on a single IC wills double every 18 months [4]. Increased transistor density has increased operating speeds at the same rate, but also caused more power consumption [5, 6]. This increased power consumption generates undesired heat, which potentially degrades performance or destroys the IC. Historically, computer architects have designed processors either for high performance or for low power depending on the application. For example, a cell phone needs low power consumption so that it will not burn the user's hand; however, a gaming console needs maximum.

Advances in telecommunication and computer systems have pushed the development of new and complex embedded systems that range from modems and routers to smart-phones, tablets and net-books. For example, the smart-phone industry alone is expected to sell over 500 million units during 2012.

To take in account the rapid development of embedded processors new computing architectures are being developed. It should be noticed that, despite some convergence between general purpose processors (e.g., the Intel Core i7) and embedded processors (e.g., the ARM Cortex or the MIPS32 processor family), these two types of processors are intrinsically different. The former targets general purpose applications with fewer power restrictions. The latter is intended for specific applications often with strict power and budget restrictions.

One of the most important components of any efficient computational system is the memory hierarchy subsystem. However, since the main primary memory is typically placed far away from the processor and works at a different frequency, the access times for reading or writing data into the main memory are often very high (e.g., hundreds of processor cycles). To mitigate this fact, cache memories are typically added to the processor to store a subset of the information in the main memory. Since cache memories are placed close to the processor and can work at the processor frequency, the access times to data in the cache are short (typically less than 10 clock cycles). Thus, cache memories allow increasing the processor performance by tens or hundreds of times. This is especially important since the performance bottleneck of many data-intensive applications is actually accessing the data.

As mentioned earlier, cache memories greatly increase the processor performance but they are also responsible for a high portion of the processor's power consumption and chip area (which affects processor cost). Thus, while increasing cache sizes can lead to increased processor performance, it also leads to a more expensive and power hungry processor. To overcome this problem, reconfigurable cache architectures should be developed that are able to implement dynamic algorithms during program execution. These algorithms can also include the powering-down of portions of the cache when they are not being used. This would lead to significant improvements in processor performance while decreasing the total power consumption of the processor.

The shift from scaling frequency to scaling the number of cores continues the trend of stressing off-chip memory bandwidth and reliance on on-chip caches. However the costs of larger caches are significant and growing. They typically occupy 40–60% of the chip area and with leakage power exceeding switching power at sub-micron technologies, they are dominant consumers of energy. Furthermore, analysis of benchmarks have shown that cache utilization is typically low - below 20% for a majority of benchmarks, with performance efficiency averaging 4.7% and energy efficiency averaging 0.17%. This is in large part due to the fact that the majority of the cache (especially L2 and L3) is idle most of the time but contributes significantly to leakage power. The state of the practice in making caches more energy efficient has been to power down cache components such as cache lines, sets or ways - turn them off or maintain them in a low voltage state. Strategies focus on when to turn of which components. Poor decisions lead to expensive misses and power up events and therefore strategies tend to be conservative.

II. LITERATURE REVIEW

Karthik T. Sundararajan, Timothy M. Jones and Nigel Topham in their work [7] titled Smart Cache: A Self Adaptive Cache Architecture for Energy Efficiency have presented a Set and way Management cache Architecture for Run-Time reconfiguration, a cache architecture that allows reconfiguration in both its size and associativity. Their results show the energy-delay of the Smart cache is on average 14% better than state-of-the-art cache reconfiguration architectures

Mehdi Alipour, Mostafa E. Salehi, and Kamran Moshari in paper [8] titled Cache Power and Performance Tradeoffs for Embedded Applications have explored the design space of cache in embedded processors to find out the cache sizes which have the optimum performance/power consumption in embedded applications.

Jongsok Choi, Kevin Nam, Andrew Canis, Jason Anderson, Stephen Brown, and Tomasz Czajkowski in 2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines presented work on [9] Impact of Cache Architecture and Interface on Performance and Area of FPGA-Based Processor/Parallel-Accelerator Systems in which they have described new multi-ported cache designs suitable for use in FPGA-based processor/parallel-accelerator systems, and evaluate their impact on application performance and area.

TPUTCACHE: HIGH-FREQUENCY, MULTI-WAY CACHE FOR HIGH-THROUGHPUT FPGA APPLICATIONS published by Aaron Severance, Guy G.F. Lemieux introduces TputCache [10], a cache designed to meet the needs of throughput processing on FPGAs, giving the throughput performance of on-chip BRAMs when the problem size fits in local memory. The design utilizes a replay based architecture to achieve high frequency with very low resource overheads.

A.Bengueddach, B.Senouci, S.Niar and B.Beldjilali in [11] Energy Consumption in Reconfigurable MPSoC Architecture: Two-Level Caches Optimization Oriented Approach have investigated the estimation of the energy consumption in embedded MPSoC system. The main contribution of their research is to explore two level data cache (L1/L2) multiprocessor architecture by estimating the energy consumption.

Seungcheol Baek, Hyung Gyu Lee, Chrysostomos Nicopoulos, Junghee Lee, and Jongman Kim in paper [12] titled Size-Aware Cache Management for Compressed Cache Architectures introduces the concept of size-aware cache management as a way to maximize the performance of compressed caches and shows an average effective capacity increase of 18.4% over the Least-Recently Used (LRU) policy, and 23.9% over the Dynamic Re-Reference Interval Prediction scheme.

Kenji Kanazawa and Tsutomu Maruyama in [13] FPGA Acceleration of SAT/Max-SAT Solving using Variable-way Cache have proposed a method to hide the access delay by using on chip memory banks as a variable-way associative cache memory. This cache memory aims to hold whole block when it is small enough, and only the head portion when it is large, to hide the DRAM access delay. With this cache memory, up to 60% DRAM access delay can be hidden, and the performance can be improved up to 26%.

III. PROPOSED APPROACH

- We propose to implement the memory and other features using HDL coding.
- After successful implementation we will also detect the cache hit/miss ratio. The data from block shall also be useful in providing inputs to the reconfigurable block to manage the capacity and properties of cache.
- We can then develop a testing strategy by generating a behavioral model that shall apply all possible combinations of test vectors to the design and observe the outputs for verification of desired results.

Scope of this design is relevant to microcontrollers or microprocessors that usually run multiple small applications. As the reconfiguration is application depended, the performance of the processor is not aggressively affected by reducing the cache capacity. Cache accounts for majority static power dissipation, the unused cache may be totally switched off to reduce static power. Dynamic reconfiguration will allow the design to achieve high performance when application requires more cache and will cause less power dissipation when less cache is required.

IV. CONCLUSIONS

We will study the available low power design architecture, by doing literature survey .One comparative statement with their limitation .

We will also focus on the design of reconfigurable controller that configures the parameters of cache like changing associativity , number of lines, replacement strategy etc.

An algorithm will them be developed to detect the cache required for a particular application and decide the capacity as well as other parameters required for cache. We will design the memory and the controller in HDL, simulate the circuits using CAD tools like ModelSim, Testing and Implementation on suitable FPGA platform.

REFERENCES

[1] Vincent Heuring, and Harry Jordan. Computer Systems Design and Architecture. Massachusetts: Addison-Wesley, 1997.

[2] Michael Powell, Se-Hyun Yang, Babak Falsafi, Kaushik Roy and T. N. Vijaykumar. "Gated-Vdd: A Circuit Technique to Reduce Leakage in Deep-Submicron Cache Memories." Proceedings of the International Symposium on Low Power Electronics and Design (2000).

[3] Parthasarathy Ranganathan, Sarita Adve, Norman P. Jouppi. “Reconfigurable Caches and their Application to Media Processing”. Proceedings of 27th international symposium on computer architecture (ISCA-27), June 2000

[4] Intel Corporation. “Moore’s Law”. 13 February 2002. http://www.intel.com/research/silicon/mooreslaw.htm

[5] David Brooks, Vivek Tiwari, and Margaret Martonosi. “Wattch: A Framework for Architectural-Level Power Analysis and Optimizations.” Proceedings of the 27th International Symposium on Computer Architecture 2000.

[6] Anoop Iyer, and Diana Marculescu. “Run-time Scaling of Microarchitecture Resources in a Processor for Energy Savings.” Proceedings of KoolChips Workshop, International Symposium on Microarchitecture, Monterey, 2000.

[7] Karthik T. Sundararajan, Timothy M. Jones and Nigel Topham. “Smart Cache: A Self Adaptive Cache Architecture for Energy Efficiency. 2011 IEEE.

[8] Mehdi Alipour, Mostafa E. Salehi, and Kamran Moshari. “Cache Power and Performance Tradeoffs for Embedded Applications”. 2011 International Conference on Computer Applications and Industrial Electronics (ICCAIE 2011).

[9] Jongsok Choi, Kevin Nam, Andrew Canis, Jason Anderson, Stephen Brown, and Tomasz Czajkowski. “Impact of Cache Architecture and Interface on Performance and Area of FPGA-Based Processor/Parallel-Accelerator Systems”. 2012 IEEE 20th International Symposium on Field-Programmable Custom Computing Machines.

[10] Aaron Severance, Guy G.F. Lemieux. “Tputcache: High-Frequency, Multi-Way Cache For High-Throughput Fpga Applications”. 2013 IEEE.

[11] A.Bengueddach, B.Senouci, S.Niar and B.Beldjilali. “Energy Consumption in Reconfigurable MPSoC Architecture: Two-Level Caches Optimization Oriented Approach”. 2013 IEEE.

[12] Seungcheol Baek, Hyung Gyu Lee, Chrysostomos Nicopoulos, Junghee Lee, and Jongman Kim. “Size-Aware Cache Management for Compressed Cache Architectures”. 2014 IEEE Transactions on Computers.

[13] Kenji Kanazawa and Tsutomu Maruyama. “FPGA Acceleration of SAT/Max-SAT Solving using Variable-way Cache”. IEEE 2014.