Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 6 – No.3, November 2016 – www.caeaccess.org

A Survey of Emerging Architectural Techniques for Improving Cache Energy Consumption

Washington Bhebhe
Department of Computing and Immersive Technologies
University of Northampton, UK

Michael Opoku Agyeman
Department of Computing and Immersive Technologies
University of Northampton, UK

ABSTRACT
The search goes on for another ground breaking phenomenon to reduce the ever-increasing disparity between the CPU performance and storage. There are encouraging breakthroughs in enhancing CPU performance through fabrication technologies and changes in chip designs but not as much luck has been struck with regards to the computer storage resulting in material negative system performance. A lot of research effort has been put on finding techniques that can improve the energy efficiency of cache architectures. This work is a survey of energy saving techniques which are grouped on whether they save the dynamic energy, leakage energy or both. Needless to mention, the aim of this work is to compile a quick reference guide of energy saving techniques from 2013 to 2016 for engineers, researchers and students.

Keywords
Fetching; Power Gating; Immediate Sleep; Dynamic and Leakage.

1. INTRODUCTION
The recent advancements of multicore and multithreading technologies have seen the CPU processing power having a great growth [85] caused by the advancements in fabrication technologies, modern chip designs with more and smaller transistors embedded in it. The reduced transistor size offers a lesser dynamic energy for switching at the cost of higher static energy because of leakage while the performance improvement of storage systems has lagged. The negative knock-on effect is that the performance gap between the CPU and storage system has continued to widen year by year [42] making the storage system a bottleneck of the whole system performance. The growing speed disparity between the processor and memory has attracted a lot of research in the hope of closing this ever-growing gap.

Attempts have been made to alleviate the disparity between the CPU performance and storage by introducing high speed multilevel cache memories found in the mainstream Chip Multi-Core Processors (CMPs) [55] which use two level on-chip cache hierarchy i.e. the private L1 cache and the public L2 cache to improve the system performance. The L2 cache is shared and has a larger capacity, a high association to provide a fast access to resources [42]. The multi-level caches in CMPs are characterised by high switching power, particularly due to the large amount of power consumed in tag-comparison operations. The power consumption is also growing because of the ever-increasing usage of the L2 caches. Also, a significant power consumption in L2 caches is caused by the fact that the L2 cache needs a high associativity to reduce the conflict misses, making large energy to be spent on tag comparisons and because the cache coherence for CMPs increases the power consumption in tag comparison.

Elsewhere, the recent advancements in video streaming, image processing and high speed wireless communication have immensely affected the way the cache is being designed [50,68]. The technological advancements place demands for high performance and low energy consumption in embedded systems. There has been a massive shift in the designs of cache architectures, shifting the objective from achieving highest peak performance to achieving highest energy efficiency. Embedded systems present a challenge because of their energy and performance budgets.

The cache memory consumes significantly a large proportion of the processor energy, approximately 42% in swarm processor and 23% in Power PC [45]. It is not practically feasible to increase the cache size indefinitely. The increase in cache size can have a negative effect on the cache speed, hit rate, cache line size and the associativity for the applications and to apply the energy optimisation techniques to reduce the static and dynamic energy without material performance degradation. The energy consumption in caches has attracted a lot of research with the objective of coming up with cache architectures that can achieve energy efficiency in caches. The objective of this work is to review some of these research techniques.

The rest of this paper is organised as follows. Section II is the background reading on the cache architecture, components of cache and its different designs. Energy consumption is also discussed under this section. Section III details the leakage energy saving techniques while Section IV focusses on the dynamic energy saving techniques. Section V reviews energy saving techniques that attempt to save both dynamic and leakage energy. Section VI is the survey of recent commercial chips implementing the energy saving techniques. Section VII there is cache architectural simulation aimed at improving cache performance by increasing associativity. Section VIII is the simulation evaluation. The conclusion is on Section IX.

2. BACKGROUND READING
The background reading is focused on the cache architecture, with a view of understanding how the cache is designed, the components of cache and different types of its organisation. The last section of the background reading is dedicated to how energy is dissipated in the cache architecture and where within the cache different types of energy consumption of are dissipated. Cache is a small, high speed memory usually made up of Static RAM (SRAM) and it contains the most recently accessed pieces of main memory [37]. It is situated between the processor and the main memory Dynamic RAM (DRAM). The cache is 75% faster than the DRAM [37]. It takes 15ns to access information from the cache compared to 60ns from the DRAM. It also takes more time and energy to fetch an instruction than to execute it, hence to avoid performance bottleneck at the input of the processor, the cache needs to be fast. The memory design

strictly centres on the principle of locality reference, meaning that at any given time, the processor access memory a small or localised region of memory. This localised region is loaded by cache. Internal 16K byte cache of a Pentium processor contains over 90% of the addresses requested by the processor making a hit rate of 90% [26]. It is not feasible to replace the main memory with SRAM to achieve more performance because SRAM is very expensive, less dense and consumes more power than DRAM. Increasing the amount of SRAM has a negative effect on performance since the processor will have more area to search, resulting in more time and dynamic energy being spent on fetching. The cache needs to be of a size that the processor can quickly determine a hit or a miss to avoid performance degradation.

A. CACHE ARCHITECTURE

The cache architecture can be a read or write policy. A read architecture can either be a Look Aside or a Look through. The write policy architecture can be a write back or write through.

Cache subsystem can be divided into three functional blocks, which are SRAM, Tad RAM and the Cache Controller. The SRAM is the memory block and holds the data and its size determines the size of the cache. The Tag RAM is a small piece of SRAM which stores the addresses of the data that is stored in the SRAM. The Cache Controller (CC) on the other hand is the main brains begins the cache. The CC is responsible for performance snoops and snarfs, updating the SRAM and TRAM and for implementing the write policy. The CC also determines if memory request is cacheable and if the request is a miss or a hit.

Cache has different organisations, and these are briefly described below,

a) Fully-Associative (FA). This allows any line in the main memory to be stored at any location in the cache. It does not use cache page, only the lines. FA organisation provides the best performance since any memory location can be stored at any cache location. However, its main disadvantage is its complexity during fetching since the current address must be compared with all the addresses in the TRAM. This requires a very large number of comparators that eventually increase the complexity and cost of implementing large caches.

b) Direct Map (DM). Under the DM, the main memory is divided into cache pages. The size of each page is equal to the size of the cache. DM cache may only store a specific line of memory within the same line of cache. DM is the least complex and less expensive compared to the FA and the SA. Its disadvantage is that it is far less flexible, making the performance much lower especially when jumping between pages.

c) Set-Associative (SA). The SA cache scheme is a combination of FA and DM. under SA, the SRAM is divided into equal sections called cache ways, the cache page is equal to the size of the cache way which is treated like a small direct mapped cache or sleep leakage mode.

Sparsh Mittal has done a similar survey of architectural techniques on energy saving in cache in 2012 [64]. However, this work from his work because in this fast growing and evolving area, three years is a long time and many techniques have been proposed since then. This work focuses on the developments that have happened between 2013 and 2016.

3. LEAKAGE ENERGY SAVING APPROACHES

B. Overview

Leakage energy is mostly spent on LLCs and most techniques switch off part of the unused cache or introduce DRAM which is denser and does not suffer a lot of leakage energy consumption. The techniques estimate the required size of the cache needed by a program before it runs and turns off any unneeded cache space. The leakage energy saving techniques are either state destroying or state preserving techniques. State preserving techniques turn off the unneeded cache area and still preserve its state. When the cache space is re-activated, there is no need for fetching the lower levels of memory. On the other hand, the state destroying techniques do not preserve the state of the switched off cache space. If the destroyed cache space is later needed, it must be fetched in the lower levels of memory. The state destroying techniques generally save more energy than the state preserving techniques for as long as the destroyed cache space is not needed. When the destroyed cache space is later needed, more energy is spent in searching lower level caches.

Various techniques used to save leakage energy have some similarities and are discussed below,

Various researchers employ the drowsy cache technique (DCT) to save the leakage energy [35]. The DCT migrates a portion of unneeded cache into a low-leakage mode, thus saving energy at granularity level. Some reaches use the immediate sleep technique to turn off the unused cache [94]. Reconfigurable cache techniques are used to estimate program miss-rate in an online manner [66]. Some techniques use power-gating to migrate blocks deemed useful to a live partition before shutting down the unused cache [5]. Some techniques focus on hybridisation of PCRAM and DRAM to introduce a bit of density to the PCRAM and cause less leakage consumption [13].

C. Discussion

Mittal et al. [62] presents a multicore energy saving technique using dynamic cache reconfiguration. The technique works by periodically allocating required amount of LLC space to each running application and turning off unused cache space to save energy. This technique is called MASTER and it uses cache colouring scheme, allocating cache at the granularity of a single cache colour. A reconfigurable cache emulator is used for profiling the behavior of running programs under different LLC sizes. The energy-saving algorithm is used to predict the memory subsystem energy of running programs for a small number of colour values. Master uses these estimates to select a configuration with minimum estimated energy and turns off the unused cache colours for sassing leakage energy. Vdd gate is also used by Master to implement the hardware of the cache block. The simulation results show that the average savings in memory subsystem energy over shared baseline LLC are 15% and 11%.

Joonho et al. [108] proposed an energy-efficient PV-aware 3D LLC architecture. The technique exploits the narrow width values to save many faulty cache lines under severe process variation which results in significant yield improvement in a highly energy-efficient manner with only a small performance loss and area overhead. The zeros are stored in the cache arrays with the faulty cache portions. A significant leakage energy is saved which also contribute to the leakage -induced yield loss reduction.

Bardine et al. [7] evaluated leakage reduction alternatives for

deep submicron dynamic non-uniform cache architecture caches. The results of the simulation show that cache decay has leakage savings and performance degradation comparable with Way Adaptable on D-NUCA cache. The drowsy cache is potentially able to get higher energy reduction with reduced performance losses. Zhu et al. [79] used a Tripple-Threshold-Voltage 9-Transistor SRAM Cell technique for Data stability and energy efficiency at Ultra-low power supply voltages. The technique scales the power supply voltage (Vdd) to enhance the integrated circuits efficiency. However, this efficiency causes the substantial degradation of reliability due to less noise margins of the CMOS circuits.

Kadjo et al. [43] proposed a novel technique called Power Gating using block migration in chip multiprocessor last level caches. This technique greatly reduces the leakage energy of Lower Level Caches (LLC) while reducing the impact on performance levels. High temporal locality blocks are migrated to facilitate power gating. The blocks expected to be used in the future are migrated from the block being shut down to a live partition at a negligible performance impact and hardware overhead. Simulations show that energy savings of 66% can be reached at only 2.16% performance degradation.

Charkrabotty et al. [10] used a technique called Performance Constrained Static Energy reduction using way-sharing target banks. This technique improves the performance of the target banks by dynamically managing their associativity. The cost of the request is optimised by adding distance as another metric reducing the performance degradation. Experiments show that static energy can be reduced by 43% and EDP by 23% for a 4MB LLC with 3% performance constraint. The underutilised banks are powered off and the requests re-mapped to target banks.

Khartan et al. [45] proposed a Hardware based approach for saving cache energy in multicore simulation of power systems. In this technique, the time domain simulation of power system is conducted and traces of analysis instructions recorded. The recorded traces are used in multicore configurations. The Drowsy cache technique (DCT)is used as a cache leakage energy saving technique. DCT transitions a portion of cache into a low-leakage mode thus saving energy at cache granularity level. The results of simulations show an effective savings, keeping performance losses at minimal. For a 2MB 2-Core system, DCT saves 54% Cache energy and for a 4MB 4-core system up to 50.2% cache energy is saved.

Rossi et al. [73] introduced BTI and Leakage aware DSV for Reliable Low Cache memories. The technique shows that the Bias Temperature Instability (BTI) induced degradation greatly benefits leakage power saving of drowsy cache memories. The simulation results show that the leakage power can be reduced by more than 35% during the first month of use, more than 48% during the first year and up to 61% in ten years of memory operation. This shows that older memories give an opportunity of saving more leakage energy.

Arima et al. [5] proposed a technique called immediate sleep for reducing energy impact of peripheral circuits in STT-MRAM caches. This technique aims to save leakage power of peripheral circuits. Immediate sleep is also a power technique used for turning off a sub array of STT-MRAM caches immediately if the next access is not crucial or will not impede the performance levels. The technique uses power gating to STT-MRAM caches at the granularity of subarray at runtime. The subarrays contain local decoders and write drivers. With every cache access, all the decoders and write drivers are activated. Non-critical events are the write access and accesses which arrive after a long interval. A next-access predictor algorithm is used to predict accesses for each subarray at runtime and this prediction is made possible because there is a small number of subarrays in an STT-MRAM LLCs. Shutting down these non- critical events subarrays can save a relatively large leakage energy without affecting performance. The immediate sleep technique has proved to save leakage energy by 32% of an STT-MRAM LLC compared to the conventional scheme with STT-MRAM LLC.

Yue et al. [97] proposed a Micro-architectural technique for Run-time Power-gating in caches of GPU for leakage energy savings when they are idle during workload execution. The mode-transition latency is used to switch in and out of the low-leakage or sleep mode whenever needed to. These latencies are micro-architecturally hidden to avoid performance degradation during workload execution. The low-leakage mode is state-retentive meaning that it does not lose contents and there is no need for flushing the caches after they wake up. Therefor the L1 cache which is private to the core can be put into low-leakage sleep mode when there are no scheduled threads and if there are no memory requests, the L2 cache can be put into sleep mode. This technique on average can save up to 54% of leakage energy.

Mittal et al. [66] proposed a Flexi way, which is a cache energy saving technique using Fine-grained Cache Reconfiguration based approach for saving leakage energy. Flexi works on the observations that access to the cache sets are not distributed uniformly making some sets seeing more accesses than others. The other observation is that of the difference in associativity of sets. A cache is subdivided into small modules called subways. Selective-ways technique is used to turn off exactly the same number of ways for all the modules, providing the fine grain reconfiguration with caches of similar associativity thereby avoiding the need to use caches of large associativity for fine-grain reconfiguration. The simulation results show that Flexi can achieve energy savings of 26.2% in dual core systems.

Mittal et al. [63] discuss a technique called CASHIER or Cache Energy Saving Technique for Quality-of-Service systems (QOS). This technique uses a reconfigurable cache emulator which estimate the program miss-rate for various cache reconfigurations in an on-line manner. CPI stakes are used to estimate program execution time under different Lower Level Caches configurations. The Energy Saving Algorithm (ESA) then uses the estimates to estimate the memory sub-system energy under different cache configurations. From the results of ESA, a suitable cache is chosen that will strike the best balance between energy saving and performance loss, thereby avoiding a deadline. The of CASHIER saves on average 23.6% energy in memory sub-systems for a 2MB L2 cache with a 5% performance slack allowed.

Wang et al. [90] propose a technique called System-Wide Leakage-Aware Energy Minimisation using dynamic voltage scaling and cache reconfiguration in multi-tasking systems. Dynamic Voltage Scaling (DVS) is integrated with Dynamic Cache Reconfiguration (DCR) techniques. The DVS and DCR make decisions judiciously so that the total amount of energy consumed is minimised. Using only the DVS or DCR in isolation lead to wrong conclusions in the overall energy savings. This proposed technique is 47.6% more efficient than the leakage-aware DVS techniques and 23.5% than the leakage-oblivious DVS and DCR techniques. Sampaio et al. [59] proposed a technique called ‘Approximation-Aware Multi-Level Cells STT-RAM Cache Architecture’. The aim of the technique

is to achieve energy-efficient reliability optimisation in STT-RAM based caches through what they called Selective approximations of the storage data. The Selective data approximation simplifies the error-protection hardware depending on the resilience levels and user-provided error tolerance of the applications. The technique aims on maximising the quality of the applications while minimising the energy consumption.

Jing et al. [40] discuss the Energy-Efficient eDRAM-Based On-Chip storage architecture for General purpose graphics processing units (GPUPUs). The use of the eDRAM is proposed as an alternative for building an area and energy-efficient on-chip storage, including the RF, shared memory and L1 caches. eDRAM is chosen because it enables higher density and lower leakage power but suffers from limited data retention time. To avoid periodic refreshing of the eDRAM which makes performance suffer, lightweight compiler techniques are applied and runtime monitoring for selective refreshing that intelligently eliminate the unnecessary refreshes.

Hsiang-Yun et al. [70] discusses a technique called 'LAP', which is a Loo-Block which is aware of inclusion properties for energy-efficient asymmetric of last level caches. The technique is designed to improve the efficiency of Non-Volatile Memory based LLCs, especially the redesign of inclusion properties and associated replacement policies to explicitly include write reductions. The technique also incorporates advantages from both non-inclusive and exclusive designs to selectively cache only part of upper-level data in the LLC. The simulation results show that the architecture out-performs other variants of selective inclusion and consumes 20% and 12% less energy than non-inclusive and exclusive STT-RAM based LLCs.

Hameed et al. [34] proposed a two row buffer bypass policies and an alternative row buffer organisation to reduce the number of row buffer in STT-RAM based Last-Level cache architectures. The experiments show that energy consumption is reduced by 19.5% compared to SRAM alone.

Bengueddach et al [8] proposed a technique called MPSoC for energy consumption reconfiguration in two-level caches. The technique uses dynamic reconfiguration approach in the multiprocessors. This memory hierarchy contribute largely in the energy consumption of the overall hardware/software architecture.

Cheng et al [13] proposed an Adaptive page allocation of DRAM/PCRAM hybrid memory architecture. This is a hybrid of DRAM and PCRAM, taking advantage of both components, reducing leakage energy. The technique uses DRAM as the last level of cache and PCRAM as the main memory. When the data stored in the DRAM block is not accessed for a period, it is ignored and the refresh operation of the DRAM block is stopped. When the data accesses are found necessary after it has been ignored, the data is reloaded from the PCRAM. The technique uses a small DRAM as cache of PCRAM memory to reduce leakage power consumption together with an adaptive page allocation scheme which is used to make better utilisation of the DRAM capacity. This results in the conflict misses of DRAM being minimised under the multi-core architecture. The architecture reduces the number of write back to PCRAM and data migration between PCRAM and DRAM is materially reduced. The results of the simulation show that both the energy consumption and access latency of PCRAM is reduced by 25%.

Hammed et al. [34] proposed a technique for architecting STT LLCs for performance and energy improvement. The technique uses a large chip of DRAM memory as LLC. The experiments show that on-chip DRAM LLC provides significantly improved performance benefits compared to an SRAM based LLC with a high cache capacity. The technique employs STT-RAM as a larger LLC because of its low leakage, non-existent refreshing energy and its scalability benefits.

4. DYNAMIC ENERGY SAVING APPROACHES

D. Overview

The techniques for saving dynamic energy normally work in the FLCs because this is where most the dynamic energy is spent. In the FLCs, the data and instructions are separate concerns unlike in the LLCs where they are unified. The techniques proposed for saving the dynamic energy dissipation are not the same and are grouped below into how they seek to improve the consumption of dynamic energy.

Some techniques save dynamic energy by employing a bloom filter. The tag-bloom techniques use tags to directly map addresses in L1 and L2 caches. [59]. The bloom filter algorithm is then used to predict the cache misses and divert the search to the main memory rather than to waste time and energy in searching the cache. Some techniques use prefetching technique and cache locking to save dynamic energy. The prefetching and locking techniques save energy by reducing the penalty costs of cache misses. Some techniques aim to reduce the number of active ways accessed in each cache access to the number of ways halted in case of a miss prediction using software or hardware. Other techniques reconfigure cache using computer software [71] while some techniques predict the program behaviour [55,1,59]. Some techniques deal with instructions in the cache [16,35].

Some techniques seek to leverage unused cache block words to reduce dynamic energy consumption [10,15,36]. These techniques flush out the unused words from the cache and keep the cache current. Some techniques introduce a new level of cache called L0. L0 is placed between the processor and the L1 cache to improve the speed at which it can be accessed [49,46,22]

E. Discussion

Lee et al. [52] proposed a technique called Filter Data Cache which is an energy efficient small L0 Data Cache architecture driven by the miss-cost-reduction. The filter data cache is placed between the processor and the L1 cache. The filter data cache is small to improve the speed at which it can be accessed and thereby reducing the dynamic energy consumption. The filter data cache consists of Early Cache Hit Predictor (ECHP), a Locality based Allocation (LA) and a No Tag-Matching Write (NTW). The ECHP predicts if there will be a miss or a hit in the filter cache. If the prediction is a miss, then the filter cache. If the prediction is a miss, then the filter cache is by-passed and the L1 cache is accessed. The LA decides of whether to allocate data on the filter cache or not. The NTW reduces the energy consumption of write-backs by recording all the cache lines in the cache filter which requested from the L1 cache. The recorded cache lines are used in future in accessing L1 cache without requiring the tag matchings. The results of the simulations show that the technique can detect 92.2% of misses which are by-passed to the L1 cache. The filter data cache can reduce data allocations by 58% on average and only 5% of the number of hits decreases. 21% of energy consumption is reduced using the filter cache.

Divya et al. [42] proposed a cache architecture called Partial

Way-Tagged cache. This cache is used to reduce the energy consumption of write-through cache systems with minimal area overhead and less performance degradation. A way-tag is attached in the L2 cache and the same information sent and stored in the L1 cache when the data is loaded in the L1. During future accesses, any write hit in L1 cache is directly mapped to the L2 cache. This direct-mapping reduces significantly energy consumption in L2 caches. An enhanced bloom filter is also used to store partial values of tags and make cache miss predictions to avoid redundant L2 cache accesses and in the process reducing energy consumption.

Xian at al. [51] used a technique called the SSD-Base Cache architecture for primary storage in the virtualisation environment. The technique reduces the duplicate data blocks in the cache device resulting in expansion of the cache space. The experiment results show that the technique can greatly improve the l/O performance and prolong the SSD device life time. The cache hit ratio can be increased by 5 times, average l/O latency can be reduced by 63% while the SSD write can be eliminated by 81%.

Datta et al. [20] presented a technique called CPU Scheduling for Power/Energy Management on multicore processors using cache miss and context switch data. Two algorithms are used namely Cache Miss Priority CPU Scheduler and Context Switch Priority CPU Scheduler. These algorithms lower the global budget in multicore processor systems while improving the performance energy metrics. The algorithms use the hardware partitions containing CPU sets operating at the same frequency. The results of the simulations show that the algorithms created power savings of 38watts and at the same times generating 15.34% performance gain. The algorithms also reduce the power consumption by taking advantage of the much unnoticed dynamic performance data.

Kadjo et al. [43] proposed a technique called Leveraging Unused Cache Block words to Reduce power in CMP Interconnect. The technique aims to find the useful words in cache lines and flush out the unused words. Unused words are leveraged by implementing a word-predictor. The word predictor records the used words in the cache block when the cache block is about to be evicted. Miss prediction rates are lowered by using two of the recent history bit vectors. All the untouched words are considered unused. The findings show that the use of the used words-predictor has accuracy of 76.69% and dynamic energy reduction of 26.62% is achievable. The use of the combination of both flit-drop and word-repeat has a potential of saving 41.75% reduction in dynamic energy.

Grani et al. [33] proposed an architecture called Flat-Topology High-Throughput Compute Node with AWGR-Based Optical-Interconnects. The study was based on simulations comparing execution time and energy consumption of optical multi-socket boards. The results show that optical solutions perform better than the electronic baseline with 70% EDP being saved exploiting the DVFS techniques. With an architecture design with no intra-socket electronic hops, it was proved that an additional 15% in performance was possible. The use of voltage, frequency scaling optimisation and the source-synchronous transmission model obtained up to 3 times reduction in energy consumption.

Mohammandi et al. [69] proposed an On-Demand Dynamic Branch Prediction (ODBP)technique. This combines both the static and dynamic prediction schemes to allow low energy and highly accurate branch prediction. Instruction binaries are annotated with prediction hints that compile time enabling the processor to choose between two schemes. The hints determine whether an instruction is a branch or non-branch type. If it's a branch type, the hints determine if it is statically not-predictable, statically taken or dynamically predictable. The use of ODBP minimises branch miss-predictions thereby increasing performance and energy efficiency. Yuan et al. [35] proposes an instruction locking using Temporal Re-Use Profile (TRP)to improve the Worst-Case Execution Time (WCET) in real time embedded applications. TRP is used to compute the cost and benefit of cache locking. TRP has the advantage of being more compact compared to memory trace thereby enabling efficient cache locking algorithms to eliminate cache conflict misses through cache locking. The result of TRP is then used to lock a memory block that will minimise the number of cache misses and reduce the dynamic energy lost.

Faramahini et al. [25] use a technique called Near-Memory Caching for improved energy consumption. The SRAM is placed closer to memory rather than closer to the processor. A small chip cache is integrated within the boundaries of a power aware multi-aware multibanked memory. This organisation is called Power-Aware CDRAM (PACDRAM). The PACDRAM improves performance, drastically reduce cache accesses in the main memory. Cache energy is reduced because of the small caches that are distributed to the memory chip reduce the cache access energy compared to large and undistributed caches. Near-Memory caches allow the access of relatively large blocks from memory which is not affordable with near processor caches. Memory energy consumption is further reduced by having the DRAM banks turned off during long idle periods. Mian Lou et al. [52] proposed a novel technique with the neglectable auxiliary overhead to reduce the power occurred in L1 cache comparison during the backward invalidation. This technique is energy-efficient two-level cache architecture for chip multiprocessors. A banked Bloom filter is exploited for the realisation and organisation of the cache. The linear feedback shift register counter is also exploited to replace the traditional predictors. The results of the simulations show that the proposed architecture can reduce the cache power by 49.7% at the cost of the acceptable performance overhead.

Anneesh et al. [3] proposed a technique called 'Power and Performance Efficient Secondary Cache' using tag bloom architecture to reduce the power consumption. The tag-bloom cache improves the performance of write through cache systems along with reduction in power consumption and minimal area overhead. The tag-bloom technique uses tags to directly map the address in the L1 and L2 caches such that when the data is deleted in the L1 cache, the processor only need to check the L2 cache because the same data will also be stored in the L1 cache. Because of the direct mapping, a write hit in L1 cache directly maps to the corresponding data in L2 cache using the way tag information hence reducing significant power consumption. The technique also uses a bloom filter algorithm to predict the cache misses and divert the misses to the main memory.

Subha et al. [79] proposed a technique called 'A Reconfigurable Cache Architecture'. This architecture enables occupied ways of selected sets to be enabled on occupancy. The architecture introduces a sequential component to cache design for cache ways. The results of the experiment show that on average 6.7% of power is saved for L1 cache of 2048 sets and 4.7% of L1 cache of 4096 with associativity of [8,16,32].

Aahn et al. [1] discusses a technique called 'Prediction Hybrid Cache' which is an energy-efficient STT-RAM cache architecture. A mechanism called Write Intensity Predictor is

proposed which is used to predict the Write Intensity of every cache block dynamically. The predictor relies on the correlation between write intensity of blocks and addresses of load instructions which result in misses of the blocks. The predictor therefor keeps track of instruction likely to load write-intensive blocks and utilise the information to predict write intensity of blocks likely to be accessed in the future. The experiments conducted show that 28% energy reduction in hybrid caches is possible and 4% energy reduction in the main memory.

Cilku et al. [18] discusses an instruction cache architecture that uses pre-fetching and cache locking to reduce cache miss rates. The proposed architecture combines pre-fetching and cache locking to reduce both the miss rates and the penalty of cache misses. The pre=fetching algorithm can pre=fetch sequential and non-sequential streams of instructions with accuracy, avoiding cache pollution or useless memory traffic. The cache locking mechanism is a dynamic one and can decrease the cache miss rate of the system by only locking the appropriate memory blocks. Both techniques work together, complimenting each other with the pre-fetching exploiting the spatial locality while the cache locking makes use of temporal locality.

Neethu et al. [60] proposed a way halted prediction cache as an energy-efficient cache architecture for embedded processors. The technique aims to reduce the number of active ways to one in case the prediction being a hit and active ways to the number of ways halted in case of a miss prediction. The technique only seeks to activate only the matched ways there by achieving dynamic energy savings over the conventional set-associative cache architecture.

Mohammad et al. [111] proposed an architecture for GPUs called the 'Efficient STT-RAM Last Level Cache'. The STT-RAM L2 cache architecture proposed can improve IPC by more than 100% while reducing the average consumed power by 20%.

Lee et al. [54] proposed a technique that partitions hybrid caches in multi-core architectures to reduce energy consumption. Utility-based partitioning is used to determine the sizes of the partitions for every core so that the number of misses is minimised. The replacement policy is re-designed to incorporate the technique into hybrid caches. When a store operation of a cache causes a miss in the shared cache, the corresponding new block is placed in the SRAM. Simulation results show that the technique improves the performance by reducing energy consumption by 3.6% on quad-core systems and energy consumption reduction of 11% on hybrid caches.

Jianwei et al. [19] proposed a technique called 'Way-tagged Cache' which is an energy-efficient L2 cache architecture using way-tag information under write-through policy. The technique improves the energy efficiency of write-through cache systems with minimal area overhead and performance degradation. The data in L2 cache is directly mapped to data in L1 cache by way of tags. During the subsequent accesses, which there is a right hit in the L1 cache, L2 cache can also be accessed in an equivalent direct-mapping. This process accounts for most L2 cache accesses in most applications thereby reducing dynamic energy consumption in L2 caches.

5. DYNAMIC AND LEAKAGE ENERGY SAVING TECHNIQUES

Jaekyu et al. [53] proposed a technique called Green Cache for exploiting the disciplined memory model of open CL on GPUS. Open CL is used because it allows applications to run on GPUS. Dynamic energy is saved by a technique called Region-Aware caching. Cache behaviours of each region is monitored either by compiler static analysis or by dynamic hardware training. Open CL specifies region information and passes it to the GPUs. The caching is directed only to regions with higher cache hit rates. Leakage energy is saved by a technique called Region-Aware Cache Resizing. The size of the cache is provided by the open CL libraries. The size of the required cache for a programme is calculated and if its smaller than the total size of the cache, then the remainder is turned off to save the leakage energy. The simulation results show that with Green Cache, dynamic energy of 56% can be saved in L1 Cache and 39% in L2 Cache. Leakage energy of 5% can be saved in L2 cache with no material performance degradation and off-chip access increases.

Alejandro et al. [82] proposed a technique called the Design of Hybrid Second Level Caches which is the hybridisation of SRAM and DRAM to minimise performance losses, energy consumption over the SRAM area and to maximise performance over the DRAM area. SRAM is the fastest existing memory but it has drawbacks of having low density and high leakage proportional to the number of transistors. DRAM on the other hand is slow and has high density. The results of the experiments show that the hybrid cache improves performance by 5.9% on average and the total energy consumption is reduced by 32%.

Valero et al. [85] designed a hybrid cache architecture with data encoding using low cost peripheral circuitry to improve energy, latency and endurance of cache simultaneously. The technique splits the input data between the STT-RAM and SRAM caches per the proportion of ones in input data. The data with zeros is stored on SRAM caches. Zeros on STT-RAM improve the performance and energy efficiency of the cache because a zero on STT-RAM cell consumes 3.5x-6.5x less energy than writing ones. The ones are written on the symmetric ST-SRAM cell which consumes very low power in writing ones. This technique achieves 42% and 53% energy efficiency and 9.3% and 9.1% performance improvement.

6. ENERGY SAVING CHIPS

This section is a survey of the most recent commercial chips designed to save energy in the cache architectures.

Warnock et al. [89] discuss a circuit and physical design of the zEnterprise EC12 Microprocessor chips and multi-chip module. This is a processor chip (CP), level-4 cache chip (SC) and the multi-chip module at the heart of the EC12 system. The chips were implemented in the IBM's high performance 32nm high-k/metal-gate SOI technology. The design of the chip is such that it contains 6 super-scalars, out-of-order processor cores, running at 5.5GHz, while the SC chip contains 192MB of eDRAM cache. Six CP chips and two SC chips are mounted on a high-performance glass-ceramic substrate providing high-bandwidth and low latency interconnections. The results of the experiments show that the EC12 MCM provide an unprecedented level of system performance improvement. The introduction of the eDRAM in the cache introduces density, resulting in low leakage consumption.

Shum et al. [77] discuss an IBM zNext chip, which is a 3rd Generation high frequency microprocessor chip. The chip has 6 cores instead of 4, dedicated-core core processors and a 48MB eDRAM on-chip shared L3. The IBM zNext chip maximizes Out-of-Order window because of its improved dispatch grouping efficiencies which are the reduced cracked instructions overhead, increased branches per group and added instruction queue for re-clumping. zNext also has the capability of accelerating specific functions because of its short-circuit

executions, dedicated Fixed-point divide engine resulting in 25-65% faster operations and millicode operations.

Haupt et al. [35] discuss a heat transfer modelling of a dual-side cooled microprocessor chip stack with embedded micro-channels. This is a cooling approach also called dual-side liquid cooling where the heat removal from the chip stack is enhanced by placing a liquid cooled interposer and thus dissipates heat from the stacked dies. This design achieves a hydraulic diameter as large a 200µm.

Warnock et al. [88] describes a technique called the 5.5GHz System z Microprocessor and multi-chip module. This chip features a high-frequency processor core for running at 5.5GHz in a 32nm high-k CMOS technology using 15 levels of metal. This chip is an upgrade of the 45nm chip with significant improvements made to the core and nest to increase the performance and through put of the design. The 32nm chip can achieve a higher operating frequency while running at lower operating voltages than the 45nm

Fischer et al. [26] discuss a performance Enhancement for 14nm High volume manufacturing microprocessor and system on-chip processes. This is a performance enhancement to Intel's 14nm high-performance logic technology interconnects. The enhancements are the improved RC performance and intrinsic capacitance for back end metal layers over a range of process versions and metal stacks offered for optimal cost and density targeted for various applications. These enhancements were implemented with no loss in reliability performance.

Choi at al. [16] discuss a sub-µw on-chip oscillator for fully integrated system-on-chip designs. This oscillator introduces a resistive frequency locked loop topology for accurate clock generation. A switched-capacitor from the topology. The oscillator is then matched to a temperature-compensated on-chip resistor using an ultra-low power amplifier. A test chip is fabricated in 0.18µm CMOS that exhibits a temperature sufficient of 34.3ppm/degrees Celsius with long-term stability of less than 7ppm.Yen et al. [72] discuss a low store energy and robust ReRAM-Based flip-flop for normally-off microprocessors. The technique of Normally-of Computing (NoC) benefits microsystems with long sleep time. NoC can turn off power to achieve zero power consumption and can activate microsystems instantly. This work is a novel ReRAM based non-volatile flip-flop (NVFF), fabricated using 90nm CMOS technology and the ReRAM process of the Industrial Research Institute. This ReRAM based NVFF can reduce store energy by 36.4%, restore time by 64.2% and circuit area by 42.8% compared with the state of the art complimentary design. The ReRAM based NVFF is also superior in reducing restoration error by 9.44% under hardship condition compared to NVFF with a single NV device.

Andersen et al. [4] discuss a 10w on-chip switched capacitor voltage regulator with Feed forward regulation capability for granular microprocessor power delivery. This on-chip switched capacitor voltage regulator (SCVR) is designed and implemented in a 32nm SOI CMOS Technology. The semi-conductor technology features the high capacitance density and low loss deep trench capacitor resulting in high efficiency and high power density on chip SCRV designs. The implemented on-chip switched capacitor voltage regulator provides a 0.7V-1.1V output voltage from 1.8V input. It achieves 85.1% maximum efficiency at 3.2W/nmm power density. The overall power consumption of the microprocessor can be reduced by 7%.

7. CONCLUSION

Various researchers continue to tackle the problem caused by inefficient consumption of energy in the cache architectures. The progress is being made towards greater efficiency in the consumption of energy and this progress is being seen even in the development of commercial chips.

This work has reviewed some of the architectural techniques for saving the dynamic and leakage energy. Commercial chips were also reviewed. This work serves as a quick reference guide for the energy saving techniques, recent commercial chips and cache architecture.

8. REFERENCES

[1] Aahn, J., Yoo, S., & Choi, K. (2016). Prediction Hybrid Cache: An Energy-Efficient STT-RAM Cache Architecture. IEEE Transactions on Computers, 65(3), 940–951.

[2] Abadal, S., Mestres, A., Martinez, R., Alarcon, E., Cabellos-Aparicio, A., & Martinez, R. (2015). Multicast On-chip Traffic Analysis Targeting Manycore NoC Design. In 2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (pp. 370–378). IEEE

[3] Aneesh Kumar, A. G., Janeera, D. A., & Ramesh, M. (2014). Power and performance efficient secondary cache using tag bloom architecture. In 2014 International Conference on Electronics and Communication Systems (ICECS) (pp. 1–5). IEEE.

[4] Andersen, T., Krismer, F., Kolar, J., Toifl, T., Menolfi, C., Kull, L., … Francese, P. A. (2016). A 10 W On-Chip Switched Capacitor Voltage Regulator with Feedforward Regulation Capability for Granular Microprocessor Power Delivery. IEEE Transactions on Power Electronics, 1–1

[5] Arima, E., Noguchi, H., Nakada, T., Miwa, S., Takeda, S., Fujita, S., & Nakamura, H. (2015). Immediate sleep: Reducing energy impact of peripheral circuits in STT-MRAM caches. In 2015 33rd IEEE International Conference on Computer Design (ICCD) (pp. 149–156).

[6] Arora, N. D., Worley, S., & Ganpule, D. R. (2015). FieldRC, a GPU accelerated interconnect RC parasitic extractor for full-chip designs. In 2015 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC) (pp. 459–462). IEEE.

[7] Bardine, A., Comparetti, M., Foglia, P., & Prete, C. A. (2014). Evaluation of Leakage Reduction Alternatives for Deep Submicron Dynamic Nonuniform Cache Architecture Caches. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 22(1), 185–190.

[8] Bengueddach, A., Senouci, B., Niar, S., & Beldjilali, B. (2013). Energy consumption in reconfigurable mpsoc architecture: Two-level caches optimization oriented approach. In 2013 8th IEEE Design and Test Symposium (pp. 1–6). IEEE.

[9] Cache Simulation Main Page. (n.d.). Retrieved from http://www.ecs.umass.edu/ece/koren/architecture/Cache/frame0.htm

[10] Chakraborty, S., Das, S., & Kapoor, H. K. (2015). Performance Constrained Static Energy Reduction Using Way-Sharing Target-Banks. In 2015 IEEE International Parallel and Distributed Processing Symposium Workshop

(pp. 444–453). IEEE.

[11] Chen, K.-C. J., Chao, C.-H., & Wu, A.-Y. A. (2015). Thermal-Aware 3D Network-On-Chip (3D NoC) Designs: Routing Algorithms and Thermal Managements. IEEE Circuits and Systems Magazine, 15(4), 45–69

[12] Chen, X., Chen, W., Lu, Z., Long, P., Yang, S., & Wang, Z. (2015). A Duplication-Aware SSD-Based Cache Architecture for Primary Storage in Virtualization Environment. IEEE Systems Journal, 1–12.

[13] Cheng, W.-K., Cheng, P.-C., & Li, X.-L. (2016). Adaptive page allocation of DRAM/PCRAM hybrid memory architecture. In 2016 5th International Symposium on Next-Generation Electronics (ISNE) (pp. 1–2). IEEE.

[14] Chien, T.-K., Chiou, L.-Y., Lee, C.-C., Chuang, Y.-C., Ke, S.-H., Sheu, S.-S., … Wu, C.-I. (2016). An energy-efficient nonvolatile microprocessor considering software-hardware interaction for energy harvesting applications. In 2016 International Symposium on VLSI Design, Automation and Test (VLSI-DAT) (pp. 1–4). IEEE.

[15] Chenxi Zhang, Xiaodong Zhang, & Yong Yan. (n.d.). Multi-column implementations for cache associativity. In Proceedings International Conference on Computer Design VLSI in Computers and Processors (pp. 504–509). IEEE Comput. Soc.

[16] Choi, M., Jang, T., Bang, S., Shi, Y., Blaauw, D., & Sylvester, D. (2016). A 110 nW Resistive Frequency Locked On-Chip Oscillator with 34.3 ppm/°C Temperature Stability for System-on-Chip Designs. IEEE Journal of Solid-State Circuits, 1–13.

[17] Chung, S. W., & Skadron, K. (2008). On-Demand Solution to Minimize I-Cache Leakage Energy with Maintaining Performance. IEEE Transactions on Computers, 57(1), 7–24.

[18] Cilku, B., Prokesch, D., & Puschner, P. (2015). A Time-Predictable Instruction-Cache Architecture that Uses Prefetching and Cache Locking. In 2015 IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops (pp. 74–79). IEEE

[19] Dai, J., & Wang, L. (2013). An Energy-Efficient L2 Cache Architecture Using Way Tag Information Under Write-Through Policy. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 21(1), 102–112.

[20] Datta, A. K., & Patel, R. (2014). CPU Scheduling for Power/Energy Management on Multicore Processors Using Cache Miss and Context Switch Data. IEEE Transactions on Parallel and Distributed Systems, 25(5), 1190–1199.

[21] De, V. (2015). Fine-grain power management in manycore processor and System-on-Chip (SoC) designs. In 2015 IEEE/ACM International Conference on Computer-Aided Design (ICCAD) (pp. 159–164). IEEE.

[22] Degnan, B., Marr, B., & Hasler, J. (2016). Assessing Trends in Performance per Watt for Signal Processing Applications. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 24(1), 58–66.

[23] Eeckhout, L. (2015). Hot Chips in an Increasingly Diverse Microprocessor Landscape. IEEE Micro, 35(2), 2–3.

[24] Ezz-Eldin, R., El-Moursy, M. A., & Hamed, H. F. A. (2015). Analysis and design of Network on Chip under high process variation. In 2015 IEEE International Conference on Electronics, Circuits, and Systems (ICECS) (pp. 508–509). IEEE.

[25] Farmahini-Farahani, A., Ho Ahn, J., Morrow, K., & Sung Kim, N. (2015). DRAMA: An Architecture for Accelerated Processing Near Memory. IEEE Computer Architecture Letters, 14(1), 26–29.

[26] Fischer, K., Chang, H. ., Ingerly, D., Jin, I., Kilambi, H., Longun, J., … Yashar, P. (2016). Performance enhancement for 14nm high volume manufacturing microprocessor and system on a chip processes. In 2016 IEEE International Interconnect Technology Conference / Advanced Metallization Conference (IITC/AMC) (pp. 5–7). IEEE.

[27] Frustaci, F., Corsonello, P., Perri, S., & Cocorullo, G. (n.d.). Leakage energy reduction techniques in deep submicron cache memories: a comparative study. In 2006 IEEE International Symposium on Circuits and Systems (p. 4). IEEE.

[28] Frustaci, F., Corsonello, P., Perri, S., & Cocorullo, G. (n.d.). A New Scheme to Reduce Leakage in DeepSubmicron Cache Memories with No Extra Dynamic Consumption. In 2006 Ph.D. Research in Microelectronics and Electronics (pp. 61–64). IEEE.

[29] Ghaemi, S. G., Monazzah, A. M. H., Farbeh, H., & Miremadi, S. G. (2015). LATED: Lifetime-Aware Tag for Enduring Design. In 2015 11th European Dependable Computing Conference (EDCC) (pp. 97–107). IEEE.

[30] Ghosh, M., & Lee, H.-H. S. (2007). Virtual Exclusion: An architectural approach to reducing leakage energy in caches for multiprocessor systems. In 2007 International Conference on Parallel and Distributed Systems (pp. 1–8). IEEE.

[31] Golubeva, O., Loghi, M., Poncino, M., & Macii, E. (2007). Architectural Leakage-Aware Management of Partitioned Scratchpad Memories. In 2007 Design, Automation & Test in Europe Conference & Exhibition (pp. 1–6). IEEE.

[32] Goudarzi, M., Ishihara, T., & Yasuura, H. (2007). A Software Technique to Improve Yield of Processor Chips in Presence of Ultra-Leaky SRAM Cells Caused by Process Variation. In 2007 Asia and South Pacific Design Automation Conference (pp. 878–883). IEEE.

[33] Grani, P., Proietti, R., Cheung, S., & Ben Yoo, S. J. (2016). Flat-Topology High-Throughput Compute Node With AWGR-Based Optical-Interconnects. Journal of Lightwave Technology, 34(12), 2959–2968.

[34] Hameed, F., & Tahoori, M. B. (2016). Architecting STT Last-Level-Cache for performance and energy improvement. In 2016 17th International Symposium on Quality Electronic Design (ISQED) (pp. 319–324). IEEE

[35] Haupt, M., Brunschwiller, T., Keller, J., & Ozsun, O. (2015). Heat transfer modelling of a dual-side cooled microprocessor chip stack with embedded micro-channels. In 2015 21st International Workshop on Thermal Investigations of ICs and Systems (THERMINIC) (pp. 1–4). IEEE.

[36] Hu, J. S., Nadgir, A., Vijaykrishnan, N., Irwin, M. J., & Kandemir, M. (n.d.). Exploiting program hotspots and code

sequentiality for instruction cache leakage management. In Proceedings of the 2003 International Symposium on Low Power Electronics and Design, 2003. ISLPED '03. (pp. 402–407). ACM

[37] Intel An Overview of Cache Page 2 2.1 Basic Model CPU Cache Memory Main DRAM Memory System Interface. (n.d.).

[38] Isaza-Gonzalez, J., Serrano-Cases, A., Restrepo-Calle, F., Cuenca-Asensi, S., & Martinez-Alvarez, A. (2016). Dependability evaluation of COTS microprocessors via on-chip debugging facilities. In 2016 17th Latin-American Test Symposium (LATS) (pp. 27–32). IEEE.

[39] Jang, H., Kim, J., Gratz, P., Yum, K. H., & Kim, E. J. (2015). Bandwidth-efficient on-chip interconnect designs for GPGPUs. In Proceedings of the 52nd Annual Design Automation Conference on - DAC '15 (pp. 1–6). New York, New York, USA: ACM Press.

[40] Jing, N., Jiang, L., Zhang, T., Li, C., Fan, F., & Liang, X. (2016). Energy-Efficient eDRAM-Based On-Chip Storage Architecture for GPGPUs. IEEE Transactions on Computers, 65(1), 122–135.

[41] Jishen Zhao, Xiangyu Dong, & Yuan Xie. (2011). An energy-efficient 3D CMP design with fine-grained voltage scaling. In 2011 Design, Automation & Test in Europe (pp. 1–4). IEEE.

[42] Divya Jebaseeli, A., & Kiruba, M. (2014). Design of low power L2 cache architecture using partial way tag information. In 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE) (pp. 1–6). IEEE.

[43] Kadjo, D., Kim, H., Gratz, P., Hu, J., & Ayoub, R. (2013). Power gating with block migration in chip-multiprocessor last-level caches. In 2013 IEEE 31st International Conference on Computer Design (ICCD) (pp. 93–99). IEEE

[44] Kalla, P., Xiaobo Sharon Hu, & Henkel, J. (2006). Distance-based recent use (DRU): an enhancement to instruction cache replacement policies for transition energy reduction. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 14(1), 69–80.

[45] Khaitan, S. K., & McCalley, J. D. (2013). A hardware-based approach for saving cache energy in multicore simulation of power systems. In 2013 IEEE Power & Energy Society General Meeting (pp. 1–5). IEEE.

[46] Khaitan, S. K., & McCalley, J. D. (2013). A hardware-based approach for saving cache energy in multicore simulation of power systems. In 2013 IEEE Power & Energy Society General Meeting (pp. 1–5). IEEE.

[47] Kim, C. H., Jae-Joon Kim, Ik-Joon Chang, & Roy, K. (n.d.). PVT-aware leakage reduction for on-die caches with improved read stability. In ISSCC. 2005 IEEE International Digest of Technical Papers. Solid-State Circuits Conference, 2005. (pp. 482–484). IEEE.

[48] Kim, C. H., Kim, J.-J., Chang, I.-J., & Roy, K. (2006). PVT-Aware Leakage Reduction for On-Die Caches With Improved Read Stability. IEEE Journal of Solid-State Circuits, 41(1), 170–178.

[49] Kim, N., Ahn, J., Seo, W., & Choi, K. (2015). Energy-efficient exclusive last-level hybrid caches consisting of SRAM and STT-RAM. In 2015 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC) (pp. 183–188). IEEE.

[50] Kudithipudi, D., Petko, S., & John, E. B. (2008). Caches for Multimedia Workloads: Power and Energy Tradeoffs. IEEE Transactions on Multimedia, 10(6), 1013–1021.

[51] Kim, N., Ahn, J., Seo, W., & Choi, K. (2015). Energy-efficient exclusive last-level hybrid caches consisting of SRAM and STT-RAM. In 2015 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC) (pp. 183–188). IEEE.

[52] Lee, J., & Kim, S. (2015). Filter Data Cache: An Energy-Efficient Small L0 Data Cache Architecture Driven byMiss Cost Reduction. IEEE Transactions on Computers, 64(7), 1927–1939.

[53] Lee, J., Woo, D. H., Kim, H., & Azimi, M. (2015). GREEN Cache: Exploiting the Disciplined Memory Model of OpenCL on GPUs. IEEE Transactions on Computers, 64(11), 3167–3180.

[54] Lee, S., Kang, K., Jung, J., & Kyung, C.-M. (2016). Hybrid L2 NUCA Design and Management Considering Data Access Latency, Energy Efficiency, and Storage Lifetime. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1–14.

[55] Lin, I.-C., & Chiou, J.-N. (2015). High-Endurance Hybrid Cache Design in CMP Architecture With Cache Partitioning and Access-Aware Policies. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 23(10), 2149–2161.

[56] Lin, I.-J., Yang, M.-J., & Hu, K.-S. (2016). Single layer differential group routing for flip-chip designs. In 2016 International Symposium on VLSI Design, Automation and Test (VLSI-DAT) (pp. 1–4). IEEE.

[57] Lockwood, J. W., & Monga, M. (2016). Implementing Ultra-Low-Latency Datacenter Services with Programmable Logic. IEEE Micro, 36(4), 18–26.

[58] Loghi, M., Golubeva, O., Macii, E., & Poncino, M. (2010). Architectural Leakage Power Minimization of Scratchpad Memories by Application-Driven Subbanking. IEEE Transactions on Computers, 59(7), 891–904.

[59] Lou, M., Wu, L., Shi, S., & Lu, P. (2014). An energy-efficient two-level cache architecture for chip multiprocessors. In Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT) (pp. 1–5). IEEE.

[60] Mallya, N. B., Patil, G., & Raveendran, B. (2015). Way Halted Prediction Cache: An Energy Efficient Cache Architecture for Embedded Processors. In 2015 28th International Conference on VLSI Design (pp. 65–70). IEEE.

[61] Matsuno, S., Tawada, M., Yanagisawa, M., Kimura, S., Togawa, N., & Sugibayashi, T. (2013). Energy evaluation for two-level on-chip cache with non-volatile memory on mobile processors. In 2013 IEEE 10th International Conference on ASIC (pp. 1–4). IEEE.

[62] Mittal, S., Cao, Y., & Zhang, Z. (2014). MASTER: A Multicore Cache Energy-Saving Technique Using Dynamic Cache Reconfiguration. IEEE Transactions on Very Large

Scale Integration (VLSI) Systems, 22(8), 1653–1665

[63] Mittal, S., Zhang, Z., & Cao, Y. (2013). CASHIER: A Cache Energy Saving Technique for QoS Systems. In 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems (pp. 43–48). IEEE.

[64] Mittal, S. (n.d.). A Survey of Architectural Techniques For Improving Cache Power Efficiency

[65] Mittal, S., Zhang, Z., & Cao, Y. (2013). CASHIER: A Cache Energy Saving Technique for QoS Systems. In 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems (pp. 43–48). IEEE.

[66] Mittal, S., Zhang, Z., & Vetter, J. S. (2013). FlexiWay: A cache energy saving technique using fine-grained cache reconfiguration. In 2013 IEEE 31st International Conference on Computer Design (ICCD) (pp. 100–107). IEEE

[67] Mittal, S., Zhang, Z., & Vetter, J. S. (2013). FlexiWay: A cache energy saving technique using fine-grained cache reconfiguration. In 2013 IEEE 31st International Conference on Computer Design (ICCD) (pp. 100–107). IEEE.

[68] Moein, S., Gulliver, T. A., Gebali, F., & Alkandari, A. (2016). A New Characterization of Hardware Trojans. IEEE Access, 4, 2721–2731.

[69] Mohammadi, M., Han, S., Aamodt, T. M., & Dally, W. J. (2015). On-Demand Dynamic Branch Prediction. IEEE Computer Architecture Letters, 14(1), 50–53

[70] Monchiero, M., Canal, R., & Gonzalez, A. (2009). Using Coherence Information and Decay Techniques to Optimize L2 Cache Leakage in CMPs. In 2009 International Conference on Parallel Processing (pp. 1–8). IEEE

[71] Nadgir, A., Kandemir, M., Guangyu Chen, & Guilin Chen. (n.d.). An access pattern based energy management strategy for instruction caches. In IEEE International [Systems-on-Chip] SOC Conference, 2003. Proceedings. (pp. 175–178). IEEE.

[72] Ray, A., & Choudhry, A. (2015). Time optimization of instruction execution in FPGA using embedded systems. In 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE) (pp. 566–572). IEEE.

[73] Rossi, D., Tenentes, V., Khursheed, S., & Al-Hashimi, B. M. (2015). BTI and leakage aware dynamic voltage scaling for reliable low power cache memories. In 2015 IEEE 21st International On-Line Testing Symposium (IOLTS) (pp. 194–199). IEEE.

[74] Sai Manoj, P. D., & Hao Yu. (2013). Cyber-physical management for heterogeneously integrated 3D thousand-core on-chip microprocessor. In 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013) (pp. 533–536). IEEE.

[75] Sampaio, F., Shafique, M., Zatt, B., Bampi, S., & Henkel, J. (2015). Approximation-aware Multi-Level Cells STT-RAM cache architecture. In 2015 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES) (pp. 79–88). IEEE.

[76] Seongmoo Heo, Barr, K., Hampton, M., & Asanovic, K. (n.d.). Dynamic fine-grain leakage reduction using leakage-biased bitlines. In Proceedings 29th Annual International Symposium on Computer Architecture (pp. 137–147). IEEE Comput. Soc.

[77] Shum, C. K., Busaba, F., & Jacobi, C. (2013). IBM zEC12: The Third-Generation High-Frequency Mainframe Microprocessor. IEEE Micro, 33(2), 38–47.

[78] Soontae Kim, Vijaykrishnan, N., Kandemir, M., & Irwin, M. J. (n.d.). Predictive precharging for bitline leakage energy reduction [microprocessor caches]. In 15th Annual IEEE International ASIC/SOC Conference (pp. 36–40). IEEE.

[79] Subha, S. (2014). A reconfigurable cache architecture. In 2014 International Conference on High Performance Computing and Applications (ICHPCA) (pp. 1–5). IEEE.

[80] Tao Li, & John, L. K. (n.d.). OS-aware Tuning: Improving Instruction Cache Energy Efficiency on System Workloads. In 2006 IEEE International Performance Computing and Communications Conference (pp. 321–330). IEEE

[81] Tsoutsos, N. G., & Maniatakos, M. (2015). The HEROIC Framework: Encrypted Computation Without Shared Keys. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34(6), 875–888.

[82] Tu, C.-Y., Chang, Y.-Y., King, C.-T., Chen, C.-T., & Wang, T.-Y. (2014). Traffic-aware frequency scaling for balanced on-chip networks on GPGPUs. In 2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS) (pp. 87–94). IEEE.

[83] Ubal, R., Sahuquillo, J., Petit, S., & Lopez, P. (2006). Applying the zeros switch-off technique to reduce static energy in data caches. In 2006 18th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'06) (pp. 133–140). IEEE.

[84] Ubal, R., Sahuquillo, J., Petit, S., Hassan, H., & Lopez, P. (2007). Leakage Current Reduction in Data Caches on Embedded Systems. In The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007) (pp. 45–50). IEEE.

[85] Valero, A., Sahuquillo, J., Petit, S., Lopez, P., & Duato, J. (2015). Design of Hybrid Second-Level Caches. IEEE Transactions on Computers, 64(7), 1884–1897

[86] Wang, W., & Mishra, P. (2010). Leakage-Aware Energy Minimization Using Dynamic Voltage Scaling and Cache Reconfiguration in Real-Time Systems. In 2010 23rd International Conference on VLSI Design (pp. 357–362). IEEE.

[87] Wang, Z., Jimenez, D. A., Xu, C., Sun, G., & Xie, Y. (2014). Adaptive placement and migration policy for an STT-RAM-based hybrid cache. In 2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA) (pp. 13–24). IEEE.

[88] Warnock, J., Chan, Y. H., Harrer, H., Rude, D., Puri, R., Carey, S., … Webb, C. (2013). 5.5GHz system z microprocessor and multi-chip module. In 2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers (pp. 46–47). IEEE.

[89] Warnock, J., Chan, Y., Harrer, H., Carey, S., Salem, G., Malone, D., … Webb, C. (2014). Circuit and Physical Design of the zEnterpriseTM EC12 Microprocessor Chips and Multi-Chip Module. IEEE Journal of Solid-State Circuits, 49(1), 9–18.

[90] Weixun Wang, & Mishra, P. (2012). System-Wide Leakage-Aware Energy Minimization Using Dynamic Voltage Scaling and Cache Reconfiguration in Multitasking Systems. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 20(5), 902–910.

[91] Wu, Y., Zhao, J., Chen, D., & Guo, D. (2016). Modeling of Gaussian Network-Based Reconfigurable Network-on-Chip Designs. IEEE Transactions on Computers, 65(7), 2134–2142.

[92] Xiaoping, H., & Jianfeng, A. (2013). A Novel Architecture to Identify the Microprocessor Chips by Implanting Timing-Fault Execution Unit. In 2013 IEEE 16th International Conference on Computational Science and Engineering (pp. 766–769). IEEE.

[93] Yan-Fang, S., Jian-Guo, S., & Yu-Qian, X. (2015). Design and Application of Distributed Intelligent Greenhouse Computerized System. In 2015 Seventh International Conference on Measuring Technology and Mechatronics Automation (pp. 331–334). IEEE.

[94] Yen, C.-H., Chen, C.-H., & Chen, K.-C. (2015). A memory-efficient NoC system for OpenCL many-core platform. In 2015 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1386–1389). IEEE

[95] Yongsoo Joo, Dimin Niu, Xiangyu Dong, Guangyu Sun, Naehyuck Chang, & Yuan Xie. (2010). Energy- and endurance-aware design of phase change memory caches. In 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010) (pp. 136–141). IEEE.

[96] Yu, S., & Zhang, W. (2008). Adaptive Drowsy Cache Control for Java Applications. In 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (pp. 185–191). IEEE.

[97] Yue Wang, Roy, S., & Ranganathan, N. (2012). Run-time power-gating in caches of GPUs for leakage energy savings. In 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE) (pp. 300–303). IEEE

[98] Zhang, Q., Li, Z., & Pei, S. (2013). A high performance inter-chip fiber communication scheme with short frame protocol. In 2013 IEEE Third International Conference on Information Science and Technology (ICIST) (pp. 850–853). IEEE.

[99] Sajith, V., & Sobhan, C. B. P. (2012). Characterization of Heat Dissipation From a Microprocessor Chip Using Digital Interferometry. IEEE Transactions on Components, Packaging and Manufacturing Technology, 2(8), 1298–1306.

[100] Gardner, A. T., & Collins, J. A. (2012). Advancements in high-performance timing for long term underwater experiments: A comparison of chip scale atomic clocks to traditional microprocessor-compensated crystal oscillators. In 2012 Oceans (pp. 1–8). IEEE.

[101] Warnock, J., Chan, Y. H., Harrer, H., Rude, D., Puri, R., Carey, S., … Webb, C. (2013). 5.5GHz system z microprocessor and multi-chip module. In 2013 IEEE International Solid-State Circuits Conference Digest of Technical Papers (pp. 46–47). IEEE.

[102] Sai Manoj, P. D., & Hao Yu. (2013). Cyber-physical management for heterogeneously integrated 3D thousand-core on-chip microprocessor. In 2013 IEEE International Symposium on Circuits and Systems (ISCAS2013) (pp. 533–536). IEEE.

[103] Isaza-Gonzalez, J., Serrano-Cases, A., Restrepo-Calle, F., Cuenca-Asensi, S., & Martinez-Alvarez, A. (2016). Dependability evaluation of COTS microprocessors via on-chip debugging facilities. In 2016 17th Latin-American Test Symposium (LATS) (pp. 27–32). IEEE.

[104] Haupt, M., Brunschwiller, T., Keller, J., & Ozsun, O. (2015). Heat transfer modelling of a dual-side cooled microprocessor chip stack with embedded micro-channels. In 2015 21st International Workshop on Thermal Investigations of ICs and Systems (THERMINIC) (pp. 1–4). IEEE.

[105] Xiaoping, H., & Jianfeng, A. (2013). A Novel Architecture to Identify the Microprocessor Chips by Implanting Timing-Fault Execution Unit. In 2013 IEEE 16th International Conference on Computational Science and Engineering (pp. 766–769). IEEE.

[106] Shum, C.-L. (2012). IBM zNext - the 3rd generation high frequency microprocessor chip. In 2012 IEEE Hot Chips 24 Symposium (HCS) (pp. 1–18). IEEE

[107] Zhu, H., & Kursun, V. (2014). Triple-threshold-voltage 9-transistor SRAM cell for data stability and energy-efficiency at ultra-low power supply voltages. In 2014 26th International Conference on Microelectronics (ICM) (pp. 176–179). IEEE.