International Journal of Emerging Trends in Science and Technology
5 (1), June 2019, 01-06
http://www.publishingindia.com/ijetst

Analysis of Energy Efficient Two Level Cache Using Counting Bloom Filter

S. Kiruthika[1*] and I. Aravindaguru[2]
[1]Assistant Professor, Dept. of EIE, M. Kumarasamy College of Engineering, Karur, Tamil Nadu, India.
Email: kiruthikas.eie@mkce.ac.in
[2]Assistant Professor, Dept. of EIE, M. Kumarasamy College of Engineering, Karur, Tamil Nadu, India.
Email: aravindagurui.eie@mkce.ac.in
*Corresponding Author

Abstract: The microprocessor system's Caches consume much of a power and energy consumption. In this paper, a set associative cache system and way tagged L2 cache and way tagged L2 cache with counting bloom filter caches are compared. The tagged cache beyond continue the method tag of level two cache in the level one cache for the duration of understand writing process, this process prepare level two cache towards operate within a corresponding map in absolute process in the time of write hits that results in mainstream of the level two cache access. Here CBFs are proposed in the way tagged L2 cache to recover the energy and swiftness of membership tests by maintain a vague and compact demonstration of a large set to be searched. Designs were developed using behavioural VHDL and synthesized in Xilinx 8.1 Spartan 2E device. Experimental results were showing that the future way tagged L2 cache with CBF consumes lesser 50% of power than the set associative cache systems.

Keywords: CBF, Energy and power, Set associative cache, Way tagged L2 cache.

I. Introduction

The level of more on-chip system cache has been extensively processed in more microprocessor system. The memory cache is a hugely quick memory that is insert into the CPU and processed independent chip. The usage of central processing unit in the cache memory to combine the instruction this are jointly enforced to execute the programs, developing global system speed and compressing power dissipation. The structure level of the on-chip cache get maximized, the energy consumption per unit area also get rising regularly [2]. To maintain the data width all over the memory hierarchy write-after and write-over line of action were usually managed. Below the mark reverse rule, an altered cache slab was modified reverse to this pertinent under level cache at most period of the block was regard to be exchanged.

The period of mark from side to side rule, each samples of a cache slab were moderated instantly following the cache block was updated at the present cache, for all that the block great and impressive power of strength was not be removed. So, the policy write-through continues same data levels sample at all in the cache hierarchy all over more the time of running the process [1]. This feature was significant as complemented metal oxide semiconductor expertise be scale keen on top of the nanometre array; the yielding error has established because a substantial consistency question within on chip store memory system. It has be finished to sole occasion more bit upset is attainment inferior inside on chip recollections [4].

These difficulties have be address next to unlike level of the plan. At the structural design stature an efficient explanations are to remain information reliable in the middle of a variety of memory levels pecking order to shield the scheme from give way due to flexible error. Usually, store mark from side to side rule is intrinsically too learnt in the direction of spongy error for the cause that the information next to everyone connected level of the store pecking order to be reserved constant [8]. This is designed for the reason that below the mark from side to side rule, caches at the lesser height practice additional access throughout write operations [9]. There is a fundamental cache design trade-off between a set-associative cache and direct-mapped cache.

II. Two-Level Cache

This is a level of one and two cache system in Fig. 1. But the level one information cache apparatus the mark reverse regulation, a mark thump within the Level one store don't require to right of right of entry the Level 2 store. Stipulation of the Level one store is writing from side to side, after that together Level one and Level two caches require to be accepted for each write process [10]. The write through rule incur additional mark access inside the Level two store, which within revolve add in the direction of the power utilization of the store scheme. Energy rakishness was at the present measured because the significant problems inside store plan. Usually On-chip caches know how to get through regarding 50% of the whole energy in high-presentation microprocessors.

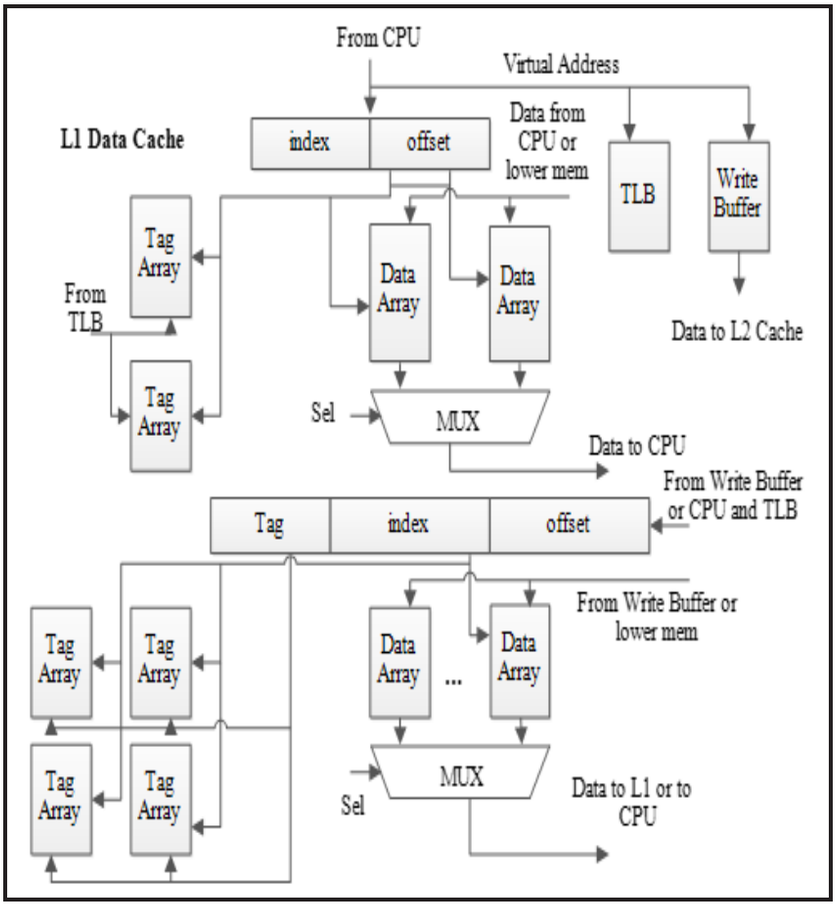

Fig. 1: Two-Level Cache Architecture

III. Way Tagged Cache

The most important task of this way-tagged cache that exploits the technique in sequence in Level two cache intended for getting better the energy effectiveness. This exploits that the fact that decrease figure of conduct accessed throughout Level two cache accesses [5]. This method tag as long as the input in order intended for the following mark access to the Level two store. Inside usual, together the understand writing and position fountain pen to paper accesses in the Level one cache might require in the direction of admission the Level two in Fig. 1 store.

Table I: Equal L2 Right of Entry Mode in Dissimilar Operation in the L1 Cache

Operations in the L1 Cache				
	Read Slap	Read Overlook	Write Slap	Write Overlook
L2	Negative access	Position associa-tive	Straight mapping	Position associative

This access guide towards dissimilar operation within the method tag store, because abridged in board one. This way-tagged cache architecture consists of a number of novel mechanisms mode label buffer, way register, way-tag arrays, way decoder in Table I.

A. Arrays Way-Tag

During this way-tagged cache, every cache row within the Level one cache keeps its Level two method [11] tag in sequence within the applicable opening of the method label array, everywhere just single Level one information collection in addition to the matching way label collection were worn in favour of effortlessness in Table II.

Table II: Process of Array Way Tag

Write	Fill in	Process
One	One	Put in writing arrays of way-tag
One	Zero	Understand writing arrays of way-tag
Zero	Zero	Not right of entry
Zero	One	Not right of entry

B. Buffer of Way Tag

The buffer of Way-tag supplies the way-tags temporarily understands writing beginning the arrays of method label.

The number of entries in the buffer of method label was alike because the mark down buffers of the Level one store and also shares the charge of indication through it. The shock absorber way tag have divide put in writing and understand writing logic in order to maintain equivalent understand writing and inscribe press like to the put in writing buffer of the Level two cache in Fig. 1.

C. Decoder Approach

The position of the decoder approach was to translate technique tags and trigger barely the determined habits in the Level two cache. While the dual signs are worn, the row dimension of method label array was designed because bits of $n = \log_2 N$, where n is the figure of approach in the Level two cache in Fig. 1. This minimized the power transparency as of the extra supports and the bang on chip region was insignificant.

D. Register Approach

This register approach was provided the tag technique in favour of the arrays approach. Designed for a technique Level two cache, labelled 11, 00, 01 and 10 be located within the method list in Table II. Every category single method inside the Level two cache. The information preserve be encumbered as of the Level two caches during the Level one cache, the matching approach tag in the method record was transmitting to the mode label arrays [10].

For the reason with the purpose of these innovative workings, the tagged approach store operates by means of diverse methods through understand writing and mark process in the table one. Into this tagged approach cache lone the method containing the needed information was processed within the Level two collection for a put in writing hit in the Level one store amateur dramatics the Level two store consistently a straight following accumulation to reduce power utilization lacking introduce presentation visual projection [9].

IV. Tagged Method of Cache With Counting Bloom Filter

A. Bloom Filter Process of Counting

For the most part of the structure methods provides on system including bloom filters to advance in the lead of power, difficulty, and wait of a range of processor design. Used for case of counting bloom filters be able to processed to recover the energy and presentation in snoop-coherent multiprocessor or multi-core systems.

It also know how to be applied to get better the measurability of cache weight preparation queues and to decrease teaching outputs by maintain in near the beginning fail to stop resolve at the Level one information accumulation. In the top of uses in counting bloom filters applied to cancelled broadcasts top of the interconnection system in multiprocessor systems and also decrease process to a great deal larger and thus a large amount slower and energy tag cache arrays or pleased addressable recollections in Fig. 2.

As shown in Fig. 3, a counting bloom filters was theoretically an array of counts indexed from first to last a disorder role of the element below devotion tests. It process three operations: 1) increase count 2) decrease count and 3) examination condition the count is nothing. The operations of two process are corresponds to calculate by one, and the operation of addition to proceeds factual or fake (bit output = single). The operations of two process are corresponds it as promoted and to the third one as a probe. It is considered by it is figure of entries and also the width of calculate per entry.

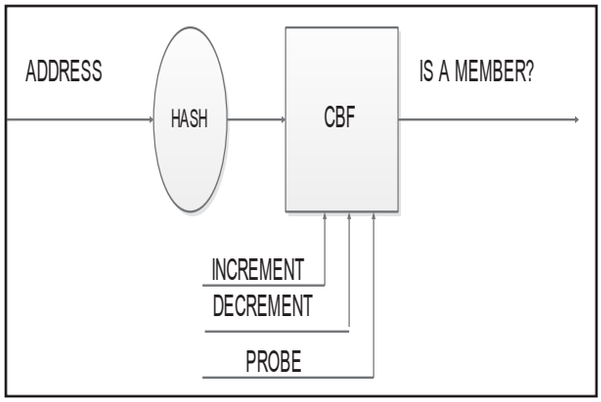

Fig. 2: CBF as a Black Box

B. Bloom Filter of Partial Tag Enhanced

In-case of a single on access, the unique Bloom filter cannot inform whether the receiving talk to is applying within the store method. So the verifying of the attendance of receiving address in-case of single ton access, this unfair method is processed in Fig. 3.

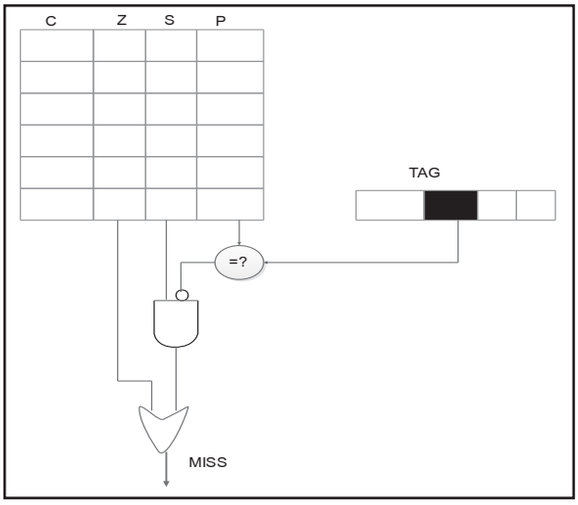

Fig. 3: Bloom Filter With Partial Tag

Sometimes we need to perform the concurrent statements also so for that partial tag is used to check the concurrent statements. Here tag has first four MSB bits. Here checking of value can be performed in the hash table, if any value present in tat hash table means data out value is 1 otherwise it will not go to inside of the operation so the overall power, energy, delay can be reduced. The proposed method tagged cache with the counting bloom filter.

V. Results and Discussions

The designs are analysed in this paper have been developed using VHDL and synthesized in Spartan 2E device of 2s600efg676-6. Here the analysis can be made in terms of power, delay, and energy in Fig. 4, 5 and 6. From the comparison results of Table III 49.80% of the power can be reduced in the proposed way tagged cache with the counting bloom filter.

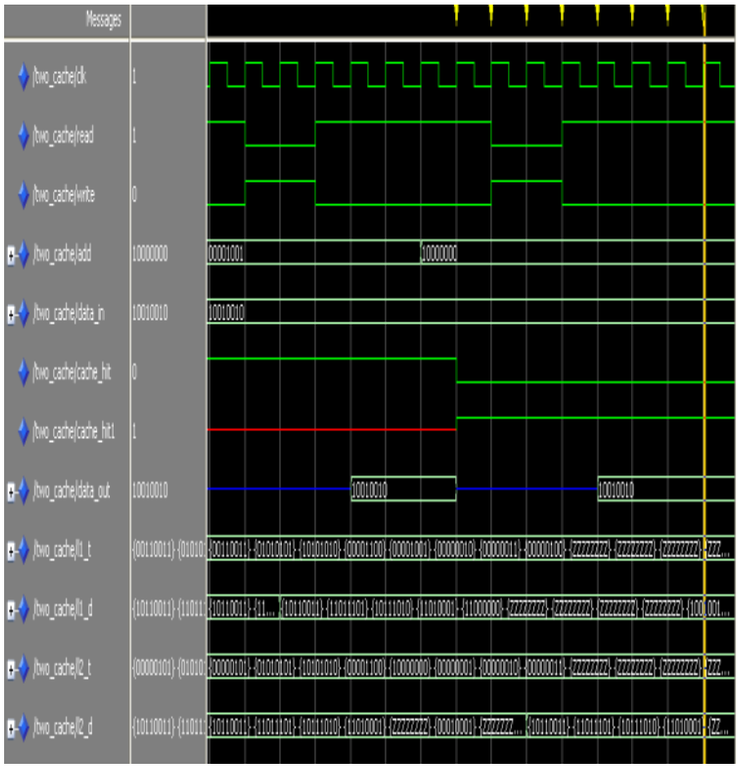

Fig. 4: Simulation Output for Two-Level Cache System

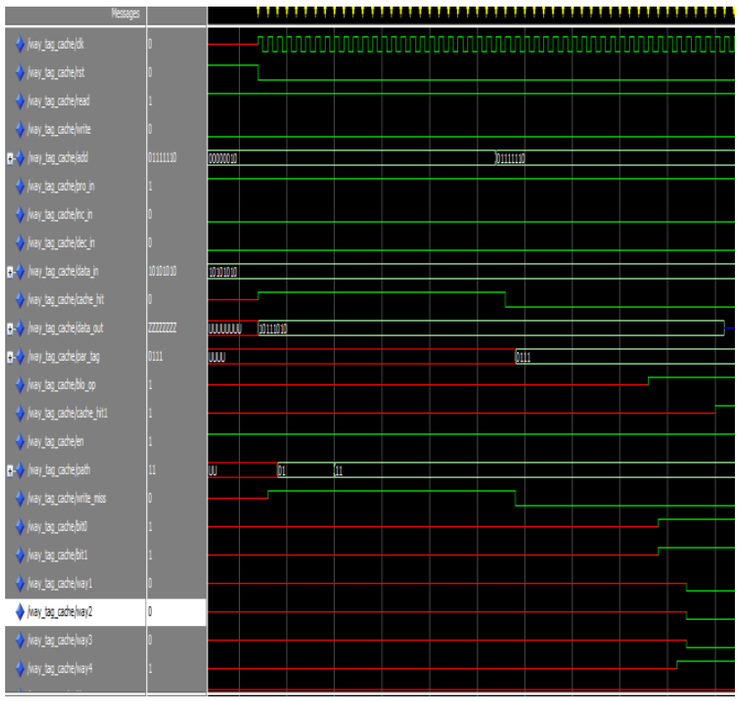

Fig. 5: Simulation Output for Way Tagged Cache With CBF

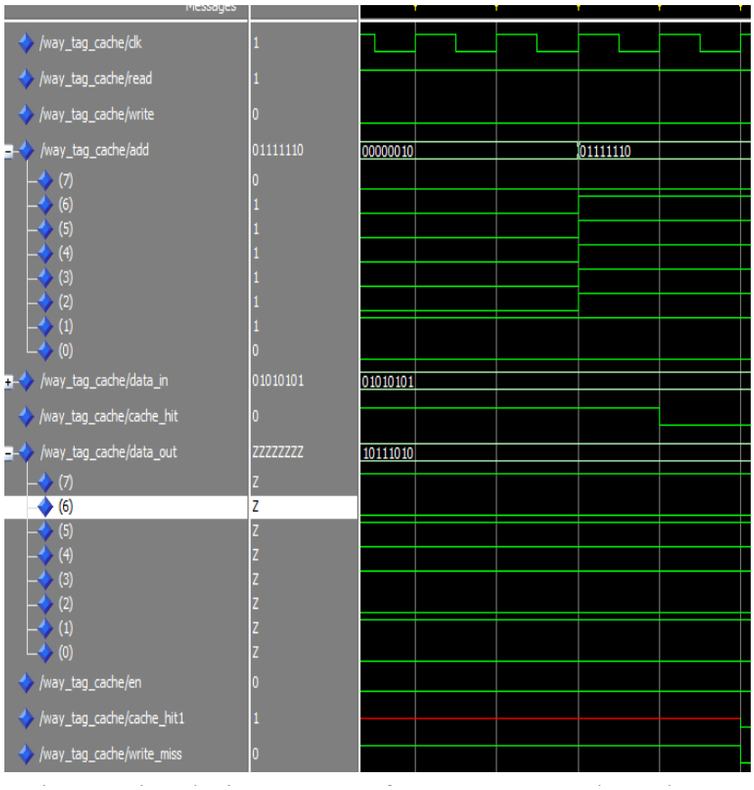

Fig. 6: Simulation Output for Way Tagged Cache

TABLE III: COMPARISON RESULTS

Parameters	Two Way Cache	Way Tagged Cache	Way Tagged Cache With CBF (Proposed)
Power (mW)	227.80	155.88	114.34
Delay (ns)	16.225	15.930	7.34
Frequency (MHz)	61.633	62.77	132.184
Power*Delay	3696.05	2483.168	839.2256
LUT RAM	5	6	4
ROM	2	2	1
Counters	1	2	1
Registers	54	77	21
Comparators	5	6	2
MUX	-	1	8
Add/Sub	1	1	-
XORs	-	-	1

VI. CONCLUSION

In this paper three kinds of cache architectures are compared such as Two-level cache, Way tagged cache, and proposed Way tagged cache with Counting Bloom Filter. From the Table III results we have concluded that the proposed Way tagged cache with CBFs has 49.80% of power lesser than the existing two-level cache system.

REFERENCES

[1] J. Dai, and L. Wang, "Way-tagged cache: An energy-efficient L2 cache architecture under write write-through policy," in *Proceedings of the 2009 International Symposium on Low Power Electronics and Design (ISLPED)*, pp. 159-164, 2009.

[2] E. Safi, A. Moshovos, and A. Veneris, "L-CBF: A low-power, fast counting bloom filter architecture," *IEEE*

Transactions on Very Large Scale Integration Systems, vol. 16, no. 6, pp. 628-638, July 2008.

[3] S. Kiruthika, R. N. Kumar, and S. Valarmathy, "Comparative analysis of 4-bit multipliers using low power 8-transistor full adder cells," *International Journal of Emerging Technology and Advanced Engineering (IJETAE),* vol. 3, no. 1, January 2013.

[4] S. G. Siddharth, M. Ramkumar, and S. Kiruthika, "Railway track scanning and surveillance robot using wireless technology," *Journal of Harmonized Research in Engineering (JOHR),* vol. 2, no. 1, pp. 194-200, 2014.

[5] S. Kiruthika, and A. V. Starbino, "Design and analysis of FIR filters using low power multiplier and full adder cells," *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE*), pp. 1-5, IEEE, 2017.

[6] S. Kiruthika, and B. Balraj, "Design of 4x4 wallace tree multiplier based on 0.12µm CMOS technology using GDI full adder," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 15 (sp. issue), pp. 3293-3300, 2018.

[7] P. Sakthi, S. Maheswari, and P. Yuvarani, "High performance vedic multiplier using compressors," *International Journal of Applied Engineering Research*, vol. 10, no. 20, pp. 16882-16886, 2015.

[8] P. Sakthi, and P. Yuvarani, "Multipliers based on Urdhva Tiryagbhyam algorithm: A survey," *Advances in Natural and Applied Sciences*, vol. 8, no. 19, pp. 100-106, 2014.

[9] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural level power analysis and optimizations," in *Proceedings of the 27th International Symposium on Computer Architecture*, pp. 83-94, Vancouver, BC, June 2000.

[10] S. Wilton, and N. Jouppi, "An enhanced access and cycle time model for on-chip caches," 1994.

[11] M. R. Stan, A. F. Tenca, and M. D. Ercegovac, "Long and fastvup/down counters," *IEEE Transactions on Computers*, vol. 47, no. 7, pp. 722-735, July 1998.