International Journal of Advanced Information Science and Technology (IJAIST) ISSN: 2319:2682
Vol.4, No.2, February 2015 DOI:10.15693/ijaist/2015.v4i2.36-40

A Bandwidth Scalable Controller Technique For Reducing Power Consumption Of Caches In Embedded Processor

B.Vidhya kumari
PGScholar/Department of VLSI Design
Sethu Institute of Technology, Pulloor.Kariapatti

Mr.S.Ramkumar M.E
Assistant professor/Department of VLSI Design
Sethu Institute of Technology, Pulloor.Kariapatti

Dr.R.Ganesan
PG Program Head/M.E-VLSI Design
Sethu Institute of Technology, Pulloor, Kariapatti

***Abstract*-This paper target the low power applications in cache memory for embedded processor using ETA technique in VHDL. The ETA technique is used to determine the destination ways of memory instructions. This method is mostly used in processors. To reduce the high power and memory consumption ,high latency and gate counts the Bandwidth Scalable Controller can be used. The proposed system consists of Bandwidth Scalable Controller efficiently performs for data array management to reduce the power consumption. The main objective of this method is used to improve the energy efficiency and reduce power consumption, latency and hardware utilizations. simulation results clearly shows that the proposed BSC unit cache achieves over 203mw power can be consumed.**

Keywords: **TLB (Translation Look aside Buffer), ETA (Early Tag Access), BSC (Bandwidth Scalable Controller),cache memory,L1 data cache.**

I.INTRODUCTION

The complexity and variety of embedded applications are constantly increasing, thus demanding systems with high computing capacities and low power consumption. To execute this applications and reduce costs, embedded systems are integrated into system-on-chip (SOC). Cache is a part of memory unit in SOC which is used to increase the performance of the system architecture in terms of power consumption and latency. The on-chip caches would consume 40% of the total chip power [1].A CPU cache is a cache used by the central processing unit (CPU) of a computer to reduce the average time to access data from the main memory. Cache is a fast and small memory, and collect the copies of data from main memory locations. Several CPUs have different independent caches, including instruction and data cache, the data cache have more hierarchy cache levels (L1, L2 etc.)

The large power dissipation causes thermal effects and performance degradation.TheL1 data cache is a physical layer and to recover a data. The L1 data cache consists of two types of array namely, tag array and data array. Tag array is used to store the address and data array is used to store the data. The set-associative caches have fewer misses than direct-mapped caches, set-associative caches and it have slower hit times. The reactive-associative cache uses way-predicting techniques, to achieve high accuracy and provides flexible associativity. The reactive associative cache employs hardware way-prediction to determine the way-number of blocks that are displaced to set-associative positions before address computation is complete. The way-prediction have I-caches and D-caches. The I-caches combined with branch prediction and D-caches do not interact with branch prediction [8].The way-halting cache is also a cache design technique and stores some lower order bits of each tag in a tag array [3].

A. Level 1 and Level 2 cache

The Level 1 cache is also known as primary cache is used for temporary storage of instructions

and data organised in blocks of 32 bytes and it is the fastest form of storage. The Level 1 cache is implemented using Static RAM (SRAM) have 16KB size [4],[5]. SRAM uses two transistors per bit and can hold data without external assistance. The second transistor controls the output known as a flip-flop.The main objective of level 2 cache is reduce data access time and to store a new accessed information. The level 2 cache is also a secondary cache, can be used to buffer a program instructions and data from memory. The aim of the Level 2 cache is to deliever a stored information to the processor without any delays. It have two sizes, 256KB or 512KB.Compared to L1 cache the level 2 cache performs lower level and slow operation.

II. RELATED WORK

The complexity and variety of embedded applications are constantly increasing, thus demanding systems with high computing capacities and low power consumption. To execute this applications and reduce costs, embedded systems are integrated into system-on-chip(SOC).Cache is a part of memory unit in SOC which is used to increase the performance of the system architecture in terms of power consumption and latency. The on-chip caches would consume 40% of the total chip power. A CPU cache is a cache used by the central processing unit (CPU) of a computer to reduce the average time to access data from the main memory. Cache is a fast and small memory, and collect the copies of data from main memory locations. Several CPUs have different independent caches, including instruction and data cache, the data cache have more hierarchy cache levels (L1, L2 etc.).

The large power dissipation causes thermal effects and performance degradation.TheL1 data cache is a physical layer and to recover a data. The L1 data cache consists of two types of array namely, tag array and data array. Tag array is used to store the address and data array is used to store the data. The set-associative caches have fewer misses than direct-mapped caches, set-associative caches and it have slower hit times. The reactive-associative cache uses way-predicting techniques, to achieve high accuracy and provides flexible associativity. Primary and Secondary cache.

The Primary cache is also known as L1 cache is used for temporary storage of instructions and data organised in blocks of 32 bytes and it is the fastest form of storage. The Level 1 cache is implemented using Static RAM (SRAM) have 16KB size. SRAM uses two transistors per bit and can hold data without external assistance. The second transistor controls the output known as a flip-flop. The main objective of level 2 cache is reduce data access time and to store a new accessed information. The secondary cache is also a L2 cache, can be used to buffer a program instructions and data from memory. The aim of the Level 2 cache is to deliever a stored information to the processor without any delays. It have two sizes, 256KB or 512KB.Compared to L1 cache the level 2 cache performs lower level and slow operation.

A Translation Look aside Buffer (TLB) is also a cache that memory management hardware used to improve virtual address translation speed. The TLB is sometimes implemented as content-addressable memory (CAM). The CAM search key is the virtual address and the search result is a physical address. The proposed architecture consists of,

I. Two-level Cache Architecture

II. Unified Cache Architecture

III. Predictive Sequential Associative Cache

III.LSQ AND L1 DATA CACHE

The conventional L1 data cache, all tag arrays and data arrays are activated simultaneously for every read/write operation and to reduce delay. The delay/latency of the L1 data cache is one clock cycle, then the latency to be higher in deeply pipelined processors. On the other hand, the tag arrays can always be finished in one cycle. The modules in cache architecture are LSQ tag arrays, LSQ data array, LSQ TLB, information buffer, way decoder, and way hit/miss decoder. The number of matches with respect to its index of the packet is called as Hit. Number of non-matching index is called as Miss. If Hit is enabled, then the way decoder is activated to deliver the data stored in data array.

LSQ tag array and LSQ data array are used to avoid the data contention with the L1 data cache (as shown in fig 1).There are two types of operations: LOOKUP and UPDATE. When the packet arrives to cache, the LSQ tag array and LSQ TLB extracts the address from the packet and searches against the address present in the LSQ tag array and LSQ TLB. This is called look up operation. If it matched, then hit is set to 1. If it is not matched, then miss is set to 1. The tag array is used to store the address and the data array is used to store the data cache. It consists of Way decoder, Information buffer and way hit/miss decoder.

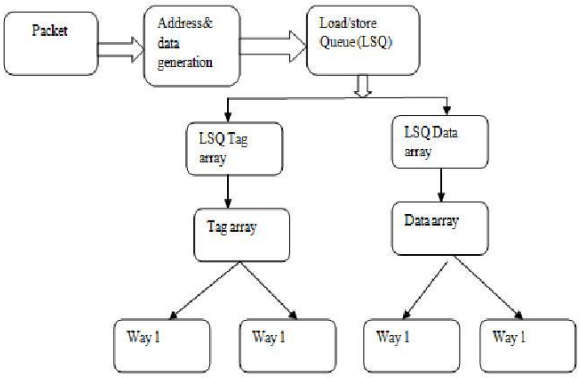

Figure 1 Pipeline of a load/store instruction between LSQ and L1 data cache (proposed)

The information buffer has separate write and read ports to support parallel write and read Operations as shown in fig 2. The write operations of the information buffer always start one clock cycle later than the corresponding write operations in the LSQ. The LSQ, LSQ tag arrays, and LSQ TLB occur simultaneously. The way information is available after the write operations in the LSQ, this information will be written into the information buffer one clock cycle later than the corresponding write operation in the LSQ. Thus, the write signal of the information buffer can be generated by delaying the write signal of the LSQ by one clock cycle. In Fig 2,the information buffer delievers a High Power Memory consumption and also produce High latency and gate counts.

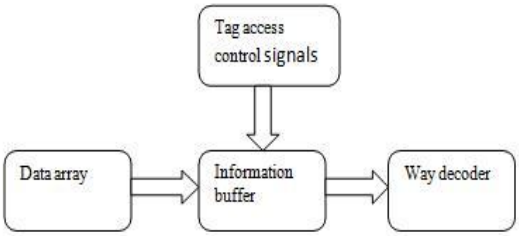

Figure 2 Information buffer

These parameters can be reduced by using bandwidth scalable controller unit. The information buffer have an additional circuitry unit that is bandwidth allocation unit as shown in Fig 5.The bandwidth allocation unit have several parameters.

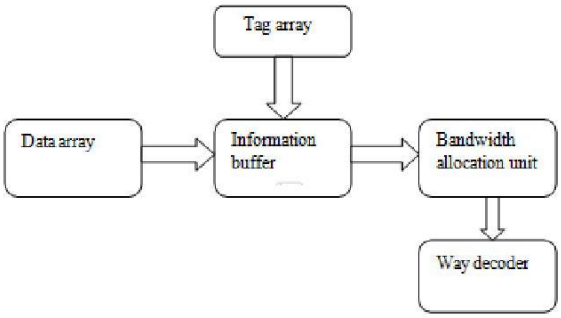

Figure 3. Proposed cache architecture

To propose an Modified bandwidth controller unit which efficiently performs for data array management to reduce the power consumption. The bandwidth controller unit have several parameters as shown in fig 4. The Bandwidth allocation unit is used to allocates the bandwidth of the packets from the information buffer.The search range prediction unit is used to determine the search area of the cache memory.The Final Search range prediction unit is used to predict the value of search range prediction unit. The BW Efficiency calculator is used to determine the efficiency of the information buffer bandwidth.

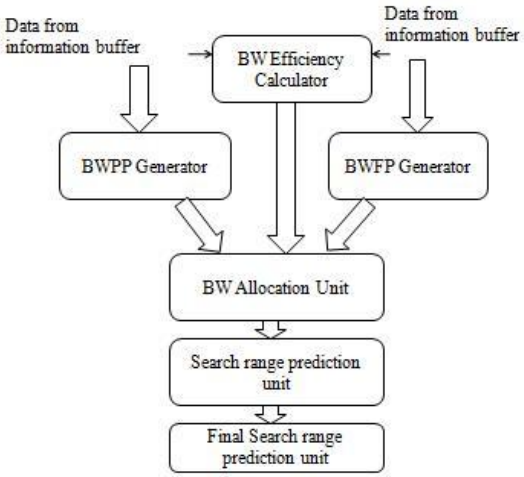

Figure 4.Bandwidth Controller Unit

The proposed enhanced BSC technique and LFSR decorrelator technique are simulated by using Xilinx ISE 9.2 simulator and modelsim 5.5e simulator. In the comparison result of the existing ETA technique and the proposed BSC and LFSR technique the

parameters are varied. To implement this technique improve energy efficiency and reduced power& memory consumption, latency, low hardware utilizations and number of gate counts. The technique is mostly used in real time processors.The list of parameter with the comparison of existing method and proposed method as shown in table 1

TABLE 1

LIST OF PARAMETER WITH THE COMPARISON OF EXISTING METHOD AND PROPOSED METHOD

Performance Parameters	Existing Method	Proposed Method
Power Consumption	203 mW	27 mW
Memory Usage	155996 KB	128992KB
Latency	5.077 ns	2.556ns
Gate Counts	1728	1456

The Existing and proposed method the parameters to be compared as shown in fig 5.

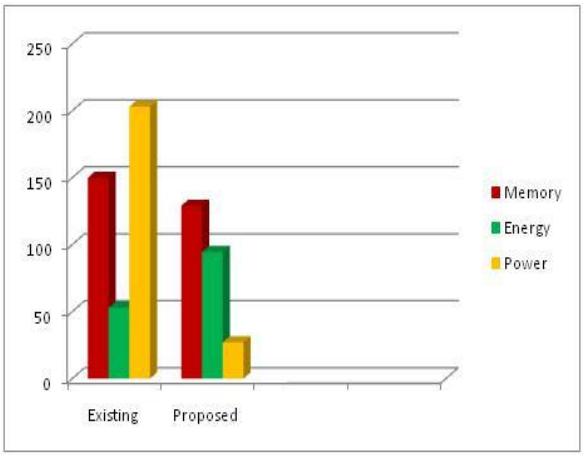

Figure 5. Performance analysis

III.RESULTS AND DISCUSSION

A. Simulation Result

Figure 6 shows that simulation result of BSC. The memory have number of bits like 000,001,….111.If we send bit 001,it is enabled in memory then remaining bits are disabled. so, the power can be reduced .

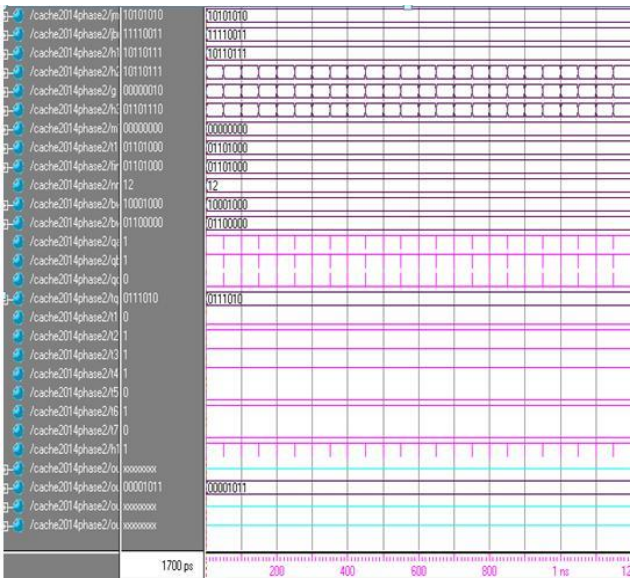

Figure 7. Simulation results

IV.CONCLUSION

The Bandwidth Scalable Controller Unit can be designed in this project which efficiently performs for data array management to reduce power consumption. Simulation presented clearly shows that the 52.8% energy reduction on average in the data cache and TLB. The power consumption was measured to be 27 mw.

REFERENCES

1. B. Batson and T. Vijaykumar (2001) "Reactive associative caches," in *Proc.Int.Conf. Parallel Archit. Compil.*
2. D. Brooks, V. Tiwari, and M. Martonosi, "Wattch (2000): A framework for architectural-level power analysis and optimizations," in *Proc. Int. Symp.Comput. Archit.*, pp. 83–94.
3. B. Calder and D. Grunwald (1996), "Next cache line and set prediction," in*Proc. IEEE Symp. High-Perform.Comput. Archit.* pp. 49–60.
4. J. Dai and L. Wang (2013), "An energy-efficient L2 cache architecture usingway tag information under write-

through policy," *IEEE Trans. VeryLarge Scale Integr. (VLSI) Syst.*, vol. 21, no. 1, pp. 102–112.
5. A. Hasegawa, I. Kawasaki, K. Yamada, S. Yoshioka, S. Kawasaki, and P. Biswas (1995) "SH3: High code density, low power," *IEEE Micro*, vol. 15,no. 6, pp. 11–19
6. K. Inoue, T. Ishihara, and K. Murakami (1999), "Way-predicting set-associative cache for high performance and low energy consumption," *Int. Symp.Low Power Electron. Design*, pp. 273–275.
7. T.Ishihara and F. Fallah (2005), "A way memoization technique for reducing power consumption of caches in application specific integrated processors," in *Proc. Design Autom. Test Eur*,pp. 358–363.
8. Jianwei Dai, Menglong Guan, and Lei Wang (2014), *Senior Member,* IEEE Transactions On Very Large Scale Integration (VLSI) Systems, VOL. 22, NO. 2, "Exploiting Early Tag Access for Reducing L1 Data Cache Energy in Embedded Processors"
9. R. Min, W. Jone, and Y. Hu, "Location cache: A low-power L2 cache system," in Proc. Int. Symp. Low Power Electron. Design, 2004, pp.120–125.
10. J. Montanaro, R. T. Witek, K. Anne, A. J. Black, E. M. Cooper, D. W.Dobberpuhl, P. M. Donahue, J. Eno, W. Hoeppner, D. Kruckemyer, T. H.Lee, P. C. M. Lin, L. Madden, D. Murray, M. H. Pearce, S. Santhanam,K. J. Snyder, R. Stehpany, and S. C. Thierauf (1996), "A 160-MHz 32-b 0.5-W CMOS RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 31,no. 11, pp. 1703–1714.
11. B. Moyer, and D. Cermak (2000), "A Low power unified cache architecture providing power and performance flexibility," in *Proc. Int.Symp. Low Power Electron. Design*, pp. 241–243.
12. S. Santhanam, A. J. Baum, D. Bertucci, M. Braganza, K. Broch,T. Broch, J. Burnette, E. Chang, C. Kwong-Tak, D. Dobberpuhl,P. Donahue, J. Grodstein, K. Insung, D. Murray, M. Pearce, A. Silveria,D. Souydalay, A. Spink, R. Stepanian, A. Varadharajan, V. R.vanKaenel, and R. Wen (1998)."A low-cost, 300-MHz, RISC CPU with attached media processor," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1829–1838
13. C. Su and A. Despain, "Cache design tradeoffs for power and performance optimization: A case study," in *Proc. Int. Symp. Low Power Electron. Design, 1997, pp. 63–68.*
14. C. Zhang, F. Vahid, and W. Najjar (2003)"A highly-configurable cache architecture for embedded systems," in *Proc. 30th Annu. Int. Symp. Comput.Archit.*, pp. 136–146.

Author's Profile

Mr.S.Ramkumar obtained hisBachelors in Engineering in Electronics and Communication from Anna University. And **M.E.,** degree (VLSI Design)from Anna University Coimbatore on 2011. His Areas of interest are Testing of VLSI circuits,VLSI security and networking.

Dr.R.Ganesan received his **B.E.,** in Instrumentation and Control Engineering from Arulmigu Kalasalingam College Of Engineering and **M.E.,** (Instrumentation) from Madras Institute of Technology in the year 1991 and 1999 respectively. He has completed his **Ph.D** from Anna University, Chennai, India in 2010. He is presently working as Professor and head in the department of M.E-VLSI Design at Sethu Institute of Technology, India. He has published more than 25 research papers in the National & International Journals/ Conferences. His research interests are VLSI design, Image Processing, Neural Networks and Genetic algorithms.

B.VidhyaKumari received her **B.E.,** degree in Electronics and Communication Engineering from Sree Sowdambika College Of Engineering Anna University Thirunelveli, in 2013.Pursuing **M.E.,** degree in VLSI Design from Sethu Institute of Technology, Anna University Chennai, India. Her research interests are Low power VLSI design and Neural Networks.