VOL. 10, NO. 11, JUNE 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DESIGN OF CACHE MEMORY MAPPING TECHNIQUES FOR LOW POWER PROCESSOR

R. Ramya[1] and T. Ravi[2]
[1]VLSI Design, Sathyabama University, Chennai, India
[2]Department of Electronics and Communication Engineering, Sathyabama University, Chennai, India
E-Mail: Ramesh.ramya3@gmail.com

ABSTRACT

The use of cache memory makes the processing of access in a faster rate. The main purpose of cache memory is to give faster memory access by which the data read should be fast and at the same period d provide less expensive and types of semiconductor memories which are of large memory size. There is correspondingly main memory which is large but slow together with a smaller as well faster cache memory. The cache memory contains a copy of instruction from main memory. The processor when it needs to read from or write to in the main memory locations, it first checks whether the cache memory contains the required data. This paper presents a two level cache in which the splitting of cache level is used by which faster access time and low power consumption can be achieved. The main focus of this project is reduced access time and power consumption.

Keywords: direct mapping techniques, fully associative mapping, set associative mapping technique.

1. INTRODUCTION

Cache systems are on-chip memory elements such that data that is needed can be stored. The miss rate that occurred in cache memory can be found by the controller. When the data that is required by microprocessor is found then the data hit is said to occur in the cache. The common usage of storing data on cache is to achieve faster data reading time but energy consumption is one of the drawbacks of on-chip. As the cache memory moves away from the CPU, the access time and the size of the cache memory storage unit increase as well. Cache memory is an additional and fast memory unit that has to be placed between the processing unit and the physical memory. The mostly used instructions and data, where this information is needed to be accessed again are stored in cache. The physical memory and external disk storage devices can be accessed faster by the internal registers and cache which are located near to CPU. Cache which is accessed faster can be considered to be more power efficient. To address this challenge and also sustainable computing goals are met by several energy efficient techniques that are proposed for the cache architecture. The static and dynamic power consumption will lead to the total power consumption. If so, the processor will reads from or writes immediately to the cache, which is much smaller and faster to read and write the data that is required. Three independent caches are used in the modern desktop and server CPUs, such as an instruction cache to speed up executable fetch from instruction, a data cache is used to speed up data fetch and store, transistion look aside table used to increase the virtual-to-physical address translation. The two levels of memory are used to reduce average access time works in principle, during the course of execution of a program. One obvious advantage of the logical cache is that cache access speed is faster than a physical cache, because this cache memory will respond before the MMU performs an address translation. The disadvantage is that the same virtual address in two different applications refers to two different physical addresses.

Modern processors employ large LLCs and their size is expected to grow even further. In fact, Rogers *et al*., [1] have shown that due to bandwidth-limitations, SRAM-based caches may occupy 90% of the chip-area in upcoming fourth CMOS generation. Thus, With each CMOS (complementary metal oxide semiconductor) technology generation will lead to considerable amount of leakage power [2 3], due to increasing size of LLCs, along with large leakage energy consumption of SRAM devices, the energy consumption by the LLCs is one of the important factor for processor energy consumption [4]. SMART technique uses cache reconfiguration to dynamically tune cache active fraction to reduce leakage energy consumption, while choosing reasonably small write latency [5]. Although the techniques designed to improve performance are also likely to conserve energy, in this survey we focused on only those techniques which aim to optimize energy efficiency and have been shown to improve energy efficiency. In this paper we have 3section; Section 1 briefly describes the different optimization parameters that cache consist of during cache design. Section 2 describes about the power efficient cache which leads to low power cache and Section 3 describes different cache mapping techniques.

2. CACHE MAPPING TECHNIQUES

The Cache Mapping technique is of three types such as direct mapping, Fully Associative technique and set Associative technique and these techniques are discussed below,

2.1 Direct mapping

In a direct mapped cache, each block has only one place that it can go. Thus when the CPU needs a certain block from the main memory, the cache memory is the only one place that it could possibly reside. The needed block can also be fetched from the lower level of memory if it is not there in the cache memory. In order to find where the block resides in the cache, the block frame address is divided by the number of blocks in the cache.

The advantage of direct mapping technique is that the mapping function that is followed in this method is of trivial. If the cache memory is needed for the data to be stored then the cache line will be immediately available and the access time to read the data from the cache line of the cache memory will be faster. In optimum case the sequential processing will be followed. The drawback of the direct mapping is, if cache not yet full line may be replaced and replacing scheme may be inadequate. In worst cases referencing always contains same line number.

2.2 Fully associative mapping

In a fully associative cache, the block can be placed anywhere in the cache since there are no restrictions as to where it has to be placed. When the CPU needs a certain memory block, the cache must be checked for each block that resides in it, to determine if the required information is present in the cache. A block is only evicted from a fully associative cache if the cache is full.

The advantage of fully associative cache is that the trivial method of mapping function is followed. When the data has to be written in the cache only if the cache is full then the cache line will be replaced. In optimum case the access of the data will be wide spread access. The disadvantage of fully associative cache is, whenever the data has to be searched it checks all the memory cell of the cache. The extended tag field is used and additional tag counter is required. Replacing scheme that is used in this type of mapping is inadequate. In worst case the area are slightly larger than the cache that are done sequentially.

2.3 Set associative mapping

In a set associative cache, certain set of places are allocated for each memory block. A set consists of a group of two or more blocks in the cache. When a block is placed into cache memory from the main memory, first the block is mapped to a set, and within that set the block is free to go anywhere. This method of set associative mapping combines the likes of both direct mapped and fully associative mapping in that the block is directly mapped to a set, and then is fully associative within that set. In order to determine the set that the block should be placed in, the block frame address is divided (modulo) by the number of sets that are in the cache.

A cache is said to be n-way set associative if there are n blocks in each set. The power consumed by the cache memory which is of set associative is more when compared to other mapping techniques like direct mapping, fully associative. The faster access can be achieved if less set associative is done where miss rate can be achieved only if more set are associated where power consumption also increases. The advantage of set Associative mapping technique is cheaper when compared to the fully associative mapping. Lower miss ratio is achieved when compared to the direct mapping cache at the cost of access time.

3. EXISTING SYSTEM

The Existing cache memory consists of two level of cache level. The data cache memory can be extended to different levels by which it has hierarial levels of cache memory. when the levels of cache memory increases the delay or the miss penalty that is associated with the cache increases by which most of the processor cache memory are of two levels. The first level cache memory consist of direct mapping technique by which the faster access time can be achieved because in direct mapping it has row decoder and column decoder by which the exact memory cell is choosen but the miss rate that may occur in direct mapping. The second level cache memory consist of fully associative mapping techniques by which the data are accessed but the speed of this mapping technique is less when compared to direct mapping but the occurance of the miss rate is less. The simulation result of existing methodology is given in Figure-2.

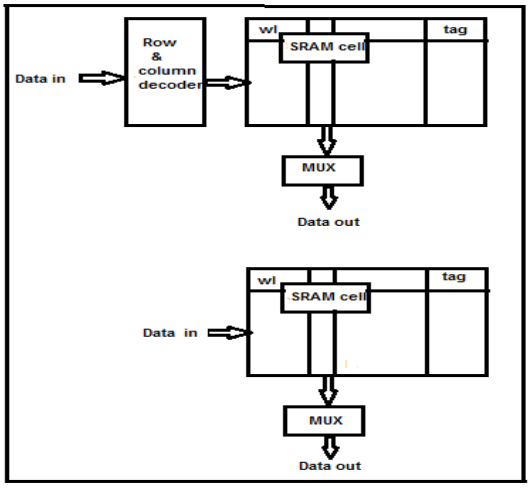

Figure-1. The Existing two level cache system.

Simulation result

Case-1

In this case, the L1 cache level is active and L2 cache level is inactive. When the word line of L1 cache is accessed then its corresponding output is shown in Figure-2 and similarly when the word line of L2 cache is not accessed then its corresponding output is shown in Figure-3.

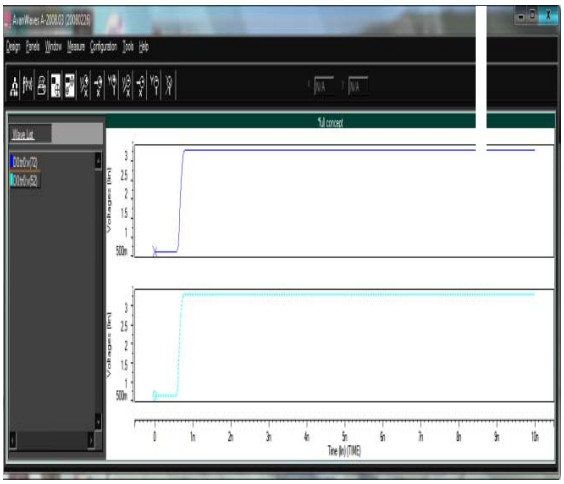

Figure-2. Simulation output when L1 cache is accessed.

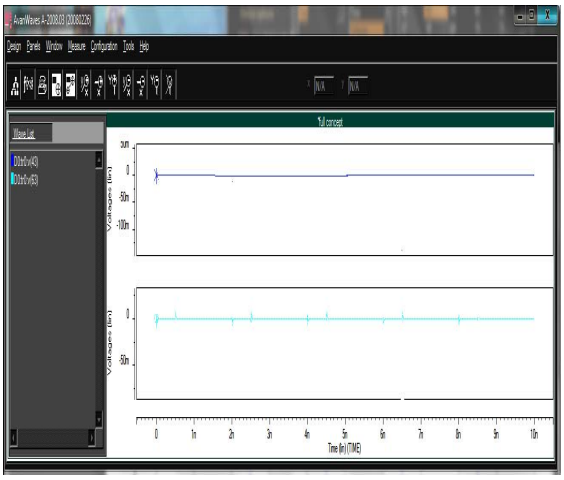

Figure-3. Simulation output when L2 cache is not accessed.

Case-2

In this case, the L1 cache level is inactive and L2 cache level is active. When the word line of L1 cache is not accessed then its corresponding output is shown in Figure-4 and similarly when the word line of L2 cache is accessed then its corresponding output is shown in Figure-5.

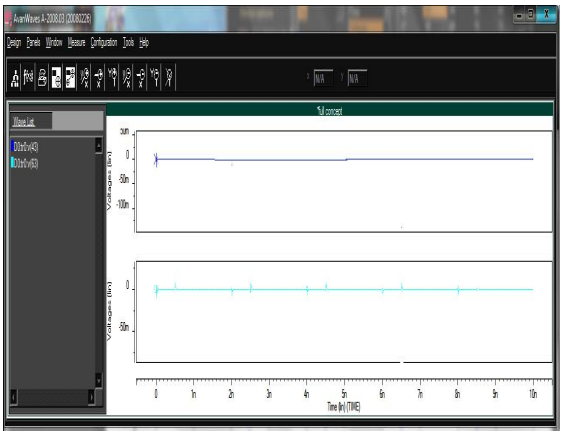

Figure-4. Simulation output when L1 cache is not accessed.

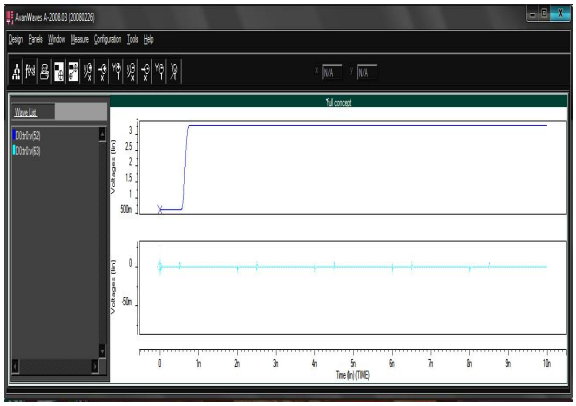

Figure-5. Simulation output when L2 cache is accessed.

4. PROPOSED WORK

The proposed methodology consist of two level cache memory by which the second level cache memory which is of fully associative can be splited into equally two levels which is of direct mapping and fully associative mapping technique by using this dividing of a single cache level which will lead to faster access and low power consumption when compared to the existing system by which the power efficiency as well the performance of the cache can be achieved as shown in Figure-6.

The cache should achieve a less delay by which the cache system will have a quick access as it will check every memory address. The total power consumption of any processors is increasing tremendously and it reaches the “powerwall” imposed by thermal limitations of cooling solutions and power delivery. Inorder to reduce the power consumption by the cache memory we can choose the mapping technique which has less power consumption. To reduce the miss rate victim cache which is of fully associative is placed at the last by which the miss rate can also be reduced mainly reduced delay, less power consumption is achieved in this work.

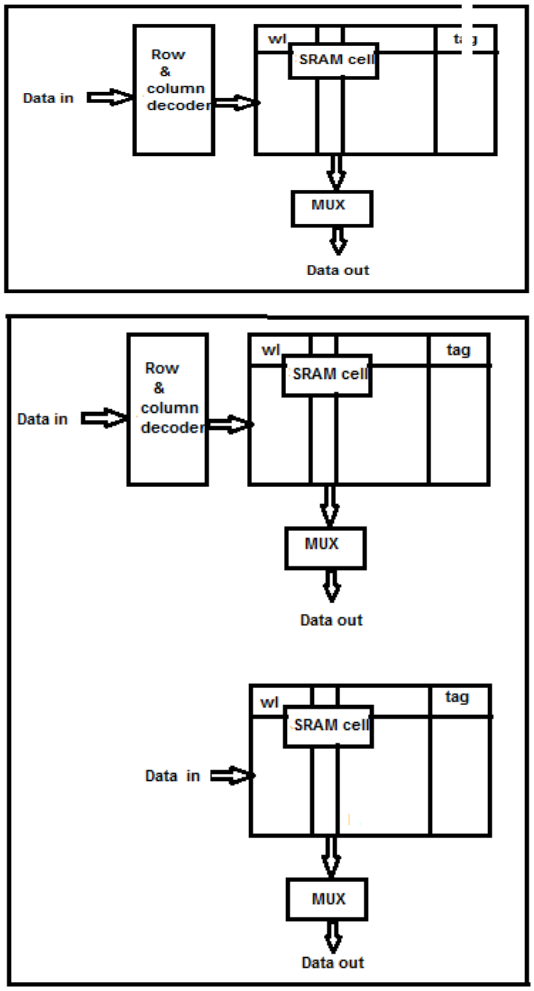

Figure-6. The proposed cache system.

Simulation result

Case-1

In this case, the L1 cache level is active and L2 cache level is non active. When the word line of L1 cache is accessed then its corresponding output is shown in Figure-7 and similarly when the word line of L2 cache is not accessed then its corresponding output is shown in Figure-8.

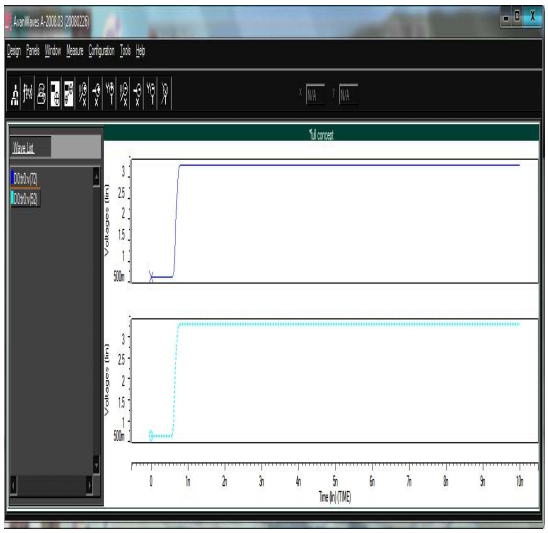

Figure-7. Simulation output when L1 cache is accessed.

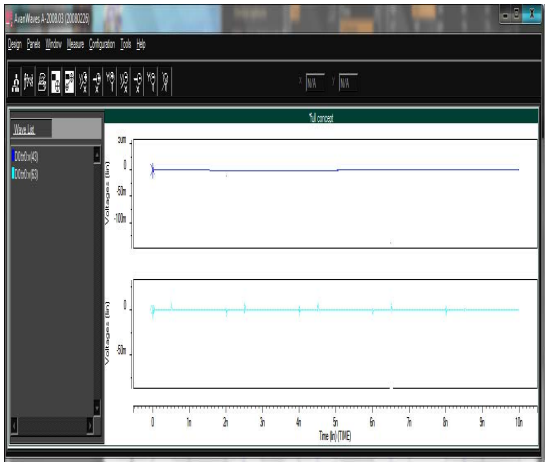

Figure-8. Simulation output when L2 cache is not accessed.

Case-2

In this case, the L1 cache level is not active and L2 cache level is active. When the word line of L1 cache is not accessed then its corresponding output is shown in Figure-9 and similarly when the word line of L2 cache is accessed then its corresponding output is shown in Figure-10.

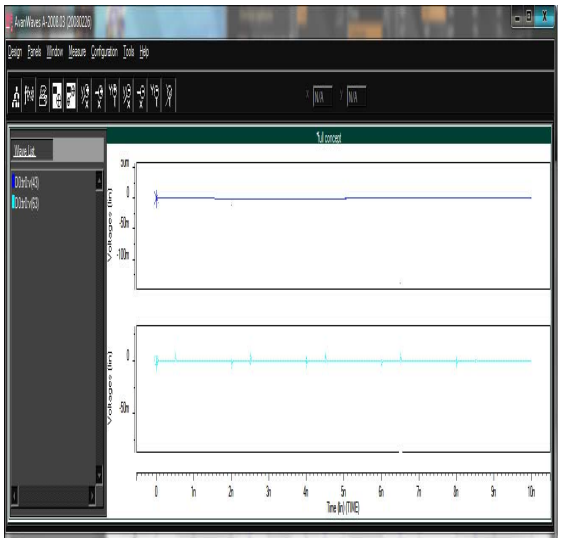

Figure-9. Simulation output when L1 cache is not accessed.

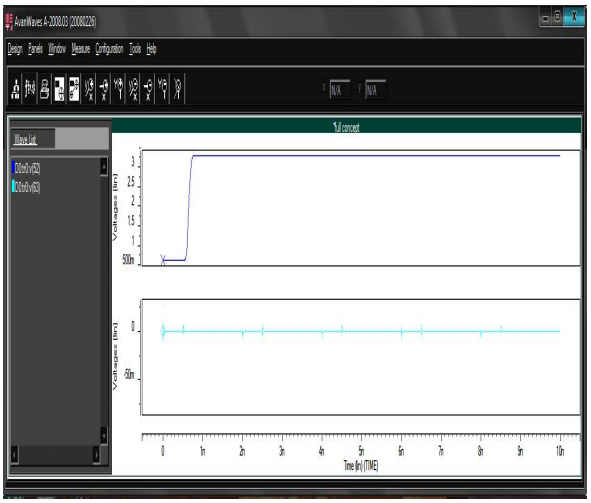

Figure-10. Simulation output when L2 cache is accessed.

Table-1. Performance analysis of existing cache.

	Existing system	
Parameter	180nm	130nm
Average power	1.3044mW	4.8034mW
Average delay	3.3162ps	9.6750ps
Power delay product	4.3256fJ	46.4773fJ

Table-2. Performance Analysis of proposed cache.

	Proposed system	
Parameter	180nm	130nm
Average power	1.3042mW	4.7816mW
Average delay	1.6207ps	9.3500ps
Power delay product	2.1137fJ	44.7830fJ

5. CONCLUSIONS

Caches memory can be implemented in different ways, but the basic concepts behind the cache technique remain indentical. A challenge in cache design is to conform that the required data as well as the instructions are presented in the cache line. Overall power consumption of any processors is increasing tremendously and it reaches the "powerwall" imposed by thermal limitations of cooling solutions and power delivery. Thus, using technological scaling higher performance can be achieved and managing the power consumption of processors has become a vital necessity. In this paper, mainly 3.36% reduction in delay and 0.45% reduction in power consumption and 3.65% reduction in power delay product is achieved at the cost of miss rate.

REFERENCES

[1] B.M Rogers, A. Krishna *et al*. 2009. Scaling the bandwidth wall: challenges in and avenues for CMP scaling. ACM SIGARCH computer Architecture.

[2] S. Borkar *et al*. 1999. Design challenges of technology scaling. IEEE.

[3] G. Gammie A. *et al.* 2010. Smartreflex power and performance management technologies for 90nm, 65nm, 45nm mobile Application processors. IEEE.

[4] Sparsh Mittal. 2013. A Survey of Architectural techniques for improving cache power efficiency. Elsevier sustainable computing: Informatics and systems.

[5] SparshMittal. 2013. A cache energy optimization technique for STT-RAM last level cache. CS.AR.

[6] Ching-Long *et al.* Cache design Trade-offs for power and performance optimization.

[7] H.Dybdahl *et al*. 2005. Destructive read in embedded DRAM impact on power consumption. Journal of embedded computing.

[8] Norman P. Jouppi. 1990. Improving direct mapped cache performance by the addition of a small fully associative cache and prefetch buffers.

[9] Bradford M. Beckmann and David A. wood. 2004. Managing wire delay in large chip multiprocessor caches. In International symposium of Micro architecture.

[10] C.L. Hwang, T. Kirihata *et al*. 2002. A 2.9ns random Access cycle embedded DRAM with a destructive read Architecture. In IEEE symposium.

[11] G.Kirsch. 2003. Parallel and distributed processing. On International symposium.

[12] Krisztianflaunter, nam sung Kim et al. 2002. Drowsy caches: simple techniques for reducing leakage power. International symposium on computer Architecture.

[13] Michael powell *et al*. 200. Gatedvdd: a circuit technique to reduce leakage in deep-submicron cache memories. International Symposium on low power electronics and design.

[14] Johnson Kin,Munisha Gupta *et al*. 1997. The filter cache: an energy efficient memory structure. International symposium on Micro architecture.

[15] Atsushi Hasegawa, Ikuya Kawasaki *et al*. 1995. High code density low power. IEEE Micro.

[16] Ravi. T, Kannan. V. 2012. Design and Analysis of Low power CNTFET TSPC D Flip-Flop based shift Registers. Applied Mechanics and Materials journal.

[17] Ravi T *et al*. 2013. Ultra Low Power Single Edge Triggered Delay Flip Flop Based Shift Registers uses 10-Nanometer Carbon Nano Tube Field Effect Transistor. In American Journal of Applied Sciences. 10(12): 1509-1520.