International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Modified Networks-in-Cache for Embedded Processor

Soly Susan Joseph[1], Anusree L. S.[2]

[1]P G Scholar, VLSI and Embedded Systems, Department of ECE, T K M Institute of Technology, Kollam, India

[2]Assistant Professor, Department of ECE, T K M Institute of Technology, Kollam, India

Abstract: ***A new cache architecture which utilizes the advantage of the tiling of cache Non Uniform Cache Architecture (NUCA) and the latency in accessing data from the tiles can be reduced by modified tag matching technique. Based on the fact that a memory system is said to be perfect if it can supply immediately any data that the CPU requests. The cache architectures pay an important role in how fast the data can be accessed from the cache by the processor. The proposed cache architecture consist of small and low latency tiles placed between different cache levels so as to reduce the inter cache latency gap between first level cache and secondary cache. The interconnection and routing of data through these tiles are done by three networks which expertise in separate cache operation so as to improve the performance. The tag matching technique presented in this brief for data protected with error correcting code is to reduce the latency and complexity. The practical error correcting code word is usually presented in a systematic form in which the data and parity parts are completely separated from each other so the advanced tag matching technique parallelizes the comparison of both parts. The incoming data matches the stored data only when a certain number of erroneous bits are corrected and a butterfly-formed weight accumulator is used for efficient Hamming Distance computing. This design flow can be attained in Verilog language is used for coding and simulated by using Model-Sim.***

Keywords: Non Uniform Cache Architecture, tag matching, error-correcting codes (ECC), systematic codes, Hamming distance.

1. Introduction

A memory system is said to be perfect if it can supply immediately any data that the CPU requests. But an ideal memory is not practically possible. Therefore, in order to cope up with the speed of the CPU an economical solution is a memory hierarchy that is organized into several levels in which each level is smaller and hence faster than the next levels. Cache is a smaller version of main memory which is placed between the CPU and the main memory so that it can be easily accessed by the processor. The ever enlarging sizes of on chip caches and the growing control of wire delays makes it necessary to cause significant changes in cache architectures or cache hierarchy design methodologies. In the traditional cache architecture which is called Uniform Cache Architecture or UCA, caches are typically partitioned into many banks and the cache access latencies are determined by the latency for the furthest cache bank. But the cache sizes and the latency difference between nearest and the farthest banks increases makes the UCA not preferred as a scalable model. Also, forcing a size and latency increment in secondary caches to reduce the latency gap between the processor and main memory causes the increase in inter-cache latency gap between secondary caches and L1 caches. The solution to reduce these gaps is either to reduce the latency of secondary caches or increasing the size of first level cache [7].

In NUCA or Non Uniform Cache Architecture, the banks are connected with an interconnect fabric and the access time for a block depends on the delays taken in traversing the network path from the cache controller to the bank that contains the block. This causes non-uniform cache access times in accessing data from the banks so that the banks closer to the CPU can be accessed faster than that residing in banks which lies away from the CPU [4][5][7]. The cache architecture used in this brief is a Non Uniform Cache Architecture called Light Power Non Uniform Cache Architecture in which the L1 cache is surrounded by a set of small tiles connected by three on chip networks which are specialized for performing three cache operations: search, transport and replacement [3]. Also, in order to reduce the number of tile access and to increase the performance two additional techniques are incorporated in LNUCA called Miss Wave Stopping and Sectoring. [2]. Recently computers use ECC codes for storing data in the memory for protection and reliability. The tag matching technique presented in this brief for data protected with error correcting code is to reduce the latency and complexity. The practical error correcting code word is usually presented in a systematic form in which the data and parity parts are completely separated from each other so the advanced tag matching technique parallelizes the comparison of both parts. The incoming data matches the stored data only when a certain number of erroneous bits are corrected and a butterfly-formed weight accumulator is used for efficient Hamming Distance computing [1].

2. Related Works

Cache memories are small, high-speed buffer memories so that information stored in cache memories may be accessed in much less time than that located in main memory. Thus the CPU with cache memories needs to spend far less time waiting for the data. Whenever the CPU wants some data it first checks whether the data is present in the cache or not. If the data is present in the cache then it takes the data from the cache and if the searched data is absent in the cache then it checks in the main memory to retrieve the data. Hence the greater number of requests served from the cache; the faster the overall system performance becomes [6].

There were a number of Non Uniform Cache Architectures introduced before LPNUCA which includes S-NUCA-1, S-NUCA-2, DNUCA etc [5][7]. The initial aim was to split the large cache banks into small ones with each banks uses private, two-way pipe lined transmission channel to serve requests and this architecture is called static NUCA-1 (S-NUCA-1). Then the focus was to reduce the impact of wire delay in LLC so the global wires are replaced with conventional 2-D mesh and worm hole routers forming static NUCA-2 (S-NUCA-2). The modifications on cache architectures were further leads to introduction of Dynamic NUCA (DNUCA) in which interbank block migration is possible. Most of the advancements are based on LLCs, improving router performance, topology etc. Most of the designs result in increase in bank size and thus corresponding routing delay for improving performance in the large LLC environment. Among various NUCA architectures, Light NUCA (LNUCA) [3] tries to close these latency gaps between secondary caches and L1 caches by enlarging the lower level cache which can be accessed by the processor at low latencies and this architecture focuses on improving topologies and data transfer. In an LNUCA cache, the L1 cache is surrounded by a set of small tiles connected by three on chip networks which are specialized for performing three cache operations: search, transport and replacement. Also, in order to reduce the number of tile access and to increase the performance two additional techniques are incorporated in LNUCA called Miss Wave Stopping and Sectoring to form Light Power NUCA (LP-NUCA) which is a specialization of Light NUCA for high performance embedded processors [2].

3. Methodology

The basic block diagram shown in fig.1 consist of CPU, tiled cache memory with a modified tag matching technique and main memory along with respective memory controller. CPU decides whether to perform read or write operation and according to which it issues either READ or WRITE requests respectively along with an address from which the data can be read or to which a data can be write. Main memory is considered as a Random Access Memory (RAM) which is of larger size than the cache memory. It stores all the data needed for the CPU in specific memory locations. The main purpose of the memory controller is to check the address given by the CPU is present in the memory and the data in the address is given to the CPU in read operation and data written to this address in write operation. Cache controller decides to which cache line the data from memory to be loaded and the cache memory architecture used is LP-NUCA [2] rather than conventional cache architecture to which a tag modification is introduced.

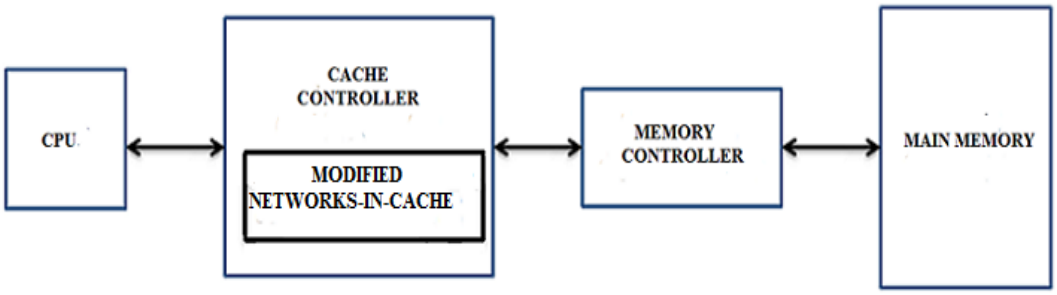


Figure 1: Block diagram incorporated with modified networks-in- cache.

Using the LP-NUCA architecture the main intention is to expand the lower level cache without overwhelming the allowable cache size so as to reduce the latency gap between secondary caches and L1 cache level. Therefore a fabric of very small tiles is placed between L1 and last level cache which not only act as the L1 cache capacity extension mechanism but also keep most recently evicted blocks close to L1 as shown in fig.2. Most of the previous NUCA designs depend on conventional routers for data movement. The main feature of LP-NUCA is that it exploits the traffic pattern within the cache using three specialized networks-in-cache which takes advantage of availability of on chip wires to minimize the network overhead. Also two techniques called Miss Wave Stopping and Sectoring are used in this architecture to reduce the number of tile access and to improve the performance [2].

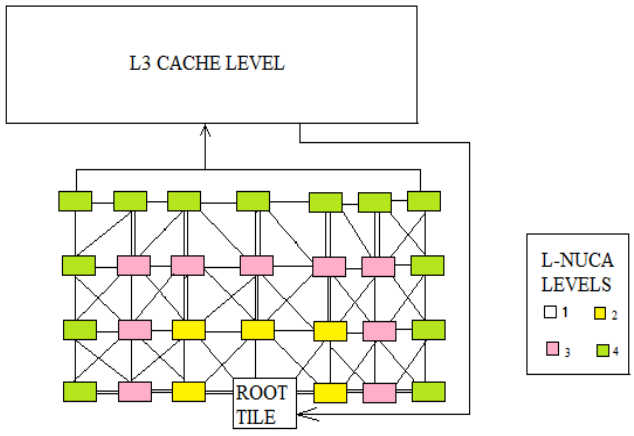
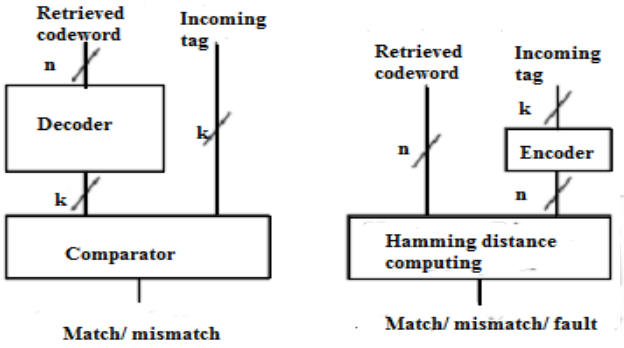


Figure 2: Tiled cache architecture

3.1 Modified Tag Matching

In order to check whether a data is present in a cache, the address of data in memory is compared to all cache tags in the same set that might contain the address. In LP-NUCA, the root tile, child tiles and other cache levels and external

memory carries data so for searching data it is necessary to check in all these storage units. Usually in a memory structure which is protected with error correcting codes, the data is encoded first and then the entire code word including the error correcting code check bits are written to the memory array. For getting original data, the code loaded from the memory has to be decoded and corrected if errors are present. Data comparison circuit which compares each tag with the given address is usually lies in the critical path of the components in the pipeline stage since comparison result decides which way in the set to load the data from the data array.



(a) (b)

Figure 3: (a) Tag matching with decoding (b) Tag matching without decoding

The comparison checks if the retrieved code word lies in the error correctable range of the code word with respect to incoming data by using Hamming Distance computing. Here the characteristics of systematic codes are considered and a low complexity processing element that computes the Hamming distance faster is used [1].

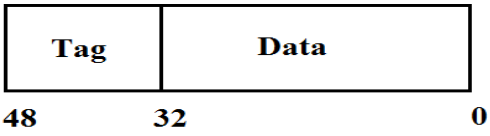


Figure 4: Tag and data size used in proposed networks-in-cache.

4. Results and Discussion

The modules are modeled using Verilog HDL and the simulation of the design is performed using Modelsim SE 6.5E to verify the functionality of the design. Main memories is basically designed as RAM memory with the controller performing read and write operations.

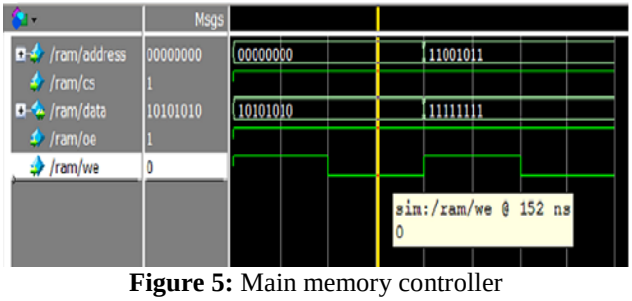


Figure 5: Main memory controller

Assume for read/write operation chip select, cs=1. For memory write, write enable, we=1 then an address to which data to be written and the data to be stored are given. For memory read, output enable, oe=1 and write enable, we=0. An address from which the data to be read is given. Output obtained is the data read from the given address is shown in fig.5.

The inputs to the networks-in-cache are clock (clk), control signal (ctrl) and search address. The root tile, child tiles and external memory are loaded with some data. When the control signal is 0001, the data stored in corresponding address given as ”search address” is obtained as output by performing tag matching in root tile and in child tiles and then in the external memory shown in fig.6. The output is obtained as ”matchdata”.

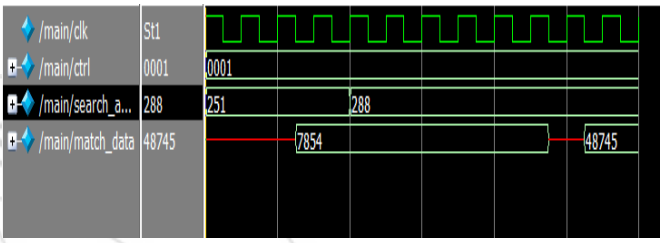


Figure 6: Networks-in-cache

5. Conclusion

It is difficult for a memory to catch up with speed of a processor. But to increase the performance the memory has to respond to the processor as quickly as the processor sends a request. Caches are used by the processor for easy accessing of data than from an external memory quickly due to their small size and proximity to the processor. Even though the hit rate of caches increases with increasing the size of cache, the latency in accessing data is also increases with cache size. Therefore, it is very difficult to model a cache architecture which compensates both the size and latency. As the secondary cache increases the latency gap between lower level caches and secondary caches also increases.

LP-NUCA consists of three networks for searching, transporting and replacement of data through the tiles forming the networks. Miss Wave Stopping and Sectoring are the two techniques using in LP-NUCA with the intention of reducing the number of tile access in each network. The use of networks of small tiles paves a way to overcome the above mentioned difficulty by expanding the size of level cache without increasing the latency in accessing a data. This makes LP-NUCA one of the prominent cache architecture.

From the result in fig.6, it can be seen that the output obtained with a considerable time delay after giving the inputs. One of the reasons for causing output delay is due to the delay in data comparison, in this case tag matching. The solution is to make low latency tag matching circuitry. The data in the memory is stored in an encoded form so the earlier technique for data comparison is to first decode the encoded data and then compare it with incoming tag. Since decoding is always complex than encoding, it is better to eliminate the decoding by encoding. The modified tag

matching can be done by encoding the incoming tag and compares it with retrieved code word using error correcting codes.

References

[1] Byeong Yong Kong, Jihyuck Jo, Hyewon Jeong, Mina Hwang, Soyoung Cha, Bongjin Kim and In-Cheol Park, " Low Complexity Low Latency Architecture for Matching of Data Encoded With Hard Systematic Error-Correcting Codes",IEEE Transactions on Very Large Scale Integration(VLSI) Systems, vol.22, no.7, July 2014.
[2] Daro Surez Gracia, Giorgos Dimitrakopoulos, Teresa Monreal Arnal, Manolis G. H. Katevenis, Vctor Vials Yfera, LP-NUCA: Networks-in-Cache for High-Performance Low-Power Embedded Processors, IEEE Transactions on Very Large Scale Integration(VLSI) Systems, vol.20, no.8, August 2012.
[3] D. Surez, T. Monreal, F. Vallejo, R. Beivide, and V. Vials, Light NUCA: A proposal for bridging the inter-cache latency gap, in Proc.12th Des., Autom. Test in Europe Conf. and Exhibition (DATE09), 2009, pp. 530535.
[4] Y. Jin, E. J. Kim, and K. H. Yum, A domain-specific on-chip network design for large scale cache systems, in Proc. 13th Int. Symp. High Performance Comput. Architecture (HPCA07), 2007, pp. 318327
[5] P. Foglia, D. Mangano, and C. A. Prete, A nuca model for embedded systems cache design, in Proc. 3rd Workshop Embedded Syst. Real- Time Multimedia (ESTImedia05),2005, pp. 4146.
[6] Gene Cooperman. "Cache Basics". 2003.
[7] C. Kim, D. Burger, and S. W. Keckler, An adaptive, non-uniform cache structure for wire-delay dominated on-chip caches, in Proc. 10th Int. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS02), 2002, pp.211222.