Annals of DAAAM for 2010 & Proceedings of the 21st International DAAAM Symposium, Volume 21, No. 1, ISSN 1726-9679
ISBN 978-3-901509-73-5, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2010
Make Harmony Between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals of DAAAM International

DEPENDABILITY ASPECTS REGARDING THE DESIGN OF A CACHE MEMORY

NOVAC, O[vidiu]; NOVAC, M[ihaela] C[ornelia]; VLADU, E[caterina] E[milia]; VARI - KAKAS, S[tefan]; MANG, G[erda] E[rica] & INDRIE, L[iliana]

Abstract: *In this paper, we will present some dependability aspects regarding the design of a cache memory. An important issue in a memory hierarchy is the achievement of a high reliability. Caches provide for efficient read/write access to memory, and their reliability is essential to assure dependable computing. To achieve high reliability we must know internal structure of the cache.*

Key words: *cache memory, cache Tag, cache RAM, reliability*

1. INTRODUCTION

Cache memory is random access memory (RAM) that a computer microprocessor can access more quickly than it can access regular RAM. As the microprocessor processes data, it looks first in the cache memory and if it finds the data there (from a previous reading of data), it does not have to do the more time-consuming reading of data from larger memory.

Cache memory is sometimes described in levels of closeness and accessibility to the microprocessor. A level 1 (L1) cache is on the same chip as the microprocessor. L2 is usually a separate static RAM (SRAM) chip. The main memory is usually a dynamic RAM (DRAM) chip (Avizienis, A et al, 2004)

A cache is a component that improves performance by transparently storing data such that future requests for that data can be served faster. The data that is stored within a cache might be values that have been computed earlier or duplicates of original values that are stored elsewhere. If requested data is contained in the cache (cache hit), this request can be served by simply reading the cache, which is comparably faster. Otherwise (cache miss), the data has to be recomputed or fetched from its original storage location, which is comparably slower. Hence, the more requests can be served from the cache the better the overall system performance is (Rao, T.R.N..et al, 1989).

In addition to cache memory, one can think of RAM itself as a cache of memory for hard disk storage since all of RAM's contents come from the hard disk initially when we turn your computer on and load the operating system (we loading it into RAM) and later as you start new applications and access new data. RAM can also contain a special area called a disk cache that contains the data most recently read in from the hard disk.

2. BLOCK DIAGRAMN OF A CACHE MEMORY

Also, we admit that we use a direct mapped cache, and we adopt write-through as writing policy.

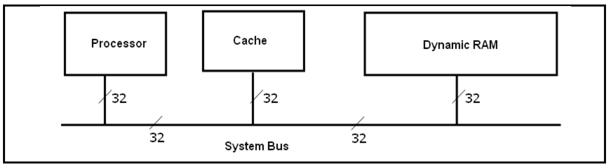

Fig. 1. The structure of a memory hierarchy with cache

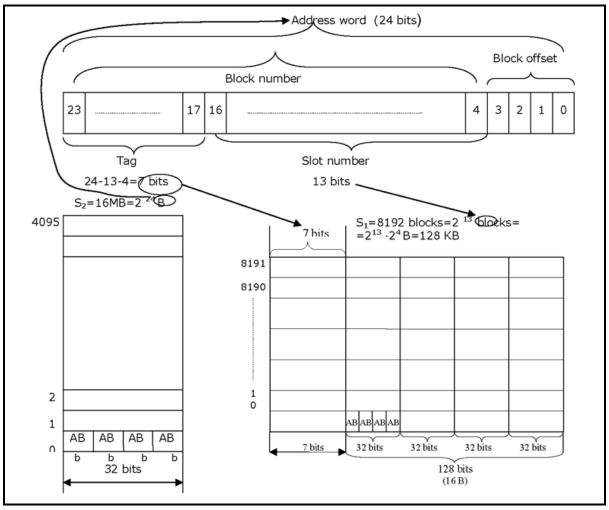

Fig. 2. The cache memory architecture

In a write-through cache, every write to the cache causes a synchronous write one in cache and one in to the backing store (main memory). (Seznec A. 1993).

We adopt a bus with 32 lines, and the cache has a size of 16 B. The structure of a memory hierarchy, in our case the design work, is given in fig. 1. In such a configuration, processor accesses information, either from cache, or from dynamic RAM.

If information is available in cache (in case of a hit) data is accessed quickly and the processor continues its work. If information is not present in cache (in case of a miss) data must be brought from Dynamic RAM memory. In this case, without losing of generality, and seeking simplicity of construction features, we will adopt low capacity features for cache and for main memory. Thus, as data inputs for the design solution we admit that we have a 128KB cache (32K x 32b), (where we have used the B notation for a byte and b for bit) and a main memory of 16 MB implemented with DRAM chips of 1KB.

A direct mapped cache specifies that if the information is in cache, there is one location where you can find that information. Cache is organized in small units called blocks. (Howard P. L. 1990).

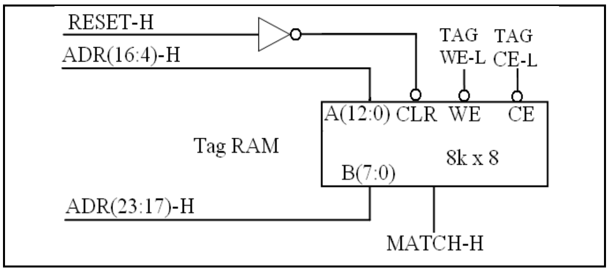

Fig. 3. The Cache Tag memory (Tag RAM)

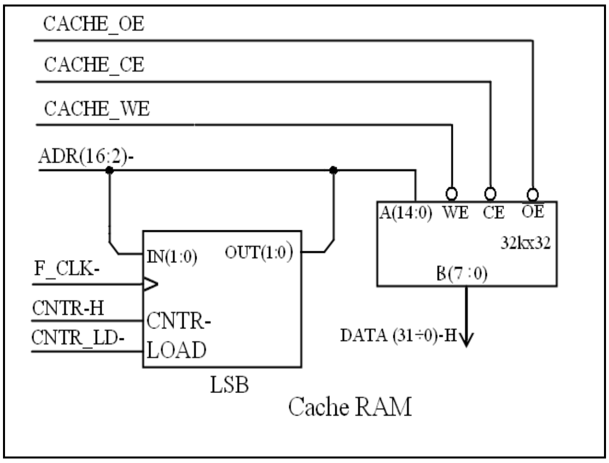

Fig. 4. The Cache RAM memory (Cache RAM)

This system will make four words transfers to fill a cache block In accordance with input data consisting of main memory capacity S_2 of 16MB, because 16MB = 2^{24} B, address bus will have 24 lines, denoted from 0 to 23, as they are presented in fig. 2. Also, having block size equal to 16 B = 2^4 B, to ensure access to the level of B (byte), we have from the 24-bit address, four bits allocated to so-called block, offset. Also, because the cache capacity is S_1 = 8192 blocks = 2^{13} blocks = $2^{13} \cdot 2^4 \cdot$ B = 128 KB, from the 24-bits of address, corresponding field of a cache slot is 13 bits. In conclusion for the Tag Field remains 24 – 4 – 13 = 7 bits.

We can say that the actual size is equal to $(7+32 \cdot 4) \cdot 8192$ bits. Cache interfaces with the processor and main memory via 24-bit address bus, a 32-bit data bus and by command lines further presented. Cache system is divided into two, a data memory part, that we will call further Cache RAM, having a size of 32K x 32 bits and a tag part, which we call Cache Tag memory (Tag RAM), having a size of 8K x 8 bits. (Chen P. M., et al.,1999, Novac O., et al 2008).

Since there are four 16-bit quantities on a cache line, a Tag location will identify four locations in cache, aspect presented in fig. 3 and fig. 5. Data lines of cache are connected to data bus of system, this way being used both, to write data in cache and to read data from the cache. Addresses to cache RAM are derived from ADR (16-2) lines of address bus. From 15 bits, 13 will be used to identify the cache line, and the following bits are used to identify in which from the locations of four bytes, from a line, byte address is founded.

The two least significant bits of address bus (LSB) are entries for one block of Cache RAM, this block is called LSB Control, aspect presented in fig. 4 and fig. 6.

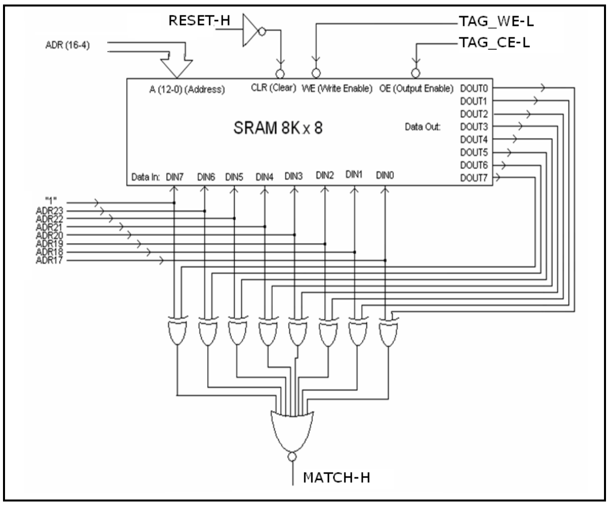

Fig. 5. Tag RAM memory (Detail)

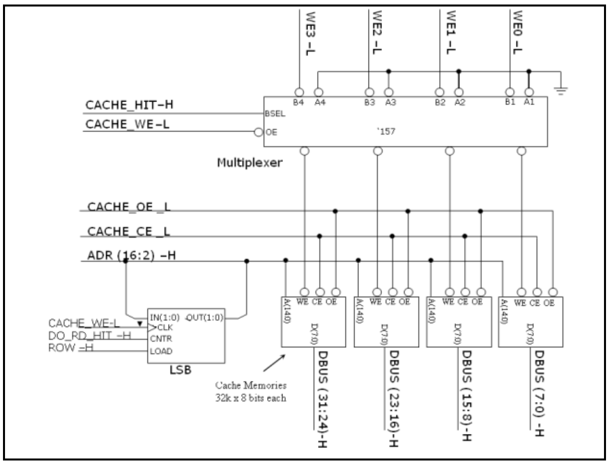

Fig. 6. Cache RAM memory (Detail)

This block contains a numerator and a multiplexer. During a normal operation, LSB two bits are supplied by the address multiplexer. When occurs the cache line loading from dynamic RAM, the two LSB bits, are obtained from the numerator, which increments the address with four locations. The available address in cache on LSB lines is the address that comes from the system, unless the signal CNTR-H is active, situation in which the address is supplied by the part that contains the cache numerator. The numerator is loaded with address, when signal CTR_LD-H is active.

There are other control lines, lines are usually in any system which contains RAM memory. Data are written in cache if signal CACHE_WE-L is active, memory RAM chip is selected by CACHE_CE-L signal, and data can be accessed from RAM when the signal CACHE_OE-L is active. In Fig. 6, are presented circuits which form cache RAM.

3. CONCLUSION

In this paper we have presented some dependability aspects regarding the design of a cache memory. In modern computer systems, at the cache level of the memory hierarchy, we can apply multiple error correction codes. This codes for detection and correction of errors are added to memories to obtain a better dependability

4. REFERENCES

Avizienis, A.; Laprie, J.-C.; Randell, B.; Landwehr, C. (2004). *Basic Concepts and Taxonomy of Dependable and Secure Computing*, IEEE Transactions on Dependable and Secure Computing, Vol.1, No.1, pp 11 - 33, January-March

Chen P. M.; Lowell D. E. (1999). *Reliability Hierarchies,* Workshop in Operating Systems.

Howard P. L. (1990). *The Design Book: Techniques and Solutions for Digital Computer Systems*, Prentice-Hall Inc., Englewood Cliffs, N. J.

Novac, O.; Vlăduțiu, M.; Vari-Kakas, Șt.; Hathazi,. F. I.; Novac, M.; (2008). *Aspects Regarding the use of SEC-DED Codes to the Cache Level of a Memory Hierarchy*, Proceedings of AIKED'08, University of Cambridge, 20-22 Feb

Paterson D.A.; Henessy J.L. (1990-1996). *Computer architecture. a quantitative approach*, Morgan Kaufmann Publishers, Inc

Rao, T.R.N.; Fujiwara, E. (1989). *Error-Control Coding for Computer Systems*, Prentice Hall International Inc., Englewood Cliffs, New Jersey, USA

Seznec A. (1993). *A Case for Two-Way Skewed-Associative Caches",* Proceedings of International Symposium on Computer Architecture, pp. 169 -178