A Cubic based Set Associative Cache encoded mapping

Vinothini S[1], Chandra Segar Thirumalai[2], Vijayaragavan R[3] and Senthil Kumar M[2]

[1]*Student, CK College of Engineering & Technology, Anna University, Cuddalore, India.*

[2] *Assistant Professor Senior, School of Information Technology and Engineering, VIT University, Vellore, India.*

[3]*Associate Professor, School of Advanced Sciences, VIT University, Vellore, India.*

Abstract - *Based on the internal or external interrupt, a bunch of words can be loaded on the cache memory. Due to the volatility nature of internal memory the cache history gets abscond once the system is deactivate. However, the performance of processor is based on the factors such as cache size and hit, write policy, type of cache mapping technique, CPU speed, front side bus, and depth of cache level. A number of standard cache addresses mapping techniques are available. This paper proposes a novel idea of set associative cache address mapping using cubic equation. The standard set associative mapping is remapped with cubic set associative technique for to secure the data in a non sequential fashion by having the standard mapping execution time. This work can also be applied to design the cache chip.*

Key Words: Cache mapping, physical addressing, cubic associative mapping

1. INTRODUCTION

To measure *set associative cache* address mapping system using cubic equation and to reduce time completion of cache address mapping system. The proposed cubic cache mapping system aims to understand easily on *associative cache* address mapping and to access fast in cache address mapping system.

In the computer architecture, some of the standard cache addresses mapping techniques are *direct mapping, set-associative mapping, and fully associative mapping.* Generally, the cache mapping techniques differs with other through the way data is fetched and stored from the main memory onto cache and correspondingly. Here the direct mapping will be working in the form of hashing where the input is the search key 'j' from this the mod function is applied with the cache size 'i' i.e., $9 \bmod 4 = 1$ which is mapped from main to cache. The data is referred from the cache through the policy called Locality of reference such as temporal and spatial reference. The temporal reference of a data refers a period of availability of data with respect to time and the spatial reference of a data refers to the location over the cache.

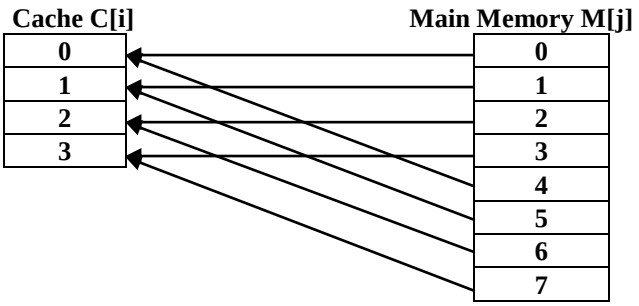

Fig -1.1: Direct mapping method

The above diagram shows the direct mapping method from the main to cache system.

The main aim of this paper is to experiment the time convolution of novel *associative mapping* with respect to standard associative mapping. Initially the central processor makes an effort to stare for a data or a word on the cache point called hit, or else then stare with an extra time as penalty on main memory level called miss. To compute a process on the system, the processor loads the set of passive instructions from the auxiliary memory to system memory to make it active for execution. In this regard, processor has to examine the sufficient space on both cache and physical memory. After the load process, the partial or complete execution of a process is stored on this memory and these changes reflect either only on cache or both cache and physical memory i.e., write policy selection. The *associative memory* refers the physical memory based on the three attributes such as *tag, block and word.* The proposed cubic cache mapping remaps the actual reference with the standard one in the linear order fashion.

In associative mapping the hit ratio of data gets increases as the number of blocks increases. As fully associative mapping has no index so the hit ratio performance comparatively much better in associative mapping.

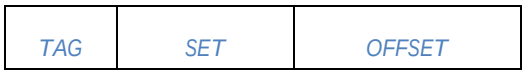

Fig -1.2: Set Associative Syntax

Fully Associative: No Index
Direct Mapping: Large Index

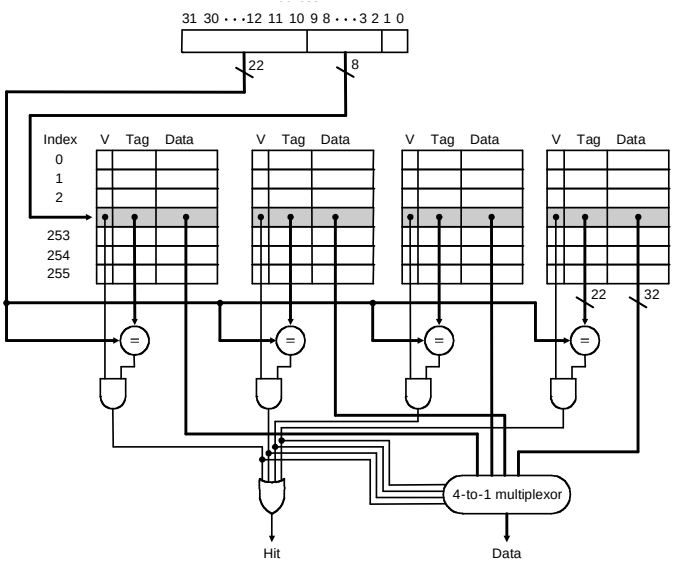

Fig -1.3: A four way Set Associative Cache

For instance, the system has the physical memory size of 8 MByte (2^{23}), the cache size 1024 KByte (2^{20}), and the block size is of 128Byte (2^7). From the width of the SET field the number of rows is to be determined. Each set maximum refers to four blocks i.e., *4-way set-associative cache* is applied onto the system. Thus the system consists of:

Number of Blocks in Physical memory
= Physical memory capacity / Per Block Size
= $2^{23} / 2^7 = 2^{16}$ blocks

Number of Sets = Number of Blocks / Number of ways
= $2^{16} / 2^2 = 2^{14}$ sets

TAG Size = Physical size / Cache Size
= $2^{23} / 2^{20} = 2^3$

OFFSET Size = Physical size / (Number of Sets * TAG Size) = $2^{23} / (2^{14} * 2^3) = 2^6$.

Hence, the Cubic System fields are *OFFSET* (= 6) and *TAG* (= 3).

2. MOTIVATION AND BACKGROUNG

Paragraph N. Chandramowliswaran et al., [1] describes that, the exponential growth of Information System needs a vital requirement to protect those data substantially from the prevention of unethical activities. To avoid such scenario in the internal memory system they are proposing a novel technique for *associative mapping* using *Graceful Code (GC) technique.* The processor performance is directly depends on the *cache, mapping technique, bandwidth, front side bus.* In paper [1], their proposed work is focused on secured mapping over *GC* technique applied to acquire a demanding result.

John S. Harper et. al., [2] describes Cache behavior is complex and also unstable, but it is a critical factor affecting program performance. Quantitative predictions of miss-ratio and information to guide optimization of cache are required to evaluate cache. Cache simulation gives accurate predictions of miss-ratio, but little to direct optimization. Hence, the program execution time always lesser than simulation time. Many analytical models have been made, but concentrate mainly on direct-mapped caches, often for specific types of algorithm, to give qualitative predictions. Analytical models of cache are presented, applicable to numerical codes consisting mostly of array operations in looping constructs. *Set associative caches* are determined, through an extensive hierarchy of cache reuse and interference effects, including numerous forms of temporal and spatial locality. An advantage is that it indicates sources of cache interference. The accuracy validated through program fragments. The predicted miss-ratios are compared with simulations it will be within 15 percent. The evaluation time of the models is independent it depends upon the problem size, in general many orders of magnitude faster than simulation.

N. Chandramowliswaran et al., [3] deals with a novel idea of *set associative cache address mapping* using linear equation. The standard *set associative mapping* is remapped with linear set associative for to secure the data in a non sequential portion by having the standard mapping execution time. This paper is mainly focused on to design the cache enhancement and improvement.

Stefano Di Carlo et al., [4] describe embedded microprocessor cache memories which affect observability and controllability problems and these factors can be observed during the system tests. Here they are also applying procedure to transform traditional march tests into software-based self-test programs *for set associative cache memories* through LRU replacement. In microprocessor testing instruction caches represents major difficulties due to limitations ing n two areas: 1) test patterns which must be composed of valid instruction opcodes and 2) test result observability: the results can only be observed through the results of executed instructions. For these purpose the proposed system will concentrate on the implementation of test programs for instruction caches. The main part of this work lies in the possibility of applying state-of-the-art memory test algorithms to embedded cache memories without any hardware or performance overheads and guarantees that detection of typical faults arising in nanometer CMOS technologies. The results got by constructing test programs for the LEON3 microprocessor show that it is possible to Protect the fault coverage of the original march

tests. The results also consider control blocks of the cache such as validity bits and control circuits, providing reasonable coverage also on these blocks. Additional fields that might be included in a microprocessor cache have not been work. Similarly to validity bits, their coverage should be tested the specific implementation of the target cache memory and whenever required, the test program should be improved to cover undetected faults.

S.Subha et al., [5] describe method to save energy in *set associative* additional information about next access to maintain in the cache ways. All the ways of the cache are put in either disable mode or low energy mode as supported by the cache. *Set associative* has a additional column called next_access_counter which contains the time of the next access of the address. The time counter is maintained to keep the track of the time cycle. During this mapping all the ways of the mapped set are enabled as in a traditional *set associative cache*. The model was simulated on SPEC2000 benchmarks with average energy savings of 19% and performance degradation of about 10%.A cache model for set associative cache that minimizes the energy consumed is proposed in this paper.

Ramy E. Aly et al., [6] describes that Variable-way set associative cache is used to maximize the cache performance and reduce the power consumption with the same performance. The *set-associative cache* Variable-way *set-associative* can be used in High performance or low-power operation modes. The proposed architecture is simulated on simplescalar simulator and tested on several Spec2000 Benchmarks. The results show on average 2% reduction in the miss rate at the high-performance mode and up to 43% reduction of the power consumption at low-power mode. We found that some sets get no performance improvement by doubling the associativity. In variable way *set associative* on sets that only get hit rate improvement.

S. Subha et al., [7] describes *set associative caches* have fixed number of sets with fixed number of ways in each set. The mapping of an address to a set is expanded to a subset of the number of sets by XOR' ing the address with 0, 1, 2, 4..., (Number of sets/2). The line is placed in any of the available ways in this mapping. If the cache is full, the cache is like *set associative cache*. In this mapping the number of ways that a line can be placed in a partially filled cache is increased. This paper proposes a modified address translation procedure in *set associative cache*. *Set associative* an average memory access time is 13%. The degradation of average memory access time for set associative memory is 256.bzip2. The consumption of energy increases as a function of the cache size.

3. CUBIC SET ASSOCIATIVE MAPPING

The system is deal with the implementation of cubic based set associative cache address mapping system using the equation $y = (a_0 + a_1x + a_2x^2 a_3x^3) \bmod n$.

3. 1 Solution Methodology

In this cubic cache based system, the concept of *set associative mapping* techniques linked with *direct and fully associative mapping.* The cache lines are grouped into sets. The number of lines, ***'n'*** in set can vary from 2 to 16. Set associative mapping will be divided into three parts.

STEP 1: *Initialize all secured variables to x,* a_0, a_1, a_2, a_3

STEP 2: *Get the inputs for real constant* a_0, *odd integer* a_1, *and pair of even integers* a_2 & a_3

If $a_1 \% 2 == 1$ *and* $a_2 \%2 == 0$ *and* $a_3 \%2 == 0$

Goto step 3

Else repeat step 1

STEP 3: *Compute* f(x) = $(a_0 + a_1x + a_2x^2 + a_3x^3)$

STEP 4: Find its equivalent remapped block,

$y = f(x) \bmod n$

STEP 5: End

For instance the private secured variables $[a_0, a_1, a_2, a_3] = [1, 3, 6, 4]$ is shown in **Table 1.**

Table -3.1: Remapped Set

X	*f(x)*	*Y Encoded Set*
0	1	1
1	14	14
2	63	15
3	172	12
4	365	13
5	666	10
6	1099	11
7	1688	8
8	2457	9
9	3430	6
10	4631	7
11	6084	4
12	7813	5
13	9842	2
14	12195	3
15	14896	0

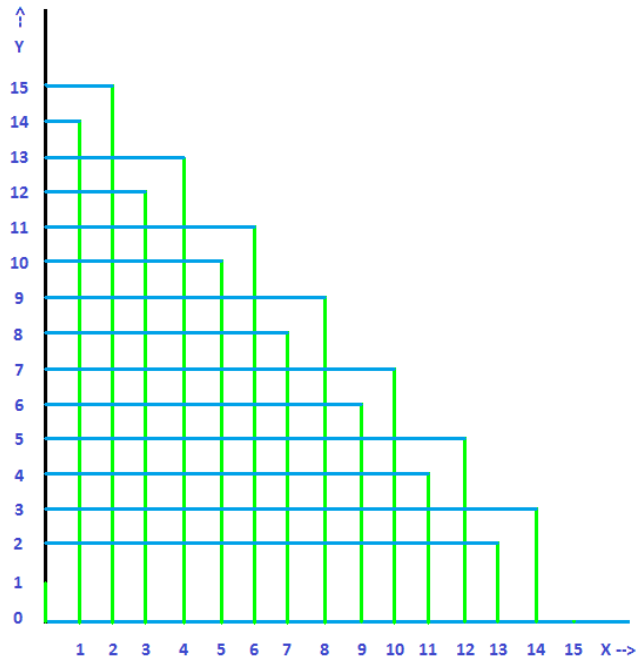

Fig. 1

Fig -3.1: One to One Remapping Graph of *[1, 3, 6,4]*

Here there are totally 16 tags are offered which is associated with remapped tags as shown in the above graph using Cubic equations.

4. CONCLUSIONS

In this work, cubic based *set associative cache* address mapping has been introduced and implemented by picking the parameters a_0, a_1, a_2 and a_4. Hence, the above remap clearly states that there exists unique one to one mapping from the search of cubic inputs, *X [0] to X [15]*. Moreover, this work can also be extended by using cubic one to one recursive mapping. The time complexity of the existing algorithm acquires only invariable time.

REFERENCES

[1] Chandramowliswaran N, Srinivasan.S and Chandra Segar.T, "A Novel scheme for Secured Associative Mapping" The International J. of Computer Science and Applications (TIJCSA) & India TIJCSA Publishers & 2278-1080 Vol. 1, No 5 / pp. 1-7 / July 2012

[2] John S. Harper, Darren J. Kerbyson, and Graham R. Nudd, "Analytical Modeling of Set-Associative Cache Behavior", Compute (1999), Pg. no: 1009-1024.

[3] Chandramowliswaran N, Srinivasan.S and Chandra Segar.T, "A Note on Linear based Set Associative Cache address System International J. on Computer Science and Engg. (IJCSE) & India Engineering Journals & 0975-3397 Vol. 4 No. 08 / pp. 1383-1386 / Aug. 2012

[4] Stefano Di Carlo, Paolo Prinetto, and Alessandro Savino," Software-Based Self-Test of Set-Associative Cache Memories", Computer (2011), Pg. no: 1030-1044.

[5] S.subha," Set Associative Cache Model with Energy Saving", Information Technology: New Generations (ITNG - 2008), Pg. no: 1249-1250.

[6] Ramy E. Aly, Bharat R. Nallamilli, Magdy A. Bayoumi, "Variable-way Set Associative Cache Design for embedded System Applications," Circuit and system (2003), vol.3., Pg. no: 1435-1438.

[7] S. Subha, "A Set Associative Cache Architecture," Information Technology: New Generations (ITNG – 2010), Pg. no: 1316 – 1317.

[8] Chandra Segar T, Vijayaragavan R, "Pell's RSA key generation and its security analysis IEEE Computing, Communications and Networking Technologies (ICCCNT), India IEEE & 978-1-4799-3925-1 Page 1 – 5/July 2013

BIOGRAPHIES

Vinothini S, completed M.E. in 2012 under the major of Applied Electronics with distinction from C.K college of Engineering and Technology and B.E in Electronics and Communication Engineering affiliated to Anna University. Her area of specialization includes Fuzzy Systems, Digital Image Processing, Computer Networks.

Prof. Chandra Segar T, currently working as Assistant Prof. Senior in School of Information and Technology and Engineering, VIT University, Vellore, India. His area of specialization includes Linear Cryptanalysis, Public Key Cryptosystems, Fuzzy Systems, Automata and Networking. About his publication, currently holds six International journals and one International conference.

Prof. Vijayaragavan, Ph.D is currently working as Associate Professor in VIT University, Vellore, India as Associate Professor in School of Advanced Sciences. His area of interest includes Cryptosystems, etc., About his publication, more than fifteen International journals and several International conferences.

Prof. M.Senthilkumar, M.S., M.Tech , is currently working as Assistant Professor (Senior) in the School of Information Technology and Eng(SITE) ,VIT University, Vellore. His area of interest includes Cloud Computing.

