ISSN: 0976-3104
ISSUE: Engineering and Technolohy

ARTICLE

OPTIMIZED DIRECT METHOD ROUTING FOR CACHE MEMORY IN RISC ARCHITECTURE

S Priya[1*], R Sathish Kumar[2], R Prabha[3], S Baskar[4]

[1,2,3] AP/Department of ECE, SNS College of Technology, Coimbatore, INDIA

[4]AP/Department of ECE, Karpagam Academy of higher Education, Coimbatore, INDIA

ABSTRACT

The growing gap between processor speeds and memory speeds is leading to progressively valuable cache misses, underscoring the necessity for classy cache hierarchy techniques. Increasing the associativity of the cache is a way to scale back its miss rate. Whereas set-associative caches generally incur fewer misses than direct-mapped caches, set associative caches have slower hit times. We tend to propose the reactive associative cache (r-a cache), that dynamically provides associativity in response to application demand. The r-a cache employs a unique multi-probe organization that uses a direct-mapped information array and a set-associative tag array. It's accessed sort of a regular direct-mapped cache for many accesses, however it additionally has the power to be accessed sort of a set-associative cache (using some way prediction), once its necessary to alleviate direct-mapped line competition. Circuit analysis indicates that a r-a cache encompasses a hit latency akin to that of a direct-mapped cache for all direct-mapped accesses and every one properly foretold set-associative accesses. Incorrectly foretold set-associative accesses can incur a further probe into the information array. Previous multi-probe cache organizations have suffered from poor initial-probe miss rates, and thus need several secondary probes, that will increase average hit time and demand for cache information measure over that of a direct-mapped cache. What is more, a r-a cache doesn't need cache block swapping, as in statically-probed multi-probe cache schemes similar to column associative and cluster associative. The feedback mechanism permits the r-a cache to live the 'predictability' of sure directions and to ban - ix - associative displacement for unpredictable accesses. Simulations show that direct map prediction resources, can vanquish column associative and prophetical successive associative, moreover as 83% reduction in area which speedups over a direct-mapped cache on a set of the SPEC95 benchmark suite.

INTRODUCTION

KEY WORDS
SPEC95, set-associative, latency, direct-mapped, routing.

Received: 10 Dec 2017
Accepted: 31 Dec 2017
Published: 12 Jan 2018

***Corresponding Author**
Email:
mailtopriya.sri@gmail.com

One of the elemental limitations of the performance of recent laptop systems is that the rate at that memory requests are often serviceable. Whereas processors have steady improved in machine performance by many orders of magnitude, memory speeds have not unbroken pace. This rift, likewise because the thought of neighborhood of memory reference, has motivated the event of memory hierarchies, within which tiny and quick SRAM cache reminiscences are wont to satisfy most requests. Sequential levels of progressively larger (and slower) caches are wont to step by step insulate the nimble processor from slower main memory DRAMs. The direct relationship between memory system performance and overall system performance has created cache style a relevant issue in laptop design. The performance of the memory system is especially necessary within the initial level of cache (L1) that is accessed by the processor. In fashionable processors, the L1 cache is on-chip, associated split between an instruction cache (L1 I cache) and a knowledge cache (L1 d cache). The high neighbor-hood and regular access patterns of instruction streams have created the L1 instruction cache perform comparatively well, however the L1 information cache of most systems still may be a vital performance bottleneck. There are 3 major reasons that this is often the case: (1) masses (and so information cache accesses) are within the crucial path of program execution, therefore there are few techniques which might hide the latency of a load operation; (2) for a range of circuit reasons, it's tough to extend the amount of cache ports to satisfy all the requests from the processor; and (3) because the wire delays begin to dominate intra-processor communication, the time to retrieve a cache block from following level within the hierarchy is growing, that makes misses within the information cache increasing costlier. During this thesis, we tend to examine optimizations to enhance the performance of the L1 information cache. The normal measures of a cache are its miss rate (the pc of accesses that are not by the cache), and therefore the hit latency (the time it takes to service cache hits). - Two - The miss rate indicates however usually following level within the hierarchy is accessed; therefore it becomes a lot of necessary because the L2 access latency will increase. The hit latency is usually within the crucial (circuit) path of processor execution, therefore it will figure conspicuously within the clock cycle of the processor. The only sort of a L1 d cache may be a direct-mapped organization, within which every cache block address is mapped to one potential location within the cache array. This structure permits for quick hit times, and conjointly achieves respectable miss rates, sometimes under V-day for typical number applications. Set associative caches permit a given cache block to reside in additional than one place (way) among the array, that will increase occupancy of reused information, and thus decreases the miss rate. The decrease in miss rate is often forceful. For a set of the SPEC95 benchmark suite [1] with an 8k L1 d cache, a 2-way set associative cache has (on average) thirty second fewer misses than a direct-mapped cache. A 4-way set-associative cache has forty first fewer misses than a direct-mapped cache. This decrease in miss rate is sadly amid associate equally forceful increase in hit latency, because of the extra logic necessary to implement the associativity. victimization the cache analysis tool CACTI [2], we tend to estimate that associate 8k 2-way set associative cache is fifty three slower than associate 8k direct-mapped cache. This creates a significant trade-off between hit latency and miss rate within the L1 information cache. In Section one.1, we tend to examine a lot of closely the circuit problems that management the hit time of direct-mapped and associative caches. the big increase in hit latency between direct mapped and a set-associative caches has motivated interest in multi-probe cache schemes, that have a hit-latency getting ready to that of a direct-mapped cache, however with miss rates nearer to associate associative cache [3].

This paper is organized as follows. Section II shortly provides survey on normal error protection mechanisms and to boot discusses previous schemes projected to chop back vulnerability of cache recollections against transient errors. Section III details tag bits errors classified into four subgroups and normal ways that elaborate draw back Definition of the prevailing associative mapping, describes our projected direct mapping. Section IV presents our experimental results for our Existing and Proposed. Finally, Section V concludes this paper.

CONVENTIONAL CACHES

[4]Statically-probed caches are around for nearly fifteen years, however haven't gained trade acceptance, because of the need for cache block swapping. Prophetical consecutive associative is equally sure, because of its serious increase in information measure utilization. We advise a brand new dynamically-probed theme known as reactive-associative (r-a), that doesn't need cache block swapping, and encompasses an information measure demand the same as that of a [5] direct-mapped cache. The r-a cache employs a completely unique organization within which the information array is direct-mapped, however the tag array is set-associative .aspect contains one data array, like during a direct-mapped cache, which may support only 1 probe at a time aspect doesn't need tag match information from the tag array on the initial probe, and so the crucial path although the information aspect is appreciate that of a direct-mapped cache. The tag aspect contains equivalent logic to the tag aspect of a set associative cache .Specifically, the tag array is split into approach banks that are probed in parallel for a given set (from the cache block address). [6] The critical-path delay of the tag aspect of a set-associative cache is such as that of a direct-mapped tag aspect. This is as a result of the set-associative tag array contains 2 or a lot of approach banks that are every smaller and quicker than a direct-mapped tag array; since the approach banks are probed in parallel, the tag array of Associate in Nursing associative cache is really quicker than that of a direct-mapped cache, that offsets the rise in different logic. [7]

The quicker approach banks that are probed in parallel).While the organization of the r-a cache is attention-grabbing, verity novelty [8] comes from the displacement/prediction subsystems. The r-a cache can solely displace conflicting blocks to line associative positions, so relieving what would be line competition (and thrashing) during a direct-mapped cache. Conflicting cache blocks are detected employing a questionable victim list that tracks recent L1 d-cache misses. By solely displacing conflicting blocks, we have a tendency to relieve pressure on the way-predictor that currently should solely track approach prediction info for displaced (contentious) cache blocks. The notion of solely displacing.

Certain blocks are mentioned as selective displacement. What is more, we have a tendency to determine that in every application, there are a bunch of directions that have poor foregone conclusion. The r-a cache employs a feedback mechanism which can live the dynamic prediction accuracy per individual (or teams of) directions. Directions with poor foregone conclusion are prohibited from accessing displaced cache blocks (i.e. the cache blocks are forced to reside in their direct-mapped positions). By limiting the candidates for associative displacement, the r-a cache can have the next overall miss rate, in general, than the sooner multi-probe cache schemes. However, this performance disadvantage is over offset by the accrued performance because of have a way lower probe0 miss rate and by eliminating the necessity for pricey block swapping. The approach predictor uses a prediction handle that is a few perform of system state that correlates to knowledge access patterns, and is on the market before the effective address. This prediction handle is employed to index into a prediction table that permits for some way prediction. prophetical consecutive associative suggested the employment of XOR approach prediction, within which the supply register contents are logically XORed with the offset price, to supply - ten -an approximation of the information address (this is comparable in favor to zero-cycle hundreds, that were projected in [9]). Sadly, it's unlikely that this technique might be used thanks to the strict temporal arrangement constraints of cache accesses. For XOR prediction, the logic operation would have to be compelled to be performed, and some way prediction table operation completed, all in the time of a traditional address computation. For r-a, we advise .A reactive associative cache as shown the [Fig. 1] Mistreatment laptop prediction that could be a weaker prediction handles that correlates the address of the memory operation (i.e., the PC value) to the approach prediction. Since the laptop is on the market several cycles prior to the memory request is sent, there's lots of time for table lookups, and there's no risk of compromising the crucial path of cache accesses. In this variety of mapping the associative memory is used to store content and addresses every of the memory word. This permits the position of the any word at anywhere among the cache memory. It's thought-about to be the fastest and conjointly the foremost versatile mapping sort. [10]

DIRECT MAPPING

Block identification: let the most memory contains n blocks (which need log2 (n)) and cache contains m blocks, of memory is mapped (at different times) to a cache block. every cache block encompasses a tag oral communication that block of memory is presently gift in it, every cache block conjointly contain a sound bit to confirm whether or not a memory block is within the cache block presently.[11]

- Number of bits within the tag: log2 (n/m)
- Number of sets within the Cache: m
- Number of bits to spot the right set: log2 (m)

The memory address is split into three parts- tag (most MSB), index, block offset (most LSB) so as to try and do the cache mapping.

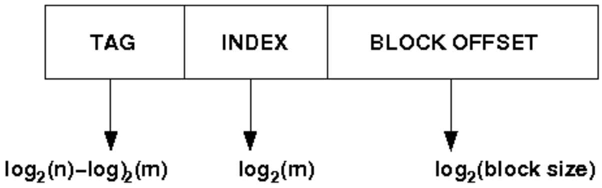

Fig. 1: A Cache mapping

1. Choose set mistreatment index, block from set mistreatment tag.
2. Choose location from block mistreatment block offset.
3. Tag + index = block address

Diagram of a direct mapped cache (here main memory address is of 32 bits and it gives a data chunk of 32 bits at a time):

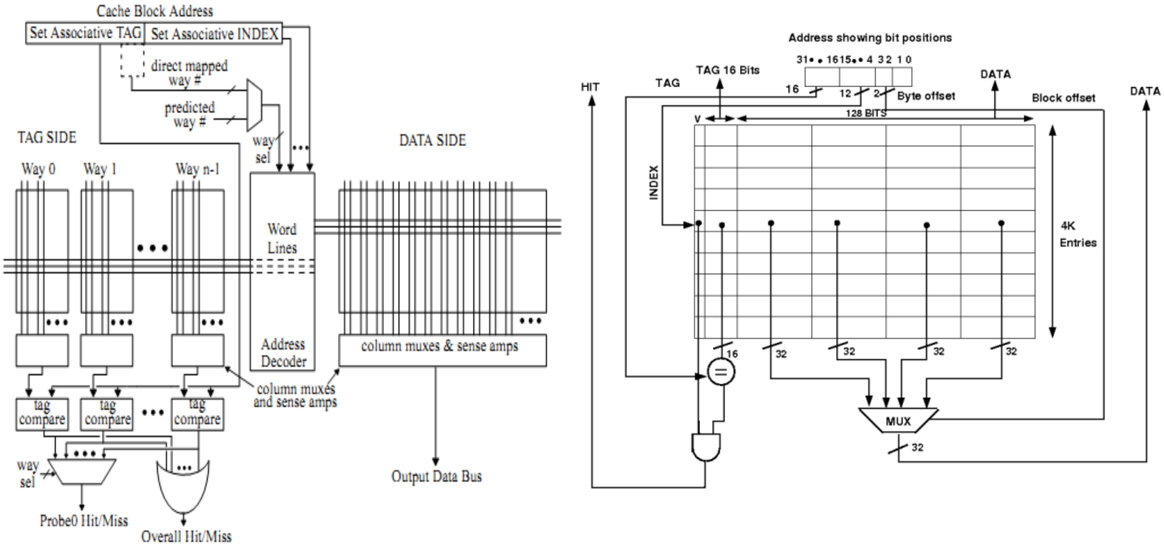

Fig. 2: A Direct mapping

If a miss occur electronic equipment bring the block from the most memory to the cache, if there's no free block within the corresponding set it replaces a block and place the new one. Electronic equipment uses completely different replacement policies to make a decision that block is to interchange. The disadvantage of the direct mapped cache is that it's simple to make. Below could be a straightforward cache that holds 1024 words or 4KB, memory address is thirty two bits. The tag from the cache is compared against the foremost important bits of the address to see whether or not the entry within the cache corresponds to the requested address because the cache has 210 or 1024 words and a block size of 1 word, ten bits are wont to index the cache, departure 32-10-2=20 bits to be compared against the tag. If the tag and therefore the most vital twenty bits of the address are equal and therefore the valid bit is on then the request Hits within the cache otherwise miss happens. No replacement policy has been enforced within the circuit. In direct mapping the RAM is made use of to store info and some is hold on among the cache. Associate in address home is split into two elements index field and tag field. The cache is used to store the tag field whereas the rest is hold on among the most memory. Direct mapping`s performance is directly proportional to the Hit quantitative relation. when the firm analysis we have a tendency to conclude direct mapping terms to be the foremost economical technique in terms of space, power and delay that has simulated within the below mere design. A given memory block is mapped into one and solely cache line. All the higher than problems are corrected mistreatment direct mapping. As shown in the [Fig. 2].

RESULTS AND DISCUSSIONS

Xilinx/ISE simulations and precision RTL of mentor graphics

The proposed adder and its corresponding blocks are described using structural VHDL and synthesized employing Xilinx Synthesis Tool (XST), Web PACK version 13.2 and Precision RTL of Mentor Graphics. The implementation was targeted to Xilinx spartan-3E low power, Selected Device: 3S250EPQ208.

The logical routing can be observed from the obtained Place and route result from the FPGA Editor option in Xilinx synthesizer. It is observed that about 10% (CLB taken from [Table 1]) area for the targeted FPGA is covered for the implementation of this System. The CLB's are connected in cascade manner to obtain the functionality for the designed system. To ensure that the hardware implementation works properly, simulation test was performed using I-Sim (0.76.xd). [12]

Impact of the proposed flow on peak memory usage, timing and area

In this paper, the conventional approach and the proposed method is analyzed based on the cost function of placer and router. As shown in the [Table 1] the number of LUT's, memory usage and timing are reduced in proposed flow due to less consumption of adder circuit in the design.[13]

As the [Table 2] show:

The routed architecture of the conventional and proposed method on Xilinx spartan-3E low power, Selected Device: 3S250EPQ208 is tabulated in [Table 1] shows the proposed method outperforms the conventional architecture in terms of placement and routing which has shown in the [Table 2] Shows the proposed method reduced the area up to 83% ([Table 2] –COMPARED THE METHODS AND TASKEN THE DIFFERENCE) when compared to conventional method.[14]

As shown in the [[Fig. 3].In terms of overhead, since the conventional approach and the proposed method only change the placement and routing of the design, as the usage of the CLB (configurable logic blocks) varies which provides the overhead and delay lesser than existing approach. In addition, no unreachable CLBs are reported by the original method and the proposed method which helps to overcome the limitation of the original approach. Hence, the conventional approach and the proposed method sustain CLB overhead It is observed that about 83 % ([Table 2] CLB's) area for the targeted FPGA is covered for the implementation of this System as shown in the [Table 2]. The CLB's are connected in cascade manner to obtain the functionality for the designed system. As the coverage area of the CBs reduces minimize route channel width. The lower delay comes from that the number of glitches is smaller when the carry propagates Quicker through the logic. The slice usage of the proposed method is reduced power up to 80% [Table 2] than the conventional approach as shown in the [Table 1].[15]

However, in this work the main target is using the direct mapping result in less area, memory, Power and delay when compare to associative mapping (As shown in the [Table 1 and 2]).

Table 1: Performance analysis for the existing and proposed architecture

	Base paper Existing	Base paper proposed
LUTs	425 out of 63,400	71 out of 4896
BELS	437	90
Delay		
Minimum input arrival time before clock	5.714ns	6.773ns
Maximum output required time after clock	1.102ns	4.283ns
Power	0.165 Watts	0.080 Watts
Memory	506756 Kilobytes	262724 Kilobytes
Flip Flops/Latches	11	43

Table 2: Performance Timing analysis for the existing and proposed architecture

	Base paper Existing	Base paper proposed
Total REAL time to PAR completion	23 secs	8 secs
Total CPU time to PAR completion	22 secs	7 secs

Power

Power dissipated to drive the input of the flip flop is due to switching power, short-circuit and leakage power. [15]

Power= $P_{switching}$ +$P_{short\ circuit}$ + $P_{leakage}$ (1)

Switching Activity Factor: α

If the signal is a clock, α = 1 then If the signal switches once per cycle, α = ½.besides For Dynamic gates: switch is either 0 or 2 times per cycle, α = ½ and for the Static gates: depending on design, but typically α = 0.1

$P_{switching}$ = a .f.Ceff .Vdd2 (2)

Short-circuit power occurred when there is a transition between VDD and GND occurs

$P_{short\ circuit}$ = Isc .Vdd.f (3)

$P_{leakage}$ = f (Vdd, Vth, W/L) (4)

The power comparison for various styles due to switching activity has shown in [Table 1]

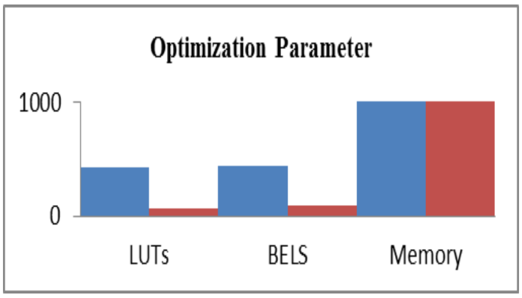

Fig .3: Optimization parameter comparison

CONCLUSION

In this analysis we have a tendency to propose the direct mapping supported the key observation that even for applications that take pleasure in accrued associativity, the common case for a direct-mapped cache may be a hit. This observation implies. That associativity is required just for conflicting block and will not be provided at the expense of upper hit latencies for all accesses. The r-a cache keeps the maximum amount knowledge in direct-mapped positions as attainable, and displaces solely conflicting blocks to set-associative positions. The direct mapping thus provides versatile associativity that will increase or decreases counting on application characteristics. The direct mapping cache may be a dynamically-probed multi-probe organization that includes a direct-mapped hit latency for the initial probe. An on the spot connect mesh routing structure is provided for interconnecting configurable logic blocks inside a programmable logic device. The structure includes multi-bit interconnect busses and a extremely regular structure distributed throughout a configurable array sanctioning high direct interconnect utilization to adjacent and non-adjacent logic blocks, high speed circuit implementation, and improved temporal arrangement characteristics. The direct connections of the invention square measure the well-liked interconnect path between logic blocks as a result of the considerably scale back the common interconnect delay, thereby permitting the programmable logic device to work at a better speed.

CONFLICT OF INTEREST

There is no conflict of interest.

ACKNOWLEDGEMENTS

None

FINANCIAL DISCLOSURE

None

REFERENCES

[1] Jeongkyu Hong, Jesung Kim, and Soontae Kim [2015], Member, IEEE, Exploiting Same Tag Bits to Improve the Reliability of the Cache Memories, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS. 23(2).

[2] An Overview of Cache Optimization Techniques and Cache, Markus Kowarschik and Christian Wei; Fakultat

furInformatik Technische University at Munchen, Germany.

[3] [1999] A Trace Cache Microarchitecture and Evaluation", Eric Rotenberg, Steve Bennett; IEEE TRANSACTIONS ON COMPUTERS. 48(2).

[4] Pendse R, Kushanagar N, Walterscheidt U. [1845] Investigation of Impact of Victim Cache and Victim Tracer on a Fully Associative Disk Cache", Department of Electrical Engineering, The Wichita State University, North Fairmount.

[5] David A, Patterson, John L, Hennessy. [2009] Computer organization and design: the hardware/software interface. ISBN 0-12-374493-8, ISBN 978-0-12-374493-7 Chapter 5: Large and Fast: Exploiting the Memory Hierarchy. 484.

[6] Zeng D, Wang FY, Liu M. [2004] efficient web content delivery using proxy caching techniques. Systems, Man, and Cybernetics, Part C: applications and Reviews, IEEE Transactions on. 34(3):270-280 (3).

[7] Rabinovich M, Spatschek O. [2002] Web caching and replication. Addison-Wesley Longman Publishing Co., Inc. 361.

[8] Baskar S, Pavithra S, Vanitha T. [2015] Optimized placement and routing algorithm for ISCAS-85 circuit, Electronics and Communication Systems (ICECS), 2nd International Conference on 2015/2/26, IEEE. 958-964.

[9] Baskar S. [2012] Error recognition and correction enhanced decoding of hybrid codes for memory application. International Journal of Advanced Research in Computer and Communication Engineering, IJARCCE. 10:816-820.

[10] Baskar S. [2014] Error recognition and correction enhanced decoding of hybrid codes for memory application, at Devices, Circuits and Systems (ICDCS), 2nd IEEE Conference. 1-6.

[11] Baskar S, Reliability-Oriented Placement And Routing Analysis In Designing Low Power Multipliers. International Journal of Applied Engineering Research. 10(44):31384-31390.

[12] Takase T, et al. [2002] A web services cache architecture based on xml canonicalization, in In Proceedings 11th Int. World Wide Web Conf. (Poster Paper), HI: Honolulu.

[13] Brooks C, et al. [1995] Application-specific proxy servers as http stream transducers, In Proceedings 4th World Wide Conference. 539-548.

[14] Lee S, Jung J, Kyung CM. [2012] Hybrid cache architecture replacing SRAM cache with future memory technology. in Circuits and Systems (ISCAS), IEEE International Symposium on. IEEE.

[15] Aggarwal N, et al. [2007] Isolation in Commodity Multicore Processors. Computer. 40(6):49-59.