Design of Cache Controller for Low Power and High Speed Application by Using Cache Memory

Mr.Parag G. Shewane[1], Ms.Amruta G. Shewne[2], Mr.Ritesh C. Ujawane[3], Mr.Sagar A Patil[4], Ms. Bhagyashree G. Mudaliar[5]

[1,3,5] *Dept. of Electrical Engineering, RTMNU/Dr. Babasaheb Ambedkar College of engineering Nagpur, Maharashtra, India*

[2]*Network Associate, Tata Communication Limited, Pune, Maharashtra, India*

[4]*Department of Electrical, Mahavir Polytechnic Nashik, Maharashtra India*

***Abstract*- We report on the design of efficient cache Controller suitable for Cache Memory use in FPGA-based processors. Cache systems are on-chip memory elements used to store data that are frequently referenced by programs. The advantage of storing data in cache, as compared to RAM, is that it has faster retrieval times, but it has the disadvantage of on-chip energy consumption. A cache-decay interval is the amount of time a cache element holds unreferenced data before being turned off (cleared). Semiconductor memory which can operate at speeds comparable with the operation of the processor exists; it is not economical to provide all the main memory with very high speed semiconductor memory. The problem can be alleviated by introducing a small block of high speed memory called a cache between the main memory and the processor.**

Index Terms- Cache Memory, Main Memory, Cache Controller.

I. INTRODUCTION

To meet the growing needs of computing power, communication speed and performance requirements demanded by today's applications, processor clock speed has to be increased. However, increasing clock speed is not viable due to heat dissipation and power consumption constraints. There are many algorithms proposed by researcher for improvement in time speed and reduces in power consumption.

The proposed system is modeled using design of cache controller. Due to the design of Cache Controller the data processing capabilities of a processor have been increased at a much faster rate when compared to the memory performance or speed in recent years. This trend can be analyzed by looking at the time to performance graph shown as Moore's law effect in figure. In this new generation, the processors are having very large main memory access latency and it has been predicted that it will be increased further.

The advantage of such a model is its simplicity, as cache coherency mechanisms are not required despite this potential limitation. This paper deals with the design of efficient cache memory for detecting miss rate in cache memory and less power consumption. This cache memory may use in future work to design FPGA based cache controller.

II. SYSTEM ARCHITECTURE

Cache controller that communicates between microprocessor and cache memory to carry out memory related operations. The functionality of the design is explained below. Cache controller receive address that microprocessor wants to access Cache controller looks for the address in L1 cache. If address present in L1 cache the data from is provided to microprocessor via data bus. If the address not found in L1 cache then cache miss will occur. Cache controller then looks same address in cache L2. If address present in L2 cache the data from location is provided to microprocessor. The same data will replace in cache L1. If the address not found in L2 cache then cache miss will occur. In our paper work we designed cache memory (L1) for detecting miss rate in cache memory and less power consumption and to track cache miss in cache memory we designed cache controller.

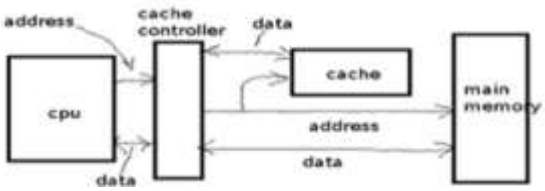

Fig1: System Architecture

III CATCH MEMORY

Cache memory is closest to the processor and fastest of all other memories. It is used to store the instructions or data which are frequently required by the processor. It has the highest level in the memory hierarchy. Whenever there is any request from the processor to read or write, cache memory is referenced first. Cache systems are on-chip memory element used to store data. Cache memory is used to increase the data transfer rate between processor and main memory. The cache has to be designed such that the data required by the processor should be readily available in it. The time required to access cache should be minimum as well. While designing the cache, few steps have to be followed as shown in figure 2

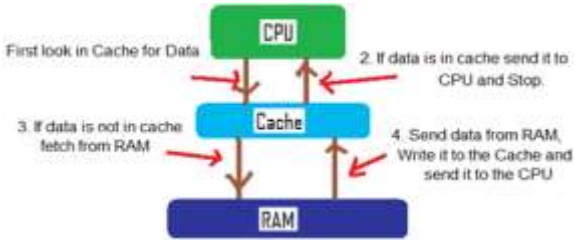

Fig2. Cache Design Steps

A .Write Operation of Cache Memory

During write operation, if there is a hit from any way, then the set corresponding to that particular way will be selected with the help of index bits e.g. if the index bits of the 12 bit address line is 0000, set number 0 will be selected. If it is a miss, the way will be selected randomly by using a counter as the replacement method. Also as the line size of cache is 16 bytes, where to write in the cache is obtained with the help of block offset bits e.g. if the block offset is 0001, the data would be written to the location of second byte from the right.

B. Read Operation of Cache Memory

While reading the cache all the four ways will be enabled. The set, from where the data has to be read, is selected, with the help of index bits extracted from the address bits coming from the processor. The byte which has to be read is selected with the help of block offset bits. Tags present in each way will be compared with the tag bits of the address coming from the processor. Comparator is used for comparing the tag. If the same tag is present in any way of the cache, it is ahit. The data will be fetched from the location corresponding to that particular tag and will be provided to the processor. If the tag is not presenting the cache, it is a miss and the data has to be brought from the main memory.

IV. CACHE CONTROLLER

To reduced cache miss we designed cache controller as shown in figure 8. Cache controller is designed on the basis of locality of the reference. spatial locality of the reference means Given an access to a particular location in memory, there is a high probability that other accesses will be made to either that or neighboring locations within the lifetime of the program. if requested address is missing then cache controller generate the address range. This address range is nothing but the neighboring addresses of miss requested address. This address range given to the microprocessor by cache controller to replace addresses between this range into cache tag memory. This will reduce the cache misses.

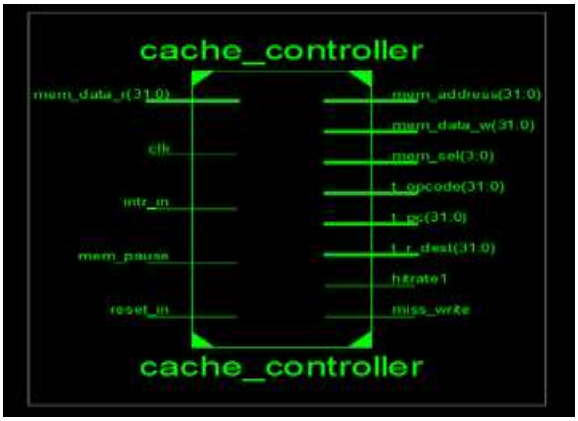

Fig3: RTL view of Cache Controller

V. SYNTHESIZING AND IMPLEMENTING THE DESIGN

The design has been synthesized using Xilinx Synthesis Tool (XST). Table 1 shows the device utilization summary of cache controller designed.

The development board that has been targeted is Virtex-6 ML605 evaluation board. It shows number of logics available which includes number of Slice Registers, number of slice LUTs, number used as logic, number used as memory, number of bonded IOBs and number of Block RAM. It has been shown that the logic utilization is 0%, 4%, 4%, 35%, 6% respectively.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	1267	184304	0%
Number of Slice LUTs	4512	92152	4%
Number of fully used LUT-FF pairs	265	5534	4%
Number of bonded IOBs	202	576	35%
Number of BUFG/BUFGCTRLs	1	16	6%

Fig4: Device utilization summary of the designed cache controller

VI. CONCLUSION

In this paper, we have presented design of cache memory on FPGA for detecting cache miss and design of cache controller for tracking induced cache miss in cache memory. Such an approach would be of great utility to many modern embedded applications, for which both high-performance and low-power are of great importance. This cache memory and cache controller may used in FPGA based processors.

REFERENCES

[1] John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, 4th ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2006.

[2] Maurice V. Wilkes, "The memory gap and the future of high performance memories", SIGARCH Comput. Archit. News, 29(1), 2–7 March, 2001.

[3] A. J. Smith, "Cache memories", ACM Computing Surveys, vol. 14, no. 3, pp. 473-530, September 1982.

[4] Bani R.R, Mohanty, S.P, Kougianos E, and Thakral G, "Design of a Reconfigurable Embedded Data Cache", inProc. Int. Symp. Electronic System Design, pp.163-168, December 2010.

[5] Chuanjun Zhang, Vahid F, and Lysecky R, "A self-tuning cache architecture for embedded systems", inProc. Europe Conf. and Exhibition on Design, Automation and Test, vol.1, pp.142, 147, 16-20 February 2004.

[6] Frank Vahid and Tony Givargis, Embedded System Design: A Unified Hardware / Software Approach. John Wiley & Sons, 2006.

[7] Chenxu Wang, Jiamin Zheng, and Mingyan Yu, "Cache Performance Research for Embedded Processors" inProc. SciVerse Science Direct Int. Conf. Solid State Devices and Materials Science, pp.1322-1328, March 2012.

[8] Maciej kurek, ioannis ilkos, wayne luk, "Customizable security-aware cache for fpga-based soft processors", Department of computing, imperial college London

[9] Nawaf Almoosa, Yorai Wardi, and Sudhakar Yalamanchili, Controller Design for Tracking Induced Miss-Rates in Cache Memories, 2010 8th IEEE International Conference on Control and Automation Xiamen, China, June 9-11, 2010

[10] Computer architecture and organisation : John .p. Hayes(Mc Graw hill publication)

[11] Computer architecture: Forth Edition, John L. hennessy and David A. Patterson.