

# IMPACTS OF LEVEL-2 CACHE ON PERFORMANCE OF MULTIMEDIA SYSTEMS AND APPLICATIONS

Abu Asaduzzaman, Manira Rani

CSE Department, Florida Atlantic University, 777 Glades Rd, Boca Raton, FL, USA

Darryl Koivisto

Architecture Modeling Group, Mirabilis Design, Inc., 830 Stewart Dr, San Jose, CA, USA

Keywords: Multimedia system, multimedia application, performance analysis, cache memory hierarchy, VisualSim.

Abstract: Multimedia systems normally suffer while processing multimedia applications because of their limited resources. The demand for tremendous amount of processing power raises serious challenges for multimedia systems and applications. Studies show that cache memory has strong influence on the performance of multimedia systems and applications. In our previous work, we optimize level-1 cache parameters to enhance the performance of portable devices running MPEG4 decoder. The focus of this paper is to evaluate the impacts of level-2 cache on the performance of multimedia systems running MPEG4 and H.264/AVC encoders. We develop VisualSim model and C++ code to run the simulation. We measure miss rates, CPU utilization, and power consumed by varying level-2 cache size. Simulation results show that the performance of multimedia systems and applications can be enhanced by optimizing level-2 cache.

## 1 INTRODUCTION

Due to the massive popularity, multimedia systems and applications have attracted researchers from every corner of the globe. The future multimedia system should support more and more functions to satisfy the growing demands. Computer architectures are changing accordingly in response to the demands to support multimedia applications. Processing multimedia applications is a significant challenge for memory sub-systems. The performance of such a system highly depends on the memory hierarchy (Asaduzzaman, 2004), (Grigoriadou, 2003), (Slingerland, 2005).

The ISO (International Organization for Standardization) standard MPEG4 and ITU-T (International Telecommunication Union - Telecommunication Standardization Sector) standard H.264/AVC are the principal multimedia applications for multimedia systems. It is important to understand the encoding and decoding algorithms (CODEC) and the composition of their data set to improve the performance of the system running them. Multimedia systems should efficiently

perform video algorithms to support these applications and meet low power and bandwidth requirements. For time-critical applications, the systems should react in real-time. Studies show that multimedia systems need a computationally intense architecture implementation in order to support multimedia applications (Asaduzzaman, 2006), (Wu, 2004), (Richardson, 2002), (Koenen, 2002), (Ely, 1995), (Schaphorst, 1999).

Cache memory is used to give bridge the processor-memory speed gap and reduce the mean memory access time. Miss rates generate significant excess cache-memory traffic. For multimedia systems, where battery power and bandwidth are limited, cache inefficiency can have a direct cost impact, requiring the use of higher capacity components that can drive up system cost (Soderquist, 1997) and (Wolf, 2005).

Figure 1 shows the memory hierarchy with level-1 (CL1) and level-2 (CL2) caches. In Figure 1(a), the data is fetched from the main memory using the shared bus in case of a CL1 miss. In Figure 1(b), it is expected that the data is fetched from CL2 in case of a CL1 miss; for a CL2 miss, the data is fetched from the main memory. Therefore, the addition of CL2

cache should improve performance and decrease power consumption by reducing the data access time. However the system performance entirely depends on the hardware/software used to build the system and the target applications.

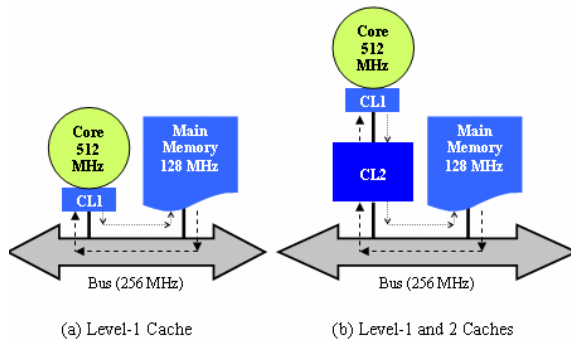


Figure 1: Memory hierarchy (a) with level-1 cache and (b) with level-2 cache.

In this work, our focus is to evaluate the impacts of level-2 cache on performance of multimedia systems and applications. In Section 2, some related articles are summarized. Two popular multimedia applications (MPEG4 and H.264/AVC encoders) are briefly explained in Section 3. Simulation details are presented in Section 4. In Section 5, the simulation results are discussed. We conclude our work in Section 6. Finally, the VisualSim model of the simulated architecture is provided in Appendix A.

## 2 RELATED WORK

Multimedia systems and applications are very interesting fields to the researchers all over the globe. A lot of work has already been done. Some related articles are discussed in this Section.

The impact of cache memory on performance of multimedia systems and applications is studied in (Slingerland, 2005) and (Asaduzzaman, 2006). According to these studies, multimedia applications exhibit higher data miss rates and comparable lower instruction miss rates. These studies also indicate larger data cache line sizes than are currently used would be beneficial in case of multimedia applications. Various contemporary techniques for optimizing memory used in embedded systems are discussed in (Wolf, 2003), (Li, 1998), and (Panda, 2001). Using trace-based techniques and WCET analysis performance can be analyzed. It has been shown that cache not only improves performance, but reduces energy consumption. In (Koenen, 2002), the trade-off between the energy dissipation of software and that of system resources like cache and

main memory is studied. It is shown that there is no straight forward way to judge the change of performance when various system parameters and application are changed. Studies indicate that simulation tools can be used for performance evaluation of multimedia systems and applications.

In (Asaduzzaman, 2006), cache modeling and optimization is conducted for portable devices running MPEG4 video decoder. Cache miss rates are measured using Cachegrind and VisualSim for varying cache parameters. Both Cachegrind and VisualSim gave lower miss rates with increased CL1 line size and associativity levels.

A general-purpose computing platform running MPEG2 application is studied in (Soderquist, 1997). Running MPEG2 generates at least two streams of encoded and decoded video being concurrently transferred in and out of main memory. Any excess memory traffic generated by cache inefficiency will further make this situation worse. Experimental results show that the addition of a larger second level cache to a small first level cache can reduce the memory bandwidth significantly.

In this work, we keep CL1 fixed and vary CL2 size and measure various performance metrics for MPEG4 and H.264/AVC encoders.

## 3 MULTIMEDIA SYSTEMS AND APPLICATIONS

Multimedia systems and applications deal with combination of various data types including video, graphics, and audio. Encoders compress input video streams by discarding less important information. Decoders decode the compressed video data (Koenen, 2002), (Ely, 1995), (Schaphorst, 1999).

### 3.1 MPEG4 Encoder

The Moving Picture Experts Group (a working group within the ISO) finalized MPEG4 (Part-2) in 1998. MPEG4 delivers professional-quality audio and video streams over a wide range of bandwidths, from cellular phone to broadband and beyond.

The MPEG4 video encoding algorithm achieves very high compression rates by removing both the temporal and spatial redundancy from the motion video. The compressed data is temporarily stored into a buffer to discard the most detailed information and preserve the less detailed picture content to control the transmission rate. The video may be compressed further with an entropy coding algorithm (Ely, 1995). To exploit temporal

redundancy, MPEG4 encoding uses motion compensation with three different types of frames – Intra (I), Predicted (P), and Bidirectional (B). Information not present in reference frames is encoded spatially on a block-by-block basis (Asaduzzaman, 2006), (Wu, 2004), and (Richardson, 2002). The encoding, transmission, and decoding order of the picture frames in a GOP are the same (non-temporal order). But the playback order is different (temporal).

### 3.2 H.264/AVC Encoder

The ISO Motion Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG) have developed Advanced Video Coding (AVC) – widely known as H.264/AVC in 2003. H.264/AVC significantly outperforms both H.263 and MPEG4 by providing high-quality and low bit-rate streaming video.

The encoder includes two dataflow paths – a “forward” path and a “reconstruction” path. The input frame is processed in units of a macro-block (corresponding to 16x16 pixels in the original image). Each macro-block is encoded in intra or inter mode. In either case, a prediction macro-block is formed based on a reconstructed frame. In Intra mode, prediction macro-block is formed from samples in the current frame that have previously encoded, decoded, and reconstructed. In Inter mode, prediction macro-block is formed by motion-compensated prediction from one or more reference frame(s) (Richardson, 2002).

In both MPEG4 and H.264/AVC, there are dependencies among frames during encoding. H.264/AVC provides higher quality and lower bit-rate video than MPEG4 does.

## 4 SIMULATION

In this paper, we focus on evaluating the impacts of level-2 cache on the performance of multimedia systems running MPEG4 and H.264 encoders. The overall performance of such a complex system entirely depends on the hardware/software used to build the system and the target applications.

### 4.1 Assumptions

Following assumptions are made to model and run the VisualSim simulation.

1. We keep CL1 parameters fixed. Only CL2 size is varied during the simulation.
2. The dedicated bus that connects CL1 and CL2 introduces negligible delay compared to the delay introduced by the system bus which connects CL2 and main memory.
3. CPU and CL1 operate at 512 MHz, CL2 at 256 MHz, bus at 128 MHz, and memory (SDRAM) at 64 MHz.
4. Hit ratio of CL1 and CL2 are very high.

### 4.2 Simulated Architecture

The simulated architecture has a single processing core and a memory system with two levels of caches as shown in Figure 2. CL1 is on-chip with the core but CL2 is off-chip. They are connected to main memory via a shared bus. The processing core reads, encodes, and writes the video streams from/into the main memory through its cache hierarchy.

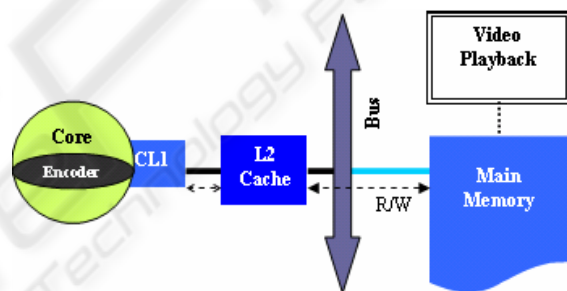


Figure 2: Single-processor architecture with two levels of caches.

Cache size and levels are important memory system design parameters. In this work, we vary CL2 size while measuring performance metrics.

### 4.3 Workload

The quality of the workload used in the simulation is important for the accuracy of the simulation results (Slingerland, 2002), (Avritzer, 2002), and (Maxiaguine, 2004). We choose MPEG4 and H.264/AVC applications because of their popularity. We characterize the workload using ARMulator. We obtain detailed traces using ARMulator and miss rates using C++ code. Detailed traces and miss rates are used to run the VisualSim simulation model. Table 1 shows read and write references at CL2.

Table 1: CL2 references for encoders.

Video Type	CODEC	L2 Refs (K)		L2 Refs	
		Read	Write	R%	W%
MPEG4	FFmpeg	307	153	67	33
H.264/AVC	JM RS (96)	21392	7352	76	24

MPEG4 encoder input file is a raw YUV 4:2:0 video data file of size 1,475 K, the output is a .mp4 file generated by the FFMPEG encoder. Similarly, H.264/AVC encoder input file is the same YUV file, but the output is a .264 file generated by JM-RS (98) encoder.

#### 4.4 Simulation Tools

The simulation tools we use include ARMulator, FFMPEG, JM-RS (96), and VisualSim (ARMulator, 2007), (FFmpeg, 2007), (JM-RS (96), 2007), and (VisualSim, 2007). ARMulator, FFMPEG, JM-RS (96), and C++ are used to characterize the workload of MPEG4 and H.264/AVC. C++ program is developed and run in Microsoft Visual C++ 6.0. VisualSim from Mirabilis Design, Inc is used to develop and run the simulation model.

#### 4.5 VisualSim Model

We develop a VisualSim model to simulate the architecture with a single-core and two-level-cache system. The model of the simulated architecture is presented in Appendix A.

## 5 RESULTS

The focus of this paper is to evaluate the impacts of level-2 cache on the performance of multimedia systems running MPEG4 and H.264/AVC encoders. The simulation results are presented in the following subsections.

### 5.1 Miss Rates

First, we discuss the impact of CL2 size on the miss rates. Our previous work shows that CL2 size smaller than 256K and bigger than 2M does not have significant impact on miss rates. In this work, we keep CL1 cache parameters fixed and vary CL2 size from 256K to 2M. Miss rates for various CL2 size is shown in Figure 3.

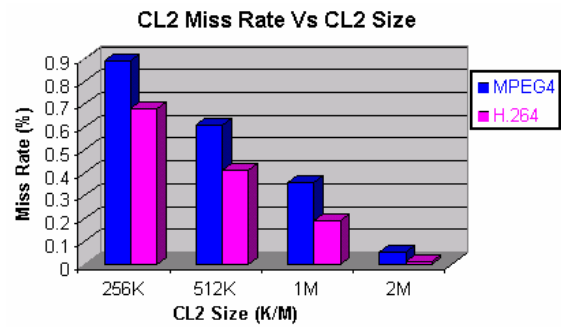


Figure 3: Miss Rates versus CL2 size.

It is observed that for CL2 size from 256K to 2M miss rate decreases sharply. It is also observed that the miss rate of H.264/AVC is smaller when compared with that of MPEG4.

### 5.2 CPU Utilization

CPU utilization is an important performance metrics. The CPU utilization is defined as the ratio of the time that CPU spent computing to the total time required to complete all the tasks. Figure 4 shows the impact of CL2 size variation on CPU utilization (with and without CL2) for MPEG4 and H.264/AVC encoders. CL1 parameters are kept fixed and CL2 size is varied from 256K to 2M. CPU utilization decreases with the increase of CL2 size from 256K to 2M.

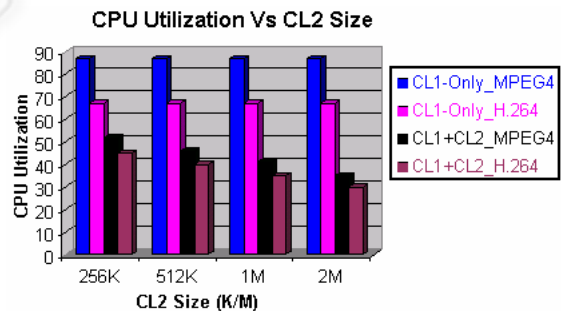


Figure 4: CPU utilization versus CL2 size.

Simulation results show that CPU utilization of H.264/AVC is smaller than that of MPEG4 encoder. The difference of CPU utilizations for smaller CL2 (256K in our simulation) is significant.

### 5.3 Memory System Power Consumption

Power consumption of the memory system due to MPEG4 and H.264/AVC encoder is shown in Figure 5 in terms of power consumption decrease in percentage for various CL2 size. We keep CL1 cache parameters fixed and obtain the total power consumed by the memory system for each CL2 size. This is an activity based power analysis and it is assumed that power required due to a CL2 miss is 100 times more than that required due to a CL2 hit.

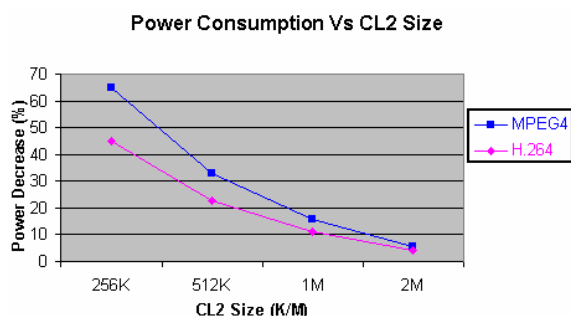


Figure 5: Power decrease (%) versus CL2 size.

Simulation results show that the percentage of decrease in power consumed by the memory system is significant for smaller CL2 cache size. It is also observed that for H.264/AVC, the percentage of decrease in power consumed is smaller when compared with that of MPEG4.

## 6 CONCLUSIONS

Due to their limited resources, multimedia systems may suffer while processing multimedia applications. The demand for tremendous amount of processing power raises serious challenges for multimedia systems and applications. Studies show that cache memory has strong influence on the performance of multimedia systems and applications. In our previous work, we optimize level-1 cache parameters to enhance the performance of portable devices running MPEG4 decoder. In this paper, we focus on evaluating the impacts of level-2 cache on the performance of multimedia systems running MPEG4 and H.264/AVC encoders. We develop VisualSim model and C++ code to run the simulation. We measure miss rates, CPU utilization, and power consumed by varying level-2 cache size. Simulation results show that the performance of multimedia systems and

applications can be enhanced by optimizing level-2 cache.

We plan to investigate the impact of level-3 cache on performance and power of a multimedia system running MPEG4 and H.264/AVC CODEC in our next endeavour.

## REFERENCES

- Asaduzzaman, A., Mahgoub, I., 2004. Evaluation of Application-Specific Multiprocessor Mobile System, SPECTS.
- Grigoriadou, M., Toulas, M., Kanidis, E., 2003. Design and Evaluation of a Cache Memory Simulation Program, Proceedings of the IEEE.
- Slingerland, N., Smith, A., 2005. Cache Performance for Multimedia Applications. Portal ACM, URL: [www.portal.acm.org/ft\\_gateway.cfm?id=377833&type=pdf](http://www.portal.acm.org/ft_gateway.cfm?id=377833&type=pdf)
- Asaduzzaman, A., Mahgoub, I., 2006. Cache Modelling and Optimization for Portable Devices Running MPEG-4 Video Decoder, MTAP06 Multimedia Tools and Applications.
- Wu, Z., Tokumitsu, M., Hatanaka, T., 2004. The Development of MPEG4-AVC/H.264, the Next Generation Moving Picture Coding Technology, Oki Technical Review.
- Richardson, I., 2002. H.264 / MPEG4 Part 10: Overview, URL: [www.vcodex.com](http://www.vcodex.com)
- Koenen, R., Pereira, F., Chiariglione, L., 2002. MPEG4: Context and Objectives, Invited paper for the Special Issue on MPEG4 of the Image Communication Journal.
- Ely, S., 1995. MPEG video coding - A simple introduction, EBU Technical Review Winter.
- Schaphorst, R., 1999. Videoconferencing and Videotelephony – Technology and Standards, Artech House.
- Soderquist, P., Leeser, M., 1997. Optimizing the Data Cache Performance of a Software MPEG-2 Video Decoder, ACM Multimedia 97 – Electronic Proceedings.
- Wolf, W., 2005. Multimedia Applications of Multiprocessor Systems-on-Chips. IEEE DATE'05 Proceedings.
- Wolf, W., Kandemir, M., 2003. Memory System Optimization of Embedded Software, Proceedings of the IEEE.
- Li, Y., Henkel, J., 1998. A framework for estimating and minimizing energy dissipation of embedded HW/SW systems. Proc. 35th Design Automation Conf.
- Panda, P., Kjeldsberg, P., et al, 2001. Data and Memory Optimization Techniques for Embedded Systems, ACM Transactions on Design Automation of Electronic Systems.
- Kannan, S., Allen, M., Fridman, J. Cached Memory Performance Characterization of a Wireless Digital Baseband Processor, IEEE V-361-364.

Slingerland, N., Smith, A., 2002. Design and characterization of the Berkeley multimedia workload, Multimedia Systems, Springer-Verlag.

Avritzer, A., Kondek, J., Liu, D., Weyuker, E.J., 2002. Software Performance Testing Based on Workload Characterization, WOSP'02, AT&T Labs, ACM

Maxiaguine, Liu, Y., Chakraborty, S., Ooi, W., 2004. Identifying "representative" workloads in designing MpSoC platforms for media processing, 2nd Workshop on Embedded Systems for Real-Time Multimedia.

ARMulator, 2007. Application Note 32: The ARMulator, www.arm.com/support/ARMulator.html

VisualSim, 2007. A system-level simulator from Mirabilis Design, Inc. www.mirabilisdesign.com

FFmpeg, 2007. A very fast video and audio converter, http://ffmpeg.sourceforge.net/ffmpeg-doc.html#SEC1

JM-RS (96), 2007. H.264/AVC Reference Software http://iphone.hhi.de/suehring/tml/download/

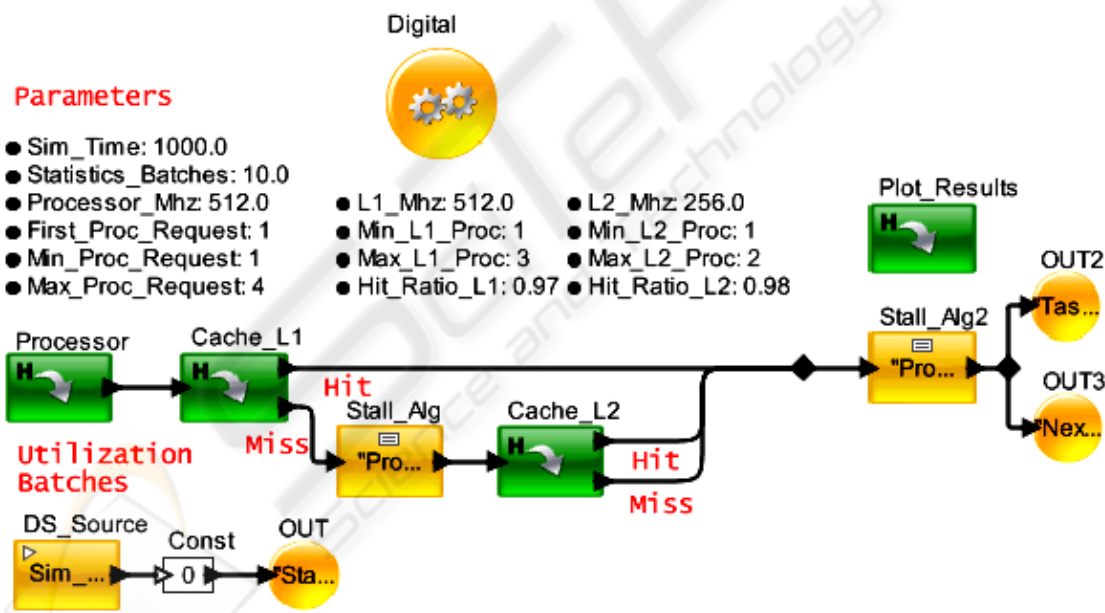
## APPENDIX

### Visualsim Block Diagram

In this work, we use VisualSim simulation tool from Mirabilis Design, Inc. (URL: www.mirabilisdesign.com/).

## Visualsim Model: Simulated Architecture

This simulation model explores the impact of level-2 cache parameters on performance of multimedia systems and applications.



In VisualSim, a system to be evaluated can be described in three parts – Architecture, Behaviour, and Workload. Architecture: elements such as Processor and cache. Behaviour: this describes the actions performed on the system. Workload: transactions that traverse the system. Mapping between behaviour and architecture is performed using Virtual Execution. Connection can be made using dedicated and/or Virtual Connections. The virtual execution capability makes re-mapping from hardware to software by just changing a parameter. The output of a block can be displayed or plotted. The Simulation Cockpit (not shown here) provides functionalities (Go, Pause, Resume, and Stop) to run the model (block diagram) and to collect simulation results. Parameters can be changed before running the simulation without modifying the block diagram. The final results can be saved into a file and/or printed for further analysis.