A Framework for Video Application in the Embedded System through Rearrangement of Cache Memory Hierarchy

Jabbar T. K.

STIC Laboratory, University of Tlemcen, Algeria

ABSTRACT

Now days embedded systems is extremely used for running period of time video applications and it\'s additionally growing quite ever before. so the challenge for to run transmission applications on the embedded design increased; as a result of the embedded system has terribly restricted resources. The Cache memory is getting used to enhance the performance by bridging the speed gap between the process core and therefore the main memory. However, the matter of adopting the cache memory into embedded systems is twofold. First, cache is power hungry – that challenges the energy and thermal constraints. Second, the cache introduces execution time unpredictability – that challenges the supporting period of time applications. during this paper explore impact of the cache parameters and cache protection on the foregone conclusion, power consumption, and therefore the performance for the period of time embedded applications. This work simulate a universally used Intel Pentium-like the single-core design that has 2 levels of the cache memory hierarchy running 2 video applications, MPEG-4 (the world video committal to writing normal for transmission applications) and H.264/AVC (the network friendly video committal to writing normal for colloquial and non-conversational applications). Experimental results show that cache protection mechanism side to associate optimized cache memory structure is extremely promising to enhance the foregone conclusion of embedded MPEG-4 and H.264/AVC with none negative impact on the performance and total power consumption. it\'s additionally ascertained that H.264/AVC has performance advantage over MPEG-4 for smaller caches.

Keywords: MPEG, cache memory, memory, vided files, embedded system.

1. INTRODUCTION

Billions of embedded systems are sold-out annually. The growing quality of embedded systems brings challenges to the designers to feature a lot of options to support period video applications. The freshly additional functionalities increase the quality and area-size of embedded systems. These devices ar expected to perform period video algorithms with efficiency to support these applications and meet low power and information measure needs. It becomes obvious that a lot of machine power is needed to subsume these needs. Increased machine power implies a lot of traffic from electronic equipment to main memory. The memory information measure isn't increasing as quick because the increase in process speed, resulting in a big processor-memory speed gap. a typical observe to subsume memory information measure bottlenecks is to use cache memory – a really quick and tiny however pricey memory. Cache runs at a speed nearly as quick as electronic equipment speed. Cache improves system performance by reducing the effective access time [1 – 8]. though multicore design is that the new style trend, single-core processor can stay because the most suitable option for a few embedded systems. Most modern processors have caches. The cache memory hierarchy ordinarily has level-1 cache (CL1), level-2 cache (CL2), and main memory. CL1 is typically split into instruction (I1) and knowledge (D1) caches and CL2 may be a unified cache. additionally to the electronic equipment and memory hierarchy, there ar interfaces that alter the embedded system to act with the external setting [17, 18].

Even though cache improves performance, cache consumes extra energy and introduces execution time unpredictability as a result of its accommodative and dynamic nature [9 – 13]. Extra energy demand becomes crucial for embedded systems, particularly after they ar operated by battery power. Execution time foregone conclusion is extremely crucial to develop any mission-critical period embedded systems. Victimization high performance processors in embedded systems to support period applications poses style challenges as embedded systems suffer from restricted resources like energy offer. For embedded systems supporting period video applications, there's no undemanding thanks to decide the trade-off among performance, power consumption, and foregone conclusion [14 – 16]. In general, an easy design that may support transmission applications victimization minimum energy is required to style such AN embedded system [19 – 22].

Improving foregone conclusion while not decreasing performance and/or while not intense extra power is extremely difficult. On the one hand, cache improves overall system performance. However on the opposite hand, cache will increase overall energy consumption. Recent studies show that cache protection could improve execution time foregone conclusion [10 – 12]. However, aggressive cache protection could scale back performance and increase energy consumption. In [9], level-1 instruction cache (I-Cache) protection in AN Intel Pentium-like single-core design is simulated victimization FFT, MI, and DFT work. Experimental results show that cache protection improves each foregone conclusion and performance up to a limit. at the moment limit, foregone conclusion are often more improved by sacrificing the performance.

In this work, we tend to explore the impact of cache parameters and cache protection on the foregone conclusion, power consumption, and performance for period embedded applications. we tend to think about AN Intel-like single-core design that has 2 levels of cache memory hierarchy. we tend to use 2 representative period applications, namely, MPEG-4 and H.264/AVC to run the simulation program. ISO normal MPEG-4 (a.k.a. MPEG4 (Part-2)) is that the international video committal to writing normal for multimedia applications [48] and ITU-T commonplace H264/AVC (a.k.a. MPEG4 (Part-10)) is that the network friendly video cryptography commonplace for colloquial (video telephone) and non-conversational (storage, broadcast, or streaming) applications [49]. This simulation platform are often reused for analyzing multicore embedded systems with some modifications.

The define of this paper is as follows. connected work is summarized in Section a pair of. Section three shortly discusses video CODEC (enCOder and DECoder pair). Section four shortly reviews vital cache parameters and their influence on performance, power, and foregone conclusion. In Section five, simulation details ar given. Some vital results ar mentioned in Section half-dozen. Finally, we tend to conclude our add Section seven.

2. LITERATURE SURVEY

Designing effective cache memory hierarchy for embedded systems may be a nice got to support period multimedia system applications. plenty of labor has been done to boost the performance, power consumption, and/or foregone conclusion in single-core systems. a number of them, we discover relevant to the present work, ar mentioned during this section.

Cache modeling and optimisation is conducted for MPEG-4 video decoder in [2]. The target design includes a process core to run the decryption formula and a memory hierarchy with two-level caches. each level-1 and level-2 caches ar thought-about to be unified caches. Cache parameters together with cache size, line size, associativity level, and cache level ar optimized to boost system performance. Cache miss rates ar measured victimisation Cachegrind and VisualSim simulation tools for varied cache parameters. each Cachegrind and VisualSim experiments show that performance is improved by lowering the miss rates by optimizing CL1 line size, CL1 associativity level, and CL2 cache size for MPEG-4 decoder. In [18], a general computing platform running MPEG-2 application is studied. The impact of cache parameters on performance (in terms of miss rate and memory traffic) is evaluated during this work. Experimental results show that cache improves MPEG-2 performance on such a system. Authors counsel that the understanding of application behavior will facilitate to boost cache potency. In [24], the matter of up performance of the memory hierarchy at system level for multitasking knowledge intensive applications is addressed . This technique uses cache partitioning techniques to seek out a static task execution order for inter-task knowledge cache misses. attributable to the shortage of freedom in rearrangement task execution, this technique improves performance by optimizing the caches additional. In [26], a access analysis is bestowed by learning the vector operations of explicit code segments which will be accustomed justify the cache memory design. on top of mentioned articles, [2][18][24][26], gift totally different cache optimisation techniques to boost the performance. However, these articles don't cowl the foregone conclusion and/or power consumption analysis that are necessary for period embedded systems.

Power consumption may be a crucial style issue for embedded systems, particularly for those who ar battery operated. A victim buffer (between CL1 and CL2) is introduced to boost the ability consumption in [25]. rather than discarding the victim blocks (from CL1) which may be documented within the close to future, victim blocks ar keep in victim buffer. Experimental results show that victim buffer will scale back energy by forty third on PowerStone and MediaBench benchmarks. In [27], cache reminiscences for embedded applications ar designed in such the way that they increase performance and scale back energy consumption. it's shown that separating knowledge cache into Associate in Nursing array cache and a scalar cache will result in vital performance enhancements for scientific benchmarks. Such a split knowledge cache can even profit embedded applications. In [29], a system-level power aware style flow is planned so as to avoid failures when months of style time spent at register transfer level gate level. the tactic in [30] uses a ballroom dancing approach – initial collects immediate knowledge concerning the applying so uses equations to predict the performance and power consumption of every of the potential configurations of the system parameters. In [31], Associate in Nursing analytical model for power estimation and average access time is bestowed. Articles in [25][27][29][30][31] discuss however overall energy consumption is reduced by optimizing varied cache parameters. additionally to energy consumption, a number of these techniques, [27][30], cowl performance analysis. However, none of those techniques discusses however execution time foregone conclusion (a crucial style issue for period systems) is improved.

For period systems, the hardware and computer code ar subject to operational deadlines from event to system response and therefore the execution time should be certain. it's been planned that necessary cache contents ought to be statically fast in cache in order that access time and cache-related preemption delay ar certain [9 – 12]. In [9], the impact of cache lockup on the foregone conclusion and performance is studied. Experimental results show that cache lockup improves each foregone conclusion and performance once the suitable cache parameters ar used and therefore the right cache blocks ar fast. foregone conclusion is any improved by sacrificing performance. varied techniques ar planned to pick out the blocks for lockup. as an example, genetic formula is employed in [12]. the key disadvantage of those articles is that no analysis is finished to indicate however power consumption is wedged. In [28], a technique is given to chop-chop notice the L1 cache miss rate of Associate in nursing application. Also, Associate in Nursing energy model Associate in Nursingd an execution time model are developed to seek out the most effective cache configuration for the given embedded application. However, this work doesn't supply any methodology to boost the foregone conclusion, power consumption, and foregone conclusion.

Performance, power consumption, and foregone conclusion ar necessary style factors for period embedded systems. On the one hand, cache improves overall system performance, however on the opposite hand, cache consumes additional energy and makes the temporal order unpredictability worse. Therefore, it's necessary to research the performance, power consumption, and foregone conclusion along for embedded systems. during this work, we have a tendency to study the impact of cache parameters and cache lockup on the foregone conclusion, power consumption, and performance victimisation period video CODEC.

3. ARCHITECTURE OF VIDEO FORMATS

In this paper, we have a tendency to examine the impact of cache parameters and cache protection on the foregone conclusion, power consumption, and performance for period of time applications. we have a tendency to use 2 multimedia system encoder and decoder pairs (CODEC) during this experiment, namely, MPEG-4 (a.k.a., MPEG-4 Part-2) and H.264/AVC (a.k.a., MPEG-4 Part-10). MPEG-4 is that the world video writing commonplace for multimedia system applications and H.264/AVC is that the network friendly video writing commonplace for informal (e.g. video conferencing) and non-conversational (e.g. video streaming) applications. Encoders compress input video streams and decryptrs decode the compressed video knowledge. There square measure dependencies among video frames throughout cryptography and cryptography video files [28, 33, 34]. owing to the dependencies among frames, the correct choice of cache parameters might considerably improve the foregone conclusion and performance for smallest energy consumption. Also, the blocks that cause cache miss are often barred within the cache for the whole execution time to boost foregone conclusion. With this regards, we have a tendency to shortly discuss video CODEC algorithms within the following subsections.

MPEG-4 delivers professional-quality audio and video streams over a large vary of bandwidths, from cell phone to broadband and on the far side. A MPEG-4 sequence video clip consists of the many teams of images (GOP). A GOP contains a minimum of one intra (I) frame and usually variety of dependent expected (P) and two-way (B) frames [2, 19, 48]. I frames contain an entire image. P frames consist primarily of pixels from the nighest previous I or P frame. B frames use the nighest I or P frames, one before and one when in temporal arrangement, as reference frames. Figure one shows the dependencies among the frames of a MPEG-4 GOP with seven frames. Within the following subsections, we have a tendency to concisely discuss the influence of cache memory on MPEG-4 CODEC (encoder and decoder pair).

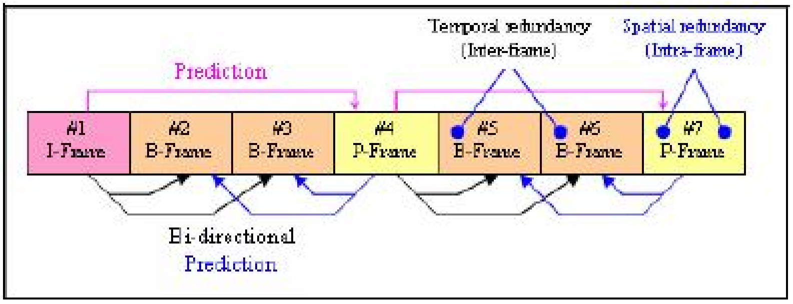

Figure 1: The frames architecture of an MPEG-4 group of pictures (GOP)

MPEG-4 coding performance is improved mistreatment cache memory. The MPEG-4 video coding formula achieves terribly high compression rates by removing each the temporal redundancy (that happens among totally different frames) and spatial redundancy (that happens inside a similar frame) from the motion video as shown in Figure one. The compressed knowledge is quickly hold on into a buffer to discard the foremost elaborated info and preserve the less elaborated image content to regulate the transmission rate. The video is also compressed more with associate entropy writing formula [32]. info not gift in reference frames is encoded spatially on a block-by-block basis [2, 20]. The coding order of the frames during a GOP is non-temporal (for Figure one, the coding order is #1, #4, #2, #3, #7, #5, and #6).

As a result, varied cache parameters influence the foregone conclusion, power consumption, and performance as they're employed by the encoder.

MPEG-4 decoder decodes the compressed video knowledge that may be displayed shortly [2, 19]. Like coding, cryptography order of the frames during a GOP is non-temporal. As a result, the presence of cache (cache parameters and cache locking) influences the foregone conclusion, power consumption, and performance of MPEG-4 decoder.

H.264/AVC (a.k.a. MPEG4 Part-10)) guarantees to considerably beat each of its folks, H.263 and MPEG-4 (a.k.a. MPEG4 Part-2)), by providing high-quality and low bit-rate streaming video. H.264/AVC CODEC includes 2 dataflow ways – a “Forward” path (encoder – left to right) and a “Reconstruction” path (decoder – right to left) [35, 36, 49]. The schematic diagram in Figure two shows vital parts of H.264/AVC CODEC formula. in contrast to MPEG-4, H.264/AVC defines the syntax of associate encoded video bit-stream in conjunction with the tactic of cryptography this bit-stream. within the following subsections, we tend to shortly discuss the influence of cache memory on H.264/AVC CODEC (encoder and decoder pair).

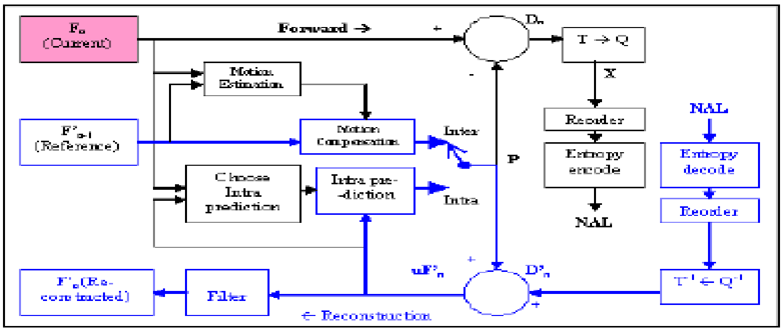

Figure 2: The components of H.264/AVC encoder and decoder algorithms

Like MPEG-4 encoder, H.264/AVC cryptography performance will be improved by with efficiency exploitation the cache memory. In Figure a pair of, Associate in Nursing input frame Fn is given for cryptography. The frame is also processed in intra or inhume mode. In Intra mode, P is created from samples within the current frame Fn that have antecedently encoded, decoded and reconstructed (uF’n). In inhume mode, P is created by motion-compensated prediction from one or a lot of reference frame(s) [36]. Therefore, the certainty, power consumption, and performance of H.264/AVC encoder square measure influenced by the presence of the cache (cache parameters and cache locking).

In H.264/AVC, the decipherment methodology is outlined with the syntax of Associate in Nursing encoded video bit-stream. The decoder receives an encoded bit-stream from the Network Abstraction Layer (NAL). As shown in Figure two, the info components area unit entropy decoded and reordered to supply a group of quantity coefficients (X). exploitation the header data from the bit-stream, the decoder creates a prediction macro-block P, a twin of the first prediction P fashioned within the encoder. P is additional to D’n to supply uF’n that is filtered to form the decoded macro-block F’n. As a result, the cache parameters and cache protection influence the certainty, power consumption, and performance of H.264/AVC decoder.

Because of the dependencies among the video frames throughout secret writing and coding video files, it's well worthy to examine the impact of cache parameters and cache protection on the certainty, power consumption, and performance for MPEG-4 and H.264/AVC CODEC before creating any style choices.

4. Issue of Cache Memory

It has been established that cache parameters and cache protection have vital impact on cache miss rate. Therefore, the study of the combined impact of standardization cache parameters and applying cache protection on period embedded multimedia system systems ought to be useful to style such a fancy system. during this work, we tend to explore the impact of cache parameters and cache protection on the sure thing, power consumption, and performance for MPEG-4 and H.264/AVC CODEC (encoder and decoder) running on embedded systems. process core uses its cache memory hierarchy to scan unprocessed video information from main memory and to write down processed video information into main memory. the choice of cache parameters and therefore the quantity of cache fastened is extraordinarily necessary to realize the optimum sure thing, power consumption, and performance. With this regards, we tend to in short discuss the cache parameters and cache protection techniques within the following subsections.

Level-1 cache is 1st introduced by IBM in late Sixties to enhance performance by reducing the speed gap between the mainframe and therefore the main memory. Today, most processors area unit having on-chip CL1 and off-chip CL2. The schematic diagram in Figure three shows cache blocks or lines (line size = Sb), associativity levels (W), sets (S), and mapping techniques. For a cache with S sets and W levels of associativity, total range of blocks (B) = S * W and total cache size is B * Sb. Cache will operate as direct mapped (W = one and S = B), set-associative (1 &lt; W &lt; B), and fully-associative (W = B and S = 1).

Even though cache improves overall system performance, it takes further energy to be operated and it makes the system additional unpredictable attributable to its dynamic behavior [9 – 12]. So, adding cache recollections to embedded systems is problematic, particularly once the system is battery operated and runs period of time applications [44 – 46]. The influence of the vital cache parameters on the sure thing, power consumption, and performance is mentioned below.

Cache Size: Larger cache permits additional video information become nearer to the methoding unit|CPU|C.P.U.|central processor|processor|mainframe|electronic equipment|hardware|computer hardware} in order that processor will process them quickly. Studies show that for MPEG-2, the increasing cache size offers performance improvement [18]. However, once a degree, the increment isn't important. during this work, we have a tendency to examine the impact of CL2 cache size on miss rate and total power consumption for MPEG-4 and H.264/AVC CODEC.

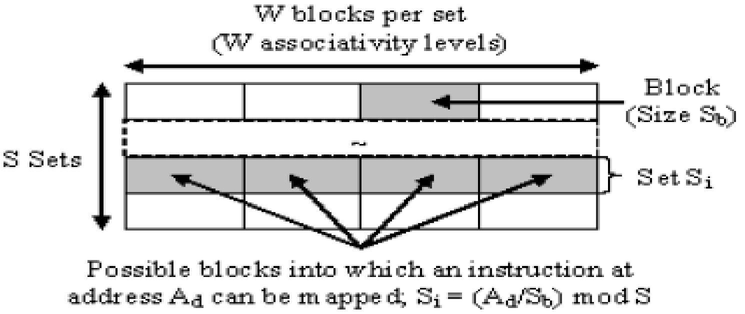
W blocks per set
(W associativity levels)
Block
(Size S_b)
S Sets
~
Set S_i
Possible blocks into which an instruction at
address A_d can be mapped; $S_i = (A_d/S_b) \bmod S$

Figure 3: Architecture of cache memory

Line Size: Larger line size tends to supply low miss rates, however need additional information to be scan and probably written back on a cache miss. For video information, overlarge line size could introduce cache pollution and reduce performance. during this work, we have a tendency to study the impact of line size on miss rate and total power consumption for MPEG-4 and H.264/AVC CODEC.

Associativity Level: just in case of direct mapped or set associative block replacement strategy, conflict misses occur once many blocks area unit mapped to an equivalent set or block frame. Studies indicate that higher associativity could improve each performance and power consumption by reducing the conflict misses [18]. For video information, aggressive increment in associativity level might not improve overall performance attributable to the redoubled complexness. during this work, we have a tendency to examine the impact of CL1 associativity level on miss rate and total power consumption for MPEG-4 and H.264/AVC CODEC.

Cache lockup may be a technique to carry sure memory blocks within the cache throughout the complete period of the execution time. The memory blocks could also be pre-selected or willy-nilly chosen (and pre-loaded) so as to boost hit rates. Once such a block is loaded into the cache, the replacement algorithmic rule excludes them to be removed till the appliance is completed. Studies show that cache lockup improves sure thing in embedded systems running period of time applications [10, 12]. However, aggressive cache lockup could decrease the performance by increasing cache misses attributable to the reduction in economical cache size. during this paper, we have a tendency to explore the impact of cache parameters and cache lockup on the sure thing, power consumption, and performance for period of time embedded applications. we have a tendency to take into account solely level-1 instruction cache lockup and that we willy-nilly choose the memory blocks to be latched within the cache. Recently printed work shows that performance will increase considerably with the rise within the latched cache size for smaller values (0% to twenty fifth of the cache size) [9]. once twenty fifth latched cache size, performance starts decreasing with the rise of the cache lockup capability.

5. PROPOSED WORK

In this paper, we tend to investigate the impact of cache parameters on the certainty, power consumption, and performance for period of time embedded CODEC. Simulation is verified to be a good technique to explore such a fancy system [39]. we tend to simulate AN Intel Pentium-like design that has one process core and 2 levels of cache memory hierarchy. we tend to run the simulation program mistreatment MPEG-4 and H.264/AVC employment. within the following subsections, we tend to discuss simulation tools, simulated design, assumptions, and employment associated with this work.

5.1 Simulation Tools

We use 2 simulation tools, namely, Cachegrind and VisualSim. mistreatment Cachegrind, we tend to produce employment for the chosen applications. we tend to collect memory references for level-1 (instruction/data), D1 (read/write), and level-2 (read/write) caches. mistreatment VisualSim, we tend to model the target design and run the simulation program. employment we tend to produce mistreatment Cachegrind is employed to run VisualSim simulation program.

Cachegrind (from Valgrind) may be a cache profiler [41]. we tend to use Cachegrind to perform elaborated simulation of level-1 and level-2 caches. Total references, misses, and miss rates for I1, D1, and level-2 caches square measure collected mistreatment Cachegrind. With Cachegrind, we tend to use FFmpeg to characterize MPEG-4 and JM-RS (98) to characterize H.264/AVC encoding/decoding algorithms [42-43]. mistreatment Cachegrind, we tend to acquire cache miss rates for varied cache parameters.

VisualSim (from caryophylloid dicot genus Design) may be a graphical and hierarchical modeling tool, effective to simulate system level design of embedded systems and period of time applications [40]. VisualSim provides block libraries for varied system parts as well as process core, cache, bus, and main memory. VisualSim simulation cockpit provides functionalities to run the model and to gather simulation results. mistreatment VisualSim, we tend to acquire total power consumed by the system for varied cache parameters and cache protection capacities.

5.2 Simulated design

In this paper, we tend to specialize in exploring the impact of cache parameters and cache protection on the certainty, power consumption, and performance for period of time embedded CODEC. despite the fact that multicore design improves performance/power quantitative relation, it makes the system a lot of complicated and for such a fancy system, it becomes terribly tough to specialize in the impact of cache parameters and cache protection on the certainty, power consumption, and performance. to create a sound analysis, we tend to simulate AN Intel Pentium-like design that features a single process core and a memory hierarchy with 2 levels of caches. The key parts of the simulated design square measure shown within the schematic diagram in Figure four. CL1 is split into information (D1) and instruction (I1) caches and CL2 may be a unified cache. Process core encodes or decodes the video streams from the most memory. The core reads the raw/encoded information from (and writes the encoded/decoded information into) the most memory through its cache memory hierarchy. We tend to keep the design setup fastened and run MPEG-4 and H.264/AVC encoders and decoders one at a time.

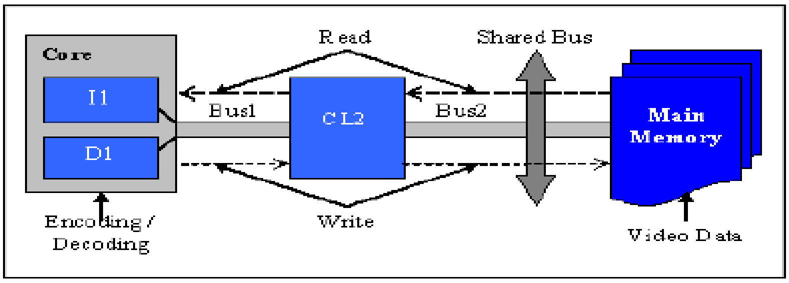

Figure 4: The components of the simulated architecture

We develop the simulation model and run the simulation program mistreatment VisualSim to get level-1 and level-2 cache miss rates and total power consumed by the system. we tend to use MPEG-4 and H.264/AVC employment to run the simulation program.

5.3 Workload

We choose 2 representative period of time applications, MPEG-4 and H264/AVC, to run our simulation program. MPEG-4 is that the international video committal to writing normal for transmission applications and H264/AVC is that the network friendly video committal to writing normal for colloquial and non-conversational applications. employment characterization is vital as a result of the standard of the employment utilized in the simulation determines the accuracy and completeness of the simulation results [37 – 39]. Cachegrind may be a well-known and wide used cache profiler. we tend to characterize each applications mistreatment Cachegrind package (with FFmpeg and JM-RS (96)) and build employment to capture all attainable situations that the target design can expertise.

5.3.1 MPEG-4 and H.264/AVC Encoder employment

Using Cachegrind, we tend to acquire the whole range of references for I1, D1 (read and write), and level-2 (read and write) caches mistreatment totally different cache parameters. Tables 1A and 1B show MPEG-4 and H.264/AVC encryption employment for I1 size sixteen computer memory unit, D1 size sixteen computer memory unit, level-2 size one MB, line size sixty four computer memory unit, and associativity level 4-way. A raw YUV 4:2:0 video file of size one,475K computer memory unit is employed because the computer file to FFmpeg for MPEG-4 encoder and to JM-RS (96) for H.264/AVC encoder [2, 42, 43]. FFmpeg (MPEG-4) generates .mp4 computer file. Similarly, JM-RS (96) (H.264/AVC) generates .264 computer file.

Table 1A: Level-1 instruction and data and level-2 references for encoders

Standard	**Codec**	**Total CL1 References (in Kilo)**		**CL1 Refs**		**CL2 Refs (in Kilo**
		I1	**D1**	**I1%**	**D1%**	
MPEG-4	FFMPEG	132,718	53,858	71	29	460
H.264/AVC	JM RS (96)	11,287,547	7,084,246	61	39	28,615

Table 1B: Level-1 data (read/write) and level-2 (read/write) references for encoders

Standard	**Codec**	**D1 References (Kilo)**		**D1 Refs**		**CL2 Refs (Kilo)**		**D1 Refs**	
		Read	**Write**	**R%**	**W%**	**Read**	**Write**	**R%**	**W%**
MPEG-4	FFMPEG	42,300	11,558	78	22	307	153	67	33
H.264/AVC	JM RS (96)	5,617,714	1,466,532	79	21	21,372	7,243	75	25

5.3.2 MPEG-4 and H.264/AVC Decoder employment

We use the .mp4 file of size 97K computer memory unit (previously generated mistreatment FFmpeg) to run MPEG-4 decoder and .264 file of size 17K computer memory unit (previously generated mistreatment JM-RS (96)) to run H.264/AVC decoder. each FFmpeg (MPEG-4) and JM-RS (96) (H.264/AVC) generate .yuv output files. Like encoders, we tend to acquire the whole range of references for I1, D1 (read and write), and level-2 (read and write) caches mistreatment totally different cache parameters for each decoders. Tables 2A and 2B show MPEG-4 and H.264/AVC decipherment employment for I1 size sixteen computer memory unit, D1 size sixteen computer memory unit, level-2 size one MB, line size sixty four computer memory unit, and associativity level 4-way.

Table 2A: Level-1 instruction and data and level-2 references for decoders

Standard	**Codec**	**Total CL1 References (in Kilo)**		**CL1 Refs**		**CL2 Refs (in Kilo**
		I1	**D1**	**I1%**	**D1%**	
MPEG-4	FFMPEG	40,157	21,363	65	35	259
H.264/AVC	JM RS (96)	120,163	74,159	62	38	191

Table 2B: Level-1 data (read/write) and level-2 (read/write) references for decoders

Standard	**Codec**	**D1 References (Kilo)**		**D1 Refs**		**CL2 Refs (Kilo)**		**D1 Refs**	
		Read	**Write**	**R%**	**W%**	**Read**	**Write**	**R%**	**W%**
MPEG-4	FFMPEG	15,142	6,221	71	29	128	131	49	51
H.264/AVC	JM RS (96)	55,765	18,394	75	25	161	30	84	16

5.4 Assumptions

Following assumptions square measure created to model the target design and to run the simulation program.

- Simulated design consists of a single-core and a memory system with 2 levels of caches. CL1 is split into I1 and D1 caches and CL2 may be a unified cache.
- The dedicated bus that connects CL1 and CL2 (Bus1 in Figure 4) introduces negligible delay compared to the delay introduced by the system bus that connects CL2 and main memory.
- Popular MPEG-4 and H.264/AVC video applications square measure chosen to run the simulation program. MPEG-4 is that the international video committal to writing normal for transmission applications and H.264/AVC is that the network friendly video committal to writing normal for colloquial and non-conversational applications.
- Random cache block replacement policy and write-back memory update strategy square measure used.
- Line size from sixteen to 256 computer memory unit, associativity level from 2- to 16-way, level-2 cache size from 128 computer memory unit to four MB, and cache protection capability from 1/3 (no locking) to five hundredth square measure used.
- Only level-1 instruction cache protection is take into account and cache blocks square measure designated every which way for cache protection.
- For cache protection functions, 16-way set-associative cache mapping and method cache protection is taken into account. Therefore, the minimum portion of the cache that may be fastened is 1/16 (i.e., 6.25%).

6. Analysis of Result

H.264/AVC is additional processor intensive than MPEG-4 to inscribe, so H.264/AVC ought to do additional computing to rewrite. However, H.264/AVC encoder makes smaller files, so H.264/AVC ought to place less strain on the device. As way because the impact of cache parameters on the sure thing, power, and performance is bothered, it's troublesome to take a position that H.264/AVC is healthier than MPEG-4 or vice versa; solely experiments could facilitate to form some realistic suggestions. during this work, we have a tendency to conduct model to look at the impact of cache parameters on the sure thing, power, and performance for period embedded CODEC. The simulated design has 2 levels of cache system; the level-1 cache is split into information and instruction caches and also the level-2 cache may be a unified cache. we have a tendency to acquire the cache miss rates (to represent the performance), and power consumption for numerous cache size, line size, associativity level, and level-1 instruction cache lockup capability. within the following subsections we have a tendency to discuss some necessary experimental results. First, we have a tendency to gift the impact of cache parameters on miss rates and total power consumption for the encoders and decoders. Then, we have a tendency to gift the impact of cache lockup on miss rates and total power consumption for the encoders and decoders.

6.1 Impact of Cache Parameters

In this following section, we have a tendency to gift the impact of cache parameters (CL1 line size, CL1 associativity level, and CL2 cache size) on miss rates for MPEG-4 and H.264/AVC encoders and decoders.

6.1.1 Miss Rates for Encoders

First, we have a tendency to gift the simulation results obtained for MPEG-4 and H.264/AVC encoders. exploitation Cachegrind, we have a tendency to acquire CL1 miss rates for various CL1 line sizes. As shown in Figure five, miss rate starts decreasing with the rise of line size as a result of required cache misses (compulsory miss happens on the primary access to a block) decrease as line size will increase. In our experiment, miss rates keep identical or increase for line size larger 128 computer memory unit. this is often as a result of large a line size could increase capability cache misses (capacity miss happens as a result of blocks area unit discarded from cache as cache cannot contain all blocks required for program execution). Also, the miss rates for H.264/AVC encoder area unit smaller than those for MPEG-4 encoder.

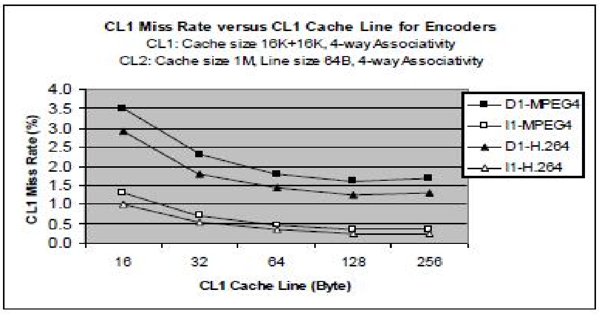

Figure 5: CL1 Miss Ratio Vs CL1 cache line for encoders

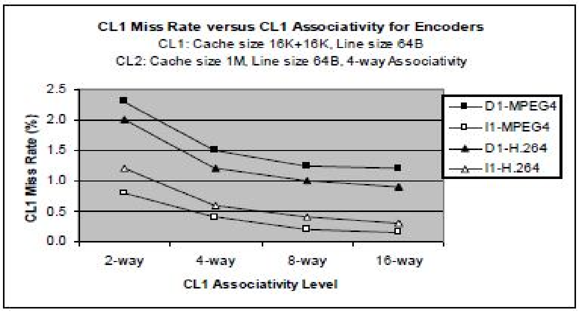

Figure 6. CL1 Miss Ratio Vs CL1 associativity level for encoders

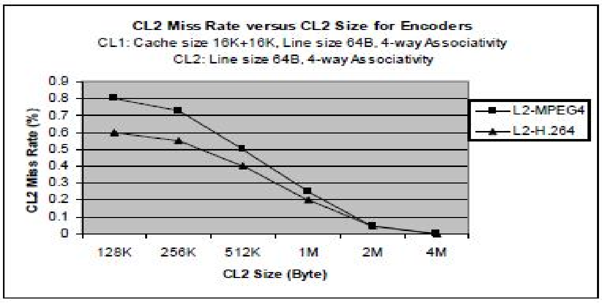

Figure 7: CL1 Miss Ratio Vs CL1 associativity level for encoders

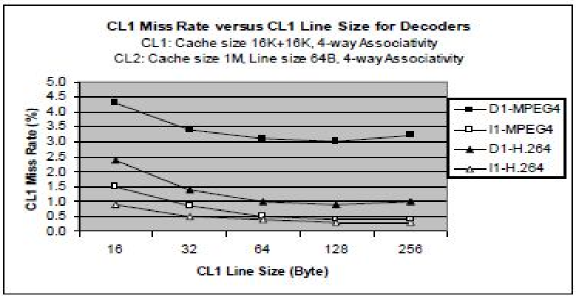

Figure 8: CL1 Miss Rate Vs CL1 cache line for decoders

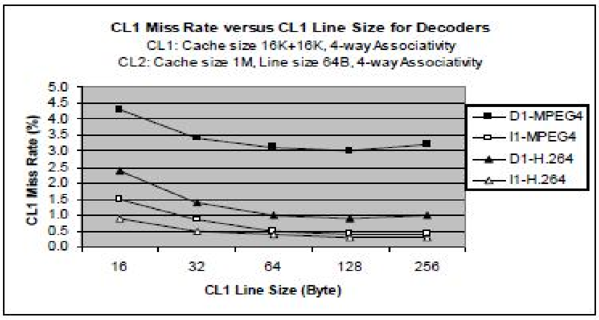

Figure 9: CL1 Miss Rate Vs CL1 associativity for decoders

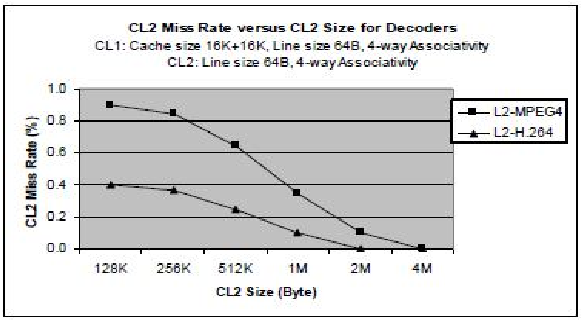

Figure 10: CL2 Miss Rate Vs CL2 cache size for decoders

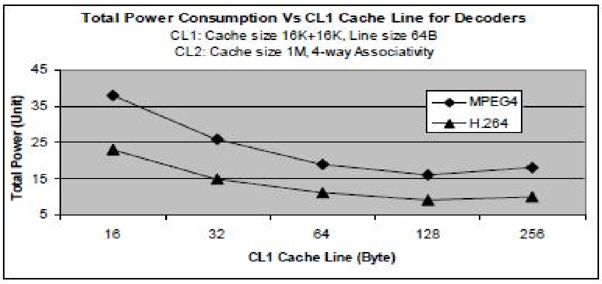

Figure 11: Total power consumed by the decoding system size

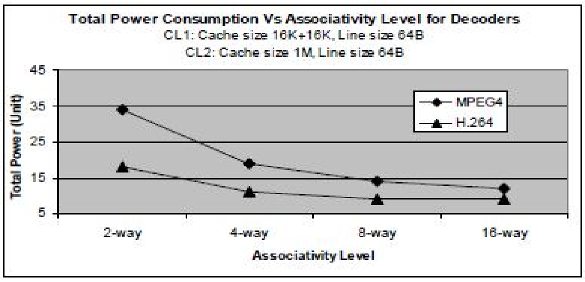

Figure 12: Total power consumed by the decoding system Vs Vs Line Associativity level

Using Cachegrind, we have a tendency to acquire CL1 miss rates by variable CL1 associativity [see Figure 6]. As illustrated within the figure, the miss rates considerably decrease from 2-way to 4-way. this is often as a result of the multiplied degree of associativity reduces the conflict cache misses (conflict cache misses occur once many blocks area unit mapped to identical set or block). However, for higher associativity (more than 8-way) the changes in miss rates aren't vital. Results for each MPEG-4 and H.264/AVC encoders follow identical pattern. Results, however, show that the miss rates for H.264/AVC encoder area unit smaller than those for MPEG-4 encoder. this is often in all probability as a result of H.264/AVC work is a smaller amount nerve-wracking.

CL2 miss rates for MPEG-4 and H.264/AVC as a result of the variation of CL2 cache size area unit shown in Figure seven. it's determined that for CL2 size from two56K computer memory unit to 2M computer memory unit the decrease in miss rates is incredibly sharp; for alternative sizes (smaller than 256 kilobyte or larger than 2 MB) the decrease within the miss rates isn't vital. it's conjointly determined that the miss rates for H.264/AVC encoder area unit smaller than those for MPEG-4 encoder. The distinction between MPEG-4 and H.264/AVC miss rates is important for smaller CL2 (between 128 kilobyte and 512 KB).

6.1.2 Miss Rate for Decoders

Second, we have a tendency to gift the simulation results obtained for MPEG-4 and H.264/AVC decoders. exploitation Cachegrind, CL1 miss rates area unit obtained for numerous CL1 line sizes. Miss rates begin decreasing with the rise of line sizes (see Figure 8). Results show that once 128 computer memory unit, miss rates either keep identical or increase. Like encoders, this is often as a result of large a line size could increase capability cache misses. The miss rates as a result of H.264/AVC decoder area unit smaller than those for MPEG-4 decoder.

Like encoders, higher associativity level is anticipated to boost decoders performance by reducing conflict cache misses. exploitation Cachegrind, we have a tendency to acquire CL1 miss rates for various associativity levels. As shown in Figure nine, CL1 miss rates considerably decrease from 2-way to 4-way; for higher associativity level the changes aren't vital. Also, the miss rates for H.264/AVC decoder area unit smaller than those for MPEG-4 decoder.

Figure ten shows the impact of accelerating CL2 size on the miss rate for MPEG-4 and H.264/AVC decoders. For CL2 size from 256K to 2M computer memory unit, the miss rates decrease sharply. it's conjointly observe that the miss rates for H.264/AVC decoder area unit smaller than those for MPEG-4 decoder. The deference of MPEG-4 and H.264/AVC miss rates for smaller CL2 (128K to 1M computer memory unit in our simulation) is important. CL2 miss rate for each MPEG-4 and H.264/AVC decoders area unit negligible for CL2 cache size 4MB or larger.

6.1.3 Total Power Consumption for Encoders and Decoders

In this following section, we have a tendency to gift the impact of cache parameters (CL1 line size, CL1 associativity level, and CL2 cache size) on total power consumption for MPEG-4 and H.264/AVC encoders and decoders.

The power consumed by any automatic data processing system varies wide betting on however ofttimes its cache memory system is getting used [47]. we have a tendency to use Associate in Nursing activity based mostly power analysis to match the whole power consumed by the system so as to run MPEG-4 and H.264/AVC CODEC. during this technique, a system element is taken into account to be one among the 3 states – active (component consumes adequate quantity of energy to be turned on and active), ideal (component consumes minimum quantity of energy simply to be turned on), or sleep (component is turned off and consumes no energy). Figure eleven shows the whole power consumption for MPEG-4 and H.264/AVC decoders for numerous line sizes. Total power consumed by the system is higher for smaller line size (between sixteen Bytes and 128 Bytes) for each decoders. Again, the system consumes less power to run H.264/AVC decoder than running MPEG-4 decoder. Similar characteristic of power consumption by the system is determined for MPEG-4 and H.264/AVC encoders.

Total power consumption for MPEG-4 and H.264/AVC decoders for numerous associativity levels is illustrated in Figure twelve. As shown within the figure, the whole power consumed by the system is higher for smaller associativity level (2-way) for each MPEG-4 and H.264/AVC encoders. Again, the system consumes less power to run H.264/AVC decoder than running MPEG-4 decoder. in line with the results, set-associativity quite 16-way might not be helpful.

Similar characteristic of power consumption by the system is determined for MPEG-4 and H.264/AVC encoders.

Figure thirteen shows overall power consumed by the system running MPEG-4 and H.264/AVC encoders for numerous CL2 size. we have a tendency to assume that power needed as a result of a CL2 miss is a hundred times quite that needed as a result of a CL2 hit. Simulation results show that the decrease in power consumed by the system is important for smaller CL2 cache sizes (between 256 kilobyte and one MB) for each encoders. this is often as a result of CL2 miss rates decreases sharply with increasing CL2 size for smaller CL2 and decreasing cache miss rates facilitate the system cut back energy consumption. it's conjointly determined that the system consumes less power to run H.264/AVC encoder than running MPEG-4 encoder. this is often as a result of H.264/AVC with lower bit rates (than MPEG-4) consumes less energy to be processed. Similar characteristic of power consumption by the system is determined for each decoders exploitation numerous CL2 size.

6.2 Impact of Cache lockup

Finally, we have a tendency to discuss the impact of cache lockup. Cache lockup improves sure thing by holding bound memory blocks within the cache throughout the whole period of the execution time. The additional cache blocks area unit bolted, the upper sure thing is achieved. As bolted cache size will increase – on the one hand, additional cache blocks area unit locked; on the opposite hand, the effective cache size decreases. These 2 contradictory phenomena create it troublesome to predict if performance and power consumption can increase or decrease once cache lockup is applied. within the following subsections, we have a tendency to gift the impact of I1 cache lockup on the performance and total power consumption.

6.2.1 Miss Rates

Experimental results show that miss rate starts decreasing with multiplied bolted cache size for each MPEG-4 and H.264/AVC encoders (see Figure 14). However, once 15 August 1945 bolted I1 cache size, miss rate begin increasing with multiplied bolted cache size for each encoders. The miss rate is also quite the miss rate while not cache lockup if incorrect numbers of cache blocks area unit bolted. Similar characteristic of miss rate for numerous bolted I1 cache size is determined for MPEG-4 and H.264/AVC decoders.

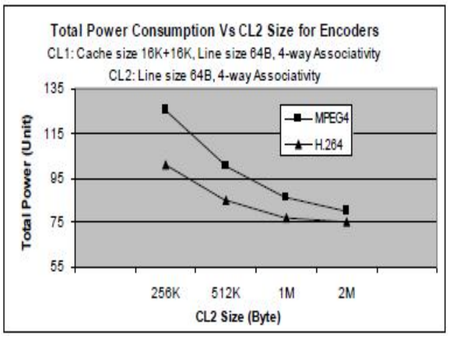

Figure 13: Total power consumed by the system Vs CL2 size

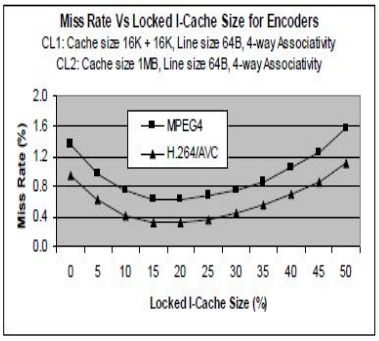

Figure 14: Miss Rates for encoding Vs Locked I-Cache size

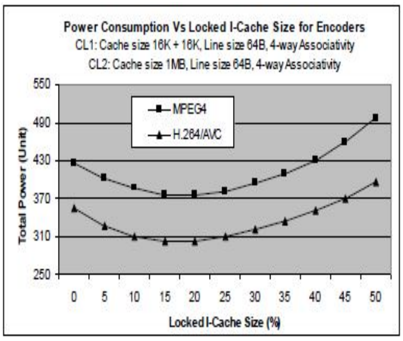

Figure 15: Total power consumed by the encoding system Vs Locked I-Cache size

6.2.2 Total Power Consumption

If aggressive cache lockup is employed and incorrect cache blocks area unit bolted, the system could consume additional power than the ability required by the system while not cache lockup. Total power consumed by the system for numerous bolted I1 cache size is shown in Figure fifteen. Total power consumption starts decreasing with the rise of cache lockup size for each MPEG-4 and H.264/AVC encoders. However, for cache lockup capability over 15 August 1945, the general power consumption starts increasing with the rise of cache lockup capability for each encoders. we have a tendency to observe similar characteristic of power consumption for MPEG-4 and H.264/AVC decoders.

Because of the dependencies among the video frames throughout secret writing and decryption video files, it's well worthy to examine the impact of cache parameters and cache lockup on the sure thing, power consumption, and performance for MPEG-4 and H.264/AVC CODEC before creating any style selections.

7. CONCLUSIONS

In order to run period video applications, higher execution time sure thing and better process speed area unit needed. Cache memory is employed to boost performance by bridging the speed gap between the most memory and central processor. However, cache needs further power to be operated and cache introduces execution time unpredictability. Embedded system has restricted resources. So, power consumption/dissipation ought to be unbroken nominal. Studies show that cache optimisation together with cache lockup may be wont to improve sure thing with none vital negative impact on performance/power quantitative relation. during this paper, we have a tendency to investigate the impact of cache parameters and cache lockup on the sure thing, power consumption, and performance for 2 wide used period applications, MPEG-4 and H.264/AVC CODEC, running on Associate in Nursing embedded system. we have a tendency to simulate Associate in Nursing Intel Pentium-like design that incorporates a single core with two-level cache memory hierarchy. we have a tendency to acquire miss rates and total power consumption by variable cache parameters for non-locking and level-1 instruction cache lockup. Simulation results show that sure thing of embedded CODEC may be improved {without Associate in Nursingy|with none} vital negative impact on performance and power consumption by applying cache lockup mechanism to an optimized cache memory organization. it's conjointly determined that the distinction between miss rates created by MPEG-4 and H.264/AVC is important for smaller CL1 line size, CL1 associativity level, and CL2 cache size. Experimental results conjointly show that H.264/AVC has performance advantage over MPEG-4 for smaller caches.

Most chip-vendors area unit deploying multicore processors to their line of products for improved performance/power quantitative relation. it'd be fascinating to explore the impact of cache lockup on sure thing Associate in Nursingd performance/power quantitative relation of multicore period embedded systems as an extension to the present work.

REFERENCES

[1] “Improving cache performance,” UMBC, 2007. URL: www.csee.umbc.edu/help/architecture/611-5b.ps

[2] A. Asaduzzaman, “A Power-Aware Cache Organization Effective for Real-Time Distributed and Embedded Systems”, accepted in Journal of Computers (JCP), 2012.

[3] O. Certner, Z. Li, P. Palatin, O. Temam, F. Arzel, N. Drach, “A Practical Approach for Reconciling High and Predictable Performance in Non-Regular Parallel,” in Design, Automation and Test in Europe (DATE'08, 2008), 2008.

[4] M. Soryani, M. Sharifi, M.H. Rezvani, “Performance Evaluation of Cache Memory Organizations in Embedded Systems,” in the Fourth International Conference on Information Technology: New Generations (ITNG'07), pp.1045-1050, 2007.

[5] C.M. Kirsch, R. Wilhelm, “Grand Challenges in Embedded Software,” in ACM Conference on Embedded Systems Software (EMSOFT’07), pp.2-6, 2007.

[6] M. Grigoriadou, M. Toula, E. Kanidis, “Design and Evaluation of a Cache Memory Simulation Program,” in IEEE Conference, 2003.

[7] W. Wolf, “Multimedia Applications of Multiprocessor Systems-on-Chips,’ in IEEE DATE’05 Proceedings, 2005.

[8] N. Slingerland, A. Smith, “Cache Performance for Multimedia Applications,” in Portal ACM, URL: portal.acm.org/ft_gateway.cfm?id=377833&type=pdf

[9] A. Asaduzzaman, F.N. Sibai, et al, "An Effective Dynamic Way Cache Locking Scheme to Improve the Predictability of Power-Aware Real-Time Embedded Systems", in ICECS-2011, Beirut, Lebanon, 2011.

[10] I. Puaut, “Cache Analysis Vs Static Cache Locking for Schedulability Analysis in Multitasking Real-Time Systems,” 2006. URL: http://citeseer.ist.psu.edu/534615.html

[11] I. Puaut, C. Pais, “Scratchpad memories vs locked caches in hard real-time systems: a quantitative comparison,” in Design, Automation & Test in Europe Conference & Exhibition (DATE'07), pp. 1-6, 2007.

[12] A.M. Campoy, E. Tamura, S. Saez, F. Rodriguez, J.V. Busquets Mataix, “On Using Locking Caches in Embedded Real-Time Systems,” in ICESS-05, LNCS 3820, pp. 150-159, 2005.

[13] J. Robertson, K. Gala, “Instruction and Data Cache Locking on the e300 Processor Core,” Freescale Semiconductor, 2006.

[14] S. Kannan, M. Allen, et al, “Cached Memory Performance Characterization of a Wireless Digital Baseband Processor,” in IEEE V-361-364.

[15] W. Wolf, M. Kandemir, “Memory System Optimization of Embedded Software,” in Proceedings of the IEEE Vol. 91, No. 1, pp. 165-182, 2003.

[16] P. Panda, P. Kjeldsberg, et al, “Data and Memory Optimization Techniques for Embedded Systems,” in ACM Transactions on Design Automation of Electronic Systems, Vol. 8, No. 2, pp. 149-206, 2001.

[17] P.J. Koopman, “Embedded System Design Issues (the Rest of the Story),” in Proceedings of the International Conference on Computer Design (ICCD’96), 1996.

[18] P. Soderquist, M. Leeser, “Optimizing the Data Cache Performance of a Software MPEG-2 Video Decoder,” in ACM Multimedia 97 – Electronic Proceedings, Seattle, WA, 1997.

[19] J. Chase, C. Pretty, “Efficient Algorithms for MPEG-4 Video Decoding,” in TechOnLine, University of Canterbury, New Zealand, 2002.

[20] Y. Li, J. Henkel, “A framework for estimating and minimizing energy dissipation of embedded HW/SW systems,” in Proc. 35th Design Automation Conf., pp.188-194, 1998.

[21] C. Kulkarni, F. Catthoor, H. DeMan, “Hardware cache optimization for parallel multimedia applications,” in Proceedings of the 4th International Euro-Par Conference on Parallel Processing, pp. 923–932, 1998.

[22] N. Slingerland, A. Smith, “Design and characterization of the Berkeley multimedia workload, Multimedia Systems,” in Springer-Verlag, pp. 315-327, 2002.

[23] Y.-K. Chen, E. Debes, et al, “Evaluating and Improving Performance of Multimedia Applications on Simultaneous Multi-Threading,” in IEEE Proceedings of the Ninth International Conference on Parallel and Distributed Systems (ICPADS’02), pp. 529-535, 2002.

[24] A.M. Molnos, B. Mesman, et al, “Data Cache Optimization in Multimedia Applications,” in Proceedings of the 14th Annual Workshop on Circuits, Systems and Signal Processing, pp. 529-532, The Netherlands, 2003.

[25] C. Zhang, F. Vahid, R. Lysecky, “A Self-Tuning Cache Architecture for Embedded Systems,” in ACM Trans. on Embedded Computing Systems, Vol. 3, No. 2, pp. 407-425, 2004.

[26] L. Nosetti, C. Solomon, E. Macii, “Analysis of Memory Accesses in Embedded Systems,” in The 6th IEEE International Conference on Electronics, Circuits and Systems, pp. 1783-1786, 1999.

[27] A. Naz, K. Kavi, M. Rezaei, W. Li, “Making a Case for Split Data Caches for Embedded Applications,” in the ACM SIGARCH Computer Architecture News, Vol. 34, Issue 1, pp. 19-26, 2006.

[28] A. Janapsatya, A. Ignjatovic, S. Parameswaran, “Finding Optimal L1 Cache Configuration for Embedded Systems,” in the Proceedings of the 2006 conference on Asia South Pacific design automation, pp. 796-801, 2006.

[29] W. Nebel, “System-Level Power Optimization”, in IEEE/DSD, 2004.

[30] T. Givargis, F. Vahid, J. Henkel, “Fast Cache and Bus Power Estimation for Parameterized System-on-a-Chip Design”, 2005.

[31] R.E. Ahmed, “Energy-Aware Cache Coherence Protocol for Chip-Multiprocessors”, in the Canadian Conference on Electrical and Computer Engineering (CCECE'06), pp. 82-85, 2006.

[32] R. Koenen, F. Pereira, L. Chiariglione, “MPEG-4: Context and Objectives,” Invited paper for the Special Issue on MPEG-4 of the Image Communication Journal, 2002.

[33] S. Ely, “MPEG video coding - A simple introduction,” in EBU Technical Review Winter, 1995.

[34] R. Schaphorst, “Videoconferencing and Videotelephony – Techonology and Standards,” in Artech House, Norwood, MA, 2nd ed. 1999.

[35] Z. Wu, M. Tokumitsu, T. Hatanaka, “The Development of MPEG-4-AVC/H.264, the Next Generation Moving Picture Coding Technology,” in Issue 200 Vol.71 No.4, Oki Technical Review, 2004.

[36] I. Richardson, “H.264 / MPEG-4 Part 10: Overview,” in www.vcodex.com, 2002.

[37] A. Avritzer, J. Kondek, D. Liu, E.J. Weyuker, “Software Performance Testing Based on Workload Characterization,” in WOSP'02, AT&T Labs, ACM ISBN 1-1-58113-563-7 02/07, Rome, Italy, 2002.

[38] K. Diefendorff, P. Dubey, “How Multimedia Workloads Will Change Processor Design,” in IEEE Computer, pp. 43–45, 1997.

[39] A. Maxiaguine, Y. Liu, S. Chakraborty, W. Ooi, “Identifying "representative" workloads in designing MpSoC platforms for media processing,” in 2nd Workshop on Embedded Systems for R-T Multimedia, pp. 41-46, 2004.

[40] “VisualSim – system-level simulator,” 2012.
URL: www.mirabilisdesign.com

[41] “Cachegrind - a cache profiler from Valgrind,” 2012. URL: http://valgrind.kde.org/index.html

[42] “FFmpeg – A very fast video and audio converter,” 2012. URL: http://ffmpeg.sourceforge.net/ffmpeg-doc.html#SEC1

[43] JM-RS (96) – H.264/AVC Reference Software, 2012. URL: http://iphome.hhi.de/suehring/tml/ download/

[44] P.F. Sweeney, M. Hauswirth, et al, “Understanding Performance of MultiCore Systems using Trace-based Visualization”, in STMCS’06, Manhattan, NY, 2006.

[45] A. Jerraya, H. Tenhunen, W. Wolf, “Multiprocessor Systems-on-Chips”, in IEEE Computer Society, pp. 36–40, July 2005.

[46] P. Gepner, M.F. Kowalik, “Multi-Core Processors: New Way to Achieve High System Performance”, in the Proceedings of the International Symposium on Parallel Computing in Electrical Engineering (PARELEC'06), pp. 9-13, 2006.

[47] “Static random access memory”, from Wikipedia, the free encyclopedia, 2012.

[48] “MPEG-4: The container for digital media,” 2012. URL: apple.com/quicktime/technologies/MPEG-4/

[49] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra; “Overview of the H.264 / AVC Video Coding Standard,” in IEEE Transactions on Circuits and Systems for Video Technology, 2003.