2009 ISECS International Colloquium on Computing, Communication, Control, and Management

Cache Optimization for Real Time MPEG-4 encoder

HU-Zhiqiang, DENG Lun-hui, RuiLv
Engineering Center of Digital Audio and Video
Communication University of China
Beijing China
arifence@cuc.edu.cn

Abstraction—**Due to the speed up of the VLSI technology, the high speed of the DSP core is not compatible with much lower speed external memory. The low speed of the off-chip memory, which is mainly responsible for the DSP delays, all that was called memory wall .The memory wall seriously confined the DSP's computation capability for the intense data exchange algorithm. This paper provides two strategies (Regulation the L2 cache ,adjust the encoding procedure) to speed up the performance of the Cache which based on the study of the architecture of the TMS320DM642 two level cache and the algorithm flow of the MPEG-4 encoder.**

Key words: MPEG-4 Cache TMS320DM642 optimization

I INTRODUCTION

Due to the rapidly growing demand of the video surveillance and other multimedia applications, The MPEG-4 video compress algorithm has been applied in so many areas such as cell phone TV and mobile video communications, security surveillance and so on. But the complexity and the intensely computation of the MPEG-4 algorithm decide that to implementation of it into real world is a tough problem. One efficient,and fast way to realize the MPEG-4 is to use the latest DSP platform such as Ti's VLIW-SIMD TMS320DM642 high performance DSP ,however the DSP's on chip cache is limited compare to the common desktop CPU (such as Intel's Core i7 which has 8MB on chip cache),which is the most important part to balance the high speed dsp core and much lower external memory for the intense data exchange algorithm . The memory subsystem has been the bottleneck for the real time system.So to remold the complex algorithm to fit the architecture of the DSP's limited cache is really critical .In this paper we proposed two novel algorithms: to change the algorithm flow and the regulation the date stream to speed up the real time encoder system.At last we realize the MPEG-4 algorithm over the TMS320DM642 platform(use the EVM board),to test our algorithm.

II RELATED WORK

A number of studies has been done on cache optimization for DSP based platform. In this section we only include some very critical work which analysis the behavior of the cache subsystem for multimedia application. The author in [2][3].both give us some methods to use processor's DMA and the Cache to accelerate the performance for multimedia application, Use of DMA enables the data transfers and the processing to be accomplished in parallel .In[4] the author analysis the cache's architecture for multimedia ,such as the line size,N- way associativity or the capacity of cache. In[5]the author give us a way to optimization the H.264 for business use IP Set-Top Box. All the references give us a common technical solution to the cache optimization, but in our paper we provide a Real Time cache tone strategy to evaluate the MPEG-4 encoder over DSP platform.

III AARCHITECTURE AND THE ALGORITHM

A TMS320DM642 and the architecture of the Two level cache

DM642 is high performance Fix-point VLIW DSP just for digital media applications, which use advanced VelociTI very-long-instruction-word (VLIW) architecture (VelociTI.2) and eight parallel units. Deep pipeline with performance of up to 5760 million instructions per second (MIPS) at a clock rate of 720 MHz.The DM642 also has Two level hierarchy of cache, L1P/D is the first-level program and date designated 16K 2-way set associative cache which has the highest communication frequency to the DSP core.L2 on chip memory/cache is 256KB configurable, can be set for various memory or cache, and the speed of L2 is half of L1 cache. When the data and the instruction which the DSP need are not in the on chip two level Cache then they must fetch from external memory which will cost serious DSP pipeline delay (Fig3) when Cache Miss happen in L2 the DSP pipeline must wait for about 8 clocks. To the deep pipeline processor is really a disaster.

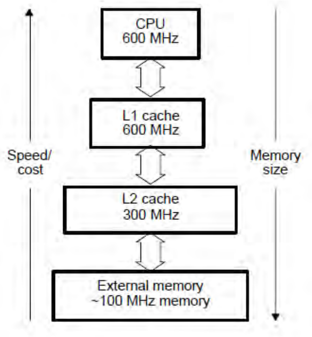


Fig1 Access Speed of Different Memories

For the off chip memory is run at much lower speed than the on chip cache. So we proposed two methods to regulation the on chip memory ,and adjust the algorithm flow to reduce the L1P/L1D cache miss rate.

B Analyze of the MPEG-4 algorithm's data exchange flow.

The motion estimation/compensation is the most computation intensely part in the MPEG-4 algorithm ,it's the bottleneck for this algorithm realized over real-time system .For our system we aim at encoding D1 resolution (720*576)raw video in real time ,the picture sample is 4:2:0 so only for one frame will taking almost 607KB (720*576*1.5Byte) to store .The MPEG-4 algorithm must access three frames(current frame, reference frame, reconstruct frame) at one encoding loop. Obviously to place these three frames all on the DM642 chip is not possible. According to Fig 2.

Firstly to getting the integer motion vector, the searching area from the reference frame, and the current MB date from the current frame must be loading to the on chip memory, Secondly, the half pixels interpolated area then loading to the on–chip memory .Due to the limited capacity of the on-chip cache all the date will be prematurely eject from cache .But in order to computing the residual frame later ,all the current, reference's MB which loaded to the cache earlier are reused now. In this point the date reuse rate is such low.

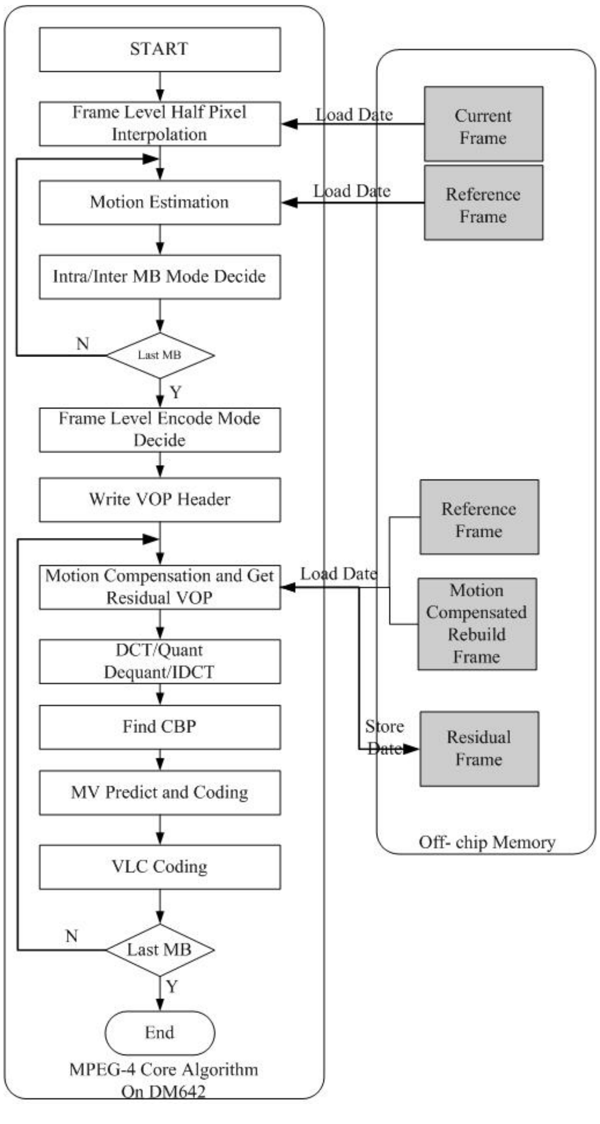


Fig2 Date Exchange Flow for Motion Estimation

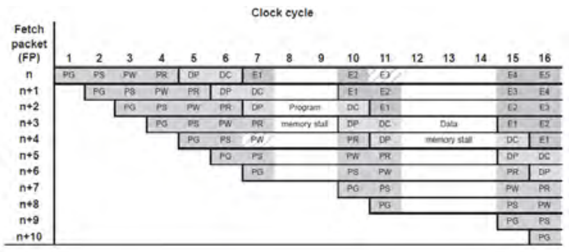


Fig3 The Pipeline Block of DM642 Processor

Under this unevaluated algorithm flow ,the off-chip date must be loading to the on chip cache frequently, loading date from external memory can caused serious pipeline stall ,such as Fig2. According to this situation we proposed a novel date Scheduling strategy to reduce the cache miss.

IV PROPOSED CACHE OPTIMIZATION

A Optimized date loading strategy

Using the on chip EDMA ,enable the date transfer and DSP processing in parallel, where the processing can be completed utilizing the full processing power, is our solution to the problem.

For the D1 resolution raw sequence (4:2:0sampling) One frame must take 607KB off-chip memory. In order to enable the DSP accessing acting date immediately we proposed a preloading strategy ,to loading the reference ， current frame date to the L2 on-chip memory for the next processing while the DSP is dealing the former workload. Refer to Fig 6(above)for the usual DSP processing time line ,we can conclude that there is a lot of time redundancy when the DSP waiting for EDMA transferring needed off-chip date. To get the half pixels precision motion vector ,the reference frame must be interpolated ,so the whole reference frame are loading to the on chip memory ,then motion searching machine start working ,the DSP must access integer reference frame and the current frame ,so the interpolated date will be kick out.

A lot of time is wasted for the data fetching in kicking out. However the EDMA subsystem could transfer the date with DSP processing in parallel ,the DSP is not smart enough that it can not preload the date for next processing , according to this we do the preloading according to the MPEG-4 algorithm flow ,refer to Fig6(below) we could preloading the reference frame (PreloadingA)the encoder is doing some of the initialization work such as get the point of the some struct(which used in the motion estimation) initialization the variables .It's all the same to the PreloadingB we preloading the current frame date for the integer motion search while the DSP are computing the searching area and the MB (macro block)encoding mode. Then the whole motion estimation flow can be compressed as Fig6(below).

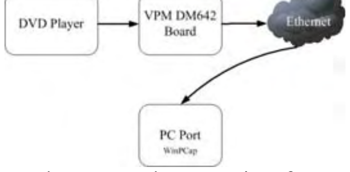


Fig4 The Hardware Implementation for MPEG-4 Encoder

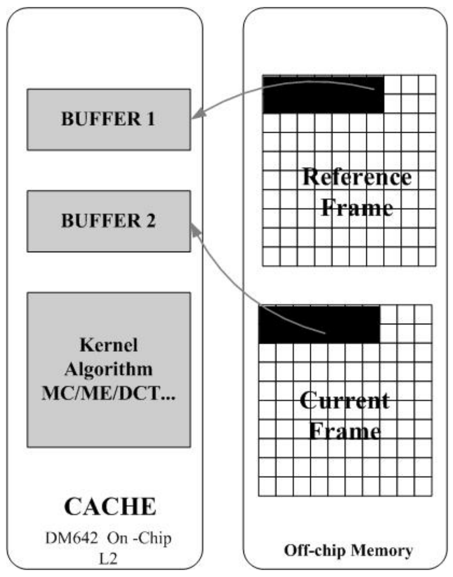
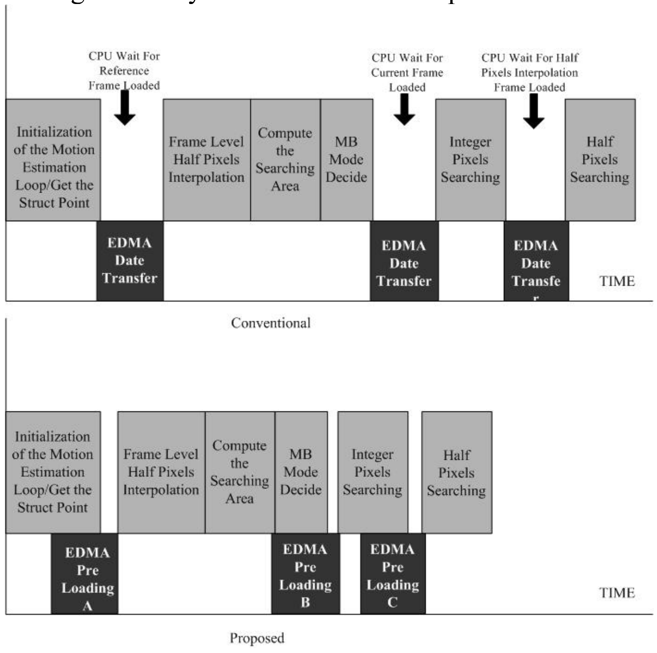


Fig5 Memory Allocation for on Chip L2 Cache


CPU Wait For Reference Frame Loaded
CPU Wait For Current Frame Loaded
CPU Wait For Half Pixels Interpolation Frame Loaded
Initialization of the Motion Estimation Loop/Get the Struct Point
Frame Level Half Pixels Interpolation
Compute the Searching Area
MB Mode Decide
Integer Pixels Searching
Half Pixels Searching
EDMA Date Transfer
EDMA Date Transfer
EDMA Date Transfe r
TIME
Conventional
Initialization of the Motion Estimation Loop/Get the Struct Point
Frame Level Half Pixels Interpolation
Compute the Searching Area
MB Mode Decide
Integer Pixels Searching
Half Pixels Searching
EDMA Pre Loading A
EDMA Pre Loading B
EDMA Pre Loading C
TIME
Proposed
Fig6 Regulation the Motion Estimation Date Flow

To realize our novel schedule strategy, we configure the L2 cache as 192K SRAM and 64K cache architecture Then we divide the 192KB SRAM into two part :Buffer and Buffer2 then we can preload the reference fram acting MB into Buffer1 ,the current frame acting MI loading to Buffer2 (Fig5) .After allocating the on-chi buffer we use the preloading strategy described before t loading the acting date. We allocate 56KB for Buffer 1 s approximately 540 acting MBs could loading to the buffe once. For D1 resolution raw sequence there ar 6480(90*72)MBs(8pixels *8pixels) one frame By th same way we also allocate 102KB buffer2 for the curren frame acting MBs. Because the reference frame date ha the higher accessing rate for motion searching.

When we have done this the DSP no longer waitin so long time for the date loaded. Then the efficiency o motion estimation loop is much higher than before. The left 34KB SRAM is reserved for the key code (such as motion estimation/compensation DCT/IDCT)which has a characteristic of frequently called ,and some global variables ,VLC coding tables.After we done this we test our encoder by several CIF resolution YUV sequences the L1D cache miss rate is bring down about 33 percent.Fig11

B Instruction cache optimization

The DM642 on-chip L1P cache only has 16KB capacity, but the original MPEG-4 source code is more than that. By studying the behavior of the instruction cache ,when a code loop starting all the loop code are loading to the on-chip cache firstly, if the loop code size surpass the 16KB instruction cache ,the instructions are loading and eject from L1P frequently ,tremendously bring up the L1P cache miss rate .In order to this we proposed a new encoding procedure which change the frame level macro block loop to three macro block encoding loops ,all code size of each loop no longer surpass the size of L1P ,The three loops are motion estimation loop (Fig 10 middle) intra text coding loop (Fig10 above) and the motion compensation (frame reconstruct)(Fig10 below) . All three loops are dealing ‘M’ Macro Blocks in one encoding flow,’M’ is defined by the Preloading MB’s number which we discussed in chapter 4.1.

In the same way we test the evaluated encoding flow by Ti CCS CACHE TONE, which could visualize the cache performance for DM642 processor ,The L1P cache miss rate is relatively lower than before optimization.Fig7.

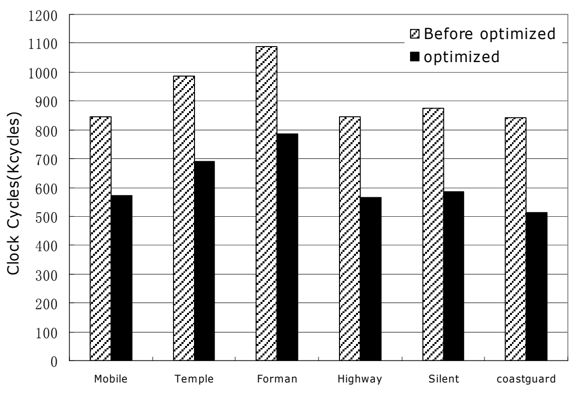


Fig7 L1P Cache Miss Rate

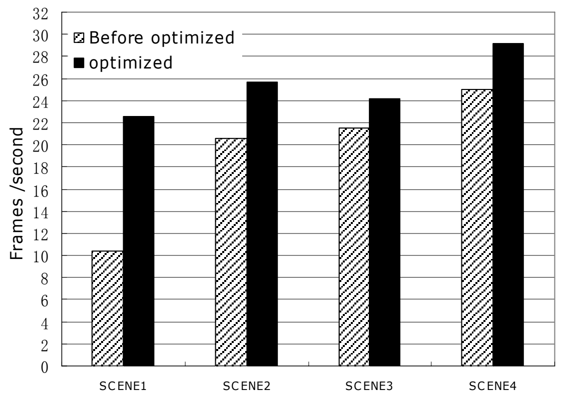


Fig8 Real Time Video Scene Test

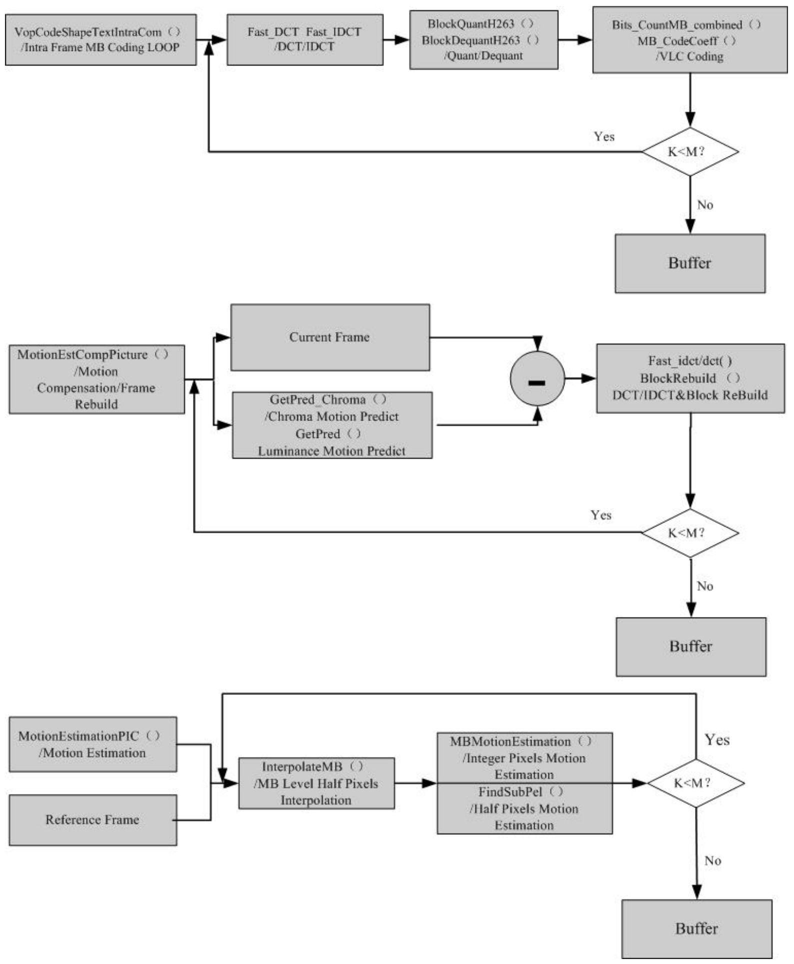


Fig9 Divide the Encoding Flow into Three Loops

V Hardware implementation and experimental results

To examines our algorithm optimization result, we implement the MPEG-4 algorithm over the real time hardware system. The whole real-time system is composed of VPM DM642 board (Evaluation Board for DM642) and a DVD player which in charge of outputting NTSC raw analog D1 video refer to Fig4 and Fig12 is our hardware system . We use the desktop PC to receive the encoded stream through the Ethernet with the 802.3 protocol. Though the EMAC port on the VPM DM642 board, the encoded stream is divided into 1024Byte a packet, then sending to the Ethernet . In the PC port we develop a receive software based on winpcap(windows packet capture)to receive and decoding the stream in real-time .

We test four different Video scenes derive form DVD player and Industry motion camera, Scene1 is a 1024 frames action scene form the movie IRON MAN ,all the components in the picture are moving violently, so we achieve the lowest encoding frames per second, for the massive computation and date exchange of motion searching ,the evaluation of L1D cache performance is the most significance .Also due to more residue date to doing VLC coding,the test score is low.Fig8. In Scene 2 and 3 we select two dialogue scene(Secne3 from INDIANA JONES IV) ,the components moving are mild ,so the test score are relatively higher than Scene1.At last we capture the Video from the motion camera in real-time，all the background is still, only the author is moving ,so a lot of MBs is skipped as SKIP MB,we achieve the highest performance. Our MPEG-4 encoder is almost in real-time encoding. So our MPEG-4 encoder(D1 resolution) can be used as security surveillance Real-Time encoder.

VI Conclusion

In this paper we proposed two different strategy to bring down the cache miss rate for DSP platform running multimedia applications. We test our algorithm in real time system, the cache miss rate is bring down about 30 percent ,and the encoding speed is about 25f/s .Fast enough to the security surveillance use.

Fig10 Real Time Test System

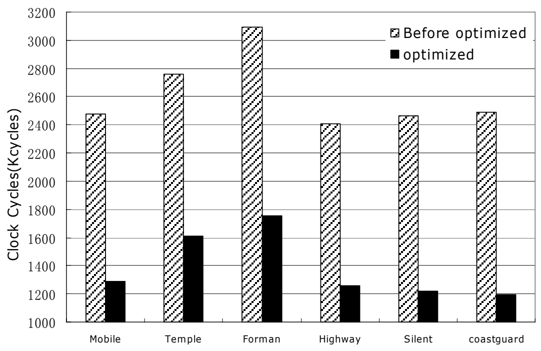


Fig11 L1D Cache Miss Rate

SCENE1 IRON MAN Action Scene
SCENE2 IRON MAN Dialogue Scene
SCENE3 INDIANA JONES4 Dialogue Scene
SCENE4Real Time Video from The Motion Camera(Author’ s Lab)

Fig 12 Four Different Video Testing Scenes

ACKNOWLEDGMENT

Thanks for sponsors of CNGI-04-12-2A of NDRC of China, important program of ME of China (106041), and 211 project of Communication University of China.

REFERENCE

[1] Zhiyong Xu, Student Member, IEEE, Sohum Sohoni, Rui Min, and Yiming Hu, Senior Member, IEEE An Analysis of Cache Performance of Multimedia Applications IEEE TRANSACTIONS ON COMPUTERS, VOL. 53, NO. 1, JANUARY 2004

[2] Abu Asaduzzaman, Imad Mahgoub, Praveen Sanigepalli, Hari Kalva, Ravi Shankar, and Borko Furht Cache Optimization for Mobile Devices Running Multimedia Applications Proceedings of the IEEE Sixth International Symposium on Multimedia Software Engineering

[3] Christian Zinner and Wilfried Kubinger ROS-DMA: A DMA Double Buffering Method for Embedded Image Processing with Resource Optimized Slicing Proceedings of the Twelfth IEEE Real-Time and Embedded Technology and Applications Symposium

[4] Shuai Hu , Zhe Zhang , Mengsu Zhang ,Tao Sheng Optimization of Memory Allocation for H.264/AVC Video Decoder on Digital Signal Processors 2008 Congress on Image and Signal Processing 72-75

[5] Gudula Rünger and Michael SchwindDepartment of Computer Science, Cache optimization for mixed regular and irregular computations

[6] A.M. Molnos, M.J.M. Heijligers, S.D. Cotofana, J.T.J. van Eijndhoven, and B. Mesman, “Data CacheOptimization in Multimedia Applications”, Proceedings of the 14th Annual Workshop onCircuits, Systems and Signal Processing, ProRISC2003, pp. 529-532, Veldhoven, The Netherlands, November 2003

[7] P. Soderquist and M. Leeser, “Optimizing the DataCache Performance of a Software MPEG-2 VideoDecoder”, ACM Multimedia 97 – ElectronicProceedings, Seattle, WA, Nov. 1997

[8] N.T. Slingerland and A.J. Smith, “Cache Performance for Multimedia Applications” portal.acm.org/ft_gateway.cfm?id=377833&type=pdf

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