International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 3, Issue 2 (January 2014), PP. 01-08

Performance analysis of VP8 Video Decoder on TMS3206713 Platform:

Basavaraju S[1], Dr.Sivakumar B[2]
[1]Reserach Scholar, PRIST University, Thanjavur, [2]Prof and HOD TC dept, Dr AIT, Bangalore

Abstract:- Digital video continues to extend visual communication to an ever-larger range of applications, and also an increasing number of developers are becoming involved in creating new video systems or enlarging the capabilities of existing designs. Video not only demands very high performance but the wide variety of video applications requires the performance to be optimized depending on the particular system. Programmable, high-performance DSPs and platforms from TI allow developers to adapt their software readily to application requirements that can vary widely in terms of transmission bandwidth, storage, image specifications and quality requirements. The inherent structure and complexity of video encoding and decoding (codec) the inherent structure and complexity of video encoding and decoding (codec) algorithms drives the optimization approach. Encoders are particularly important because they must adapt to the application and they represent a major portion of the heavy processing load of video applications. The goal for video compression is to encode digital video using as few bits as possible while maintaining acceptable visual quality. While encoders are based on the mathematical principles of information theory, they may still require implementation trade-offs, which can be quite complex. TI encoders/decoders are highly configurable, providing an easy-to-use system interface and performance optimization for a wide range of video applications. Even if the developer is new to DSPs or video algorithms, TI video encoders/decoders offer a number of techniques that can make the optimization process easier and more effective in terms of compression and picture quality. So ported and optimized VP8 video decoder on TMS3206713 platform and analyse the results.

Keywords:- Video, VP8, Bitrates, PSNR, H264, MPEG, TMS3206713

I. INTRODUCTION

Video teleconferencing, High definition television, streaming Internet video content, every day we receive and exchange more and more data. Video[7] manipulation becomes each day a more challenging process. This problem is complex : first because of the diversity of the video content available but also because of the different capacities of all the devices requiring these videos.Video[1] is an significant part of modern life, from movies and television shows, to news, sports, home videos and events. The Internet has been expanding in bandwidth usage at an enormous rate, and Internet video tops the growth demographics, In the last fifteen years, there has been a continuous evolution of video and there is no sign that this evolution is at an end. MPEG-2[8] is started in full format in earlier days, and been overtaken in respect to the degree of achievable compression by H.263, MPEG-4 and in 2003 by H.264 [2]. Along with H.261, MPEG-1 and MPEG-2 were the first codec's to combine multiple ways (algorithms) of removing redundancy in one codec. Now VP8 Video codec's is now an "elderly" codec, having been standardized by international authorities in the period Essentially, each video frame is split into blocks and matching blocks between successive frames are sought. Only the difference after it has been further encoded is then transmitted or stored. Each of the contributory algorithms has been refined over the years as intensive and competitive global research has taken place.

Much of our work is taken up with the impact of those algorithms in the next two decade. The latest codec to emerge, H.264 and after this VP8 recently released by ON2 , has taken advantage of the hardware bonus, as achievable computational complexity has increased in line with Moore's law (a doubling of processing power every 18 months). In particular, the size of the blocks that are compared has been reduced and made more flexible, which reduces the difference data that remains to be encoded. Of course, improved compression allows either a reduction in the spectrum required to transmit the same programmes or improved video quality using the existing spectrum or a combination of both. With the advent of second generation High Definition Television (HDTV) in Europe there will be an inevitable demand for greater bandwidths, as bit-rates of about three times those of Standard Definition Television (SDTV) are expected. It is likely that H.264[1] will provide many times the compression that was once achieved by MPEG-2 but that the gain will vary according to the size, resolution and quality of the image, either HDTV, SDTV or one of the smaller formats for handheld devices. However, an increasing number of services and growing popularity of high definition TV are creating greater needs for higher coding efficiency. Moreover, other transmission media such as Cable Modem, xDSL, or UMTS offer much lower data rates than broadcast channels, and enhanced coding efficiency can enable the transmission of more video channels or higher quality video representations within existing digital transmission capacities

II. INTRODUCTION ABOUT VP8 VIDEO CODEC

a) VP Video codec:

The complete block diagram of VP8 Video codec [3] is shown in fig 1 Basically VP8 Video codec is released by On2 technologies and then Google acquired On2 calls for Google to release the VP8 source code. At this moment, libvpx is the only software library capable of encoding VP8 video streams and also the same libvpx is capable of decoding VP8 video streams. TheVP8 video codec good strength is it offers the "Better quality real-time video delivery",

This Video codec also started to give the attention (attract) for wide interest in the video coding research community from both industry and as well as academia.

One more important factor is VP8 uses three different types of reference frames for inter prediction: in this usually the “last frame”, a “golden frame” (one frame worth of decompressed data from the arbitrarily distant past) and also last frames called as “alternate reference frame.” VP8 bit stream initially separating the compressed data into two categories, one for macro block coding modes and motion vectors and one for quantized transform coefficients. This Video codec also can make the good use of computation power in especially modern hardware for improving compression efficiency even while maintaining fast decoding speed on majority devices connected to the web applications.

And major features of VP8 Video codec is it will support for low bandwidth applications and major features are Web video format, Hybrid transform with adaptive quantization, Flexible reference frames, Efficient intra prediction and interpretation, sub-pixel interpolation, Adaptive in-loop deblocking filtering, even Frame level adaptive entropy coding, and friendly data partitioning.

There are two decoders 1) VPX decoder 2) Simple Decoder the only difference between the two is that simple decoder just simply decodes a stream and nothing else, while the vpxdec can do plenty of other things.

2.1 VP8 Video Codec Features

From the very beginning of VP8’s[4] development, the developers were focused on Internet/web-based video applications. This focus has led to a number of basic assumptions in VP8’s overall design:

Low bandwidth requirement: One of the basic design assumptions is that for the foreseeable future, available network bandwidth will be limited. With this assumption, VP8 was specifically designed to operate mainly in a quality range from “watchable video” (~30dB in the PSNR metric) to “visually lossless” (~45dB).

Heterogeneous client hardware: There is a broad spectrum of client hardware connected to the web, ranging from low power mobile and embedded devices to the most advanced desktop computers with many processor cores. It must, therefore, be possible to create efficient implementations for a wide range of client devices.

Web video format: VP8 was designed to handle the image format used by the vast majority of web videos: 420 color sampling, 8 bit per channel color depth, progressive scan (not interlaced), and image dimensions up to a maximum of 16383x16383 pixels.

The push for compression efficiency and decoder simplicity under these design assumptions led to a number of distinctive technical features in VP8 [3], relative to other known video compression formats, such as MPEG-2[8], H.263 and H.264/AVC. The following list highlights the technical innovations in VP8:

Hybrid transform with adaptive quantization: VP8 uses 4x4 block-based discrete cosine transform (DCT) for all luma and chroma residual signal. Depending on the prediction mode, the DC coefficients from a 16x16 macro block may then undergo a 4x4 Walsh-Hadamard transform.

Flexible reference frames: VP8 uses three reference frames for inter prediction, but the scheme is somewhat different from the multiple reference motion compensation scheme seen in other formats. VP8’s design limits the buffer size requirement to three reference frame buffers and still achieves effective de-correlation in motion compensation.

Efficient intra prediction and inter prediction: VP8 makes extensive uses of intra and inter prediction. VP8’s *intra* prediction features a new “TM_PRED” mode as one of the many simple and effective intra prediction methods. For *inter* prediction, VP8 features a flexible "SPLITMV" mode capable of coding arbitrary block patterns within a macroblock.

High performance sub-pixel interpolation: VP8's motion compensation uses quarter-pixel accurate motion vectors for luma pixels and up to one-eighth pixel accurate motion vectors for chroma pixels. The sub-pixel interpolation of VP8 features a single stage interpolation process and a set of high performance six-tap interpolation filters.

Adaptive in-loop deblocking filtering: VP8 has a highly adaptive in-loop deblocking filter. The type and strength of the filtering can be adjusted for different prediction modes and reference frame types.

Frame level adaptive entropy coding: VP8 uses binary arithmetic coding extensively for almost all data values except a few header bits. Entropy contexts are adaptive at the frame level, striking a balance between compression efficiency and computational complexity.

Parallel processing friendly data partitioning: VP8 can pack entropy coded transform coefficients into multiple partitions, to facilitate parallel processing in decoders. This design improves decoder performance on multi-core processors, with close to zero impact to compression efficiency and no impact to decoding performance on single core processors.

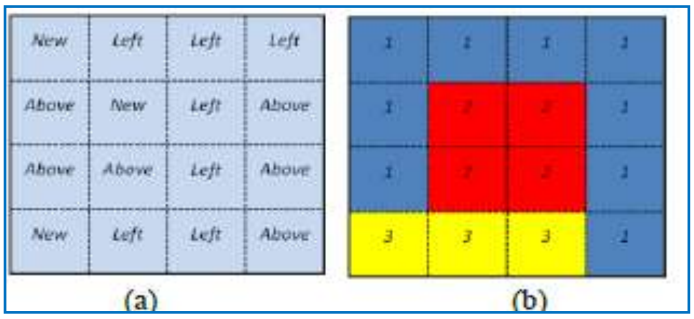


Fig 1: Illustration of VP8 inter prediction mode SPLITMV.

PSNR can be calculated using below eq:

$$p(X,Y) = 10\log_{10}\frac{255^2 n\, m}{\sum_{i=1,j=1}^{n,m}(X_{ij} - Y_{ij})2}$$

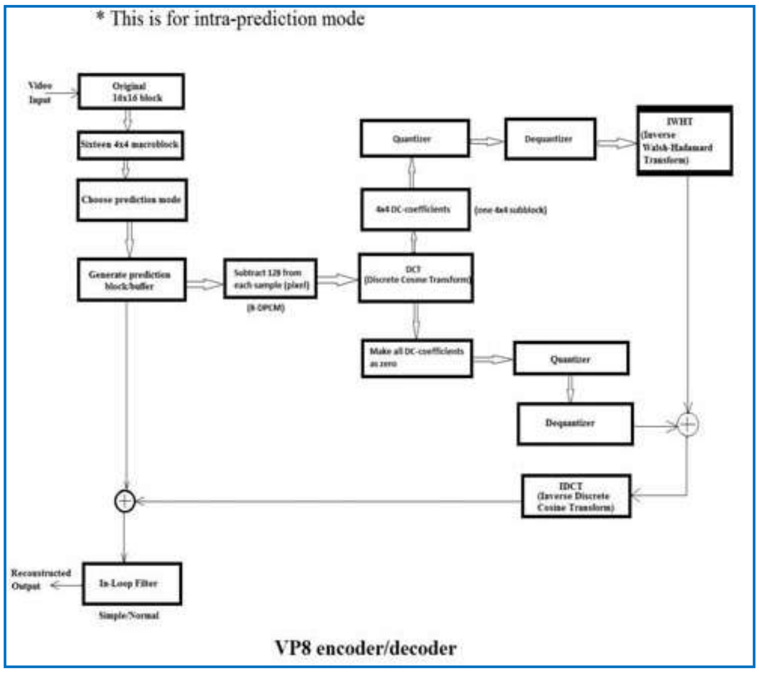

* This is for intra-prediction mode
Video Input
Original 16x16 block
Sixteen 4x4 macroblock
Choose prediction mode
Generate prediction block/buffer
Subtract 128 from each sample (pixel)
(8 DPCM)
DCT (Discrete Cosine Transform)
4x4 DC-coefficients
(one 4x4 subblock)
Quantizer
Dequantizer
IWHT (Inverse Walsh-Hadamard Transform)
Make all DC-coefficients as zero
Quantizer
Dequantizer
IDCT (Inverse Discrete Cosine Transform)
Reconstructed Output
In-Loop Filter
Simple/Normal
VP8 encoder/decoder

Fig 2: VP8 Video Codec

III. ABOUT TMS3206713

The experiments described are performed on a C6713[6] DSP board. The initial DSP frequency is 225MHz and has 8 pipelines. It could work at 1.35 giga-floating-point operations per second (GFLOPS). It brings the highest level of performance in the C6000 family of floating point DSP.

3.1 C6713 Memory architecture

Figure 3 presents the TMS320C6713 memory architecture. This architecture is organized as describe bellow: The C6713 has a two levels architecture memory

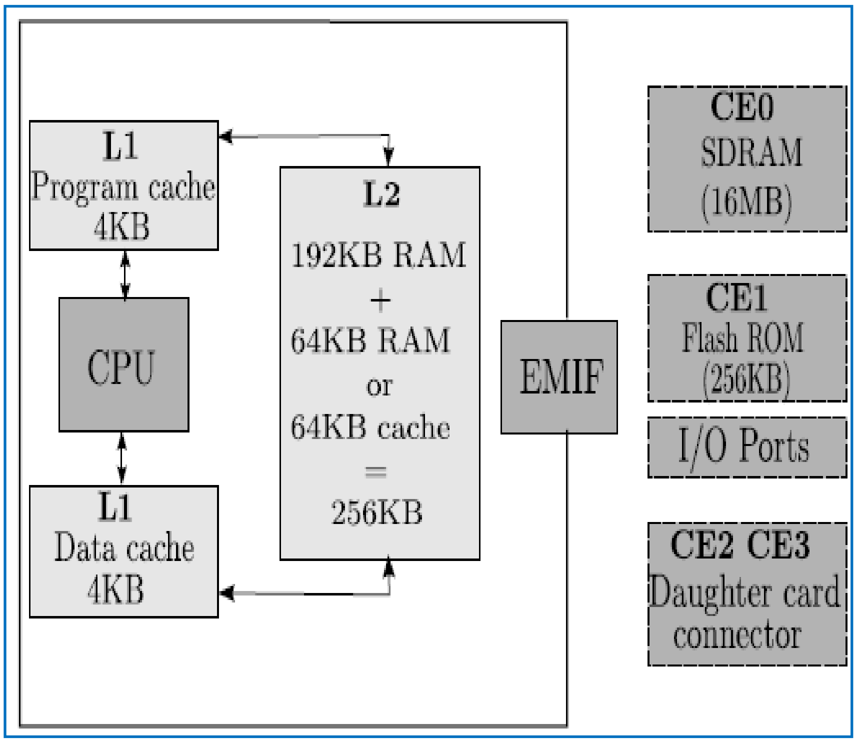


Fig 3: 6713 Memory architecture.

- Level 1 memory is always used for cache. There are 2 caches: one for the data, one for the program. Each one is 4KB.The two level 1 caches are always available and are single cycle access. If the data is not on the L1 cache, L2 is accessed.
- Level 2 memory is 256KB. 64KB can be made cache, the other 192KB are always RAM (data or program).

The External Memory interface (EMIF)[6] is broken in four 128 Mbytes external range, each one has a dedicated strobe (CEx) The DSK uses 4 external Memory regions (CEx).

- CE0 is 16MB and dedicated to SDRAM.
- CE1 is 256KB and used for Flash memory nd I/O Ports
- CE2 and CE3 are available via daughter card connector

The Cache Configuration Register (CCFG) is a 32b register.

Each external memory address space of 16 Mbytes is controlled by one Memory Attribute Register bit (0: non cacheable, 1: cacheable). 16 MAR registers control the memory caheability. MAR0 at address range 0184 8200 controls CE0 range from 8000 0000 to 80FF FFFF. By default the memory is not cachable. This is why the decoding process was so slow. As soon as possible in the program, L2 cache is configured.

The enableL2cache function as written above allows using 64KB of cache. Figure 4 shows the influence of the cache size on the software speed. Tests are done with no cache, 16 KB cache, 32 KB, then 48 KB and finally 64KB. For more information on the C6713 memory management, there is a Texas Instrument datasheet center on this specific point

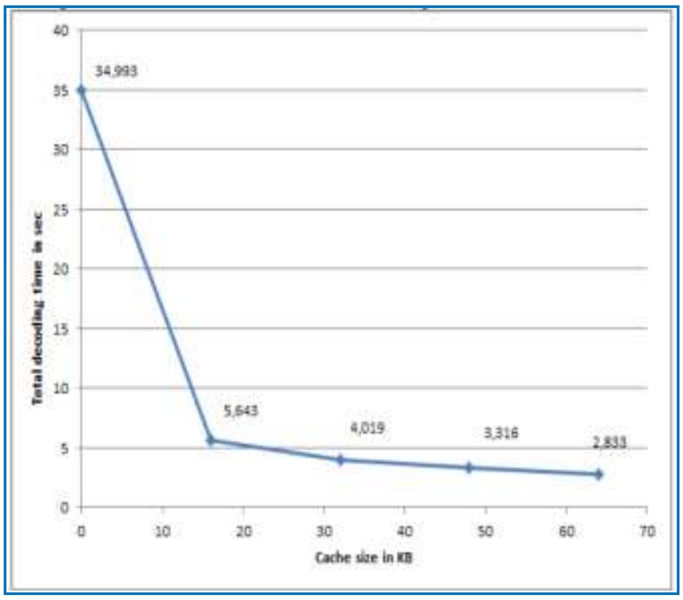


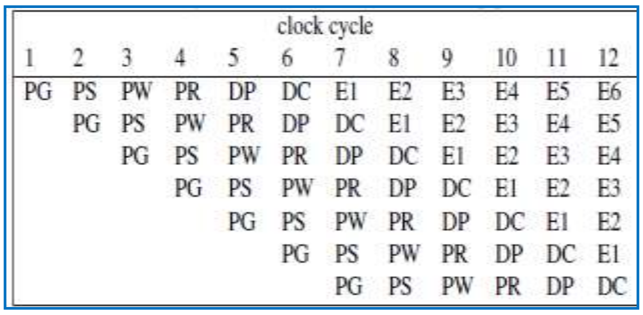
Fig 4: Cache influence: time needed for decoding a 50frames video

3.2: C6713 Pipeline:

All the C6000 DSP family is based on VLIW architecture. The C6713, the last one in the family, is designed to perform floating operation at a high level of performance. The objective of such architecture is to take advantage of instruction level parallelism. In traditional approach of parallelism, all decisions are made internally and the processor has to schedule itself instructions in its pipeline. It implies higher cost, energy consumption and complex hardware. The VLIW[7] approach is totally different. The following section is an introduction on DSP pipeline, a more detailed presentation is available on Texas Instrument user guide. It is not the processor but the program that decides if instructions will be executed simultaneously and how. Consequently the hardware is simpler and the compiler more complex. On the C6713 DSP platform an instruction is 32 bit,
and the Program bus is 256 bits wide. It means that each clock cycle, the C6713 fetches 8 instructions. The 256 bits fetched forms a Very Long instruction Bit Word (VLIW). The CC6713 has 2 data path A and B, and both have 4 functional
units:

- D for loading storing and arithmetic operations
- L for Logical and arithmetic operations
- M for Multiply operations
- S for Branch, bit manipulations and arithmetic operations

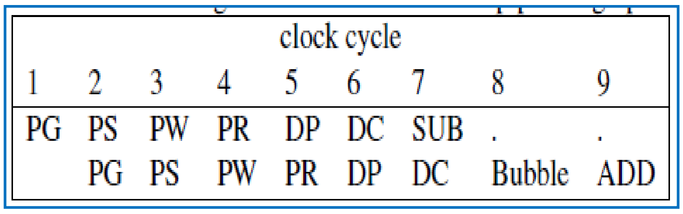
Both of them can perform one operation per cycle. At 225MHz and with 8 units it means 1800 million instructions per second (MIPS). C6713 is a floating point processor. The L M and S units can perform one float operation per cycle, the platform can perform 1.35 giga float operation per second (GFLOPS)



clock cycle											
1	2	3	4	5	6	7	8	9	10	11	12
PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5	E6
	PG	PS	PW	PR	DP	DC	E1	E2	E3	E4	E5
		PG	PS	PW	PR	DP	DC	E1	E2	E3	E4
			PG	PS	PW	PR	DP	DC	E1	E2	E3
				PG	PS	PW	PR	DP	DC	E1	E2
					PG	PS	PW	PR	DP	DC	E1
						PG	PS	PW	PR	DP	DC

Fig 5: An ideal and non realist pipeline

A first instruction is stored on the first line at clock 1. At clock 2 this first instruction is still processing (step Program address send) but a new instruction can be fetched in parallel. At clock 11 the pipeline is full because all eleven instructions are proceeding through the various phases E1 -E10. Most instruction tie up their functional unit for only one phase (E1). This process has limitation and the pipeline presented in figure 5 is not realist. In a real, ninstruction flow bubbles are introduced in our pipeline process. These bubbles represent extra idle cycles.



clock cycle								
1	2	3	4	5	6	7	8	9
PG	PS	PW	PR	DP	DC	SUB	.	.
	PG	PS	PW	PR	DP	DC	Bubble	ADD

Fig 6: A more "actual" pipelining operation

Figure 6 gives a good example, if an instruction needs the result of a previous instruction; the instruction will wait the end of the execution of the previous one. We start with a subtraction operation. The subsequent instruction uses the result of this subtraction as an input parameter. The addition instruction execution cycle is supposed to start at 8 but an extra cycle is added to be sure the ADD will read the result of the subtraction and not the result before the addition occur.

3.3 Code Composer Studio: Build properties:

A first option, necessary for building the project is increasing the stack and the heap size. Running the command valgring ./vp8decode.exe give a heap summary information.

```
==9752== HEAP SUMMARY:
==9752== in use at exit: 0 bytes in 0 blocks
==9752== total heap usage: 1,383 allows, 1,383 frees, 132,601,784 bytes allocated
==9752==
==9752== All heap blocks were freed – no leaks are possible
```

We decide to increase the stack size until the limit of the internal memory. The heap size is also fixed to be as big as possible. On the following experiment, the fixed size is:

–stack size 0xfe00

–heap size 0x800000

3.4: optimization level:

When -O3 [9] is specified, the compiler optimizes primarily for performance. Need adjust the priorities between performance and code size by using the code size flag –opt for space (short in ms). The -ms0, -ms1, -ms2 and -ms3 options increasingly favor code size over performance as seen on **Figure 7.**

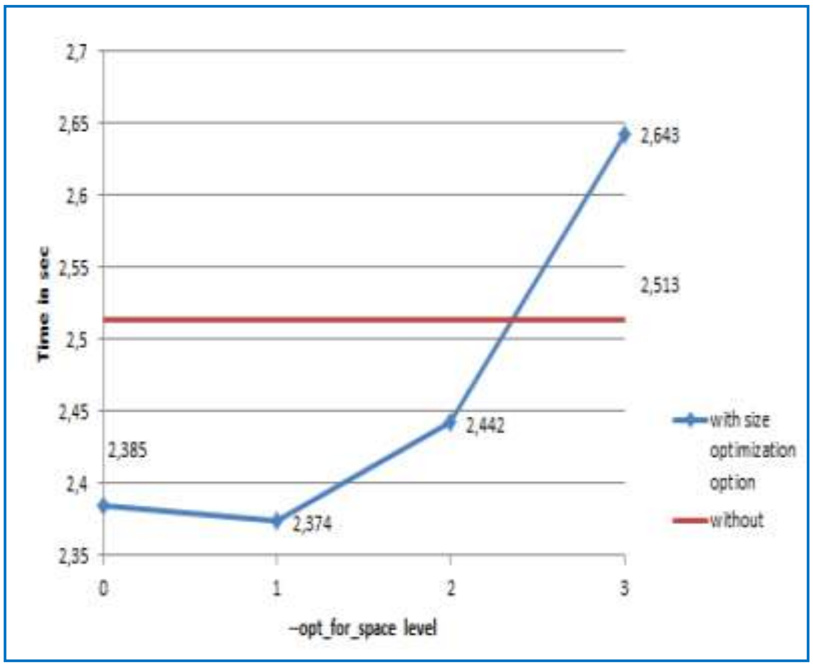


Fig 7: Influence of –opt for space option on decoding time

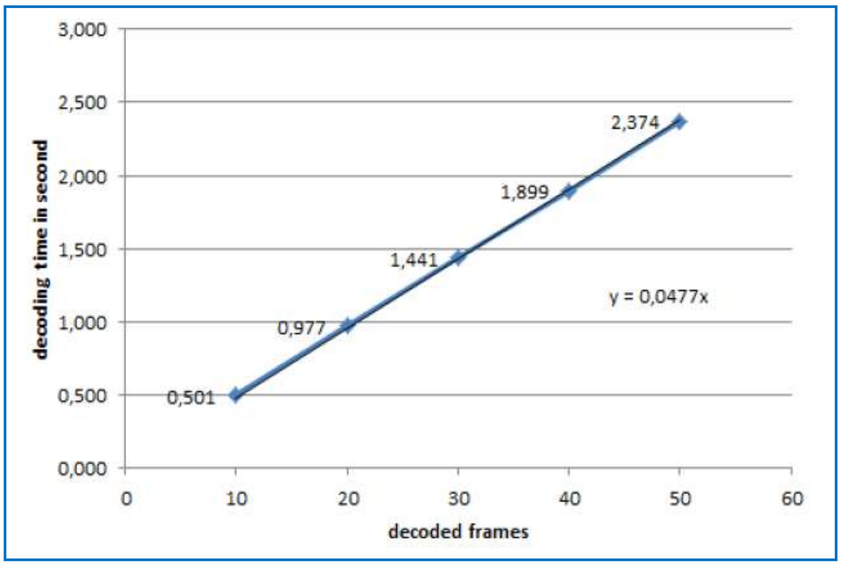
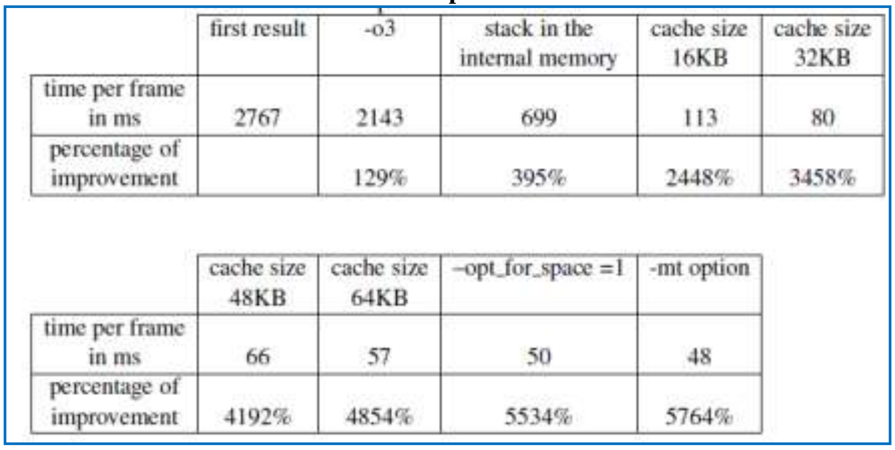


Fig 8: Decoding time, a linear function of frame number

IV. RESULTS AND IMPROVEMENTS

Table 1: Improvements



	first result	-o3	stack in the internal memory	cache size 16KB	cache size 32KB
time per frame in ms	2767	2143	699	113	80
percentage of improvement		129%	395%	2448%	3458%

	cache size 48KB	cache size 64KB	–opt_for_space =1	-mt option
time per frame in ms	66	57	50	48
percentage of improvement	4192%	4854%	5534%	5764%

Fig 9: VP8 Video decoder output onTMS3206713

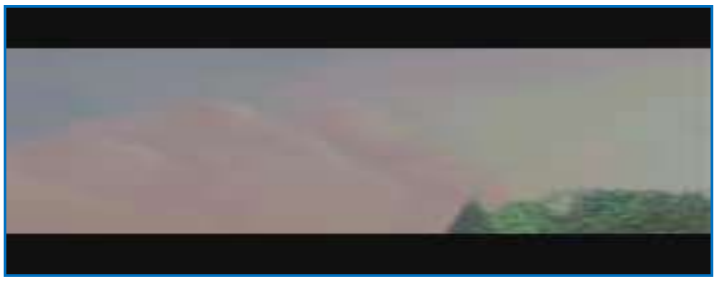


Fig 10: VP8 Video decoder output onTMS3206713

Table-2: Performance analysis of VP8 video decoder

Sequence	PSNR	Bit Rate (Kbps)
vp80-00-comprehensive-002	29.67	33.95
vp80-00-comprehensive-007	37.57	51.85
vp80-00-comprehensive-015	39.32	67.34

4.1. Result analysis

In this paper fig 2 shows that VP8 Codec diagram which we are considering to porting on TMS3206713, and fig 3 showing the memory details of TI6713 platform, similarly fig 4-8shows about cache and pipeline structure of TI6713,Fig 9 and 10 shows the decoded output on TI platform, Similarly Table and Table 2 will give the information about improvements of VP8 video decoder after applying optimization techniques on TI platform and also the PSNR and bitrates[7] results of VP8 video decoder.

Conclusion and Future work

Here we have successfully ported and optimized and also analyse the performance of VP8 video decoder on TMS3206713 platform, This results shows that much improvement than compared Windows based PC and also H.264 on same platform, Also This results are more impressive for multimedia communication applications, Like wise planning to porting also analysis of the VP8 video codec is possible on different platforms.

REFERENCES

[1]. Y. Wang, J. Ostermann, Y. Q. Zhang, Video Processing and Communications, Prentice Hall, 2002. Chapters 9,11,13

[2]. "Text of ISO/IEC FDIS 14496-10/Draft ITU-T H.264: Information Technology – Coding of Audio-Visual Objects: Advanced Video Coding", International Organization for Standardization, 2003

[3]. J. Bankoski, P. Wilkins, Y.Xu, "VP8 Data format and Decoding Guide," http://www.ietf.org/internet-draft/draft-bankoski-vp8-bitstream-01.txt,Jan 2011

[4]. Video Network Traffic and Quality Comparison of VP8 and H.264 SVC by Patrick Seeling, Frank H.P. Fitzek and Gergö Ertli

[5]. ITU-T Recommendation P.910, "Subjective video quality assessment methods for multimedia applications," Sep. 999.

[6]. Texas Instruments. TMS320C6000 Optimizing Compiler v 6.0, July 2005.

[7]. David Strachan "Video Compression "published in SMPTE Journal, February 1996

[8]. MPEG-2, *MPEG-2 Test Model5 (TM5)* Doc.ISO/ IEC/ JTC1/SC29/WG11/N0400, Test Model Editing Committee, April 1993.

[9]. Texas Instruments. TMS320C67x/C67x+ DSP CPU and Instruction Set Reference Guide, November 2006. Literature Number: SPRU733A.

Basavaraju.S$_1$ holds the Master degree from VTU, Karnataka, and currently pursuing PhD at PRIST University Thanjavur, India, and Earlier worked at several software industries in signal and multimedia processing domain and currently working as Assistant professor in Sapthagiri College of engineering, Bangalore.

Dr B Sivakumar$_2$ holds PhD degree from Anna University Tamilnadu, and currently working as Professor and head of the Telecom dept at Dr AIT, Bangalore.