A Loss Less video compression for H.264 video Decoding

M.Chandrasekhar[1], S.Nagaraju[2]
[1]*Student, Department of ECE, VLITS,Vadlamudi,India*
[2]*Asst.prof., Department of ECE, VLITS,Vadlamudi,India*

***Abstract* –The rapid increase of mobile technology in the past decade the number of people using mobile devices is increasing day by day. The number of people using the multimedia is increasing day by day. For proper, efficient and effective usage of multimedia in communication, we have to compact the data. External bandwidth is one of the most critical issues of video coding for high performance and low power concerns. With the limited size of bandwidth more number of users can't use properly. With the increasing technology more number of people are opting for the high quality video. For high quality video the size is more. so that they acquire more bandwidth, for that purpose compression of video is needed. In this golomb rice coding is used to encode the video on raspberry pi board.**

***Index Terms*—lossless compression, embedded compression, video coding.**

I. INTRODUCTION

External Memory access to DRAM dominates the overall performance and power consumption in a modern complex video codec chip, which becomes worse with the increasing demands of higher resolution videos and more and more complex video compression algorithm. One way to solve this problem is to compress data to the external memory and decompress them when needed as shown in Fig. 1, which are often referred as embedded compression and decompression since it operates transparently in the overall processing flow. With embedded compression, we can reduce the transferred data amount, which implies less power consumption, less bandwidth demands, and thus lower cost. To fit above purpose, various embedded compression engines have been proposed, either in lossless or lossy compression. For the lossy compression methods, used DCT or DWT to find the importance of the data and then applied high quantization factor or even discards the less important one to meet target compression ratio. For the lossless compression, Golomb-Rice Coding (GR Coding) has been used extensively for its simplicity and good coding efficiency. In which, used the statistical data to predict the probability distribution of the encoded value for coding, used DWT for better coding efficiency, reduced redundancy by multiple differential data method, and further adopts the Truncated Bit Packing to achieve compression. However, these lossless compression approaches did not address the uncompressible case, that is, the bit stream size could be larger than the original data size. They also failed to address its effectiveness in a real DRAM access scenario, which could reduce the possible access savings due to DRAM latency. Besides, the additional bit stream header is not minimized, which reduces the compression ratio.

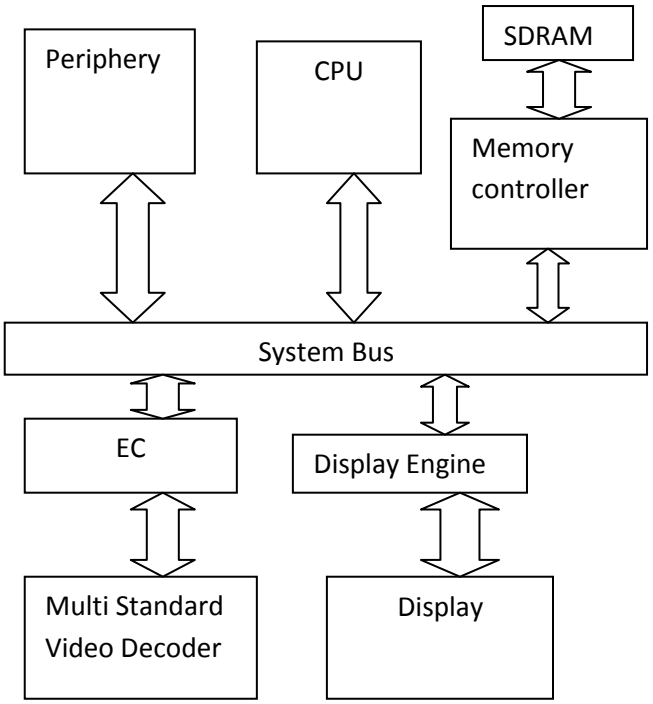

Fig. 1 architecture diagram for embedded compression system

To solve above problem, this paper proposed a simple adaptive differential data and modified GR coding for compression with decoding error resilient scheme for uncompressible case. The final implementation shows it can achieve HD size video compression easily with much lower hardware cost.

II. PROPOSED ALGORITHMS

A. Golomb rice coding

Given a constant M,any symbol S can berepresented as a quotient and remainder.

Where

$$S=Q*M+R$$

If S is small(relative to M) then Q will also be small.rice coding represents Q as a unary value and R as a binary

value. While encoding ,consider the bit length as ,k.compute the k value by the equation M=2^k.then the AND operatio between S&(M-1) and write out S>>K in unary.these two codes are the output from the encoding module.

During decoding process.by using the equation m=2^k,then determine q by counting the number of 1s before the first 0.and determine r value by reading the k bits as a binary value.then write out Q*M+R.

Golomb rice is not meant to be shorter than the shortest binary encoding for one particular number.rather by providing a specific kind of variable length encoding,they reduce the average length per encoded value compared to fixed width encoding,if the encoded values are from a large range,but are using only a small fraction of that range most of the time.

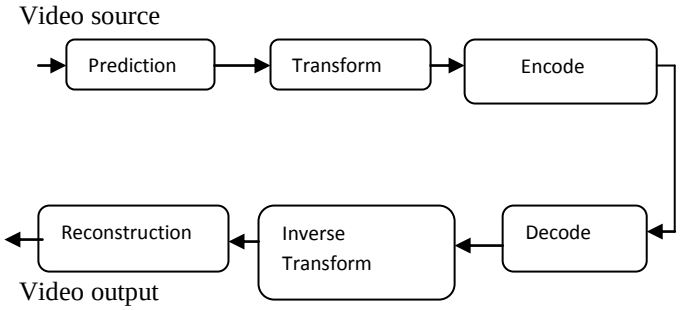

Fig 2: H.264 Encoding and Decoding Process

III HARDWARE IMPLEMENTATION

The hard ware design can be implemented by using raspberry pi .Raspberry pi is a ultra low cost credit card sized linux computer.with video core IV Gpu video signals processing can be done.by writing code for golomb rice algorthim that program can be compiled using gcc compiler.the compressed video can be seen .video is captured by using the raspberry pi camera module.

Fig 3: Raspberry pi Board

For capturing of video raspvid function is used.we can select the no.of frames to be captured for every second.the quantization parameter is defautly selected as 18.raspbin is the Operating system used in the raspberry pi.

Firstly we need to decide which linux distribution will be used in raspberry pi and then the SD card is flashed with operating system. Here raspbian OS is used. The raspberry pi build of raspbian OS includes a desktop environment Lightweight X11 Desktop Environment(LXDE)

IV EXPERIMENTAL RESULTS

In this a video is taken as input, After encoding the video, it is compressed frame by frame the compression is in fig 4 and fig 5.

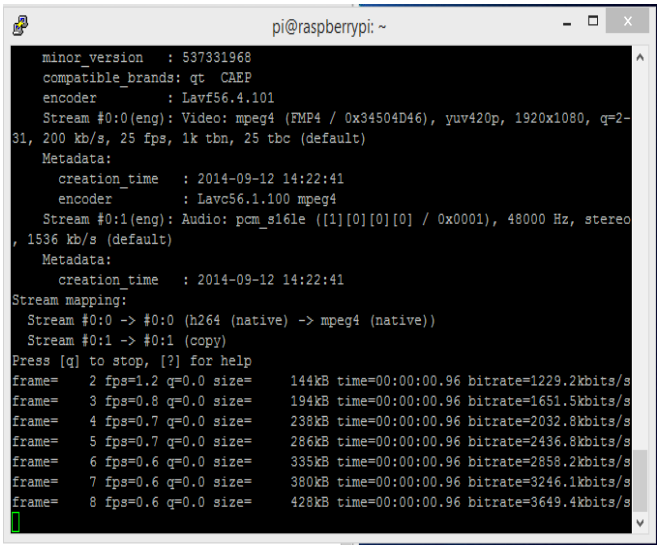

Fig 4: Video Encoding

Initially the video size is 87.79 MB. After encoding the video size is 16.89 MB. Here the encoding is done frame by frame by using golomb rice algorithm.

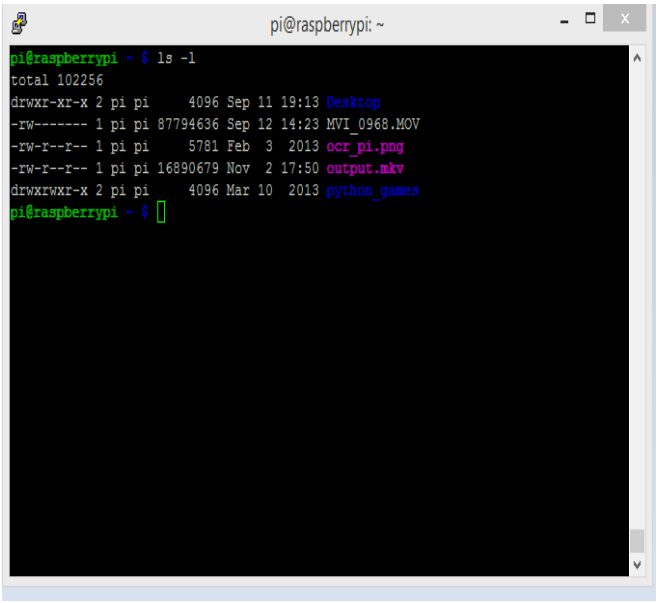

Fig 5: Video encoding

Here the input file is named as MVI_0968 and the output file is named as output.The video Compression ratio which indicates compression capability is defined as

$$CR=1-\frac{COMPRESSED\ DATA\ SIZE}{ORIGINAL\ DATA\ SIZE}$$

From the above formulae, the compression capability is 20%.

V. CONCLUSION

This paper presents a lossless Video compression engine
to solve the memory bandwidth problem in H.264video decoding. Our approach is simple and elegant and can save up to 41% data size. Further extension to lossy compression is possible by extending the approach with appropriate quantization scheme.

REFERENCES

[1] “A lossless embedded codec engine for HD video decoding”
[2] C.-C. Cheng, and et al, ”Multi-Mode Embedded Compression Codec Engine for Power-Aware Video Coding System” in IEEE Transaction on Circuits and System for Video Technology, pp. 141-150, Feb, 2009.
[3] W.-Y. Chen, and et al, "Architecture design of high performance embedded compression for high definition video coding" in IEEE ICME, 2008.
[4] Y.-H. Lee, and et al, "A high-speed lossless embedded compression codec for high-end LCD applications" in IEEE A-SSCC, 2008.
[5] H. C.-Chou "A Lossless Embedded Compression Codec
Engine Integrated with H.264/AVC”