

Design and Implementation of H.264/AVC Encoder Using 3D DCT Architecture

¹Nithin Kurup, ²Mr. Ananthan.P

¹M.Tech Scholar, ²Associate Professor

¹Department of VLSI & ES,

²Department of Electronics and Communication

Abstract— H.264/AVC is a joint project of ITU and MPEG. It provides high quality compression for various services like IP streaming media, SDTV and HDTV broadcast and video on demand etc. Motion estimation, input buffer, summation unit and reference frame section form complex design of H.264/AVC encoder. It is the most advanced video standard. In this paper, H.264/AVC encoder is designed in verilog with better performance and optimized structure. Design is finally implemented on FPGA platform.

Index Terms - H.264/AVC, 3DDCT, Encoder, Implementation, FPGA, Complexity

I. INTRODUCTION

H.264/AVC is the newest international video coding standard. MPEG-2 video coding standard which was developed was an enabling technology for digital television systems world wide. It is widely used for the transmission of standard definition (SD) and high definition (HD) TV signals over satellite, cable, and terrestrial emission and high quality SD video signals onto DVDs. However, an increasing number of services and growing popularity of HDTV are creating greater needs for higher coding efficiency. High quality video coding has evolved through the development of the ITU-T H.261, H.262 (MPEG-2), and H.263 video coding. In order to double the video coding efficiency, a new video coding standard H.264/AVC (Advanced Video Coding) developed in March 2003. Compression quality is balanced and implementation cost is reduced. It supports more flexibility in the selection of motion compensation block sizes and shapes than any other video coding standards. In this paper, H.264/AVC is designed in such a way to deliver better performance without compromising design functionality. It is difficult to design the architecture for the H.264/AVC encoder. H.264/AVC encoder is optimized for the proposed design. A new design model is made of basic encoder components like input buffer, reference frame, summation unit and motion estimation. Then the design is simulated using Xilinx ISE. In the proposed design, 3D DCT architecture is used for final optimized model. Second model is also simulated using Xilinx ISE. Both encoders are implemented on FPGA platform. Results of both encoders are compared. Section II shows details about proposed optimized model and implementation of encoder model using 3D DCT architecture. Section III shows results and Section IV draws conclusion.

II. PROPOSED DESIGN

Input Buffer & Reference Frame Section

The input image is divided into macro blocks. Each macro block consists of the three components Y, Cr and Cb. Y is the luminance component which represents the brightness information. Cr and Cb represent the color information. Due to the fact that human eyes are less sensitive to chrominance than to luminance, the chrominance signals are both sub sampled by a factor of 2 in horizontal and vertical direction. Therefore, a macro block consists of one block of 16 by 16 picture elements for the luminance component and of two blocks of 8 by 8 picture elements for the color components. These macro blocks are coded in inter or intra mode. In Inter mode, a macro block is predicted using motion compensation. For motion compensated prediction a displacement vector is estimated and transmitted for each block (motion data) that refers to the corresponding position of its image signal in an already transmitted reference image stored in memory. In the intra mode, former standards set the prediction signal to zero such that the image can be coded without reference to previously sent information. The prediction error, which is difference between the original and predicted block, is then transformed, quantized and entropy coded. In order to reconstruct the same image on the decoder side, the quantised coefficients are inverse transformed and added to prediction signal. The result is the reconstructed macroblock that is also available at the decoder side. This macroblock is stored in a memory. Macroblocks are typically stored in raster scan order. Whereas the memory contains one video frame in previous standards, H.264/AVC allows storing multiple video frames in the memory. In the encoder, a prediction scheme is also used in intra mode that uses image signal of already transmitted macroblocks of the same image in order to predict the block to code.

Inter –Frame Prediction Process

An interceded frame is divided into blocks known as macro blocks. After that, instead of directly encoding the raw pixel values for each block, the encoder will try to find a block similar to the one it is encoding on a previously encoded frame, referred to as a reference frame. This process is done by a block matching algorithm. If the encoder succeeds on its search, the block could be encoded by a vector, known as motion vector, which points to the position of the matching block at the reference frame. The process of motion vector determination is called motion estimation. In most cases the encoder will succeed, but the block found is likely not an exact match to the block it is encoding. This is why the encoder will compute the differences between them. These

residual values are known as the prediction error and need to be transformed and sent to the decoder. To sum up, if the encoder succeeds in finding a matching block on a reference frame, it will obtain a motion vector pointing to the matched block and a prediction error. Using both elements, the decoder will be able to recover the raw pixels of the block.

Motion Estimation

Motion estimation is a process of determining motion vectors that describe the transformation from one 2D image to another; usually from adjacent frames in a video sequence. It is an ill-posed problem as the motion is in three dimensions but the images are a projection of the 3D scene onto a 2D plane. The motion vectors may relate to the whole image or special parts, such as rectangular blocks, arbitrary shaped patches or even per pixel. The motion vectors may be represented by a translational model or many models that can approximate the motion of a real video camera, such as rotation and translation in all three dimensions and zoom.

DCT & Quantization

The DCT is a one dimensional process that has 8 basic functions, which represent the frequency domain. Each basis function is the pixel pattern that results when that particular DCT coefficient is set to its maximum value and all the other coefficients are set to zero. The quantization step involves multiplying the output of the DCT stage with a set of predefined values from a quantization table. Quantization is basically a division process that is converted into multiplication by simply inverting the quantization table values.

Zigzag Scan

Each block of data that is output by the quantization module needs to be reordered in a zigzag fashion before being forwarded to the entropy encoder. This reordering is achieved using an 8 by 8 array of registers organized in a fashion similar to the transpose buffer.

Entropy encoder

The function of the entropy encoder is to code the quantized coefficients from the encoder model using the variable length encoding. Run length coding is a simple thought that is accomplished by assigning a code, run length and size, to every non-zero value in the quantized data stream. The size is a category given to the non-zero value which is used to recover the value later. Huffman coding is a technique which will assign a variable length codeword to the input data item. Huffman coding assigns a smaller codeword to an input that occurs more frequently. It is very similar to Morse code, which assigned smaller pulse combinations to letters that occurred more frequently. This is the last step in the encoding process. It organizes the data stream into a smaller number of output data packets by assigning unique codewords that later during decompression can be reconstructed without loss.

Proposed Design Using 3D DCT

Optimized model is replaced by proposed model using 3D DCT architecture. 3D DCT is designed by calling 1D DCT (Discrete Cosine Transform) three times. A discrete cosine transform expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering, from lossy compression of audio and images to spectral methods for the numerical solution of partial differential equations. The use of cosine than sine functions is critical for compression, since it turns out that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of boundary conditions.

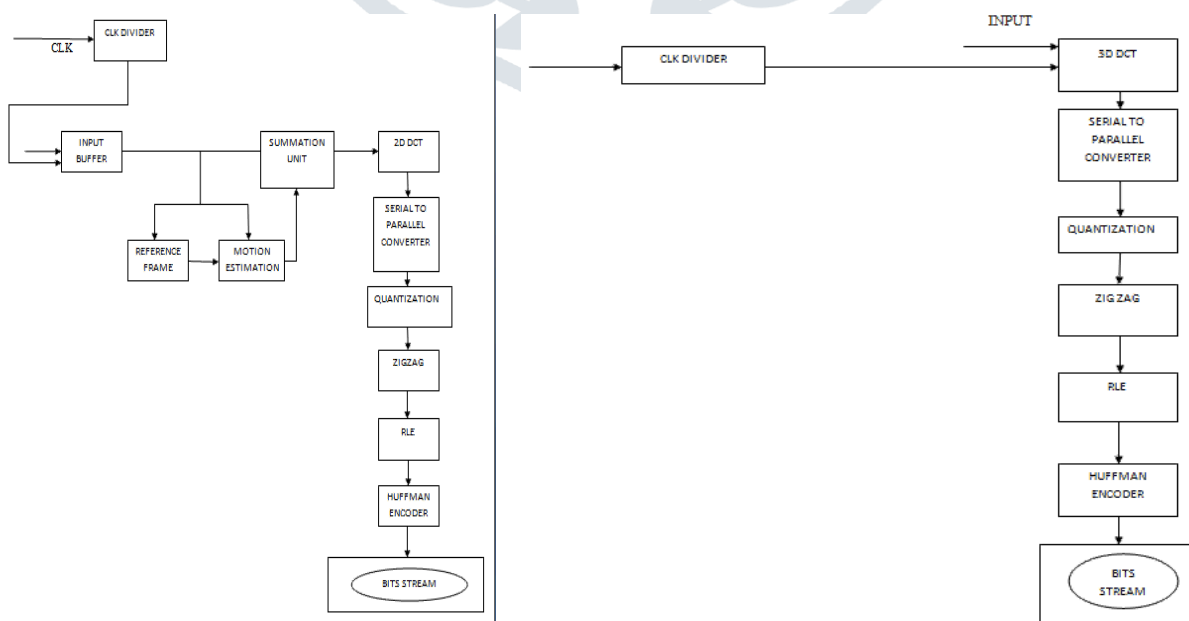


Fig 1. Architecture Of H.264 Encoder (Optimized Model) And Proposed Design Using 3D DCT

III. RESULTS

After the functional simulation of both optimized design and proposed design using 3D DCT are simulated using Xilinx ISE 14.2.Both designs are designed using verilog HDL.(Hardware Description Language).Clock divider is used in both designs. Simulation results are obtained and device utilization reports are observed.

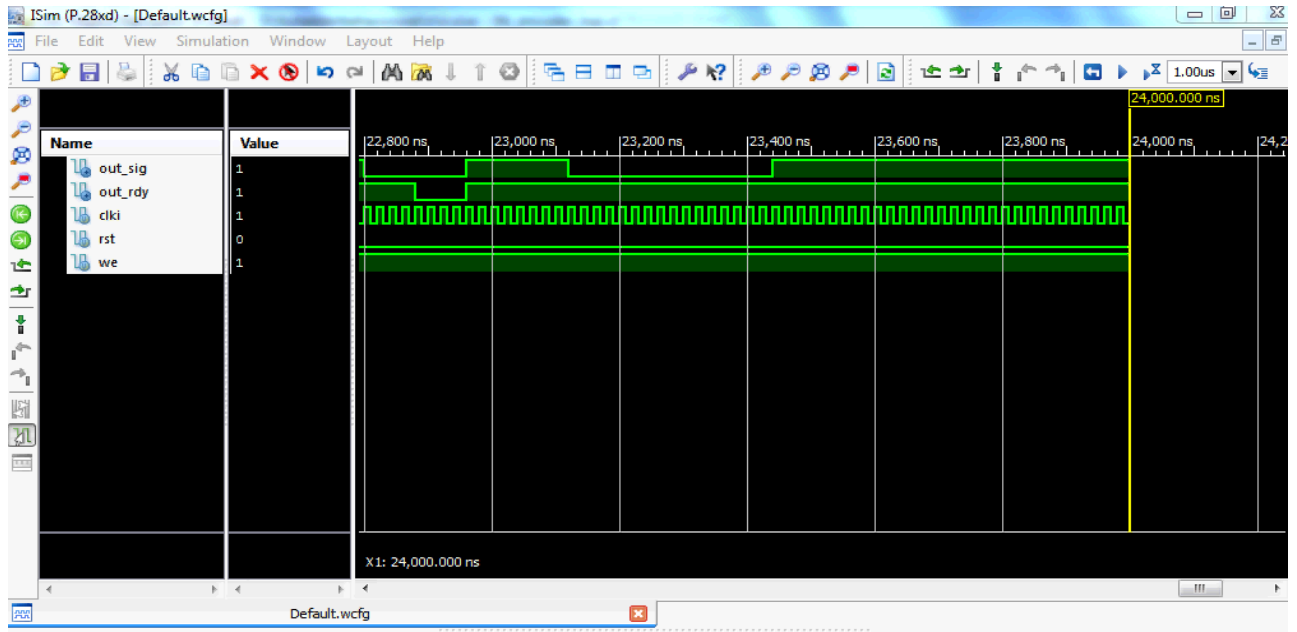


Fig 2.Simulation Result For Proposed Design

From the results obtained for optimized model, it can be seen that 94% of total number of slices are used, thereby resulting more memory usage. Complexity in design is another disadvantage of this design even though it reduces H.264/AVC Encoder model. In the proposed design, complex blocks like motion estimation, summation unit, input buffer and reference frame are replaced. Hence complexity is reduced compared to that of optimized design .Delay is also reduced in proposed design. Complexity in coding is also reduced in proposed design.

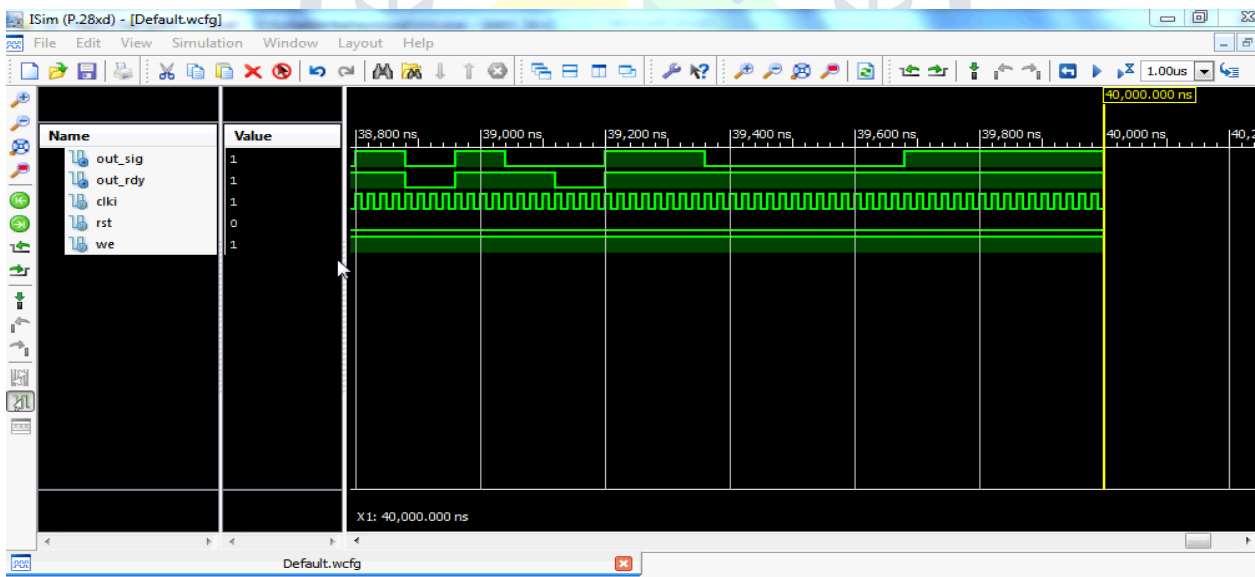


Fig 3.Simulation Result For Optimized Design

```

Device utilization summary:
-----
Selected Device : 3s250etq144-4

Number of Slices:                2320 out of 2448 94%
Number of Slice Flip Flops:      3103 out of 4896 63%
Number of 4 input LUTs:         2300 out of 4896 46%
    Number used as logic:        1596
    Number used as Shift registers: 512
    Number used as RAMs:         192
Number of IOs:                   6
Number of bonded IOBs:           6 out of 108 5%
Number of BRAMs:                 1 out of 12 8%
Number of MULT18X18SIOs:        12 out of 12 100%
Number of GCLKs:                 2 out of 24 8%
    
```

Fig 4.Device Utilization Report For Optimized Design

Previous designs utilized 94% of slices, resulting high memory usage. Delay is also high in case of previous design. Slice utilization reduced to 22% compared to design without 3D DCT. Complexity in coding of H.264/AVC encoder is reduced and motion estimation section is replaced using 3D DCT architecture. It can be understood that previous designs had memory usage high thereby resulting complexity in design and in coding. The proposed design with minimum slice utilization

```

Device utilization summary:
-----
Selected Device : 3s250etq144-4

Number of Slices:                553 out of 2448 22%
Number of Slice Flip Flops:      188 out of 4896 3%
Number of 4 input LUTs:         1223 out of 4896 24%
    Number used as logic:        1031
    Number used as RAMs:         192
Number of IOs:                   6
Number of bonded IOBs:           5 out of 108 4%
Number of MULT18X18SIOs:        12 out of 12 100%
Number of GCLKs:                 2 out of 24 8%
    
```

Fig 5.Device Utilization Report For Proposed Design

IV. CONCLUSION

It can be concluded that proposed design has better performance compared to that of previous designs. Previous designs utilized 94% of slices, resulting high memory usage. Delay is also high in case of previous design. Slice utilization reduced to 22% compared to design without 3D DCT. Complexity in coding of H.264/AVC encoder is reduced and motion estimation section is replaced using 3D DCT architecture. Proposed design ensures less memory usage and delay.

V. ACKNOWLEDGMENT

Thanks to my project guide MR.ANANTHAN.P for his proper guidance and support for completion of this project.

REFERENCES

- [1] Teng Wang,Chih-Kuang Chen,Qi-Hua Yang,and Xin-An Wanganxinwang@pku.edu.cn FPGA IMPLEMENTATION AND VERIFICATION SYSTEM OF H.264/AVC ENCODER FOR HDTV APPLICATIONS,D.Jin and S.Lin (Eds.):Advances in CSIC Vol.2,AISC 169,pp,345-352,2012
- [2] Tung-Chien Chen,Shao-Yi Chien,Yu-Wen Huang,Chen-Han Tsai,Ching-Yeh Chen,and Liang-Gee Chen,Fellow,IEEE,Analysis and Architecture Design of an HDTV 720p 30 Frames/s H.264/AVC Encoder IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY,VOL.16,NO.6,JUNE
- [3] Thomas Wiegand,Gary J.Sullivan,Senior Member,IEEE,Gisle Bjontegaard,and Ajay Luthra,Senior Member,IEEE Overview of the H.264/AVC Video Coding Standard IEEE Overview of the H.264/AVC Video Coding Standard IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY,VOL.13,NO.7,JULY 2003