Implementation of H.264 Decoder

Shriram K Vasudevan
Senior Engineer, Aricent Technologies
India.

Subashri V
I Year, M.Tech Computer Science
B.S.Abdur Rahman University, India

Sasikumar P
Assistant Professor, VIT
Vellore, TamilNadu
India.

ABSTRACT
The H.264/AVC is the most recent standard of video compression/decompression for future broadband network. This standard was developed through the Joint Video Team (JVT) from the ITU-T Video Coding Experts Group and the ISO/IEC MPEG standardization committee. In this project H.264 decoder functional block such as Context based Binary arithmetic coding (CABAC), Inverse Quantization and Inverse Discrete Cosine Transform are designed using Verilog. CABAC includes three basic building blocks of context modeling, binary arithmetic coding and Inverse binarization. Here the compressed bit-stream from NAL unit is expanded by CABAC module to generate various syntax elements. Here the basic arithmetic decoding circuit units are designed to share efficiently by all syntax elements. Inverse Quantization and Inverse Discrete Cosine Transform functional blocks are used to reconstruct the original image pixels values

Keywords
MPEG-2, H.264, MPEG-4 Part 10, AVC, digital video codec standard, Lossy compression, lossy transform codecs, lossy predictive codecs

1. INTRODUCTION
The H.264 is a new video compression scheme that is becoming the worldwide digital video standard for consumer electronics and personal computers. In particular, H.264 has already been selected as a key compression scheme (codec) for the next generation of optical disc formats, HD-DVD and Blu-ray disc. (Sometimes referred to as BD or BD-ROM) H.264 has been adopted by the Motion Picture Experts Group (MPEG) to be a key Video compression scheme in the MPEG-4 format for digital media exchange. H.264 is sometimes referred to as "MPEG-4 Part 10" (part of the MPEG-4 Specification), or as "AVC" (MPEG-4's Advanced Video Coding). This new compression scheme has been developed in response to technical factors And the needs of an evolving market:

- MPEG-2 and other older video codec's are relatively inefficient.
- Much greater computational resources are available today.
- High Definition video is becoming pervasive, and there is a strong need to store And transmit more efficiently the higher quantity of data of HD (about 6 times more than Standard Definition video)

H.264, MPEG-4 Part 10, or AVC (for Advanced Video Coding), is a digital video codec standard that is noted or achieving very high data compression. It was written by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC Moving Picture Experts Group (MPEG) as the product of a collective partnership effort known as the Joint Video Team (JVT). It aims to increase compression rate significantly while transmitting high quality image at both high and low bit rates. Three profiles have first been defined, each with several levels.

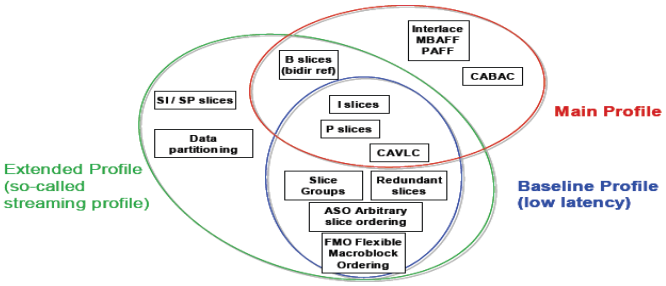

Fig 1. Overview

2. PROBLEM STATEMENT
Nowadays, a large number of consumer products such as digital cameras, Personal Digital Assistants, video telephony, portable DVD player as well as storage, broadcast and streaming of standard definition TV are common practice. All those applications demand efficient management of the large amount of video data. This motivated a large body of research in industry as well as in academia to develop advanced video coding technology. H.264/AVC video coding standard developed by the ITU-T/ISO/IEC is the latest international standard developed by ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group.

The new standard provides gains in compression efficiency of up to 50% over a wide range of bit rates and video resolutions compared with the former standards. Averagely 30~40 cycles are needed to decode a single bin on DSP. That means for such typical 4M bit stream, averagely about 1.5× 100× 30=4500 cycles are needed simply to implement the arithmetic decoding task for one MB, while the cycles for other controls are not counted in. This speed is unacceptable for real time applications, where 30 frames of D1 resolution are required to be decoded within 1s with 100MHz clock, i.e. a MB has to be decoded within at most 2000 cycles.

So hardware acceleration is necessary for a commercially viable H.264/AVC based video application, especially with increase in image size and quality settings in the future. To speed up the decoding process, multiplication-free logic is used to calculate subinterval range.

3. METHODOLOGIES
3.1. Background
H.264/AVC is a new recommendation international standard published jointly by ITU-T VCEG (Video Coding Experts Group) and ISO/IEC MPEG (Moving Picture Experts Group) the main purpose of this standard is to provide a broad

range of multimedia applications with higher reliability and efficient encoding and decoding performance when transporting regular video through various networks compares to former standards. As H.264/AVC achieves enhanced compression rate and better error resilience by employee some unique techniques, the computation complexity of coding is also increased.

3.2. Procedure

3.2.1. Basic concepts of video compression

3.2.1.1 Compression:

The process of coding that will effectively reduce the total number of bits needed to represent certain information. Compression is useful due to reduction in volume of multimedia information and reduction in bandwidth needed for transmission of Multimedia.

3.2.1.2 Lossy compression

A lossy compression method is one where compressing data and then decompressing it retrieves data that may well be different from the original, but is close enough to be useful in some way. Lossy compression is most commonly used to compress multimedia data (audio, video, still images) especially in applications, such as streaming media and internet telephony.

There are two basic lossy compression schemes:

In lossy transform codecs, samples of picture or sound are taken, chopped into small segments, transformed into a new basis space, and quantized. The resulting quantized values are then entropy coded.

In lossy predictive codecs, previous and/or subsequent decoded data is used to predict the current sound sample or image frame. The error between the predicted data and the real data, together with any extra information needed to reproduce the prediction, is then quantized and coded. In some systems the two techniques are combined, with transform codecs being used to compress the error signals generated by the predictive stage.

3.2.1.3 Lossless compression

Lossless compression is a class of compression algorithms that allows the exact original data to be reconstructed from the compressed data. It is often used as a component within lossy data compression technologies. Lossless compression is used when no assumption can be made on whether certain deviation is uncritical.

3.2.2 Pixel

The basic unit of the composition of an image on a television screen, computer monitor, or similar display. To produce a digitized version of the camera output, each scan line can be divided into a series of small areas, called "pixels", generally, the smallest addressable unit on a display screen or bitmapped image.

Each pixel can be encoded as a set of 3 integers: Red, Green, and Blue or Luminance, Y, Blue color difference, Cb, and Red color difference, Cr (Cg can be computed from these.)

3.3Block Diagram

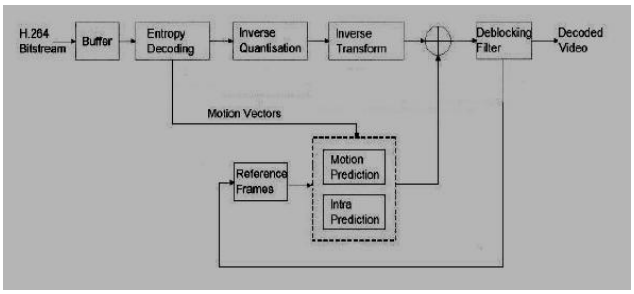

Fig 2. Block diagram

3.3.1 Functional Blocks of H.264 Decoder

- Entropy decoding
- Inverse Quantization
- Inverse Transform
- Motion compensation
- Intra prediction
- De-blocking filter

4. TOP LEVEL MODULE AND SIMULATION & SYNTHESIS RESULTS

4.1. Top level modules and simulation results

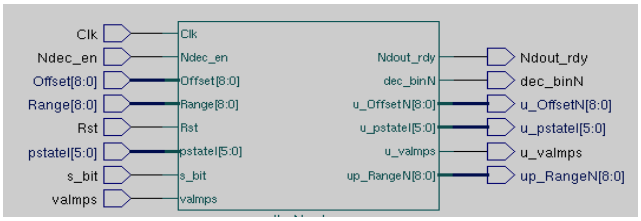

Fig 4.1 Top level Module for Normal decoding process

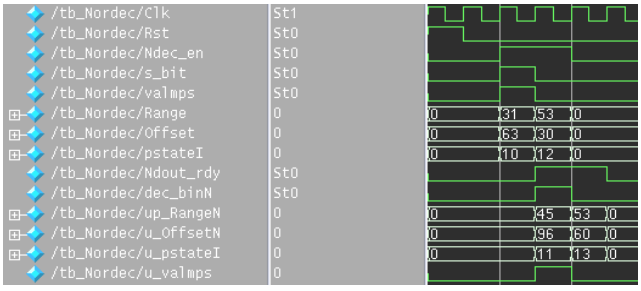

Fig 4.2 I/O waveforms for Normal decoding process

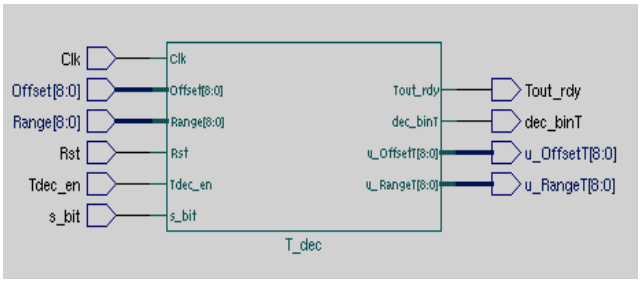

Fig 4.3 Top level Module for Terminal decoding Process

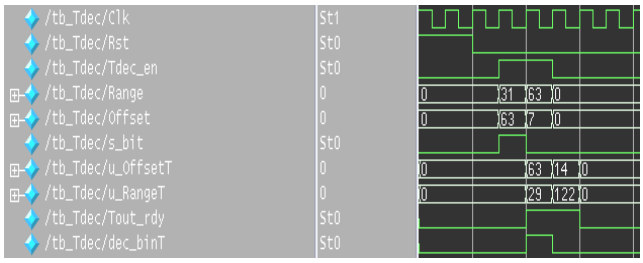

Fig 4.4 I/O waveform for terminal decoding process

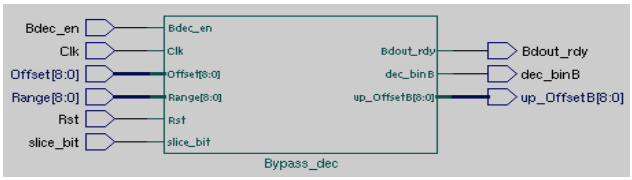

Fig 4.5 Top level Module for Bypass decoding block

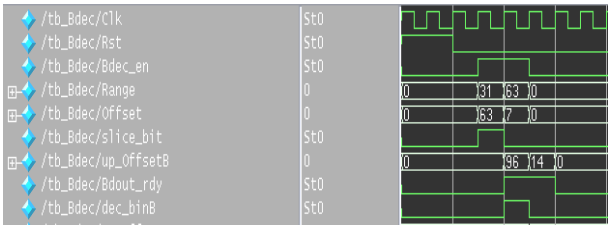

Fig 4.6 I/O waveforms for the By-pass decoding process

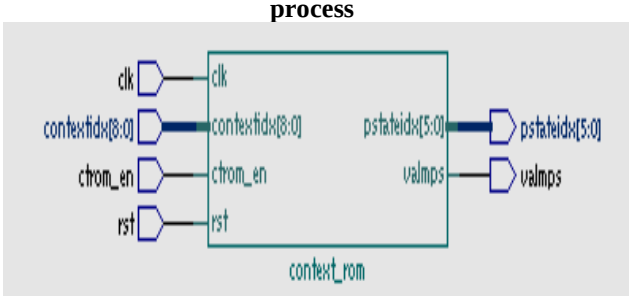

Fig 4.7 Top Level for Context Rom

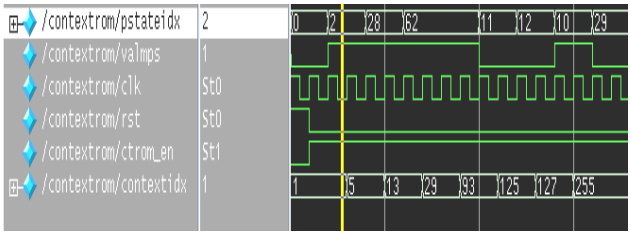

Fig 4.8 I/O wave form for context Rom

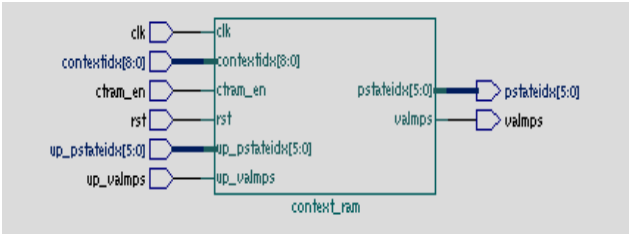

Fig 4.9 Top level Module for context Ram

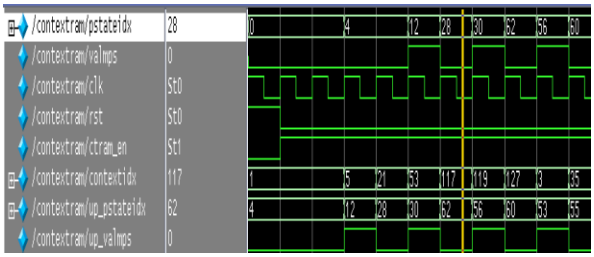

Fig 4.10 I/O waveform for Context Ram

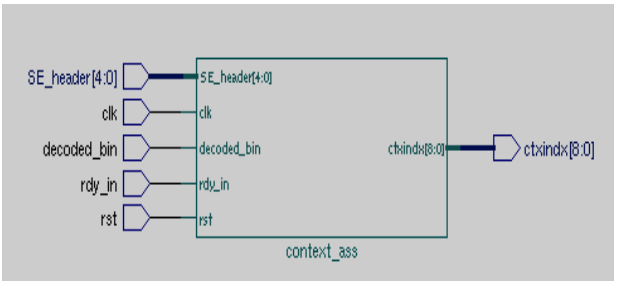

Fig 4.11 Top level module for Context assignment Bin

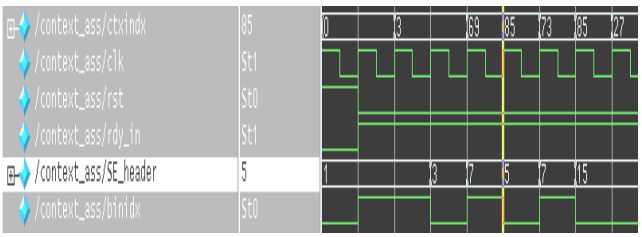

Fig 4.12 I/O Waveform for Context assignment Bin

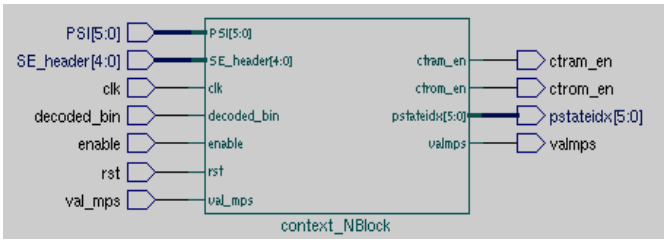

Fig 4.13 Top level Module for Context NBlock

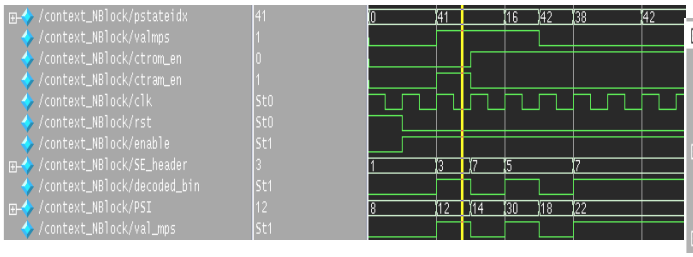

Fig 4.14 I/O wave forms for Context NBlock

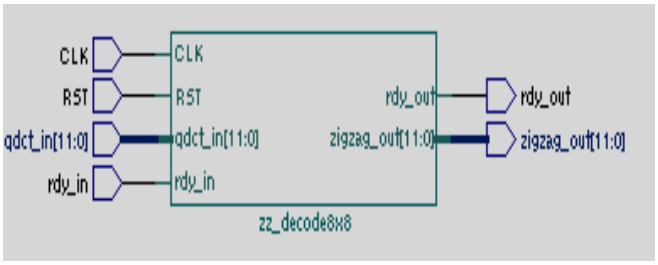

Fig 4.15 Top level module inverse zigzag coding

4.2 Simulation Result for Zigzag Decoding

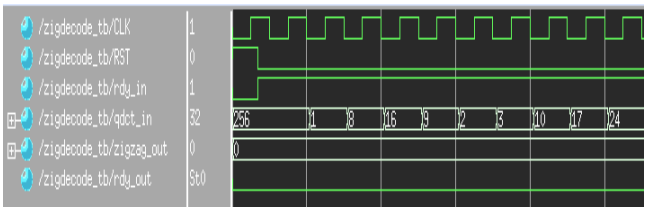

Fig 4.16 Input Waveform

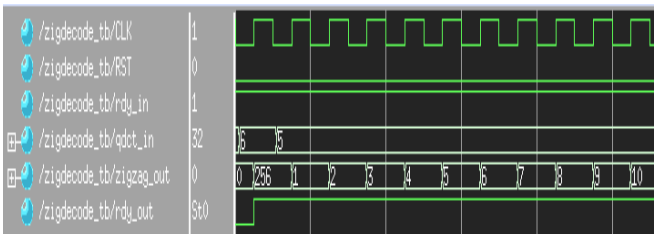

Fig 4.17 Output Waveform

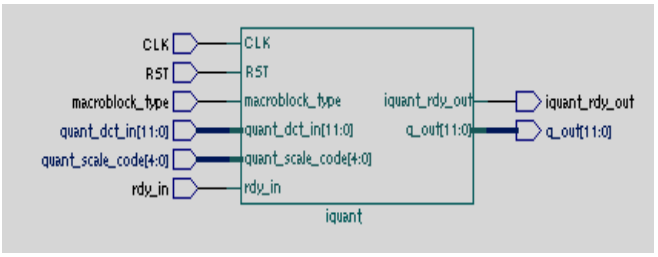

Fig 4.18 Top level module for Inverse Quantization

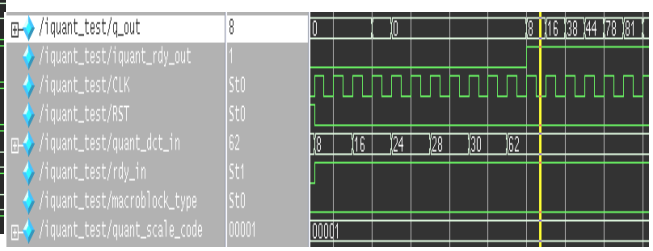

Fig 4.19 I /O waveform for inverse Quantization

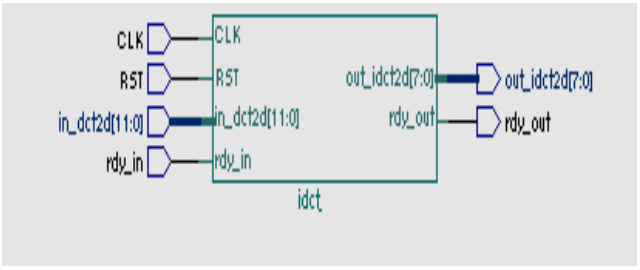

Fig 4.20 Top level module for IDCT

4.3 Simulation Results for IDCT

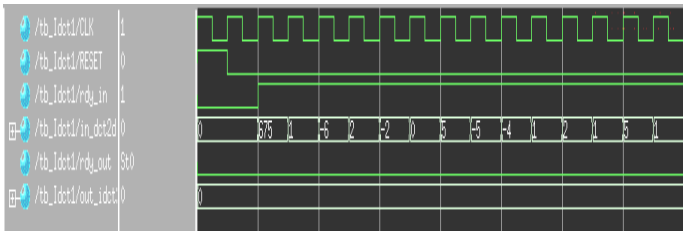

Fig 4.21 Input waveform

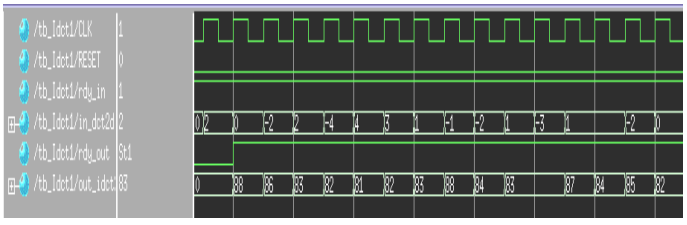

Fig 4.22 Output waveform

4.4 Synthesis Report

```
*********************************************
Device Utilization for 2V500fg256
*********************************************
Resource                  Used   Avail  Utilization
-------------------------------------------------------
IOs                         59     172     34.30%
Global Buffers               1      16      6.25%
Function Generators       5114    6144     89.24%
CLB Slices                2557    3072     89.24%
Dffs or Latches           2622    6660     39.37%
Block RAMs                   0      32      0.00%
Block Multipliers           28      32     87.50%
Block Multiplier Dffs        0    1152      0.00%
-------------------------------------------------------
```

Fig 4.23 Synthesis Report

```
data required time  (default specified - setup time)  5.56

-------------------------------------------------------------------

data required time     5.56

data arrival time      4.29

                     ----------

Slack                  1.26

                     ----------

-------------------------------------------------------------------
```

4.5 Comparison

Here in this project comparing to the other technologies of the same type, this has better video quality, less storage requirement and better compression rate.

5. CONCLUSION

In this project H.264 decoder functional blocks such as Context based Binary arithmetic coding (CABAC), Inverse Quantization and Inverse Discrete Cosine Transform are designed using Verilog to increase the speed of decoding operation. Since CABAC decoding is a highly time consuming process, CPU or DSP is not being the appropriate choice for real-time CABAC decoding applications. This project work shows that the hardware design of CABAC Decoder, Inverse Quantization and Inverse Discrete Cosine Transform are possible for a commercially viable H.264/AVC based video application, especially with increase in image size and quality settings in the future.

6. FUTURE DEVELOPMENTS

In this project work, CABAC decoder, Inverse Quantization and Inverse Discrete Cosine Transform are designed using Verilog to increase the speed of decoding operation. Since CABAC is a key technology adopted in H.264/AVC standard, it offers a 16% bit-rate reduction when compared to baseline entropy coder while increasing access frequency from 25% to 30%.So CABAC decoding is a highly time consuming process. Multiple decoding engines and shared memory between the modules can be implemented in future to increase the decoding speed especially to suite for high bit rate applications such as HDTV, High Definition DVD, Broadcast and Streaming, Digital Television. So Much space is left for real-time applications of higher video quality and larger image resolutions in the future.

4. ACKNOWLEDGMENTS

My sincere thanks to all the co authors.

5. REFERENCES

[1] Vcodex White Paper, http://www.vcodex.com/h264overview.html , Copyright (c) Iain Richardson, 2007-2008.

[2] H. 264 Video coding tutorial, http://www.brighttalk.com/webcast/7022 22/03/2010

[3] International Journal of Digital Multimedia Broadcasting Volume 2009 (2009), Article ID 141986, 8 pages doi:10.1155/2009/141986, Intra-Skip in Inter-Frame Coding of H.264/AVC, Hui Su.

[4] Intra/inter algorithm for B frame processing in H.264/AVC encoder in INTERNATIONAL JOURNAL of COMMUNICATIONS, by Zoran M. Milicevic and Zoran S. Bojkovic, www.naun.org/journals/communications/c-02.pdf

[5] Journal of Electronic Imaging, Adaptive mode-dependent scan for H.264/AVC intracoding, J. Electron. Imaging, Vol. 19, 033008 (2010); doi:10.1117/1.3476330,Published 11 August 2010.