International Journal of Mechanical Engineering

A Design of H.264 High Profile Intra Frame Encoder using FPGA

Kibum Suh
Professor, Dept. of Railway electrical system, Woosong University,
171 Dong-Deajeon-Ro Dong Ku,Deajeon,34640 South Korea
kbsuh@wsu.ac.kr

Abstract

Background/Objectives: In this paper, H.264 HP intra frame encoder using FPGA, which incorporates video input module(VIM), intra prediction module, context-based adaptive variable length coding, and DDR2 DMAC module, is proposed.
Methods/Statistical analysis: In order to verify the hardware, the simulation of hardware using test vector generated by reference C is performed. The proposed modules of encoder is operated in 440 cycle for one-macroblock, respectively.
Findings: The designed encoder is implemented on the Huins's SOC platform which has Vertex5 FPGA and DDR2 memory. It uses of 200 MHz for DMAC, 50 MHz for the Encoder module, and 25 MHz for VIM. It generates the H.264 bitstream in real time, and the generated bitstream is verified using software h.264 decoder.
Improvements/Applications: Since the proposed design can be implemented on a small FPGA, it is suitable for low cost applications.

Keywords: high profile, intra frame encoder, H.264/AVC, DDR memory, low cost FPGA

1. Introduction

H.264 was developed for the purpose of providing a means to improve picture quality more than any other existing video compression standards. The basic concept itself is similar to H.263 and MPEG-4, but it adopts a largely changed method for detailed internal implementation. Referring to the case of motion estimation/ compensation (motion estimation/compensation) is H.264 H.263 provide the shape and size of the other blocks, unlike that provided by the same size, and all blocks, 1/4 pel motion estimation , Multiple reference frame selection, etc. are supported.
In transform, it is possible to perform high-speed operation by using an integer-based transform instead of the discrete cosine transform used by previous video compression standards. Also, this method has no error due to mismatch in the inverse transform. Entropy coding methods include universal variable length coding (UVLC), context-based adaptive variable length coding (CAVLC), and context-based adaptive binary arithmetic coding. , CABAC) is applied, and more efficient compression is achieved.
In this paper, HDL was designed and integrated into an encoder suitable for H.264/AVC high profile level 4[1], and ported to Vertex 5 xilinx FPGA. In order to verify the proposed encoder structure, a reference code was developed from JM13.2[2], and a test vector was extracted from the reference code to verify the designed circuit. Micron's DDR2 SDRAM was used as a frame buffer[3], and it was confirmed that real-time encoding was performed by porting it to Huins' SOC Platform. The scope of this paper is to develop an encoder that codes only Intra-frame among H.264 high profile encoders.
The scope of this paper is to design an intra frame encoder that encodes only I-frames in an H.264 high profile encoder. Figure 1 shows the block diagram of the H.264 high profile encoder, and the module inside the red square is a block to be used in this study, and is intended to be implemented in a small FPGA.
In the case of such H.264 encoder, there are research results [4] on ASIC implementation for intra frame, and in the case of implementation using FPGA, there are results of several studies [5-8]. In the case of this paper, since the research result is for intra frame, we will compare it with research [4].

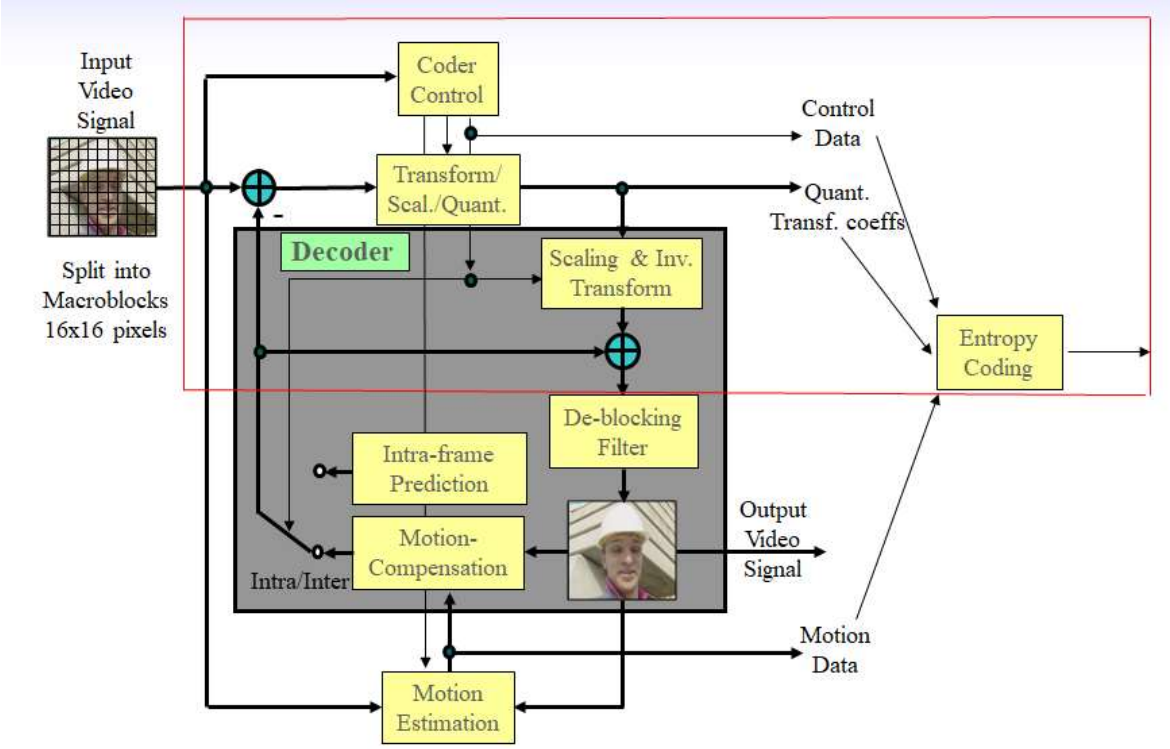


Figure 1. The scope of the proposed encoder

2. Hardware Architecture

The overall architecture of the encoder used in this study is shown in Figure 2. Inside the FPGA, there are 6 modules of the encoder, each of which is a VIM module that receives image data from CIS (CMOS image sensor), a DDR2 control module that enables use of DDR2 memory, and an intra DCT/Q module that plays the role of H.264 intra prediction encoding. CAVLC module that plays the role of variable length coding, and controller module that controls the entire encoder. After receiving the image from the CIS camera and storing it in the DDR2 memory the image data is read in MB units from the DDR2 memory, transferred to SRAM buffer and transmitted to the Intra DCTQ/Q module. In the Intra DCT/Q module, intra-prediction encoding is performed, the data is encoded using CAVLC, and encoded bitstreams are transmitted to a PC via FT245 using a USB interface.

The encoder consists of five pipelines that can continuously process video. The number of cycles required for one macroblock is limited to 440, and even if one macroblock is processed before 440 cycles, a new macroblock is processed in the next pipeline after waiting for the remaining cycles.

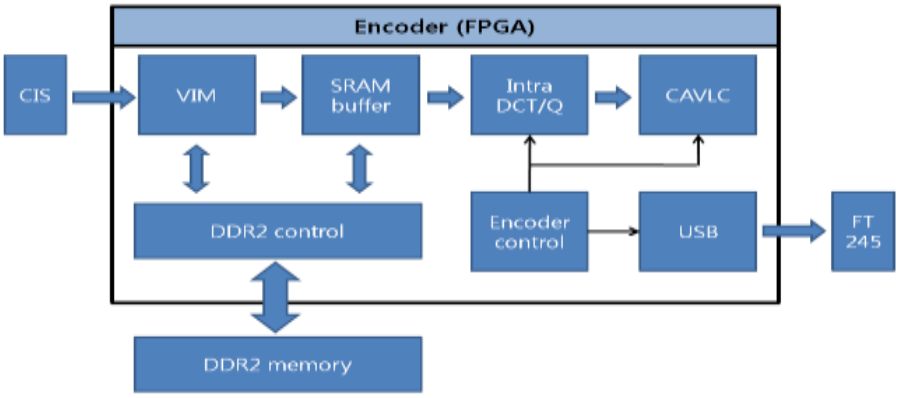


Figure 2. Hardware architecture of implemented encoder

2.1. DDR2 controller

The DDR2 Control (DMA) hardware module supplies and stores the data required for the intra prediction coding module[9-10]. When the image input module saves the original image from the image sensor in the SRAM buffer area as shown in Figure 3, the ddr controller stores the saved data in the original image area of the SDRAM.

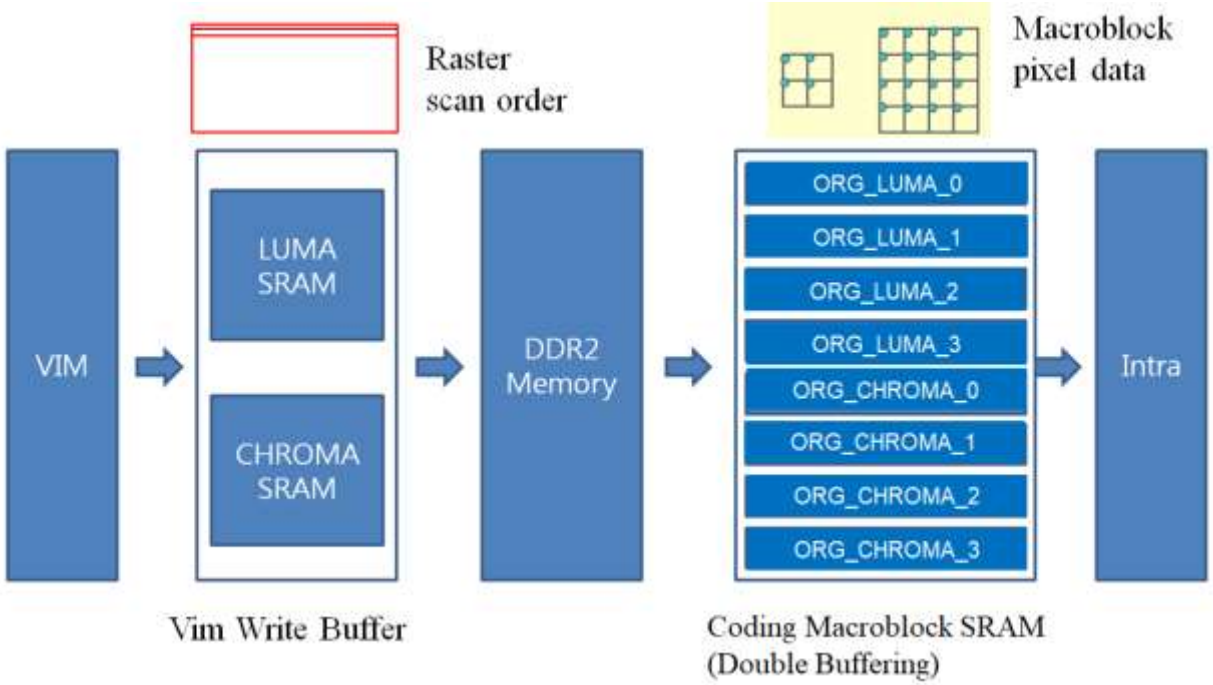


Figure 3. The structure of DMA buffer

After 16 lines of image input are stored in SDRAM, the ddr controller starts reading the image in macroblock units from the SDRAM and transmitting it to the intra prediction module. If Figure 3 is explained in the order of processing, the image input through the VIM is stored in the VIM Write buffer (Luma SRAM,Chroma SRAM) and then stored in the DDR2 memory in the raster scan method.

The memory block (SRAM in macroblock unit) that exists after the DDR2 memory in Figure 3 is controlled by the DDR2 controller. DDR controller reads macroblock pixels from SDRAM and stores them in ORG_LUMA_x, ORG_CHROMA_x buffers. Later, this macroblock unit information is used to be read by the encoder module. The size of the VIM write buffer (Luma SRAM) is the size that can store 3840 pixels and is the same size as CHROMA (chroma SRAM). The Org LUMA buffer consists of 4 RAMs that can store 256 pixels each, and the CHROMA buffer is designed with 4 RAMs that can store 128 pixels. The reason for using multiple buffers is to enable double buffering. Double buffering has the effect of reducing the processing speed by hiding the time to read and write data. Actually, only 2 SRAMs are needed, but 4 SRAMs were used by adding 2 SRAMs to be used for motion estimation later. The ddr2 controller (DMA) operates in two ways depending on the operation period.

2.1.1 Data transfer from VIM to DDR2

Data is transferred from VIM to DDR2 in a cycle as shown in Figure 4 below. Assuming that the operating frequency of DDR is 200 MHz and the operating frequency of the encoder is 100 MHz, one line of data is transmitted to DDR2 for 136 cycles.

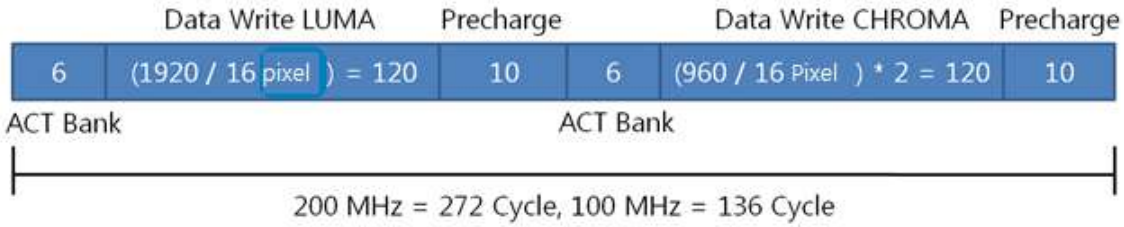


Figure 4. Data Transfer cycle from VIM to DDR2

2.1.2 Data transfer from DDR2 to Intra

In a cycle as shown in figure 5 below, data of 16x16 blocks are transferred from DDR2 to the Intra module. Assuming that the operating frequency of the DDR is 200 MHz and the operating frequency of the encoder is 100 MHz, the data of one macroblock is transmitted to the Intra module for 75 cycles.

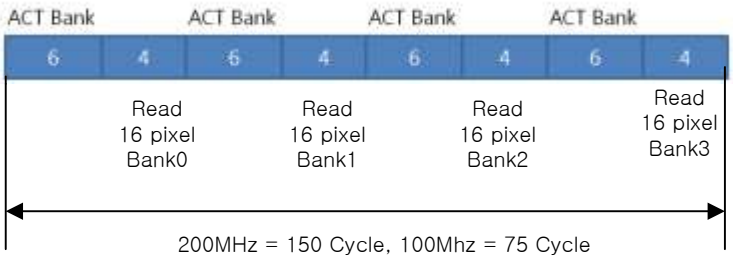
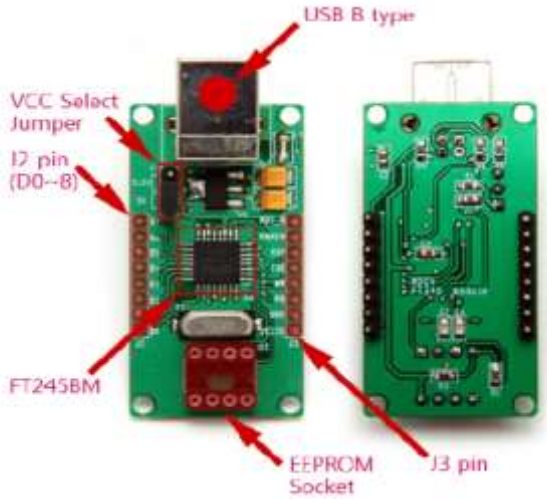


Figure. 5 Data Transfer cycle from DDR2 to Intra module

Figure. 6 Pin configuration of FT245

2.2. USB module design

The USB module functions to deliver the final output bitstream of the encoder to the PC through the external FT245 module. It receives 32-bit data as an input and transfers it to the FT245 module by 8 bits inside the USB module. Figure 6 shows the FT245 module to be connected to the USB module.



2.3 Others module design

In the case of other modules to be used in the encoder to be implemented with FPGA, Intra prediction & DCT module was modified and used by the one developed by the paper [10]. Also, in case of entropy coding, it was developed as verilog code. Total gate amount is 250,000 gate(2-input nand).

3. Experimental Results

The JM13.2 code was analyzed and the reference C code was coded based on it. To check whether the operation was correct after coding, the test vector was extracted from JM13.2 and compared with the reference code. The hardware was designed using verilog HDL, a hardware design language, and the function of the hardware was verified by comparing it with the test vector generated by this reference code. Figure 7 is the result of comparing the test vector output from the reference code using the Modelsim simulator, and shows that there are no errors in the output.

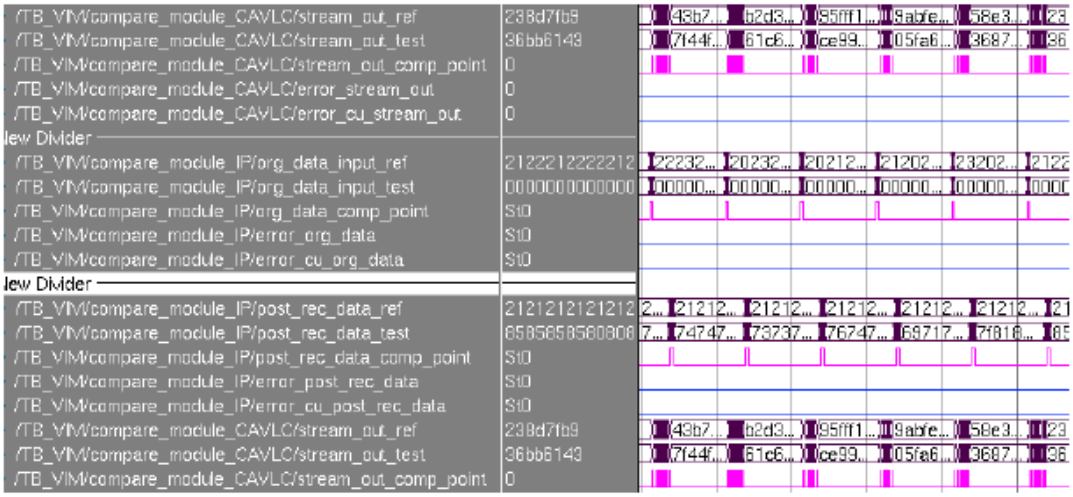


Figure 7. Simulation Results using Modelsim

After performing RTL simulation, it was implemented using Xilinx Virtex 5 XC5VLX330 devices, and Synopsys' Synplify Pro and Xilinx's Xilinx ISE Design Suite were used.

The designed encoder was designed to be verified using Huins' SOC platform. It was confirmed that the H.264 bitstream data output from the FPGA was saved as a file in a PC through the USB interface, and then decoded as an image when it was decoded as an image using an H.264 software decoder. Figure 8 shows the process of encoding a video by porting the HDL implemented in this study on the Huins SOC platform, and showing the decoded video on a PC.

Figure 8. FPGA implementation and software decoding

Since the LCD size of Huins SOC platform for FPGA verification is 480 × 272, the bitstream is encoded with this size in the experiment. The proposed hardware is capable of encoding Full HD (1088p) video at 108 MHz at 30 frames per second. The result of this paper is compared with the paper in [4], it is possible to process at a lower frequency of 32MHz compared to 140MHz when processing the same frame.

Table 1: The performance analysis of encoder

DDR clock	CIS clock	Encoder clock	cycles/ MB	Performance
200 MHz	25 MHz	50 MHz	440(FPGA verification)	480 × 727 30f/sec
200 MHz	50 MHz	60 MHz	440	720p 30f/sec
200 MHz	50 MHz	108 MHz	440	1088p 30f/sec
[4]		140 MHz	441	1088p 30f/sec

4. Conclusion

Implemented hardware was confirmed on the board that the FPGA Virtex5 XC5VLX330 operates at DDR2 DMA 200 MHz, Encoder 50 MHz, VIM 25 MHz, and 43099 LUT, which is about 20% of FPGA, was used. As a result of synthesis by Chartered 0.18um process, it was confirmed that it operates at 120 MHz. When the bitstream of the signal is transmitted to the PC through the USB interface and decoded in software, it is confirmed that it is decoded without a problem.

5. Acknowledgment

This research is based on the support of 2020 Woosong University Academic Research Funding.

6. References

1. ISO/IEC 14496-10:2004 FDAM 1, Information technology Coding of audio-visual objects-part 10:Advanced Video Coding, Amendment 1: AVC professional extensions, 2004.
2. Joint Video Team (JVT) software JM 13.2
3. Micron double data rate sdram, 512Mb_DDR_x4x8x16_D1.fm-512Mb DDR: Rev. L; Core DDR Rev. A 4/07 Available from: https://www.micron.com/products/dram/ddr-sdram
4. Lin.Y.K,Ku,C.W.,Li,D.W.,& Chang, T.S., A 140MHz 94K Gates HD1080P 30-Frmaes/s Intra-Only profile H.264 encoder,In IEEE Trans Circuits and System Technology, vol 19,Issue 3, March,2009
5. Z. Li, J. Li, Y. Zhao, C. Rong and J. Ma, "A SoC Design and Implementation of H.264 Video Encoding System Based on FPGA," 2014 Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics, 2014, pp. 321-324, doi: 10.1109/IHMSC.2014.179.
6. Noelle BGC,Maria Carmina Rae Villanueva Gonzaga;Hazel Deluso Laure Anastacia Ballesil Alvarez;Maria Theresa Gusad De Leon;Chris Vincent Jagunap Densing;John Richard Ereso Hizon;Marc Driz Rosales,Design of Multiple Prediction Complexity Configurations for an FPGA-Based H.264 Baseline Profile Encoder, TENCON 2018 - 2018 IEEE Region 10 Conference
7. M. Shafique et al., Application-Guided Power-Efficient Fault Tolerance for H.264 Context Adaptive Variable Length Coding, in IEEE Transactions on Computers, vol. 66, no. 4, pp. 560-574, 1 April 2017, doi: 10.1109/TC.2016.2616313.
8. D. Engelhardt, J. Moller, J. Hahlbeck and B. Stabernack, FPGA implementation of a full HD real-time HEVC main profile decoder, in IEEE Transactions on Consumer Electronics, vol. 60, no. 3, pp. 476-484, Aug. 2014, doi: 10.1109/TCE.2014.6937333.
9. Kibum suh, The Architecture of Intra-prediction

and DCTQ Hardware for H.264, Journal of the institute of Electronic and Information Engineer. May 2010. pp360~368 Available from: https://scienceon.kisti.re.kr/commons/util/originalView.do?cn=JAKO201029335105755&dbt=JAKO&koi=KISTI1.1003%2FJNL.JAKO201029335105755

10. Kibum Suh, Seongmo Park, Hanjin Cho. An Efficient Hardware Architecture of Intra Prediction and TQ/IQIT Module for H.264 Encoder, ETRI Journal 2005 May. 27(5):511-524 DOI:10.4218/etrij.05.0905.0032