Implementation of Speed Efficient Image Compression Unit for 640x480 pixel using VHDL

Mr. Shubhdeep Bhatt [1], Prof. Vinod Kapse [3]
Embedded Systems and VLSI Design [1,2]
Department of Electronics & Communication [1,2]
M.tech-IVth semester [1] , Professor[2]
Gyan Ganga Institute of Technology and Sciences [1,2]
Rajeev Gandhi Technical University, Jabalpur, Madhya Pradesh, India [1,2]

Abstract- **This research work is devoted to design high speed Image Compression Unit, which ultimately performs jpg to jpg/bmp to jpg to jpg compression operation. In which JPEG Encoder is the key element of digitized image compression unit, which is used in many places such as for multi-image disk storage, sending of many images through email etc. Speed is the most prominent factor of this JPEG Encoder being used recently. That's why it's imperative to design high speed Encoder, which can enhance the efficiency of those modules which lies upon the image compression operations performed by JPEG Encoder. To enhance speed of this unit we have used Ping-Pong buffer that gives much better result in comparison to other conventional buffer in terms of speed. Further we have analyzed hat 2-D DCT provides much better results in comparison to other transformations like Fourier Transform, Discrete Wavelet Transform with respect to speed. And found that the 2-D DCT with Ping-Pong buffer gives better outcome. Then we have proposed that design for the JPEG Encoder unit design. This proposed image compression Unit is able to perform image compression operation at high speed. And after designing each module individually, it has been integrated to design final module, in which reset and clock functionality has been provided to have better control on the circuitry. To design proposed JPEG Encoder unit Verilog hardware description language (HDL) has been used to design test bench, VHDL has been used to design modules and Matlab has been used to fetch the images from folder. For this design the target FPGA belongs to Virtex-5 (family), XC5vsx92t (device), ff1136 (package) with speed grade of -2. For synthesis purpose Xilinx synthesis tool (XST) of Xilinx ISE-13.2i has been used. For the behavioral simulation purpose ISE simulator has been used.**

Keywords – **Ping-Pong Buffer, 2-D DCT.**

I. INTRODUCTION

1.1 Definition of image compression:-

Meaning of image compression is to minimize the size in bytes of a graphic file without any degradation in quality of images to an unacceptable level. Reduction in size multiple image is required to store those within a limited disk space. This advantage supposes us sending or down loading image from server it helps to reduce our time taking for image purpose.

1.1.1 Types if image compression:-

There are mainly two types of image compression those are lossy and lossless image compression, whose description is given as follows:-

1. Lossless image compression: – it is a compression in which image compressed without introducing errors but only up to certain extent that's called lossless.

Types of Lossless image compression: -

a) Run-length encoding: - this encoding run data (consecutive data elements with identical value) and store in a single value data terms and count that.

b) DPCM and predective encoding: - It means differential pulse code modulation it's a lossless coding that decode the image and original image have same value at every correspondent element.

2. Lossy image compressions: - on lossy image compression the program removes some data from original(main) file and save the image with a reduced size it's depend upon developer that how much data to be developed by setting image compression rate.

Types of Lossless image compression: -

a) Transform encoding:-this encoding used in jpeg pictures. This transform the color image and with the help of coding this can convert the fewer color from the original file reduce with images block used DCT.

b) Chroma sub-sampling:- In sub-sampling a portion where image information stored is being stored known as sample. Sub-sampling can also be considered as a portion where some specific pixels collect the chrome information that is used as representative of all pixels. Sub-sampling is a part, or fraction, section where all chrominance sampling in an image.

II. IMPLEMENTATION OF PROPOSED IMAGE COMPRESSION UNIT

After thorough analysis we have seen that buffer is the most time-consuming part in the Image compression unit, to enhance timing efficiency with the use of Ping-Pong buffer. Following diagram shows the proposed image compression unit:-

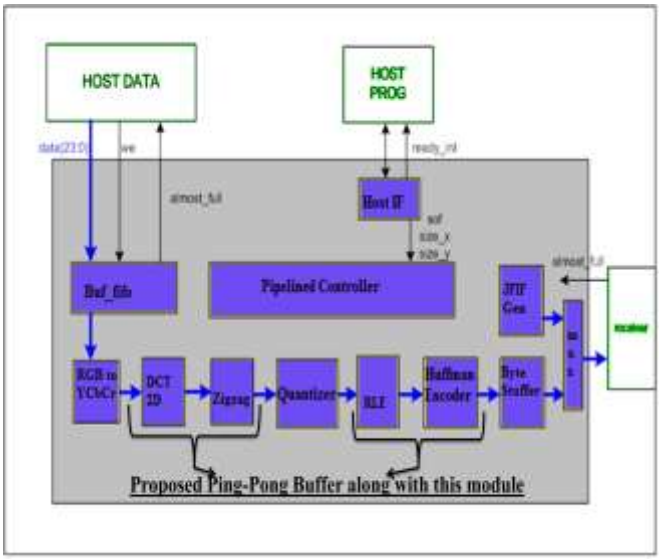

Figure 1 Proposed JPEG Encoder Unit

High level description about above block diagram has been explained in the following: -

1. HOST PROGRAMMER: - encoding chain started by host data programming. By host data interface data goes continuous in buffer FIFO until FIFO full signal received, after this it will be stop and wait for signal FIFO will become almost full. In this condition it should we write continue.
2. Pipelined Controller: - In encoding process "pipeline and controller ", every pipelined stage process 8*8 block of sample at a time.

3. Finally, encoded bit stream is byte stuffed and after that it should be stored to output (RAM or FIFO).

In the following we have described different blocks, belonging to the JPEG encoder: -

1. HOST INTERFACE: - Host interface is an interface between host (CPU) and jpeg encoder ip core.

2. BUFFER FIFO: - Host data interface writes input image into buffer FIFO and also this minimize latency between raw image being loaded and encoded for further usage.

This Buffer FIFO unit contains following blocks: -

1. RAM WRITE: - In this host write image line by line.
2. RAM will write (0 TO N-1) with 8-pixels in each line as well as per FIFO.
3. PIXEL_COUNT (15:0) is incremented after each write enable (WE) from host and incremented with every write enable (WE) from host and get reset at end of each line it gets recorded.
4. RAM READ: - 8*8 blocks are read in the form of sequential scanning from left to right and top to bottom.

3. RGB TO Y Cb Cr conversion: - Following equations are used for conversion from RGB to Y Cb Cr.

Y = (0.299*R) + (0.587*G) + (0.114*B).

Cb = (-0.1687*R) - (0.3313*G) + (0.5*B) + 128.

Cr = (0.5*R) - (0.4187*G) - (0.0813*B) + 128.

We have proposed to use 4:2:2 chroma sub-sampling. This will have better compression without any noticeable change in the image. Following chroma sub-sampling diagram is self-explanatory in itself: -

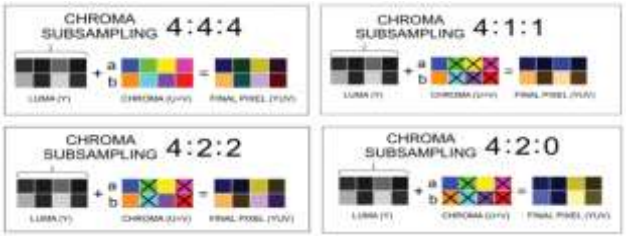

Figure 2 Chroma Sub-Sampling

4. DCT: - DCT is mathematical tool which converts signals from time domain to frequency domain. If we recall to the Fourier transform, that is conventionally used for conversion of signals from time domain to frequency domain. In Fourier transform we get sine and cosine waveform which is time spatial, where sine and cosine waves are replica of each other, which is un-necessary time-consuming, So the best way is to use only one value for this cosine value is used in DCT. There is another method known as C karhunen loeve transform, that is best possible method but it is complex in nature. We have proposed DCT core in our design, as this does not use multiplier but ROM based look up table which makes it super efficient. When the pipe line is full, 8x8 blocks are processed in 64 cycles! Quantizer and Zigzag also take 64 cycles when the pipeline is full. We have proposed DCT core in our design, as this does not use multiplier but ROM based look up table which makes it super efficient. When the pipe line is full, 8x8 blocks are processed in 64 cycles! Quantizer and Zigzag also take 64 cycles when the pipeline is full.

Host read image data in form of 8 bits groups line by line and places that data in buffer fifo, Which is then it breaks this 8*8 blocks like eg. Chess and every block gets converted according to their comparative frequency by DCT whose amplitude get placed as per pre-specified quantization level, which is then converted to the binary values. We have proposed to use ping pong buffers with this module. ping pong is used for data processing in parallel manner, means in which time value is providing in ping pong for read on particular time, value also writing meanwhile ping pong parallel works. Ping pong used in memory optimization system design in altera and we used buffer ping pong for parallel which provide read and writing data at a time.

5. SUB RAM: - This is memory used as line buffer.

6. Zigzag: - This unit is responsible to perform zigzag scan. It is simply reorder of samples positions in one 8x8 block and accordingly we have taken quantization coefficient value and value of DCT coefficient are converted in binary format in form of 0 and 1, and with series of 0's are arranged in ascending process in last. We have proposed to use ping pong buffers in this module. Ping pong is used for data processing in parallel manner, means in which time value is providing in ping pong for read on particular time, value also writing meanwhile ping pong parallel works. Ping pong used in memory optimization system design in altera and we used buffer ping pong for parallel which provide read and writing data at a time.

7. RLE: - We have given an example to explain this unit. For example suppose 10 (zeros) we are getting so if we write this it take space if here we apply run length encoding which help to write these 10(zeros) as 10*0 form and reorder of those value is done using zig-zag unit, which contain 0 values for high frequency signal and the value which is less than zero it is low frequency signal. We have proposed to use ping pong buffers in this module. ping pong is used for data processing in parallel manner, means in which time value is providing in ping pong for read on particular time, value also writing meanwhile ping pong parallel works. Ping pong used in memory optimization system design in altera and we used buffer ping pong for parallel which provide read and writing data at a time.

In short zigzag core perform reordered input sample accordingly with zigzag sequence. ZigZag Core is performing reordering of input samples according to zig-zag sequence. For this purpose REORDER ROM is used.

FIFO CTRL: - FIFO CTRL is read controller for FIFO. It read FIFO output until it's not empty. Also it generates data valid signal which simply delayed FIFO read requirements.

8.QUANTIZER: - It perform input sample by defined quantization work sample used with fill the value 64 different quantization coefficient to internal ram(64*8 bits).

Following equation is used:-

FOUT (u, v) = round (FINPUT (u, v) / Q (u, v)).

- u, v –rectangular coordinates
- FINPUT – input sample to quantizer

FOUTPUT – output sample from quantizer.

Two signals AC and DC obtained as outcome. AC-signal which has frequency signal's high in terms of sine/cosine and this value are placed in last and dc-signal which has fix amplitude and it's taken first. Compression ratio depends upon level of quantization and the particular value of quantization coefficient matrix and zig-zag approach.

9. Huffman Encoder:- it perform Huffman encoding operation that is used for conversion of parallel data stream into serial data stream. We have proposed to use ping pong buffers in this module.

Ping pong is used for data processing in parallel manner, means in which time value is providing in ping pong for read on particular time, value also writing meanwhile ping pong parallel works. Ping pong used in memory optimization system design in altera and we used buffer ping pong for parallel which provide read and writing data at a time.

9. Byte Stuffer: - contains those values which are left on sample of input and make jpeg.hex with the help of jfif generator.

10. JFIF Header Generated:-

Default JFIF header is generated from on-chip RAM. Out of which most of header fields are hardcoded, however some as configurable and must be programmed by Host.

11. JFIF Generator:-

Following is the algorithm of JFIF Generator.

When start_jfif asserts and eoi = 0:

- rd_cnt is reset. rd_cnt then counts from 0..622 (JFIF header size=623 bytes). This reads Header RAM whose output is then saved to Output RAM.
- After all bytes from to be Header RAM are transmitted to ready_jfif.

When start_jfif asserts and eoi = 1:

- EOI marker "FF D9" is written after coded byte stream to "close" image.

EOI Writer (End of Image Writer):-

Writes two Header RAM: - it's Size is 2048x8 bits that has Dual port RAM. Most of the content taken from HEX initialization file. Some fields are configured by Host IF.

III. CONFIGURATION REQUIRED TO SUPPORT OUR PROPOSED DESIGN

- WINDOWS EDITION: - WINDOWS-7, WINDOWS-XP.
- SYSTEM TYPE: - 64-BIT OPERATING SYSTEM.
- RAM REQUIRED: - 4 GB.

The basic platform of here we designing To design and implement high speed area optimized JPEG image compression (encoder) using VHDL and synthesized for FPGA.

- ✓ First convert bitmap image into text (test.txt) using matlab img2txt utility given in the folder.
- ✓ To convert bitmap image into text, place any *.bmp file in the same directory where img2txt utility is going to run and rename it to test.bmp
- ✓ Run the utility and copy test.txt into the directory where all VHDL files are placed.

- ✓ Now run the simulation for at least 4.342ms.
- ✓ Compressed image is generated at a place where all VHDL code stored.

When we run 7.9ms then we get the compressed image 43kb. On the basis of which we get following outcomes of RTL Schematic and other stuff.

IV. TIMING CONSTRAINTS

1. Minimum period: 5.531ns.

2. Minimum input arrival time before clock: 5.130ns.

3. Maximum output required time after clock: 2.86ns.

4. Maximum combinational path delay: No Path Found.

V. COMPARATIVE RESULTS

References	DEVICE	[1]	Proposed
NUMBER OF SLICE REGISTER	XILINX VIRTEX-5	9360	4003
NO OF DSP BLOCKS	XILINX VIRTEX-5	40	10
MEMORY USED	XILINX VIRTEX-5	1728 KB	1044 KB
NO. OF ALUTS	XILINX VIRTEX-5	33150	4384

VI. APPLICATION

Our proposed image compression unit does not used only for getting high level of efficiency to store any disk image with less memory space, but also can be used in the following areas also :-

- To transmit compressed image over the low speed network, So that images can be send at high speed without any concerns due to limit of low speed network.
- Our proposed unit can also be used in real time Image and video processing.
- Further it can also be used in medical, research and defense field it's highly required to process heavily on images at Real time basis, which can be efficiently processed with the use of our proposed unit.

VII. REFERENCES

[1] MRS. KALEEM FATIMA S. VIJAY GOPAL N. ZEESHAN NADEEM, "A NOVEL ARCHITECTURE FOR THE COMPUTATION OF 2D-DWT AND ITS IMPLEMENTATION ON VIRTEX-II PRO FPGA".

[2] KALEEM FATIMA, VIJAY GOPAL SARVEPALLI, ZEESHAN NADEEM NAKHI, *"A NOVEL ARCHITECTURE FOR THE COMPUTATION OF THE 2D-DWT"*, *CIS*, 2007, 2013 NINTH INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND SECURITY, 2013 NINTH INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND SECURITY 2007, PP. 531-535, DOI:10.1109/CIS.2007.172.

[3] DAMIAN MODRZYK, AND MICHAŁ STAWORKO, *"A HIGH-PERFORMANCE ARCHITECTURE OF JPEG2000 ENCODER"*, 9TH EUROPEAN SIGNAL PROCESSING CONFERENCE (EUSIPCO 2011), BARCELONA, SPAIN, AUGUST 29 - SEPTEMBER 2, 2011.

[4] TRANG T.T. DO, BINH P. NGUYEN, "A HIGH-ACCURACY AND HIGH-SPEED 2-D 8x8 DISCRETE COSINE TRANSFORM DESIGN".

[5] R. RAMPRAKASH, K. DEEPTHI, *"DESIGN AND IMPLEMENTATION OF JPEG IMAGE COMPRESSION AND DECOMPRESSION"*, INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING AND TECHNOLOGY (IJIET), VOL. 2 ISSUE 1 FEBRUARY 2013, ISSN: 2319 – 1058.

[6] DURGA PATIDAR, PROF. JAIKARAN SINGH, PROF. MUKESH TIWARI, *"A VHDL IMPLEMENTATION OF JPEG ENCODER FOR IMAGE COMPRESSION"*, INTERNATIONAL JOURNAL OF DIGITAL APPLICATION & CONTEMPORARY RESEARCH WEBSITE: WWW.IJDACR.COM (VOLUME 1, ISSUE 2, SEPTEMBER 2012)

[7] PRASHANT CHATURVEDI, PROF. TARUN VERMA AND PROF. M. ZAHID ALAM, *"A NOVEL VLSI BASED ARCHITECTURE FOR COMPUTATION OF 2D-DCT IMAGE COMPRESSION"*, 1(2): 65-70(2012). INTERNATIONAL JOURNAL OF ELECTRICAL, ELECTRONICS ISSN NO. (ONLINE) : 2277-2626 AND COMPUTER ENGINEERING.

[8]HTTP://WWW.IMAGE-ENGINEERING.DE/INDEX.PHP?OPTION=COM_CONTENT&VIEW=ARTICLE&ID=522
(How Does JPEG Work?)

[9] Prashant Chaturvedi, Prof. Tarun Verma and Prof. M. Zahid Alam, *"A Novel VLSI based Architecture for Computation of 2D-DCT Image Compression"*, International Journal of Electrical, Electronics and Computer Engineering 1(2): 65-70 (2012), ISSN No. (Online): 2277-2626.

[10] T.Pradeepthi and Addanki Purna Ramesh, *"PIPELINED ARCHITECTURE OF 2D-DCT, QUANTIZATION AND ZIGZAG PROCESS FOR JPEG IMAGE COMPRESSION USING VHDL"*, International Journal of VLSI design & Communication Systems (VLSICS) Vol.2, No.3, September 2011.

[11] S.Vijayaraghavan, Dr.C.Parthasarathy, "*Fast DCT Based image compression Using FPGA*", international journal of innovative research in electrical, electronics, instrumentation and control engineering, Vol. 2, Issue 1, January 2014.

[12] Prashant Chaturvedi, Tarun Verma, Rita Jain, *"FPGA Implementation of 2-D DCT Architecture for JPEG Image Compression"*, international journal of advanced electronics & communication systems Approved by CSIR-NISCAIR ISSN NO: 2277-7318.

[13] *"Ping-Pong Buffer"*, By Prawat Nagvajara, Electrical and Computer Engineering, Martin, 2013.

