FPGA Based Image Splitting of Video Streaming Data for High Speed Data Transmission

Mr. UNDI SIDDESH [1] **Mr. PRASHANTHA H A** [2]

[1]***Lecturer,*** Dept. of Electronics & Communication Engineering, Government Polytechnic, Kudligi - 583135
[2]***Lecturer,*** Dept. of Computer Science and Engineering, Government Polytechnic, Devadurga - 584111

Abstract**—**The aim of the project is to split the image or video into blocks and transferring video data over the network. Video Splitting is the process of dividing the video into non overlapping parts. Then row mean and column mean or each part is obtained. After applying transform on these, feature sets can be obtained to be used in image retrieval. By using splitting higher precision and recall can be obtained. Video Streaming refers to transferring video data such that it can be processed as a steady and continuous stream over the network. With streaming the client browser or plug in can start displaying the multimedia data before the entire file has been transmitted. Also comparative study has been made for device utilization between JTAG simulation and implementation using the simulink.

Index Terms**—**Image, Video, Splitting, Simulink.

INTRODUCTION

Video Splitting is the process of dividing the video into non overlapping parts. Then row mean and column mean or each part is obtained. After applying transform on these, feature sets can be obtained to be used in image retrieval. By using splitting higher precision and recall can be obtained. Video Streaming refers to transferring video data such that it can be processed as a steady and continuous stream over the network. With streaming the client browser or plug in can start displaying the multimedia data before the entire file has been transmitted. An image is defined as a two dimensional function, f(x, y), where x and y are spatial coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y and the intensity values of f are all finite, discrete quantities, we call the image a digital image [1]. Digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are called picture elements, image elements, pels, and pixels. Pixel is a term used most widely to denote the elements of a digital image. Pixels are normally arranged in a two dimensional grid and are often represented using dots or squares. Number of pixels in an image can be called as resolution. Associated with each pixel is a number known as Digital Number or Brightness Value that depicts the average radiance of a relatively small area within a scene. The matrix structure of the digital image is shown in Fig 1.

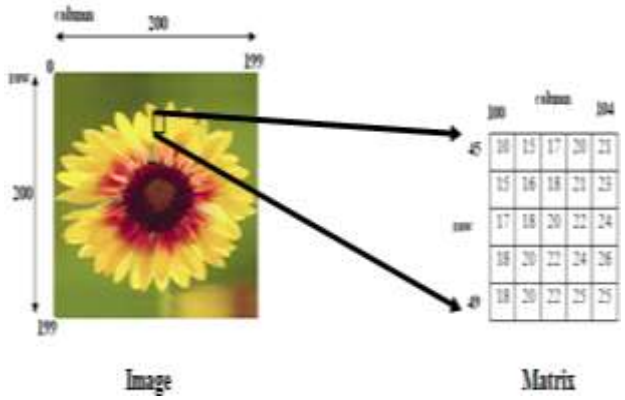

Fig. 1 Structure of digital image

The term video commonly refers to several storage formats for moving pictures. Video can be recorded and transmitted in various physical media: in magnetic tape when recorded as PAL or NTSC electric signals by video cameras, or in MPEG-4 or DV digital media when recorded by digital cameras. Quality of video essentially depends on the capturing method and storage used. Digital television is a relatively recent format with higher quality than earlier television formats and has become a standard for television video. 3D-video, digital video in three dimensions, premiered at the end of 20th century. Six or eight cameras with real time depth measurement are typically used to capture 3D-video streams. The format of 3D-video is fixed in MPEG-4 Part 16 Animation Framework extension (AFX). Video streaming services are services in which continuous video and audio data is delivered to an end user. Video streaming enables users to view videos within system dependent play out delay after the end user begins receiving the data. This is in contrast to other schemes that require the user to wait for the entire video to download before it can be viewed. Since it can often take several minutes and longer to download whole video files, video streaming offers the advantage of being able to view the video soon after it begins downloading. This service option is generally used in multimedia information and message

retrieval, video-on-demand, Pay-TV, interactive news retrieval and search, and other multimedia information broadcasting. Longer playout delay is allowed for Video Streaming Services because of its one-way transmission nature and buffering at the terminal. Consequently, transmission errors can be dealt with by utilizing retransmission. Terminal buffering is used to minimize the influence of delay jitter and retransmission delays, and to achieve seamless playback. This service is initially expected to be a point-to-point service, and use of other techniques, such as multicast, is for further study. The structure of the video streaming is show in Fig. 2. The streaming services are asymmetric between the sender and the receiver, since the video stream only flows in one direction, from a server to one or more clients. On the sender side, the video streaming services will include content creation and transmission.The video stream is encoded from the video and audio signals with corresponding codecs if the content is encoded in real time, or if it is retrieved in precoded format from storage devices. The encoded information may be sent in packet or circuit mode. The receiving terminal decodes the data with the corresponding video and audio decoders. The output is then sent, to be played back at local video and audio devices. The video streaming services will also include system control protocols for setting up connections between parties involved with the streaming services, negotiating various options and capabilities, and communicating with and controlling the various source codecs that the video streaming services use.

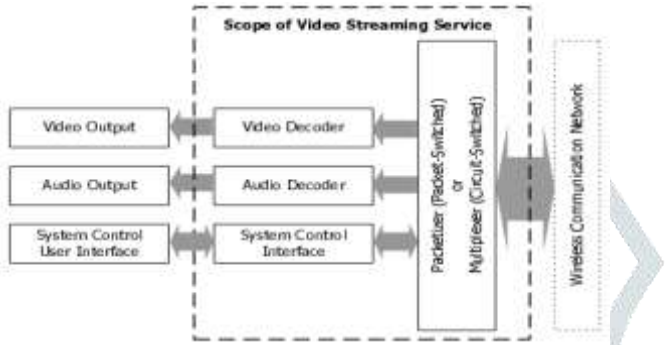

Fig. 2 Scope and structure of video streaming services

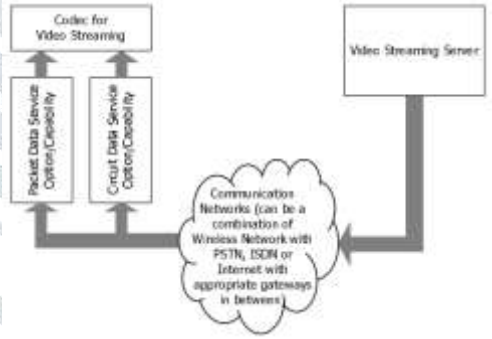

Fig.3 Relationship between video streaming services and other data services

The Video Streaming Services can be built over other data service options, e.g., Packet Data Service Option and Circuit Data Service Option. The relationship between video streaming services and other data services is shown in Fig.3. Although it is possible that the Video Streaming Services could utilize the "Multimedia Call Model" (i.e., calls with multiple simultaneous connections of the same or different traffic types including voice, circuit switched data, and packet data), it is recommended that the practice of the common international standards to transmit all the contents through one single connection to be followed. This practice ensures compatibility with other systems, and avoids the extra requirements of synchronizing the data sent through separate connections. When the Packet Data Service Option is used, the Video Streaming Service shall packetize all of the data streams before the data is transferred to the Packet Data Service. When the Circuit Data Service Option is used, the Video Streaming Services needs to multiplex all the data streams before the data is transferred to the Circuit Data Service.

Proposed Block Diagram Of Image/Video Splitting The block diagram of the design is as shown in Fig. 4. Video from the web camera at 30 frames per second is applied to the video input block. Resize block enlarge or shrinks image size. Captured video will be in RGB format. It is converted into croma and luma components. Luma represents the brightness in an image and it represents the achromatic image without any color while the croma component represents the color information. Image split block splits the image into number of blocks. Each spitted block is resized using bicubic interpolation technique, as the bicubic interpolation does no suffer the step line boundary problem of nearest neighborhood interpolation and copes with linear interpolation blurring as well. Bicubic interpolation is used in raster displays that enable zooming with respect to arbitrary point. If nearest neighbor methods are used, areas of the same brightness would increase. Bicubic interpolation preserves fine details in as image as well. The split image is displayed using the video display output block.

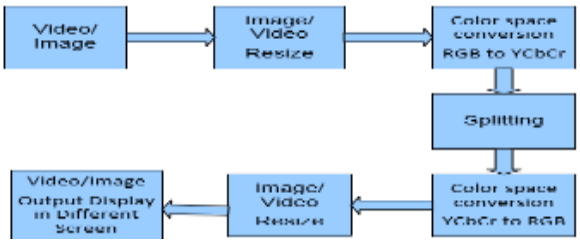

Fig 4. Block diagram of image/video splitting

throughout this document and are identified in italic type, within parentheses, following the example. PLEASE DO NOT RE-ADJUST THESE MARGINS. Some components, such as multi-leveled equations, graphics, and tables are not prescribed, although the various table text styles are provided. The formatter will need to create these components, incorporating the applicable criteria that follow.

DESIGN AND IMPLEMENTATION

SOFTWARE REFERENCE MODEL: The software reference model for splitting the image into four parts is as show in Fig. 5. The original image in RGB format. Resize block resizes the image to the specified value say 256 x 256. Color conversion block converts the input RGB format to YCBCR format. Y stands for intensity component (luma). Cb and Cr stands for blue difference and red difference (croma) components respectively. These croma and luma components is applied to the splitting block. Splitting block is a subsystem which is having one input and four outputs. This splitting block consists of submatrix block. Submatric is used to select the image matrix. Each Splitting block contains four outputs of dimension 128 x 128. One output from each splitting block is concatenated.ie first output from splitting block, splitting1 block and splitting2 block respectively. When we do this we will get one entire image. Similarly this procedure is carried out for all the outputs of splitting block. This image is in YCbCr format. The image is converted back to RGB format using color conversion block. Splitted bocks are resized to the original dimension of the input image. For resizing we use bicubic interpolation technique. Resized images are displayed using video display blocks.

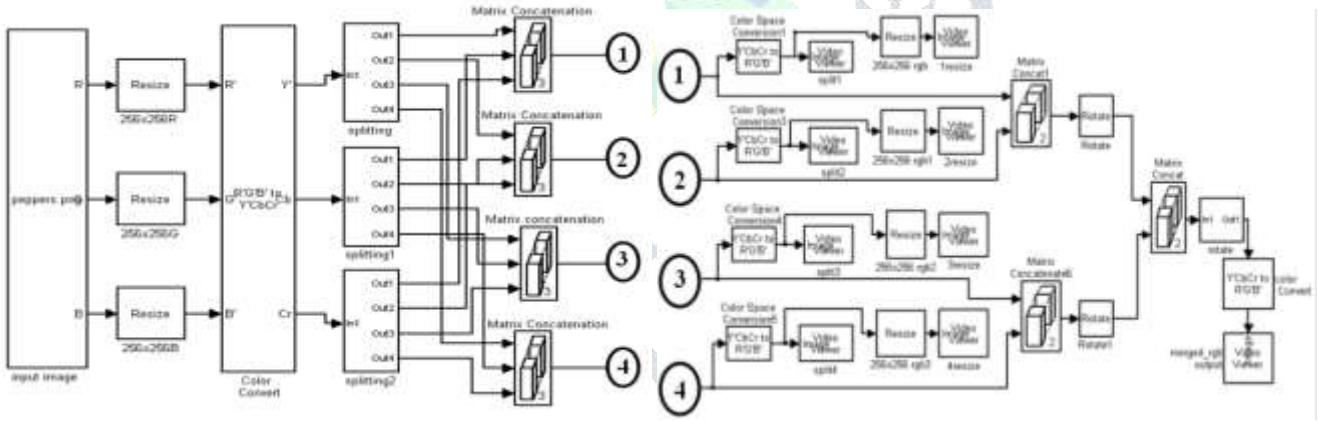

Fig. 5 Software Reference Model

HARDWARE MODEL FOR SPLITTING COLOR IMAGE: The hardware model of image splitting color image is as shown in the Fig. 6. The input image is in RGB format. That image is enlarge or shrink using the resize block. the resized output is in RGB format is converted in YCbCr component. The output of the color conversion block is converted into serial format using the serial conversion blocks m2s, m2s1, m2s2 as shown in model. The output of conversion blocks are stored in separate memory. Next comes the splitting block which is used to split the image pixel values. These splitted image pixel values are stored in separate memory as shown. The pixel values read from memory are converted into image format using the subsystem r16 to r23 in the Fig.6. The image interpolated is carried out in subsystems r16 to r23. Each color component output is concatenated to obtain an image. This image is in YCbCr format is converted into RGB format. The output image is displayed on the screen.

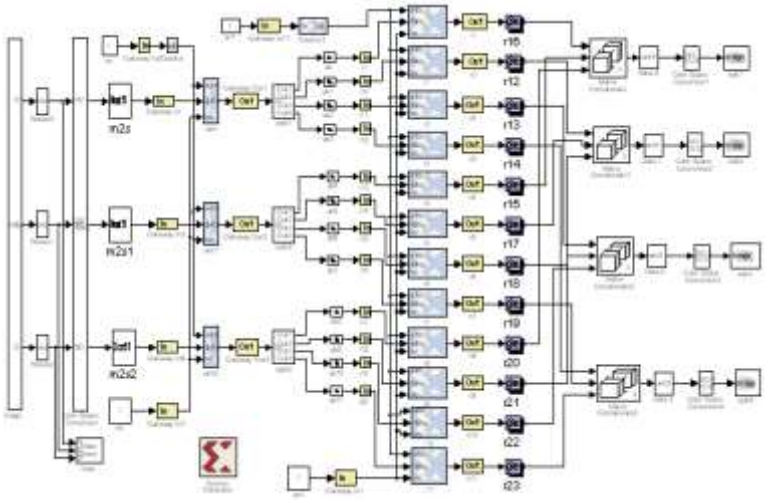

Fig. 6 Hardware model of color image splitting.

FPGA IMPLEMENTATION

In market, one of the most advanced FPGA family is the FPGA series produced by Xilinx. The vertex user programmable gate array comprises two major configurable elements: configurable logic blocks (CLBs) and input/output blocks (IOBs). Each CLB is composed of two slices. A slice contain 4-input, 1-output LUTs and two registers. Interconnections between these elements are configured by multiplexers controlled by SRAM cells programmed by a user's bit stream. The LUTs allow any function of five inputs, and two functions of four inputs, or some functions of up to nine inputs to be created within a CLB slice. This structure allows a very powerful method of implementing arbitrary, complex digital logic. Virtex FPGAs are programmed using verilog HDL: a popular hardware description language. The language has capabilities to describe the behavioral nature of a design, the data flow of a design, a design's structural composition, delays and a waveform generation programming and development environment, Xilinx ISE foundation series tools have been used to produce a physical implementation for the virtex FPGA. Fixed programmable gate arrays provide a new implementation platform for the discrete wavelet transform. Here we use Xilinx 10.1 version and Vertex 2pro development board.

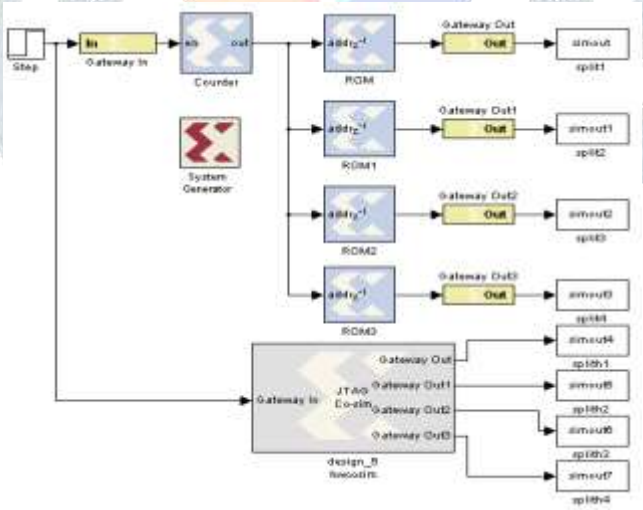

Fig. 7 JTAG implementation of gray image splitting

RESULTS

SIMULATION RESULTS : Simulation results show counter output, and values read from ROM of the hardware model show in Fig 7.1 ROM output consists of coeficients of the splitted image.

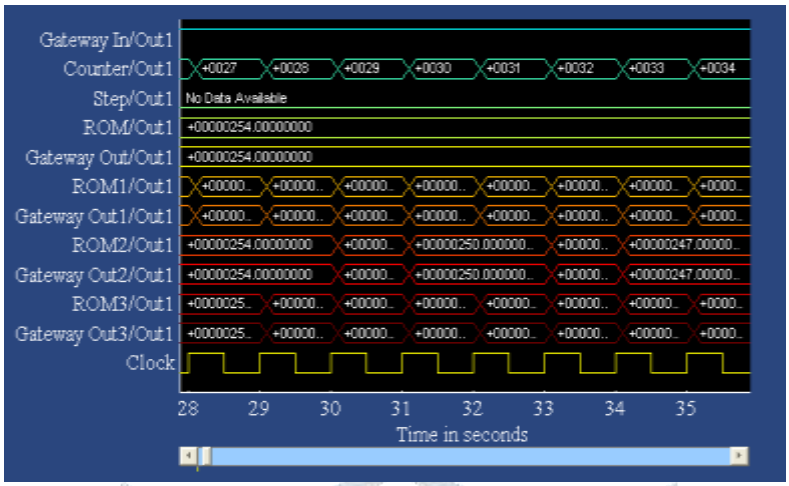

Fig. 8 Simulation waveform of Fig. 7

SYNTHESIS RESULTS : Device utilization and the timing summary for two designs shown in Fig. 6 and Fig. 7 are compared in tables 1, 2, 3 and 4 respectively

Selected device: 2vp30ff896-7

No of slices	20 out of 13696	<1%
No of slice flip flops	9 out of 27392	<1%
No of 4 input LUTs	35 out of 27392	<1%
No of IOs	1	
No of bonded IOBs	1 out of 556	<1%
No of GCLKS	3 out of 16	18%

Table 1 Device utilization summary for Fig. 7

Table 2 Timing report for Fig. 7

Maximum period	1.203ns
Maximum frequency	831.601MHz
Minimum input arrival time before clock	3.347ns
Maximum output required time after clock	0.701ns
Maximum combinational path delay	0.780ns

Table 3 Device utilization summary for Fig. 6

No of slices	127 out of 13696	<1%
No of slice flip flops	208 out of 27392	<1%
No of 4 input LUTs	220 out of 27392	<1%
No of IOs	1	
No of bonded IOBs	1 out of 556	<1%
No of GCLKS	3 out of 16	18%

Table 4 Timing report for Fig. 6

Maximum period	1.203ns
Maximum frequency	831.601MHz
Minimum input arrival time before clock	4.950ns
Maximum output required time after clock	0.701ns
Maximum combinational path delay	0.921ns

CONCLUSION

From the above synthesis report comparisions we come to conclusion that model shown in Fig. 7 is best in terms of device utilization compared to the another shown in Fig. 6. From the results discussed from the previous section we can conclude that using JTAG co-simulation we can easily implement the design on the FPGA kit(VERTEX-2 PRO FPGA KIT). The analysis showed that input image/video is splitted according to the design. The simulink input image splitted efficiently. To prove this statement, when splitted blocks are merged together we get back the original image. Similar results are seen in system generator and on hardware. Comparing the output obtained by software model and hardware model we can conclude the desired job is achieved.

REFERENCES

1. Rafael C. Gonzalez and Richard E. Woods, "*Digital Image Processing"*, Pearson Prentice Hall, 3 Edition, 2008.
2. Ian kuon and Jonathan Rose, *"Measuring the gap between FPGAs and ASICS"*, IEEE transcation on Computer-Aided of Integrated Circuits and systems, 26(2): 203-215,2007
3. Xilinx incorporated , virtex-6 Family overview, 2009.
4. Marco Aurelio,Nunu maganda, "*Real time FPGA based architecture for bicubic interpolation: an application for digital image scaling",* Proc of the International conference on reconfigurable computing and FPGAs, page 5-14, 2005.
5. Alain Merigot, "*Revisiting image splitting"*, Proc of 12th international conference on image analysis and processing, page 314-319, 2003
6. Gregory H. Rose, Raleigh F.Johmon and Don G. Brunder, "*Interpolation algorithm for 3D Reconstruction of magnetic resonance images",* Proc of 12th Southern Biomedical Engineering Conference, page 43-45, 1993.
7. Belgacem Ben Yousef and Jim Bizzochi, "*Enhance Pixel-Based Video Frame Interpolation Algorithms",* IEEE International symposium on Signal Processing and Information Technology, page 23-28, Dec 2007.
8. T.Saidani, D. Dia, W. Elhamzi, M. Atri, and R. Tourki, "*Hardware Cosimulation for Video Processing using Xininx System Generator"*, Proc of the World Congress on Engineering, Vol 1, page 1-3, 2009.
9. Matthew Own by, Dr Wagdy.H. mhmoud, "*A design methodology for implementing DSP with xilinx system generator for Matlab"*, processdings of 35th south eastern symposium, Vol 15, page 2226-2238, 2006.
10. Fatma T. Arslan and Artyom M. Grigoryan, *"Method of image enhancement by splitting signal"*, IEEE international Conference on Acoustics, speech and signal processing, vol 4, page iv/177 - iv/180, March 2005.
11. Akhtar. P and Azhar. F, *"A single image interpolation scheme for enchanced super resolution in Bio-Medical Imaging"*, 4th International Conference on Bio-informatics and Bio medical Engineering, pages 1-5, June 2010.
12. Hajizadeh.M, Helfroush. M.S, Tashk.A, *"Improvement of Image zooming using least directional differences based on linear and cubic interpolation"*, 2nd International conference on computer computing and communication, pages 1-6,2009.
13.Patil.V, Kumar.R and Mukherjee.J, *"A Fast Arbitrary Factor Video Resizing Algorithm"*, IEEE transactions on circuits and systems for video technology, Vol 16, page 1164-1171, Sep 2006.
14. Ince I.F, Ilker Yengin, Salman. Y.B, Hwan-Gue Cho and Tae-Cheon Yang, *"Design CAPTCA Algorithm: Splitting and Rotating the image against OCRs"*, 3rd international conference on Convergence and Hybrid Information Technology ,Vol 2, page 596-601,2008.
15. Lei Zhang and Xiaolin Wu, *"An Edge Guided Image interpolation algorithm via directional filtering and data fusion"*, IEEE transactions on Image Processing, Vol 15, page 2226-2238, Aug.2006.
16. Matlab user guide, help navigator.2007.
17. Tien-Ying kuo and Lin-ying chuong, *"Fast global motion-compensated frame interpolator for very low-bit rate quality enhancement"*, IEEE International conference on Acoustics, Speech, and Signal processing, vol 3, page 111-113- April 2003.
18. Samir Palnitkar, *"Verilog HDL a Guide to Digital Design and Synthesis"*, Pearson Education, 2 Edition, 2009.