CAM Based High-Speed Compressed Data Communication System Development using FPGA

Tribeni Prasad Banerjee[1], Amit Konar[2], Ajith Abraham[3].

[1]*Central Mechanical Engineering Research Institute,Durgapur.Durgapur-713209, West Bengal,India*
[2] *Department of Electronics & Tele communication Engineering,,Jadavpur University, Kolkata-32*
[3]*Norwegian University of Science and Technology, Norway*
E-mails: t_p_banerjee@yahoo.com, konaramit@yahoo.co.in, ajith.abraham@ieee.org

Abstract

In this paper we propose a Lossless data compressor in high level throughput using re programable FPGA technology.Real time data compression is expected to play a crucial role in high rate data communication applications. Most available approaches have largely overlooked the impact of mixed pixels and subpixel targets, which can be accurately modeled and uncovered by resorting to the wealth of spectral information provided by hyperspectral image data. In this paper, we proposed an FPGA-based data compressor on the concept of CAM and Dictionary based compression technique has been proposed in this paper . It has been implemented on a Xilinx Spartan3 -II FPGA formed by several millions of gates, and with high computational power and compact size, which make this reconfigurable device very appealing for onboard, real-time data processing.

Keyword: **Data-Compression, CAM, Image and Video Compression, Digital Signal Processing , FPGA , ASIC , SoC .**

I. Introduction

The use of lossless data compressor can bring about a number of increasingly important benefits to an electronic system. With the audio and video compression systems (such as JPEG and MPEG) which are lossy and hence only recreate an approximation of the original data. The most obvious benefit of data compression is reduction in the volume of data which must be stored .This is important where the storage media itself is costly (such as memory) or the other parameters, such as power consumption, weight or physical volume, are critical to product feasibility. Using the data compression reduce the total storage requirement, thus effecting the cost saving. The push to roll out high definition video enabled video and imaging equipment is creating numerous challenges for video system architects. The increased image resolution brings with it higher performance requirements for basic video data path processing and next-generation compression standards, outstripping that which standalone digital signal processors (DSPs) can provide. In addition, the system specifications require designers to support a range of standard and custom video interfaces and peripherals usually not supported by off-the-shelf DSPs. While it is possible to go the route of application specific integrated circuits (ASICs) or use application specific standard products (ASSPs), these can be difficult and expensive alternatives that might require a compromised feature set. Furthermore, these choices can hasten a short product life cycle and force yet another system redesign to meet varied and quickly changing market requirements. Field programmable gate arrays (FPGAs) are an option that can bridge the flexibility gap in these types of designs. Additionally, with the increasing number of embedded hard multipliers and high memory bandwidth, the latest generation of FPGAs can enable customized designs for video systems while offering a manifold performance improvement over the fastest available stand-alone DSPs. Designers now have the ability with state-of-the-art FPGA co-processor design flows to implement high-performance DSP video and image processing applications. This new generation of tools facilitates the design of a system architecture that is more scalable and powerful than traditional DSP-only designs while at the same time taking advantage of the price and performance benefits of FPGAs. We have been researching high performance loss less data compressor /decompression hardware as the means to achieve the high throughput target [1]-[5].

978-1-4244-5612-3/09/$26.00 ©2009 IEEE

II. Video Compression Application Requirements

A wide variety of digital video applications currently exist. They range from simple low-resolution and low-bandwidth applications (multimedia, Picture phone) to very high-resolution and high-bandwidth (HDTV) demands. The present requirements of current and future digital video applications and the demands they place on the video compression system. The importance of video compression, the transmission of digital video television signals is presented. The bandwidth required by a digital television signal is approximately one-half the number of picture elements (pixels) displayed per second. The analog pixel size in the vertical dimension is the distance between scanning lines, and the horizontal dimension is the distance the scanning spot moves during ½ cycle of the highest video signal transmission frequency. The bandwidth is given by Equation

Bw = (Cycle /Frame) (FR)
= (Cycle/Line)(NL)(FR)
= (0.5) (aspect ratio)(NL)(FR)(RH)/0.84
=0.8 (FR)(RH)(ML)

Where

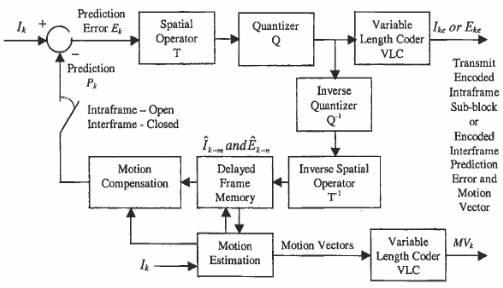
Bw = system bandwidth.
FR = number of frames transmitted per second (fps),
N_L = number of scanning lines per frame,
R_H = horizontal resolution (lines), proportional to pixel resolution.

Video compression and video CODECs will therefore remain a vital part of the emerging multimedia industry for the foreseeable future, allowing designers to make the most efficient use of available transmission or storage capacity. In this paper we introduce the basic components of an image or video compression system. We begin by defining the concept of an image or video encoder (compressor) and decoder (decompress or). We then describe the main functional blocks of an image encoder/decoder (CODEC) and a video CODEC. We are emphasizes on the real implementation of the CODEC on the VLSI system into a FPGA (Xilinx Spartan 3 kit).

III. Basic Video Compression Scheme

Video or visual communications require significant amounts of information transmission. Video compression, as considered here, involves the bit rate reduction of digital video signal carrying visual information. Traditional video-based compression, like other information compression techniques, focuses on eliminating the redundant elements of the signal. The degree to which the encoder reduces the bit rate is called its coding efficiency, equivalently; its inverse is termed the compression ratio:

Coding efficiency
= (compression ratio)-1
= encoded bit rate/decoded bit rate.



I_k
+
Prediction Error E_k
−
Prediction P_k
Intraframe – Open
Interframe - Closed
Spatial Operator T
Quantizer Q
Variable Length Coder VLC
I_{ke} or E_{ke}
Transmit Encoded Intraframe Sub-block or Encoded Interframe Prediction Error and Motion Vector
Inverse Quantizer Q^{-1}
$\hat{I}_{k-m}$ and $\hat{E}_{k-n}$
Motion Compensation
Delayed Frame Memory
Inverse Spatial Operator T^{-1}
I_k
Motion Estimation
Motion Vectors
Variable Length Coder VLC
MV_k

Figure 1. A General video compression Scheme

IV. Basic Video Transmission Scheme

Applications of digital video range from low quality videophones and teleconferencing to high resolution television. The most effective compression algorithms remove the time redundancy with motion compensation. Local image displacements are measured from one frame to the next, and are coded as motion vectors. Each frame is predicted from the previous one by compensating for the motion. An error image is calculated and compressed with a transform code. The MPEG standards are based on this motion compensation. For teleconferencing, color images have only 360 by 288 pixels. A maximum of 30 images per second are transmitted, but more often 10 or 15. If the images do not include too much motion, a decent quality video is obtained at 128kb/s, which can be transmitted in real time through a digital telephone line. The High Definition Television (HDTV) format has color images of 1280 by 720 pixels, and 60 images per second. The resulting bit rate is on the order of lo3 Mb/s. To transmit the HDTV[11] through channels used by current television technology, the challenge is to reduce the bit rate to 20 Mb/s, without any loss of quality.

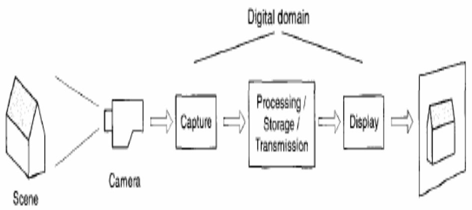


Figure 2. Basic Transmission Scheme

V. A Simple Scheme for Soc Implementation into The FPGA

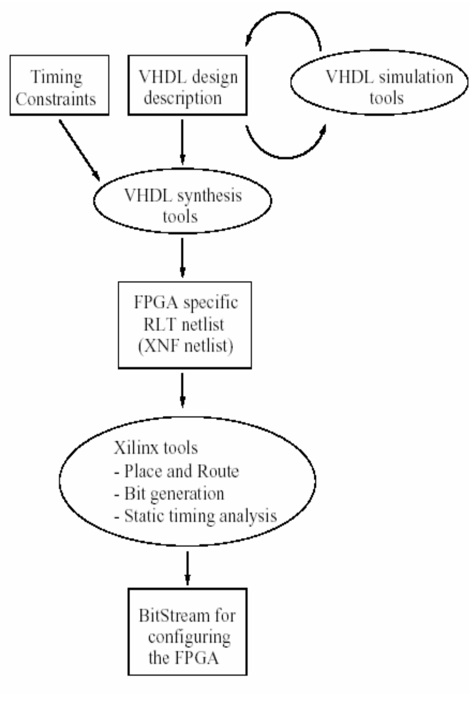


Figure 3. FPGA synthesis and implementation

VI. The Basic Overview and Structure of FPGA

Field Programmable Gate Array (FPGA is a semiconductor device containing programmable logic components and programmable interconnects [1]. The programmable logic components can be programmed to duplicate the functionality of basic logic gates (such as AND, OR, XOR, NOT) or more complex combinatorial functions such as decoders or simple math functions. In most FPGAs, these programmable logic components (or logic blocks) also include memory elements, which may be simple flip-flops or more complete blocks of memories. FPGAs require three major types of elements [2]:

- Combinational logic
- Interconnect
- I/O pins

Figure 4 shows the basic architecture of FPGA that incorporates these three elements.

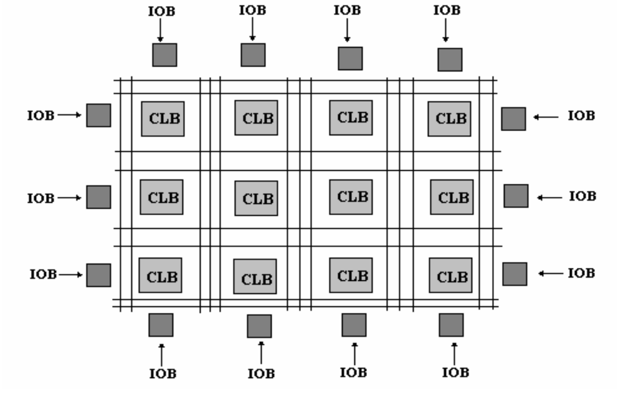


Figure 4. Basic Architecture of FPGA

VII. A Design Architecture of the Compressor / Decompress or

The design is a dictionary style compressor based around a dictionary implemented in the form of a content addressable memory (CAM).The length of the CAM varies from 16 to 1024 tuples (4-byte location) trading complexity for the compression. Typically, the device complexity has been increased 1.5 times the dictionary doubles. It has also master control logic , target control logic and DMA controller . The architecture has 2 independent compression and decompression engines that can work simultaneously in full-duplex mode. Each of this engine has 2 32 –bit ports that are used to move data to and from the PCI interface logic. Each of these ports has own buffering scheme formed by dual port SRAM memory blocks plus control logic that enable a smooth flow of data during compression and decompression operations. A hand –shaking protocol formed by bus request, bus acknowledge and wait signals is used by each of these 4 ports to input or output data to the interface core. Compression and decompression commands are issued through a common 32-bit control data port .A 4-bit address is used to access the internal registers that store the commands, information related to compressed and uncompressed block sizes and CRC (Cyclic Redundancy check) codes to verify the compression operations. Each channel includes a CRC unit that accesses the compression engine or that leaves the decompression engine. A total of 10 registers form the registers bank. 5 registers are used to control the compression channel and the other 5 for the decompression channel. The first bit in the address line indicates if the read /write operation accesses compression or decompression registers. The device includes a test mode that simultaneously decompresses the block being compressed and reports any mismatches in the CRC codes using an interrupt. The architecture is based around a block of CAM to realize the dictionary.

A CAM[14,17,18] is used to store data, much the same as a RAM. The write mode of CAM and RAM is similar to some degree, but the read mode differs significantly. With

RAM we input an address, and get data out. With CAM we input data, and if this data is stored in the CAM, we get the address of that data out. There is an address at the output even if there is no match so with a CAM we need a Match bit to indicate if the CAM contains the input data.

Fig shows a CAM has been applied for string matching in this work. The data to be matched is sent to the CAM as a Byte Strum. In parallel, it is compared to all strings (i.e. words) stored in the CAM. If a match is found, it is indicated by the Match bit. The Match Address reports the "address" of the string that matched in the CAM. Exact string matching is performed and thus, only one (or none) string will give a match. The string matcher has not yet been integrated with the Snort program, although this is the intention.

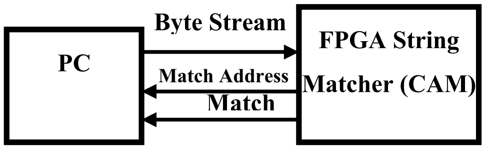


Figure 5. CAM dictionary Interface

The contribution of this paper is also to implement a Variable Word-Width Content Addressable Memory (CAM) in FPGA for string matching. Thereby great flexibility is obtained with respect to how the Snort rule set is defined. In order to design a system for an FPGA, a Hardware Description Language (HDL) is needed. VHDL[18,19,20] was chosen for this work in spite of its somewhat restricted programming features.

VIII. The Dictionary Architecture

In our architecture CAM based dictionary has 16,32 or 64 tuples.The n-tuple dictionary is formed by a total no of n*32 CAM cells. Each cell stores one bit of data tuple and it can maintains its current data, or load the data present in the cell above. The dictionary architecture has been shown in the above fig . The architecture compares the search data with the data present in the dictionary using one XOR gate to do the comparison of each input bit plus ($\log_2$(dictionary width)) 2 input and gates tree to obtain a single comparison bit per dictionary position. The delay of the search operation although in principle is independent of dictionary length, in fan outs and long wires of large dictionaries its speed considerably. An adaptation vector named *move* in fig and whose length equal to dictionary length defines which cells keep its current data and which cells load data from its north-neighboring cell.

We implemented the CAM in hardware by using Xlinx ISE 8.1i.

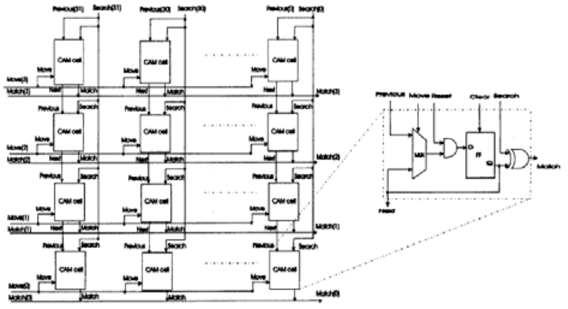


Figure 6. CAM based dictionary Architecture

IX. Implementation and RTL Overview at ime

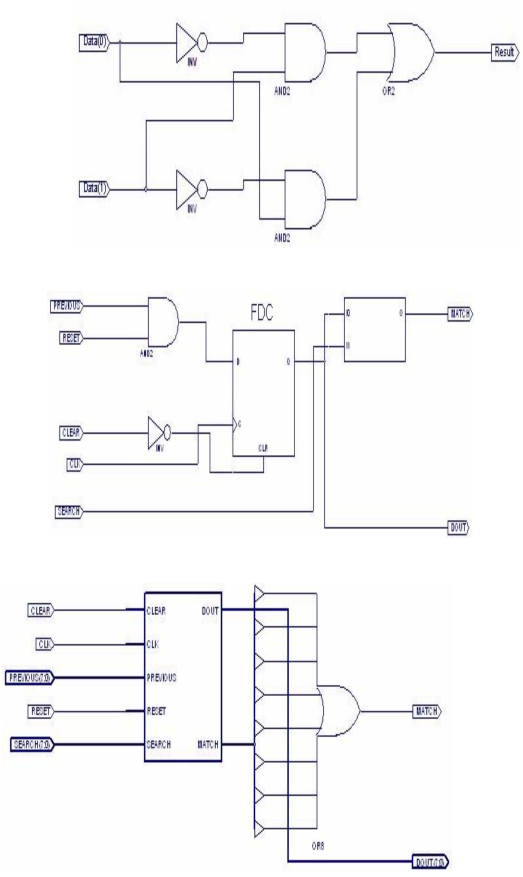


Figure 7. RTL view of CAM

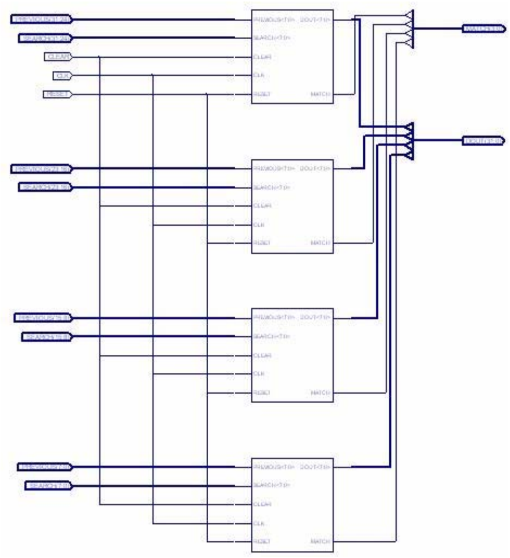


Figure 8. RTL View of the CAM

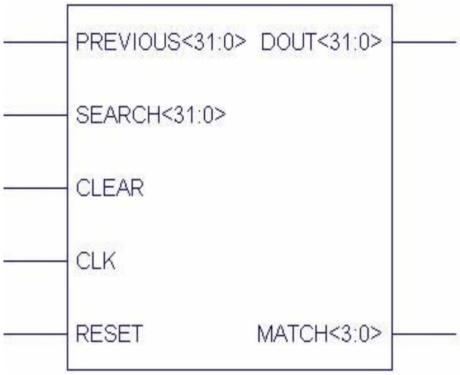


Figure 9. Complete View of the CAM

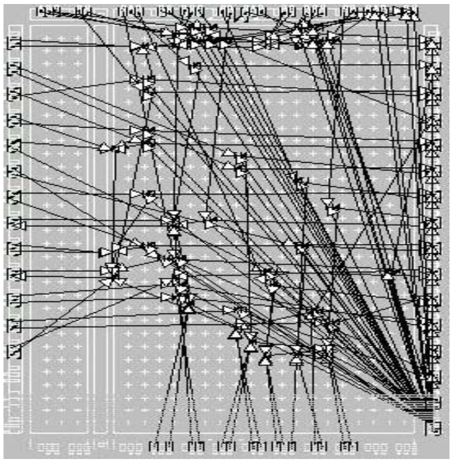


Figure 10. Floor Plan of Proposed CAM Architecture

X. Conclusions

In this paper we presented the starting portion of our research project of ASIC design implementation and verification of large video data transformation SoC[24] design in FPGA System. In future complete pre coder design will be appended to the coders and a complete video compression system or scheme will be realized after mathematical verification of algorithm. The intention is to use XILINX Spartan 3E FPGA for implementation of ASIC Design. More over the design should be tested in real time using XILINX ISE 8i.

XI. Acknowledgment

I am thankful to Prof Amit Konar for his kind guidance. And also thankful to the lab support of the Embedded system Lab of CMERI, Durgapur for complete my work.

XII. References

[1] ISO/IEC 11 172 (MPEG-l),"Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to About 1.5 Mbit/s."

[2] ISO/IEC 13818 (MPEG-2),"Information Technology-Generic Coding of Moving Pictures and Associated Audio Information.

[3] ISO/IEC 10918 (JPEG), "Information Technology-Digital Compression and Coding of Continuous-Tone Still Images.

[4] R. Koenen, F. Pereira, and L. Chiariglione, "MPEG-4: Context and Objectives," Image Communication Journal, Vol. 9, No. 4, 1997.

[5] MPEG-7: ISO/IEC JTCl/SC20/WG211, N2207, "Context and Objectives," March 1998.

[6] G. Wallace, W. B. Pennebaker, and J. L. Mitchell] "Draft Proposal for the Joint Photographic Expert Group (JPEG) ," JPEG-&R6, June 24, 1990.

[7] ISO/IEC WD15444 (V3.0) "JPEG2000 Lossless and Lossy Compression of Continuous-Tone and Bi-level Still Images," 1999.

[8] H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic Coding for Data Compression," Communications of the ACM, Vol. 30, No. 6, June 1987.

[9] A. N. Netravali and B. G. Haskell, Digital Pictures. Plenum Press, New York, 1988.

[10] P. Tong P. Ruetz and P. Ang. "A video-rate JPEG chip set". Journal of VLSI. Signal Processing, 5:141-150, 1993.

[11] Shaw-Min Lei and Ming-Ting Sun. "A parallel variable-length-code decoder for advanced television applications". Proc. 3rd Int. Workshop on HDTV, Aug. 1989.

[12] MPEG-2 International Standard. Video Recommendation ITU- T H.262. ISO/IEC 13818-2, Jan. 20, 1995.

[13] J. Lucas, R. Hoare, I. Kourtev, and A. K. Jones. LURU: Global scope FPGA technology mapping with content addressable memories. In Proceedings of the IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2004.

[14] R. Neale. Is content addressable memory (CAM) the key to network success Electronic Engineering, 71(865):9{12, February 1999.

[15] M. Chang, M. Teklap and M. Ibrahim Sezan, 'Simultaneous motion estimation and segmentation',IEEE Trans Im. Processing 6(9), 1997.

[16] S.J.E. Wilton, `FPGA Embedded Memory Architectures: Recent Research Results", in IEEE Pacific Rim Conference on Communications, Computers and Signal Processing , Aug 1999.

[17] C.J. Jones, S.J.E. Wilton, "Content-addressable Memory with Cascaded Match, Read and Write Logic in a Programmable Logic Device", U.S. Patent 6,622,204. Issued Sept. 16, 2003

[18] IEEE, IEEE standard VHDL Language Reference Manual, std 1076-1993, New York, 1993.

[19]A. Rushton, VHDL for Logic Synthesis, John Wiley, 1998.

[20] J.E. Vuillemin, P.Bertin, D. Roncin, M. Shand, H.H. Touati and P. Boucard, Programmable Active Memories: reconfigurable systems come of age, IEEE Trans. on VLSI Systems, 4:56-69, March 1996.

[21] K. Illgner and F. Muller, 'Image segmentation using motion estimation', in tirne-varying Image Processing and Image Recognition, Elsevier Science, 1997.

[22] R. Castagno and T. Ebrahimi, 'Video Segmentation based on multiple features for interactive multimedia applications', IEEE Trans. CSVT 8(5), September, 1998.

[23] E. Steinbach, P. Eisert and B. Girod, 'Motion-based analysis and segmentation of image sequences using 3-D scene models', Signal Processing, 66(2), April 1998.

[24] S.J.E. Wilton, R. Saleh, `Progammable Logic IP Cores in SoC Design: Opportunities and Challenges", in the IEEE Custom Integrated Circuits Conference, San Diego, CA, May 2001, pp. 63-66.