Design and Implementation of Lossless Data Compression Coprocessor using FPGA

Udaya Kumar H
PG Student(VLSI Design and Embedded Systems)
SIET, Tumkur
Karnataka, India

Madhu B C
Assistant Prof., Dept. of E&C,
SIET, Tumkur,
Karnataka, India

***Abstract* - An implementation of Field Programmable Gate Array (FPGA)-based lossless data compression coprocessor using a compression method developed by Rice. We are implementing the Rice code (both encoder and decoder) for 8 bit/sample data on an FPGA Xilinx spartan 6 Nexys 3.**
The code has been designed to be optimal on 1.5 < H < 7.5 bits/sample, where H is Entropy that is usually required in lossless image compression. The co-processor will reduce the burden of host processor for data compression.The encoder and decoder can achieve 11.6 MHz and 19.4 MHz clock, respectively, where a 10 MHz clock corresponds to a 10Mbits/s throughput. The coprocessor interacts with three major external subsystems Host processor, Memory and channel I/O. The coprocessor consists of a Control unit and a Data path unit. A Host processor ultimately controls the coprocessor to perform its functions.

Keywords: ***Entropy, Rice code , Lossless compression.***

I. INTRODUCTION

The paper describes a Field Programmable Gate Array (FPGA) lossless data compression coprocessor implementing a compression method developed by R. F. Rice. Data compression reduces the number of bits normally required for representing digital datahence, data compression allows more bits to be transmitted through rate limited channels or to be stored in limited storage space.

In a lossless and near–lossless image coding that Rice coder performs well (comparable to arithmetic coder, AC), and in some cases of lossless wavelet image coding it outperforms a Huffman coder. In this paper, we show that such a high–performance coder can be implemented in hardware in a much simpler complexity than that of AC or Huffman coders.

A hardware implementation allows a dedicatedCompression subsystem to be integrated into a system without putting any computation burden to the host processor. In anticipating the need for compression core modules, such as in intellectual property (IP) modules, we have designed the coder hardware to be reusable for other hardware design. The FPGA platform is selected as the target platform to verify the design.

We have implemented the Rice coder (both encoder and decoder) for 8 bit/sample data on an FPGA Xilinx Spartan 6 Nexys3 . The encoder and decoder can achieve more than 1.74 Mbit/s throughput.The *lossless algorithms* are the algorithms which can reconstruct the original message exactly from the compressed message without loss of data. The Rice algorithm which is a lossless data compression method is used. The Rice compression algorithm is one example of a loss less compression algorithm which is suitable for compression of certain types of data under lossless compression. It was developed in part by Robert Rice. This method is a special case of Golomb coding where the concatenation of unary and binary representations of the same symbol is done. By restricting these coefficients to powers of 2 that is if it is power of 2 then it is called as rice coding.

II .COPROCESSOR

We shall now design a coprocessor to implement a coder. We first define the design assumptions especially its external operating environment. We then choose a basic coprocessor computing model. The model consist of control unit(CU) and data path unit(DPU).

A. .Design Assumptions

We assume in real environment the coder consists of encoder and decoder separated by long distance communication channel(See Fig 1) The channel is bit oriented, meaning data are transmitted from encoder to decoder bit by bit serially. Data input of encoder are available locally, stored in memory. The encoder can access memory through bus orientedchannel. Similarly output of the decoder must be stored to memory for future use.

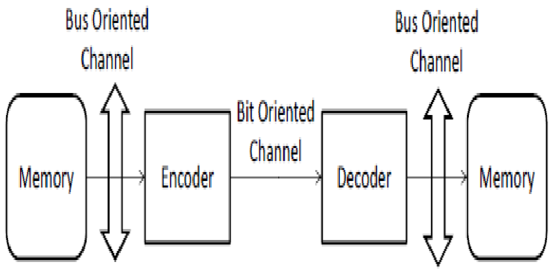

Fig1: Encoder accepts data from memory and producing bit streams to serial channel decoder does the other way around

B. *Basic Coprocessor Computing model*

We propose to use a basic coprocessor computing model shown in figure 2 to satisfy the above environmental requirements The Coprocessor interacts with three major external subsystems The Host processor, memory and channel I/O. The Coprocessor Consists of control unit and data path unit The Host Processor ultimately controls the Coprocessor. Giving commands to coprocessor to perform its functions.

Both the encoder and decoder use the same computing model. An encoder DPU gets input data from memory. Performs actual data compressionand sends data to channel I/O. Conversely decoder DPU gets same data from channel, performs decoding processes, and stores results in memory.

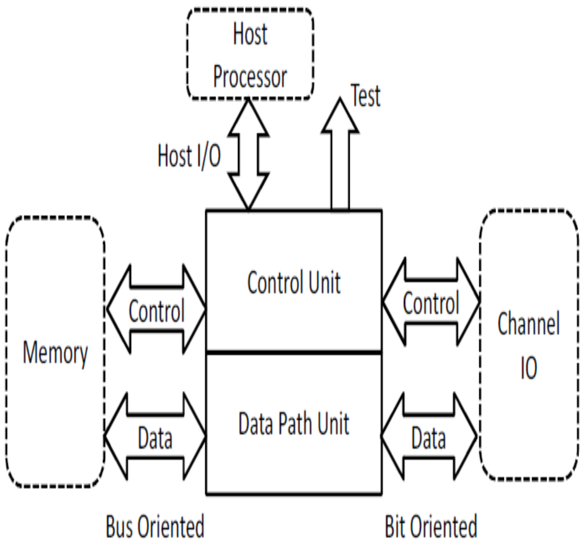

Fig 2 : Coprocessor computing model

The Control unit manages and controls the DPU to ensure synchronised interactions with external sub systems .The CU accepts commands and giving status signals to Host processor through Host I/O. Interactions with memory are controlled through bus control signals. The Channel control signals manage interactions with transmission channel.

The Control unit controls the data compression encoder for the data compression the data compression encoder is designed for the 8 bits of the data and the Control unit controls the buffer the storage element used to store the data for compression the control unit controls through the control signals generated by the control unit based on the requirement the encoded data is converted serially by parallel to serial convertor.

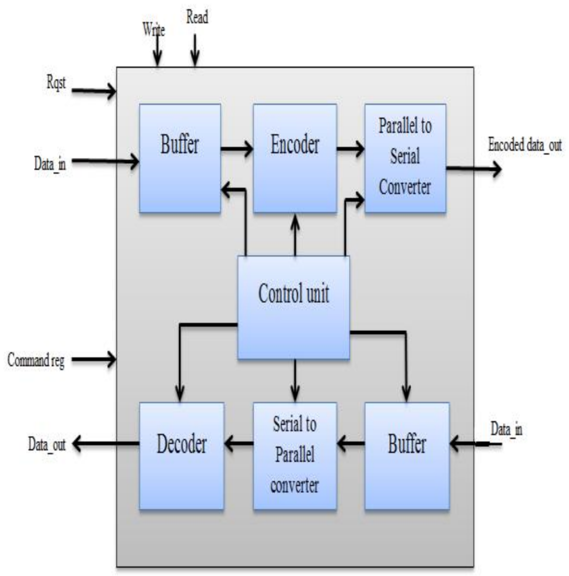

Fig 3: Control unit

The task of compression consists of two components, an encoding algorithm that takes a message and generates a “compressed” representation (hopefully with fewer bits), and a decoding algorithm that reconstructs the original message or some approximation of it from the compressed representation.

If the data is ready for the compression then the control unit provides control signal to get the data for the compression then the compression encoder encodes the data the encoded data is available serially by the parallel to serial converter.

The control unit also controls the decoder and the memory. If the data is available for the decompression then the data is first applied to serial to parallel converter to get the parallel data then the data is given to the decoder to get the original data.

III. RICE CODER

The Rice algorithm uses a set of variable-length codes to achieve compression. Each code is nearly optimal for a particular geometrically distributed source. Variable-length codes, such as Huffman codes and the codes used by the Rice algorithm, compress data by assigning shorter code words to symbols that are expected to occur with higher frequency. By using several different codes and transmitting the code identifier, the Rice algorithm can adapt to many sources from low entropy (more compressible) to high entropy (less compressible).

It is known that Rice codes are a special case of more general Golomb codes where the parameter m is a power of 2, $m = 2^k$, with $k >= 0$. Rice codes have 2^k codes of length, starting with a minimum length of $k + 1$. A significant feature of Rice codes is that it is a very simple coding algorithm. Once the parameter k has been defined, the code is easily constructed by simply separating the k Least

Significant Bits (LSB) of the integer n will become the LSB of code. These will follow the j =[n/2^k] bitsin unary code. These codes are easily constructed with few operations which are notcomputationally expensive. This is an important feature. Finally, it is required to computethe length of a given code constantly, thus a simple equation is desired. Suitably, thelength of a Rice code for an integer n coded using a parameter k can be easily computedas 1 + k + [n/2^k].The k parameter of a Rice coder must be chosen carefully in order to obtain the expected compression ratios for a given set of data.

There are a few well-known Lossless compression techniques, including Huffman coding, arithmetic coding, Ziv-Lempel coding, and variants of each. After extensive study and performance comparison on a set of test images, the Rice algorithm was selected. Some of thelossy data compression techniques are JPEG, MPEG, MP3.

A block diagram of the Rice data compression encoder architecture is shown in figure 4.This method is a special case of Golomb coding where the concatenation of unary and binary representations of the same symbol is done. By restricting these coefficients to powers of 2 that is if it is power of 2 then it is called as rice coding. Rice coding makes the coding process simple resulting in a coding system that can be more efficiently employed in electronic or computerized systems with less complexity.

In the block diagram given in the figure.4 The encoder architecture as many compression options these options are selected based on the parameter K value calculated it is chosen for best compression options. If the option does not provide any compression then the option no compression is considered. The encoder gets the input file for compression the code will be coded such that the encoder gets the correct data for encoding from the input file. After compression the encoder generates the compressed file which will require less amount of the size for storage.

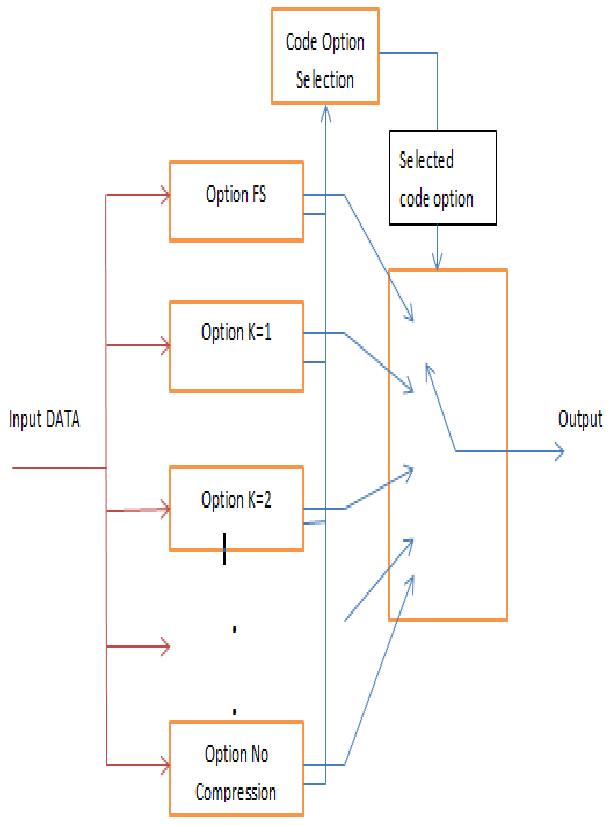

Fig 4 : Encoder Architecture

For each block, the coding option achieving the best compression is selected to encode the block. the entropy coder selects the option that minimizes the number of encoded bits. This includes the identifier bit sequence that specifies which code option was selected.

The option F S means the Fundamental sequence that is for K=0. The data gets encoded by selecting the K Value the K value is selected for the best encoding that is for minimum encoded bits. The option no compression is selected where the data compression is not possible. By using the encoded value the K value used for the encoding can be easily calculated. The encoder architecture is given in the figure the 8 bits of the input data is encoded by using the encoder architecture.

The decoder architecture is given in the figure 5 while decoding the encoded value first the K value is calculated. Then the original message is decoded.

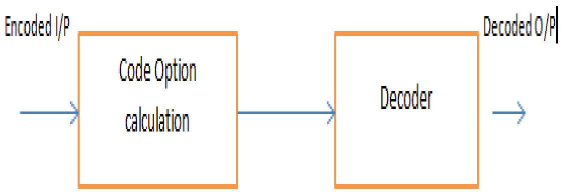

Fig 5 : Decoder Architecture

The encoded value is taken for decoding first the value of K is calculated by using the encoded value then the encoded value can be easily decoded by the decoder.

Before the encoding the value of K is chosen correctly and in the decoding the value of K is calculated by using the enoded value and decoding of encoded value is achieved.To ensure synchronized interactions with external subsystems, we design the
encoder and decoder coprocessor to interact with various signals. Through the signals the coprocessor interacts with host processor, memory, and channel I/O.Both the encoder and decoder use the same computing model. An encoder DPUgets input data from memory, performs the actual data compression, and sends data to channel I/O. Conversely, a decoder DPU gets input from channel,performs decoding processes, and stores the results into memory.

When the coprocessor is ready to receive commands from host processor to process a block of data, the coprocessor issues RDY to host. The host the issues START to trigger the coprocessor. The coprocessor turns off the RDY and start processing the input data from channel. When all block has been processed, the coprocessor activates RDY to tell the host that data are available in the memory and it is ready to process the next block from channel.

We distinguish between the lossless algorithms and lossyalgorithms .*lossless algorithms*, which can reconstruct the original message exactly from the compressed message, and *lossy algorithms*, which can only reconstruct an approximation of the original message. Lossless algorithms are typically used for text, and lossy for images and sound where a little bit of loss in resolution is often undetectable, or at least acceptable. Lossy is used in an abstract sense, however, and does not mean random lost pixels, but instead means loss of a quantity such as a frequency component, or perhaps loss of noise

Example Encode the 8-bit value 18 (0b00010010) when K = 4 (M = 16)

1. S & (M - 1) = 18 & (16 - 1) = 0b00010010 & 0b1111 = 0b0010
2. S >> K = 18 >> 4 = 0b00010010 >> 4 = 0b0001 (10 in unary)

So the encoded value is 100010, saving 2 bits.

Decode the encoded value 100010 when K = 4 (M = 16)

1. Q = 1
2. R = 0b0010 = 2
3. S = Q × M + R = 1 × 16 + 2 = 18.

.

IV. RESULT

After validating the architecture with the Verilog simulations. The implementation of the Rice coder (both encoder and decoder) for 8 bits of data on an FPGA Xilinx Spartan 6 Nexys 3 has been done.The RTL Schematic of the data compression coprocessor is given in the figure 6

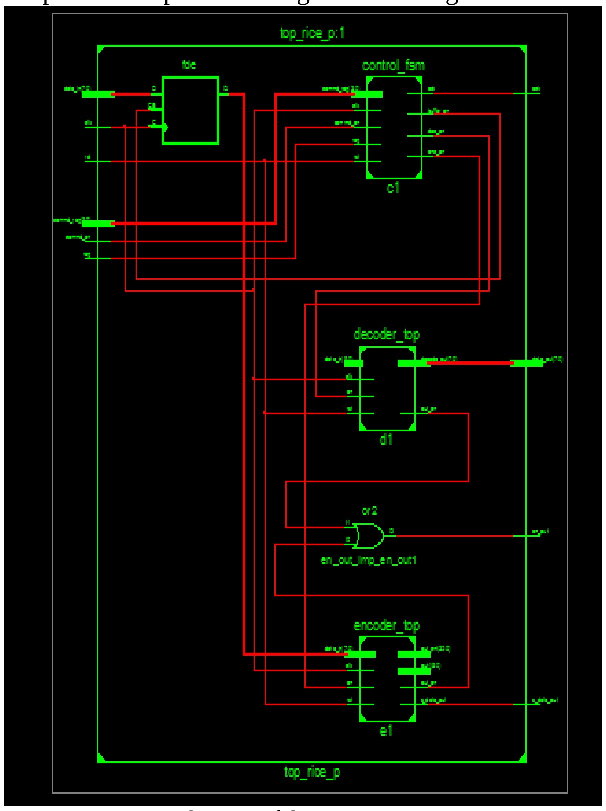

Fig 6: RTL Schematic of data compression coprocessor

In the RTL schematic the control unit, encoder and decoder has been shown all the blocks are simulated and the simulation waveform of the encoder is shown in the figure 7 For the encoder the data has been given in the form of file and the compressed file can obtained from the data compression encoder. The compressed file will have less number of bits then theoriginal file. The original file can be obtained from the decompression process
Furthermore in this particular implementation, theencoder and decoder can achieve 100 MHzclock ratesand 100 MHz clock rate corresponds to a 100Mbits/s throughput, The encoder and decoder are designed and simulated using the

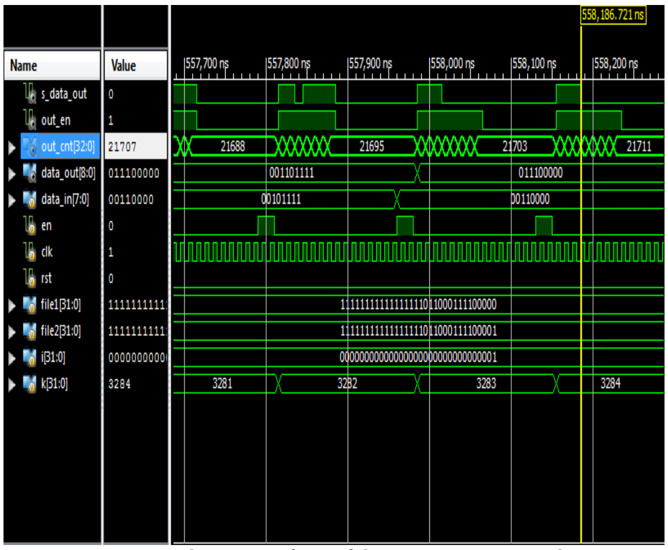

Fig 7: Simulation waveform of data compression encoder

Verilog for the 8bits of data on Field programmable gate arrays (FPGA) The number of bits required to represent the data can be reduced after the data compression.

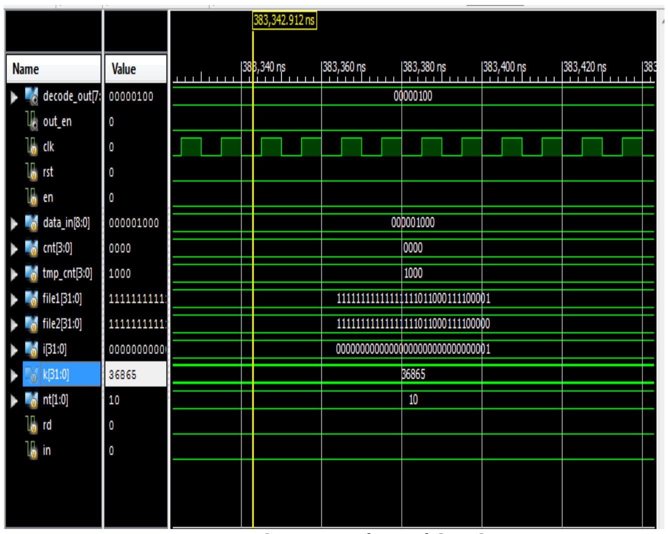

Fig 8: simulation waveform of decoder

The decoder simulation waveform is given in the figure 8. The compressed file is given has input for encoder to obtain the original file

V. CONCLUSION

In anticipating the need for compression core modules, such as in intellectual property (IP) modules, we have designed the coder hardware to be reusable for other hardware design. The FPGA platform isselected as the target platform to verify the design. We have implemented the Rice code (both encoder and decoder)for 8 bit/sample data on an FPGA.Data compression coprocessor with data path unit and control-unit will be implemented. Soft IP core of data compression co-processor will be made.hardware implementation allows a dedicatedCompression subsystem to be integrated into a system without putting any computation burden to the host processor.

ACKNOWLEDGMENT

I wish to thank Mr.Madhu B C, Assistant Professor, department of Electronics and Communication engineering for accepting to be my internal guide and I express my deep sense of gratitude to Dr. M.N Eshwarappa, Professor and Head, Dept. E&C for providing me support required for preparing this paper.

REFERENCES

[1] An FPGA implementation of simple lossless data compression coprocessor, Armein Z.R Langi ITB Research center on information and communication technology.

[2] A. Langi and W. Kinsner, “Wavelet compression for image transmission through bandlimited channels”, ARRL QEX Experimenters’sEchange,(ISSN: 0886–8093, USPS 011–424), No. 151, pp. 12–21, Sep. 1994.

[3] Spartan-3E FPGA Starter Kit Board User Guide UG230 (v1.1) (June 20, 2008.).

[4] A Robust Image Compression Algorithm using JPEG 2000 Standard with Golomb Rice Coding. IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.12, December 2010.

[5] A. Langi, “Review of data compression methods and algorithms”,TechnicalReport,DSP-RTG–2010–9,InstitutTeknologi Bandung, Sep.2010.

[6] A. Langi, “A VLSI Architecture of a Counter-Based Entropy Coder”, ITBJournal of Engineering Science, submitted Feb 2011.

[7] A. Langi, “Lossless Compression Performance of a Simple Counter-Based Entropy Coder”, ITB Journal of information and communication technology, submitted Dec 2010.

[8] A Hardware Architecture of a Counter Based EntropyCoder,Armein Z. R. LangiITB J. Eng. Sci., Vol. 44, No. 1, 2012, 33-48.

[9] Cui, W., New LZW data compression algorithm and its FPGA implementation, Proc. 26th Picture Coding Symposium (PCS 2007),November 2007.

[10] Heliontech.com, Compression Systems, (available at http://www.heliontech.com/comp_sys.htm, accessed August 24, 2011).

[11] Jilani, S.A.K. &Sattar, S.A., JPEG Image Compression Using FPGA With Artificial Neural Networks, IACSIT International Journal of Engineering and Technology, **2**(3), p 252-257, June 2010.

[12] Implementation of LOCO-I Lossless Compression Algorithm using Fuzzy logic. ManasaLekha.J, Mr.Chetan.H. International Journal of Application or Innovation in Engineering & Management (IJAIEM). Volume 3, Issue 7, July 2014.