Journal of Advanced Research in Microelectronics and VLSI
Volume 2, Issue 2 - 2020, Pg. No. 1-5
Peer Reviewed Journal

Article

Design and Execution of Losses Data Comparison using FPGA

Sandhyarani T[1], Sandhana C[2], Basavaliga Swamy[3]

[1,2,3]Assistant Professor, Department of VLSI Design and Embedded System Lingaraj Appa Engineering College, Bidar, Karnataka, Andhra Pradesh, India.

I N F O

Corresponding Author:
Basavaliga Swamy, Department of VLSI Design and Embedded System Lingaraj Appa Engineering College, Bidar, Karnataka, Andhra Pradesh, India.
E-mail Id:
basushastri22@gmail.com
Orcid Id:
https://orcid.org/0000-0002-3616-2189
How to cite this article:
Sandhyarani T, Sandhana C, Swamy B. Design and Execution of Losses Data Comparison using FPGA. *J Adv Res Microelec VLSI* 2020; 2(2): 1-5.

Date of Submission: 2020-09-25
Date of Acceptance: 2020-10-15

A B S T R A C T

A usage of Field Programmable Gate Array (FPGA)-based lossless information pressure coprocessor utilizing a pressure strategy created by Rice. We are executing the Rice code (both encoder and decoder) for 8 bit/test information on Altera Cyclone IV FPGA. The code has been intended to be ideal on 1.5 < H < 7.5 pieces/test, where H is Entropy that is typically needed in lossless picture pressure. The co-processor will decrease the weight of host processor for information pressure. The encoder and decoder can accomplish 11.6 MHz and 19.4 MHz clock, individually, where a 10 MHz clock relates to a 10Mbits/s throughput. The coprocessor collaborates with three significant outside subsystems Host processor, Memory and channel I/O. The coprocessor comprises of a Control unit and a Data way unit. A Host processor eventually controls the coprocessor to play out its capacities.

Keywords: Entropy, Rice Code, Lossless Compression

Introduction

The paper portrays a Field Programmable Gate Array (FPGA) lossless information pressure coprocessor executing a pressure strategy created by R. F. Rice. Information pressure lessens the quantity of pieces typically needed for speaking to computerized information thus, information pressure permits more pieces to be sent through rate restricted channels or to be put away in restricted extra room. In a lossless and close lossless picture coding that Rice coder performs well (tantamount to number-crunching coder, AC) and now and again of lossless wavelet picture coding it beats a Huffman coder. In this paper, we show that a particularly superior coder can be executed in equipment in a lot easier unpredictability than that of AC or Huffman coders.

An equipment usage permits a committed Compression subsystem to be incorporated into a framework without putting any calculation weight to the host processor. In envisioning the requirement for pressure center modules, for example, in Intellectual Property (IP) modules, we have planned the coder equipment to be reusable for other equipment plan. The FPGA stage is chosen as the objective stage to confirm the plan. We have executed the Rice coder (both encoder and decoder) for 8 bit/test information on a FPGA Xilinx Spartan 6 Nexys 3. The encoder and decoder can accomplish more than 1.74 Mbit/s throughput. The lossless calculations are the calculations which can remake the first message precisely from the packed message without loss of information.

The Rice calculation which is a lossless information pressure procedure is utilized. The Rice pressure calculation is one occasion of a misfortune less pressure calculation which is proper for pressure of certain kinds of information under lossless pressure. It was inherent part by Robert Rice. This technique is a special case of Golomb coding where the chain of unary and binary depictions of the same symbol is accomplished. By limiting these coefficients to powers of 2 that is if it is power of 2 then it is called as rice coding.

Reconfigurable architecture: Reconfigurable computing is developing an in-evitable necessity in the domain of high-functioning computing architecture. Reconfigurable computing may be well-defined as a calculation- method which uses a specific hardware that can adapt at the logic level to resolve difficulties. The reconfiguration process provides huge benefits such as power/ size/ cost reduction, hardware reuse, obsolescence avoidance, and application portability. In recent times, the reconfigurability feature of FPGAs has allowed many new applications to be emulated on it.

Data compression can also be completed using software-based techniques. Though software-based computation is adaptable, but it degrades the system functioning as well as bandwidth. Hence, a reconfigurable hardware implementation will not only make the execution speed faster but also will allow the user to execute a plethora of algorithms. Data compression Techniques Recent advances in the field of Information Technology have resulted in enormous amounts of data being generated every second. Subsequently, almost certainly, information stockpiling and transmission will increment to an immense rate. Due to restricted assets, procedures for information pressure were proposed to diminish the size of data being prepared or communicated. The decrease calculation is utilized to move negligible pieces across the organization, moderate stockpiling limit and transmission capacity and effectively and productively send information also.

There are two kinds of information pressure: lossy information pressure and lossless information pressure. Various sorts of pressure procedures are utilized in lossless and lossy pressure of information, for example, Shannon-Fano, Huffman encoding, Arithmetic encoding, Lempel - Ziv, and so on Arrangement of Data pressure Techniques: Lossy information pressure Lossless information pressure Lossy Data Compression: At the point when lossy information pressure is utilized for the pressure of the information, there is a deficiency of some number of pieces for example the packed information and the decompressed information are detached. The capacity productivity is expanded by utilizing the lossy information pressure procedure. Lossless information pressure: Lossless pressure implies compacting information so that when pressure is switched, the first informational collection will be totally re-established.

Coprocessor

We will presently plan a coprocessor to execute a coder. We initially characterize the plan presumptions particularly its outside working climate. We at that point pick an essential coprocessor computing model. The model consist of Control Unit (CU) and Data Path Unit (DPU).

Design Statements: We accept in genuine climate the coder comprises of encoder and decoder isolated by significant distance correspondence channel (See Figure 1). The channel is bit arranged, which means information are communicated from encoder to decoder step by step sequentially. Information contribution of encoder are accessible privately, put away in memory. The encoder can get to memory through transport arranged channel. Additionally yield of the decoder should be put away to memory for some time later, Figure 1.

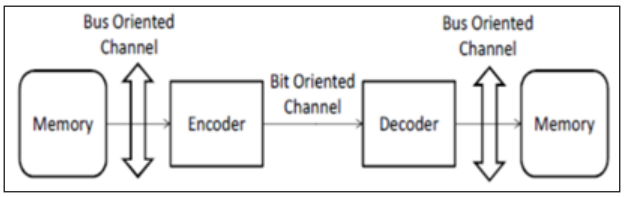

Figure 1.Encoder accepts data from memory and producing bit streams to serial channel decoder does the other way around

Basic Coprocessor

Computing model we propose to utilize an essential coprocessor figuring model appeared in Figure 2 to fulfil the above ecological prerequisites The Coprocessor cooperates with three significant outside subsystems The Host processor, memory and channel I/O. The Coprocessor Consists of control unit and information way unit. The Host Processor at last controls the Coprocessor. Providing orders to coprocessor to play out its capacities. Both the encoder and decoder utilize a similar figuring model. An encoder DPU gets input information from memory. Performs genuine information pressure and sends information to channel I/O. On the other hand decoder DPU gets same information from channel, performs unravelling cycles and stores brings about memory.

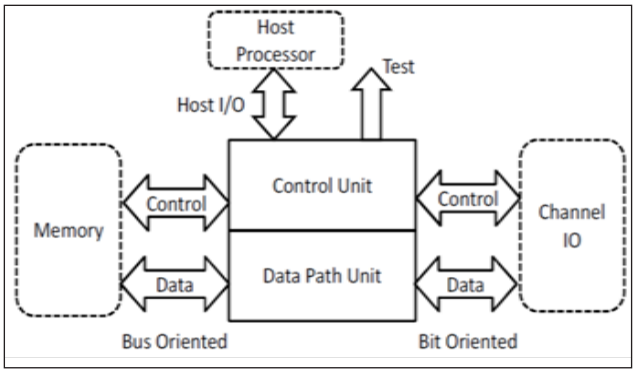

Figure 2.Coprocessor computing model

The Control unit oversees and controls the DPU to guarantee synchronized communications with outer sub frameworks. The CU acknowledges orders and giving status signs to Host processor through Host I/O. Cooperations with memory are controlled through transport control signals. The Channel control signals oversee connections

with transmission channel. The Control unit controls the information pressure encoder for the information pressure the information pressure encoder is intended for the 8 pieces of the information and the Control unit controls the support the capacity component used to store the information for pressure the control unit controls through the control signals produced by the control unit dependent on the prerequisite the encoded information is changed over sequentially by parallel to serial convertor.

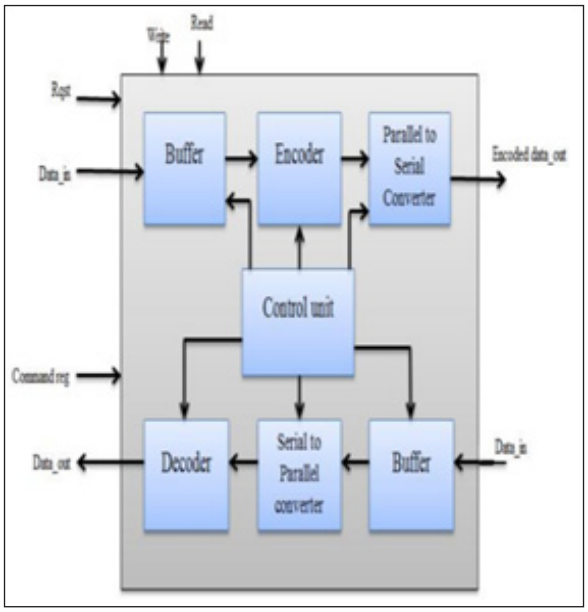

Figure 3.Control unit

The assignment of pressure comprises of two parts, an encoding calculation that takes a message and creates a "compacted" portrayal (ideally with less pieces), and a translating calculation that remakes the first message or some estimate of it from the packed portrayal. On the off chance that the information is prepared for the pressure, at that point the control unit gives control sign to get the information for the pressure then the pressure encoder encodes the information the encoded information is accessible sequentially by the corresponding to chronic converter. The control unit additionally controls the decoder and the memory. On the off chance that the information is accessible for the decompression, at that point the information is first applied to sequential to resemble converter to get the equal information then the information is given to the decoder to get the first information

Rice Coder

The Rice algorithm utilizes a bunch of variable-length codes to accomplish pressure. Each code is almost ideal for a specific mathematically dispersed source. Variable-length codes, for example, Huffman codes and the codes utilized by the Rice calculation, pack information by appointing more limited code words to images that are required to happen with higher recurrence. By utilizing a few unique codes and communicating the code identifier, the Rice calculation can adjust to numerous 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 = 2k , with k > = 0. Rice codes have 2k 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/2k] bitsin unary code. These codes are easily constructed with few operations which are not computationally expensive. This is an important feature.

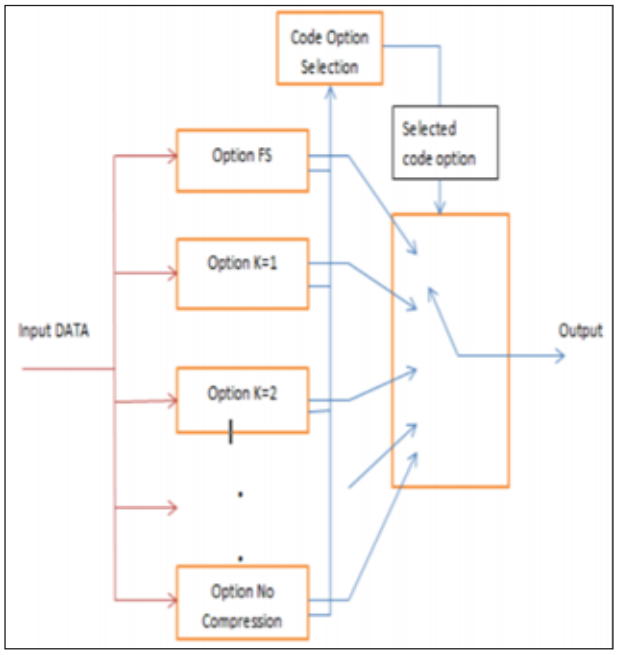

Figure 4.Encoder Architecture

Finally, it is required to compute the length of a given code constantly, thus a simple equation is desired. Suitably, the length of a Rice code for an integer n coded using a parameter k can be easily computed as 1 + k + [n/2k]. 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 methods, 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 the lossy data compression techniques are JPEG, MPEG, MP3. A block diagram of the Rice data compression encoder architecture is shown in Figure 4.

This strategy is an extraordinary instance of Golomb coding where the link of unary and parallel portrayals of a similar

image is finished. By confining these coefficients to forces of 2 that is in the event that it is force of 2, at that point it is called as rice coding. Rice coding makes the coding cycle straightforward bringing about a coding framework that can be all the more proficiently utilized in electronic or mechanized frameworks with less intricacy. In the square chart given in the Figure 4. The encoder engineering the same number of pressure alternatives these choices are chosen dependent on the boundary K worth determined it is picked for best pressure choices. On the off chance that the choice doesn't give any pressure, at that point the alternative no pressure is thought of. The encoder gets the info document for pressure the code will be coded with the end goal that the encoder gets the right information for encoding from the information record. After pressure the encoder creates the packed record which will require less measure of the size for capacity. For each square, the coding choice accomplishing the best pressure is chosen to encode the square. The entropy coder chooses the alternative that limits the quantity of encoded bits. This incorporates the identifier bit succession that determines which code alternative was chosen. The alternative F S implies the Fundamental succession that is for K=0. The information gets encoded by choosing the K Value the K worth is chosen for the best encoding that is for least encoded bits. The alternative no pressure is chosen where the information pressure is preposterous. By utilizing the encoded esteem the K worth utilized for the encoding can be effortlessly determined. The encoder engineering is given in the figure the 8 pieces of the info information is encoded by utilizing the encoder design. The decoder engineering is given in the figure 5 while deciphering the encoded esteem first the K worth is determined. At that point the first message is decoded.

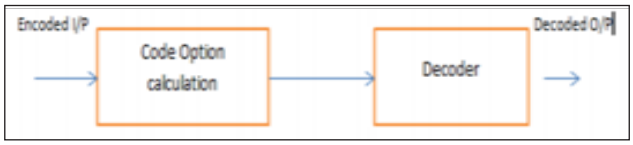

Figure 5.Decoder Architecture

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 utilize a similar registering model. An encoder DPU gets input information from memory, plays out the real information pressure, and sends information to channel I/O. On the other hand, a decoder DPU gets contribution from channel, performs translating cycles and stores the outcomes into memory. When the coprocessor is prepared to get orders from have processor to handle a square of information, the coprocessor issues RDY to have. The host the issues START to trigger the coprocessor. The coprocessor kills the RDY and begin preparing the info information from channel.

At the point when all square has been prepared, the coprocessor initiates RDY to tell the host that information are accessible in the memory and it is prepared to handle the following square from channel. We recognize the lossless calculations and lossy calculations. lossless calculations, which can remake the first message precisely from the packed message, and lossy calculations, which can just recreate a guess of the first message. Lossless calculations are ordinarily utilized for text, and lossy for pictures and sound where a tad of misfortune in goal is frequently imperceptible or possibly worthy. Lossy is utilized from a theoretical perspective, in any case, and doesn't mean arbitrary lost pixels, yet rather implies loss of an amount, for example, a recurrence segment, or maybe 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 = 23. S = Q × M + R = 1 × 16 + 2 = 18.

Results and Analysis

After validating the architecture with the VHDL in simulations. The implementation of the Rice coder (both encoder and decoder) for 8 bits of data on an FPGA Altera Cyclone IV has been done. The RTL Schematic of the data compression coprocessor is given in the Figure 6.

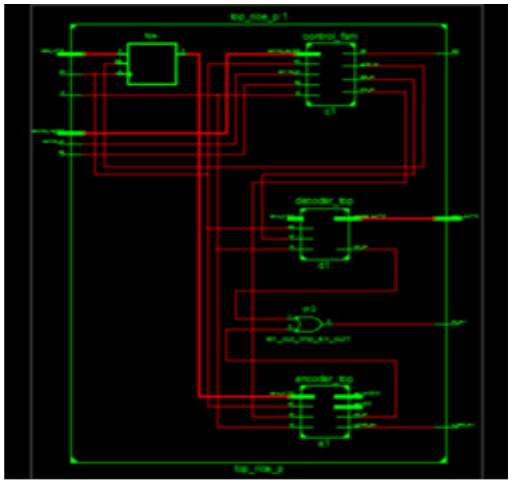

Figure 6.RTL Schematic of data compression coprocessor

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 a smaller number of bits then the original file. The original file can be obtained from the decompression process Furthermore in this particular implementation, the encoder and decoder can achieve 100 MHz clock rates and 100 MHz clock rate corresponds to a 100Mbits/s throughput, The encoder and decoder are designed and simulated using the VHDL 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 The decoder simulation waveform is given in the Figure 8. The compressed file is given has input for encoder to obtain the original file.

Figure 7.Simulation waveform of data compression encoder

Figure 8.Simulation waveform of decoder

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 is selected 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 dedicated Compression subsystem to be integrated into a system without putting any computation burden to the host processor.

References

1. An FPGA implementation of simple lossless data compression coprocessor. Armein ZR Langi ITB Research center on information and communication technology.
2. Langi A, Kinsner W. Wavelet compression for image transmission through bandlimited channels. ARRL QEX Experimenters'sEchange. 1994; 151: 12-21. ISSN: 0886–8093. USPS 011– 424.
3. A Robust Image Compression Algorithm using JPEG 2000 Standard with Golomb Rice Coding. *IJCSNS International Journal of Computer Science and Network Security* 2010; 10(12).
4. Langi A. Review of data compression methods and algorithms. Technical Report, DSP-RTG–2010–9,Institute Teknologi Bandung. 2010.
5. Langi A. A VLSI Architecture of a Counter-Based Entropy Coder. *ITB Journal of Engineering Science* 2011.
6. Langi A. Lossless Compression Performance of a Simple Counter-Based Entropy Coder. *ITB Journal of information and communication technology* 2010.
7. A Hardware Architecture of a Counter Based Entropy Coder, Armein ZR, Langi ITB. *J Eng Sci* 2012; 44(1): 33-48.
8. Cui W. New LZW data compression algorithm and its FPGA implementation, Proc. 26th Picture Coding Symposium (PCS 2007), 2007. Heliontech.com, Compression Systems, (available at http://www.heliontech.com/comp_sys.htm, accessed August 24, 2011).
9. Jilani SAK, Sattar SA. JPEG Image Compression Using FPGA With Artificial Neural Networks. *IACSIT International Journal of Engineering and Technology* 2010; 2(3): 252-257.
10. Implementation of LOCO-I Lossless Compression Algorithm using Fuzzy logic. Manasa Lekha.J, Mr. Chetan H. *International Journal of Application or Innovation in Engineering & Management (IJAIEM)* 2014; 3(7).