RESEARCH ARTICLE OPEN ACCESS

Design and implementation of a reconfigurable hardware for data compression techniques: a Survey

B Jaysree, Harshith D, Deekshith P and Dr. Ipsita Biswas Mahapatra
(Department of electronic and communication, Atria Institute of Technology, Banglore-55
Corresponding author: B Jaysree, Harshith D

ABSTRACT
Explosive data growth in the digital world leads to the need for efficient data storage and transmission technology. Because of limited resources, techniques for data compression are proposed to minimize the size of data being stored or communicated. Since data compression principles result in optimal use of available storage area and bandwidth for interaction, various solutions have been established in several ways. To examine how data compression strategies and their hardware implementations have developed, a comprehensive analysis is performed on many different data compression techniques and their hardware implementations to discuss current data reliability, coding systems, software form and applications specifications. A comparative study of hardware implementations is also carried out to determine the importance of the methods examined in terms of their area, power, features, basic principles, theoretical variables, and weaknesses.
Keywords: Data compression, Data redundancy, Data storage, Data transmission, Hardware implementation

---

Date of Submission: 25-02-2020 Date Of Acceptance: 05-03-2020

I. INTRODUCTION

Reconfigurable architecture

Reconfigurable computing is becoming an in-evitable requirement in the domain of high performance computing architecture. Reconfigurable computing may be defined as a computation- technique which uses a specialized hardware that can adapt at the logic level to solve specific problems. The reconfiguration process provides huge advantages 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 performed using software- based techniques. Though software based computation is flexible but it degrades the system performance 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. As a result, it is likely that data storage and transmission will increase to a huge rate. Because of limited resources, techniques for data compression were proposed to reduce the size of information being processed or transmitted. The reduction algorithm is used to transfer minimal bits across the network, conserve storage capacity and bandwidth, and easily and efficiently transmit data as well.

There are two types of data compression: lossy data compression and lossless data compression.

Different types of compression techniques are used in lossless and lossy compression of data, such as Shannon- Fano, Huffman encoding, Arithmetic encoding, Lempel - Ziv, etc.

Classification of Data compression Techniques:
Lossy data compression
Lossless data compression

Lossy Data Compression**:** When lossy data compression is used for the compression of the data, there is a loss of some number of bits i.e. the compressed data and the decompressed data are indifferent. The storage efficiency is increased by using the lossy data compression technique.

Lossless data compression**:** Lossless compression means compressing data in such a way that when compression is reversed, the original data set will be completely restored.

II. DATA COMPRESSION TECHNIQUES

Huffman encoding

Huffman Coding also called as Huffman Encoding is a famous greedy algorithm that is

used for the lossless compression of data. This uses encoding of variable length where all characters are assigned variable length codes depending on how often they appear in the given data. The most frequently occurring character gets the smallest code and the less frequently occurring character gets the largest code. Huffman coding uses the Bottom- up approach to design a binary tree.

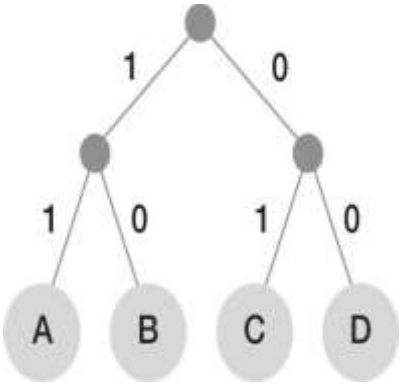

Figure 1: Huffman tree

Shannon-Fano coding

This is the first compression method developed at MIT & Bell Lab by Claude Shannon and RM Fano. It is based on the symbol probabilities in the data. Then build the table on the probability basses containing a number of probabilities. The symbols are divided into groups and subgroups so that sum of probabilities is approximately equal. In this case, it is important to encode the data compression to use binary tree to solve the decoding problem for variable length codes.

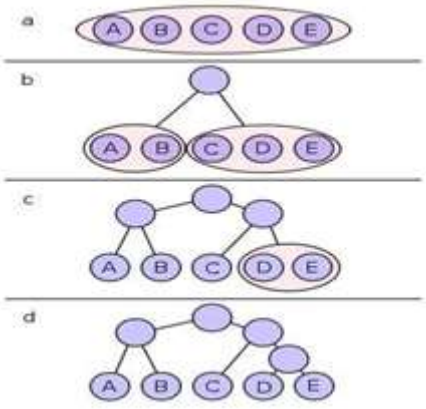

Figure 2: Shannon tree

Lempel-Ziv algorithm

The Lempel-Ziv compression algorithms were developed in 1977-78. Jacob Ziv and Abraham Lempel developed various algorithms of compression based on dictionary concepts such as LZ77 and LZ78.

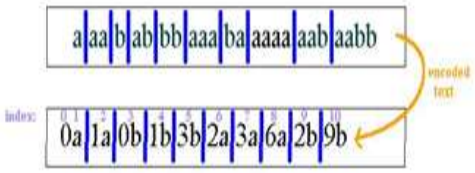

Figure 3: Lempel-Ziv

It has been widely used; while they are enhanced by newer schemes, they provide a straight forward approach to understand universal algorithms of data compression.

III. RELATED WORKS

Huffman Coding

[1] This paper aims at designing and implementing an Enhanced Huffman Tree Coding system which compresses data by encoding it through a sorting table which keeps track of compressed data by assigning dedicated keyword characters. The proposed algorithm increases the decoding operations speed based on the decoder-partitioning principle of the Variable Length Coding. The primary objective of this paper is to evaluate a Huffman high-speed data compression and decompression encoding system. It is proposed to encode an Enhanced Huffman Tree Coding Algorithm (EHTC) via a sorting tree that tracks compression data by assigning keyword characters. The architecture was implemented with HDL in Altera Quartus 16 and modeled in MATLAB [7].Using Altera Cyclone FPGA, the executed design is synthesized.

[4]In this article, the parallel processing features of FPGA and parallel coding of Huffman are used. Unlike the conventional sorting algorithm, this paper uses FPGA's parallel features using a complete parallel comparison algorithm. Huffman coding's general procedure is to construct a Huffman tree by comparing the weights step by step, and then encode or decode the tree. Here we present a FPGA-based Huffman coding implementation involving coding performance, resource utilization and clock frequency issues. The design proposed is developed using Verilog HDL. The simulation, synthesis and implementation is done by IDE of Xilinx FPGA.

[21]This paper describes one of the most popular variable length coding (VLC) known as Huffman Coding. In Huffman algorithm, the basic idea is to assign a shorter codeword for more frequent (probability) data and assign a longer codeword for less frequent data. Here the

parallel Huffman decoder has been designed and implemented successfully using 50,000 gate FPGA (FLEX10K20 from Altera).Parallelism is important to be used in the design to achieve a high frame rate. A code word is guaranteed to be processed within a single clock cycle by means of a parallel technique. The codeword to be processed is matched to the one stored in the Look Up Table (LUT). Parallel Huffman Decoder is a very important technique for ensuring that decoded data is generated at each clock cycle.

Shannon Fano Coding

[8]This paper describes the implementation by VHDL coding of a Shannon-Fano data compression algorithm. One can easily determine how many bits can be saved or how much information can be compressed during transmission using VHDL implementation, and can also see the encoding of the respective data transmission symbol. In Shannon-Fano algorithm sorting is performed according to their frequency or occurrence of the same symbol. Then these sorted symbols are grouped according to their probabilities for their occurrence. ModelSim SE 6.4 simulators are used to implement this algorithm in VHDL.

Lempel-Ziv Coding

[9In this article, they introduce the Lempel-Ziv- Welch (LZW) algorithm's very high speed hardware description language (VHDL) modeling framework for binary data compression to enable definition, validation, simulation and hardware implementation. During the entire process, a data structure known as Dictionary is maintained. The layout of this information consists of no sequence. The encoder compares the substrings extracted from the original text and identifies it in the dictionary; if a correct match is found, a link to the dictionary in the encrypted document would overwrite the substring. It does not do any analysis of the incoming text. Instead, it just adds every new string of characters it sees to a table of strings. In MATLAB / Simulink, the application reference model of data compression utilizing LZW is modeled and a FPGA is used for the hardware implementation of the same.

[18]The main goal of this paper is to improve the efficiency of the Lempel-Ziv algorithm implementation of the systolic-array method. The theory of the LZ algorithms is to seek the longest match duration with the incoming string in the buffer containing the newly obtained information string and represent it with the longest match location and width in the buffer. Since the replicated information is correlated with an older version, there is a more succinct description and compression. The proposed implementation is efficient in area and speed. The compression frequency is improved by over 40% and the development region is reduced by about 30%. The effect on the compression ratio of the chosen buffer size is evaluated. The suggested model will be applied by the FPGA. It checks that on - the-fly data can be compressed and decompressed.

[14]This paper introduces a high-speed low-complex Register Transfer Logic (RTL) design and Implementation of the lossless Lempel-Ziv Welch (LZW) algorithm for High Bandwidth Applicationson Xilinx Virtex II computer family. The proposed design portrays the basic, LZW method adapted for high bandwidth requirements, being applied effectively and the design takes full advantage of the intrinsic compatibility between the two processes and Provides an effective dictionary-based application that allows for high performance over 1 Gbit / s with high efficiency to be achieved. The architecture descriptions provided in this paper demonstrate that the area- and speed-based design improvements take full advantage of the LZW algorithm's flexibility and high compression performance, positioning it among the currently high bandwidth and compact implementations. Compared to other commercially available architectures, the FPGA implementation of LZW algorithm addressed in this paper provides prominent quality benefit in terms of high throughput / slice.

[11]In this paper the dictionary for the LZW algorithm is built based on the Memory Addressable Material (CAM) array. In the dictionary, which uses fewer bits (5 bits) than its ASCII encoding, the code for each character is also available. In this article, the finite state machine applies the LZW software compression algorithm, so the text information can be compressed effectively. Compression of LZW replaces character sequence with keys. The LZW compressor keeps records that have been read from a directory to be condensed with characters. Every character in the dictionary is represented by an index number. They have suggested an optimization scheme for data compression in this article. By using it, memory dictionaries for content access are built into the proposed system. That dictionary character is substituted by a symbol that is less than its ASCII value. The proposed LZW algorithm is assessed in VHDL using finite state machine technology Precise simulation tests were achieved using Xilinx techniques which illustrate an increase in the

lossless data compression scheme by growing the storage space to 60.25% and increasing the compression speed by 30.3%**.**

[20]In this paper they suggest multiple sequential one- dimensional and parallel two-dimensional systolic arrays for compression of Lempel Ziv information .The LZ algorithm's rule is to consider the longest equivalent duration in the buffer containing the recently received data string with the incoming string and represent it with the best match location and length in the buffer. A VLSI processor is designed and evaluated to execute our maximal linear range. The design designed for the array is scalable. The Systolic-Array LZ Compressor (SALZC) is a VLSI chip that is fully custom designed and manufactured using 0.8 m CMOS single-poly double- metal technology for K and. It has 16 PEs and the most cost-effective array architecture, i.e. Type-. Because the range is very standard, full-custom layout is simple. Furthermore, there was only one hand-laid cell (PE). Theother 15 PEs are only copies thereof. Because the set is systolic, it also simplifies routing.

Reconfigurable Architecture

[3]This paper describes the FPGA-supported static and dynamic reconfiguration and its implementation in various applications. Reconfigurable technique has been widely used in FPGA due to various advantages of reconfigurable computing such as increasing efficiency and software flexibility in the hardware computing algorithm. Different benefits of dynamic reconfiguration are addressed along with potential range, such as less area overhead and low power dissipation. FPGA is used in various applications and will dominate in future applications based on hardware and software algorithms due to reconfiguration.

[7]This paper investigates the availability of reconfigurable computing. This paper analyzes the challenges facing reconfigurable computing across three areas, reconfigurable chip, design method and compilation software. Reconfigurable computing technology blends the benefits of General Processor and Domain Specific Processor, narrowing the gap between software and hardware in computational performance. Reconfigurable computing shows prominent benefits in the bad environment, such as high reliability, high efficiency and low power consumption in the field of aerospace.

[10This paper discusses the Field Programmable Gate Arrays that are attractive platform for reconfigurable computing due to their inherent versatility and low cost of entry compared to custom built-in circuits. In many applications, FPGAs have also been shown to outperform general purpose processors by reconfiguring their hardware to the specific application. A partially reconfigurable, multi-tile structure is presented in this article. In this method, an FPGA is divided into tiles, each with the characteristic of being able to contain a full soft processor or common hardware accelerator and also partially reconfigured.

[13]The paper describes an integrated circuit (IC) of a multipurpose lossless data compression coprocessor based on a Field Programmable Gate Array (FPGA) using simple counters that implement a compression Method developed by R. F. Rice. The Rice code (both encoder and decoder) is implemented for 8 bit / sample data on the Xilinx XC4005 FPGA, achieving a throughput of at least 1.74 Mbit/s. This also demonstrates that an X4005 is necessary for the encoder and decoder to be implemented. The quality is good, comparable to that of Huffman than arithmetic coding which is computationally more expensive. In hardware, such a high-performance encoder can be implemented in a much simpler complexity (i.e., using simple counters) than AC or Huffman encoders. For other hardware design, the model coder hardware should be reusable. The FPGA system is selected as the target platform to validate the model.

[16]This paper proposes a method for efficient teaching of reconfigurable computing. This argues for the value of reconfigurable systems in education and proposes a multimedia device complemented by a prototyping system based on FPGA, which could lead to the successful teaching of reconfigurable computing. Hardware definition languages are actually the golden mean for reconfigurable computer model entry methods. They allow high-performance circuits to be created, optimized from the point of view of resource use, the development time is not so long and it is not very difficult to change the design, with such highly programmable platforms having one or more programmable processor(s) and/or reconfigurable logic, derivative designs can be created without a new system- on-chip (SOC) being developed. This multimedia tool is of great value as it offers the opportunity to create, physically execute and test a very complicated system with very minimal hardware description language knowledge.

[19] This paper discusses about designing an Application Specific Integrated Circuit which performs the data compression which

leads to save energy and weight, and improve the integration rate. They also explain about the Compression of data that can be done in two ways, compression with or without loss. The purpose of this paper is to present the PRDC (Payload Rice Data Compressor), the ASIC data compressor implementing the Rice Data Compression Algorithm. The aim of this project is to develop a proper test board with basic functionalities that can supply the ASIC with a data stream to be compressed.

IV. RESULTS AND DISCUSSIONS

The data compression techniques for communication purposes like Military communication, Acoustic underwater communication, CDMA etc, make use of different lossless Data compression algorithms in order to give better results with respect to efficiency, storage space, time consumption or hardware implementation. The below table gives the comparison of the various approaches used for the hardware implementation of various lossless data compression techniques.

Table Comparison:

Author	Year	Technique	Advantages/scope
Sukrut Kesari Pasumarthi, Gabriel Fei,Nan Wang	2018	Huffman(Variable length tree encoding and decoding)	greatly reduces the decoding time ,Saves Bandwidth
Yi Chen, Guo Chun Wan, Ling Yi Tang, and Mei Song Tong	2017	Huffman(parallel full comparison algorithm)	Takes only one clock and small clock cycles to get the sorted data
Z.Aspar, Z. Mohd Yusof , I. Suleiman	2000	Huffman(Serial and parallel Implementation of Huffman decoder)	Decoder is easily modified to customize for a dedicated application
Mahesh Vaidya, Ekjot Singh Walia, Aditya Gupta	2014	Shannon-Fano(Compression through VHDL coding)	All code word lengths are within one bit. Used in ZIP file Compression
Armein Z. R. Langi	2011	Implementation of data compression using rise algorithm	High performance encoder can be implemented in a much simpler complexity

Amandeep Singh, Mamta Khosla ,Balwinder Raj	2017	Reconfigurable architecture using FPGA	Flexible and fast solution for implementing hardware algorithms, static and dynamic reconfiguration is possible
Raymond J. Weber , Justin A. Hogan , Brock J. LaMeres	2013	Reconfigurable architecture using Xilinx Virtex-6 FPGA	Power efficiency
Iouliia Skliarova	2009	Reconfigurable computing using multimedia tool and HDl	Complicated system can be executed and tested with minimal HDL knowledge
Ms. Agrawal Arohi K1 , Prof. S.V. Kulkarni2	2014	Lempel-Ziv (Welch)	Faster compression
Mohamed A. Abd El ghany, Aly E. Salama and Ahmed H. Khalil	2007	Parallel algorithm for LZ based data compression	Area and speed efficient

Shih-Arn Hwang, Cheng- Wen W	2001	Serial one dimensional and parall el two dimensional systolic array	Better hardware cost and testability, higher compression rate
Simrande ep kaur, V.Suloch ana Verma	2012	LZ algorithm(Wel ch - dictionary based)	Reduced storage space and higher compression rate
Saud NAQVI, Rameez NAQVI, Raja Ali RIAZ1, Faisal SIDDIQ UI1	2011	Dictionary based implementatio n(LZW algorithm)	High speed low complexity RTL design and higher throughput efficiency

V. CONCLUSION

We have presented a survey of the various lossless techniques of data compression and its hardware implementation that could be used in satellite communication, military communication, and underwater acoustic communication. The context of this paper was restricted to the commonly applied data compression strategies such as Huffman, Shannon Fano and Lempel-Ziv algorithms. A distinction is also made on the basis of these existing techniques. This paper has been written to understand the hardware implementation of the data compression algorithms in a better manner and relate it to the future work.

REFERENCE

[1] Sukrut Kesari Pasumarthi, Gabriel Fei, Nan Wang, PhD "Enhanced Huffman Coding Algorithm and its FPGA Implementation " [2018]
[2] Yong Liu, Bing Li "Implementation of LZO real-time lossless compression on FPGA" [2017]
[3] Amandeep Singh, Mamta Khosla, Balwinder Raj Reconfigurable Architecture Using FPGA for Low Power Circuit Application" [2017]
[4] Yi Chen , Guo Chun Wan ,Ling Yi Tang , Mei Song Tong "Huffman Coding Method Based on Parallel Implementation of FPGA" [2017]
[5] Xia Zhao, Bing Li "Implementation of the LZMA Compression Algorithm on FPGA" [2017]
[6] Xin Zhou, Yasuaki Ito, Koji Nakano "An Efficient Implementation of LZW Decompression in the FPGA" [2016]
[7] Y Hai, X Zhao, Y Liu " Reconfigurable Computing Availability and Developing Trends" [2015]
[8] Mahesh Vaidya, Ekjot Singh Walia, Aditya Gupta "Data compression using Shannon Fano algorithm implemented using VHDL" [2014]

[9] Ms. Agrawal Arohi K1, Prof. V. S Kulkarni2 "FPGA based implementation of data compression using dictionary based LZW Algorithm" [2014]

[10] Raymond J. Weber, Justin A. Hogan, Brock J. LaMeres "Power Efficiency Benchmarking of a Partially Reconfigurable, Many-Tile System Implemented on a Xilinx Virtex- 6 FPGA" [2013]

[11] Simrandeep Kaur, V.Sulochana Verma "Design and Implementation of LZW Data Compression Algorithm" [2012]

[12] Ivan Shcherbakov, Christian Weis, Norbert Wehn "A High-performance FPGA-Based Implementation of the LZSS Compression Algorithm" [2012]

[13] Armein Z. R. Langi "An FPGA Implementation of a Simple Lossless Data Compression" [2011]

[14] Saud NAQVI, Rameez NAQVI, Raja Ali RIAZ, Faisal SIDDIQUI "Optimized RTL design and implementation of LZW algorithm for high bandwidth application" [2011]

[15] Santana Gil, A.D Benavides Benitez, J.I Hernandez Calviño, M. Herruzo Gómez, E "Reconfigu rable Cache implemented on an FPGA"[2010]

[16] Iouliia Skliarova "A Multimedia Tool for Teaching Reconfigurable Computing" [2009]

[17] Suzanne Rigler, William Bishop, Andrew Kennings "FPGA -Based Lossless Data Compression using Huffman and LZ77 Algorithms" [2007]

[18] Mohamed A. Abd El ghany, Aly E. Salama, Ahmed H. Khalil "Design and Implementation of FPGA-based Systolic Array for LZ Data Compression" [2007]

[19] R. Vitulli "An ASIC device for lossless data compression implementation" [2004]

[20] Shih-Arn Hwang, Cheng- Wen Wu "Unified VLSI systolic array design for LZ data compression" [2001]

[21] Z. Aspar, Z. Mohd Yusof, I. Suleiman "Parallel Huffman Decoder with an Optimize Look UP Table Option on FPGA" [2000]

[22] P.K. Lala, A. Walker "An on-line reconfigurable FPGA architecture" [2000]

B Jaysree, Harshith D, " Design and implementation of a reconfigurable hardware for data compression techniques: a Survey" *International Journal of Engineering Research and Applications (IJERA*), vol.10 (03), 2020, pp 33-4.