Design and Implementation of a Reconfigurable hardware for Source Encoding Algorithms

B Jaysree[1], Harshith D[2], Deekshith P[3], Spoorthi S P[4]

[1,2,3]*Student, Department of Electronics and Communication, Atria Institute of Technology, Karnataka Bangalore, India*
[4] *Professor, Electronics and Communication Engineering, Atria Institute of Technology, Karnataka Bangalore, India*

---***---

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.*

Key Words: *Data compression, Data redundancy, Data storage, Data transmission, Hardware implementation*

1. INTRODUCTION

Signal processing, multimedia and high-speed communications are the major application domains that require lossless compression techniques. In this work, we propose to design a re-configurable hardware which will have the ability to customize its internal architecture to match the computation and data flow of different source coding algorithms.

1.1 Data compression

Data compression is the art of representing the information in a compact form rather than its original form. More concisely, data compression comprises the identification and removal of redundant elements of source data. The main objective of data compression is to reduce the size of file to store or transmit over communication channels. Data compression was 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.

1.2 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.

1.3 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.

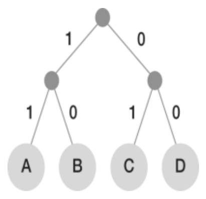


Fig-2: Huffman tree

1.4 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.

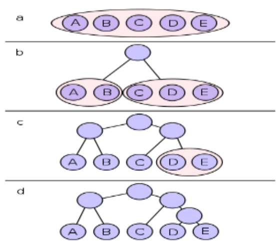


Fig-3: Shannon Fano tree

In this work, we propose to design a re-configurable hardware which will have the ability to customize its internal architecture to match the computation and data flow of different source coding algorithms. Our proposed technique is a combination of Huffman encoding and Shannon-Fano encoding algorithms. Software implementation for the proposed system cannot fulfill the performance and real time requirements, though software based computation is flexible but it degrades the system performance as well as bandwidth, where as the hardware implementation of the proposed system will be more efficient.

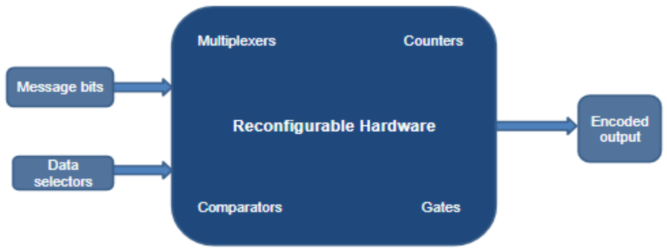


Fig-1: Basic block diagram

2. LITERATURE SURVEY

[6] An Enhanced Huffman Tree Coding Algorithm and its FPGA Implementation

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. Data compression plays a vital role in today's communication, where limitation in bandwidth leads to slower communication.

[1] Low Power Text Compression for Huffman Coding using Altera FPGA with Power Management Controller

This paper aims to design and implement Huffman coding based on binary trees using FPGA (Field Programmable Gate Arrays), and proposing a novel method of clock gating and frequency scaling to achieve low power consumption and reliability of design. The proposed Huffman achieved a 47.95% saving percentage in data size. While, reduce power consumption up to 52.52% compared to traditional design.

[3] Reconfigurable Architectures Using FPGA for Low Power Circuit Application

This paper presents a survey on reconfigurable computing, types of reconfiguration, its implementation on FPGA, selection of FPGA for typical application and factors responsible for it, its advantages and applications. It also describes the static and dynamic reconfiguration supported by the FPGA and its implementation in different applications. Various advantages of dynamic reconfiguration like less area overhead and low power dissipation are discussed along with future scope.

[8] Data compression using Shannon-Fano algorithm implemented by VHDL

In digital communication while transmitting the data it is well desired that the transmitting data bits should be as minimum as possible, so to compress the data there are several techniques used. In this paper we have implemented a Shannon-Fano algorithm for data compression through VHDL coding. Using VHDL implementation we can easily observe how many bits we can save or how much data gets compressed during transmission, and we can also see the encoding of the respective symbol of transmit data.

[10] Hardware implementation Of the Huffman encoder for Data Compression using Altera De2 Board

This work will aim to describe hardware implementations of static and dynamic Huffman encoders written in VHDL. The flexibility of the design allows for hardware-based implementations using FPGAs.

3. METHODOLOGY

The main goal of the project is to design a hardware that performs data compression techniques like Huffman encoding and Shannon Fano encoding. This chapter discusses the details of the design and implementation of the proposed system. Details of various functional blocks as well as their FPGA implementation will also be described.

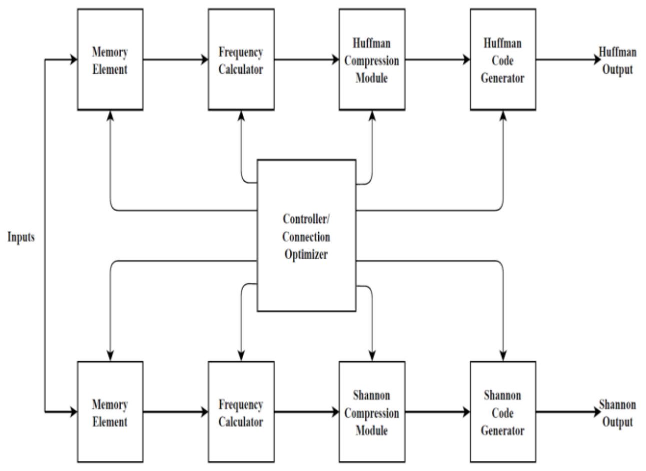


Fig-3: Detailed Block Diagram

3.1 Components used

Hardware: Kintex 7 FPGA

Software: MATLAB, Vivado HLS, Xilinx ISE

3.2 Controller/Connection optimizer

The main feature of a dynamic Huffman coder is that the histogram is calculated from the input data. The purpose of this module is to take input, processing data by enabling other modules and store output result into memory. After performing necessary operation, the data will be written into given memory location. The main purpose of this module is to coordinate between the different modules in the design and to optimize the connection between the modules to make the design reconfigurable.

3.3 Memory element

Memory element consists of 15 bit register, this block stores the input data which is passed down to the frequency calculator, status signal, made1 once the complete data is passed on to the Occurrence Calculator, clock is the clock signal which activates the occurrence calculator on rising edge, reset is the reset signal, which when high, initializes all the memory locations to zero.

3.4 Frequency calculator

This block calculates the occurrences of the number of unique words present in the data. It generates a control signal to the Huffman compression module as status S, to indicate the completion of occurrence calculation and activates the Huffman compression module (when S=1) ,clock signal which activates the frequency calculator on rising edge, reset when high, all the occurrence values are initialized to zero. Data_in is the 15 bit input data fed to the frequency calculator from the memory element.

3.5 Huffman and Shannon Fano Compression Module

This module is responsible for second level compression in transmission unit. At first, it computes the frequency of the input characters. Based on the frequency of input data set, it constructs Huffman tree and Shannon Fano tree. Later, this Huffman tree and Shannon Fano Swill be used to encode each character of the source data. Huffman and Shannon Fano compression module is the most complex module of this project. It is divided into several sub-modules which will be described in following sections.

3.5.1 Frequency Sorting Module

Sorter has input values that are taken from the frequency calculator and sorts them is descending order descending order, when the enable signal is active high and the reset signal is active -low. The sorted values are passed to the Adder.

3.5.2 Adder

Adder module takes two numbers 'NUM_1' and 'NUM_2' from the output of the frequency sorting module and performs addition. Basically this addition is required to build Huffman tree where two least frequency will be summed up and added as a new node. Result of the addition is another number which will be sent through the port 'SUM'.

3.5.3 Huffman and Shannon Fano Code generator

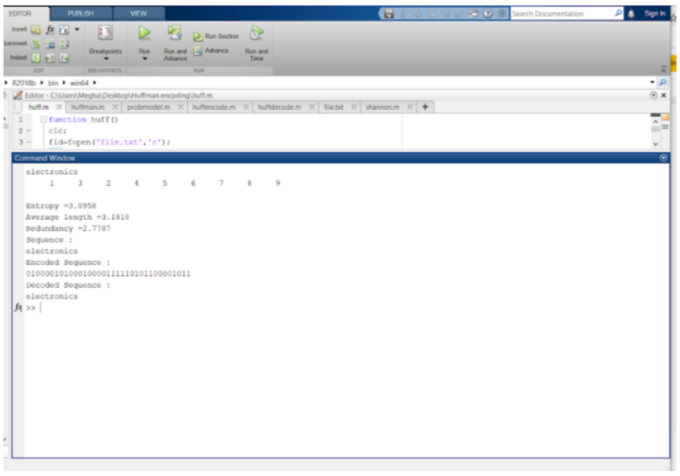
This is the block where the actual Huffman code is generated .The Huffman Code generator takes the input from the Adder and the frequency sorter to generate the Huffman Tree. The code generated in the code generator is passed to the encoder block as input where each input data is encoded with its corresponding code. It performs the compression by replacing each of the unique word with its equivalent code words and stores the compressed data.

3.6 IMPLEMENTATION

The implementation part will be carried out by MATLAB implementation (codes) of the selected algorithms. MATLAB simulation is done for checking the functionality of the proposed system. The MATLAB codes are converted to HDL codes for the hardware implementation process.

The simulation of the HDL codes is done using Vivado HLS for scrutinizing the functionality. Subsequently the RTL schematics are generated. The reconfigurable architecture for the system is designed by comparing and extracting the similar blocks from the obtained RTL schematics.

Implementation of the proposed hardware on the Kintex FPGA will be accomplished. The input for this hardware will be in terms of 14 bit array of characters, each character is assigned with an optimized binary code which results in the compressed data at the output. One sample window showing the MATLAB simulation results for the designed hardware is shown in the figures 4 and 5.

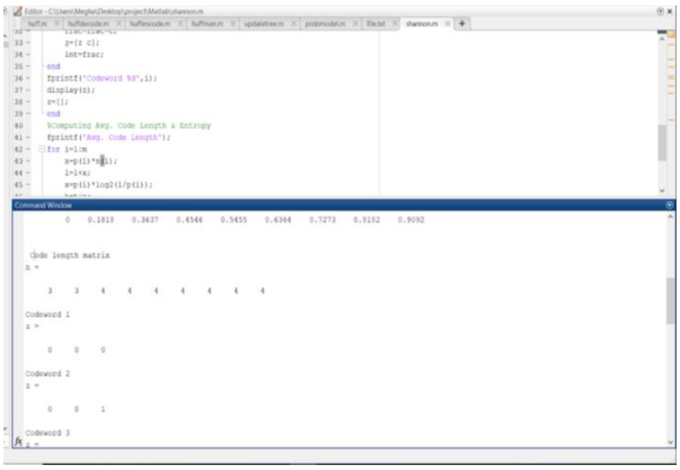


```
function huff()
clc;
fid=fopen('file.txt','r');
```

```
electronics
     1     3     2     4     5     6     7     8     9

Entropy =3.0958
Average length =3.1818
Redundancy =2.7787
Sequence :
electronics
Encoded Sequence :
0100001010001000011111010110000101l
Decoded Sequence :
electronics
>>
```

Fig-4: Simulation of Huffman algorithm



```
        z=[z c];
        int=frac;
    end
    fprintf('Codeword %d',i);
    display(z);
    z=[];
end
%Computing Avg. Code Length & Entropy
fprintf('Avg. Code Length');
for i=1:m
    x=p(i)*n(i);
    l=l+x;
    x=p(i)*log2(1/p(i));
```

```
         0    0.1818    0.3637    0.4546    0.5455    0.6364    0.7273    0.8182    0.9092


Code length matrix
n =

     3     3     4     4     4     4     4     4     4

Codeword 1
z =

     0     0     0

Codeword 2
z =

     0     0     1

Codeword 3
z =
```

Fig-5: Simulation of Shannon Fano algorithm

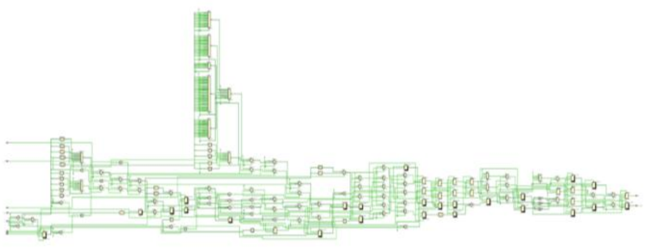


Fig-6: RTL schematic of Huffman encoding

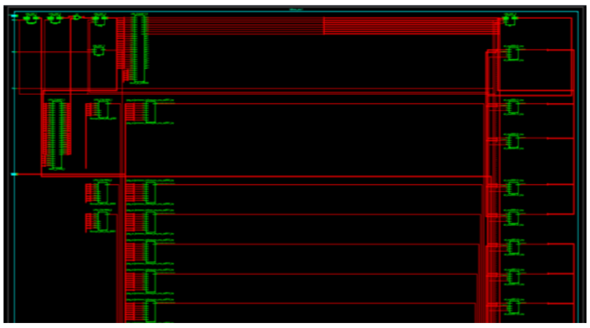


Fig-7: RTL schematic of Shannon-Fano encoding

4. APPLICATIONS AND ADVANTAGES

Source encoding plays a very important role in processing issues. By encoding text one can secure the data and which requires less memory for storage which in turn leads for faster transfer of data.

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.

5. CONCLUSION

Reconfigurable hardware is successfully made use of by the proposed system for the Data compression applications. The reconfiguration process provide huge advantages such as power/size/cost reduction, hardware reuse and application portability The proposed system successfully generates the encoded outputs by utilizing Huffman and Shannon Fano algorithms for the given text input. The propounded hardware is computational intensive viz. a single hardware carry out multiple tasks which needed grid computing. The computational pace to perform the encoding process using the hardware is comparatively more than the software execution.

6. FUTURE SCOPE

The proposed hardware can be emulated on an Application Specific Integrated Circuit (ASIC) which can be used in the encoding algorithms where smaller area is necessary. Utilizing the parallelism and pipelining feature of the FPGA various Data compression algorithms (Huffman, Lempel-Ziv, Arithmetic and Shannon-Fano) can be delineated.

ACKNOWLEDGEMENT

A project report is a job of great enormity and it can't be accomplished by an individual all alone. We are grateful to a number of individuals whose professional guidance and encouragement have made it a pleasant endeavor to undertake this project.

REFERENCES

[1] Maan Hameed, Hussein Shakor, Intensar Razak "Low power Text compression for Huffman Coding using Altera FPGA" [2018]M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[2] Yong Liu, Bing Li "Implementation of LZO real-time lossless compression on FPGA" [2017] K. Elissa, "Title of paper if known," unpublished.

[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 enhanced Huffman tree coding Algorithm" [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] Ms D.M Kate, "Hardware implementation of Huffman encoder for Data compression using Altera De2 board"

BIOGRAPHIES

B Jaysree
Bachelor of Electronics and Communication Engineering

Harshith D
Bachelor of Electronics and Communication Engineering

Deekshith P
Bachelor of Electronics and Communication Engineering

Prof. Spoorthi S P
Assistant professor
Department of Electronics and Communication Engineering