RESEARCH ARTICLE OPEN ACCESS

Implementation of Huffman Decoder on Fpga

Safia Amir Dahri[1], Dr Abdul Fattah Chandio[2], Nawaz Ali Zardari[3]
Department of Telecommunication Engineering, QUEST NawabShah, Pakistan[1,3]
Principal, QUCEST Larkana, Pakistan[2]

Abstract
Lossless data compression algorithm is most widely used algorithm in data transmission, reception and storage systems in order to increase data rate, speed and save lots of space on storage devices. Now-a-days, different algorithms are implemented in hardware to achieve benefits of hardware realizations. Hardware implementation of algorithms, digital signal processing algorithms and filter realization is done on programmable devices i.e. FPGA. In lossless data compression algorithms, Huffman algorithm is most widely used because of its variable length coding features and many other benefits. Huffman algorithms are used in many applications in software form, e.g. Zip and Unzip, communication, etc. In this paper, Huffman algorithm is implemented on Xilinx Spartan 3E board. This FPGA is programmed by Xilinx tool, Xilinx ISE 8.2i. The program is written in VHDL and text data is decoded by a Huffman algorithm on Hardware board which was previously encoded by Huffman algorithm. In order to visualize the output clearly in waveforms, the same code is simulated on ModelSim v6.4. Huffman decoder is also implemented in the MATLAB for verification of operation. The FPGA is a configurable device which is more efficient in all aspects. Text application, image processing, video streaming and in many other applications Huffman algorithms are implemented.
Keywords*:* Huffman algorithm; FPGA; Variable length coding; communication;

I. INTRODUCTION

Recent developments in the field of hardware design make it possible to achieve higher speed and parallelism. Due to which software based systems or algorithms used in various communication systems and digital signal processing are implemented in Hardware. Huffman decoder is one of the most widely used algorithms to transmit variable length code in order to reduce bandwidth, higher efficiency and higher data rate. The Huffman decoder algorithm can also be implemented on field programmable gate array (FPGA). The FPGA is reconfigurable and programmable device through which hardware can be formed and it can achieve more speed than software counterpart. In order to achieve these benefits, we need to create hardware and software platform. Realization is done on Spartan-3E FPGA board. To program FPGA, Xilinx ISE8.2i software is used, which performed synthesis, design, implementation and bit file generation for the Huffman design file. That file will be downloaded into the FPGA by an iMPACT tool. The result verification is done by simulating the design and MATLAB is used to perform Huffman decoding.

Huffman algorithm is mostly used in data compression in software applications. In this paper, Huffman algorithm decoding is implemented on FPGA. MATLAB is also used to verify and simulate the data. The ModelSim is used to simulate the data for clear visualization of results. The motivation of hardware implementation of Huffman decoder is clear from the fact that, the hardware implementation is fast, and every algorithm is being implemented on FPGA. Digital signal processing algorithms, fast fourier transforms and filter design is now implemented on hardware i.e. Field Programmable Gate Array.

In recent years, the demand for lossless data compression is increased drastically and software implementation is not sufficient, hence Huffman decoding algorithm is needed to implement on hardware.

A Huffman decoder algorithm is used in data communication and other fields. It requires less transmission time and storage space because it has a minimum average length. It has unique prefix properties, it does not require end of character delimiter. It is easy to implement and provides a lossless compression and minimum redundancy.

It gains popularity due to its variable length coding. It is used in data transmission, video and voice streaming, wireless coding where transmission and reception of data is needed. Some fields where Huffman encoding and decoding is most widely used, are, Text application (Zip and Unzip) For JPEG compression For image/video coding standards H.261, H.263 and MPEG1 and 2. MPEG format compression and decompression.

II. PREVIOUS WORK

This is the type of source coding that was developed by David A. Huffman in 1952. This is also called entropy coding algorithm because it follows the principle that the average number of bits per

message should be less or equal to the average bits of information per message which means entropy [1]. To achieve the higher efficiency, we have to reduce the redundancy and for that purpose variable length coding algorithms were developed. Both the Shannon fanno algorithm and Huffman algorithm produce variable length codes because according to these theories the small numbers of bits are assigned to frequently occurring symbol as compared to rare occurred symbols.

Shannon fanno algorithm and Huffman algorithm working on the same principle and depends upon the probability of occurrence of each symbol or message, but the difference is that unlike Shannon fanno algorithm, the Huffman algorithm produces an average number of bits per message equal to the entropy [1].

Currently, some features of embedded systems such as speed, time and power consumption make the system more reliable and to be used in a wide variety of applications [2]. It is the reason, software based algorithm and digital signal processing algorithms are implemented in hardware. Recent advances in the field of programmable devices, i.e. FPGA make realization simple on hardware. The FPGA is programmable hardware. After programming it behaves as hardware. It is used in many applications on a software level.

A few decades earlier, Hatsukazu Tanaka, presents his work about Huffman codes and its applications in various fields [3]. In 2000, Zulifkar Aspar et al presents his research paper about parallel Huffman decoder on FPGA. He describes that parallelism is important in some applications like JPEG and MPEG. He also describes serial and parallel bit Huffman decoder [4]. In 2004, Ying Chen et al describes the JPEG application to make it lossless. They used integer reversible transformation. They also compare iJPEG and JPEG. More about this can be found at [5]. In 2009, Tomas Frysa et al contributed in the field of wireless transmission for video streaming. In order to code the video streaming, Huffman coding is used because of any advantages. Their approach was to insert special symbols in the Huffman codeword [6]. In 2010, Hoang-Anh Pham et al described the adaptive Huffman algorithm in the applications of MP3 encoding and decoding [7]. They used single side growing approach for Huffman coding. In 2012, Ke Zhu et al presented his work about Huffman decoder implementation in the JPEG application [8]. He presented the hardware JPEG decoder with three functions for embedded systems. This decoder could decode a JPEG image, and widely used in digital camera, mobile phones and tablet PC, etc. It uses soft IP core for processing.

III. FPGA TECHNOLOGY AND PROGRAMMING

The FPGA gardware contains logic blocks to program and build specific function or circuit rather than planes of AND/OR gates. There are more advantages of FPGA over ASIC (Application Specific Integrated Circuit). Major advantages are reconfigurability, higher performance, higher speed and cost effectiveness. The only disadvantage can be considered is that the circuit formed inside the FPGA is larger than the ASIC design [9], [10], [11]. The FPGA architecture is also called FPGA fabric. The architecture consists of three main components.

- Logic Element
- Interconnects
- Input / Output Pins

The Fig.1. shows the generic structure of FPGA fabric. Logic elements or CLB (Combinational Logic Block) is the smaller unit and it consists of Look-Up Table (LUT) which usually forms the functions of most logic gates. These logic elements are combined to form a larger design. The Logic elements are connected through interconnects that can be programmed. The FPGA has different types of interconnect depending on the distance between the logic elements to be connected as shown in Fig. 2.

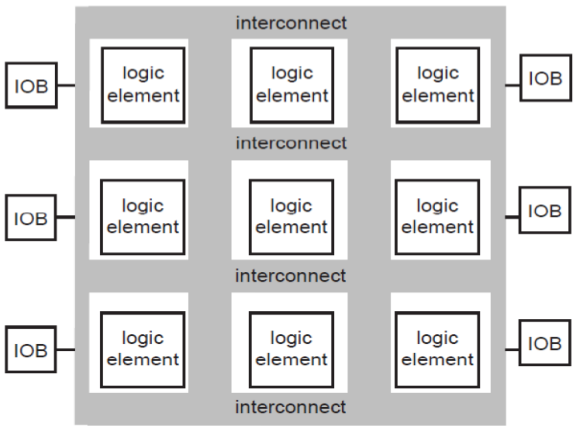

Figure 1. FPGA Fabric/architecture

The third element I/O pins also known as IOBs (I/O Blocks) to connect the function implemented to the outside world [11]. The FPGA is discussed so far, can be programmed by language, and is called Hardware Description Language (HDL). There are two major types HDLs which are most widely used languages are VHDL and Verilog HDL. The program written in these two languages are behaving as a higher level of abstraction. In this language the hardware is built in the form of codes, which means it represents the textual representation of Hardware. These languages can support RTL, Switch, logic and other abstraction levels [12]. There are following

processes used in order to generate a bitstream file for FPGA programming, i.e. Synthesis, translate, map, place & route and bit file generation. Bit file is downloaded into the FPGA by using impact tool.

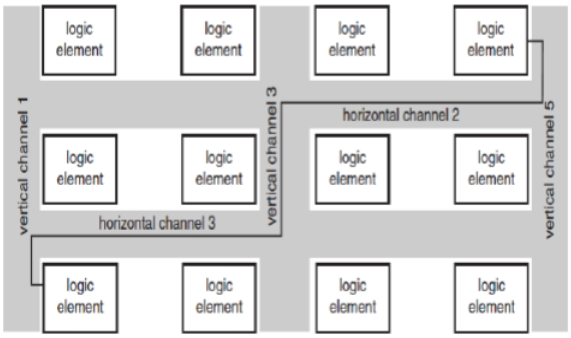

Figure 2. FPGA Interconnects

IV. METHODOLOGY AND IMPLEMENTATION ON SPARTAN-3E

There are two main manufacturers of FPGA boards, Xilinx and Altera. The board used in this paper is from Xilinx Spartan series i.e. Spartan3E. The Spartan3E can be programmed with the couple of steps required. The software used to program the FPGA on Spartan3E FPGA board is Xilinx ISE 8.2I, provided by Xilinx have been used. The below flow chart shows the methodology through which Huffman coding is done in HDL.

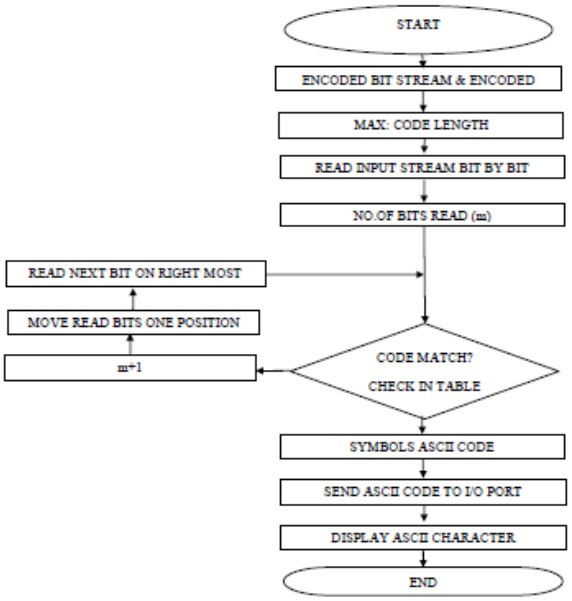

Figure 3. Proposed Methodology (Flowchart)

The language used to program the FPGA is VHDL, which is described in detail below.

A. VHDL

In this paper, Code is written in VHDL language. In 1980, there were needed to make a standard language in order to structure and function of integrated ICs. For that purpose, a very high speed integrated IC program (VHSIC) was initiated in the USA under the contract from DRAPA, which results in the development of VHDL Language [13].

The structure of the VHDL program consists of two parts, entity and architecture. The entity is declarative part and contains the input output ports. In other words, it represents the black box model of the circuit, while architecture referred to the internal circuitry or internal structure of the actual logic.

VHDL contains a wide variety of data types and its conversions. It is not case sensitive. VHDL is like any other programming language, it contains identifier, comments, variables, constants, numbers, string, bitwise operations, and enumeration types and many more features [13]. VHDL also fairly deals with an array to analyzed data in one, two or more dimensions. Advanced VHDL covers signal resolution and buses, concurrent statements and assertions in procedure call, etc. There are some disadvantages of VHDL; it does not cover the all levels of abstractions.

When the code is downloaded into the FPGA and run the several process, it shows the device utilization summary for a particular FPGA as shown in below Fig. 4.

Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Latches	5	9,312	1%	
Number of 4 input LUTs	145	9,312	1%	
Logic Distribution				
Number of occupied Slices	74	4,656	1%	
Number of Slices containing only related logic	74	74	100%	
Number of Slices containing unrelated logic	0	74	0%	
Total Number of 4 input LUTs	145	9,312	1%	
Number of bonded IOBs	10	232	4%	
IOB Latches	7			
Total equivalent gate count for design	1,164			
Additional JTAG gate count for IOBs	480			

Figure 4. Device Utilization summary for Spartan 3E

V. RESULTS AND DISCUSSION

In this paper, Huffman decoder is implemented and its operation is verified on FPGA. In order to verify the Huffman decoder, firstly it is designed and implemented in Matlab. Then for testing purpose a text file is taken that contains the data i-e Safia Amir Dahri (11-MCME-04) and applied to Huffman encoder. The encoder has generated the encoded bitstream and table that shows code for individual data as shown in Fig. 5.

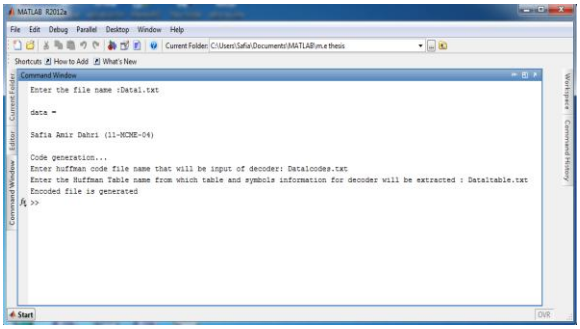

Figure 5. MATLAB Program for encoding

The figure 5 shows that after running the Huffman encoder program, it will ask for the text file that is to be compressed so the Data1.txt is the name of file and in the next line the data of the file is shown. Then for code generation program requires the file on which encoded bitstream and table is to be saved, for that purpose Data1codes.txt and Data1table.txt is provided respectively.

After verification of the Huffman decoder on MATALB, we have developed code in VHDL and simulated by using ModelSim v6.4 and actual data in waveforms is found in Fig. 6.

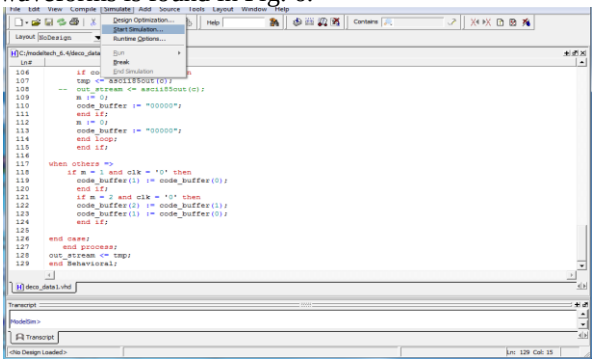

Figure 6. Modelsim for Simulation

The simulation results are shown in below Fig. 7.

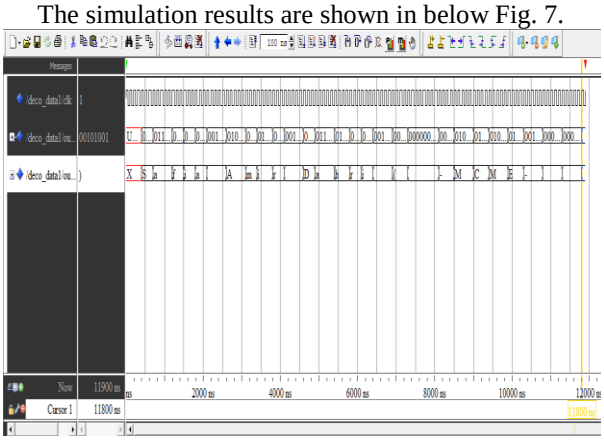

Figure 7. Results

VI. CONCLUSION

In this paper, Huffman decoding is studied and implemented on Field Programmable Gate Array (FPGA). Huffman algorithm is most widely used algorithm in data communication because of several reasons. In order to implement the Huffman decoder on hardware, we need hardware platform. The hardware platform needed, to make it possible, to configure the FPGA device. For that purpose, Xilinx ISE 8.2i tool is used. The programming is done in VHDL language and whole algorithm is described in that language, then in order to compile the code synthesis process is needed. Design implementations and bit file generation, logic, are ready to implement in FPGA. In order to verify the operation of Huffman algorithm, some data is selected and encoded by Huffman algorithm and those codes are placed in the VHDL code file and run the algorithm on hardware and verify that the results are being decoded properly. To visualize the output properly, the design file is also simulated in ModelSim v6.4. To validate the design, the same logic is performed on MATLAB as well.

REFERENCES

[1] J.S.Chitode. (2009). *Information coding techni ques.* Pune: Technical publications

[2] Azevedo Dias, W.R.; Moreno, E.D.; Nattan Palmeira, I. (2013), "A new code compression algorithm and its decompressor in FPGA-based hardware," *Integrated Circuits and Systems Design (SBCCI), 2013 26th Symposium*, vol., no., pp.1-6.

[3] Tanaka, Hatsukazu, (1987) "Data structure of Huffman codes and its application to efficient encoding and decoding (Corresp.)," *Information Theory, IEEE Transactions* , vol.33, no.1, pp.154-156.

[4] Aspar, Z.; Yusof, Z.M.; Suleiman, Ishak, (2000) "Parallel Huffman decoder with an optimized look up table option on FPGA," *TENCON 2000. Proceedings* , vol.1, no., pp.73,76 vol.1.

[5] Chen Ying; Pengwei Hao, (2004) "Integer reversible transformation to make JPEG lossless," *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference*, vol.1, no., pp.835,838 vol.1.

[6] Fryza, T.; Prokopec, J. (2009), "Experimental application of channel coding for wireless transmission of compressed video data," *Ultra Modern Telecommunications & Workshops, 2009. ICUMT '09. International Conference*, vol., no., pp.1-4

[7] Hoang-Anh Pham; Van-Hieu Bui; Anh-Vu Dinh-Duc, (2010) "An Adaptive Huffman

Decoding Algorithm for MP3 Decoder," *Electronic Design, Test and Application, 2010. DELTA '10. Fifth IEEE International Symposium*, vol., no., pp.153-157.

[8] Ke Zhu; Wei Dong Liu; Jiang Du, (2012) "Hardware JPEG Decoder and Efficient Post-Processing Functions for Embedded Application," *Computer and Information Technology (CIT), 2012 IEEE 12th International Conference* , vol., no., pp.814-817.

[9] Baumann, C. (2010). Field Programmable Gate Array (FPGA). *Embedded System Architecture.*

[10] Gutiérrez, F. M. (2009). *Implementation of a Tx/Rx OFDM System in a FPGA.*

[11] Wolf, W. (2004). *FPGA Based System Design.* USA: Prentice Hall Professional Technical Reference Pearson Education Inc.

[12] Nasri Sulaiman, Z. A. (2009). Design and Implementation of FPGA-Based Systems - A Review. *Australian Journal of Basic and Applied Sciences* , 3575-3596.

[13] Ashenden, P. J. (1990). *VHDL cookbook.*