Lempel - Ziv Implementation for a Compression System Model with Sliding Window Buffer

Ahmad AbdulQadir AlRababah
Faculty of Computing and Information Technology in Rabigh, King Abdulaziz University,
Rabigh 21911, Kingdom of Saudi Arabia

***Abstract*—Proposed compression system architecture based on Lempel-Ziv algorithm with sliding window history buffer, this architecture may be realized on FPGA, and can provide input data streams from multiple sources and context switching. Base requirements to impression system and compression system architecture were proposed. Compression system architecture should provide quick reconstruction possibility for building another system with other technical characteristics and architecture features (like reconfigurable system architecture features) on given architecture base. Digital signal processing may comprise lined or non-lined procedures. Non-lined signal processing is strictly associated with no lined structure sympathy then can apply in this period, regularity, and patio-temporal fields.**

Keywords*—Digital signal processing; FPGA; RAM; dual-port RAM; token; literal

I. Introduction

Information technology penetration into various areas of life and industry provokes avalanche-like increasing information sites, which should be processed, saved and transported [1,4]. In addition to this increasing user's number of metropolitan, and wide area networks causes necessity of economical and more effective usage of data transmission for providing quality service, to all users [2,6,7]. While data processing algorithms, which perform some functions, are application specific, data transmission and saving algorithms must be universal for providing the ability to use them in various areas [5,12,14].

Compression systems used for more effective using of saving and transmission system resources [3]. Lossless compression systems, which provide restoring compressed information without distortion, should be distinguished. Arithmetic encoding algorithms various modifications of Huffman encoding algorithms and Lempel-Ziv compression algorithms are examples of Universal Lossless compression algorithms [8,11]. Basic demands to compression systems, compressor and decompressor architecture, based on Lempel-Ziv algorithms with sliding window history buffer, are considered in this work [12,14].

II. Basic Demands to Compression Systems

The main task of compression systems is to decrease information volume, which should be transferred or saved, typical demands are used for most of them, although, various application areas can require they won specific demands. We can distinguish such basic demands to compression systems:

- Providing high compression ratio for various data types. This parameter is algorithm dependent.
- Minimization of information volume was increasing (efficiency loss) when an unsuccessful condition for giving algorithm is met. Some compression algorithms can increase information volume during processing input data without or with low redundancy. Efficiency loss minimizations possibility is dependent from given algorithm and compression system architecture. For example, we can use input data stream analyzer, which can switch to another compression algorithm or turn off compression (bypass input stream to output without compression), when efficiency loss take place on given data stream.
- Providing high speed of information compression. Implementation of this demand depends on the compression algorithm and selected realization methods. For example, some algorithms perform preceding analysis before the start of compression process (two-pass algorithms). Data compression speed depends of selected realization methods (such as ASIC, FPGA, DSP, universal normal CPU, etc.), clock rate and other device specific parameters too.
- Providing possibility to process several independent input stream (multi-channel devices) with high-speed switching between them (context switching). This demand is very useful for network systems, where multiple input transmission channels multiplexed into one or several output channels.
- Providing a suitable interface for connecting compression systems with other systems.
- Universality providing. Compression system architecture should designed for using in different application areas without changing it for each new application.

III. Compressor Architecture

Compression processor architecture, based on random access memory proposed in this chapter. Architecture, based on register memory, when history buffer realized as big shift register with comparison in all cells at a time is more effective but needs to design new chip with complex signalization scheme from each cell. This architecture cannot be realized on FPGA due significant hardware requirements.

IV. FIELD PROGRAMMABLE GATE ARRAY (FPGA)

FPGAs are programmable semiconductor campaigns that are established about an environment of Configurable Logic Blocks (CLBs) linked over programmable intersects. As per contrasting to Application Specific Integrated Circuits (ASICs), wherever the method is habit constructed aimed at the precise strategy, FPGAs can be automated to the favorite solicitation or functionality necessities. While One-Time Programmable (OTP) FPGAs are vacant, the governing kind are SRAM-based which can be reprogrammed as the project progresses.

FPGAs permit designers to modification their projects exact late in the project phase– unfluctuating subsequently the completion product has been mass-produced and installed in the field. Totaling, Xilinx FPGAs tolerate for field improvements to remain finalized tenuously, rejecting the charges allied with re-designing or manually apprising automatic methods.

An **ASIC** (application-specific integrated circuit) is a chip intended for a superior solicitation, such equally a specific type of diffusion procedure or a hand-held processer. You might compare it with universal combined circuits, such as the microchip and the random access memory chips in your PC. ASICs are used in a wide-range of submissions, as well as auto emanation controller, conservational monitoring, and personal digital assistants (PDAs). An ASIC may be pre-contrived for a distinct solicitation or it can be norm contrived (classically by modules from a "building block" archive of modules) for a specific client presentation.

Digital signal processing (DSP)

Digital signal processing (DSP) is the calculated employment of an evidence signal to transform or progress it in certain tactic. It is categorized by the depiction of detached period, separate occurrence, or other distinct dominion indications by a series of records or ciphers and the handling of these indications.

The aim of DSP is typically to ration, sieve and/or poultice incessant actual similarity signs. Commonly, the first stage is translation of the indication from an analog to a digital form, by selection and then digitizing it by an analog-to-digital converter (ADC), which cracks the analog signal into a torrent of separate numerical standards. Frequently, still, the requisite production sign is likewise analog, which necessitates a digital-to-analog converter (DAC). Smooth if this procedure is extra composite than analog handling and has a detached rate choice, the solicitation of computational control to sign handling consents for various benefits completed analog handling in numerous presentations, such as fault discovery and modification in communication as fine as data compression.

Digital and analog signal processing are subparts of signal processing. DSP submissions embrace audial and dialog sign handling, sonar and detector signal processing, sensor collection processing, phantom approximation, arithmetical sign processing, cardinal image processing, signal handling for infrastructures, controller of systems, amongst others. DSP procedures have elongated stayed route on regular processers, as well as on specific computers named alphanumeric signal processors, and on purpose-constructed hardware such as application-specific integrated circuit (ASICs). Presently, nearby are supplementary machineries used for digital signal processing counting extra controlling universal determination computer chip, field-programmable gate arrays (FPGAs), alphanumeric gesture regulators , and tributary mainframes, among others.

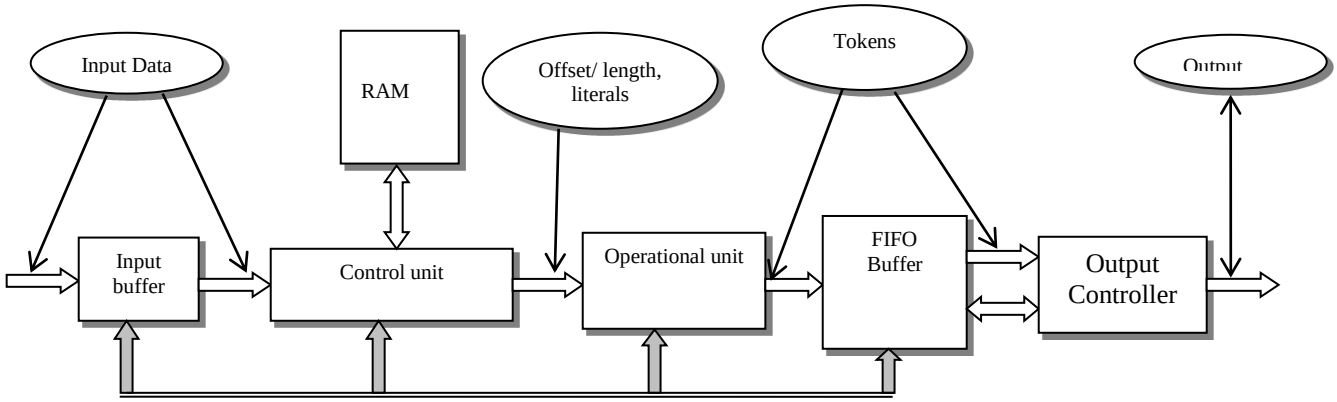

Fig. 1. Compressor Architecture

Compressor consists of the following units:

- Input buffer;
- RAM;
- Control unit;
- FIFO buffer;
- Output controller.

Input buffer used for buffering input data stream. Control unit intends to input data stream analysis, redundancy searching in this stream, token generation (literals and offset/length pair) and controlling other compressor units. Control unit is a main block in the system and consider as a more complex unit. Compressor productivity depends of

efficiency of control unit and RAM cooperation. Operational unit used for literals and offset/length pair's modification and token generation according to algorithm specification. Operational unit controlled by a control unit. FIFO buffer devoted for tokes accumulation when output controller cannot produce output stream on the fly or when external device-destination of compressed information, is slower than compressor. Output controller used for forming compressed output data stream from tokens sequence. Output data stream may be either parallel or serial, so output controller form required output stream from tokens with variable length.

The main advantage of this architecture is simplicity and commonality of used component parts, what allow to build system on already created and tested components, reduce cost and time designing process. In addition, this architecture can be realized on FPGA, can provide input data streams form multiple sources and context switching, although, it needs additional memory for all possible sources and complex control unit.

One more of the main disadvantages of this architecture is significant duration of compressing process. This caused by consecutive nature of buffer memory, what provokes necessity of scanning whole buffer, for performing comparison, for each data element from input stream. It needed to complicate control unit and using, additional memory for storing intermediate comparison results.

V. DECOMPRESSOR ARCHITETURE

Decompressor consists of the following units:

- Input controller
- Operational unit
- Token buffer
- Main controller
- Circular buffer
- Dual-port RAM
- Control logic

Input controller is used for passing compressed input data stream and separate it into tokens. Tokens, in this algorithm, are independent parts of compressed information, which can belong to one of the two types: literal or offset/length token pair. Literal is modified (modification is algorithm dependent) part of information, which was not compressed. Offset/length pair is compressed part of information where offset is offset from start of history buffer where replication begins and length-length of replication part. Operational unit is used for tokens restoring from modifications, which can took place during compression process. Token buffer is used for storing tokens when output stream is much bigger than input stream, what can occur during processing compressed information with high compression ratio. Main controller is devoted for controlling all other units. Main controller takes tokens from token buffer and sends them to circular buffer, where, dependent from token type, some operations are fulfilled. Circular buffer (history buffer) is used for storing previously decompressed information and using it in next stages of decompression process. Circular buffer consists of Dual-port RAM unit and Control logic unit. Control logic is used for building circular buffer from Dual-port RAM and for tokens processing.

Proposed architecture is flexible and can be adapted for different variants of decompression algorithm. Most of changes will be applied to Input and Operational unit. This architecture can be used in multi-channel mode with context switch ing. Decompressor needs to have Token buffer and Dual-port RAM unit for each stream to achieve this,goal.

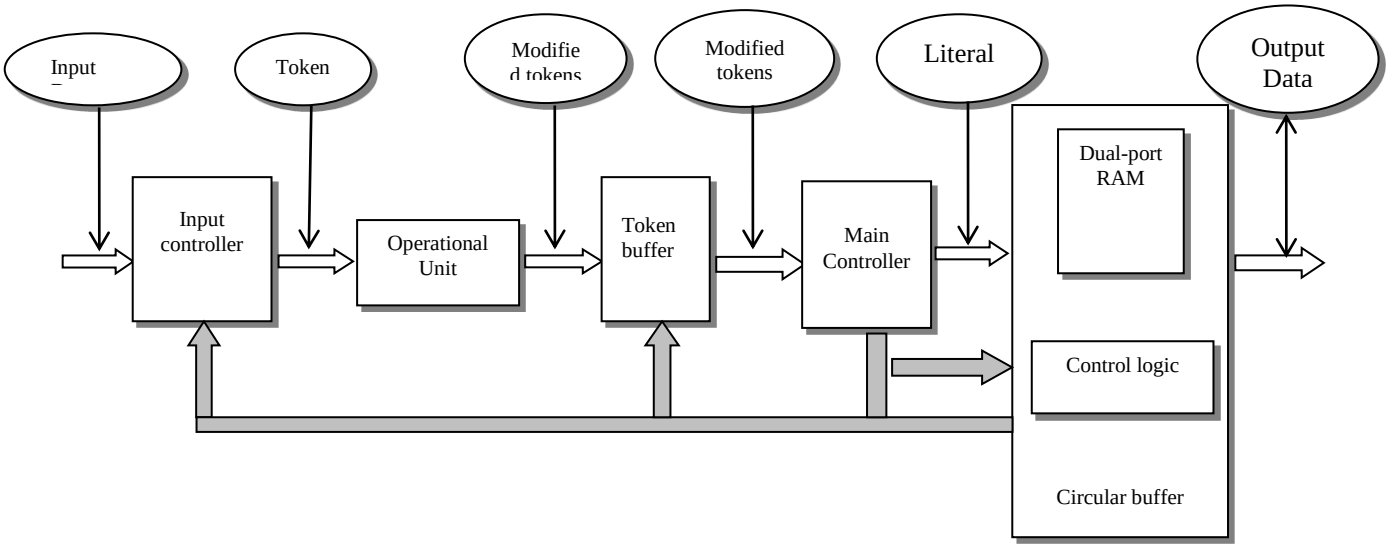

Fig. 2. Decompressor Architecture

V. CONCLUSION

Proposed compression system architecture based on Lempel-Ziv algorithm with sliding window history buffer, corresponds for more demands to compression systems and is an efficient solution for modern digital systems. One of the most important advantages of proposed architecture is the possibility to realize it in FPGA, that simplifies testing, reduces designing time and time for putting into operation.

REFERENCES

[1] Khashman A., Dimililer K., (2008)," Image Compression Using Neural Networks And Haar Wavelet", Wseas Transactions On Signal Processing, Issn: 1790-5052, Issue 5, Vol. 4, Pp. 330-339, May.

[2] Sahoolizadeh H., Abolfazl A.,(2009)," Adaptive Image Compression Using Neural Networks", 5th International Conference: Sciences Of Electronics,Technologies Of Information And Telecommunications, March 22-26.

[3] Yi Xing Z., Zhang Y., Hou Y., Jia L., (2007), " On Generating Fuzzy Systems Based On Pareto Multi-Objective Cooperative Coevolutionary Algorithm", International Journal Of Control, Automation, And Systems, Vol. 5, No. 4, Pp 444-445.

[4] Hasanzadeh R., Moradi H., Sadeghi H.,(2005)," Fuzzy Clustering To The Detection Of Defects From Nondestructive Testing", International Conference: Sciences Of Electronic, Technologies Of Information And Telecommunications, March 27-31, Tunisia.

[5] Nahm, U.Y. and Mooney, R.J. (2008) "Text mining with information extraction." *Proc AAAI-2009 Spring Symposium on Mining Answers from Texts and Knowledge Bases*. Stanford, CA.

[6] Azizah Jaafar. "Information Retrieval Technology", Springer, 2014.

[7] Thomas M. "Elements of Information Theory",2nd Ed. Willey-Interscience, 2006.

[8] David J. C. Mackay, "Information Theory, Inference and Learning Algorithms", Cambridge University Press, 2003.

[9] John G. Prookis, "Digital Signal Processing" 4th Edition, Prentice Hall, 2006.

[10] Alan V. Oppenhein, "Discrete-Time Signal Processing" 3rd Edition, Prentice Hall, 2009.

[11] Dimitris G. Monolakis, "Applied Digital Signal Processing: Theory and Practice", Cambridge University Press, 2011.

[12] Nikola Stosic, "Screw Compressors: Mathematical Modelling and Performance Calculation", Springer, 2010.

[13] Laung-Terng Wang, "System-On-Chip Test architectures: Nanometer Design for testability (Systems on Silicom)", Morgan Kaufman, 2007.

[14] [14] Daniel Glover, "Compressing Subbanded image data with Lempel-Ziv-based coders", NASA,1993.