JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Loss Less Compression And Decompression

Nalli Rojamani, M. Tech, Research Scholar, Department of ECE, International School Of Technology and Sciences (Women), Affiliated to JNTUK, NH-16, East Gonagudem, Rajanagaram, Rajamahendravaram, AndhraPradesh, 533294, India.
Miss.B.Mamatha, M. Tech, Assistant Professor, Department of ECE, International School Of Technology and Sciences (Women), Affiliated to JNTUK, NH-16, East Gonagudem, Rajanagaram, Rajamahendravaram , Andhra Pradesh, 533294, India.

ABSTRACT

The compression is the process that is required either to reduce the volume of information.The compressed data can be stored on some storage media like a floppy disk or compact disk.the compression is important because it helps to reduce the consumption of expensive resources such as disk space or connection bandwidth.data compression is the process of encoding information using fewer bits or information units.the compression only works both the sender and receiver of the information message.The compressed data has to be decompressed before use the decompression technique is just the reverse of compression.
Keywords- statistical methods,dictionary methods, Xmatchpro,CAM comparator, data compression.

I. INTRODUCTION

With the increase in silicon densities,it is becoming feasible for compression systems tobe implemented in chip.A system with distributed memory architecture is based on having data compression and decompression Engines working independently on different data at the same time.This data is Stored in memory distributed to each processor.The objective of the project is to design a lossless data compression system which operates in high-speed to achieve high compression rate.By using the architecture of compressors,the data compression rates are Significantly improved.Also inherent scalability of architecture is possible.The main parts of the system are the Xmatchpro based data compressor and the control blocks providing control signals for the data compressors,allowing Appropriate control of the routing of data into and from the system.Each Data compressor can process four bytes of data into and from a block of data in every clock cycle. The data entering the system needs to be clocked in at a rate of 4bytes in every clock cycle. This is to ensure that adequate data is present for all compressors to process rather than being in an idlestate.

II. XMATCHPRO BASED SYSTEM

The Lossless data compression system is a derivative of the XMatchPro Algorithm which originates from previous research of the authors [15] and advances in FPGA technology. The flexibility provided by using this technology is of great interest since the chip can be adapted to the requirements of a particular application easily. The drawbacks of some of the previous methods are overcome by using the XmatchPro algorithm in design. The objective is then to obtain better compression ratios and still maintain a high throughput so that the compression/decompression processes do not slow the original system down.

III. PROPOSED METHOD

The authors gives the basic idea about how the DataCompression is carried out using the Lempel-Ziv Algorithm and how it could be altered for Parallelism of the algorithm. The author describes the Lempel-Ziv algorithm as a veryefficient universal data compression technique, based upon an incremental parsing technique, which maintains code books of parsed phrases at the transmitter and at the receiver. An important feature of the algorithm is that it is not necessary to determine a model of thesource, which generates the data. According to the author, in an attempt to increase the speed of thealgorithm on general-purpose processors, the algorithm has been parallelized to run on two processors.

IV. CONTENT ADDRESSABLE MEMORY

The compression architecture is based around a block of CAM to realize the dictionary. This is necessary since the search operation must be done in parallel in all the entries in the dictionary to allow high and data-independent throughput.The number of bits in a CAM word is usually large, with existing implementations ranging from 36 to 144 bits. A typical CAM employs a table size ranging between a few hundred entries to 32K entries, corresponding to an address space ranging from 7 bits to 15 bits. The length of the CAM varies with three possible values of 16, 32 or64 tuples trading complexity for compression.The input to the system is the *search word* that is broadcast onto the *searchlines* to the tableof stored data. Each stored word has a *matchline* that indicates whether the search word and stored word are identical (the match case) or are different (a mismatch case, or miss). The matchlines are fed to an encoder that generates a binary *match location* corresponding to the matchline that is in the match state. An encoder is used in systems where only a single match is expected.The overall function of a CAM is to take a search word and return the matching memory location.

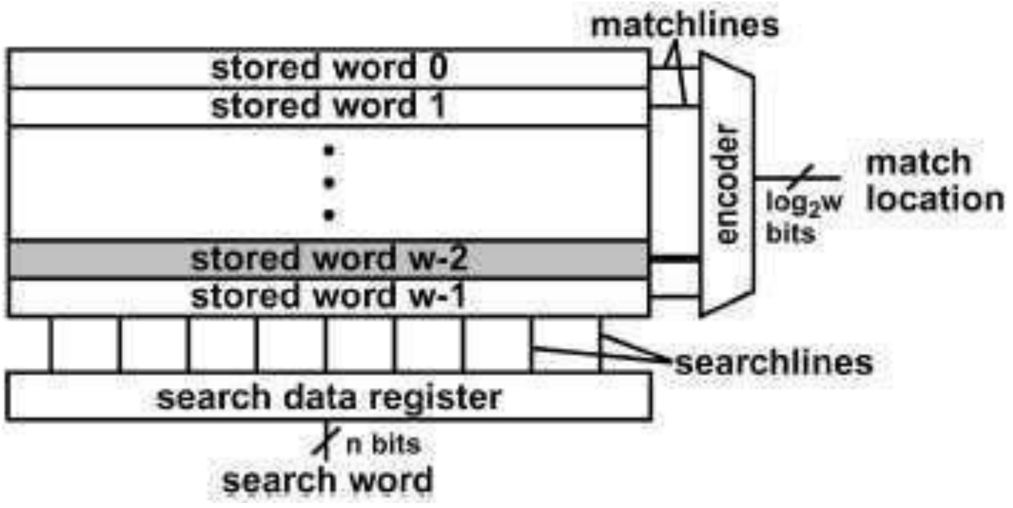


FIG. CONTENT ADDRESSABLE MEMORY

V. DESIGN METHODOLOGY

The XMatchPro algorithm is efficient at compressing the small blocks of data necessary with cache and page based memory hierarchies found in computer systems. It is suitable for high performance hardware implementation.The XMatchPro hardware achieves a throughput 2-3 times greater than other high-performance hardware implementation.The core component of the system is the XMatchPro based Compression / Decompression system. The XMatchPro is a high-speed lossless dictionary based data compressor. The XMatchPro algorithm works by taking an incoming four-byte tuple ofdata and attempting to match fully or partially match the tuple with the past data.

VI. BLOCK DIAGRAM OF 32-BIT COMPRESSION SYSTEM

The block diagram gives the details about the components of a single 32 bit Compressor. There are three components namely, COMPARATOR, ARRAY, CAMCOMPARATOR. The comparator is used to compare two 32-bit data and to set or reset the output bit as 1 for equal and 0 for unequal. The CAM COMPARATOR searches the CAM dictionary entries for a full match of the input data given.If a full match occurs, the match-hit signal is generated and the corresponding match location is given as output by the CAM Comparator.. If no full match occurs, the corresponding data that is given as input at the given time is got as output.Array is of length of 64X32 bit locations. This is used to store the unmatched incoming data and when a new data comes, the incoming data is with all compared the data stored in this array.

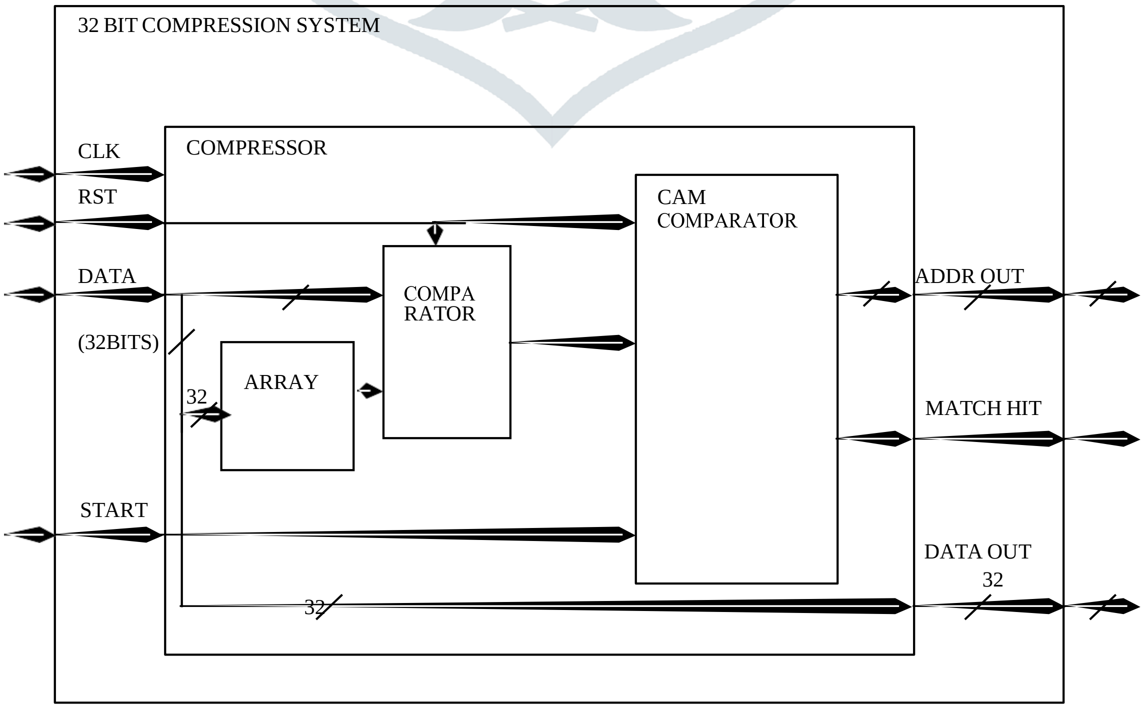


FIG. BLOCK DIAGRAM OF 32-BIT COMPRESSION SYSTEM

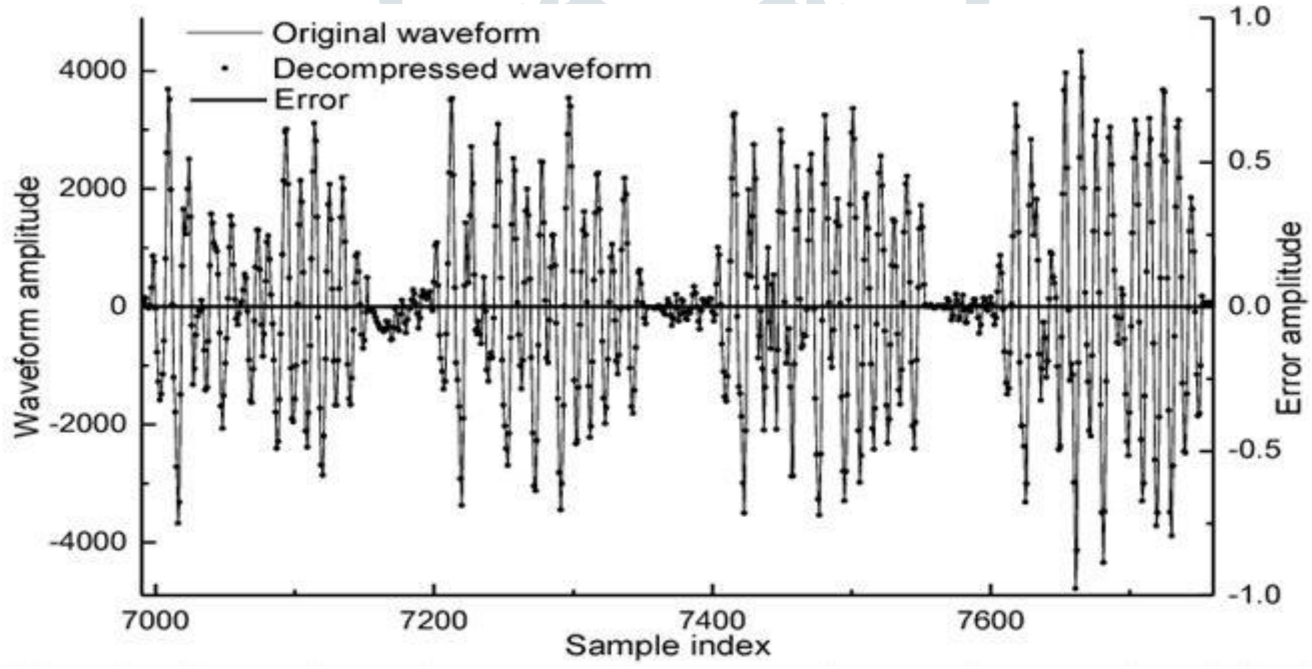
VII.INTODUCTION OF VLSI

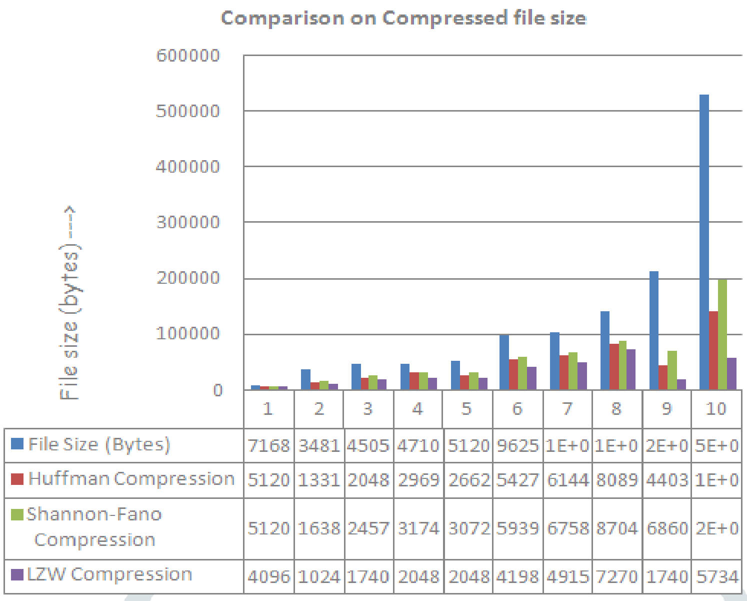
Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistor-based circuits into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device. The term is no longer as common as it once was, as chips have increased in complexity into the hundreds of millions of transistors.The first semiconductor chips held one transistor each. Subsequent advances added more and more transistors, and, as a consequence, more individual functions or systems were integrated over time. The first integrated circuits held only a few devices, perhaps as many as ten diodes, transistors, resistors and capacitors, making it possible to fabricate one or more logic gates on a single device. Now known retrospectively as "small-scale integration" (SSI), improvements in technique led to devices with hundreds of logic gates, known as large-scale integration (LSI), i.e. systems with at least a thousand logic gates. Current technology has moved far past this mark and today's microprocessors have many millions of gates and hundreds of millions of individual transistors.

APPLICATIONS OF VLSI

- Electronic system in cars.
- Digital electronics control VCRs
- Transaction processing system, ATM
- Personal computers and Workstations

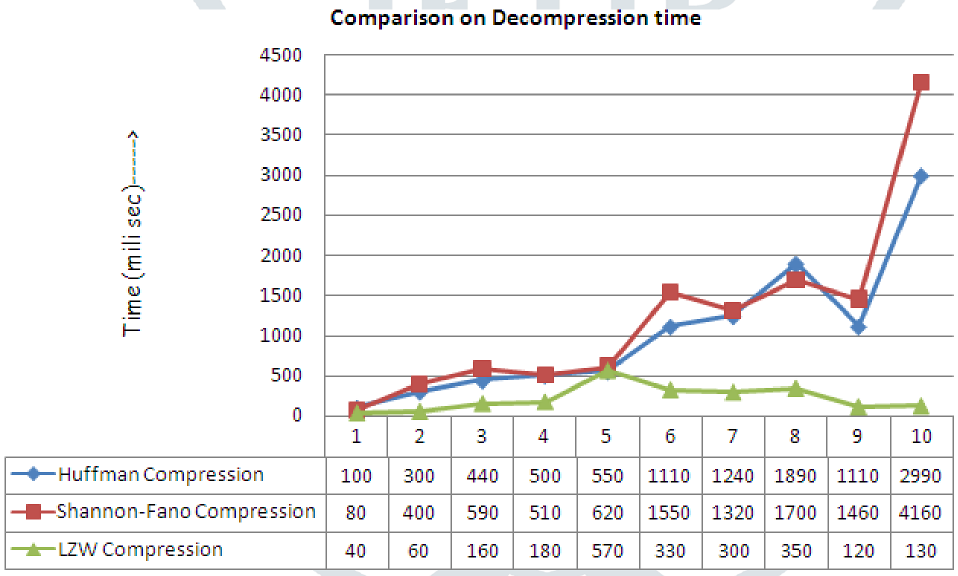
OUTPUT WAVEFORMS



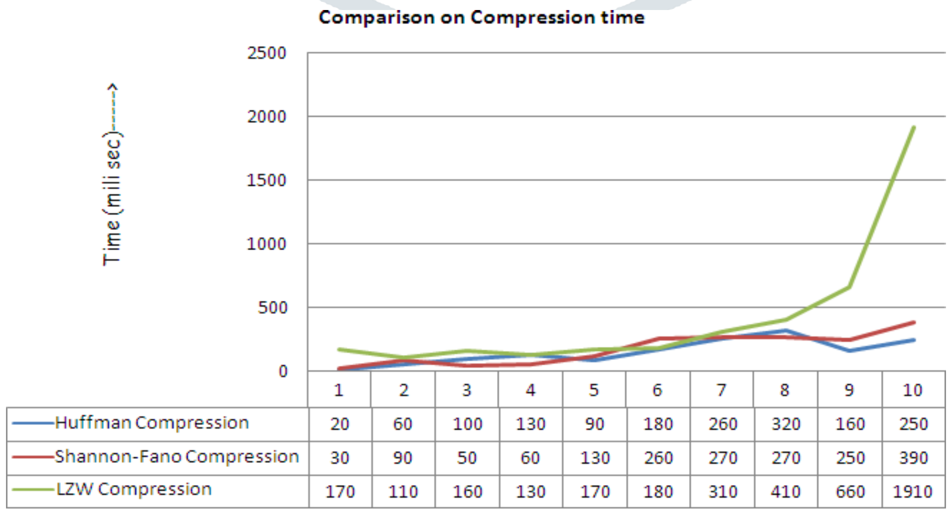


	1	2	3	4	5	6	7	8	9	10
🟦 File Size (Bytes)	7168	3481	4505	4710	5120	9625	1E+0	1E+0	2E+0	5E+0
🟥 Huffman Compression	5120	1331	2048	2969	2662	5427	6144	8089	4403	1E+0
🟩 Shannon-Fano Compression	5120	1638	2457	3174	3072	5939	6758	8704	6860	2E+0
🟪 LZW Compression	4096	1024	1740	2048	2048	4198	4915	7270	1740	5734

FIG. COMPRESSED FILE SIZE



	1	2	3	4	5	6	7	8	9	10
Huffman Compression	100	300	440	500	550	1110	1240	1890	1110	2990
Shannon-Fano Compression	80	400	590	510	620	1550	1320	1700	1460	4160
LZW Compression	40	60	160	180	570	330	300	350	120	130



	1	2	3	4	5	6	7	8	9	10
Huffman Compression	20	60	100	130	90	180	260	320	160	250
Shannon-Fano Compression	30	90	50	60	130	260	270	270	250	390
LZW Compression	170	110	160	130	170	180	310	410	660	1910

APPLICATIONS

- Compressed data is read/written faster than original data.
- It is byte order independent.
- It is variable dynamic range.

ADVANTAGES

- It saves storage space.
- It saves transmission time.
- It produce very good quality

CONCLUSION

The various modules are designed and coded using VHDL. The source codes aresimulated and the various wave-forms are obtained for all the modules. Since the Compression/Decompression system uses XMatchPro algorithm, speed of compression throughput is high.The total time required to transmit compressed data is less than that of transmitting uncompressed data. This can lead to a performance benefit, as the bandwidthof a link appears greater when transmitting compressed data and hence more data can be transmitted in a given amount of time .There is a potential of doubling the performance of storage / communication system by increasing the available transmission bandwidth and data capacity with minimum investment. It can be applied in Computer systems, High performance storage devices.

REFERENCES

[1] Pu, I.M., 2006, Fundamental Data Compression, Elsevier, Britain.
[2] Kaufman, K. and T. Shmuel, 2005. Semi-lossless text compression, Intl. J. Foundations of Computer Sci., 16: 1167-1178.
[3] Kesheng, W., J. Otoo and S. Arie, 2006. Optimizing bitmap indices with efficient compression, ACM Trans. Data base Systems, 31: 1-38.
[4] Introduction to Data Compression, Khalid Sayood, Ed Fox(Editor), March 2
[5] Shannon, C.E. (July 1948). "A Mathematical Theory of Communication". Bell System Technical Journal 27: 379–423.
[6] Fano, R.M. (1949). "The transmission of information". Technical Report No. 65 (Cambridge (Mass.), USA: Research Laboratory of Electronics at MIT).
[7] Terry Welch, "A Technique for High-Performance Data Compression", IEEE Computer, June 1984, p. 8–19.
[8] Jacob Ziv and Abraham Lempel; Compression of Individual Sequences Via Variable-Rate Coding, IEEE Transactions on Information Theory, September 1978.
[9] Improved word aligned binary compression for text indexing, Vo Ngoc and M. Alistair, 2006. IEEE Trans. Knowledge & Data Engineering, 18: 857-861.
[10] Data compression using dynamic Markov modeling, Cormak, V. and S. Horspool, 1987. Comput. J., 30: 541–550.
[11] Bounds on the redundancy of Huffman codes, Capocelli, M., R. Giancarlo and J. Taneja, 1986. IEEE Trans. Inf. Theory, 32:854857.
[12] Data compression for estimation of the physical parameters of stable and unstable linear systems, Gawthrop, J. and W. Liuping, 2005. Automatica, 41: 1313-1321.