Research Report

DATA COMPRESSION DEVICE BASED ON MODIFIED LZ4 ALGORITHM

Prepared by: **MUHAMMAD FAIZAN (飞赞)**

Roll Number: **M202161026**

School of Computer and Communication Engineering

University of Science and Technology Beijing

Submitted to: **Du Liping (杜利平)**

1st November 2021

Summary

This report discusses the data compression device based on real-time FPGA kit implementation of modified LZ4 algorithm, its technological advancements, background problems and its role in overcoming them. A review of the available literature provides insights into the modifications in LZ4 algorithm for hardware implementation, comparison between proposed architectures and improvements in I/O latency, compression ratio and performance as compared to the previous LZ algorithms.

Key findings include:

- Software implementation of data compression reduces performance significantly so proposed hardware implementation has increased throughput to 1.92Gbps with compression ratio of 2.05, higher of all times.
- Nowadays, high-end SSDs require throughput speed of 1.6Gbps, but current designs don't meet such criteria. Proposed model can be implemented in higher-end SSDs to increase their storage performance and lifetime.
- Hash computation is improved for compression ratio and low output latency is achieved.
- No hash calculation during backward matching, hash conflict, limited data input and uncertain output delay problem in LZ4 hardware implementation has been resolved in MLZ4 hardware implementation.
- MLZ4-1 and MLZ4-2 architectures are proposed but performance of second architecture is 20% in compression and 47% in decompression is higher from all previous designs due to optimized pipeline architecture.

The information presented in this report is mainly gathered from "Data Compression Device Based on Modified LZ4 Algorithm" research paper published in IEEE Journal in 2018 and secondary sources.

The report has been prepared for submission as final term evaluation for the course of Advance Communication Coding Theory at School of Computer and Communication Engineering, University of Science and Technology Beijing.

TABLE OF CONTENTS

1. Introduction

With the evolution of NAND-flash memory from Single Level Cell (SLC) to Multiple Level Cells (MLC), number of Program Erasable Cycles is reduced which means limited lifetime. Software based compression improves lifetime but it reduces overall performance of SSDs. Nowadays, high-end SSDs require throughput of 1.6 Gbps but existing designs have limited performance. Also, it's not possible to store all text in calculating hash as output delay will be uncertain and input data will be limited by the address width of hash table. To increase lifetime and performance, amount of data written to and from SSDs should be reduced. Recent studies show that applying data compression in firmware using hardware accelerator is best approach. This report discusses the data compression device based on real-time FPGA kit implementation of modified LZ4 algorithm. Proposed hardware architecture can achieve high throughput of 1.92Gbps with compression ratio of 2.05. We will also address modifications in LZ4 algorithm, comparison between proposed architectures and previous designs and optimized I/O latency, compression ratio and performance.

1.1. Methodology

Information for this report was sourced from "Data Compression Device Based on Modified LZ4 Algorithm" research paper published in IEEE Journal in 2018 and secondary sources, all listed in the Reference List. Data from publications by the IEEE Transactions on Consumer Electronics also proved valuable on technological as well as industrial scale. This report is a comprehensive review of the available literature and provides a broad overview of the topic.

1.2. Scope of Report

In the high-end SSDs which require big data to process with high throughput and compression ratio, this report could be useful in designing such system. With the advancement in compression algorithms, this report can also be referenced in ongoing research work. Mian goal is to achieve high compression ratio and I/O latency to meet the current industrial and technical needs.

2. Findings

LZ4 algorithm is modified for real-time hardware implementation. Two hardware architectures of MLZ4 algorithm implemented on FPGA evaluation kit are proposed with both compressors and decompressors and high throughput of 1.92 Gbps with a compression ratio of 2.05 is achieved. Here are some key findings explained in detail:

2.1. MLZ4 vs LZ4 vs LZ78 vs Lz77 : Difference

LZ77 algorithm uses sliding window to examine whether input sequence is matched with previously compressed data and a pointer is used to point repeated strings from dictionary while LZ78 create dictionary of phrases and when a match with phrases occurs, the encoder will output phrase index. LZ4 algorithm performs hash computation, matching, backward matching, parameter calculation and data output operations to compress data composed of LZ4 sequences including token, literal length, offset and match length while in MLZ4 algorithm, limit is set on literal length and match length to make it suitable for hardware implementation.

2.2. LZ4 vs MLZ4 : Improvements

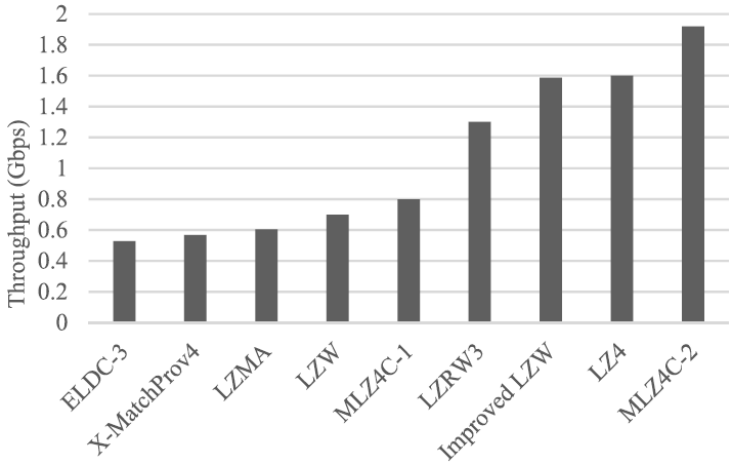
As LZ4 algorithm was software based, so there were some problems in its hardware implementation which were addressed briefly during hardware implementation of modified LZ4 algorithm. Hash calculation was only performed while matching in LZ4 algorithm but in MLZ4 algorithm, hash calculation is also performed while backward matching. More number of clock cycles were required in LZ4 algorithm in case of hash conflict while in MLZ4 algorithm, hash dictionary corresponding to hash table is included. Data stored in hash dictionary can be compared when reading the addresses. Input data was limited by the address width of hash table and so LZ4 algorithm was not able to compress data constantly while in MLZ4 valid bit is added in hash table along with data cleaning. Output delay in LZ4 algorithm was uncertain because of calculation of literal and match length was necessary before outputting data but in MLZ4, when literal length reaches 300 bytes, the backward matching can be ignored.

2.3. MLZ4-1 vs MLZ4-2 : Comparison

Both MLZ4-1 and MLZ4-2 architectures are proposed along compressors and decompressors. MLZ4C-1 runs at the frequency of 100MHz and its throughput is 0.8Gbps with compression ratio of 1.65 to 2.05 while MLZ4C-2 runs at the frequency of 240MHz and its throughput is 1.92Gbps. MLZ4D-1 runs at the frequency of 120MHz and its throughput is 0.96Gbps while MLZ4D-2 runs at frequency of 260MHz and its throughput is 2080Mbps. Throughput of MLZ4C-2 and MLZ4D-2 is 2.14 and 2.16 times of MLZ4C-1 and MLZ4D-1. This is because of optimized pipelined design of MLZ4-2.

2.4. LZ Algorithms FPGA Implementation : Performance

LZ algorithms such as LZ77, LZ78, LZW and LZ4 are previously implemented on FPGA kits in tapes, disks, SSDs and other storage devices before. All of them were software based which means reduced performance. LZ4 is the fastest algorithm till date implemented in SSDs but now LZ4 algorithm is modified to be implemented on FPGA kits. Below is the performance comparison of all LZ algorithms FPGA implementations in which MLZ4-2 outperforms.



Performance Comparison with LZ Algorithms

3. Conclusion

With the advancement in technology and growing number of internet as well as local users across the globe, there's a remarkable rise in data production and so optimized systems with high I/O latency and data compression ratio are required to manage such huge amount of data also known as big data. Different compression models were proposed time to time such as LZ4 algorithm was proposed in 2011 and it was the fastest compression algorithm till date implemented in SSDs. Nowadays, high-end SSDs with throughput of 1.6Gbps are required and to meet this demand, original LZ4 algorithm is modified for efficient hardware implementation and such changes have improved compression and decompression speeds. Proposed models can achieve 16% faster compression speeds than the required speed. The proposed MLZ4 and its hardware architectures can be used to increase storage performance as well as lifetime of high-end SSDs.

4. Recommendation

The information collected for this report provides a broad overview of key changes in LZ4 algorithm to make it work on hardware-based system and amazingly, it provided high compression and decompression output. Following recommendations could be considered in order to improve the proposed model more in future to meet the growing storage needs on low and high scale:

- Proposed model can work well on high-end SSDs to achieve desired throughput but in the recent few years, for the low-end SSDs such as used in PCs also require more optimized performance. Such model can also be implemented on low-end SSDs to further enhance their storage performance and speed.
- LZ4 algorithm is a variant of LZ77 algorithm. Although LZ77 algorithm has similar compression speed as its successor LZ78, but still decompression speed of LZ78 is quite high due to a smaller number of dictionary strings. LZ78 algorithm has already implemented on software-based models however it can be modified for hardware implementation and even high throughput can be achieved.
- Proposed MLZ4 algorithm is implemented in SSDs only but it can also be used to transmit data over communication networks more efficiently.

5. Reference List

Weiqiang Liu, Faqiang Mei, Chenghua Wang, Maire O'Neill and Earl E. Swartzlander, Jr., "Data Compression Device Based on Modified LZ4 Algorithm", IEEE Transactions on Consumer Electronics, vol. 64, no. 1, Feb. 2018.

S. Lee, J. Park, K. Fleming, Arvind, and J. Kim, "Improving Performance and Lifetime of Solid-State Drives Using Hardware-Accelerated Compression," IEEE Transactions on Consumer Electronics, vol. 57, no. 4, pp. 1732–1739, Nov. 2011.

M. Bartik, S. Ubik, and P. Kubalik, "LZ4 Compression Algorithm on FPGA," IEEE International Conference on Electronics, Circuits and Systems, Cairo, Egypt, 2015, pp. 179–182.

S. Deorowicz, "Universal Lossless Data Compression Algorithms," Ph.D. Dissertation, Faculty of Automatic Control and Computers Science, Silesian University of Technology, Gliwice, Poland, 2003.

6. Appendix

Comparison of LZ Compression and Decompression Implementations

Compression Device	Algorithms	FPGA Technology	Complexity	Clock Speed (MHz)	Troughput (Gbps)
X-MatchProv4 [7]	X-MatchPRO	180nm	5367 LUTs	50	0.567 (Compression)
LZW [8]	LZW	120nm/150nm	332 Slices 631 LUTs 247 Slices 474 LUTs	50	0.700 (Compression) 1.120~1.282 (Decompression)
ELDC-3 Core [9]	CGF, GZIP, ELIC, PNG	90nm	5900 Slices	75	0.400~0.528 (Compression)
Improved LZW Processor [10]	New LZW	90nm	3218 Slices 272 Kb RAMs	124	1.587 (Compression)
LZMA [11]	LZMA	40nm	NA	125	0.604 (Compression)
LZRW3 Core [12]	LZRW3	28nm	227 Slices 789 FFs 4~36 BRAMs	210	1.300 (Compression)
LZ4 [13]	LZ4	28nm	266 Slices 17 BRAMs 3 DSPs	200	1.600 (Compression)
MLZ4C-1	Modified LZ4	28nm	571 Slices 76.5 BRAMs	100 (Compression)	0.800 (Compression)
MLZ4D-1			365 Slices 32.5 BRAMs	120 (Decompression)	0.960 (Decompression)
MLZ4C-2	Modified LZ4	28nm	345 Slices 69 BRAMs 4 DSPs	**240 (Compression)**	**1.920 (Compression)**
MLZ4D-2			155 Slices 20 BRAMs	**260 (Decompression)**	**2.080 (Decompression)**

Comparison of LZ Implementations