HARDWARE IMPLEMENTATION OF LOSSLESS LZMA DATA COMPRESSION ALGORITHM

Parekar P. M. 1, Thakare S. S. 2
1,2 Department of Electronics and Telecommunication Engineering, Amravati University
Government College of Engineering, Amravati (M.S.), India

Abstract: Data compression is the science and skill of representing information in a compact form. Storage, transmission and processing of data are the vital component of information system. Enormous data demands additional resources hence resource optimization is required. For improving transmission speed, implementation of lossless data compression algorithm over hardware platform is an alternative. Among various lossless data compression algorithms, LZMA (Lampel Ziv Markow Chain Algorithm) is found to be better suitable for real time data compression. In this paper, hardware description of LZMA algorithm is given by VHDL and it is simulated and synthesized on modelsim and quartus II software respectively.

Keywords: Lossless Data Compression, LZ77, LZMA, VHDL.

I. INTRODUCTION

Data is a combination of alphanumeric characters which in unison called as message. Data compression is nothing but the encoding of data which takes less storage space as that of originally required. Hence, compression is the only enviable resolution for data storage and transmission application. Smaller files are desirable for data communication as it can be transferred with high speed, less power and bandwidth. There are two types of compression techniques based on the recovery of data, lossless compression and lossy compression. But as we deal with the data, lossless compression technique is more preferable. There are various methods available to achieve lossless compression namely RLE, Lossless predictive coding, Entropy coding and Arithmetic coding [1].

Field Programmable Gate Array (FPGA) technology has become a reasonable platform for the implementation of various data compression algorithm [12]. In a distributed environment, large data files remain a major bottleneck. Data compression is an only solution available for creating file sizes of manageable and transmittable dimensions without loss of any important information. When high-speed media or channels are used, high-speed data compression is desired. Software implementations are often not fast enough [11].

For real time data compression execution speed needed must be very high, but while implementing compression algorithm over computer systems speed becomes processor specific. Therefore, in real time compression, a fast compressor with low hardware complexity is required [2]. Use of LZMA algorithm reduces the maximum redundancy and able to achieve maximum compression ratio as compare to dynamic dictionary compression [3]. Here, the LZMA data compression algorithm is described in VHDL language. The individual blocks of compression algorithm is verified and analysed by using ModelSim and Quartus-II software respectively and it is implemented over the Altera DE-2 development board.

Any type of data files are needs to be converted into binary file format before applying them to the hardware. Hence, for converting any data files into binary format we need specific codec and vice versa.

Corresponding Author: Ms. Parekar P. M., Department of Electronics and Telecommunication Engineering, Amravati University
Government College of Engineering, Amravati (M.S.), India
Email Id: priyankaparekar28@gmail.com

© 2014 PISER Journal

http://piserjournal.org/
Generated binary file is given to the hardware. Hardware is configured with the proposed LZMA algorithm which will output the data in compressed and reconstructed data in binary file format.

Here, Section II describes the related technologies available. Section III describes the detailed LZMA algorithm. Section IV shows an experimental setup. Section V describes the project results and discussion. Section VI concludes the paper.

II. RELATED TECHNOLOGIES AVAILABLE

There are two techniques available for data compression namely, lossy compression and lossless compression. In lossy compression technique data loss occurs means it is impossible to recover exact original data. Such method of compression is not good for critical data, such as text file data, spreadsheet data. Lossless compression technique is free from loss of data means original message can be exactly decoded. Lossless data compression technique is ideal for text compression. By using this compression technique we achieve better compression ratio but it is less as compared to lossy technique.

There are various entropy based methods available for lossless data compressions are:

**Huffman Encoding:** Huffman coding is based on the concept of mapping an alphabet. Probabilities of symbols are calculated on fly and the symbols with a high probability of occurrence have assigned smaller representation than those that occur less often [4], [5], [6].

**LZ77:** LZ77 encoding algorithm is the first simple compression algorithm described by Lempel and Ziv in 1977. Here, the search window act as a dictionary consists of all processed input string. When new groups of symbols are being read in, the algorithm searches for matches with strings found in the previous data already read in. Then the matches are encoded as pointers and sent as the output. LZ77 encoding is simple and faster. LZ77 is effective only when the input data is highly redundant or repetitive [5], [6].

**LZ78:** LZ78 builds its dictionary with of all of the previously seen symbols in the input text rather than having a limited-size window into the preceding text. But instead of having access to all the symbol strings in the preceding text, a dictionary of strings is built a single character at a time [3].

**LZW:** LZW is a dictionary based data compression that compresses content character by character. These characters are combined to form a string. A special code is assigned to new strings and strings are added to the dictionary [7], [8], [9], [10].

**LZMA:** LZMA is the recent variant of LZ coding. The LZMA was proved to be effective for unknown byte stream compression for reliable lossless data compression. It is based on LZ77 and uses a delta filter and a sliding dictionary algorithm to achieve high compression ratio [3].
III. LAMPEL ZIV MARKOW
CHAIN ALGORITHM (LZMA)

LZMA algorithm uses delta encoder and sliding dictionary encoder along with the concept of LZ77 dynamic dictionary encoding. The output of LZMA encoder is represented in the form of tuple (offset, length, new symbol).

**Delta Encoder and Decoder**

The Delta Filter shapes the input data stream for effective compression by the sliding window. It stores or transmits data in the form of differences between sequential data. The output of the first byte delta encoding is the data stream itself. The subsequent bytes are stored as the difference between the current and its previous byte. While, the output of delta decoder keeping first data stream as it is and subsequent byte are stored as the addition between current and previous data byte. For a continuously varying real time data, delta encoding makes the sliding dictionary more efficient.

**Example of Delta Encoder**

Sample input sequence: 2, 3, 4, 6, 7, 9, 8, 7, 5, 3, 4
Output sequence encoded: 2, 1, 1, 2, 1, 2,-1,-1,-2, 1
Number of symbols in input: 8
Number of symbols in output: 4

**Example of Delta Decoder**

Sample input sequence: 2, 1, 1, 2, 1, 2,-1,-1,-2, 1
Output sequence encoded: 2,3,4,6,7,9,8,7,5,3,4
Number of symbols in input: 4
Number of symbols in output: 8

**Sliding Dictionary Algorithm**

There are two types of dictionaries namely static dictionary and adaptive dictionary. In static dictionaries the entries are predefined and constant according to the application of the text whereas in adaptive dictionaries, the entries are taken from the text itself and created on-the-fly. A search buffer is used as dictionary and the sizes of these buffers depend on the size of data set and parameters of the implementation. Patterns in text are assumed to occur within range of the search buffer. The offset and length is encoded separately, and a bit-mask is also encoded. Use of suitable data structure for the buffers will reduce the search time for longest matches. Sliding Dictionary encoding is more difficult than decoding as it needs to find the longest match.

**Fig. 2: Encoder and Decoder Block of LZMA Algorithm**

**Example of Sliding Dictionary Encoder**

<table>
<thead>
<tr>
<th>Input Data</th>
<th>Compressed Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>010001</td>
<td>0101</td>
</tr>
</tbody>
</table>

**Fig. 3: Example of sliding dictionary Encoder**

**Fig. 4: Example of Sliding Dictionary Decoder**

<table>
<thead>
<tr>
<th>input</th>
<th>compressed data</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0,0,a)</td>
<td>(0,0,b)</td>
</tr>
<tr>
<td>(0,0,b)</td>
<td>(0,0,r)</td>
</tr>
<tr>
<td>(0,r)</td>
<td>(3,1,c)</td>
</tr>
<tr>
<td>(3,1,c)</td>
<td>(2,1,d)</td>
</tr>
<tr>
<td>(2,1,d)</td>
<td>(7,4,d)</td>
</tr>
</tbody>
</table>

© 2014 PISER Journal  
http://piserjournal.org/
Experimental Setup
The experimental setup for the proposed hardware architecture for LZMA is shown in Figure 5. The proposed architecture is tested on Altera DE-2 development board with Quartus II 9.1i. The real data input is given from PC through serial communication to the FPGA development board. The FPGA is programmed using USB programmer. The compressed and decompressed data are displayed on LCD.

IV. RESULT DISCUSSION
The HDL synthesis Summary and the device utilization details for encoder and decoder are given in Table 1. Here we taken input as a different string of 16 byte data size with different redundancies and calculate its compression ratios. The LZMA gives better compression ratio over a range of input data sizes and the average compression ratio in percentage achieved by using proposed algorithm is 71.57% which is comparably high as compare to the existing dynamic dictionary encoding method (60%). Therefore, the average amount of storage space saved is 10.55% which is less than 7zip algorithm implemented on software which is 12.25% for the similar dataset. But, the storage space required can be reduced if size of the dataset will increased and dataset having maximum redundancy. The compression ratio (CR) and Storage space are calculated as follows.

\[
CR = \frac{\text{Output Data Size}}{\text{Input Data Size}} \times 100
\]

\[
\text{Storage Space} = (1 - \frac{\text{Compressed Data}}{\text{uncompressed Data}}) \times 100
\]

<table>
<thead>
<tr>
<th>TABLE I: Analysis and Synthesis Results of Compression/ Decompression Module</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total logical elements</strong></td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td><strong>Compression Module</strong></td>
</tr>
<tr>
<td><strong>Decompression Module</strong></td>
</tr>
<tr>
<td><strong>Overall System</strong></td>
</tr>
</tbody>
</table>

V. CONCLUSION
Among the various lossless data compression algorithm LZMA proves to be an effective for getting higher compression ratio. Data compression optimize the resources such as power, bandwidth, memory required for transmission and storage of data. Implementation of LZMA algorithm on hardware improves the execution speed and it is compatible with the arrival of real time.
data. High performance FPGA with parallel architecture and inbuilt embedded multipliers maximizes system performance. Further, this architecture can be extended to application specific integrated circuits (ASIC) so as to design a specific hardware chip for the LZMA compression algorithm so as to get more robustness in data compression and decompression.

REFERENCES


