

An FPGA Implementation of the Fast Minimum-Redundancy Prefix Coding

Chia-Hsiung Chen

April 13, 2009

Motivation

Imagery data captured by satellite lens provides valuable information for weather forecast, geo-environmental monitoring, and oil drilling. However, efficient compression of satellite images requires a data coding scheme with the characteristics of low-complexity, high compression ratio, and lossless image quality. In search for an efficient data coding scheme for satellite images therefore becomes important and may significantly influence the efficiency and quality of the compressed data.

Project Highlight

In this project, an FPGA realization of the minimum-redundancy prefix coder [1] will be explored. Furthermore, optimization methods discussed in course will be applied to improve the performance/throughput of the hardware design.

Prior Art

In 1971, Rice [2] proposed a low-complexity, prefix-free entropy coder. As opposed to Huffman coding [3], decoder of Rice coding scheme does not require original codeword table. Although Rice coding is recommended by the Consultative Committee for Space Data Systems (CCSDS), it is not optimal in practice, and requires $O(n \log n)$ time if the input data is unsorted. Recently, a new minimum-redundancy prefix coding scheme is proposed in [1], which provides a linear-time, prefix-free coding for space data compression. Similar to the Rice coding, no codeword table is required to send to the decoder in minimum-redundancy prefix coding scheme.

Approach

1. Understand the algorithm behind the minimum-redundancy prefix coder.
2. Perform suitable transformation to the algorithm for hardware implementation. Although the original algorithm contains certain hardware thinking, it still includes annoying operations such as loops and sorting. Apply loop transformation and incorporate speedup mechanism to the loop is necessary. Sorting and reordering operations should be carefully handled or removed in order to avoid unnecessary cycle count waste.
3. Apply scheduling and pipelining techniques for further performance speedup. This step is indeed overlapped with the algorithm transform stage. Proper scheduling of

operations help hide latency. Pipelining and retiming should be performed to remove long latency path and therefore increase working frequency.

4. Verify correctness of the design by sending the input testbench to both software and hardware blocks, and make sure the output from both software and hardware should be the same.

Expected Result

A high throughput, efficient FPGA implementation of the minimum-redundancy prefix coder will be proposed. Techniques and designs of the hardware will be described in detail in the final report.

Task Planning

- 4/1~4/7 – algorithm investigation
- 4/8~4/26 – algorithm reformulation and FPGA implementation
- 4/27~5/8 – code cleanup, further optimization

References

- [1] B. Huang, "Fast Minimum-redundancy Prefix Coding for Real-Time Space Data Compression," Proc. SPIE, 6683, 66830H, 2007.
- [2] Lossless Data Compression. Recommendation for Space Data System Standards, CCSDS 121.0-B-1. Blue Book. Issue 1. Washington, D.C.: CCSDS, May 1997.
- [3] D.A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proc. Inst. Radio Eng., vol. 40, iss. 9, pp. 1098-1101, Sept. 1952.