International Journal of Reconfigurable and Embedded Systems (IJRES)
Vol. 3, No. 1, March 2014, pp. 25~30
ISSN: 2089-4864

Implementation of LOCO-I Lossless Image Compression Algorithm for Deep Space Applications

P. Praveena
Departement of Electrical and Electronics Engineering, MVJCE, Bangalore

Article Info

Article history:

Received Jan 10, 2014
Received Feb 20, 2014
Accepted Mar 1, 2014

Keyword:

Deep space communication
FPGA
LOCO-I
Lossless compression

ABSTRACT

Present emerging trend in space science applications is to explore and utilize the deep space. Image coding in deep space communications play vital role in deep space missions. Lossless image compression has been recommended for space science exploration missions to retain the quality of image. On-board memory and bandwidth requirement is reduced by image compression. Programmable logic like field programmable gate array (FPGA) offers an attractive solution for performance and flexibility required by real time image compression algorithms. The powerful feature of FPGA is parallel processing which allows the data to process quicker than microprocessor implementation. This paper elaborates on implementing low complexity lossless image compression algorithm coder on FPGA with minimum utilization of onboard resources for deep space applications.

Corresponding Author:

P. Praveena
Dept of EEE,
MVJ College of engineering,
Channasandra, Near ITPB,
Bangalore – 560067.
Email: praveenapuvvada@gmail.com

1. INTRODUCTION

The exploration and utilization of deep space are emerging trends for future remote sensing applications. Deep space relates to the outer space more than 2 million kms away from the earth. Deep space communications takes important part in deep space exploration applications. Communication between earth and other planets is termed as deep space communications. With common terra and satellite communication deep space communication presents more challenging area for data communications. The key technologies that play important role in deep space communications are image source coding, channel coding, modulation & demodulation and deep space network [1].

The storage and transmission of image data (such as images of landforms and physiography of remote planets etc) occupy large amount of onboard resources. Bandwidth required to transmit these data also increases. This conflicts the limited storage and transmission capability of deep space communication. In order to fulfill the requirement for bandwidth and storage capacity, high efficient image compression coding method is one of key technology of deep space communication.

Digital images are comprised of lot of redundant information. Thus they are compressed to remove redundancy and hence minimizing the storage space or transmission bandwidth. Lossless image compression is a process of removing redundancy, which achieves exact reconstruction of original image; otherwise it is lossy image compression [2]. As space data is rare-one and subjected to processing, information in the data has to be preserved. So, lossless compression [3] has been suggested for space applications.

Real time or near real time image processing will be challenging technology for future remote sensing satellites. Conventional computing systems are not suitable for real time proceesing because computing power required is very large. Contemporary computer architecture comes into bottlenecks like

sequential execution and large memory, which slows the computing. Any way large memory and pipelining can eliminate these bottlenecks to some extent but not completely.

Parallel processing computers can offer reasonable solution for some of these bottlenecks. But these specialized computers found fewer acceptances in space missions due to enormous problems in processing parallism of algorithms and need of high reliable systems in hostile space environment. Moreover lacking behind flexibility of changing operational requirements and more production as well as development cost. Hence for efficient processing of images in space require computing architectures with less complexity, high flexibility and cost effective.

On-board satellite compression implementation used based on the options like software, hardware or both software and hardware solution [4]. DSP or high speed CPU achieves software-based solution for complex and massive data processing. Emerging trends use hardware implementations based on ASICs and recently on FPGAs due to its higher throughput. Radiation hardended FPGAs cover the way for reconfigurable hardware in space. The fast development of reconfigurable devices in terms of density, programmable interconnections, memory arrays and communication interfaces make them more attractive choice in space applications.

In addition to good compression performance, onboard compression requires low encoder complexity and hardware friendliness [5]. Since sensors generate very high data rates, it is important that encoder must be of low complexity to operate in real time. Onboard compression algorithms are implemented on FPGA or ASIC to meet processor speed. A fast implementation of image compression algorithm is paramount importance in all space applications. Therefore algorithm design should be able to support the simple hardware implementation with use of available resources.

Coding scheme of LOCO-I outperforms few lossless image compression algorithms like FELICS [8], CALIC [9] and JPEG-LS [10] in terms of best complexity and compression ratio. This paper presents an efficient implementation of LOCO-I algorithm tailored for field programmable gate array (FPGA) for deep space communications. Implementation takes advantage of sequential nature of algorithm and exploits pipelined architecture with minimum utilization of onboard resources.

2. IMAGE COMPRESSION METHOD

Image compression method explores the concept of entropy, correlation and compression rate. Average of information content of all possible events from memory-less source is termed as entropy. The performance of the compression system is measured in terms of encoded data rate i.e. bits per sample or entropy. Image exhibits some kind of interdependency is characterized by correlation. Image compression processes basically have a decorrelator to map image information into a lower correlation and entropy coder to optimize output information entropy.

Scientific data return from space borne instruments present some characteristics that contribute to compression: low entropy caused by atmospheric effects that results concentration of signal at mid levels and strong correlation between different regions of an image. Remote sensing data are compressed and decompressed without prior knowledge of input stream statistics. This type of compression method is called universal. A Universal Coder consists of a Modeler and a Coder [6]. The function of modeler is to determine contexts, which uniquely describe the source sequence. The coder in turn encodes the sequence using statistics provided by model. The Universal coder is suitable for satellite application due to its ability of compressing data on the fly.

The Universal coders can be classified into two main classes: (i) Universal Rice Coder (URC) and (ii) Context based Compression coders. The universal rice coder [3] is very simple to implement and better suited for compression of linear data. A Context Based Compression method infers idea that the context of a pixel can be used to predict (estimate the probability of) the pixel. This method compresses the image data by scanning each and every pixel and coding it, with respect to its neighboring pixels. FELIC, CALIC, LOCO-I, JPEG-LS are some of well known lossless compression methods comes under context based compression methods.

3. LOCO-I HARDWARE IMPLEMENTATION

LOCO-I encoder outperforms the other algorithms in terms of complexity and compression ratio [7]. Hence for deep space image coding this is best choice. This section focuses on hardware implementation issues and solutions that have been derived for LOCO-I. The solutions that have been chosen to implement an efficient management of the run time memory and effective use of onboard resources.

In hardware implementation based on FPGA, algorithm is simplified by using regular mode toencode the pixel; no run length coding used because of its high complexity. Recent work explores the

feasibility of complete implementation of LOCO-I with VHDL modeling [12]. With reference to that design the amount of on-chip memory for context table and two image rows occupies about 80% of total chip area. Therefore in FPGA implementation this is slowest phase in the design. LOCO-I compression algorithm is made of limited set of operations. These operations are executed sequentially on pixel values. This suggests the need of pipelined implementation. This is possible only when all execution phases are data independent. LOCO-I holds an exception that when two immediate pixels are having same context parameters, since both hold address at same context location.

3.1. Context Memory Management

As described in [11], LOCO-I updates the context parameters after coding/decoding phases of each pixel. In a pipeline scheme if continuous pixels points different context parameter does not create a problem. Conversely if continuous pixels points same address, the later pixel extended based on outdated values. After the current pixel processing, context update is computed. In case immediate pixels point the same context then pipeline process has to stall for few cycles as a trivial solution. But this degrades the compression algorithm performance.

To address the issue without modification to the algorithm, the proposed design updates the context parameters before addressing the immediate pixel of same context memory location. Denoting 'f' as pixel frequency, the solution is based on implementation of two-block pipeline stage. Consequently t_i defines discrete time steps of pipeline. In particular, during positive half cycle of t_i first block reads the context table parameters, while during negative half cycle of t_i second block updates the context table itself, as required by current pixel processing. As a consequence, since at t_{i+1} the same context memory location addressed to process the following pixel, the values that are read from context table are up-to-date. This is graphically explained in Figure 1, showing the time evolution of described execution.

The Figure 1 explains how the pipeline functions with the help of single clock. In presence of the same context address for two continuous pixels, the read process of the context value for pixel *i+1* (denoted as $ABCD_{i+1}$), read the up-to-date value written to the context memory by pixel *i* ($uABCD_i$).

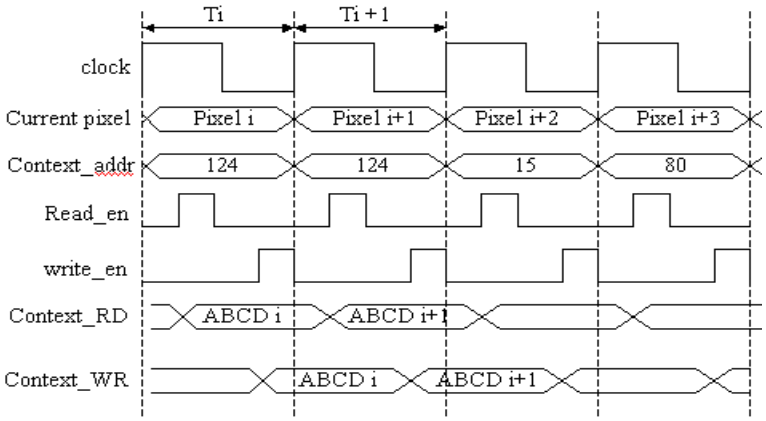

Figure 1. Time evolution context parameter: single clock architecture

3.2. Memory Management

Memory requirements for LOCO-I algorithm during runtime directly depend on context table dimensions, and the presence of the neighbouring pixels that are needed for context ascertainment and pixel decoding. As far as context table memory is concerned, its size depends on the image local gradients granularity chosen to quantize. With respect to pixel storage, the computation of current pixel requires to access the context pixels. But context pixels unfortunately belongs current and previous pixel rows.

Common LOCO-I implementations allocate memory for context table as well as for current and previous pixel rows, so as to readily access the required pixels [12]. These pixels row memories are swapped at the end of each row processing. But image row is quite large, e.g. 1024 pixel wide range or more sometimes. Then the memory requirement for current and previous pixel rows will be quite large. Therefore in implementation of LOCO-I for space applications care has been taken to reduce the memory occupation, since this represent problem in FPGA target platform [7].

Figure 2. Context of pixel P

In this point of view implementation carries simple shift register concept for the memory management and context determination, which allows minimizing the considerable memory occupation. As explained in [12], the context is made of five neighboring pixels that we name as a, b, c, d, and e and shown in above Figure 2. Two shift registers are used for current and previous row pixel values of size three times the pixel depth. For retrieving the pixel values clock of 'f' pixel frequency is used. In the positive half cycle current pixel value is read and pushed to present register (Prs_reg). Similarly in negative half cycle previous pixel value is read and pushed to the corresponding shift register previous register (Prv_reg). Therefore all the pixel values are retrieved from corresponding registers for context determination and prediction. In each clock cycle one left shift takes place in both present and previous registers, so that all neighbouring pixels are available for next pixel.

This is graphically shown in Figure 3 below. The pixel values read from memory are denoted by pixel_val. The positive half cycle value represents the current pixel i.e 'p'. The negative half cycle value represents the previous line pixel i.e. 'd'. Shift register used for present line is denoted by prs_reg and for previous line by prv_reg.

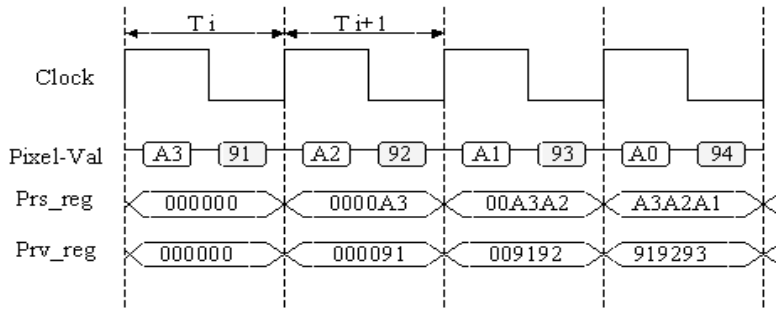

Figure 3. Neighbouring pixel determination with single clock architecture

4. OVERALL IMPLEMENTATION

The algorithm is broken into limited set of elementary operations and organized into dataflow graphs. The resulting dataflow graphs are made of combinational blocks whose cascaded propagation delays are process latency; hence the clock frequency and combinational blocks are so designed to maximize the performance. Based on the dataflow graph the RTL structure of encoder is designed. The total implementation process carried by three major mode of operations as follows:

1st Mode: Down loading image from RS232 cable to external memory.
2nd Mode: Carries compression process.
3rd Mode: Writing back compressed data stream to external memory.

VHDL model of whole design simulated its behavior and implemented it in Actel Proasic3 FPGA. Furthermore the design was tested as a compression engine for a transmission system conceived of RS232 cable and wearable computer. The various stages of compression process are shown below in Figure 4.

1st stage: Context memory initialization and generating compression start signal.
2nd stage: Retrieving current and previous pixel from memory.
3rd stage: Extracting the context and computing local gradients.
4th stage: Compute prediction and bias correction of current pixel.
5th stage: Computing context parameters and residual error values and k value.
6th stage: Generating compressed bit stream and writing to memory

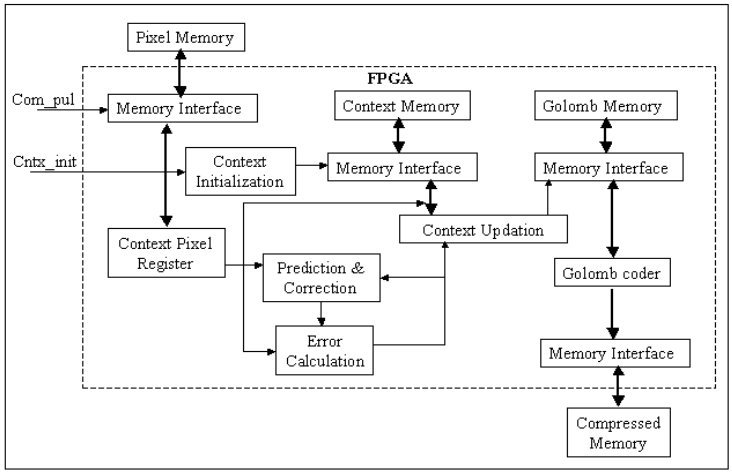

Figure 4. Compression process flow

5. TEST RESULTS

The proposed architecture was tested on Actel Proasic3 FPGA, with 24,576 logic elements and 144K RAM bits to evaluate the compression process. RS232 cable of 250Kbps feeds image data to the FPGA board, which tests in real time operation. The maximum clock frequency attained by circuit realization of encoder about 16MHz. A large number of images, ranging from standard CCSDS remote sensing imaging were used to evaluate the architecture on a wide variety of conditions.

Table 1. FPGA Resource Utilization

Logic Utilization	No. Of units Available	No. Of units Utilized	Percentage of utilization
Core cells	25,576	3740	15%
I/O Cells	300	111	37%
RAM/ROM bit blocks	32	10	31%

The compression engine developed is synthesized on single device. The FPGA synthesized compressor is able to process blocks of 512X4 Pixels, 8-bit pixel with 16MHz clock frequency and utilization of resources tabulated in Table 1 and represented in a bar chart in Figure 5.

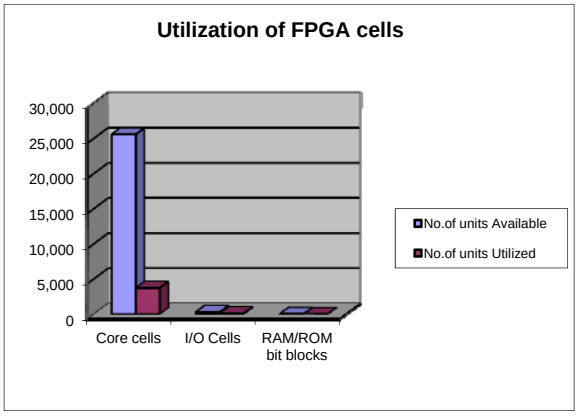

Figure 5. FPGA Utilization

Simultaneously C software implementation of LOCO-I encoder and decoder were developed to evaluate compression process. Bit-bit comparison of original and reconstructed image is evaluated. The resulting encoded data is verified continuously in reference to hardware-encoded data.

The successful implementation results, combined with the algorithm analysis, indicate the potentiality offered by the proposed solution for onboard image compression and explores the basic sense of functional and reliable end item hardware for deep space applications.

6. CONCLUSION

LOCO-I relatively efficient in terms of compression rate, low complexity and memory requirements which is very attractive for hardware implementation purpose in deep space applications. Comparing to other methods LOCO-I is best choice for lossless applications. In this paper, the design of LOCO-I compression architecture is fully compatible with standard JPEG-LS is presented. A pipelined architecture of encoder has been discussed. Simple procedure for memory determination and row memory management drastically reduced the resource occupation.

Hence the proposed architecture is qualified for FPGA implementations. Synthesis and implementation figures show design with limited resource occupation and good performance, enabling its use for image compression in deep space missions. Additional work will be carried out on packetization process to improve the fault-tolerance of compression process.

The main problem in development of space hardware for deep space applications is not the compression method itself but also the hardware in which the data will process. Hence the use of low complexity solution is spreading over the device options. FPGAs that is strategically decisive in present scenario. The proposed implementation is carried on space qualified ACTEL Proasic3 FPGA with 15% of core cells, 37% of I/O cells and 31% of RAM/ROM blocks utilization.

REFERENCES

[1] Xiao Song, Li Yunsong, Bai Baorning, Zhou Youxi. *The key Technologies of Deep Space Communications. China communications.* 2006.
[2] Ming Yang, Nikolaos Bourbakis. An Overview of lossless digital image compression techniques. *IEEE.* 2005.
[3] *Lossless Data compression*. CCSDS 120.0-G-1 Green Book. 1997
[4] Yu G, Vladimirova T, Sweeting MN. *Image Compression Systems on board satellites*. Acta Astronautica. 2009; 64: 988-1005.
[5] M Cabral, R Trautner, R Vitulli, C Monteleone. *Efficient Data Compression for spacecraft including Planetary Probes*
[6] J Rissanen, GG Langdon jr. Universal modeling and coding. *IEEE Trans. Inform. Theory.* 1981; IT-27: 12-23.
[7] Pierantonio Merlino, Antonio Abramo. A Fully Pipilined Architecture for the LOCO-I Compression Algorithm. *IEEE Transactions on very large scale integration (VLSI) Systems.* 2009; 17(7).
[8] PG Howard, JS Vitter. *Fast and efficient lossless image compression*. Proc. IEEE Int Conf Data compression, 1993: 501-510
[9] X Wu, ND Memon. *Context-based, adaptive, loss-less image coding.* IEEE Trans. Comm., 1997; 45(4): 437-444.
[10] Marcelo J Weinberger, Gadiel Seroussi, Guillermo Sapiro. *The LOCO-I Loss-less Image Compression Algorithm: Principles & Standardization into JPEG-LS.* Hewlett-Packard Laboratories, Palo Alto, CA 94304.
[11] Marcelo J Weinberger, Gadiel seroussi & Guillermo Sapiro. *LOCO-I: A Low Complexity, Context-Based, Loss-less Image Compression Algorithm*. Hewlett-Packard Laboratories, Palo Alto, CA 94304.
[12] M Klimesh, V Stanton, D Watola. *Hardware implementation of a lossless image compression algorithm using a field programmable gate array*. NASA JPL, California Inst. Technol., Pasadena. 2001

BIOGRAPHY OF AUTHORS

P. Praveena currently working as Assistant Professor in Electrical and Electronics Engineering Department at MVJ college of engineering, Bangalore. She received the M.Tech degree in Digital Electronics from VTUniversity, Belgaum in 2012 and B.Tech degree in Electrical and Electronics Engineering from JNT University, Hyderabad. Her area of interest is image processing, digital electronics, and control systems, FPGA design. She is lifetime member in ISTE, India.