Advances in Engineering Research, volume 150

4th International Conference on Machinery, Materials and Computer (MACMC 2017)

FPGA Implementation of JPEG-LS Remote Sensing Image Coding Algorithm

Hairong Wang[1, a]

[1] Network College, Haikou College of Economics, Haikou, Hainan, 571127

[a]email,

Keywords: FPGA; JPEG-LS; VHDL; Integrated Rate

Abstract. In this paper, through the research of JPEG-LS encoding algorithm, the algorithm is implemented with FPGA chip XC4VSX55-12ff1148, which is implemented on the ise14.7 development platform of Xilinx company. The algorithm adopts modular design idea，from the image data collection and storage, variable update, Golomb encoding, regular mode encoding, run length mode encoding, encoding output and storage to achieve modular, each module is described in VHDL language. Finally, the coding rate, synthesis rate and chip utilization of single image coding are obtained.

Introduction

JPEG-LS is an image coding algorithm for ISO / ITU standard for non-destructive / near lossless compression of static continuous tone image [1]. It is a low complexity image compression algorithm proposed by HP Lab [2]. The JBIG compression algorithm is mainly used for binary image, which is applied to the inside of the printing system. JPEG2000 image compression algorithm uses wavelet technology to obtain the region of interest by using its local resolution. DPCM is an early lossless compression algorithm, mainly through the prediction and differential coding to reduce the redundancy, in order to achieve the compression of the compressed data stream, wavelet computing hardware to achieve high complexity, the need to store large amounts of data, not easy to FPGA implementation; The algorithm is a context-based predictive differential coding method, and the flat region in the image is encoded by the long-distance pattern, otherwise the conventional algorithm Mode coding, compared with the above compression algorithm, JPEG-LS in the field of lossless compression with high fidelity, low complexity, easy to achieve hardware.

At home and abroad for JPEG-LS algorithm improvement and optimization in theoretical research and program implementation has been relatively mature. For the realization of the hardware, there is also a lot of progress, metaphor for the algorithm VLSL implementation [5], for image lossless compression IP core FPGA design [6], and JPEG-LS algorithm within a coding method Improved hardware implementation and so on. The design of the design using modular design ideas, from the image data acquisition and storage, variable update, Golomb coding, conventional mode coding, run mode coding, coding output and storage to achieve modular, the module using VHDL language to describe, and The algorithm is implemented on the FPGA chip XC4VSX55-12ff1148.

Realization Principle of JPEG - LS Remote Sensing Image Coding Algorithm

JPEG-LS remote sensing image coding algorithm is a very effective image compression coding algorithm. The biggest advantage of this algorithm is simple calculation, low complexity, no need for DCT transform and arithmetic coding, only differential prediction and entropy coding, high quality to restore the original image, easy to achieve hardware.

Image Sample Input and Encoding Mode Selection. In the process of coding, some samples of the image data are scanned and a "context" model is established. The current sample value is modeled by the correlation between the current sample value and its neighbors, the image sample value is input to the differential predictor, the coding mode is selected Structure shown in Figure 1. In the predictor, the local gradient is calculated by the image sample value. If the calculated

gradient value is 0 or all NEAR (near the lossless compression ratio control factor), the run length coding mode is selected. Otherwise, the normal mode code is selected.

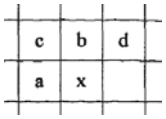

Figure 1 coding mode selection structure

Implementation of Conventional Mode Coding. If you enter the normal coding mode, according to the prediction, prediction error coding, variable update three steps to carry out.

Before we make the forecast, we will quantify and merge the gradient values obtained above. This is to reduce the number of contexts. After entering the forecast, we will complete the following four steps:

Step1: boundary detection, that is, calculate the predicted value Px;

Step2: Predicted value correction, the value of Px is corrected, Px value should be constrained in [0 ... MAXVAL], MAXVAL is the maximum possible value of an image;

Step 3: Predictive error calculation, where the prediction error is e, the actual image sample value is Ix, then e = sign (Ix-Px), sign = {- 1,1};

Step4: Predictive error quantization and reconstruction value, at NEAR = 0, reconstruction value Rx = Ix, at NEAR> 0, quantify e, quantize and then rebuild Rx.

After the determination and quantization of the predicted value is completed, the coding error of the prediction error is Golomb coding. The coding process is completed in three steps:

Step 1: Golomb variable calculation, calculate the Golomb coding variable k, k value is the least significant bit of the error map value, calculated by A [0 ... 364] and N [0 ... 364], k = {0,1,2, 3,4 ...};

Step2: Error mapping;

Step3: Mapping error coding.

Variable update: the value of the variables A [Q], B [Q], C [Q], N [Q] is updated, and the updated process must be carried out according to the value of the prediction error.

Achievement Principle of Run Length Coding. If you enter the run length coding mode, according to the run-time scan, run length coding, run-time interrupt value encoding three steps to complete

Run scan: scan the image data, the first line scan, the actual value of x xx, ABS (Ix-Ra) <= NEAR, then continue to scan, and run length n plus 1, a line scan is completed, the interrupt scan;

If the run is terminated at the end of a line, the output codeword is '1' if n> 0, and then a value is taken to perform a code similar to Golomb; if the run length is n = 0,

Run Interrupt Code: If the run is not at the end of the image, the new interrupt value is encoded (similar to Golomb's encoding).

After the end of the run code also need to follow the normal mode code to update the relevant variables.

Implementation of Key Technologies in JPEG - LS Remote Sensing Image Coding Algorithm

JPEG-LS coding algorithm is clear and easy to implement in hardware language. In the FPGA implementation, the modular design idea is adopted, which can be divided into the following modules: data cache module, location of image data and establishment of context model; gradient calculation Module, complete the calculation of the gradient value of the context model, and the selection of the coding mode; the conventional coding module, the run length coding module and the updating module of the parameters. The whole algorithm uses the pipeline in the realization of the FPGA, while the local use of parallel design, both to reduce the depth of the water, but also improve the coding efficiency of the system [7], in the use of VHDL language to achieve the algorithm, the coding process 2 shows:

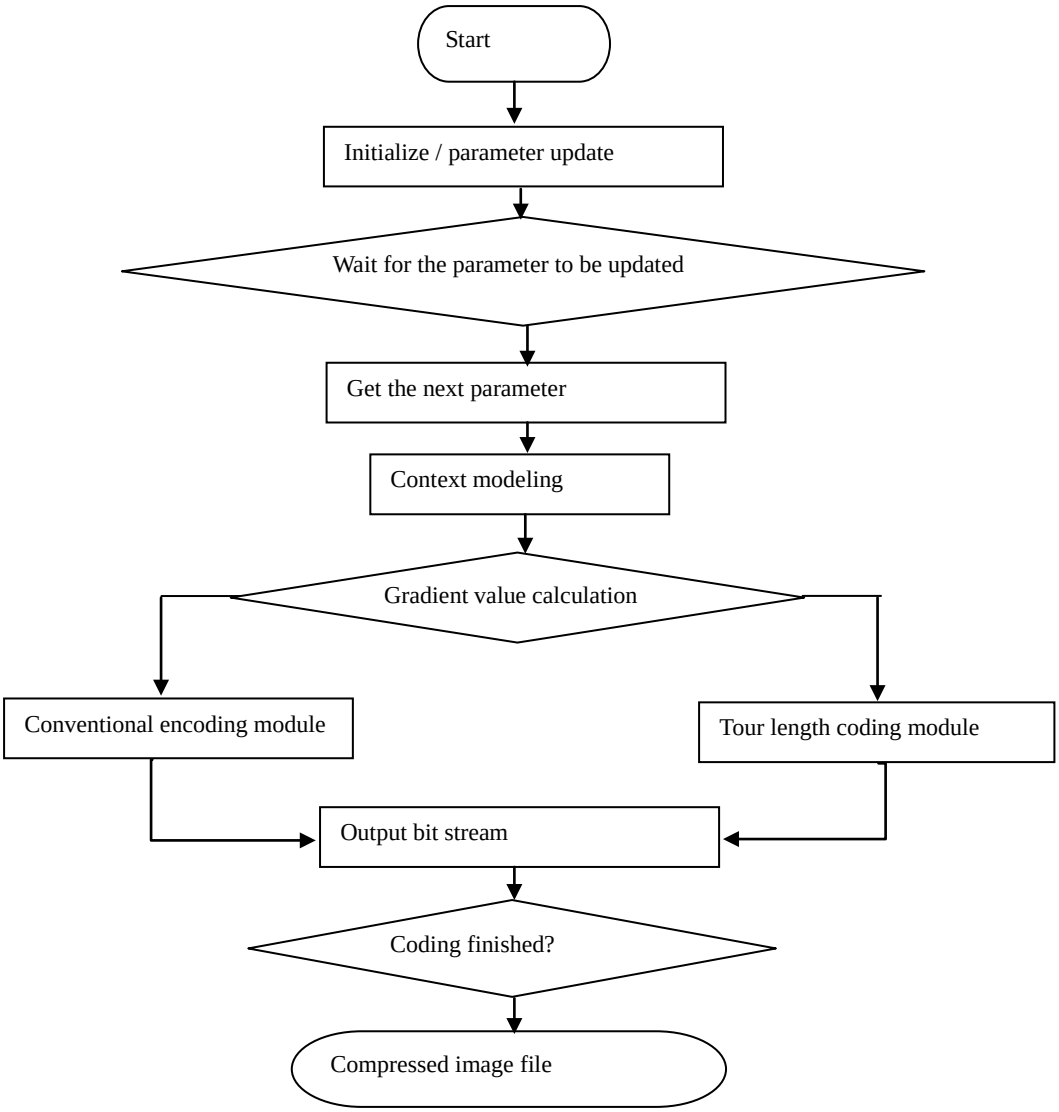

Figure 2 JPEG-LS coding flow chart

IP Core Used in the Encoding Module. In the JPEG-LS encoding module, in the source image data collection and storage issues, the rational use of ise14.7 provided in the IP core aqram, fifora, fifstream, aqram for the internal single-port block memory, used to store variables A [Q], B [Q], C [Q], N [Q] [8], are used to update the variables at any time. fifora used to store the results of the uplink coding for the current line of coding to provide b, c, d data, the encoding of each data can be sent to the FIFO in a timely manner, so that the timely update of the data. fifostream The bitstream data used for temporary caching.

Conventional Encoding Module. The design of the main module (including conventional coding and run-length coding) in a process to achieve, because the need to use the previous data coding to calculate the context, which constitutes a feedback, so the use of state machine to complete, first through the gradient of the operation to determine the entry The normal mode code or enter the run mode code, the judgment process in the state S0, S0 also calculate the value of SIGN.

In the use of VHDL language to achieve the normal mode coding, the use of state machine to achieve, the state jump as shown in Figure 3:

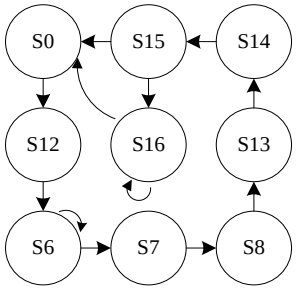

Figure 3 Conventional encoding module status jump map

S0: selection of coding mode;

S12: Calculate the predicted value Px and make boundary processing for it;

S6: Read the values of A [Q], B [Q], C [Q], N [Q]; correct Px; calculate k value;

S7: Initialize the value of aqram; border processing the corrected Px; calculate the value of MErrval based on m, zero_num and k values;

S8: Calculate the value of Errval based on the MErrval value; if it is the first 255th column, do not read fiforam, otherwise enoram;

S13: Rx is calculated according to SIGN; the values of the variables A [Q], B [Q], C [Q], and N [Q] are updated at the same time;

S14: adjust the values of Ra, Rb, Rc and Rd; adjust the values of A [Q], B [Q], C [Q], N [Q]

S15: the value of the adjusted A [Q], B [Q], C [Q], N [Q] is written to aqram;

S16: for line breaks.

Run Length Encoding Module. There are two cases of run length coding, the first case (run 1), if the run code is entered, and at least one run data is output before the end of the run; the second case (run 2), although satisfying the run conditions, Does not output run-time data, but directly into the run-off. The state jump for run 1 is shown in Figure 4:

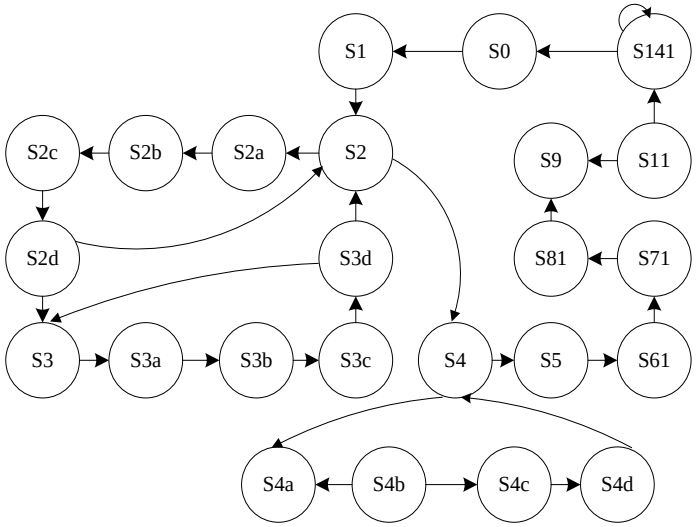

Figure 4 State 1 of the run jump

S2: determines the encoding state, and outputs the compressed encoded data, where S2a to S2d state is the clock added to the output data.

S3: adjust the value of Ra, Rb, Rc and Rd by reading and writing control signals to fifora; adjust RUNindex, judge whether the length of the travel code is completed according to the value of 2J [RUNindex]:

S4: Adjust the values of Ra, Rb, Rc and Rd;

S5: read out fifora data; adjust the value of Ra, Rb, Rc and Rd;

S [Q], B [Q], C [Q], N [Q]; the value of SIGN is evaluated according to the relationship between the absolute value of Ra-Rb and NEAR;

S61, S71, S81 have the same function as S6, S7, and S8 in the conventional mode code.

S9: Calculate the value of Errval;

S11: update the values of A [Q], B [Q], C [Q], N [Q]; calculate the value of Rx;

Aq365, Bq365, Cq365, Nq365, Nnq365 or Aq366, Bq366, Cq366, Nq366, Nnq366; the end of the line is assigned to the updated A [Q], B [Q], C [Q] Do it.

Run 2 is basically the same as the conventional encoding, except that the value of Q is 365 or 366, and is not described here. Runtime 2 status jumps as shown in Figure 8:

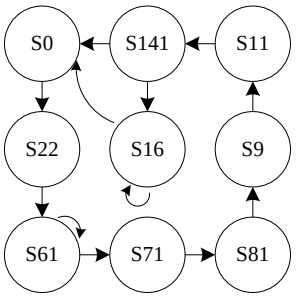

Figure 5 The state jump graph for Run 2

Simulation and Synthesis Results

In this paper, the JPEG-LS coding algorithm based on FPGA is used to perform the system-level and module-level pre-simulation and post-simulation, and the simulation waveform is obtained by using Modelsim software with the image with precision of 10, high 1024 and width of 256 as input test image. Continuous input of the test image data, the simulation results are shown in Figure 6:

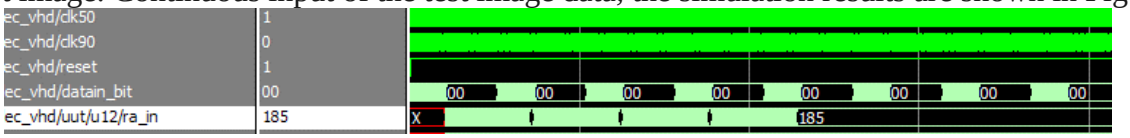

Figure 6 Image Input Test Results

The encoder is mainly responsible for a sub-image compression coding, ise14.7FPGA development platform in the XC4VSX55-12ff1148 chip integrated after the ENC module configuration shown in Figure 7:

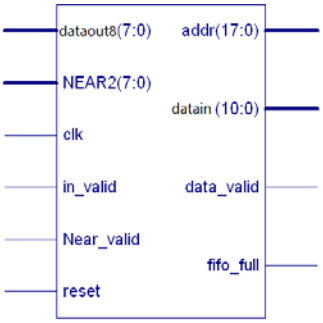

Figure 7 JPEG-LS encoder integrated chip configuration diagram

The main port is described as follows: dataout8 (7: 0) Output stream data; datain (10: 0) Encoder input data; addr (17: 0) Encoder input data corresponding address

Table 1 lists the JPEG-LS encoding algorithm in the FPGA chip XC4VSX55-12ff1148 on the realization of the use of hardware resources, and the speed of coding that is a time required to compress an image.

Table 1 JPEG-LS encoding module integrated performance table

Module name	Occupies the number of slices	Use the slice rate	On-chip storage	Memory usage	Maximum frequency （MHZ）
Coding module	4414	17%	11	3%	135.09

From the data in the table, it can be concluded that the number of logical units occupied by the

encoder is only 17%, the system occupies less area, has optimized the area, and can expand some other functions on the chip.

Conclusion

In this paper, Xilinx XC4VSX55 chip is used as the hardware implementation platform, and the hardware description language VHDL is used to realize the JPEG-LS coding algorithm. The modular design idea is adopted, and the pipelining design method is adopted on the whole system, and the parallel processing And improve the data throughput of the system, and can take full advantage of XC4VSX55 chip internal resources to achieve the design of the area for the hardware to achieve lossless compression provides a solution. In the realization of coding, XC4VSX55 chip with only 17% of the resources, but also in this on the basis of other functions to achieve the expansion, such as image data, multi-channel parallel input, improve image compression efficiency.

Acknowledgements

Fund Project: Haikou Institute of Economics scientific research project "JPEG-LS multi-channel parallel coding algorithm FPGA implementation" (hjkz15-03 key projects);

References

[1] WEINBER GERMJ, SEROUSSIG, SAPIROG. The LOCO-I Lossless Image Compression Algorithm: Principle sand Standardization into JPEG-LS[J]. IEEE Transactions on Image Processing, 2000, 9(8):1309-1324.

[2] An Improved Image Compression Based On JPEG-LS[A]. Li-xin ZHANG and Pei-pei WANG. Proceedings of 2015 International Conference on Computer Science and Information Engineering[C].2015

[3] Efficient lossless multi-channel EEG compression based on channel clustering[J]. Behzad Hejrati, Abdolhossein Fathi, Fardin Abdali-Mohammadi. Biomedical Signal Processing and Control. Volume 31,January 2017, Pages 295-300

[4] Chen Jun, Wang Huaichao, Gu Xiaodong, et al.Study on lossless compression of spaceborne image based on LOCO-I algorithm [J]. Microelectronics and Computers, 2011,28 (11): 169-173.

[5] NIE Yong-kang, LEI Jie, LI Yun-song, et al. VLSI structure design of JPEG-LS near lossless image encoders [J]. Journal of Xidian University (Natural Science Edition), 2016, 43 (4): 75-80.

[6] HAN Jun-ping, CHENG Yong-qiang, DAI Xin. IP solid-core design of JPEG-LS image lossless compression[J]. Journal of Taiyuan University of Technology, 2010, 41 (6): 713-716.

[7] Wei Yahui.Study and Implementation of JPEG-LS Compression Algorithm for Remote Sensing Image Based on FPGA [J]. Journal of Xinyang Agriculture and Forestry College, 2016,26 (2): 107-110.

[8] HAN Tao, WU Heng, HOU Hai-xiao, et al.Application of improved JPEG-LS coding method in flight test [J]. Computer Measurement and Control, 2016, 24 (6): 311-314.