Approach To Implementation On Fpga Of Data Compression Algorithm C Language Descriptions By The Means Of Vivado Package

Valya Khomits [1], Valeriy Hlukhov [2]

1. Computers Department, Lviv Polytechnic National University, UKRAINE, Lviv, S. Bandera street 12, E-mail: blueferrari5252@gmail.com

2. Computers Department, Lviv Polytechnic National University, UKRAINE, Lviv, S. Bandera street 12, E-mail: valeriygl@ukr.net

Abstract – The features of devices for monochrome images lossless compression by JPEG-LS method in modern element base are discussed. Capabilities of Vivado package (Xilinx) for JPEG-LS algorithm C- to suitable for implementation in FPGAs VHDL-descriptions transformation were tested and described. C language structures, which can not be processed by specified means and possible circumvention of such structures were defined.

Keywords: lossless compression, FPGA, JPEG-LS, software implementation, hardware implementation.

I. Introduction

The possibilities of the Vivado package (Xilinx ~~file~~) for converting the existing description of the algorithm JPEG-LS in C language into VHDL descriptions suitable for implementation in the FPGA are described in the paper. The C language constructs which can not be processed by the indicated tools are defined. Also possible ways of bypassing such structures are defined. The hardware implementation of image compression using the JPEG-LS method is less flexible than software and requires specifications in the size of the data to be processed. Also, unlike the program, the future FPGA compression unit has to processes the video stream of data, rather than individual images, which adds specification to the implementation method of such a device. Such units are used in security systems, systems for collecting information, satellite, underwater and other systems of photo-video surveillance, as well as in astronomical instruments and telescopes.

II. Development of the hardware implementation of the jpeg-ls method by means of Vivado

There is a well-known software implementation of the algorithm JPEG-LS [3], which allows you to make changes to the algorithm of work, to model its work and to determine the characteristics of the method. Also well-known example [1] of successful hardware implementation of the compression node on the FPGA by the JPEG-LS method exists and the means [2], which allow the conversion of C-descriptions into VHDL descriptions, are suitable for the development of FPGA topology.

The aim of the work is to explore the possibility of using existing and custom-made C-descriptions of compression algorithm and the means of their transmission in VHDL descriptions to create the topology of an FPGA, which satisfies the requirements of customers.

C-descriptions that were not created for implementation to FPGA were used. The main task to be solved for a successful hardware implementation of jpeg-ls v2.2 is to adapt the existing C-code for its use in the Vivado HLS environment.

To do this, a project is created in the Vivado HLS environment, which contains all jpeg-ls v2.2 program files.

When creating a project, you must specify the main function of the project (in this case it is main ()) and choose or create a test-bench file to verify the program's performance.

At this stage of the work there are serious complications, because the code implementation and code that can process and synthesize Vivado HLS, are significantly different.

In particular, during synthesis the C / C ++ constructs that are not synthesized by Vivado HLS should be eliminated or replaced. Such constructions are described below.

III. System calls

System calls can not be synthesized because they are actions that are related to the execution of some queries from the operating system in which the C program runs.

Vivado HLS will automatically ignore frequently used system calls that are used only to display data and have no effect on the execution of the algorithm (for example, printf () and fprintf (STDOUT)), so system calls can not be synthesized and should be removed from code (functions) before synthesis.

Other examples of such calls are getc(), time(), sleep(), etc., which make requests to the operating system.

IV. Dynamic memory usage

Any system calls that control the allocation of memory in the system, such as malloc (), salloc (), and free () use the operating system memory resources which are created and released when the program is running.

For hardware synthesis, such constructions should be modified so that they become completely autonomous, with precisely defined by all necessary resources.

System requests for memory allocation should be removed from the code before the synthesis. However, since the dynamic allocation of memory is used to implement the functional units of a project, it must be transformed into equivalent representations that are to be synthesized.

V. Recursive functions

Recursive functions can not be synthesized. This also applies to functions that can create endless recursion.

There are also certain limitations when using pointers. The pointer definition is not supported in the general case, but is supported for internal types in C.

During the synthesis of the jpeg-ls v2.2 codec, 23 designs were identified, which were discussed above and should be replaced for successful synthesis of the VHDL description.

Errors occur when working with external files, system calls such as clock, putc, and dynamic memory usage functions.

The first step in eliminating these errors is to clearly identify the working file name and avoid entering the initial information (parameters) from the command line.

With hardware implementation, it is not possible to choose a filename because the information will be received continuously and will be compressed only with one scenario.

The code should be modified so that the compression does occur without the input of parameters from the keyboard. Therefore, in the C-description, you must specify a file name and remove the keyboard-name input that is not already necessary. By highlighting the problematic constructions of the code with the __SYNTHESIS_ macro, you can achieve code synthesis, but also loss its functionality.

As the result of the Vivado synthesis, HLS generates three groups of files in different hardware description languages – VHDL, Verilog, and SystemC.

With this approach, the functionally reduced VHDL code is created since the constructs that can not be synthesized by the Vivado HLS package were removed from the source code.

The result of the synthesis allows further research (review of the code itself and the technological scheme that corresponds to it) with the standard tools of designing the topography of the FPGA, that is, allows you to work out the full design technology.

After successful synthesis of the created vhdl-description you can view the RTL and technological scheme of the project.

VI. Further research direction

The results allowed us to determine the areas of C-descriptions that can not be processed by any tools of the Vivado HLS package.

The found parts of the code were removed from the descriptions with the help of macros __SYNTHESIS__, which allowed to completely pass the process of creating the topology of the FPGA from the original existing description in C.

The next step is to replace the source code segments with such descriptions that can be processed by means of the Vivado HLS package, as well as the creation of technological tools that can automatically search for certain problematic areas in the original C-descriptions, followed by their automatic replacement (with the possible participation of a person) to equivalent descriptions that do not violate the algorithm's operation and are processed by tools of the Vivado HLS package.

Conclusions

In this work we tested the implementation of the algorithm with the tools of the Vivado HLS package (Xilinx) with the transformation of existing descriptions of the C algorithm into VHDL descriptions, suitable for implementation in the FPGA. Constructs of the language C, which can not be processed by the indicated means, and provided methods for correcting the initial descriptions language C were defined.

Complications in working out existing C language descriptions arise when data streams and files, dynamically distributed memory, system calls and pointers are in the program.

Due to the isolation of problematic areas, a functionally limited VHDL description is created, which allows to run a complete technological path to create an FPGA topology.

References

[1] M. Weinberger, G. Searcy, G. Sapiro, "Loko-I Compression of Impressions Without Loss," Technical Report No. Hewlett-Packard Laboratories No. HPL-98-193R1, November 1998 Transformed October 1999, Algorithm: Principles and Standardization in JPEG-LS. "IEEE Trans. Image Processing, t. August 9, 2000, p.1309-1324.

[2] M. Weinberger, G. Searcy, G. Shapiro, "LOCO-I: Low Context-Based Compression of Lossless Imagery" Algorithm, Proceedings IEEE Data Compression Conference, Snowbird, Utah, March-April 1996 .

[3] DISCRETE MATHEMATICS: ALGORITHMS. JPEG, JPEG2000, JPEG-LS. Compression of images with loss and without [Electronic resource]: [Web site]. – Electronic data. – Access mode http://rain.ifmo.ru/cat/view.php/theory/data-compression/jpeg-2006

[4] JPEG-LS encoding (numeric and almost no offset) Design specification. 2007 – 2013 ALMA Technologies. http://www.ict.edu.ru/ft/002147/sb4_page86_95.pdf

[5] File Converter . Online and Free. [Electronic resource] – Access mode: https://convertio.co

[6] Knowledge. Golomb code [Electronic resource]: [Website]. – Electronic data. – Access mode: http://www.wikiznanie.ru/wikipedia/index.php/%D0%9A%D0%BE%D0%B4_%D0%93%D0%BE%D0%BB%D0%BE%D0 % BC% D0% B1% D0% B0

[7] Hewlett-Packard Laboratory [Electronic Resource]: [Website]. – Electronic data. – Access mode: http://www.labs.hp.com/research/info_theory/loco/locodown.htm

[8] Wikipedia. Netpbm Format [Electronic Resource]: [Website]. – Electronic data. – Access mode: https://en.wikipedia.org/wiki/Netpbm_format

[9] Valerii Hlukhov, Valya Khomits "Compression of lossless images by JPEG in FPGA" – Third scientific seminar "Cyber-physical systems: achievements and challenges". PROGRAM. National University "Lviv Polytechnic", June 13-14, 2017 Lviv, Ukraine.

[10] Valerii Hlukhov, Adolf Lukenyuk, Sergii Shenderuk. Satellite scientific data collection and accumulation system as a basis for cyber-physical systems construction. Advances in Cyber-Physical Systems. Volume 1. Number 2. Lviv Polytechnic National University. 2016. P. 77 – 86.