Using the same type of IP-cores in the Virtex-6 family FPGA-architecture for distributed image processing

V M Zakharov¹, S V Shalagin¹ and B F Eminov²

¹Kazan National Research Technical University named after A.N. Tupolev–KAI, Karl Marks street 10, Kazan, Russia, 420111
²SimbirSoft Ltd., Fedoseevskaja street 58, Kazan, Russia, 420111

E-mail: gilvv@mail.ru

Abstract. Distributed image processing in real time is a relevant task. An effective solution to this problem at the hardware level is possible on the basis of creating prototypes of devices on FPGA. In this regard, the paper solves the problem of distributed computation of two-dimensional fast Fourier transform (TDFFT) based on the same type of IP-cores implemented in the architecture of the Virtex-6 family FPGA-architecture. The possibility of distributed implementation of each of the stages of the TDFFT, performed using four transformations of the “butterfly” type (TrB) over four elements of the processed data array, is shown. When using a specialized CAD-system, estimates of the time and hardware complexity of the IP-core implementing TrB are obtained. The estimated operating frequency of this IP-core is approximately 108 MHz. Hardware complexity estimates by the number of configurable elements involved in the Virtex-6 family FPGA-architecture are determined. TrB must be performed over an array of \( N^2 \) elements \( a \cdot N^2 / 4 \) once, where \( a = \log_2 N \) is the number of TDFFT stages, and \( N = 2^k \). The ability to parallelize the specified operations at each of \( a \) stages by using IP-cores that implement four TrB is shown. The results can be used to estimate hardware and time costs for distributed TDFFT execution over a square data array \( N \times N \)-dimension.

1. Introduction

The problem of real-time image processing is topical today. Software implementation of algorithms employing this problem on a general-purpose computer are limited by the features of the von Neumann architecture. A way out of the current situation is using special-purpose computers, particularly those embedding the hardware accelerators for various purposes that implement distributed computing based on ASIC [1, 2]. In this regard, the task of creating prototypes of devices that implement image processing on the FPGA is relevant.

In this paper, we solve the problem of creating a prototype of a device that implements two-dimensional fast Fourier transform (FFT) using distributed computing based on the same type of IP cores made in the FPGA-architecture [3]. Based on the estimates of hardware complexity and IP-cores functioning delay time, estimates of hardware and time complexity have been obtained regarding the execution of two-dimensional FFT for an image of a given dimensionality.

An algorithm is known to calculate a two-dimensional FFT (TDFFT) based on the one-dimensional FFT procedure [4, 5]. A weak point of that algorithm is the fact that it is executed in two stages. At the first stage, the patterns are computed for rows, while at the second stage for columns. Or vice versa,
columns at the first stage and rows at the second one. In any case, until all operations have been performed for the first stage, one cannot go to executing the operations at the second stage. Algorithm proposed in this paper does not have this weak point: Operations are performed over the elements of a two-dimensional data array in parallel, without being divided into stages. The development of the subject [3], described in this paper, consists in the study of pipeline operations on floating-point numbers in the Virtex-6 family FPGA-architecture, which corresponds to the architecture of specialized graphics accelerators that serve for various tasks.

The obtained results of estimates of time and hardware complexity devices that implement TDFFFT on FPGA-class are applicable to solving problems of distributed image processing in real time using specialized graphics accelerators.

2. Two-dimensional fast Fourier transform

Let us consider the TDFFFT for number array \( \{ x_{mn} \} = X \) sized \( N \times N, N = 2^a \). It can be represented as follows [4, 5]:

\[
G_{uv} = N^{-2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x_{mn} \cdot W_N^{mu+nv},
\]

where \( W_N^{mu+nv} = \exp\left( -2\pi \frac{mu + nv}{N} \right) \), \( u = 0, N-1, \ v = 0, N-1 \), and \( \{ G_{uv} \}_{N \times N} = G \) is a pattern for \( X \).

Similarly, the reverse TDFFFT is executed:

\[
x_{mn} = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} G_{uv} \cdot W_N^{-mu-nv}, \ m = 0, N-1, \ n = 0, N-1.
\]

The system indicated (1) can be represented similarly to one-dimensional FFT, as follows:

\[
G_{uv} = \begin{bmatrix} S_{2n_1,2m_0} + W_N^{u} \cdot S_{2n_1,2m_0+1,2m_1} + W_N^{v} \cdot S_{2n_1,2m_0+1,2m_1+1} + W_N^{uv} \cdot S_{2n_1,2m_0+1,2m_1+1} \end{bmatrix},
\]

where \( u = 0, N-1, \ v = 0, N-1 \), and \( S_{2n_1,2m_0} = \sum_{n_0=0}^{N/2-1} \sum_{m_0=0}^{N/2-1} x_{n_1,m_0} \cdot W_N^{2(n_1u+n_0v)} \).

As a result, the TDFFFT can be implemented on single-type operations executed over the elements of matrices \( D \) sized \( 2^d \times 2^d \), submatrices \( \{ x_{mn} \} \), \( m = 0, N-1, \ n = 0, N-1 \), \( d = 1, a \), \( a = \log_2 N \):

\[
D_{uv} = \begin{bmatrix} D_{u,v} + W_N^u \cdot D_{u+2^d-1,v} + W_N^v \cdot D_{u,v+2^d-1} + W_N^{uv} \cdot D_{u+2^d-1,v+2^d-1} \end{bmatrix},
\]

\[
D_{u+v+2^d-1,v} = \begin{bmatrix} D_{u,v} - W_N^u \cdot D_{u+2^d-1,v} + W_N^v \cdot D_{u,v+2^d-1} - W_N^{uv} \cdot D_{u+2^d-1,v+2^d-1} \end{bmatrix},
\]

\[
D_{u,v+2^d-1} = \begin{bmatrix} D_{u,v} + W_N^u \cdot D_{u+2^d-1,v} - W_N^v \cdot D_{u,v+2^d-1} - W_N^{uv} \cdot D_{u+2^d-1,v+2^d-1} \end{bmatrix},
\]

\[
D_{u+v+2^d-1,v+2^d-1} = \begin{bmatrix} D_{u,v} - W_N^u \cdot D_{u+2^d-1,v} - W_N^v \cdot D_{u,v+2^d-1} + W_N^{uv} \cdot D_{u+2^d-1,v+2^d-1} \end{bmatrix},
\]

where \( u = 0, 2^{d-1} - 1 \) and \( v = 0, 2^{d-1} - 1 \).

**Note.** Number of matrices \( D \) sized \( 2^d \times 2^d \) was found to be \( 4^a-d \), \( d = 1, a \), \( a = \log_2 N \).

Let us represent the system indicated (3) in matrix form as a complex of single-type “butterfly” transforms (BTr) used in computing the one-dimension FFT:

\[
\begin{pmatrix} D_{u,v} \\ D_{u+v+2^d-1,v} \\ A_1 \\ A_2 \\ B_1 \\ B_2 \end{pmatrix} = \begin{pmatrix} 1 & W_N^u \\ 1 & -W_N^u \end{pmatrix} \begin{pmatrix} A_1 \\ A_2 \\ B_1 \\ B_2 \end{pmatrix},
\]

\[
\begin{pmatrix} D_{u+v+2^d-1,v} \\ D_{u,v+2^d-1} \\ B_1 \\ B_2 \end{pmatrix} = \begin{pmatrix} 1 & W_N^v \\ 1 & -W_N^v \end{pmatrix} \begin{pmatrix} D_{u,v} \\ D_{u+v+2^d-1,v} \\ A_1 \\ A_2 \end{pmatrix},
\]

(4)
As a result, to perform one operation indicated (3) over four elements of number array $X$, according to (4), four BTrs are required. The above operations must be performed over an array of $N^2$ elements $\log_2 N$ times. According to (3) and (4), the following statements hold true:

Statement 1. Implementing a two-dimensional FFT over number array $X$ sized $N \times N$ requires executing $a \cdot N^2/4$ operations indicated (3) or $a \cdot N^2$ BTrs.

Statement 2. Single-type operations indicated (4) over number array $X$ sized $N \times N$ can be performed in parallel.

Statements 1 and 2 substantiate the applicability of special-purpose computers implementing a single-type operation indicated (3) or (4).

3. Complexity of implementing the same-type “butterfly” transformation on the FPGA

Let us consider the implementation of a hardware module implementing BTr (hereinafter, the “Module”) as an IP-core in FPGA-architecture. The module implements an operation of the form (3) and operates with 16-bit numbers represented according to the IEEE 754c standard as half-precision floating-point numbers. In this case, each complex number is represented by 32 digits, 16 of which are allocated for the real and imaginary parts, respectively. The functional logic diagram of the Module implemented in ISE Design Suite 14.7, a sheet-oriented CAD editor manufactured by Xilinx, Inc., includes:

- Three blocks of a complex number multiplication,
- Eight adders, four of which perform the addition operation, the rest-the subtraction operation, and
- Registers to ensure pipelined data processing.

The Module is implemented on FPGA XC6VCX195T-2FF784 (Virtex-6 family manufactured by Xilinx, Inc.) that comprises $D$-triggers, generators of Boolean functions of six variables (LUTs), and input/output units (I/O units).

- 7,672 of 249,600 $D$-triggers available (less than 3.1 %);
- 5,814 of 124,800 LUTs (about 4.7 %);
- 2,609 of 31,200 Slices (about 8.4 %) comprising four LUTs each, implementing Boolean functions of five or six variables and eight $D$-triggers; and
- 258 of 400 I/O units (about 64.5 %), of which 128 are allocated for inputs and 128 for outputs, and 2 more for clock signal and for resetting triggers, respectively.

The maximum function delay time of the Module is 9.280 ns. Values at outputs are computed 30 clock periods upon the relevant values having arrived at the input, i.e. upon 278.4 ns.

4. Discussion

According to [6, 7], a FPGA-based combinational circuit close to the optimal implementation requires involving at most 0.5 of the resources of each type, i.e., $D$-triggers, LUTs, and Slices.

According to the complexity estimates obtained, no more than two Modules can be placed on one FPGA of the Virtex-6 family. Limiting factor is the number of I/O units.

$a \cdot N^2/4$ FPGAs of the Virtex-6 family are required to calculate a two-dimensional FFT for a number array sized $N \times N$. Moreover, to calculate the operation described according to (4), one FPGA of the above family is required, each of which accepts at the input and returns by four elements $(D_{u,v}, D_{ur2^{d-1},v}, D_{u,v+2^{d-1}},$ and $D_{u+2^{d-1},v+2^e})$. Estimating the operation frequency of a distributed programmable-architecture system (DPAS) [8] that includes this number of FPGAs makes at least 108 MHz. Number of delay periods in distributed computing the values of elements $D_{u,v}, D_{ur2^{d-1},v}, D_{u,v+2^{d-1}}$
\[ D = a_1 \cdot 2^{d-1} + \ldots + a_n \cdot 2^{d-1}, \] according to (4), has been 30 \cdot a since the elements to be processed arrived at the input. Generally, true is

Statement 3 [3]. To process an array sized \( N \times N \) on a DPAS comprising \( P \) FPGAs that receive for processing \( q \) elements per clock period and implement a transform indicated (4) per \( T \) clock periods, \( z \) time units each, at least \( \log N \cdot \left( \left[ \frac{N^2}{(4 \cdot P)^z} + 29 \right] \right) \) time units are required.

For example, let us find the lower estimate of time required to process a number array sized \( N \times N \), \( N = 2^{10} = 1024 \). We will perform the two-dimensional FFT in 10 stages, at each of which \( N \times N = 2^{20} \) elements are required. Each Virtex-6 FPGA accepts four elements to be processed, i.e., \( q = 4 \). If there are \( P \) of the above-mentioned FPGAs in DPAS, then the stage is computed within \( 3a \left[ \frac{N^2}{(4 \cdot P)^z} + 29 \right] \) clock periods. If we set the number of Virtex-6 FPGAs to 512, as in modern DPASes [5], the one stage of the two-dimensional FFT is implemented within 541 clock periods 9.280 ns each, while 10 stages within 5,410 clock periods, which makes about 50.2 \( \mu \)s to process one stage and about 50.2 \( \mu \)s to process the entire array. About 19.9 thous of arrays sized 1,024 by 1,024 can be processed within one second.

In case of processing an array sized 2,048 by 2,048 on a DPAS comprising 512 Virtex-6 FPGAs, the lower estimate of processing time, according to Statement 3, is 212 \( \mu \)s, while about 4,716 arrays of the above size can be processed within one second.

Due to the parallel-serial input of number array \( X \) into the FPGA, the number of the IP-cores implementing the Module and configured on the Virtex-6 FPGA can be increased significantly.

For a module implemented on an XC6VCX195T-2FF784 FPGA, the restriction factor is the number of Slices used for its implementation. In this case, it is possible to place 5 IP cores implementing the Module on the specified FPGA. This requires an additional 256 D-triggers to store elements of the number array \( X \). In general, for parallel-serial input to the FPGA (or output from it) of \( 2^t \) binary bits, \( 2^t \) additional clock cycles of the device and \( 2^t \) I/O units are required, if \( f = t + i \). Consider a special case when bitwise input four 32-bit elements and to output bit-for-bit results, four elements of 32 bits each, requires 128 I/O units and 2 cycles of operation. As a result, the number of clock cycles for performing an operation of the type (3) in the Module is 34. According to Statement 3, if \( P = 512 \), \( q = 2 \) \( T = 34 \), and \( z = 9.280 \mu \)s, the lower estimate of the processing time of the number array sized \( N \times N \), \( N = 2^{10} = 1024 \), is about 98.1 \( \mu \)s. Within one second, about 10.92 thous of arrays sized 1,024 by 1,024 can be processed. In case of processing an array sized 2,048 by 2,048, the lower estimate of the operation time is approximately 383 \( \mu \)s, and about approximately 2,610 arrays of that size can be processed within one second.

5. Conclusion
Based on estimating the time and hardware complexity of the Module as a single-type IP-core in the FPGA-architecture of the Virtex-6 family, we have evaluated the function delay time and the hardware complexity of a prototype device implementing the pipelined computing of a TDFFT accompanied by time decimation on DPAS. Relevant estimates for the Module were executed using a special-purpose FPGA CAD, ISE Design Suite 14.7.

The results obtained in this work allow us to assess the potential for implementing a TDFFT.

Acknowledgements
The authors also acknowledge E.M. Kirillov, master student at KNRTU-KAI, for his help in obtaining experimental data.

References
[1] Raikhlin V A, Vershinin I S, Gibadullin R F and Pystogov S V 2016 Reliable Recognition of Masked Cartographic Scenes During Transmission over the Network IEEE: 2016
International Siberian Conference on Control and Communications (SIBCON) DOI: https://doi.org/10.1109/SIBCON.2016.7491657

[2] Gibadullin R F, Vershinin I S and Minyazev R S 2018 Development of Load Balancer and Parallel Database Management Module IEEE: 2018 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM) DOI: https://doi.org/10.1109/ICIEAM.2018.8728629

[3] Zakharov V M, Shalagin S V and Eminov B F 2019 Distributed image processing based on the same IP-cores in FPGA-architecture CEUR Workshop Proceedings (Data Science Session at the 5th International Conference on Information Technology and Nanotechnology, DS-ITNT 2019; Samara; Russian Federation; 21-24 May 2019) Vol. 2416 126–133 URL: http://www.scopus.com/inward/record.url?eid=2-s2.0-85071029159&partnerID=MN8TOARS

[4] Gonzalez R C and Woods R E 2007 Digital Image Processing (Prentice-Hall) p 976

[5] Dudgeon D, Mersereau R and Merser R 1983 Multidimensional Digital Signal Processing (Prentice Hall) p 400

[6] Shalagin S V 2004 Computer evaluation of a method for combinational-circuit synthesis in FPGAs Russian Microelectronics 33 1 46–54

[7] Shalagin S V 2016 Implementing digital devices in FPGA architecture when using distributed computing in Galois fields KNRTU-KAI Press (Kazan) p 228 [In Russian]

[8] Dordopulo A I, Kalyaev I A, Levin I I et al. 2010 High-Performance Reconfigurable Computer Systems Supercomputers 3 (3) 44–48 [In Russian]