Hardware Implementation of Video Processing System Using RNS

Nikolai I. Chervyakov[1], Pavel A. Lyakhov[2],
Andrey S. Ionisyan, Maria V. Valueva
Department of Applied Mathematics and Mathematical Modeling
North-Caucasus Federal University
Stavropol, Russia
[1]k-fmf-primath@stavsu.ru, [2]ljahov@mail.ru

Dmitry I. Kaplun[1], Vyacheslav V. Gulvanskiy[2]
Department of Automation and Control Processes
St. Petersburg Electrotechnical University "LETI"
St. Petersburg, Russia
[1]dikaplun@etu.ru, [2]vvgulvanskii@etu.ru

***Abstract*— This paper considers the creation of video signal processing device. Important performance criteria of these devices are speed and power consumption. We used an Alinx AX309 board containing FPGA Xilinx Spartan6-xc6slx9 as a hardware basis for the implementation of the system. OV7670 video camera was used to obtain a video signal. The output of the processed video was carried out on the Alinx AN430 LCD display and on the standard VGA port. We used the Residue Number System (RNS) to accelerate calculations. It allowed to improve device speed by 28% compared to using traditional two's complement number system.**

***Keywords*— residue number system; digital video processing; system-on-chip**

I. Introduction

The most information about world, conditions and the habitat people receive by vision. Questions of modern digital electronics communication with users by video are so relevant. Note that in the most general case, video processing may be implemented by a computer with loss of performance and increased power consumption. Specialized hardware solutions let to achieve the highest possible speed of video processing at a significantly lower energy consumption [1]. For example, a work station (PC or laptop) consume from 60 W to 600 W of electric power. In this paper we will propose a solution consumes not more than 5 W electric power.

Universal computers are more prone to disruptions because they are less protected from interference of electrical network and a large number of interacting processes of the operating system of a computer, hardware drivers and other applications [2]. Specialized electronics are easier to shield from external interference and connect to a more stable than public electric network a power source. It is more difficult to break work of device driver operation. The application process of video processing implemented in hardware is more reliable. It cannot be infected with a virus and all the clocks of his work are clearly regulated.

This work was supported by the Government of the Russian Federation (state order no. 2.6035.2017/BCh), the Russian Foundation for Basic Research (project no. 18-07-00109 A), and by the Presidential Grant of the Russian Federation (project no. SP-2245.2018.5).

II. Mathematical model of video signal processing

Let's consider a typical scheme of video processing. A source signal is formed on semiconductor crystals of a semiconductor matrix of a video camera. At regular intervals is supplied in the form of electrical pulses to the corresponding contacts-outputs of a video camera or other video capture device. The resulting video signal is buffered in memory of computer and it is subjected to special mathematical processing (filtering). An image I consisting of R rows and C columns is a two-dimensional function $I(x,y)$, where $0 \le x < R$ and $0 \le y < C$ are spatial coordinates and the amplitude I at any point with a pair of coordinates (x,y) is called an intensity or a gray level of the image at this point. For digital grayscale images the intensity represents by unsigned integer numbers range from 0 to 255 with 8-bits representation [3–5].

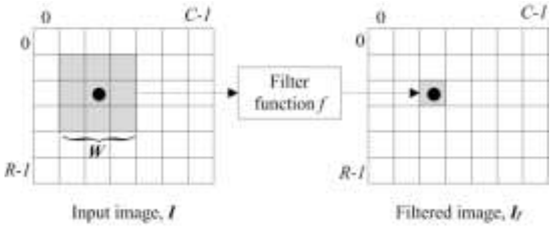

Fig. 1. Image filtering scheme

The filtering of image can be represented in the form:

$$I_f(x,y) = \sum_{i=0}^{d-1}\sum_{j=0}^{d-1} W_{i,j} I(x+i, y+j), \quad (1)$$

where I_f is a processed image and

$$W = \begin{bmatrix} w_{0,0} & \cdots & w_{0,d-1} \\ \vdots & \vdots & \vdots \\ w_{d-1,0} & \cdots & w_{d-1,d-1} \end{bmatrix} \quad (2)$$

is a filter mask of dimension $d \times d$ [6] consisting of real numbers of the IEEE-754 format. Visualization of the image filtering is shown in Fig. 1. After the calculation of new brightness values of all frame pixels, they are displayed on the external display or saved in external memory. The hardware implementation of real arithmetic usually is not effective in terms of speed and resource consumption (transistors, resistors and capacitors). Therefore, we replace the filter mask (2) by its approximate form:

$$V = \frac{1}{D}\begin{bmatrix} v_{0,0} & \cdots & v_{0,d-1} \\ \vdots & \vdots & \vdots \\ v_{d-1,0} & \cdots & v_{d-1,d-1} \end{bmatrix}, \qquad (3)$$

where $v_{i,j}$ and D are integer numbers, furthermore D is nonnegative integer power of 2:

$$\begin{bmatrix} v_{0,0} & \cdots & v_{0,d-1} \\ \vdots & \vdots & \vdots \\ v_{d-1,0} & \cdots & v_{d-1,d-1} \end{bmatrix} \cdot D \approx \begin{bmatrix} w_{0,0} & \cdots & w_{0,d-1} \\ \vdots & \vdots & \vdots \\ w_{d-1,0} & \cdots & w_{d-1,d-1} \end{bmatrix} = W. \qquad (4)$$

Under this assumption, all operations can be performed with minimal hardware costs using standard VHDL or Verilog libraries. Implementation of calculations by the formula (2) can be realized even more effectively using calculations in the RNS.

III. Background on RNS

In RNS, numbers are represented in the basis of mutually prime numbers called modules $\beta = \{m_1, \dots, m_n\}$, $\gcd(m_i, m_j) = 1$, $i \neq j$. The product of all RNS modules $M = \prod_{i=1}^{n} m_i$ is called the dynamic range of the system. Any integer $0 \le X \le M$ can be uniquely represented in RNS as a vector $\{x_1, x_2, \dots, x_n\}$, where $x_i = |X|_{m_i} = X \bmod m_i$ [7].

The dynamic range of RNS is usually divided into two approximately equal parts, so that about half of the range was represented by positive numbers, and the rest of the range by negative numbers. Thus, any integer number satisfying one of the following two relations can be represented in the RNS:

$$-\frac{M-1}{2} \le X \le \frac{M-1}{2}, \text{ if } M \text{ is odd}, \qquad (5)$$

$$-\frac{M}{2} \le X \le \frac{M}{2} - 1, \text{ if } M \text{ is even}. \qquad (6)$$

Operations of addition, subtraction, and multiplication in RNS defined by the formulas showing the carry-free parallel nature of RNS:

$$A \pm B = \left(|a_1 \pm b_1|_{m_1}, \dots, |a_n \pm b_n|_{m_n} \right), \qquad (7)$$

$$A \times B = \left(|a_1 \times b_1|_{m_1}, \dots, |a_n \times b_n|_{m_n} \right). \qquad (8)$$

Reverse conversion of number X from residues $\{x_1, x_2, \dots, x_n\}$ based on Chinese Remainder Theorem (CRT) [7, 8]

$$X = \left| \sum_{i=0}^{n} \left| |M_i^{-1}|_{m_i} x_i \right|_{m_i} M_i \right|_M, \qquad (9)$$

where $M_i = \frac{M}{m_i}$ and $|M_i^{-1}|_{m_i}$ means a multiplicative inverse of M_i modulo m_i.

IV. Architecture of the Proposed Hardware Solution

The proposed hardware solution is shown in Fig. 2 and consists of 8 large functional sectors, containing filtration in RNS, filtration in BNS, RAM manager, output video stream to VGA display and LCD one, capture video stream from video camera and I/O buffering [9, 10].

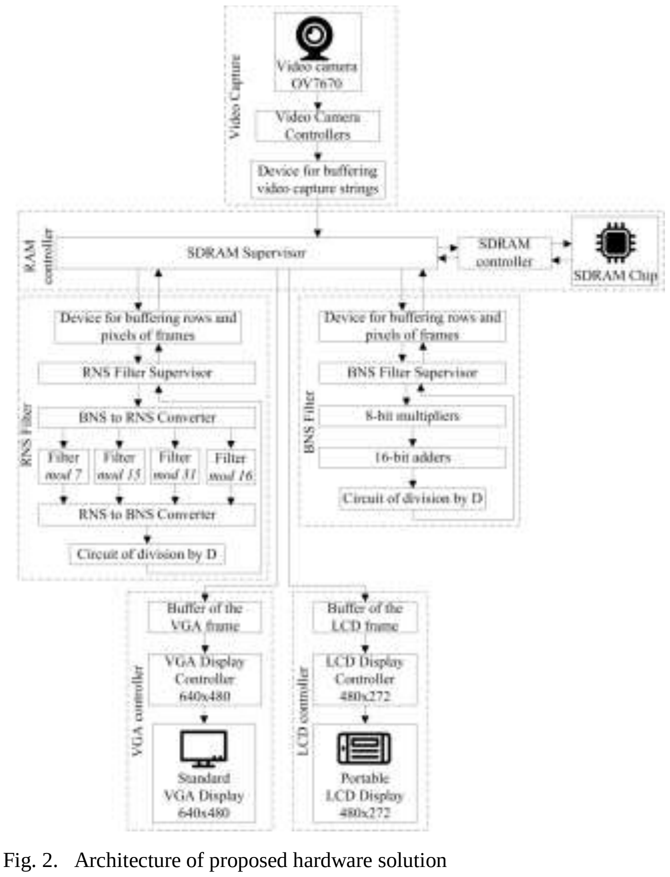

Fig. 2. Architecture of proposed hardware solution

The original signal is captured by the video capture device OV7670 and transferred to RAM line-by-line through the system of buffers. To synchronize work, the number of the current line of the frame is directly transferred to the RAM manager. Two buffers of video capture with RAM (an even-numbered buffer and an odd-numbered buffer) were used to the video capture device guaranteed to write the full line to the buffer and the RAM manager guaranteed to read the full line of this buffer and without the delay of the operation of other devices. While the video capture device writes data to the even-numbered buffer the RAM manager reads data from the buffer of an odd row. But it will start reading data from the even-numbered buffer when the video capture device fills the even-numbered buffer and will begin to fill the buffer of an odd string.

A similar interaction scheme is used for buffers of RNS filter with RAM manager and BNS filter with RAM manager. So that the filters do not fill the buffers faster than the RAM manager reads information from them, the read synchronization tactic from the RAM of the new data line for the corresponding filter is applied. But this approach is not applied for the video capture device. In this case, it is necessary to reduce the reference frequency of the video camera or to transfer lines from the video camera to buffers through one. The RAM manager saves the image received from the video camera in the memory bank of the SDRAM chip.

Simultaneously with the video capture device, VGA and LCD controllers operating at 25,175 MHz and 4 MHz reference frequencies respectively claim to have access to RAM as the highest priority. These devices must receive pixels at their inputs with a specified frequency regardless of whether the brightness characteristics of the pixels were calculated by filters in time. VGA and LCD video controllers constantly scan and display the contents of high-speed dual-port memory type BRAM storing data of only one active line and remembering the contents of the pixels from the previous output line while the RAM controller will not fill this line with new data. Thus, on the display each new line is a mixture of those pixels that the RAM manager managed to write to a two-port memory and pixels of the previous line. This technology allowed to output the video filter result simultaneously to the VGA display and a portable LCD display without loss of quality.

Additionally, in order to have as much current information in the VGA and LCD controller buffers we used "read-ahead" technology. This means that VGA and LCD controllers constantly transmit to the RAM manager the number of the current line displayed on the screen and the RAM manager, knowing this number, starts to read in advance the information from the SDRAM chip for the next row of pixels. Filtration sectors request new data from RAM or write the result of their work in RAM more manageable than video capture controllers and VGA/LCD. The work of each filters can be suspended at any time (if RAM manager is busy).

To perform a comparative analysis of the BNS and RNS filters performance, their architectures were made identical. Differences are only available in the calculation modules. The key feature of the implementation of the filter architecture is minimizing the number of accesses to the RAM manager for reading/writing data. In the high-speed BRAM memory of each filter, the last three loaded lines of pixels are stored at once. When a new line is loaded, the access to the row buffers is recombined so that the filter always has the current information for a quick calculation using formula (2). In addition, a new line of pixels for analysis is read only when the processing of the previous row of pixels is completed and the result is written in RAM.

To accelerate the calculations in the RNS, the brightness characteristics of the pixels convert when they are loaded from RAM into the buffers of the RNS filter strings. That is, the input buffers of the RNS filter strings do not store 8-bit brightness characteristics of pixels. In the input buffers of the RNS-filter rows, a 16-bit packing of the remainders from division the 8-bit brightness value by the RNS moduli $\{7,15,31,16\}$. The RNS filter expends more memory, but it works faster because it does not waste time converting nine times the same numbers from BNS to RNS. At the end of the calculation using formula (2), both filtering modules (RNS filter and BNS filter) correct the response. If brightness characteristic of a pixel after calculation by formula (2) turned out to be less than zero then we set it equal to zero. If brightness characteristic of a pixel after calculation by formula (2) turned out to be more than 255 then we set it equal to 255. All intermediate calculations in the BNS filter are carried out over 16-bit numbers in the two-complement code. All intermediate calculations in the RNS filter are carried out over are performed on tuples from the equivalent BNS range $[-26040;26039]$. To simplify the hardware implementation of determining the sign of a number, all numbers exceeding 16384 in the two-complement 16-bit code after transfer from RNS to BNS. It narrows the output range to [-26040, 16383], but does not lead to fatal errors in solving the filtration problem.

In the general case, the division in the RNS is considered difficult to implement. In our implementation of the filtration process, the division is carried out in the final phase of the calculation after converting the result from RNS to binary number system (BNS) by extracting 8 bits of previously known positions. In this way, the division by denominator is carried out after conversion from RNS to BNS.

The operation of the created device is reduced to a large number of activations of the BNS filter modules and the RNS filter, calculating numbers according to formula (2). In the RNS calculation module, all calculations are carried out in parallel and independently for each of the bases of RNS. In addition, in the output stages of the RNS filter, a response is converted from the RNS to the MRC (on the bases 7, 15, 31) and further to the BNS. The finally calculated and corrected brightness characteristic are transferred from the calculation modules to the controller of the corresponding filter, which writes them into high-speed BRAM-buffers of the result of the filter. The possible delays in the filtering process do not mean anything to the RAM manager since the RAM manager reads data not from the buffer that the filter manager fills, but from the buffer complimentary to it. The RAM sector is responsible for consistent read/write of information between the pixel row buffers and the SDRAM controller (the read/write operation takes 3–6 clock cycles). The technology of reading/writing paired (even/odd) buffers is actively used. The information about which of the paired buffers is currently occupied by the corresponding device directly passes to the RAM manager which was implemented as a finite state machine.

V. Simulation results

Devices possessing technical characteristics presented in Table 1 were selected when designing a video processing device (filtering) with calculations in RNS. Alinx board AX309 is built on the basis of Xilinx Spartan6-xc6slx9 chip, SDRAM chip 256 Mbit and has the ability to directly connect a video camera OV7670, VGA display, AN430 portable display. Clock Generator AX309 sets the reference frequency 50 MHz, which was increased to 100 MHz by means of PLL microcircuits Spartan-6. As the development environment, the integrated environment Xilinx ISE 14.7 and hardware description language VHDL. The source texts of the modules of the device we created in the VHDL language are published on the Internet [11]. Tables 2 and 3 demonstrates the results of simulation. In this implementation, the RNS filter shows a 28% superiority in performance relative to the BNS filter. We assume that this result can be reproduced on devices with more advanced production technology.

TABLE I. MAIN TECHNICAL CHARACTERISTICS OF THE PROJECT

Device	Characteristic	Value
Video camera OV7670	Resolution	640×480
	Frequency	9 million pixel/sec
VGA display	Resolution	640×480
	Frequency	60 Hz
LCD display Alinx AN430	Resolution	480×272
	Frequency	30 Hz
Accumulator battery Li-Polymer	Capacity	10 A/h
	Battery life	8 hours
	Current	1 A
Alinx board AX309	Chip	Xilinx Spartan6-xc6slx9
	RAM Capacity	256 Mbit
	RAM Frequency	100 MHz

TABLE II. CHARACTERISTICS OF FILTERS IN VARIOUS NUMBER SYSTEMS

Module	Characteristic	Value
BNS filter	Bit depth	16
	Frequency	25 MHz
	Filter window size	128×240
	Range of filter coefficients	$[-64, 63]$
	Values of D	2^n, $n = 0,1,...,7$
	Number of processed pixels per minute	5517339
RNS filter	Bit depth	16
	Frequency	25 MHz
	Filter window size	128×240
	Range of filter coefficients	$[-64;63]$
	Values of D	2^n, $n = 0,1,...,7$
	Moduli set	$\{7,15,16,31\}$
	Number of processed pixels per minute	7649649

TABLE III. QUANTITATIVE TECHNICAL CHARACTERISTICS OF THE PROJECT

Characteristic	Value
Device Frequency	140 MHz
SDRAM Frequency	100 MHz
Number of Slice LUTs	3014
Number of Slice Registers	2545
Number of BRAM	24
Number of PLL	1

VI. Conclusions

We created a high-performance device, implements video processing at the hardware level with intermediate calculations in the RNS. This work has practical and scientific significance and it can be used as a platform for further scientific research of students, graduate students and engineers, which interested in the problem of building high-performance video processing solutions in the RNS.

References

[1] Abeydeera M., Karunaratne M., Karunaratne G. and Silva K. “4K Real-Time HEVC Decoder on an FPGA,” IEEE Transactions on circuirs and systems for video technology, Vol. 26, No. 1, January 2016.

[2] Loques O.G. and Kramer J. "Flexible fault tolerance for distributed computer systems," in IEE Proceedings E - Computers and Digital Techniques, vol. 133, no. 6, pp. 319-332, November 1986.

[3] Gonzalez R.C., Woods R.E. and Eddins S.L. “Digital Image Processing Using MATLAB,” Pearson Prentice Hall, 2003, 609 P.

[4] Bovik Al. “Handbook of image and video processing,” Texas: Elsevier, 2005, 1372 p.

[5] Pratt W.K. “Digital Image Processing,” Wiley-Interscience; 4 edition, 2007, 812 P.

[6] Vasalos E., Bakalis D. and Vergos H.T. “RNS Assisted Image Filtering and Edge Detection, Digital Signal Processing (DSP),” 2013 18th International Conference on, 1-3 July 2013, pp. 1-6.

[7] Cardarilli G.C., Nannarelli A. and Re M. “Residue number system for low-power DSP applications,” Proc. 41st Asilomar Conf. Signals, Syst., Comput, 2007, pp. 1412-1416.

[8] Chervyakov N.I., Molahosseini A.S., Lyakhov P.A., Babenko M.G. and Deryabin M.A. “Residue-to binary conversion for general moduli sets based on approximate Chinese remainder theorem,” International journal of computer mathematics, Vol. 94, No. 9, pp. 1833–1849, 2017.

[9] Zhang X., Sun H., Chen Sh. and Zheng N. “VLSI Architectuer Exploration of Guided Image Filtering for 1080P@60Hz Video Processing,” IEEE Transactions on circuirs and systems for video technology, Vol. 28, No. 1, January 2018.

[10] Kao Ch.-Ch., Lai J.-H. and Chien Sh.-Y. “VLSI Architecture Design of Guided Filter for 30 Frames/s Full-HD Video,” IEEE Transactions on circuits and systems for video technology, Vol. 24, No. 3, March 2014.

[11] https://github.com/anserion/ov7670_filter_rns