Implementation of FPGA-based camera video adapter for vehicle identification tasks based on EUR 13 classification

A V Ilyukhin¹, V S Seleznev¹, E O Antonova¹, M Y Abdulkhanova¹, E V Marsova¹

¹ Department of Automation of Production Processes, Moscow Automobile and Road Construction State Technical University, 64, Leningradsky Avenue, Moscow, 125319, Russia

Email: unit.ussr@gmail.com

Abstract. This article describes the first stage of the research work on the development of FPGA-based cameras for vehicle identification tasks being widespread in automated points of weight and dimensional control. Since FPGA represents an alternative to conventional processors, which features the ability to perform multiple tasks in parallel, a camera equipped with FPGA will be able to perform vehicle detection and identification functions simultaneously. Thus, the camera will perform not only the transmission of the image but also the transmission of the processing result, after which the server will only need to confirm or deny the results of the camera operation, which will significantly reduce the processing time of images from all automated points of weight and dimensional control. A simple VGA port board, a program of a static image for displaying it on a monitor in 640x480 resolution, and a program for a pixel counter were implemented in the development phase. EP4CE6E22C8 is used as FPGA, which power is more than enough to achieve the result.

1. Introduction

Using the example of automated points of weight and dimensional control in the road industry, it may be noted that conventional cameras of photo and video fixing are used instead of "smart" cameras. All stages of processing images received from cameras, including determining the registration number of the vehicle, identifying the vehicle according to the EUR 13 classification, etc., are performed on the servers of information processing centers, as well as in traffic management centers. The use of cameras equipped with a field-programmable gate array (FPGA) will reduce the number of computing operations on the server, reducing the load on it by implementing primary image processing directly on the camera itself [1]. The latest generation of gate arrays called FPGA (Field-Programmable Gate Array) has great potential for performing various image processing methods, such as threshold or adaptive binarization, and others [2].

Since for the task of identifying vehicles according to EUR 13 classification, it is necessary to perform all the stages of image processing, such as preprocessing, segmentation, detection, and post-processing on FPGAs, the first stage in the development of a camera equipped with FPGAs is to implement a VGA controller on FPGA.

2. Implementing VGA on FPGA

To visualize the image processing process, a VGA monitor is used, which is a matrix of pixels. The screen resolution is 640 horizontal by 480 vertical. The monitor operates through the sequential transmission of information on each pixel in RGB format, where each color (red, green, blue) is
transmitted as an analog signal. Since the monitor is not equipped with video memory and cannot remember information about each pixel, a continuous stream of information is required to monitor to save the image. The VGA sync frequencies are shown in Table 1.

Table 1. VGA sync frequencies

| Name          | Frequency          |
|---------------|--------------------|
| Vertical      | 60 Hz              |
| Horizontal    | 31.46875 KHz       |
| Pixel         | 25.175 MHz         |

The debug board on which the project is implemented is equipped with a quartz resonator with a vibration frequency of 50 MHz. Hence the first task arises: implementing a frequency divider to obtain the pixel frequency. Based on the value of the reference frequency, to get the pixel frequency equal to 25.175 MHz, it is necessary to multiply it by the division factor equal to 0.5035. The coefficient can be represented as a fraction, where the numerator is 5035, and the denominator is 10000. Reducing this fraction, the numerators are 1007, and the denominator is 2000. Since a meander is needed, the duration of the active level should be equal to the duration of the pause between individual pulses. The part of the VHDL code ensuring the frequency divider implementation is shown in Figure 1.

The next task is to get horizontal and vertical sweeps from the pixel frequency. Horizontal and vertical sweep time delays are presented in Tables 2 and 3.

Table 2. Horizontal delays

| Pixel plot                | Number of pixels | Time           |
|---------------------------|------------------|----------------|
| Horizontal visible area   | 640              | 25.422045680238 μs |
| Leading edge time         | 16               | 0.63555114200596 μs  |
| Synchronizing pulse       | 96               | 3.8133068520357 μs  |
| Trailing edge time        | 48               | 1.9066534260179 μs  |

Table 3. Vertical delays

| Pixel plot                | Number of pixels | Time           |
|---------------------------|------------------|----------------|
| Horizontal visible area   | 480              | 15.253227408143 ms |
| Leading edge time         | 10               | 0.31777557100298 ms  |
| Synchronizing pulse       | 2                | 0.063555114200596 ms  |
| Trailing edge time        | 33               | 1.0486593843098 ms |

The total number of pixels for horizontal scanning should therefore correspond to 800 for a time interval of 31.777557100298 μs, and for 525 pixels vertical in 16.683217477656 ms [3]. To simplify the task, horizontal pixels can be represented as an abscissa axis, while the vertical ones can be shown as an ordinate axis. Part of the Verilog code for setting the pixel count is shown in Figure 2.
It is possible to check the performance of the programs by connecting the oscilloscope to the output contacts (pins) of the horizontal and vertical sweep on the board to evaluate the frequency characteristics. The measurement results are shown in Figures 3 and 4.

Based on the oscilloscope readings, it turns out that the vertical and horizontal sweep often do not fall within the acceptable range. Consequently, in order to check the functionality of the program, a more accurate frequency counter is required, or it is required to generate an image and then transfer it directly to the monitor.

Based on this, the last task arises, which consists in generating a static image on board the FPGA with the subsequent transfer of the color scheme to the monitor. Since, according to the VGA standard, color is transmitted by an analog signal, and the FPGA operates on digital signals, it is necessary to develop the following electrical circuit through which it will be possible to obtain the analog values required for the VGA monitor operation from parallel digital signals [4]. The electrical circuit is shown in Figure 5, the computer model in Figure 6.
Now it is necessary to generate a static image in the form of three squares of different colors. Since the VGA monitor uses RGB additive color, it is necessary to define the areas of these squares. Since the visible area of the image is known to be 640 pixels horizontally and 480 pixels vertically, it is possible to calculate the vertices of three squares so that the result is represented by the coordinates of the beginning of the square for each color and its end. When implementing the electrical circuit shown in Figure 5, the RGB555 color transmission is used, which in turn means that 5 bits transmit the value of one color. The implementation procedure of the static image program is shown in Figure 7 and the result of the program is shown in Figure 8.
The device is designed using Intel's Quartus software. To simplify the program, it is better to use PLL (Phase-Locked Loop) as the main frequency divider. The number of used logic gates will therefore be reduced by an order of magnitude and will amount to only 91 logic gates, which is about 1% of the total number of FPGA Cyclone IV E (EP4CE6) FPGA gates.

3. Conclusion
Having performed the first stage of the research work on the development of an FPGA-based camera for vehicle identification tasks, which consists in implementing a VGA port board and writing a program for displaying a static image, as well as writing a pixel counter program, the following feature of connecting an FPGA to a VGA monitor was identified: since the monitor is not equipped with built-in video memory, there is a need for a continuous flow of information to preserve the image.

To obtain the required pixel frequency with a value of 25.175 MHz, a code was written that describes the reference frequency divider with a factor of 0.5035. Then, using the obtained frequency values, horizontal and vertical sweeps were extracted with a total number of pixels of 800 and 525 respectively.

The last work stage comprised the development of an electrical circuit converting a parallel digital signal into an analog signal, which is required for the operation of a VGA monitor and displaying a static image.
Acknowledgments
This research work was performed with the financial support of the Fund for Assistance to the Development of Small Forms of Enterprises in the Scientific and Technical Sphere within the framework of the UMNIK program in accordance with research project No. 15411GU/2020.

References
[1] Palagin A V, Yakovlev Yu S, Eliseeva E V 2012 Features of the approach to the choice of FPGAs for the design of PIM-systems Mathematical machines and systems 3 p 19-28
[2] Ilyukhin A V, Zaripova I I, Marsova E V, Seleznev V S 2018 Digital automation devices (Moscow: Moscow Automobile and Road Construction State Technical University (MADI))
[3] Ivashinnikov B A, Kovalenko M V, Leskov O M 2016 Implementation of VGA-interface based on FPGA of ALTERA Company "Young Scientist Publishing House" Limited Liability Company 17 p 121
[4] Strogonov A V, Babich D A 2012 Development of a VGA controller in the VHDL language in the QUARTUS II environment Bulletin of Voronezh State Technical University 6
[5] Gorodnichev M G, Marsova E V, Gematudinov R A, Dzhabrailov K A 2020 Technical vision for monitoring and diagnostics of the road surface quality in the smart city program E3S Web of Conferences 2019 Topical Problems of Green Architecture, Civil and Environmental Engineering Volume 164 03013
[6] Dzhabrailov K A, Gorodnichev M G, Gematudinov R A, Chantieva M E 2020 Development of a Control System for the Transportation of Asphalt Mix with the Maintenance of the Required Temperature VIII International Scientific Siberian Transport Forum. TransSiberia 2019. Advances in Intelligent Systems and Computing Volume 1116 Springer
[7] Ilyukhin A V, Marsov V I, Marsova E V, Syetina T A, Dzhabrailov K A 2019 Improving the efficiency of frozen soils excavation using high-frequency vibration IOP Conference Series: Materials Science and Engineering Volume 643
[8] Gorodnichev M G, Gematudinov R A, Dzhabrailov K A, Potapchenko T D 2019 The concept of an automated weight and size control system for measuring the mass of freight vehicles in a traffic flow (Weight-In-Motion) Systems of Signal Synchronization, Generating and Processing in Telecommunications, SYNCHROINFO IEEE
[9] Marsov V I, Gematudinov R A, Seleznev V S, Dzhabrailov K A 2019 Dependence of the Efficiency of the Distribution of Anti-icing Material on the Type of Suspension of the Weighing Conveyor Systems of Signals Generating and Processing in the Field of on Board Communications IEEE
[10] Baytesh K, Gematudinov R G, Chantieva M E, Dzhabrailov K A, Gorodnichev M G 2019 Development of an automatic method for determining vehicle dimensions through computer vision International Scientific Conference Energy Management of Municipal Facilities and Sustainable Energy Technologies Volume 1614 Journal of Physics: Conference Series