Kalpa Publications in Engineering
Volume 1, 2017, Pages 428–433
ICRISET2017. International Conference on Research and Innovations in Science, Engineering &Technology. Selected Papers in Engineering

Implementation of Edge Detection Algorithm Using Nexys 4 FPGA

C S Patel, R R Vashi, A A Vahora [1,2,3]
[1,2,3]Assistant Professor
[1,2,3] B.V.M. Engg. College
Anand, Gujarat, India
cspatel@bvmengineering.ac.in, ronak.vashi@bvmengineering.ac.in,
Anish.vahora@bvmengineering.ac.in

Abstract

Image processing requires extensive computation and usually done on Personal Computer or CPU. Due to its sequential processing method of Image processing or CPU task takes long time to get desire output. However FPGAs can be one of the options to speed up image processing without increasing the clock speed. One of the main features of FPGA is allowing parallel processing which speed up the processing of Image and get desire output in limited time-bound. In this paper, *Digilent Nexys 4* XC7A100T-1CSG324C FPGA is used to implement the edge detection operation on image. The Sobel algorithm is used to detect the edge in an image and is implemented on FPGA using Hardware Description Language (VHDL).

1 Introduction

Image processing is a technique to extract required information, improve raw image data. Image processing is broadly used in industries for quick quality testing, in security fields for threat monitoring and detection, and in entertainment field for production of high quality media. Mainly GPUs and CPUs are used for image processing, are mainly designed on the CPUs or GPUs. The processes on the CPUs or GPUs are quite slower compared to the FPGAs and ASICs [1].

The main aim of this paper is to design an FPGA based system that can perform image processing tasks faster than CPUs and GPUs. Captures image and detect the edges using the Sobel edge detection method. The system has a camera that captures image of the real world, FPGA board to process the Sobel operation and a monitor to display the output.

For this task required following kit and material.

- Digilent Nexys 4 FPGA Development Board
- Display
- Image Capturing Apparatus

A. Shukla, J.M. Patel, P.D. Solanki, K.B. Judal, R.K. Shukla, R.A. Thakkar, N.P. Gajjar, N.J. Kothari, S. Saha, S.K. Joshi, S.R. Joshi, P. Darji, S. Dambhare, B.R. Parekh, P.M. George, A.M. Trivedi, T.D. Pawar, M.B. Shah, V.J. Patel, M.S. Holia, R.P. Mehta, J.M. Rathod, B.C. Goradiya and D.K. Patel (eds.), ICRISET2017 (Kalpa Publications in Engineering, vol. 1), pp. 428–433

- Simulation and Design Software (Xilinx Vivado Design Suite)

Edge detection is method of detecting discontinuities or sharp intensity change. This is possible by simple derivation of intensity in 2D plane and detects maxima of derivative. There are several algorithms developed to do the same.

There are several edge detection methods (algorithms) [1]

- Sobel
- Prewitt
- Roberts
- Canny

These methods have been proposed for detecting transitions in intensity in various parts of images. Early methods determined the best gradient operator to detect sharp intensity variations. Commonly used method for detecting edges is to apply derivative operators on images. Derivative based approaches can be categorized into two groups, namely first and second order derivative methods of gradient and laplacian.

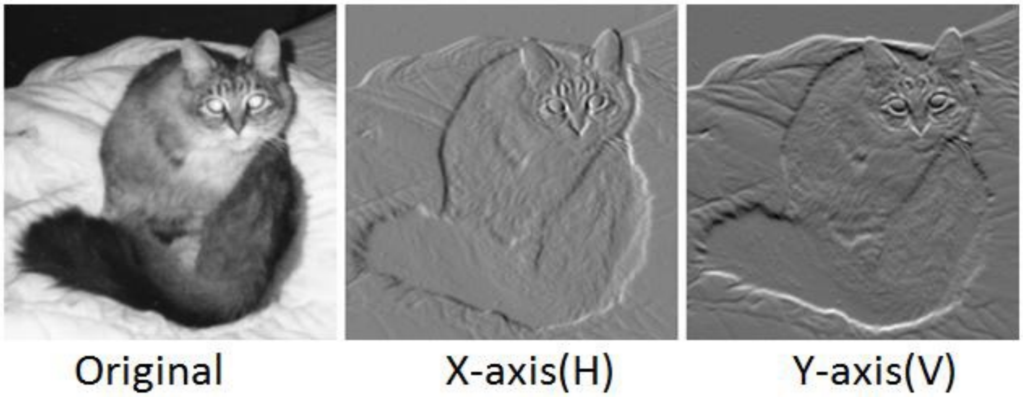


Figure 1: Sobel operator output of Horizontal and Vertical gradient**.**

In this paper, sobel edge detection technique is used. It has two masks, one along the horizontal direction and the other in vertical direction. These masks are generally 3x3 matrix. The image is convolved with only two kernels and this technique also has some smoothing effect to the random noise of the image. The image is convolved with the kernels estimating the gradient in the x-direction and the other in the y-direction.

2 Implementation On FPGA

The system architecture design is highly focused on parallel processing and minimum clock cycle utilization. In this implementation *Digilent Nexys 4* XC7A100T-1CSG324C FPGA is used.

The structural block diagram show the 5 main blocks of the process.

- BROM
- Sobel top module
- BRAM
- VGA display
- VGA timing block

The BROM and BRAM are designed using the IP core function of the Xilinx ISE suite. The grey level values of the image are stored in BROM. As no operation is to be performed on the image data within the memory, we have used read only memory (ROM), where we can just read the image data but not write or edit the image data. This image pixel values are feed as input to the sobel top module through the 8 bit data line.

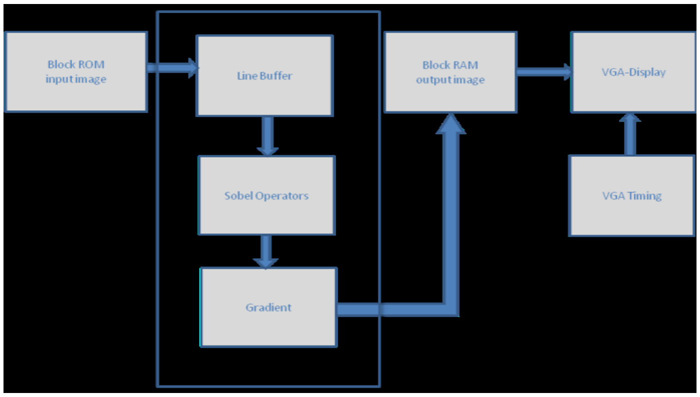


Figure 2: Structural Block Diagram

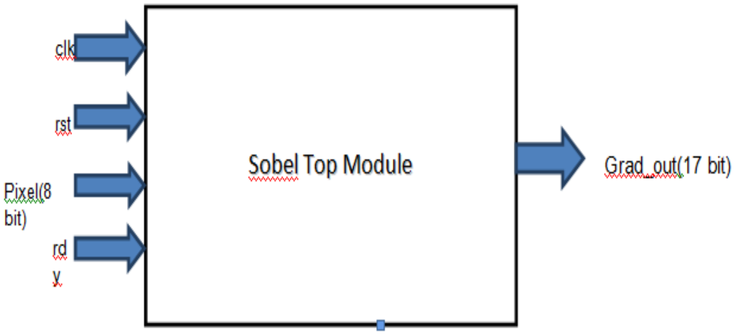


Figure 3: Sobel Top Module

The sobel top module is designed with four input lines and one output line. Out of these four input line one is, 8-bit data line that is used to provide image pixel data as input to the sobel operator via line buffer. The other three are the control lines – clk (clock input), rst (reset), rdy (ready). The clk is used for the clock input for the sobel top module, rst is used to reset the module and the rdy indicates that all the required data for the sobel operation are ready and the sobel operation can now be performed on that data. The ready signal used here is active high i.e. when the signal is high, it indicates that the image pixel data is ready .Here edge detection operation on 3x3 image matrix can be started as soon as the element at 3rd row and 3rd column is ready. Starting from this point, you can calculate the operations for every new incoming byte and generate the output for edge and direction. The output of this sobel top module is a 17 bit data that is further stored in the BRAM.

The sobel top module consists of, Line buffer block followed by sobel operator and later on gradient. The line buffer acts as a delay and generates sample space between the pixel values. These values of images are convolved with the sobel X and sobel Y operators and then the gradient is calculated to have the final output values of the edge detected image. In sobel operator block, we have generated a combinational logic for performing arithmetic operation, which provides gradient of the image as a result i.e. output. These output values are then stored in the BRAM.

The processed image is then displayed on a monitor through the VGA controller. A separate timing unit is designed for the control of the VGA controller. For the most common VGA mode

640x480 with the refresh frequency of the entire screen equal to 60 Hz, the timings for the synchronization signals are reported in the table below. In the current implementation of the VGA Driver is configured the VGA driver to display on monochrome pixel on 640x480 resolution at 60 Hz refresh rate. Pixel Clock is generated using LogiCORE IP Clock Generator (v4.03a) [4]. According to the standard Time specifications for this configuration, sync pulses are generated with an FSM, Pixel Data are fetched from the Block RAM and all are forwarded to the VGA output.

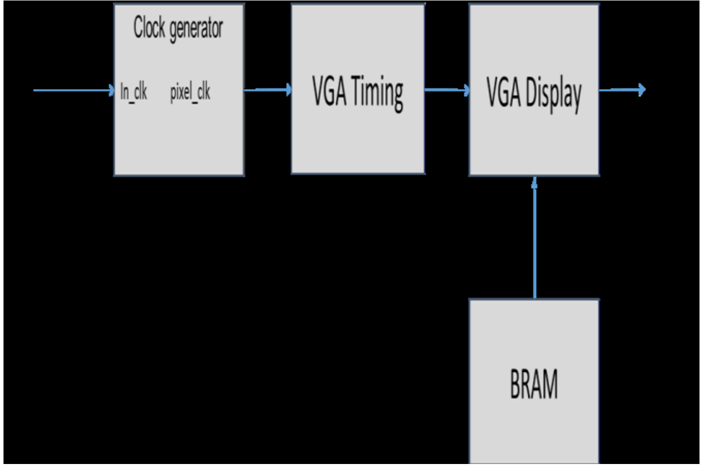


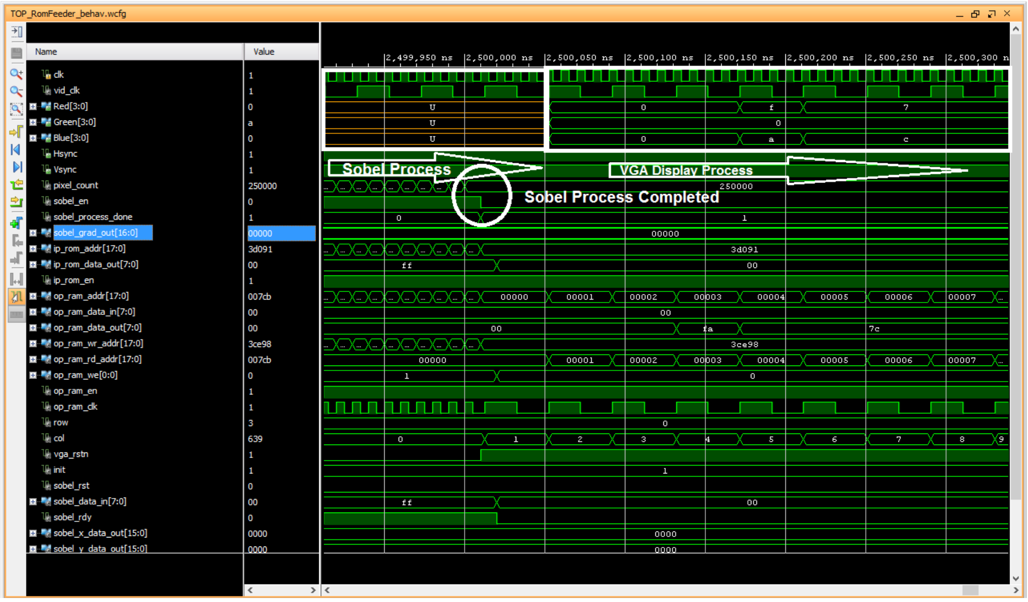
Figure 4: VGA Display Block Diagram

There are two blocks involved in the process of displaying the processed image as final output, one is VGA timing and the second one is VGA display. Here VGA operates at standard clock frequency 25.175 MHz as the image processing is carried out at 100 MHz at initial stage whereas the displaying process in VGA is done at 25.175 MHz; we need a PLL device in order to change the operating frequency. A phase-locked loop (PLL) is a voltage or current-driven oscillator that is constantly adjusted. PLL is used to control the frequency of the output signal. Hence, in our project we have reduced 100 MHz operating frequency to 25.175 MHz, which is the standard frequency requirement for VGA display.

The block including phase-locked loop is called Clock generator. Here VGA timing block gives the H_sync and V_sync as output, whereas VGA display gives R, G and B as its output. Here H_sync, V_sync, R, G and B initially are digital signals and in order to display our final image we need to convert R, G and B from digital to analog using a DAC converter. At the end the final processed i.e. edge detected image is displayed using monitors.

3 Simulation Results

From the Fig. 5 and Fig. 6 of simulation results, it is observed that image is captured from BRAM of FPGA and processed by Sobel module and then edge detected image display on VGA.


TOP_RomFeeder_behav.wcfg

Name	Value
clk	1
vid_clk	1
Red[3:0]	0
Green[3:0]	a
Blue[3:0]	0
Hsync	1
Vsync	1
pixel_count	250000
sobel_en	0
sobel_process_done	1
sobel_grad_out[16:0]	00000
ip_rom_addr[17:0]	3d091
ip_rom_data_out[7:0]	00
ip_rom_en	1
op_ram_addr[17:0]	007cb
op_ram_data_in[7:0]	00
op_ram_data_out[7:0]	00
op_ram_wr_addr[17:0]	3ce98
op_ram_rd_addr[17:0]	007cb
op_ram_we[0:0]	0
op_ram_en	1
op_ram_clk	1
row	3
col	639
vga_rstn	1
init	1
sobel_rst	0
sobel_data_in[7:0]	00
sobel_rdy	0
sobel_x_data_out[15:0]	0000
sobel_y_data_out[15:0]	0000

Figure 5: Simulation Results

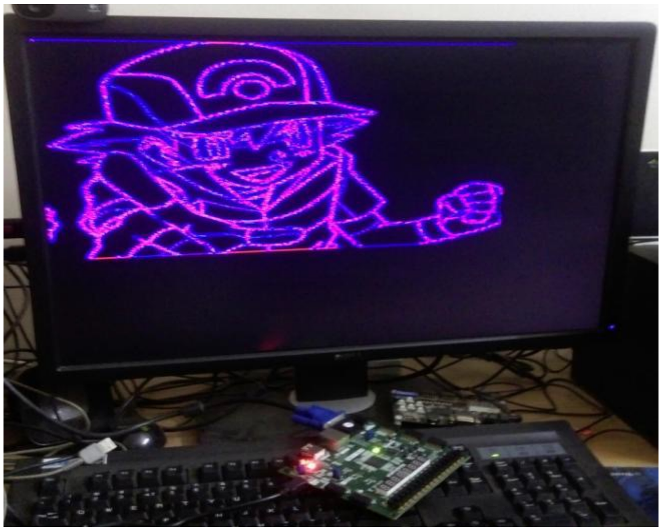


Figure 6: Edge detected and Hardware Implementation

4 Conclusion

For displayed an image on screen through VGA. Image pixel data stored in block ram which was displayed. This is primary block for any dedicated image processing application.

Data manipulation modules can be interfaced between the VGA output port and the image storage element. Such data manipulation modules are for image processing which are essentially digital signal processing blocks. These blocks incorporates image processing algorithms and processed image data can be buffered or directly given to output which is, in this case VGA. For fast image processing compared to CPU, GPU and DSP processor with scalable to SoC and cost effective with rapid development than ASIC.

In future, adding a module of contour detection and shape recognition to upgrade the Edge detection. Using Hardwired ARM Cortex A9 Embedded in Xilinx ZYNQ XC7Z010-1CLG400C with camera interface also make a complete SoC.

References

Maini R. and Aggarwal D.,"Study and Comparison of Various Image Edge Detection Techniques," International Journal of Image Processing, p. 12.

T. M. a. Y. Y. Shuichi Asano,(2009) "Performance Comparison of FPGA, GPU And CPU In Image Processing," IEEE, p. 6, 2009.

Xilinx, "IP Processor Block RAM (BRAM) Block (v1.00a) Data Sheet," 3 January 2011.

[Online Available: http://www.xilinx.com/support/documentation/ip_documentation/bram_block.pdf.

Xilinx, "Xilinx DS614 LogiCORE IP Clock Generator (v4.03a), Datasheet," 18 January 2012. OnlineAvailable:http://www.xilinx.com/support/documentation/ip_documentation/clock_generator/v4_03_a/clock_generator.pdf.

S. Larson,(2013) "VGA Controller (VHDL) - Logic - eewiki," 01 August 2013. [Online].Available:https://eewiki.net/pages/viewpage.action?pageId=15925278.

M. V. G. Rao, P. R. Kumar and A. M. Prasad (2016), "Implementation of real time image processing system with FPGA and DSP," 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), Durgapur, 2016, pp. 1-4.