International Journal of Advance Research in Engineering, Science & Technology(IJAREST),
ISSN(O):2393-9877, ISSN(P): 2394-2444,
Volume 2,Issue 8, August- 2015, Impact Factor: 2.125

AUTOMATIC NUMBER PLATE RECOGNITION AND ITS FPGA IMPLEMENTATION

Mr. Prasad Prakash Sutar[1], Prof. Pravin C. Latane[2],

[1]Department of Electronics and Telecommunication, Sinhgad Institute of Technology, Lonavala, Maharashtra, India.
prasad.sutar31@gmail.com
[2]Department of Electronics and Telecommunication, Sinhgad Institute of Technology, Lonavala,Maharashtra, India.
pclatane@gmail.com

Abstract

This paper reveals about the design and development of automatic number plate recognition [ANPR]. Since it is simpler and faster than the traditional system, it has all the potential to replace the existing system. In this system we are going to work on the video. Video captured by ANPR camera and processed using dynamic image processing technique. After that edge detection and template matching is done. Finally result is obtained in ASCII character which will be converted in alphanumeric character using MAT LAB. Number plate localization stage is very important to identify the license plate number in ANPR system. This system provides very low complexity. Also provide number plate detection rate is high by using canny edge detection algorithm. The proposed architecture has been successfully implemented and tested using Model-Sim 6 and Field Programmable Gate Array (FPGA) Altera Cyclone IV family development board. The proposed architecture will be implemented as a real time application for automated toll collection. It will be saves users valuable time by reducing the queue length in front of the toll counter.

Keywords*- Automatic License plate recognition, canny edge detection, Modelsim-6.3f and FPGA Altera Cyclone IV family.*

I. INTRODUCTION

Automatic license plate recognition has been widely used in intelligent transportation management systems. For some applications such as highway traffic monitoring and tracking specific cars in the highway, real time video processing is necessary, and for applications such as parking lots, automatic toll collection, automatic congestion charge systems, access control, tracing of stolen cars and identification of dangerous drivers, still image processing is satisfactory and applying it provides more accurate results.
The fundamental requirements of an ANPR system are image capture using an ANPR camera and processing of the captured image.

The image processing part, which is a computationally intensive task, includes three stages:

1] Number plate localization (NPL)
2] Character segmentation and
3] Optical character recognition (OCR)

NPL is the stage where the Number plate is detected in the input image from the ANPR camera. The character segmentation stage is an important pre-processing step before applying OCR, where each character from the detected Number Plate is segmented before character recognition so that only useful information is retained for recognition. In the last stage, optical character information will be converted into encoded text by pre-defined transformation model.

1. OBJECTIVE

The main goal of this work is to design and implement efficient and novel architectures for automatic number plate recognition (ANPR) system using FPGA. A separate ANPR algorithm is developed and optimized, by taking advantage of technical features of FPGAs which accelerate digital image processing algorithms. The investigation of the algorithm and its optimization focused on real time image and video processing for license plate (LP) or number plate localization (NPL), LP character segmentation (NPS) and optical character recognition (OCR) in particular, which are the three key stages of the ANPR process. ANPR often forms part of an intelligent transportation systems. Its applications include identifying vehicles by their number plates for policing, control access and toll collection.

2. BLOCK DIAGRAM

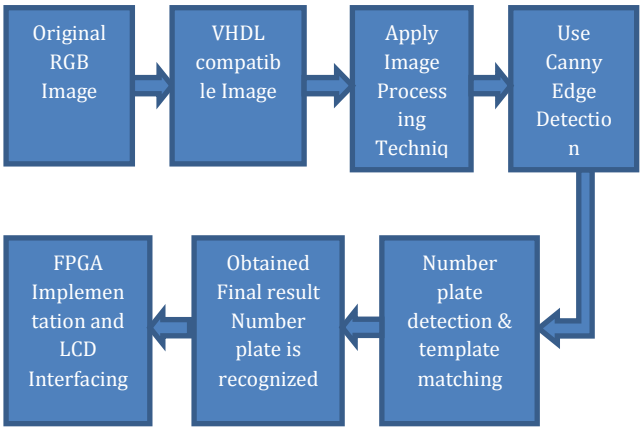


Figure 1: System Block Diagram

Above Figure 3.1.shows that System Block Diagram. This diagram shows the system overview. Original Car number plate video captured by ANPR camera. Then this video gets processed by MATLAB with rows, column and no of frames. After that we get VHDL compatible image with grayscale or binary image format. This image then processed by applying image processing techniques. Here canny edge detection algorithm is applied for edge detection.
Then Number plate detection or template matching procedure is done. Once this procedure is done we get VHDL output i.e. nothing but final result is obtained Number plate recognition is done. But this output is in ASCII character format. Now this output can be converted into alphanumeric character with the help of MATLAB. Now all the system code dumps in to the FPGA development board DE2-115 Cyclone IV family. LCD module is getting interfaced to DE2-115 board to obtain recognised number plate. Recognised number which can be displayed on LCD module. This is the whole system overview.

II. METHODOLOGY

This project aims to focus on the image processing algorithm in ANPR system which is simulated in Mat lab software. Images that are taken should be clear enough to be processed and should not contain any defects in the number plate for example missing characters. Number plate area is identified using colour contrast method. This is because number plates in India are in black and white. First, the upper half of the image is removed as it is found out that the number plate occurs at the bottom half of the image. By comparing the neighboring pixels horizontally and vertically, the summation of difference between neighboring pixels for each column and row is computed. These signals are then passed through a filter that removes values that are less than the average of the signal. Few candidates for the number plate area are then detected. The area with the highest contrast is then identified as the number plate area. The area is then cropped out. Here canny edge detection algorithm is used for edge detection.

After segmenting the number plate area, the characters are then recognized through OCR by number plate recognition algorithm. First, the image is converted to black and white. Then, objects that are bigger or smaller than the characters in the number plates are removed. The characters are then separated individually. Each character is then compared with a set of template. The template with the highest correlation coefficient corresponding to the image defines its identity.

1. CANNY EDGE DETECTION

Among the edge detection methods proposed so far, the canny edge detector is the most rigorously defined operator and is widely used. The popularity of the Canny edge detector can be attributed to its optimality according to the three criteria of good detection, good localization, and single response to an edge.

Figure 2. Original Image

Figure 3. Edge Detected Image

Although the optimization process described by Canny rests on solid grounds, its simple approximation,
which almost all implementations are based on, has some problems. A typical implementation of the Canny edge detector follows the steps below.

1. Smooth the image with an appropriate Gaussian filter to reduce desired image details.
2. Determine gradient magnitude and gradient direction1 at each pixel.
3. If the gradient magnitude at a pixel is larger than those at its two neighbors in the gradient direction, mark the pixel as an edge. Otherwise, mark the pixel as the background.
4. Remove the weak edges by hysteresis thresholding.

2. FEATURES OF EDGE DETECTION

- ☐ 1. Smoothing: Blurring of the image to remove noise.
- ☐ 2. Finding gradients: The edges should be marked where the gradients of the image has large magnitudes.
- ☐ 3. Non-maximum suppression: Only local maxima should be marked as edges.
- ☐ 4. Double thresholding: Potential edges are determined by thresholding.
- ☐ 5. Edge tracking by hysteresis: Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

III. SOFTWARE IMPLEMENTATION

1. Requirement of Software

- ✓ MATLAB
- ✓ MOELSIM SE 6.3f
- ✓ ALTERA QUARTUS II 10.1

Dynamic Image processing i.e. Video processing part is carried out toolbox of MATLAB and Modelsim SE 6.3f while hardware implementation part is carried out by writing code in Altera Quartus-II software.

2. Altera Quartus- II

The Altera Quartus II design software provides a complete, multiplatform design environment that easily adapts to your specific design needs. It is a comprehensive environment for system-on-a-programmable-chip (SOPC) design. The Quartus II software includes solutions for all phases of FPGA and CPLD design

The Quartus II software includes a modular Compiler. The Compiler includes the following modules (modules marked with an asterisk are optional during a compilation, depending on your settings):

- ✓ Analysis & Synthesis
- ✓ Partition Merge
- ✓ Fitter
- ✓ Assembler
- ✓ Time Quest Timing Analyzer
- ✓ Design Assistant
- ✓ EDA Netlist Writer
- ✓ Hard Copy Netlist Writer

IV. HARDWARE IMPLEMENTATION

1. HARDWARE REQUIREMENT

- ✓ ALTERA Cyclone IV family DE2-115 board
- ✓ LCD module

Figure 4. ALTERA Cyclone IV family DE2-115 board with LCD module

2. FPGA SPECIFICATIONS

In present work we have used Altera's DE2-115 FPGA board from Cyclone-IV family for implementation purpose. The DE2-115 board has many features that allow the user to implement a wide range of designed circuits, from simple circuits to various multimedia projects.

The following hardware is provided on the DE2-115 board:

- ✓ Altera Cyclone® IV 4CE115 FPGA device
- ✓ Altera Serial Configuration device – EPCS64
- ✓ USB Blaster (on board) for programming; both JTAG and Active Serial (AS) Programming modes are supported
- ✓ 2MB SRAM
- ✓ Two 64MB SDRAM
- ✓ 8MB Flash memory
- ✓ SD Card socket
- ✓ 4 Push-buttons
- ✓ 18 Slide switches
- ✓ 18 Red user LEDs
- ✓ 9 Green user LEDs
- ✓ 50MHz oscillator for clock sources
- ✓ 24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks
- ✓ VGA DAC (8-bit high-speed triple DACs) with VGA-out connector
- ✓ TV Decoder (NTSC/PAL/SECAM) and TV-in connector
- ✓ 2 Gigabit Ethernet PHY with RJ45 connectors
- ✓ USB Host/Slave Controller with USB type A and type B connectors
- ✓ RS-232 transceiver and 9-pin connector
- ✓ PS/2 mouse/keyboard connector
- ✓ IR Receiver
- ✓ 2 SMA connectors for external clock input/output
- ✓ One 40-pin Expansion Header with diode protection
- ✓ One High Speed Mezzanine Card (HSMC) connector
- ✓ 16x2 LCD module

3. LCD SPECIFICATIONS

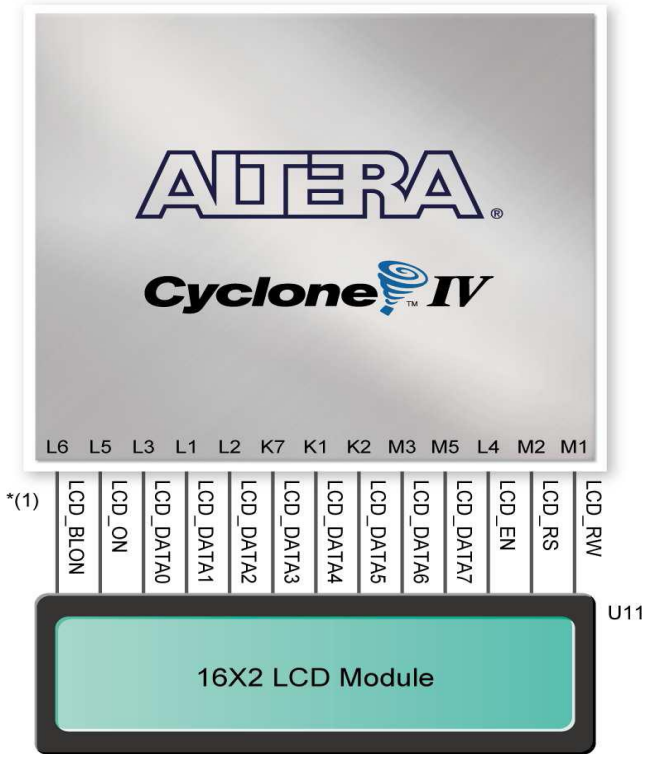


Figure 5. Connections between the LCD module and Cyclone IV FPGA

Table 1. Pin assignments for the LCD module

Signal Name	FPGA Pin No.	Description	I/O Standards
LCD_DATA[0]	PIN_L3	LCD Data[0]	3.3V
LCD_DATA[1]	PIN_L1	LCD Data[1]	3.3V
LCD_DATA[2]	PIN_L2	LCD Data[2]	3.3V
LCD_DATA[3]	PIN_K7	LCD Data[3]	3.3V
LCD_DATA[4]	PIN_K1	LCD Data[4]	3.3V
LCD_DATA[5]	PIN_K2	LCD Data[5]	3.3V
LCD_DATA[6]	PIN_M3	LCD Data[6]	3.3V
LCD_DATA[7]	PIN_M5	LCD Data[7]	3.3V
LCD_RW	PIN_M1	LCD Read/Write Select, 0 = Write, 1 = Read	3.3V
LCD_EN	PIN_L4	LCD Enable	3.3V
LCD_RS	PIN_M2	LCD Command/Data Select, 0 = Command, 1 = Data	3.3V
LCD_BLON	PIN_L6	LCD Back Light ON/OFF	3.3V

4. SYSTEM FLOW CHART

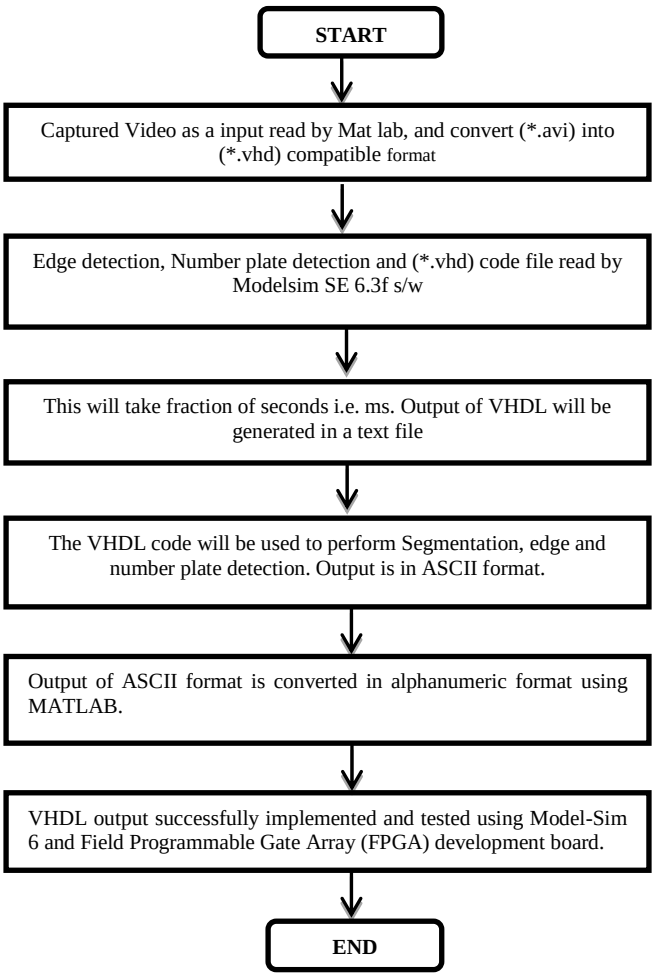


Figure 6. System Flow Chart

The overview of has following steps:

- ✓ **Image Capture:** Capture the Video of the vehicle.
- ✓ **Plate localization:** Responsible for finding and isolating the plate on the picture
- ✓ **Character segmentation and Edge detection:** Finds the individual characters on the Plates.
- ✓ **Database comparison:** Comparing segmented characters with Database characters.
- ✓ **Optical character recognition:** Recognizes individual characters through index values.
- ✓ **Authentication:** It authenticates the legal license plates.

V. RESULT

1. MATLAB AND MODELSIM 6.3f SIMULATION

Figure 7. Input Video Image

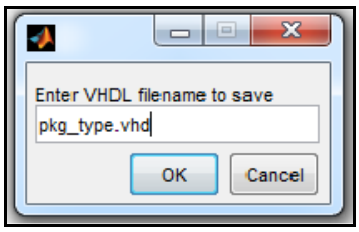


Figure 8. Enter File Name

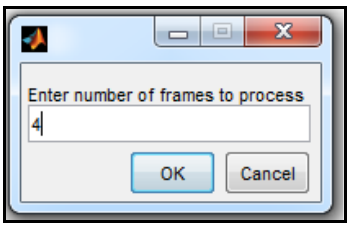


Figure 9. Enter No of Frames

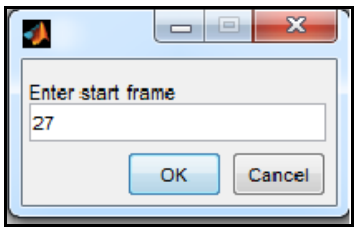


Figure 10. Enter start of Frame

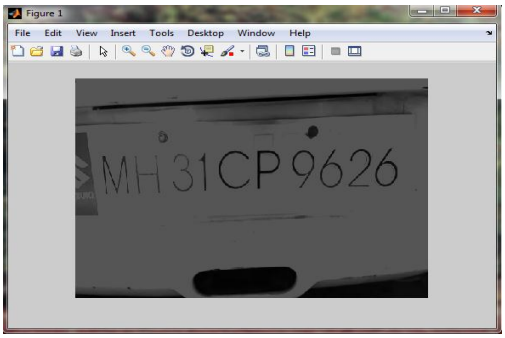


Figure 11. VHDL Compatible Image

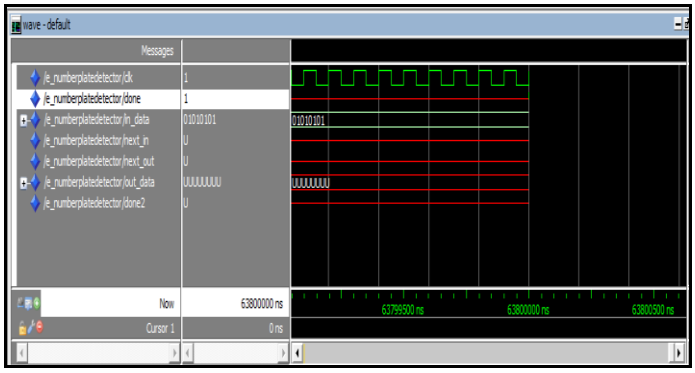


Figure 12. Simulation Result

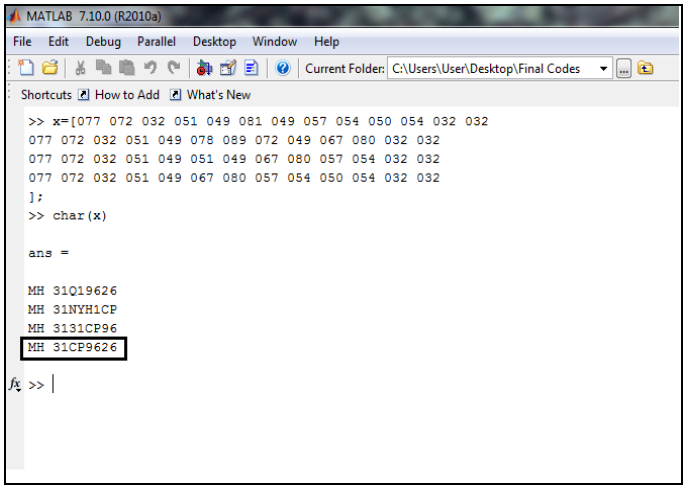


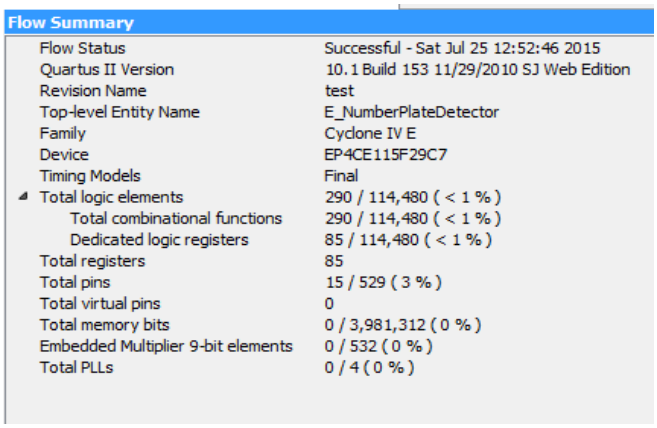
Figure 13. Detected Number Plate

2. FPGA IMPLEMENTATION

Figure 14. Recognized Number on FPGA platform

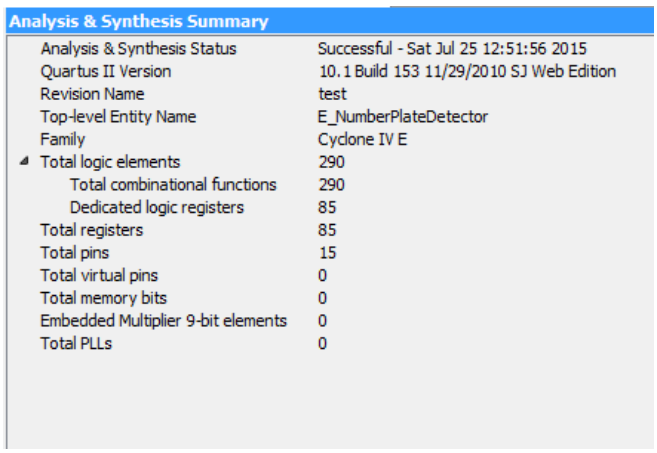
3. RESOURCE UTILIZATION OF FPGA

Resource utilization of Altera's Cyclone-IV FPGA board is shown in figure below:



Flow Summary	
Flow Status	Successful - Sat Jul 25 12:52:46 2015
Quartus II Version	10.1 Build 153 11/29/2010 SJ Web Edition
Revision Name	test
Top-level Entity Name	E_NumberPlateDetector
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	290 / 114,480 (< 1 %)
Total combinational functions	290 / 114,480 (< 1 %)
Dedicated logic registers	85 / 114,480 (< 1 %)
Total registers	85
Total pins	15 / 529 (3 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	0 / 4 (0 %)

Figure 15. Resource utilization.



Analysis & Synthesis Summary	
Analysis & Synthesis Status	Successful - Sat Jul 25 12:51:56 2015
Quartus II Version	10.1 Build 153 11/29/2010 SJ Web Edition
Revision Name	test
Top-level Entity Name	E_NumberPlateDetector
Family	Cyclone IV E
Total logic elements	290
Total combinational functions	290
Dedicated logic registers	85
Total registers	85
Total pins	15
Total virtual pins	0
Total memory bits	0
Embedded Multiplier 9-bit elements	0
Total PLLs	0

Figure 16. Analysis and Summary

VI. CONCLUSION

- ✓ In this thesis, video processing on FPGA was studied. Three main steps of the work i.e. design, simulation and implementation, are accomplished. For design and simulation, VHDL design language is used due to its closer structure to software.
- ✓ The main distinctive feature of the plate, i.e. high contrast variation, is used to detect the location of the plate on the image.
- ✓ The applied algorithms are tested with static images and video streams both in MATLAB and FPGA. In order to provide more accurate and realistic results, images taken in different time intervals and different weather conditions are tested.
- ✓ This project reveals about the design and development of automatic number plate recognition [ANPR]. Since it is simpler and faster than the traditional system.
- ✓ In this system we have worked on the video. Video captured by ANPR camera and processed using dynamic image processing technique. After that edge detection and template matching is done using MODEL SIM-6 . Finally result is obtained in ASCII character which can be converted in alphanumeric character using MAT LAB.
- ✓ Number plate localization stage is very important to identify the license plate number in ANPR system. This system gives very low complexity.
- ✓ Also provide number plate detection rate is high by using canny edge detection algorithm.
- ✓ The proposed architecture has been successfully implemented and tested using Model-Sim 6, Altera Quartus II 10.1 and Field Programmable Gate Array (FPGA) Altera cyclone IV family DE2-115 development board.
- ✓ The proposed architecture will be implemented as a real time application for automated toll collection. It will be saves users valuable time by reducing the queue length in front of the toll counter.

Use footnotes sparingly (or not at all) and place them at the bottom of the column on the page on which they are referenced. Use Times 8-point type, single-spaced. To help your readers, avoid using footnotes altogether and include necessary peripheral observations in the text (within parentheses, if you prefer, as in this sentence)

.

VII. REFERENCES

[1] M. Ashourian, N. Daneshmandpour, O. Sharifi Tehrani, P. Moallem "Real Time Implementation of a License Plate Location Recognition System Based on Adaptive Morphology" IJE TRANSACTIONS B: Applications Vol. 26, No. 11, (November 2013) 1347-1356

[2] Xiaojun Zhai, Faycal Bensaali, Soodamani Ramalingam "Improved number plate localization algorithm and its efficient field programmable gate arrays implementation" IET Circuits Devices Syst., 2013, Vol. 7, Iss. 2, pp. 93–103

[3] Zoe Jeffrey, Xiaojun Zhai, Faycal Bensaali, Reza Sotudeh and Aladdin Ariyaeeinia "Automatic Number Plate Recognition System on an ARM-DSP and FPGA Heterogeneous SoC Platforms"University of Hertfordshire, Hatfield, UK, AL10 9AB.

[4] X. Chen and A. L. Yuille. Detecting and reading text in natural scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages II–366–II–373, 2004.

[5] V. Shapiro, G. Gluhchev, and D. Dimov. Towards a multinational car license plate recognition system. Machine Vision and Applications, 17(3):173–183, August 2006.

[6] X. H. W. Jia, H. Zhang and M. Piccardi. Mean shift for accurate license plate localization. In Proceedings of the

IEEE Conference on Intelligent Transportation Systems (ITSC), pages 566–571, 2005.

[7] J. Matas and K. Zimmermann. Unconstrained license plate and text localization and recognition. In Proceedings of the IEEE Conference on Intelligent Transportation Systems (ITSC), pages 572–577, Vienna, Austria, September 2005.

[8] L. Dlagnekov and S. Belongie. Recognizing cars. Technical Report CS2005-0833, UCSD University of California, San Diego, 2005.

[9] C. Rahman, W. Badawy, and A. Radmanesh. A real time vehicle's license plate recognition system. In Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS), pages 163–166, 2003.

[10] R. Owens, "Lecture 6", Computer Vision IT412, 10/29/1997.

[11] S. Price, "Edges: The Canny Edge Detector", July 4, 1996.

[12] M.B. Ahmad and T.S. Choi , Local Threshold and Boolean Function Based Edge Detection, IEEE Transactions on Consumer Electronics, Vol. 45, No 3. August 1999.

[13] APEX 20K Programmable Logic Device Family. Data Sheet, February 2002, ver. 4.3, 1-116, http://www.altera.com

[14] Quartus II - Programmable Logic Design Software. January 2002, ver.2.0, 1-45, http://www.altera.com