International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Special Issue 31(September, 2015), PP. 255-260

AUTOMATIC VEHICLE COUNT AND CLASSIFICATION USING FPGA

Ekta K Patil **(1) ,** ***AssistantProf*** **Shweta Ashtekar (2)**, ***AssistantProf*** **Amruta Chintawar (3)**
Electronics Department , Mumbai University
RAIT College of Engineering , Nerul, Navi Mumbai,Maharashtra,India
ektapatil19@gmail.com, shweta_sa06@yahoo.com, amruta-chintawar239@gmail.com

Abstract *- The paper presents an advance technique of traffic maintenance for vehicle counting and classification, using infrared sensor emulator module. The infrared sensor emulator module used contains the binary image patterns of the vehicle stored in it and is controlled by the field programmable gate array (FPGA) system. The selected pattern are processed and analyzed by FPGA to give the final count and the classification of the vehicle. Vehicle classification is done on the basis of physical parameters of the vehicle.*

Keywords—FPGA, IR emulator ,RLE algorithm

I. INTRODUCTION

A vehicle count along a particular road is done either electronically or manually. It keeps a count of vehicles for a specific highway segment. Along with this the vehicles can be classified according to their classes. Vehicle counts can be used by traffic controlling authority to identify which routes are used most, and to either improve that road or provide an alternative if there is an excessive amount of traffic. To maximize the capacity of highways for instance, the traffic operators can use variable speed limits in function of traffic volume to prevent traffic jams from occurring and instead keeping the traffic flowing at lower speeds.[1] Continuous traffic monitoring efforts are carried out in several countries to understand seasonal, day of-week, and time-of-day traffic volume patterns. Traffic parameters such as number of vehicles, traffic density are widely used by traffic controlling systems. Also vehicle class is required for access control and statistic purpose. It can also be used at the toll collection system which determines the configuration of the vehicle for the purpose of charging the appropriate toll to the user. Automatic vehicle counter-cum-classifier (AVCC) system refers to the various components and processes of the toll collection system with which the toll equipment is able to determine the configuration of the vehicle [2]. An AVCC system consists of sensor devices installed in a lane to record the physical characteristics of vehicles and a processing unit to aggregate the input from the various sensor devices and interpret this input to assign a class to each vehicle passing through the lane. A vehicles class can be determined on the basis of the physical attributes of the vehicle such as number of axles, length and height of the vehicle etc. This system provides a low cost means of recording vehicle classification without interruption to traffic flows. Because of an increased requirement for information on traffic flows from a variety of sources sought by national traffic authorities, law enforcing authorities and consultants undertaking traffic studies in connection with congestion reduction, traffic management and development plans, it has been necessary to use AVCC systems on highways and at different toll-booths.

II. BASIC OF METHODOLOGY

The scope for dissertation has been proposed to Design a portable automatic vehicle counter and classifier System for the vehicles passing on a designated section of the road. Further this system should meet the accuracy in classifying to levels better than 95 percent. The proposed work contains following steps- To accept binary image as input from the IR sensor emulator. To apply appropriate algorithms implemented in FPGA on input binary image to calculate different vehicle parameters. Carry out the counting as well as classification of the vehicles. A vehicles class can be determined on the basis of the physical parameters of the vehicle such as number of axles, length of the vehicle body and the height of the vehicle or a combination of these parameters. Determinants of vehicle class include following features. Total length of the vehicle body. Maximum height of the vehicle. Percentage length corresponding to maximum height. Total number of axles. These physical parameters can be calculated from the input pattern applied by means of image processing algorithms like run length encoding (RLE) which performs the operation of counting the consecutive (i.e.) running lengths of 0s and 1s in the given binary sequence.[4][7]

III. VEHICLE CLASSIFICATION TECHNIQUE

Vehicles are classified by the physical parameters of the like number of axles spacing between the axle length of the body and the height of the vehicle combination of these parameters.

Features in figure 1 are required for classification of vehicles - 1.Length of the vehicle. 2. Height of the vehicle. 3. Percentage of total length/Length of the maximum height. 4. Distance between start of vehicle and front axle. 5. Distance between rear axle and end of vehicle. 6. Distance between two axles. 7.

Total number of axles.[2]

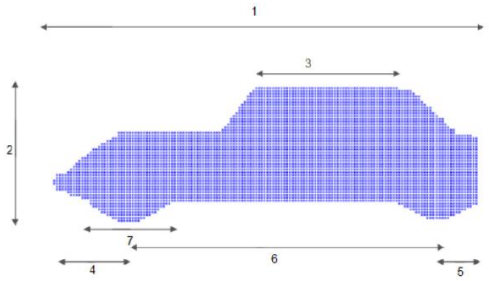

Figure 1 Feature extraction

IV. BASIC WORKING OF AVCC SYSTEM

In this system we are going to count the number of vehicle and classify them according to their classes as described above by considering various parameters. Figure 2 shows the basic working of AVCC. Main part of system consists of Infra-red sensor emulator which contains pre stored binary image of different vehicles for various classes, FPGA is used for processing the received binary image and it classifies the vehicle and count of vehicle, computer system is used for storing the data as a main database.[9]

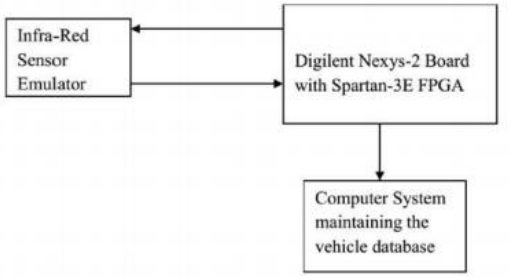

Figure 2 Basic working of AVCC

.

V. ARCHITECTURE FOR AVCC ON FPGA

A. Block level architecture for hardware on FPGA:

Infra-red sensor emulator is used which contains pre stored binary image of different vehicles for various classes. IR emulator gives the image of vehicle it sends data serially via serial port. Data received is in the form of raw binary image. This raw binary image is stored in buffer memory and then to RAM designed on FPGA.

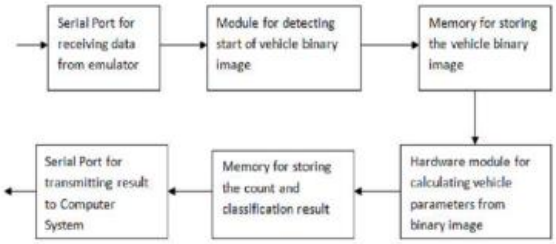

Figure 3 Block level architecture for hardware on FPGA

We design hardware module for calculating vehicle parameters as discussed above Features extracted are stored in main memory and transmitted serially to the computer system which stores the complete data base of vehicles.

B. Hardware architecture for AVCC on FPGA:

Data when gets received to IR emulator it sends to the buffer memory via UART receiver as shown in figure 4. The data send is in the form of raw binary image. This raw binary image is stored in the ten bytes RAM. RAM stores one sequence at a time. This sequence is further send to the storage memory that is hundred bytes RAM where the sequence data gets stored in every location. Then further proceeding is carried out for different parameters. Height of each sequence is stored in the memory. Length of vehicle is calculated by considering the start of vehicle sequence and end of sequence in raw binary image of the vehicle.

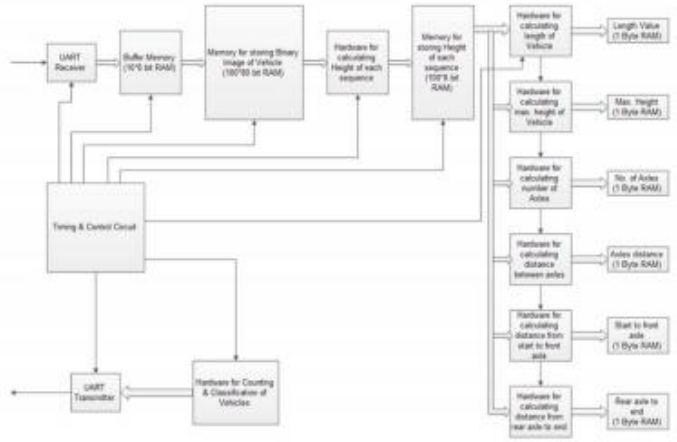

Figure 4 Hardware architecture for AVCC on FPGA

Number of axles are counted when the sequence gets constant with rise and fall on the same sides this indicates the presence of the axle. Once axles are calculated then the length between two axles is calculated. After the axle length calculation distance between start of vehicle and front axle and distance between rare axle and end of vehicle is calculated by knowing all this parameters total length of the vehicle is obtained. Thus by knowing height length and axles we can easily classify the vehicles.

C. IR sensor emulator details:

IR Sensor Emulator is a device that contains the binary side profiles of different classes of vehicles stored in the form of data bytes into its memory. Each vertical sequence of the vehicle is stored in 16 bytes of data in emulator. This data stored in an emulator, when triggered by external control keys, serially transmits the data corresponding to the side profile of selected vehicle. From 16 bytes, the first byte is ”0x7B” which indicates a start of new sequence. The last means 16th byte is ”0x7D” which indicates end of current sequence. 10 data bytes starting from 2nd byte to 11th byte represent actual side profile of the vehicle. In side profile of vehicles stored in emulator, each vertical sequence consists of 80 bits which require 10 bytes of data. 12th byte in the sequence contains crucial information regarding the presence of vehicle. The LSB of 12th byte indicate whether a vehicle is present during that particular sequence or not. It this bit is ’0’ then it indicates that vehicle is present, and if it is ’1’ then vehicle is not present. Thus this bit can be used to check presence of vehicle. This saves the calculation efforts by a system of FPGA. The calculations can be applied on received on sequence only when this bit is high, eliminating the necessity to perform calculation during all sequences. This bit has important significance. The transition from ’0’ to ’1’ indicates that vehicle has passed, and this event can be used ’timing and control unit’ to trigger the hardware unit for vehicle classification. 13th, 14th and 15th bytes do not contain any significant data; hence these bytes can be ignored from the data required for calculations and classification. Consider the following image which shows the data transmitted by the emulator, which is serially received on a computer system. The tool used for this operation is ’Docklight’. The emulator transmits the data serially with a fixed baud rate of 57600 bits/sec. The ’docklight’ tool has a facility to transmit and receive data serially from and to the external devices with serial communication. The baud rate in the tool is set equal to that of emulator i.e. 57600 bits/sec. The received data is analyzed for the pattern in which the data is transmitted by emulator.

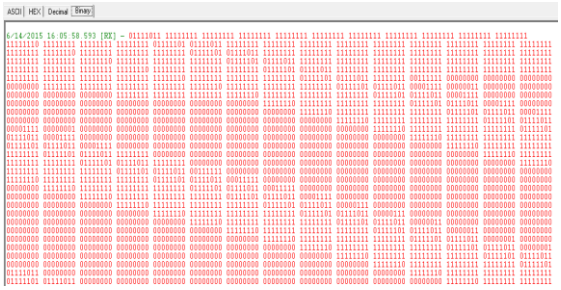

Figure 5 Data bytes received from IR sensor emulator

After removing start byte, stop byte and other unimportant data bytes, the extracted data bytes contain the bytes corresponding to side profile of the present sequence. When bits in these data bytes are extracted and arranged in appropriate order, the obtained binary sequence is a side profile corresponding to the current vertical sequence, which is 80 bit long.

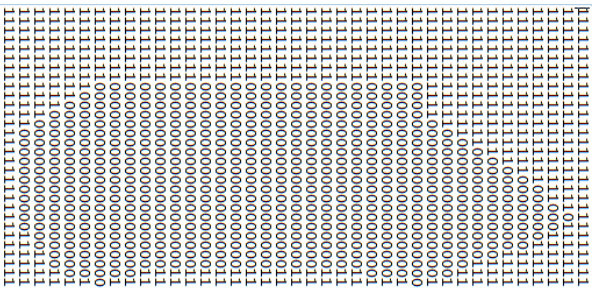

Figure 6 Binary side profile of the light vehicle

Consider figure 5 which represents such appropriately arranged sequences for typical light vehicle The above image i.e figure 6 is obtained by plotting the binary sequences of the side profile of the typical light vehicle in the order they are received from an emulator.

VI. IMPLEMENTATION

The hardware platform used for implementation of the AVCC system is field programmable gate array (FPGA). The spartan-3E family of FPGA is a superior alternative to ASICs. FPGAs avoid the high initial cost, the lengthy development cycles, and the inherent inflexibility of conventional ASICs. Also, FPGA programmability permits design upgrades in the field with no hardware replacement necessary. The device used in the dissertation is Spartan-3E XC3S500E-FG320. The features of this device are as follows. 1. Very low cost, high-performance logic solution for high-volume, application specific development. 2. Multi-voltage, multi-standard I/O pins (3.3V, 2.5V, 1.8V, 1.5V, and 1.2V signaling). 3. 500,000 system gates available on-chip which is equivalent to around 10,000 logic cells. 4. 73Kbits of Distributed RAM and 360Kbits of Block RAM which provides efficiency in memory operations. 5. 20 Dedicated Multipliers available on-chip which can be utilized by system developed. 6. 4 Digital Clock Multipliers which provides clock-skew elimination, frequency synthesis, multiplication and division. The hardware units in the system are first implemented using VHDL code. This code defines the functionality of the hardware block. Once the code is checked for its syntax, the same is synthesized in Xilinx tool. The process of synthesizing creates a net-list of logic gates for the hardware blocks. Once this process is complete it generates entity level and RTL schematic diagrams for the hardware unit. Then designed system is checked for its functionality. Once this process is successful, the system is implemented using same Xilinx tool. The bit-stream generated after implementation is downloaded into FPGA, which then performs the operation same as that of system designed. The software tool used for implementing the system is FPGA development tool from Xilinx- which is a tool developed by Xilinx and supports system development for large variety of

FPGAs from different manufacturers. This tool supports FPGA based system development right from a design entry (by means of schematic, state flow or HDL code) up to the generation of bit-stream file which is to be downloaded into FPGA in order for the device to function as the desired system. Consider Figure which represents the typical flow of FPGA based system development. The process starts with entering the design of the desired system. The design can be entered in various forms like schematic entry, state flow entry or Hardware Description Language (HDL) entry. In this project the design entry is done with the help of HDL entry using VHDL language codes. While designing digital circuits, they are usually described at high level of abstraction than a transistor level. It declares the registers and the combinational logic by using constructs in VHDL. This level is called Register- Transfer Level (RTL).Using synthesis tools, this description is converted into gate-level description of the circuit. Synthesis also performs the optimization of logic of the circuit. The synthesis results are used by placement and routing to carry out physical design of the system. It is also used by simulation tools which are used for checking the correctness of the functionality of the system designed.

VII. RESULT

A. Synthesis and simulation of emulator based system:

This section explains the implementation of the system using emulator. The implementation of this system involves defining entity, defining functionality of each hardware unit through VHDL code, synthesizing, simulating, floor-planning and placement-routing of the hardware unit. The process also generates a report which gives the resource utilization summary.

1) Timing and control unit*:* This simulation waveform as shown in figure 7 is obtained by writing a test-bench for the designed hardware unit, and simulating it with different input patterns applied. From figure it is clear that the unit generates a divided clock required for serial communication. Also it generates different signals to trigger different other hardware units. The synthesis tool in Xilinx design environment also generates a summary report for utilization of resources available on FPGA.

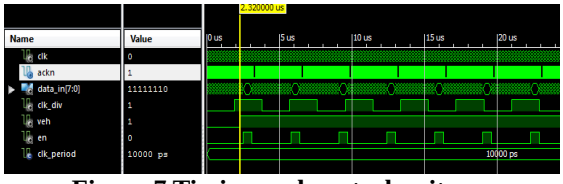

Figure 7 Timing and control unit

2) Serial data receiver*:* Simulations in figure indicates the correct functionality of the designed serial receiver. It represents how the serially received data is copied into internal memory which is copied to output of the system.

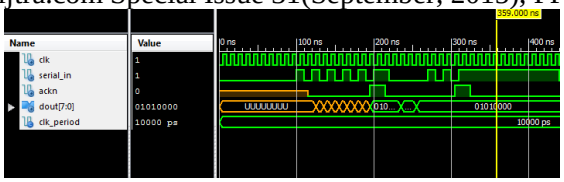

Figure 8 Serial data receiver

3) Vehicle profile extraction unit: Simulations the output of the system as the actual side profile bits extracted from serially received data from an emulator.

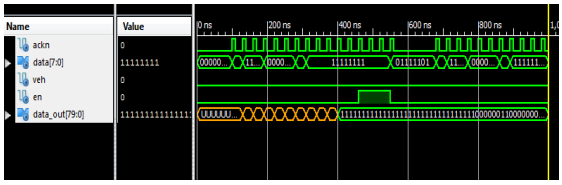

Figure 9 Vehicle profile extraction unit

4) Height calculation unit*:* Simulations indicates the functionality of the height calculation unit to be correct.

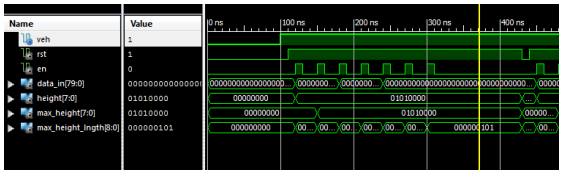

Figure 10 Height calculation unit

5) Length calculation unit: Simulations the output of the system as the actual side profile bits extracted from serially received data from an emulator.

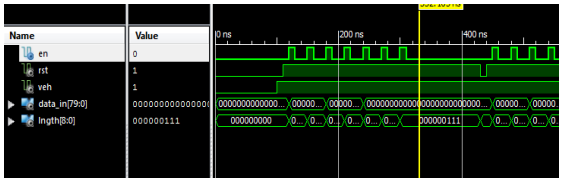

Figure 11 Length calculation unit

6) Axle calculation unit: The simulation waveforms indicate the functionality of the axle calculation unit to be correct.

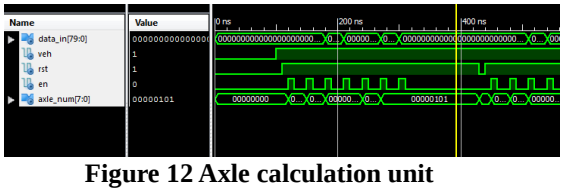

Figure 12 Axle calculation unit

7) Vehicle counting and classification unit: The simulation waveforms indicate the functionality of vehicle counting and classification unit to be correct.

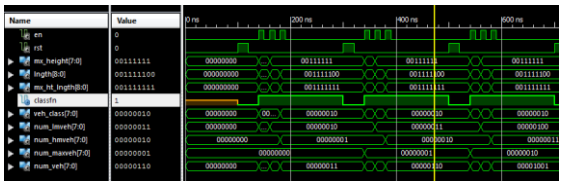

Figure 13 Vehicle counting and classification unit

8) Serial data transmitter: Simulations in the figure indicate the correct functionality of the designed serial transmitter. It represents how the data stored in RAM is serially transmitted by the hardware unit designed.

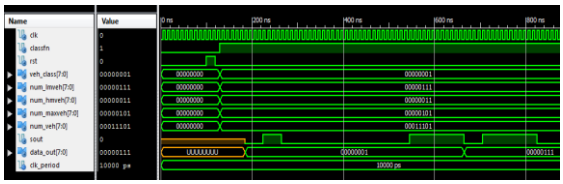

Figure 14 Serial data transmitter

9) Assigning package pins to system: Once synthesis and simulation of the system are completed, pins of the device package are assigned to input-output signals of the system designed. Process is carried out using Xilinx Plan-ahead tool, which provides a GUI to assign pin. Figure 15 shows the GUI based tool from Xilinx- Plan-Ahead, where pins can be assigned to the synthesized system. The tool generates user constrain file (.ucf), which contains the details about the pin assignment in text format.

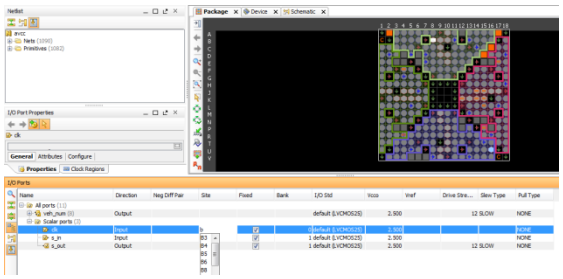

Figure 15 Assigning package pins to system

10) Configuring target device and testing: This is final step in implementation of FPGA system. In order for working of device the bit-stream (.bit) file of the system has to be downloaded into the FPGA device.Initally bit file stream is generated in Xilinx Development Tool. The bit-stream file is then downloaded into Nexys-2 board from Digilent with the help of GUI based tool Adept. The configuration file can be downloaded either directly into FPGA device or to Flash PROM chip available on the board.

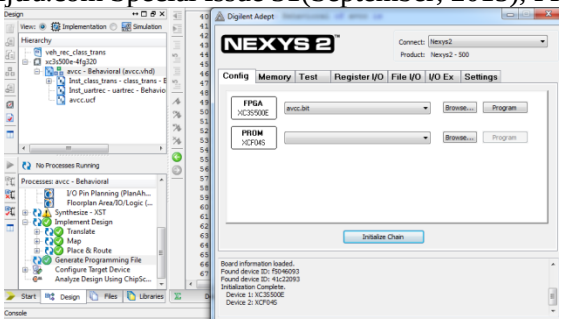

Figure 16 Configuring target device and testing

11) Graphical user interface for testing: Docklight tool is used for implementation purpose. It is a GUI based tool which can transmit or receive the data sequences serially with standard baud-rates from 110 to 256000 bits/sec. The user can also set the custom baud rate for specific applications. The user can also specify serial communication parameters like parity, data bits in one byte, start-stop bits and modes of operation like send-receive or only receive.

VIII. CONCLUSION

The low cost infrared system is developed for vehicle counting and classification purpose. Database of various vehicles are collected and stored in IR emulate module. Vehicle profile extracted is compared with database on the basis of height, length and axle count for classification purpose. .The result is stored in the computer database of the user .An infrared based vehicle counting and classification working for car, bus truck and multi- axle vehicle.

ACKNOWLEDGMENT

The successful completion of any master thesis requires a lot of guidance and assistance .I take this opportunity to express my deep sense of gratitude and sincere thanks to my project guide Prof Shweta Ashtekar and Co guide Amruta Chintawar for spending their valuable time and guidance given to me for successful completion of my project work.

REFERENCES

[1] Evaluation of infrared based automatic vehicle counters cum classifiers (avcc) under Indian traffic conditions. 2011.

[2] Gunilla Borgefors.Distance transformations in digital images. 2000.

[3] K. Subramanian C. H. Messom, S. Demidenko and G. Sen Gupta. Size position identification in real-time image processing using run length encoding. 2002.

[4] Gourab Sen Gupta Christopher H. Messom and Serge N. Demidenko. Hough transform run length encoding for real-time image processing ieee transactions on

instrumentation and measurement. 2007.

[5] K. Tano J. Harada H. Saito, Y. Arakawa and S. Shioda. Experiments on binary sensor networks for estimation of target perimeter and size. 2009.

[6] R. Kumar S. Suri J. Singh, U. Madhow and R. Cagley. Tracking multiple targets using binary proximity sensors. 2007.

[7] Y. Vijay kumar K. Babulu. Fpga realization of rle scheme for high-speed data transmission with improved compression rate. 2013.

[8] J.-Y. Choi W. Kim, K. Mechitov and S. Ham. On target tracking with binary proximity sensors. October 2005.

[9] SusmitaA.Meshram, A.V.Malviya. Traffic Surveillance by counting and classification of Vehicles from Video using Image Processing. International Journal of Advance Research in Computer Science and Management Studies. Volume 1, Issue 6, November 2013