Neural Network Based Sign Language Recognition Using Verilog HDL

R.Smitha, Ushus S Kumar, S.Suresh

Abstract— **In today's world, the field programmable gate array (FPGA) technology has advanced enough to model complex chips replacing custom application-specific integrated circuits (ASICs) and processors for signal processing and control applications. This project proposes a hardware/software co-simulation methodology using hardware description language (HDL) simulations on FPGA as an effort to accelerate the simulation time and performance [1, 2]. To accelerate the computation of the gesture recognition technique, an HW/SW implementation using field programmable gate array (FPGA) technology is presented in this project. The testing part of the neural network algorithm is being hardwired to improve the speed and performance. The American Sign Language gesture recognition is chosen to verify the performance of the approach. The major benefit of this design is that it takes only few milliseconds to recognize the hand gesture which makes it computationally more efficient.**

Index Terms— **American Sign Language, HDL, FPGA, ASICs, Neural network, VLSI**

I. INTRODUCTION

Sign language is a form of manual communication which has developed as an alternative to speech amongst the deaf and vocally impaired. Although many deaf people can speak clearly (particularly those whose hearing impairment was acquired after early childhood) and can use skills such a slip-reading when communicating with hearing people, such methods of communication are generally inappropriate for communication within the Deaf community. Therefore the hands have become the primary means of communication within these communities. The hands are also widely utilized during communication between the vocal community, with gestures often used to augment speech. However such gestures bear very little similarity to the signs that make up sign language.

First these gestures serve only an auxiliary role, rather than being the primary focus of communication as they are in signing. Second such gestures have no defined meaning, but instead are interpreted in the context of the accompanying speech. In contrast the hand gestures used in sign language are highly formalized, with each gesture having a defined meaning, in much the same manner as the spoken or written word. This allows the construction of sign-language dictionaries in which each sign of the language is equated to one or more words in a spoken language.

Hence a sign language consists of a vocabulary of signs in exactly the same way as a spoken language consists of a vocabulary of words. The task of recognizing sign-language has been the focus of many researchers, while to others has been interested in applications such as robotic control. Many of the gesture-recognition techniques developed are not specific to the application being considered and could readily be adapted to use in a range of applications. Dependent as it is on recent developments in hand-measuring technology, the field of hand-gesture recognition is a relatively young one.

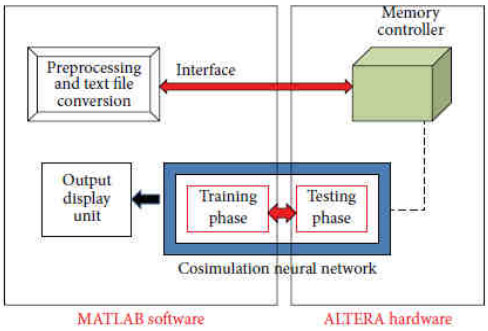

Fig. 1: System architecture of gesture recognition

As a result, the majority of the systems described in the literature are only partially completed. The literature is also spread through a wide variety of fields, such as technology for the disabled, pattern recognition, artificial intelligence, human-computer interaction and virtual reality.

II. RELATED WORK

The communication between human and machines or between people can be done using gestures called sign language [3]. The use of sign language plays an important role in the means of communication method for the hearing impaired community [4]. American Sign Language (ASL) is the 3rd most-used language and the choice for most deaf people in the United States. 500,000 and 2,000,000 people use sign language as their major daily communication tool. It seems that 3.68%of the total population is found to be hard of hearing and 0.3% of the total population is functionally deaf, out of a total population of about 268,000,000 (2005) in the US [5].

Gesture recognition is generally based on two different approaches. Primarily, glove-based analysis [6–8] where either mechanical or optical sensors are attached to a glove that transforms finger flexions into electrical signals to determine the hand posture [7]. Currently, the vision-based analysis [9–11] is used mostly, which deals with the way human beings perceive information about their surroundings. The database for these vision-based systems is created by selecting the gestures with predefined meaning, and multiple

Manuscript received May 06, 2019

R.Smitha, PG Student,M.E VLSI Design, Sree Sastha Institute of Engineering and Technology

Ushus S Kumar, AP/ECE , Sree Sastha Institute of Engineering and Technology

S.Suresh, AP/ECE , Sree Sastha Institute of Engineering and Technology

samples of each gesture are considered to increase the accuracy of the system [10]. In this paper, we have used the vision-based approach for our gesture recognition application. Several approaches have been proposed previously to recognize the gestures using soft computing approaches such as artificial neural networks

(ANNs) [12–16], fuzzy logic sets [17], and genetic algorithms [18]. Some statistical models [6] used for gesture recognition include Hidden Markov Model (HMM) [19, 20] and Finite-State Machine (FSM) [21]. ANNs are the adaptive self-organizing [22, 23] technologies that solved a broad range of problems such as identification and control, game playing and decision making, pattern recognition medical diagnosis, financial applications, and data mining [23, 24] in an easy and convenient manner [25,26]. Murakami and Taguchi in [12] presented the Japanese Sign Language recognition using two different neural network systems. Back Propagation algorithm was used for learning postures, taken using data gloves, of Japanese alphabet. The system is simple and could successfully recognize a word. The proposed automatic sampling and filtering data proved to help improve the system performance. However, the learning time of both network systems was extremely high varying from hours to days.

Maung [13] used the real-time 2D hand tracking to recognize hand gestures for Myanmar Alphabet Language. The system was easy to use and no special hardware was required. The input images were acquired via digitized photographs and the feature vector obtained using histograms of local orientation. This feature vector served as the input to the supervised neural networks system built. Implementing the system in MATLAB tool box made the work easy because of the simplicity in design and easy use of toolbox.

Bailador et al. [15] presented Continuous Time Recurrent Neural Networks (CTRNNs) real-time hand gesture recognition system. The work was based on the idea of creating specialized signal predictors for each gesture class [15]. The standard Genetic Algorithm (GA) was used to represent the neuron parameters, and each genetic string represents the parameter of a CTRNN. The system is fast, simple, modular, and a novel approach. High recognition rate of 94% is achieved from testing the dataset. This system is limited by the person's movements and activities which caused a higher noise that has significant effect on the results. The dependency of segmentation operation on the predictor proved to greatly affect the segmentation results.

Stergiopoulou and Papamarkos [16] presented static hand gesture-recognition- based Self-Growing and Self-Organized Neural Gas (SGONG) Network. Digital camera was used for input image. For hand region detection, YCbCr color space was applied and then threshold technique used to detect skin color. SGONG proved to be a fast algorithm that uses competitive Hebbian learning algorithm (the learning starts with two neurons and grows) in which a grid of neurons would detect the exact shape of the hand. However, the recognition rate was as low as 90.45%.

In all the previous works mentioned, they are purely software- based approaches.The most common software used are MATLAB and Microsoft Visual C# both of which are very powerful tools. It is preferred based on the previous research developed by Maung [13], Maraqa and Abu-Zaiter, [14] and on the review in [25, 26]. It integrates programming, computation, and visualization in user-friendly environment where problems and solutions are presented in common mathematical notation [27]. FPGAs minimize the reliability concerns with true concurrent execution and dedicated hardware since they do not use the operating system [28]. Thus, the results are not dependent on the skin color or the background of the image gestures obtained.

III. PROPOSED METHOD

This system includes following three major steps:

A. Database Generation
B. Text File Conversion
C. Training and Testing

A. Database Generation

The images for database are capture during a Cannon camera which produces image frames of RGB pixels. The ASL alphabet gestures are used to obtain the hand images. The database contains two different subjects with two different backgrounds. The Cannon camera is actually not stationary since it is not in a fixed position but the background is maintained as either black or white. The images are stored in .jpg format The size of the image frame is 640 × 480 pixels. To maintain the resolution and decrease redundancy, the frames are resized to 64 ×64 pixels. These files are stored in the memory of the ALTERA-ModelSim using command "$readmemh."The ALTERA-ModelSim is being called from the MATLAB using HDLDAEMON. HDLDAEMONcontrols the server that supports interactions with HDL simulators

B. Text File Conversion

The grayscale images are converted into text files which contain the hexadecimal value of the pixels. Each image is configured and stored with the appropriate configuration ID. The text file contains 4096 values to be read and stored onto the memory of the ALTERA-ModelSim. The text files obtained from the image files are stored onto another file in the memory locations of the ALTERA-ModelSim. The files contain data represented by hexadecimal values and hence contain 16 digits of length. For every negative edge of the clock cycle, the data is read into the memory location. The value test design/k shows the database set images ranging from 1 to N (the last image depending on the number of database sets considered).

```
0001 0001 0001 0001 0001 0001 0002 0001 0000 0001
0002 0002 0002 0001 0001 0001 0001 0002 0002 0001
0002 0002 0001 0002 0002 0002 0002 0002 0002 0002
0002 0002 0001 0002 0002 0001 0002 0002 0001 0001
0001 0002 0002 0001 0002 0002 0002 0002 0002 0003
0002 0002 0003 0004 0003 0002 0002 0002 0003 0003
0002 0003 0003 0002 0003 0003 0005 0004 0002 0001
00CF 00D3 00DF 00E3 00E2 00E0 00D7 00CB 00DC 00DE
00B3 00C4 00C5 00C4 00C8 00CE 00D6 00DA 00DB 00DE
00D9 00D7 00D1 00C7 00C0 00C2 00C1 00BA 00BC 0089
0004 0003 0004 0003 0067 00C8 00CE 00B0 00A8 00B0
00C6 00D1 00CB 00C2 00B5 009D 00A8 00B3 00AD 0095
00A4 008A 00A5 00A5 00A6 00B8 00CF 00D8 00DD 00DA
00AE 00AD 00A6 009C 0098 008E 0088 0081 007A 0069
0002 0002 0002 0002 0002 0001 0001 0002 0002 0001
0003 0003 0003 0003 0003 0004 0004 0004 0005 0004
0005 0006 0006 0005 0006 0006 0007 0007 0004 0005
0008 0008 0009 0007 0006 0005 0004 0004 0005 0004
0004 0002 0003 0004 0002 0002 0002 0003 0002 0002
0003 0003 0004 0003 0003 0003 0003 0002 0003 0005
```

Fig. 2: Text values in memory location

C. Training and Testing

We have used four different sets of data for training and testing of the co- simulation neural network designed. The memory module of the design and the testing of the network are being shifted onto the hardware level to speed up the performance. The neural network contains 16 input neurons; each database image is processed and stored as a feature vector of 16 values. The image is being compressed from 4096 pixel values into 16 feature vector values. The Once the input pre-processed image is obtained, the edge image is calculated. The edge image is split into four quadrants and the maximum location of each quadrant is calculated. The distance value to be stored as the feature value is the distance between the centre of the image and each quadrant maximum image. The compression ratio is 256 times that of the input values. Hence, only 0.39% of the image is being used as the feature vector to train and test the neural network designed. As the database set gestures involved in the application may vary very rapidly, it is highly essential to keep the feature vector as low as possible with no trade-off with respect to accuracy and performance. In this particular application, the feature vector is maintained as 16-bit vector which makes the system memory efficient and also highly redundant.

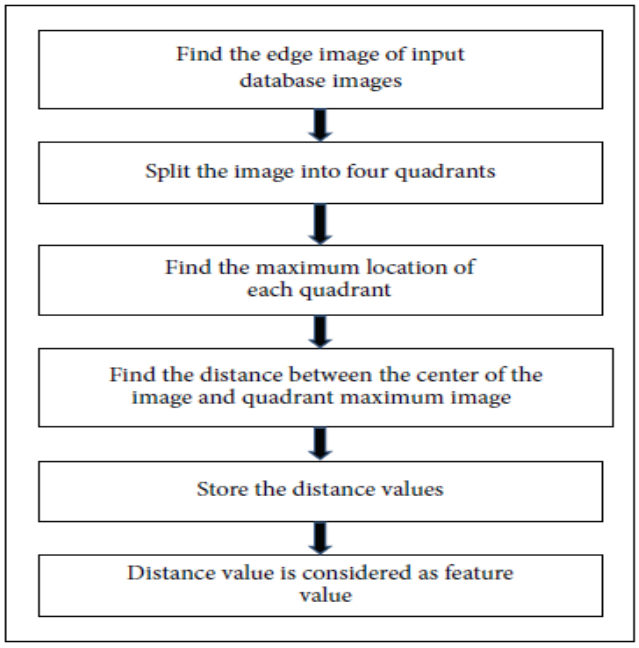

Fig. 3: Feature Extraction Algorithm for distance values

1. Network Architecture

Perhaps the most fundamental property of a neural network is its architecture or topology. This defines the number of nodes contained in the network and also the manner in which they are interconnected. The power of neural systems arises from their connectivity. The leftmost layer in this diagram is the input layer, and these neurons have no input connections from other neurons. The activation of the input-layer neurons is set by placing the input data values directly into them. These activations then feed forward into the second layer (the hidden layer) and are used to calculate the activation of the neurons in that layer. In turn the activation of the hidden nodes is fed forward and used to calculate the activation of the nodes in the output layer. The activations of the output layer are taken as the output of the network.

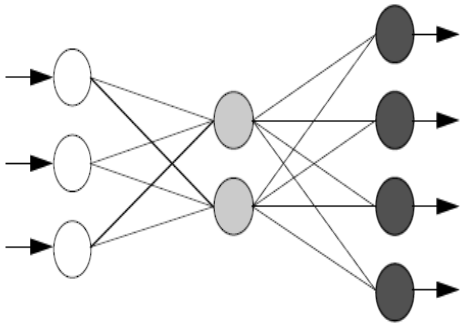

Fig. 4: A small feed-forward network with a single hidden layer

Determining the correct number of layers and neurons in a network to enable it to learn and perform a particular task remains more of an art than a science. Various rules of thumb have been proposed, but all of these are very application dependent The determination of a suitable topology depends greatly on the experience of the person controlling the network's training. The approach taken in this research was to run a small number of trials with varying network structures to obtain a 'feel' for the nature of the problem. On the basis of these exploratory trials suitable parameters were selected for use in the main body of experiments.

Examples of this approach include is the Cascade-Correlation algorithm, and the self growing counter-propagation network. However none of these methods have as yet proved advantageous enough to encourage widespread use.

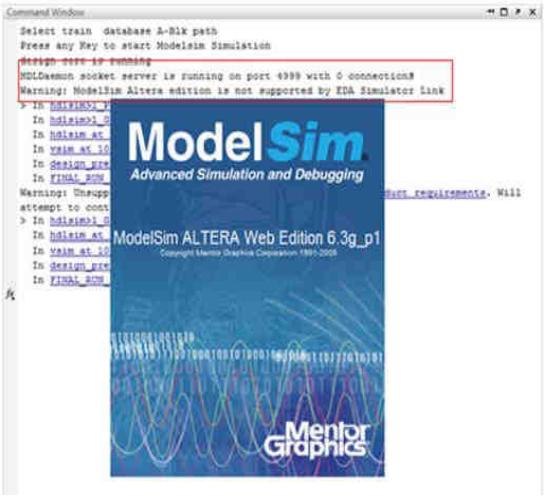

Fig. 5: HDLDAEMON as the interface between MATLAB and ALTERA-ModelSim

2. Co-Simulation Neural Network

Neural networks are based on the parallel architecture of neurons present in human brain. It can be defined as a multiprocessor system with very high degree of interconnections and adaptive interaction between the elements. The choice of neural networks to recognize the gestures automatically is due to the following aspects like adaptive learning (using a set of predefined database sets), self- organization from the training module, real-time operation with parallel computations, and high fault tolerance capability [30, 31]. Since static hand postures not only can express some concepts, but also can act as special transition states in temporal gestures recognition, thus estimating that static hand postures play an important role in gesture recognition applications.

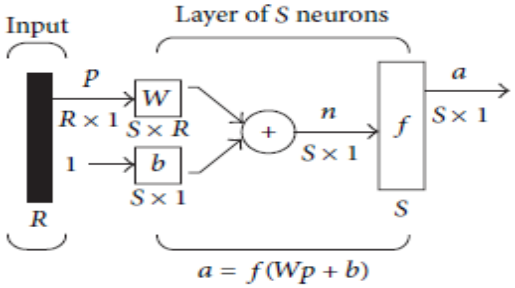

Fig. 6: Neural network model

A gesture recognition system takes an image as an input, processes it using a transform that converts the image into a feature vector, which will then be compared with the feature vectors of a training set of gestures. A new technique called co-simulation neural network is being adopted. In this method, a part of the neural network is designed on the hardware with dedicated ports. An interface is being introduced among different levels of the neural network to communicate with one another on two different platforms. Our network is built of 16 input neurons in the input layer, 50 neurons in the first hidden layer, 50 neurons in second hidden layer, and 35 neurons in the output layer. Since the features extracted from the image to be used for recognition was 16 the input layer has 16 neurons. The output is displayed as a visual representation of the gesture image and hence is a 7 × 5 (35neurons) grid display of rows and columns.

3. Device Utilization Factor

To understand the resource constraints, Xilinx ISE simulations are performed. Device Utilization indicates the FPGA elements, such as flip-flops, LUTs, block RAM, and DSP48s. Estimated indicates the number of FPGA elements that the algorithm might use based on the current directive configurations. Total indicates the total number of FPGA elements available on the FPGA target. Percent indicates the percentage of the FPGA elements that the algorithm might use.

Fig. 7: Device utilization Summary Report for memory storage

From the summary reports developed, it is observed that only 33% of the IOBs are being used on the hardware platform. The report is being developed on a Xilinx FPGA SPARTAN 3E with target device XC3S250e-5tq144. Each image is being read and stored onto the memory of the ALTERA-ModelSim.

IV. EXPERIMENTAL RESULTS

The signs for all the alphabets from A to Z are being recognized using the combinational neural networks architecture. The advantage of using the algorithm is high processing speed which can produce results in real- time manner. The speed of processing is increased due to the neural network architecture and design of HW/SW co-simulation. It is also advantageous as even noise corrupted almost up to 48%, the signs can still be retrieved. For future extensions processing of words and sentence gestures can be included.

Fig. 8 a) American sign language Database

Fig. 8 b) American sign language Database

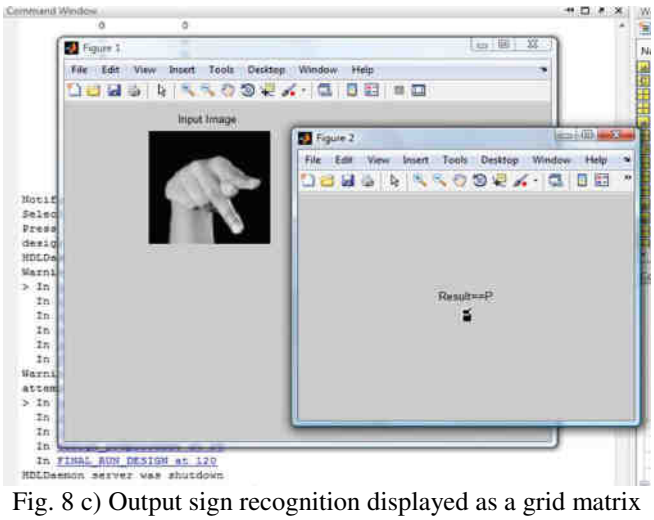

Fig. 8 c) Output sign recognition displayed as a grid matrix

TABLE I Comparison of SW versus HW/SW simulation platforms

TABLE I: RECOGNITION ACCURACY

	Software simulation (MATLAB)[5]	HW/SW co-simulation(MATLAB and ModelSim)
Performance	0.999722	0.999716
Epochs	1111/2000	1088/2000
MSE	0.998182/0.1	0.0999716/0.1
Gradient	$0.00269182/1e-006$	$0.00204379/1e-006$
Recognition Time	0.0058 secs	$5.7656e-004$ secs

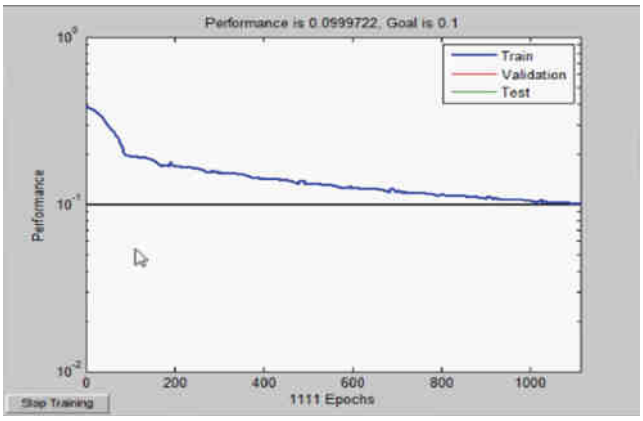

Fig. 9: Performance curve for gesture recognition using SW(MATLAB) simulation analysis

CONCLUSION

The signs for all the alphabets from A to Z are being recognized using the combinational neural networks architecture. The advantage of using the algorithm is high processing speed which can produce results in real-time manner The speed of processing is increased due to the neural network architecture and design of HW/SW co-simulation. It is also advantageous as even noise corrupted almost up to 48%, the signs can still be retrieved. For future extensions processing of words and sentence gestures can be included case the grammar and syntax play an important role in deciding the efficiency and speed of the structure

REFERENCES

[1] Pratibha Pandey, Vinay Jain, "Hand Gesture Recognition for Sign Language Recognition: A Review", International Journal of Science, Engineering and Technology Research (IJSETR), Volume 4, Issue 3, March 2015

[2] Mathavan Suresh Anand, Nagarajan Mohan Kumar, Angappan Kumaresan, "An Efficient Framework for Indian Sign Language Recognition Using Wavelet Transform" Circuits and Systems, Volume 7, pp 1874 -1883, 2016.

[3] Mandeep Kaur Ahuja, Amardeep Singh, "Hand Gesture Recognition Using PCA", International Journal of Computer Science Engineering and Technology (IJCSET), Volume 5,Issue 7, pp. 267-27, July 2015

[4] Sagar P.More, Prof. Abdul Sattar, "Hand gesture recognition system for dumb people"-2017

[5] Chandandeep Kaur, Nivit Gill,"An Automated System for Indian Sign Language Recognition", International Journal of Advanced Research in Computer Science and Software Engineering.2015

[6] Sunitha K. A, Anitha Saraswathi.P, Aarthi.M, Jayapriya. K, Lingam Sunny, "Deaf Mute Communication Interpreter-A Review", International Journal of Applied Engineering Research, Volume 11, pp 290- 296, 2016.

[7] Neelam K. Gilorkar, Manisha M. Ingle, "Real Time Detection And Recognition Of Indian And American Sign Language Using Sift", International Journal of Electronics and Communication Engineering & Technology (IJECET), Volume 5, Issue 5, pp. 11-18 , May 2014

[8] P. Vijayalakshmi and M. Aarthi, "Sign language to speech conversion," in 2016 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, India, April 2016, pp.1-6

[9] Das, L. Yadav, M. Singhal, R. Sachan, H. Goyal, K. Taparia, R Gulati, A. Singh, and G. Trivedi, "Smart glove for sign language communications," in 2016 International Conference on Accessibility to Digital World (ICADW), Guwahati, India, Dec 2016, pp. 27–31.

[10] M. Boulares and M. Jemni, "Toward a mobile service for hard of hearing people to make information accessible anywhere," in 2013 International Conference on Electrical Engineering and Software Application, Hammamet, Tunisia, March 2013, pp. 1-5

[11] H. V. Verma, E. Aggarwal, and S. Chandra, "Gesture recognition using kinect for sign language translation," in 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), Shimla, India, Dec 2013, pp. 96–100

[12] Dong, M. C. Leu, and Z. Yin, "American sign language alphabet recognition using microsoft kinect," in 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Boston MA, USA, June 2015, pp. 44–52

[13] N. VAPNIK, "AN OVERVIEW OF STATISTICAL LEARNING THEORY," IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 10, NO. 5, PP. 988–999, SEP 1999

[14] T. N. T. Huong, T. V. Huu, T. L. Xuan, and S. V. Van, "Static hand gesture recognition for vietnamese sign language (vsl) using principle components analysis," in 2015 International Conference on Communications, Management and Telecommunications (ComManTel), DaNang, Vietnam, Dec 2015, pp. 138-141

[15] P. C. Badhe and V. Kulkarni, "Indian sign language translator using gesture recognition algorithm," 2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS), Bhubaneswar, 2015, pp. 195-200.

[16] Shreyashi Narayan Sawant and M. S. Kumbhar ,"Real Time Sign Language Recognition using PCA", 2014 IEEE International Conference on Advanced Communication Control and Computing Technologies (lCACCCT),2014.