

Design of Filter for Fingerprint Application on FPGA

S Gayathri¹ Tanveerahamad Rozewale²

¹Associate Professor

^{1,2}Department of Electronics and Communication Engineering

^{1,2}SJCE, Mysuru, India

Abstract— This paper describes the design of filter for fingerprint application using parallel architecture which makes fast processing, reduces the processing time, increases speed and will make it useful for biometric authentication. This paper also describes about the comparison of MATLAB TO FPGA Implementation and presents a set of efficient criteria for implementing fingerprint image on FPGA Virtex5 board with optimization in device utilization.

Key words: Fingerprint; Preprocessing; Parallel architecture; Filtering, Device utilization

I. INTRODUCTION

Every human beings fingerprint is unique. Each fingerprint has a unique characteristic and pattern that is made up of lines and spaces. The lines are called ridges while the gap between the ridges are called valleys. The significances about using fingerprint biometrics is that more widely accepted, reliable and convenient than other forms of physical identification in case of privacy is considered.

The Fingerprint image recognition process involves pre-processing, feature extraction and post-processing [5] the preprocessing involves The normalization, orientation, filtering, binarization and thinning image. Normalization: The gray levels of the input image is compared with that of standard values, normalize the pixel values of the image if the gray levels of the image are not in the standard range The input image, which is to be processed, is first stored inside a ROM as a text file. This image pixels mean is calculated using Accumulator and divider blocks. Then the mean used to compute The variance of the image. Then mean and variance outputs of the image applied to the Comparator and Adder/Subtractor block to get the Normalized image. Orientation: The orientation field of a fingerprint image defines the local orientation of the ridges contained in the fingerprint. Distinct ridges provide accurate extraction of the minutiae from the fingerprint images. Binarization: Compares each pixel to particular threshold which then changes its value to either pure white (0x00) or pure black (0xFF). The threshold used is local block mean. Thinning: Thinning process is used to skeletonize the binary image by decreasing all lines to a individual pixel thickness. Filtering: This is a mechanical process of manually finding and removing ridge structures that probably do not belong and by filling in gaps in ridges that do not belong. The reason for using Gabor filter is because of its characteristics spatial localization, orientation selectivity and spatial frequency selectivity.

This paper shows the filtering of generic image or also data base image where filtering is the one of the major step involved in fingerprint recognition process in pre-processing steps. The basic idea of fingerprint recognition process can be explained by following figure

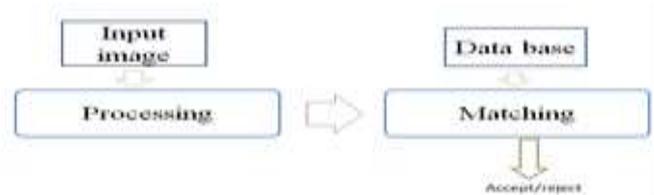


Fig. 1.1: depicts the basic idea involved in Fingerprint recognition process.

when the given input image is processed it will match from data base image after undergoing all the steps involved pre-processing steps of fingerprint recognition process as mentioned, if it matches one can observe the score of match with considering accept/reject image.

In order to compare two fingerprints, a set of invariant and discriminating features have to be extracted from them. It is well known for fingerprint research that minutiae are discriminating and reliable factor. It is custom for fingerprint verification systems required to provide a high degree of security are so far based on minutiae matching. Unfortunately, noise, contrast deficiency, improper image acquisition, geometrical transformation, deformation and skin elasticity often make reliable minutiae detection very difficult. Spurious minutiae can be produced and valid minutiae can be hidden due to the low image quality of the fingerprint. This paper gives a perfect solution of filtering the image with fast processing that is use of parallel architecture FPGA(Field Programmable Gate array) implementation.

Two different approaches serial and parallel architectures can be implement to filter out the noise present in the oriented fingerprint image. The filtered output in both the cases resembles with each other, except for the change in processing time and hardware utilization. Parallel method of processing is preferred to filter the image using Gabor function

II. RELATED WORK

The objective of design and simulation of gabor filter using verilog code[1] in fingerprint recognition is to segment the texture of fingerprint. This includes the design of a single filter to segment multiple (> 2) textures and the design of multiple filters to segment multiple textures. The computational efficiency becomes increasingly important as the size of images, number of Gabor prefilters, number of Gaussian postfilters, and number of textures increases. This[1] work has completed only with simulation, in contrast to its future scope the current paper gives well solution for it.

The use of gabor filters in the paper[2] is 16 which presents matching scheme that utilizes a ridge feature map to match fingerprint images. The technique described here obviates the whose spatial frequencies correspond to the average inter-ridge spacing in fingerprints, is used to capture the ridge strength at equally spaced orientations. The

genuine accept rate of the Gabor filter based matcher is observed to be ~ 10% to 15% higher than that of minutiae-based matcher at low false accept rates. Fingerprint feature extraction and matching takes nearly 7.1 seconds on a Pentium IV, 2.4 GHz processor. However we are presenting paper which works on the latest Virtex5 board with speed in terms of 1us/1ps for the simulation of filtering.

The implementation of orientation block [3] has been done in virtex2 pro with better performance but the future scope remained there with to implement the Filtering block on FPGA. This current paper implements the filtering block on Virtex5 FPGA with low speed grade of -2.

In this work[4], the Log-Gabor filter is singularity in the log function at the origin, the spatial domain is not accurately constructed so the filters are constructed in the frequency domain. This work shows the construction with help of the Fourier transform (FT) for the whole image and implement the filtering with a bank of Log-Gabor filters in frequency domain

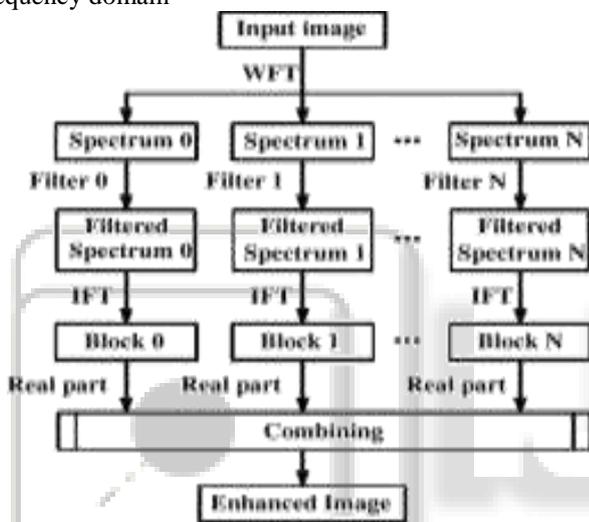


Fig. 2.1: shows the one of the algorithm for fingerprint enhancement

III. METHODOLOGY

A. Parallel Architecture:

In parallel architecture the filter block of pre-processing stage is implemented using symmetric extension method, to filter out noise present in the fingerprint image. Symmetric extension is useful in reducing the boundary distortion. Symmetric extension is accomplished by reflecting the rows and columns at the image boundary. The image is divided into 8 X 8 matrix and symmetric extension is applied, resulting in 10 X 10 matrix. This 10 X 10 matrix is applied to a processing element which is nothing but a filter.

The processing element makes use of sobel filter to remove the noise in the oriented image. The filtered output is stored in the RAM for further processing. Since parallel architecture as shown in figure is employed, the computation required for filtering is 1/4 th of that of serial architecture. The filtered output is same in both the cases but only the resource utilization differs. Parallel architecture requires more hardware in comparison with serial architecture. Sobel operator is used for edge detection Vertical Detection and Horizontal Detection. The vertical mask of vertical detection as shown in Table 3.1. When

mask is applied on the image, it works like a first order derivate and calculates the difference of pixel intensities in pixel edge region. As the center column consists of zero it does not include the original values of an image but instead it calculates the difference of right and left pixel values around that edge. Also the center values of both first and third column is 2 and -2 respectively. This gives more weightage to the pixel values around the edge region. This increases the edge intensity and it becomes enhanced compared to original image.

i11	i11	i12
1	i11	i12
i21	i21	i22

Fig. 3.1: shows 3x3 image

x1	x2	x3	y1	y2	y3
x4	x5	x6	y4	y5	y6
x7	x8	x9	y7	y8	y9

Fig. 3.2: shows sobel values X and Y

The horizontal mask of vertical detection is as shown in figure3.2 It calculates the difference among the pixel intensities of a particular edge. As the center row of mask consists of zeros it does not include original values of edge in the image but rather it calculates the difference between above and below pixel intensities of particular edge. Thus increase the sudden change of intensities making the edge more visible.

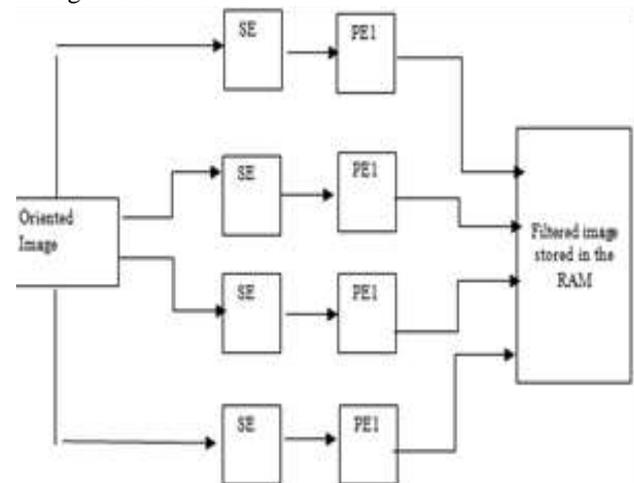


Fig. 3.3: shows the block diagram of parallel architecture

In symmetric extension method 10x10 image there is no loss of original data. The image coefficients are multiplied with sobel x values giving the Gabor coefficients.

B. Design:

A Gabor filter is linear filter whose impulse response is defined by a harmonic function multiplied by Gaussian

function. The Fourier transform of a Gabor filter's impulse response is the convolution of Fourier transform of harmonic function and the Fourier function of Gaussian function.

$$g(x, y) = s(x, y) wr(x, y) \quad (1)$$

where $s(x, y)$ is a complex sinusoidal, known as the carrier, and $wr(x, y)$ is a 2-D Gaussian-shaped function, known as the envelop. The general form of filter [1] using Gabor function can be represent as

$$G(x, y, \theta, f_0) = \exp\left\{-\frac{1}{2}\left(\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2}\right)\right\} \cdot \cos(2\pi f_0 x_{\theta})$$

$$\begin{bmatrix} x_{\theta} \\ y_{\theta} \end{bmatrix} = \begin{bmatrix} \sin\theta & \cos\theta \\ -\cos\theta & \sin\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

where θ is the ridge orientation with respect to vertical axis, f_0 is the selected ridge frequency in x_{θ} - direction, σ_x and σ_y are the standard deviation of Gaussian function along the x_{θ} and y_{θ} axes respectively and the $[x_{\theta}, y_{\theta}]$ are the coordination of $[x, y]$ after a clockwise rotation of the Cartesian axes by an angle of $(90-\theta)$ [8].

By tuning a filter to specific frequency and direction, the local frequency and orientation information can be obtained

IV. IMPLEMENTATION

Implementaion of Filter: Consider the flow chart of parallel architecture to implement Gabor filter on FPGA

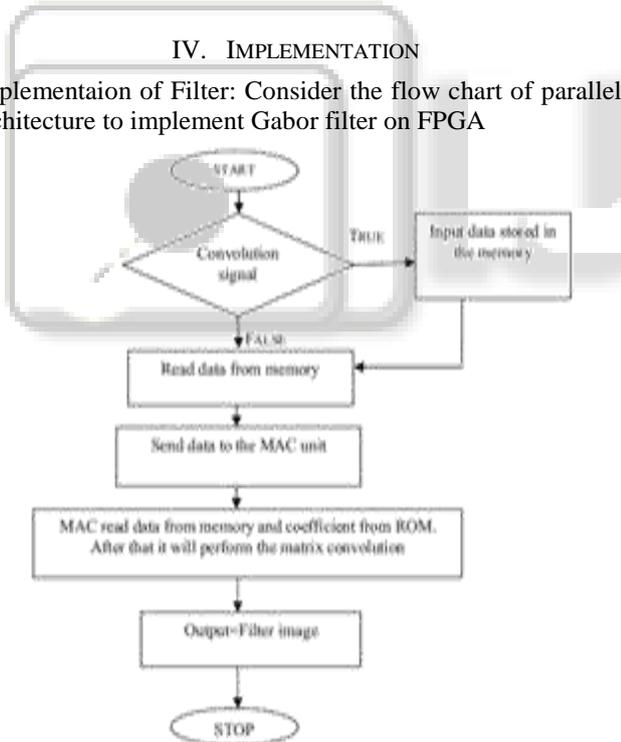


Fig. 4.1: Depicts the flow chart of parallel architecture

- The Fingerprint image is presented in matrix form or in pixel.
- The orientated image pixels are convolved with coeffecient values in MAC unit(convolution between image pixels with kernel values).
- The control unit takes the data from memory location and sent it to the multiplication-accumulator (MAC) unit.
- In MAC, ROM stores the coefficient or kernel value of the filter.

- When both input data and kernel values are available multiplication and accumulation would take place in the MAC.
- The result of MAC is the filtered image.

The proposed work is to implement Gabor filter based on memory base architecture for real-time convolution with variable kernels. Firstly store the input data (image pixel) in the memory. The size of the memory depends on the pixel size. If the image is of size 16x16 then the memory size will be 16x16. It means that every memory location will store a value for 1 image pixel. Then start the convolution process. The control unit takes the data from memory location and sent it to the multiplication-accumulator(MAC) unit. In MAC, ROM stores the coefficient or kernel value of the Gabor filter. When both input data and kernel values are available multiplication and accumulation would take place in the MAC. The result of MAC is the filtered image. The flowchart of Gabor filter is shown in the Figure 3.4 Input Memory block stores the incoming signal in the form of image pixel in the memory. Convolution block performs the process between the input image and the Kernel values defined by the Gabor filter. The kernel value defines the ridge and valley region of fingerprint. MAC block is used to perform the convolution operation. The convoluted signal with the Gabor coefficient is transformed into a matrix format. Simulation of the filter is carried out using verilog HDL language using Xilinx software and synthesized code is generated. The simulation will generate the schematic according to the code.

The flow of the convolution process between image pixels and coefficient kernels [5] is as follows. Method of convolution and the sequence of memory reading are as shown in Figure 4.2 and Figure 4.3. Firstly the pixel D11 is convoluted with the kernel. The value of D11 after convolution is given by

$$D11 = (D00 \times W11) + (D01 \times W12) + (D02 \times W13) + (D10 \times W21) + (D11 \times W22) + (D12 \times W23) + (D20 \times W31) + (D21 \times W32) + (D22 \times W33) \quad (3)$$

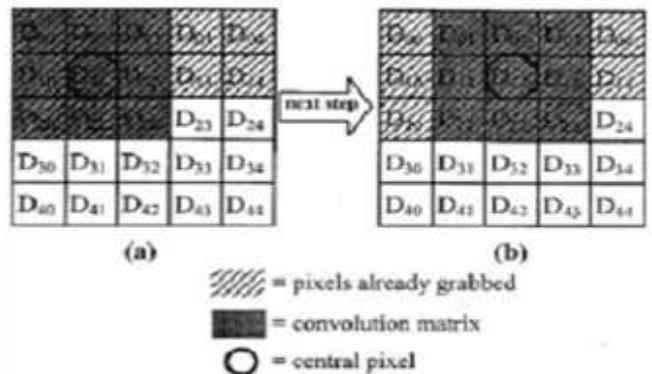


Fig. 4.2: Method of Convolution process

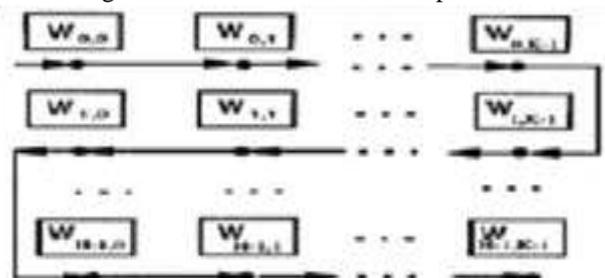


Fig. 4.3: Memory readings

Comparison of MATLAB and FPGA Implementation.
 MATLAB Implementation: The MATLAB is a high level language which is here a software reference model for hardware implementation. In this MATLAB code a resized 128x128 image for which we have 0 to 255 gray scale level



Fig. 4.4: Resized image 128x128

which is divided into 16x16 cells that is 256 cell which further divided into 8x8 image for parallel processing of image. This is a 8bit processing such that will have four cells of each 8x8 image processing, here again for filtering 3x3 matrix of the pixel values which multiplies with coefficients matrix (3x3) (sobel values) gives us filtered value such that n terms after matrix multiplication will have n+1 terms in matrix multiplication as shown in the figure 4.2 and 4.3.

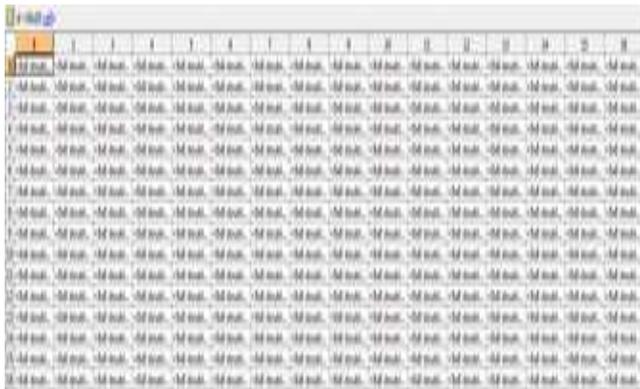


Fig. 4.5: shows 128X128 image broken down to 8x8 cells totally 16 X 16 = 256 cells

	1	2	3	4	5	6	7	8
1	0	1	2	3	4	5	6	7
2	128	129	130	131	132	133	134	135
3	256	257	258	259	260	261	262	263
4	384	385	386	387	388	389	390	391
5	512	513	514	515	516	517	518	519
6	640	641	642	643	644	645	646	647
7	768	769	770	771	772	773	774	775
8	896	897	898	899	900	901	902	903

Fig. 4.6: shows Individual Cell

	1	2	3	4	5	6	7	8
1	15480	15481	15482	15483	15484	15485	15486	15487
2	15608	15609	15610	15611	15612	15613	15614	15615
3	15736	15737	15738	15739	15740	15741	15742	15743
4	15864	15865	15866	15867	15868	15869	15870	15871
5	15992	15993	15994	15995	15996	15997	15998	15999
6	16120	16121	16122	16123	16124	16125	16126	16127
7	16248	16249	16250	16251	16252	16253	16254	16255
8	16376	16377	16378	16379	16380	16381	16382	16383

Fig. 4.7: shows {16,16}

Here the matlab code is written in such fashion to get the output of gabor filter image, noise removal image. Considering generic image/database image it has to

convert it into text file then have to call as penguins data in verilog code implementation

1) *FPGA Implementation:*

Filter block will be design and implement Filter block is implemented using Gabor function on FPGA, Verilog code simulation using Xilinx software and synthesized code is generated. The simulation will generate the schematic according to the code. Then generating programming file and configuring target device with analyzing the device the Program running is done.

V. EXPERIMENTAL RESULTS

MATLAB results: The MATLAB code is performed in MATLAB2013 version.



Fig. 5.1: a Input image

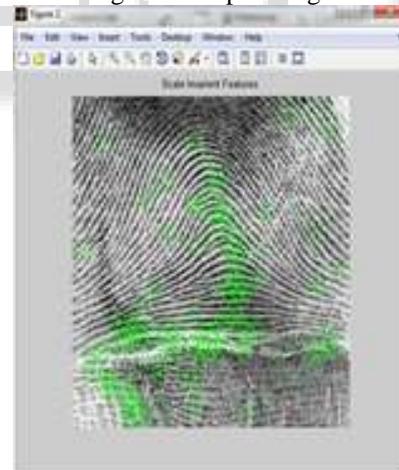


Fig. 5.1: b Scale invariant features

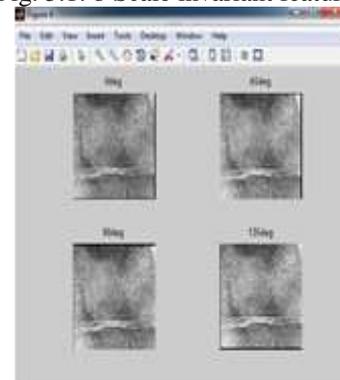


Fig. 5.1: c Orientation filtering



Fig. 5.1: d Gabor Filter output image

Fig. 5.1: a,b,c,d shows MATLAB output figures files

FPGA results: This Implementation is performed with Xilinx ISE tool Here four output indicates the parallel processing of 8x8 image simultaneously

```

Minimum period: 4.201ns (Maximum Frequency: 238.015MHz)
Minimum input arrival time before clock: 1.685ns
Maximum output required time after clock: 2.826ns
Maximum combinational path delay: No path found

Timing Detail:
-----
11 values displayed in nanoseconds (ns)

-----
Timing constraint: Default period analysis for Clock 'clock'
Clock period: 4.201ns (frequency: 238.015MHz)
Total number of paths / destination ports: 31548 / 1902
    
```

Block	Used	Available	Utilization
Block 1	1	1	100%
Block 2	1	1	100%
Block 3	1	1	100%
Block 4	1	1	100%

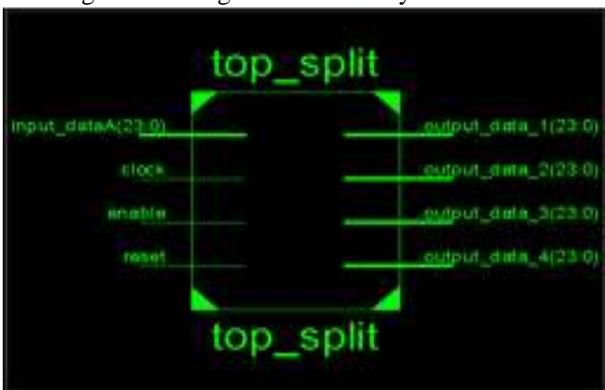


Fig. 5.2: RTL Schematic

The top_split block is constituted after adding sources to the top_split block program which can be seen as follows in figure 5.3.



Fig. 5.4: shows the device utilization and output simulation of Gabor filter

The following figure shows RTL schematic of final chip block which is obtained after making master ucf file

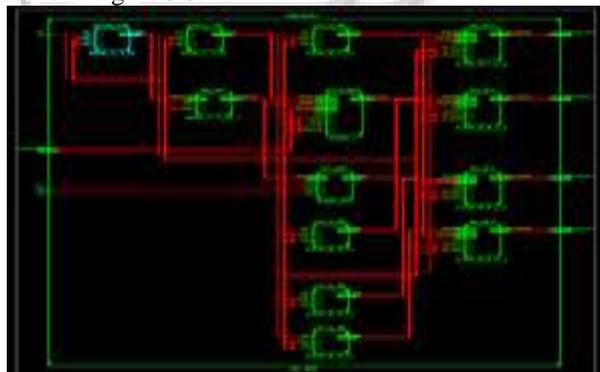


Fig. 5.3: RTL schematic of internal of the top_split block

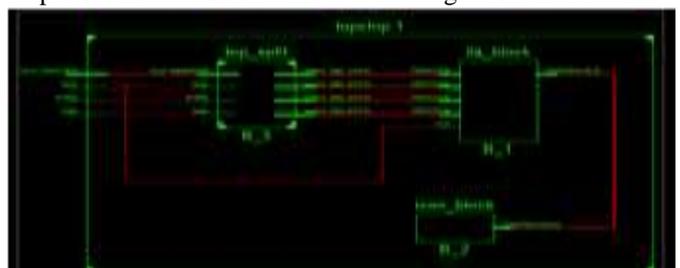


Fig. 5.5: Shows the RTL schematic of final chip block FPGA virtex LX110T ML505 model

Device Utilization summary:

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	392	69120	0%
Number of Slice LUTs	784	69120	1%
Number of fully used LUT FF pairs	392	784	50%
Number of bonded IOBs	123	640	19%
Number of Block RAM/FIFO	19	148	12%
Number of BUFG/BUFGCTRLs	1	32	3%



Fig. 5.6: Implementation after configuring the target device on FPGA Virtex5 XUPLX505T

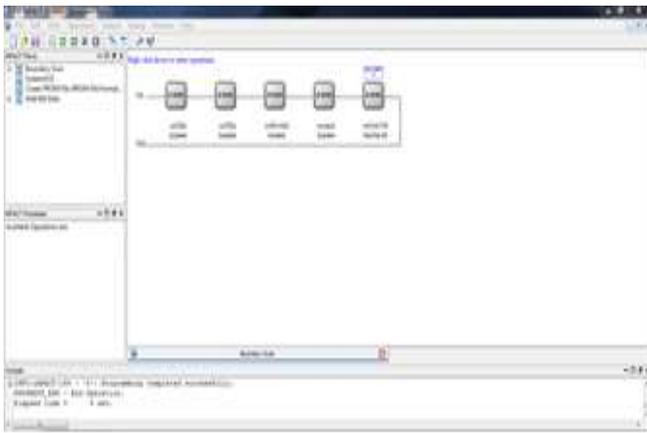


Fig. 5.7: shows successful run of program in device5

VI. CONCLUSION AND FUTURE SCOPE

The input image either from generic /database image is filtered using gabor function in MATLAB which is served as a reference software for hardware implementation on FPGA using Hardware Description Language Verilog code. The image is converted into text file (penguins data) which is filtered using verilog code and finally implemented on FPGA. The filtering is done with 8 bit which can be further modified, filtering is implemented using Gabor function because of orientation selectivity, so one can perform this by using other techniques [6] and even ASIC implementation [7] in future.

REFERENCES

- [1] S. Gayathri Research scholar SJCE, Mysore, Karnataka, India, Dr V. Sridhar Principal PECSE, Mandya, Karnataka, India "Design and simulation of Gabor filter using verilog HDL".
- [2] Muhammad Umer Munir and Dr. Muhammad Younas Javed College of Electrical and Mechanical Engineering, National University of Sciences and Technology Rawalpindi, Pakistan "Fingerprint Matching using Gabor Filters".
- [3] S. Gayathri and V Sridhar Dept of E & C, SJCE, Mysore, Karnataka, India Dept of E&C, PESCE, Mandya, Karnataka "FPGA Implementation of Orientation Field Estimation of Fingerprint Recognition Process".
- [4] Wei Wang *, Jianwei Li, Feifei Huang, Hailiang Feng Pattern Recognition Lab, Optoelectronic Engineering College, Chongqing University, Chongqing 400030 Communicated by T. Tan "Design and implementation of Log-Gabor filter in fingerprint image enhancement".
- [5] Feng Zhao*, Xiaoou Tang Department of Information Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong Received 4 June 2005; received in revised form 17 June 2006; accepted 6 September 2006."Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction".
- [6] Ching-Tang Hsieh, Eugene Lai, You-Chuang Wang* Department of Electrical Engineering, Tamkang University, Tamsui, Taipei Hsien, Taiwan 25137, Republic of China Received 21

December 2001" An effective algorithm for fingerprint image enhancement based on wavelet transform".

- [7] S. Gayathri and V Sridhar Dept of E & C, SJCE, Mysore, Karnataka, India Dept of E&C, PESCE, Mandya, Karnataka "ASIC Implementation of image enhancement technique for fingerprint recognition process".
- [8] S. Gayathri, V. Sridhar Dept of E&C, Principal SJCE, Mysore, Karnataka, India PESCE, Mandya, Karnataka, India "Image enhancement technique for fingerprint recognition process".