Design and Simulation of Cascaded Gabor Filter Using Verilog HDL

Vijay Laxmi[1], Dr. P. K. Chaturvedi[2]
[1, 2] Department of Electronics & Communication
[1, 2] SRM University, Modinagar, Ghaziabad, India

Abstract- *This paper presents the Design and Simulation of Cascaded Gabor filter for fingerprint recognition using Verilog HDL. The application of Gabor Filter technique to enhance the fingerprint image and it is used to define the ridges and valley regions of fingerprints is by convoluting the image pixel with Gabor filter coefficient. The experimental result was the signal convoluted with the Gabor coefficient. The effect of image enhancement has an impact on features extraction and the fingerprint identification of recognition rate, according to the research of traditional fingerprint image enhancement technology, Gabor filtering for fingerprint image enhancement technology is proposed.*

Keywords- Gabor filter, image enhancement, MAC, verilog HDL, Xilinx.

I. INTRODUCTION

A Gabor filter is a linear filter whose impulse response is defined by a harmonic function multiplied by a Gaussian function. Because of the convolution property, the Fourier transform of a Gabor filter's impulse response is the convolution of the Fourier transform of the harmonic function and the Fourier transform of the Gaussian function. Gabor filters are directly related to Gabor wavelets, since they can be designed for number of dilations and rotations. A Gabor filter is a linear filter used for edge detection in image processing which is named after Dennis Gabor. Gabor filter frequency and orientation representations are similar to those of human visual system, for texture representation and discrimination it has been found to be remarkably appropriate. Designing Gabor filter will help enhancing the quality of fingerprint image. In the digital signal processing the digital filter is the most important part to filter out the signal noise thus eliminating the unwanted image in fingerprint image. This project used Gabor type filter to segment the finger print texture. Texture segmentation is the most important in fingerprint recognition process. This filter enhances the image quality as its make ridges clearly differentiated from each other. The designing of Gabor filter for fingerprint recognition using verilog HDL, this filter can be run using full version software.

II. DESIGN OF GABOR FILTER

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.

$$g(x, y) = s(x, y)\, w_r\,(x, y)$$

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 function of Gabor filter can be represent as

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

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

where θ is the ridge orientation with respect to vertical axis, $f0$ is the selected ridge frequency in xθ – direction,σx and σy are the standard deviation of Gaussian function along the xθ and yθ axes respectively and the [xθ, yθ] are the coordination of [x,y] after a clockwise rotation of the Cartesian axes by an angle of (90-θ). Referring to the function in (1), the function G(xθ,yθ, $f0$) can be decomposed into two orthogonal parts, one parallel and the other perpendicular to the orientation θ.

$$G(x, y, f_0)\big|_{\theta=90^\circ} = G_{BP}(x, f_0)\, G_{LP}(y)$$

$$G_{BP}(x, f_0) = \exp\left\{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}\right)\right\}\cos(2\pi f_0 x)$$

$$G_{LP}(y) = \exp\left\{-\frac{1}{2}\left(\frac{y^2}{\sigma_y^2}\right)\right\}$$

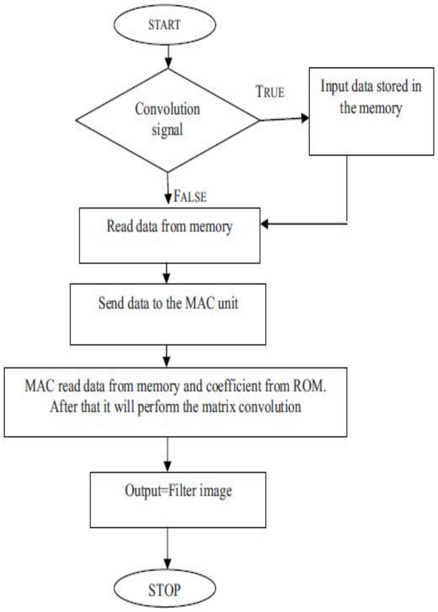
Where GBP is only a band-pass Gaussian function of x and $f0$ parameters while GLP is only a low-pass Gaussian filter of y parameter.

2.1 Methodology

There are three major factors needed to be considered, maximization of speed, minimization of area and power consumption. In this work, minimization of area consumption will be the main priority.

2.1.1 Design

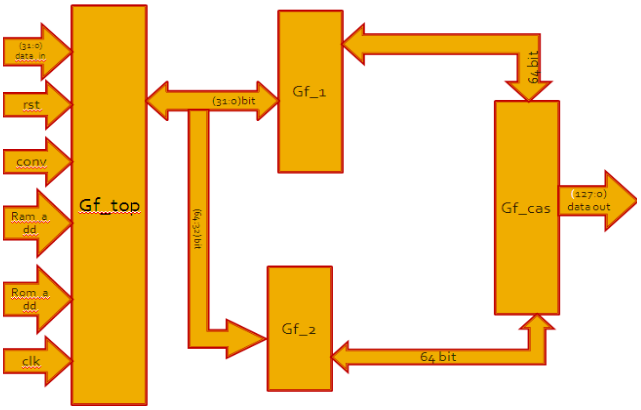
The design of the new multiplication-accumulation unit must be done precisely. This is due to the sensitivity of the transition in a single data path. Below is the design flow of the filter. Firstly, when the convolution signal is '0' the input data which is in pixel format will enter the filter and stored in the memory. The size of the memory depends on the pixel size. If the pixel is 16x16 then the memory size will be 16x16 too. It means that every memory location will be stored for value for 1 image pixe. When the convolution signal is triggered to '1' the convolution process starts. The controller will read the image that is stored in the memory and send the data to the arithmetic unit. The controller will call the data from the determined memory location. In arithmetic unit there is also a ROM which will permanently store the coefficient kernel value. The value of kernel will also be called by the control into the convolution circuit. When both data has entered the convolution circuit the process of multiplication and accumulation will take place. Only one series of data will be convoluted at a time. The counter will count for 9 convolution operation before giving out the result of filtered image. The reason why count for 9 consecutive cycle stands for the 9 coefficient kernel value. This will also be the result of the filter.


START
Convolution signal
TRUE
Input data stored in the memory
FALSE
Read data from memory
Send data to the MAC unit
MAC read data from memory and coefficient from ROM. After that it will perform the matrix convolution
Output=Filter image
STOP

Result and Discussion

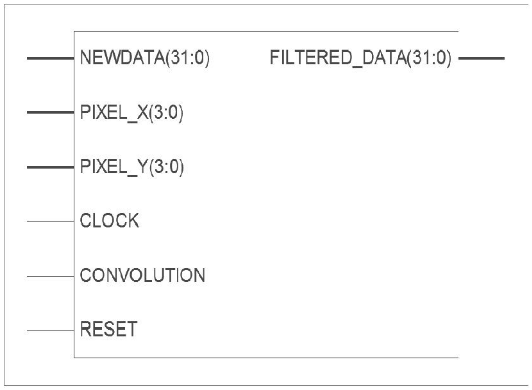
After redesigning the Gabor filter in verilog using Xilinx 10.1 software, the code was then synthesized. The summary of the design was shown in figure.

Cascaded Design



A. Top level

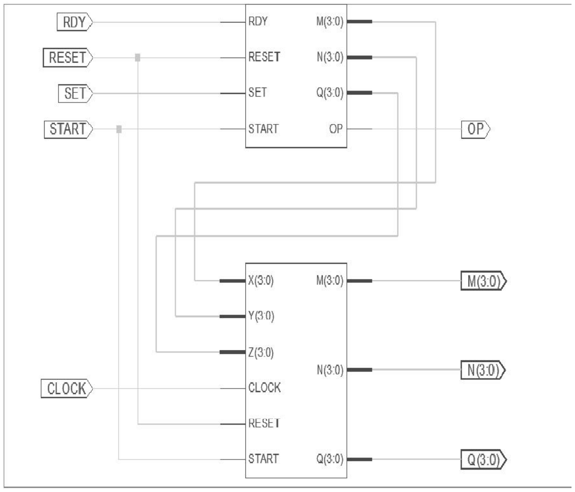
Figure shows the schematic view of the top level filter. There were 6 input pins and one output pin on the top level. Newdata stands for an unfiltered 32-bits image data. Pixel-X and Y hold the position of the memory when the write memory occurred. Clock and reset pins indicates the generated clock with 40ns period and reset button for the filter. The 'convolution' signal is to indicate the operation of the filter. If the signal is high then the convolution process takes place. If it is low then the filter receives image input and stores it to the memory based on the input location.



B. Controller (CLU)

The control logic unit functions as controller for the data flow in the filter. It gives instruction to the other blocks to do their job. Basically, it gives the memory address to read data to the MEMORY and give address of coefficient to the ALU. This CLU will only generate the address location when the 'START' signal is high. This signal indicates the convolution process that has taken place but if the signal is low, it indicates that the writing of image data into the memory takes place. This CLU contains only 2 different blocks. One is the counter for the coefficient and memory address, and the other one is the counter decoder. The design of the counter gives the relationship between the coefficient and memory address. When the coefficient address was counted up until 9, the memory address for Y- direction will count a plus one. And the X-direction address must wait until Y-direction counts until 16 then it counts a plus one.

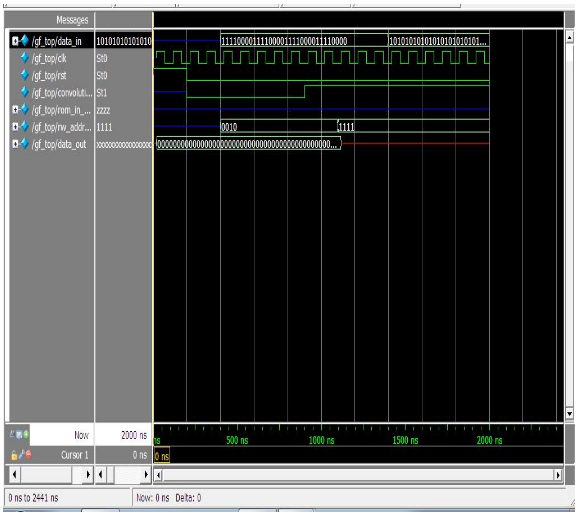
This CLU will also read feedback from the arithmetic unit which is the 'SET' and 'RDY' signals. These feedbacks from the arithmetic unit are used to control the operation 'OP' of the arithmetic unit. When the 'OP' signal was high, the convolution process at the arithmetic unit starts. When the 'OP' signal is low, the convolution process stops. The 'OP' was designed this way to control accurate series data sent to the arithmetic unit so there won't be any mismatch of data. The memory decoder decodes the data and sends the correct memory address and coefficient address separately to the memory and the arithmetic unit. Figure shows the schematic view of the controller.



C. Memory

The memory block is used to store the image pixel. The decoder only decodes address for Y-direction only. The clock was removed from the decoder so the decoded Y-direction can arrived at the same clock cycle with the X-direction. The adress for X-direction is supplied directly from the CLU or from the filter input. The image input is also connected directly from the filter input. The writenable signal indicates whether the operation is a write data or read data.

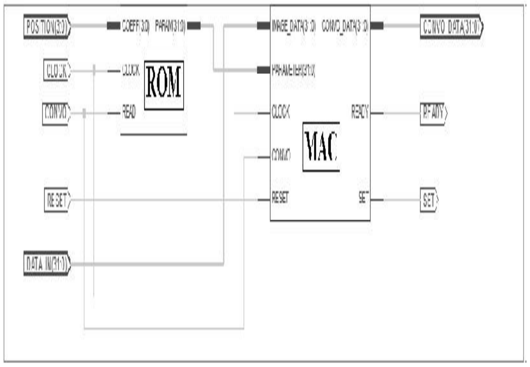
First the 'WRITENABLE' signal is high to indicate the writing process is taking place. Then the signal goes low to read the data in the memory. The memory will give the output on the same clock cycle as the address location enters.

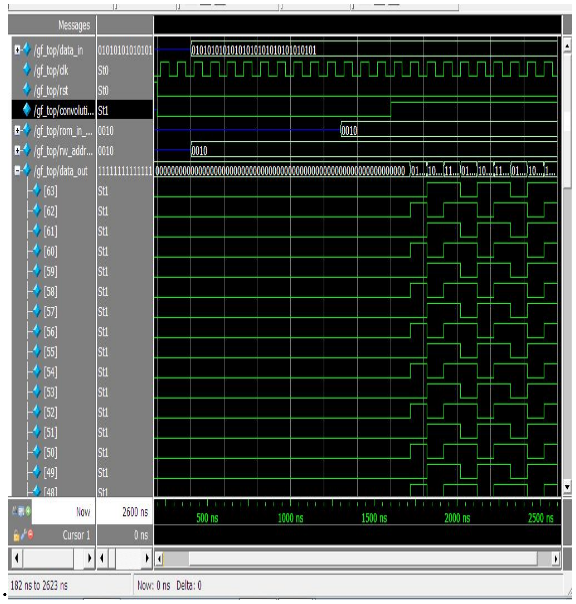


Verification of memory unit

D. Arithmetic (ALU*)*

This is the main part of this work, arithmetic unit. This is where the convolution process takes place. It consists of two parts: the ROM and the MAC. The ROM is used to store the 9 coefficient values that are needed to convolute with the image while MAC consists of a buffer, a multiplier, an adder and a counter. The crucial part of this design was to make sure that the convolution process happened align with the correct image data and coefficient. The 'CONVO' signal plays important role to ensure there was no mismatch of data read. From the figure 11 below, the 'CLOCK' and the 'CONVO' both were connected to the ROM and MAC. When the 'CONVO' went from low to high, the convolution process starts. The feedback 'READY' and 'SET' were sent to the CLU indicates convolution process completed. The CLU then will push the 'CONVO' signal from high to low before the next convolution takes place. These processes take 9 complete convolutions before sending the convoluted data out. Figure shows the detailed structure inside the MAC. The buffer was used to hold the 'CONVO' operation for 1 cycle before the multiplier. The intention was to wait for the correct data sent from the memory for the convolution process. The Multiplier and the adder are connected in series. The design was done in such a way to lessen the area consumption of the filter. After 9 consecutive multiplications and additions, the counter in the MAC will gives the expected result. Since the design is a single data path also known as pipeline, the multiplication and the addition will take a longer period of time. The total cycles required for convolution for this design is 222 clock cycles with a time period of 40ns per cycle.





Verification of ALU

III. CONCLUSION

The design enhancement proposed for Gabor Filter has successfully reached. The area of the design has been significantly reduced while the function of the filter is perfectly maintained.

By adjusting the memory and the controller unit, the functionality of a complete and correct digital Gabor Filter is obtained.

ACKNOWLEDGEMENT

I sincerely acknowledge in all earnestness, the patronage provided by our Director, **Dr. Manoj Kumar Pandey**, Engineering and Technology, to endeavour this project.

I wish to express my deep sense of gratitude and sincere thanks to our Professor and Head of the Electronics and Communication Department, **Mr. Pankaj Singh** for his encouragement, timely help and advice offered to me.

I express my thanks to our Project Coordinator **Dr. P. K. Chaturvedi**, Professor, and **Mr. Manoj Vishnoi** Assistant Professor, Electronics and Communication Department, for his encouragement and advice.

I am very grateful to my guide **Dr. P. K. Chaturvedi** Professor, Electronics and Communication Department, who has guided me with inspiring dedication, untiring efforts and tremendous enthusiasm in making this project successful and presentable.

I extend my gratitude and heart full thanks to all the staff, nonteaching staff of Electronics and Communication Department and to my parents and friends, who extended their kind co-operation by means of valuable suggestions and timely help during the course of this project work.

REFERENCES

[1] Razak, A.H.A. Taharim, R.H. “Implementing Gabor Filter for Fingerprint Recognition using verilog HDL,” IEEE explorer, March 2009.

[2] P. H. W. L. Ocean Y. H. Cheung, Eric K.C. Tsang, Bertam E.SHi, "Implementing of Gabor-type Filters on Field Programmable Gate Arrays," 2005.

[3] A. P. Arrigo Benedetti, Nello Scarabottolo, "Image Convolution o FPGAs: the implementation of a multi-FPGA structure," 1998.

[4] K. S. Vasily G. Moshnyaga, Keikichi Tamaru, "A Memory based architecture for real-time convolution with variable kernels," 1998.

[5] Clifford E. Cummings, “Verilog-2001 Behavioral and Synthesis Enhancement,” Dec 2001.