THOMSON REUTERS ENDNOTE

[Prajapati, 5(1): January, 2016] **ISSN: 2277-9655**
(I2OR), Publication Impact Factor: 3.785

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

RTL IMPLEMENTATION OF GAUSSIAN FILTER USING VHDL

Harshal Prajapati
Electronics and Communication Engineering Department, Government Polytechnic, Gandhinagar, India

ABSTRACT
A Gaussian smoothing also called blurring is frequently used in Image processing application. A Gaussian blurring is same as convolving Gaussian kernel with the image. Applying a Gaussian blur is reducing a high frequency component in the image so it is Gaussian low pass filter. To perform smoothing operation is to reduce a noise in Image. In first phase it has been designed with VHDL and verifies using MATLAB simulation and power is estimated by X power Estimator..

KEYWORDS: Gaussian Filter, Convolution, Smoothing, ASIC..

INTRODUCTION
A Gaussian smoothing is commonly used with edge detection of image. A Gaussian Blur filter before edge detection aims to reduce the level of noise in the image which improves the result of the edge detection [1, 2]. A Gaussian smoothing is also used as a pre processing stage in computer vision algorithm to enhance image at different values [3]. A Gaussian filters have the properties of having no overshoot to input step function while minimizing the rise and fall time. A Gaussian blur has the effect of reducing the image's high-frequency components so a Gaussian blur is a low pass filter. A Gaussian blur effect is typically generated by convolving an image with a kernel of Gaussian values. In practice, it is best to take advantage of the Gaussian blur's separable property by dividing the process into two passes. In the first pass, a one-dimensional kernel is used to blur the image in only the horizontal or vertical direction. In the second pass, another one-dimensional kernel is used to blur in the remaining direction.
The main objective has been to lay the groundwork for converting software driven image application to hardware implementation by using Application Specific Integrated Circuit (ASIC). The goal is achieve for image enhancement by removing Gaussian noise. ASIC offers high performance at the price of high development cost [4]. The same has been implemented through VHDL to obtain clear and readable image.
The paper is organized as follows. Section I is introduction of the work. Section II is about the Gaussian filter and method use for design of Gaussian filter. Section III shows the result for Gaussian filter. Finally Section IV concludes piece of work.

GAUSSIAN FILTER & METHOD FOR FILTER DESIGN
A Gaussian filters whose impulse response is a Gaussian function. A Gaussian theoretically has infinite support but we need a filter of the finite size [5, 6]. A Gaussian filters smoothes an image by calculating weighted averages in a filter box. The standard deviation 6 of the Gaussian function play import role in its behaviour. The standard deviation 6 of the Gaussian determines the amount of smoothing [7]. Large values for sigma will give large blurring for larger image sizes. Gaussian filtering is used to blur images and remove noise. The Gaussian filter is a non uniform low pass filter [8]. Central pixels have a higher weighting than those on the periphery. A Large value in Gaussian kernel is producing a wider peak for greater blurring. Gaussian curve with a 2-dimensional domain as shown in figure 1.

 ISSN: 2277-9655
(I2OR), Publication Impact Factor: 3.785

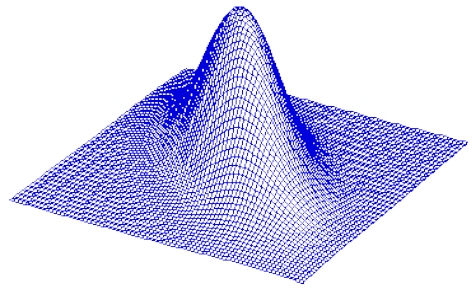

***Figure 1** Gaussian curve with a 2 dimensional domain*

$$G(x, y, 6) = \frac{1}{2\pi 6^2} e^{-(x+y)/26^2} \qquad (1)$$

Where 6 is standard deviation of the distribution

The scale space of an image is defined as a function L(x, y, 6) that is produced from the convolution of a Gaussian kernel G(x, y, 6) with an input image I(x, y):

$$L(x, y, 6) = G(x, y, 6) * I(x, y) \qquad (2)$$

*Where * is convolution operation between Gaussian kernel and image*.

Gaussian filtering is done by convolving each point in the input array with a Gaussian kernel [5]. The Gaussian filter first convolves each row with 1D filter. Then convolve each column with 1D filter.

RESULTS

A. Simulation waveform of Gaussian Filter

The Simulation Waveform of Gaussian filter is as shown in figure 2.

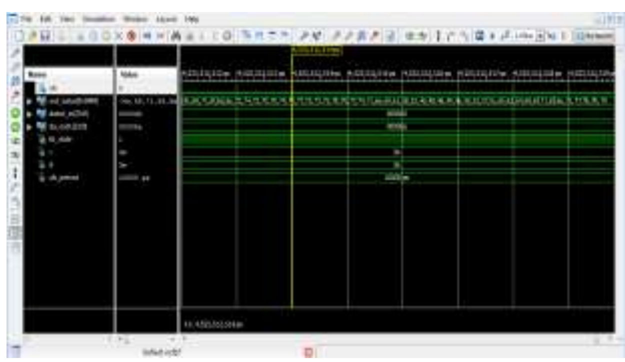

Figure 2 Simulation Waveform of Gaussian filter

B. RTL Schematic of Gaussian Filter

RTL Schematic of Gaussian filter is as shown in figure 3. Internal structure of Gaussian filter is as shown in figure 4.

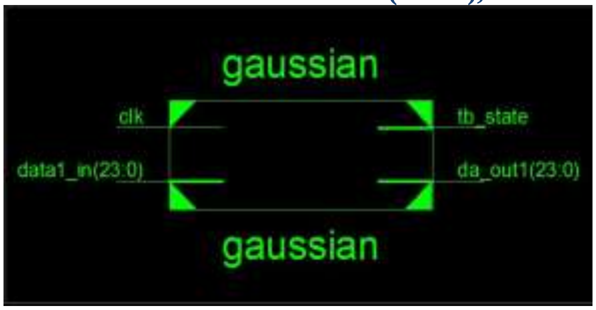

Figure 3 RTL Schematic of Gaussian Filter

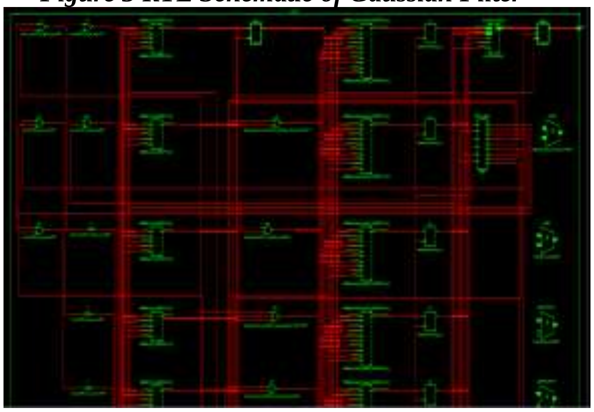

Figure 4 Internal Structure of Gaussian filter

C. Gaussian Filter Results

A Gaussian filter replace the value of pixel by convolve with rows and columns of image with Gaussian kernel. The Gaussian noise is removed by Gaussian filter as shown in figure 5(a) and (b). This result is produce by the use of Xilinx and verify by Matlab Software.

(a) (b)

Figure 5 (a) Degraded Image (b) Gaussian filtered Image

D. Devise Utilization Report

This synthesis report is generated after synthesis of design for the targeted Xilinx VERTEX 4 based xc4vlx60-ff668 FPGA Device. This report contains about how many component used as shown in Table 1.

Table 1. Device Utilization Summery

Device Utilization Summary		
Logic Utilization	Used / Available	Utilization
Number of Slices Register	6064 / 26624	22%

Number of Slices FFs	1425 / 11420	12%
Number of 4 input LUTs	9985 / 53248	18%
Number of Bonded IOBS	2 / 448	0%

E. Timing and power Summery

This synthesis report is generated after synthesis of design for the targeted Xilinx VERTEX 4 based xc4vlx60-ff668 FPGA Device. Power summary report is generated by Xilinx X-Power Estimator (XPE) [10].

Timing Summary:
Speed Grade: -12
Minimum period: 15.374ns
Maximum Frequency: 65.044MHz

Power summary:
Total quiescent power consumption (mw): 486 mw
Total operating power consumption (mw): 11 mw
Total estimated power consumption: (mw): 497 mw

F. Comparative Analysis between FPGA Devices

Spartan and Vertex FPGA family are used to measure the performance of the Gaussian filter. Table 2 shows performance Comparison between various Spartan FPGA Devices. Table 3 shows the performance Comparison between various Vertex FPGA Devices.

Table 2. Performance Comparison between various Spartan FPGA Devices

Family	Device	No. of Slices	Total No. of LUTs	No. of Slice ffs	No's of Bonded IOBs	Max. Freq.
Spartan 3	xc3s5000-fg900	6770 / 33280 (20%)	7666 / 66560 (11%)	10843/ 66560 (16%)	50 /633 (7%)	40.267MHz
Spartan 3E	xc3s1600e-fg320	6776 / 14752 (45%)	7845 / 29504 (26%)	10811/29504 (36%)	50/ 250 (20%)	45.612MHz
Spartan 6	Xc6x100-fgg484	10740 /126576 (8%)	6792/ 63288 (10%)	1498 / 12238 (12%)	50 /326 (15%)	38.758MHz

Table 3. Performance Comparison between various Vertex FPGA Devices

Family	Device	No. of Slices Register	Total No. of LUTs	No. of Slice ffs	No’s of Bonded IOBs	Max. Freq.
Vertex 4	xc4vlx60 -ff668	6064 / 26624 (22%)	6719 / 53248 (12%)	9985 / 53248 (18%)	2 / 448 (0%)	65.044 MHz
Vertex 5	xc5vlx20t-2ff323	9995 / 12480 (80%)	6473 / 12480 (51%)	1425 / 11420 (12%)	2 / 172 (1%)	120.023MHz
Vertex 6	xc6vcx75t-2ff484	9900 / 93120 (10%)	6179 / 46560 (13%)	1280 / 11180 (11%)	2 / 240 (0%)	107.897MHz
Vertex 7	xc7vx330t-ffg1157	9900 / 385600 (2%)	6179 / 192800 (3%)	1280 / 11180 (11%)	2 / 600 (0%)	121.846MHz

CONCLUSION

A Gaussian filter is the best filter to remove Gaussian noise from the image. A Gaussian filter is use for Image Enhancement. The proposed design is suitable for reduction of components. The simulated and synthesized results shows that proposed design can work at an estimated frequency of 65.044 MHz by using Vertex 4 FPGA device and estimated power consumption is 497mw. The simulated and synthesized results shows that proposed design can work at an estimated frequency of 38.725 MHz by using Spartan 6 FPGA device.

REFERENCES

[1] Shapiro, L. G. & Stockman, G. C: "Computer Vision", page 137, 150. Pretence Hall, 2001

[2] http://www.regentsprep.org/regents/math/algtrig/ats2/normallesson.htm, dated 1/1/2016

[3] Mark S. Nixon and Alberto S. Aguado. Feature Extraction and Image Processing. Academic Press, p. 88, 2008

[4] Panchami Padmasana, Mihir Narayan Mohanty, Member, IEEE, Hemanta Kumar Sahu, “VHDL Implementation of Spatial Filter for Image Enhancement”, International Conference on Communication and Signal Processing, India, 2014.

[5] https://en.wikipedia.org/wiki/Normal_distribution, dated 1/1/2016

[6] *Hazewinkel, Michiel , ed. (2001)* , "Normal distribution", Encyclopedia of Mathematics, Springer, dated 1/1/2016.

[7] R.A. Haddad and A.N. Akansu, "A Class of Fast Gaussian Binomial Filters for Speech and Image Processing," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 39, pp 723-727, March 1991

[8] http://www.stat.wisc.edu/~mchung/teaching/MIA/reading/diffusion.gaussian.kernel.pdf.pdf, dated 1/1/2016.

[9] John C. Ross. Image Processing Hand Book, CRC Press. 1994.

Hiral Pujara, Pankaj Prajapati, “RTL Implementation of Viterbi Decoder using VHDL”, *IOSR Journal of VLSI and Signal Processing*, PP 65-71, Mar. – Apr. 2013.