Hardware based algorithm for Chaotic and Edge Enhanced Error Diffusion

R.Anne Selva Rose
PG Scholar, Department of ECE
Kumaraguru College of Technology
Coimbatore – 641 049 . Tamilnadu

S.Govindaraju
Professor,Department of ECE
Kumaraguru College of Technology
Coimbatore – 641 049. Tamilnadu

ABSTRACT
Digital halftoning quantizes a grayscale image tone bit per pixel for display and printing on binary devices. It is a crucial technique used in digital printers to convert a continuous tone image into a pattern of black and white dots. Halftoning is used since printers have a limited availability of inks and cannot reproduce all the color intensities in a continuous image. Error Diffusion is an algorithm in halftoning that iteratively quantizes pixels in a neighborhood dependent fashion. Halftones are created through a process called dithering. The standard error diffusion algorithm was introduced by Floyd and Steinberg. Though it has the advantage of producing high visual quality images at low cost ,still it suffers from the problem of introducing worm-like artifacts in smooth regions. To overcome such problem, a chaotic and edge enhanced error diffusion method for image enhancement is proposed.

General Terms
Error diffusion, Floyd and Steinberg algorithm, Conventional Error Diffusion algorithm, Edge directed Error Diffusion algorithm, Chaotic and Edge enhanced Error Diffusion algorithm.

Keywords
Digital Halftoning, Electronic paper, Field Programmable Gate Array, edge enhancement.

1. INTRODUCTION
Halftone is the reprographic technique that simulates continuous tone imagery through the use of dots, varying either in size, in shape or in spacing. It is a technique to obtain more observable levels for lower level devices. It can also be used to refer specifically to the image that is produced by this process. Where continuous tone imagery contains a infinite range of colors or gray, the halftone process reduces visual reproductions to a binary image that is printed with only one color of ink. This binary reproduction relies on a basic optical illusion. Halftone applications are image printing[1]-[5], watermarking[6], image coding[7], digital display[8] and the innovative electronic paper[9][10].

Error diffusion is the most commonly used technique of digital halftoning. It is an adaptive algorithm that uses an threshold error feedback to produce patterns with different spatial frequency content, depending on the input image values. Unlike a point process, error diffusion takes neighborhood information into account to determine the output.

1.1 Related work
In digital halftoning, many algorithms have been proposed for error diffusion [3][7][8][9][16][15]. The standard error-diffusion algorithm was introduced by Floyd and Steinberg[15]. Though it has the advantage of producing high visual quality halftone images with quite low computational cost, it still suffers from the problem of introducing worm-like artifacts in some of the smooth regions. To overcome this the proposed algorithm is discussed. In section II, we present the Floyd and Steinberg algorithm . In section III, we present the Edge Directed Error diffusion algorithm . In section IV, we present the Chaotic and edge enhanced error diffusion algorithm. In section V, simulation results are discussed for random number of array inputs. In Section VI future work is discussed. the paper. Section VII concludes the paper.

2. FLOYD AND STEINBERG ALGORITHM
In this section, image dithering is reviewed which reduces the error. This algorithm achieves dithering by diffusing the quantization error of a pixel to its neighboring pixels according to the distribution.

Floyd-Steinberg dithering is commonly used by the manipulation software, if an image is converted into GIF format then it is restricted to a maximum of 256 colors.

$$\begin{bmatrix} & * & 7 \\ 3 & 5 & 1 \end{bmatrix}$$

(*) indicates the pixel currently being scanned. And this algorithm scans input from left to right and top to bottom quantizing pixels one by one. Each and every time the quantization error is transferred to the neighbouring pixels while not affecting the pixels that already have been quantized. If a number of pixels have been rounded downwards, then the next pixel is rounded upwards, such that on average the quantization error is close to zero. For optimal dithering, the counting of quantization errors should be of sufficient accuracy to prevent rounding errors from affecting the result. This method gives the high visual quality images with low cost, but produces the worm-like artifacts in smooth region and the regular patterns cannot be avoided.

Some methods used to overcome the drawback like extending the support of adjacent pixels[2][3], adjusting the diffusion coefficients or threshold[4][5], using non-raster scans or rotated dispersed dither[11][12], and adopting an edge directed diffusion scheme[13]. After implementing these methods too, regular patterns cannot be avoided. Although these modifications can reduce worm-like artifacts, few of them can be easily implemented by hardware.

This is the basic image dithering algorithm for error diffusion.

3. EDGE DIRECTED ERROR DIFFUSION ALGORITHM

There are two possible approaches for edge directed error diffusion algorithm. One possible approach is to stop the diffusion at the edge location.

g(x,y)

Algorithm 1: Edge –Stopping Error Diffusion

1. Diffuse the error by,

$u(m,n) = x(m, n) - \sum_{(k,l)\in N} h(k,l)e(m-k,-l)$ (1)

2. Perform the quantization

$$b(m, n) = Q[u(m, n)] = \begin{cases} \lambda, & u(m,n) \geq \lambda/2 \\ 0, & otherwise \end{cases} \quad (2)$$

3. Calculate the error term

$$e(m, n) = b(m, n) - u(m, n) \quad (3)$$

Edge blurring is reduced in conventional error diffusion, since the diffusion process is stopped at the edge boundary. But the computational complexity is increased. Error diffusion filtering coefficients can also be modified and so the diffusion direction matches the edge orientation.

Algorithm 2: Edge –Adaptive error Diffusion

Error is diffused based on the following rule.

If $\sum_{(k,l)\in N} c(k,l) = 0$, then it is non-edge pixel and use the standard filter h(k, l) in equation (1).

If $\sum_{(k,l)\in N} c(k,l) > 0$, then it is edge pixel.

Edge weighting :

$h_w(k, l) = h(k, l)*c(k, l)$.

Filter normalization :

$h'(k, l) = h_w(k, l) / \sum_{(k,l)\in N} h_w(k,l)$.

The first step is used modified filter h'(k, l) in u(m, n) and perform the above two steps as in algorithm 1.

And this edge directed error diffusion halftoning is difficult to completely avoid resampling operations in the publication process.

4. CHAOTIC AND EDGE ENHANCED ERROR DIFFUSION

This proposed Chaotic and edge enhanced error diffusion is the four steps fast approach for error diffusion.

The input gray level image and output binary value are denoted as g(x, y) and b(x, y) respectively, where (x, y) represents the position of the currently processed pixel. H is a diffusion filter and Q is the quantizer. First this algorithm scans input in the raster scan order and checks whether each pixel in g(x, y) is an edge point or not. If it is then the corresponding quantization threshold is generated. Then the algorithm quantizes the modified value m(x, y) and diffuses the quantized error e(x, y).

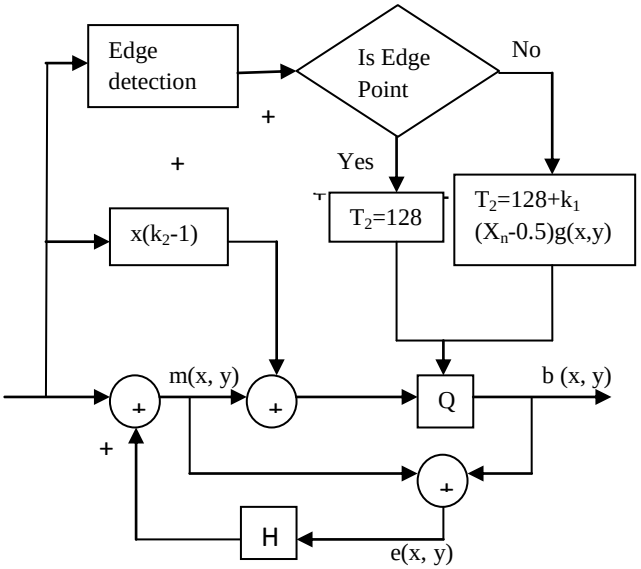

Fig1:Block diagram of chaotic and edge enhanced error diffusion algorithm

4.1 Gradient –Based Edge Detection

In this subsection, we discuss about the steps in edge detection. Edge detection is the one of the fundamental steps in image processing, image analysis, image pattern recognition and computer vision analysis. First, the image is filtered to improve the performance of the edge detector with respect to noise. Then the filtered image is enhanced and detected. And finally edge is located accurately and the edge orientation is estimated.

The proposed edge detection is the gradient based edge one, to reduce the computational complexity and it requires only four gradient values along with error diffusion directions.

$G_1 = |g(x, y)- g(x+1,y)|$

$G_2 = |g(x, y)- g(x-1,y+1)|$

$G_3 = |g(x, y)- g(x,y+1)|$

$G_4 = |g(x, y)- g(x+1,y+1)|$

4.2 Threshold Generation

Thresholding is the simplest method of image segmentation. From a grayscale image, thresholding can be used to create binary images.

If one of the above gradient values is not smaller than a predefined threshold T_1, the processed pixel is treated as an edge point and the quantization threshold T_2 is set to 128. Otherwise the quantization threshold is set to

$$T_2 = 128 + k_1 (X_n – 0.5)g(x,y)$$

where k_1 is the scaling factor, and its value is set to 1/64, generally. It can be adjusted according to different images to overcome worm-like artifacts.

X_n is a random number , whose value is generated by a quadratic iterator.

$$X_n = 4X_{n-1}(1 – X_{n-1})$$

The initial value of X_0 can be randomly generated from the interval [0,1]. In the chaos theorem, the generated values of X_n will remain in the interval, and chaos governs the whole interval from 0 to 1.The generation of X_n has brought us the following advantages. The computational complexity and

simplicity are obviously improved. Since a chaotic system behaves random behavior it can be used to avoid producing regular patterns. Worm-like artifacts can be effectively reduced by threshold T_2.

4.3 Quantization

Quantization involved in image processing is a lossy compression techniques achieved by compressing a range of values to a single quantum value. When the number of discrete symbols in a given stream is reduced the stream becomes more compressible. If we reduce the number of colors required then it reduces its file size. DCT data quantization used in JPEG and DWT data quantization used in JPEG 2000. When the quantization threshold is determined, the modified value m(x,y) is quantized and the output binary values are determined. The output value is given by,

$$b(x,y) = Q\{m(x,y) + (k_2 - 1)g(x,y)\}$$

$$= \begin{cases} 255 \text{ if } [m(x,y) + (k_2 - 1)g(x,y)] \geq T_2 \\ 0 \text{ ,otherwise} \end{cases}$$

Where k_2 is the predefined constant for edge enhancement. When k_2 is larger, the image information included in the thresholding decision will be larger, which makes the edge to be enhanced. The modified value m(x, y) can be written as,

$$m(x, y) = g(x, y) + \sum_{(i,j) \in N} h(i,j)e(x-i,y-j)$$

where the set N is a causal neighborhood and it can be

$$N = \{(i, j) \mid (i, j) = (1,1), (0, 1), (-1, 1), or(1,0)\}$$

The filter coefficients are same as in the standard error diffusion algorithm and are defined as h(1, 1) = 1/16, h(0, 1) =5/16, h(-1, 1) =3/16 and h(1,1) =7/16 respectively.

4.4 Error Diffusion

Compared with the standard error diffusion algorithm, the calculation for the error signal is not changed for chaotic and edge enhanced error diffusion algorithm.

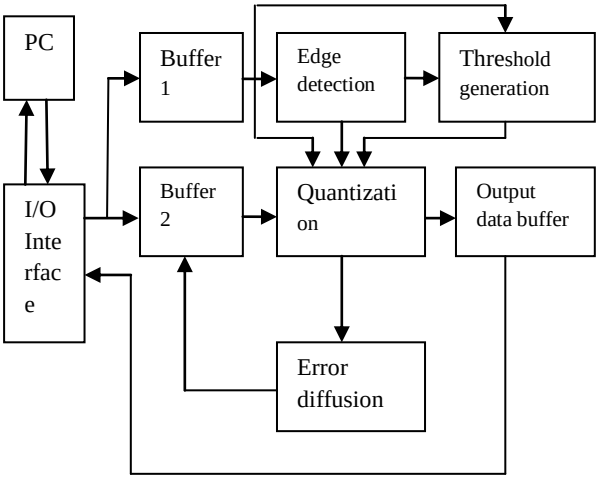

Fig 2 : The block diagram of chaotic and edge enhanced error diffusion algorithm

The quantized error is calculated by,

$$e(x, y) = m(x, y) - b(x, y).$$

The input value g(x, y) is multiplied by $(k_2 - 1)$, to control the amount of edge enhancement. The inclusion should be removed out while the quantization error is being calculated to keep the output gray level same as that of the input image.

The input image is loaded into two image data buffers. Data buffer 1 is used for edge detection and chaotic threshold generation. Data buffer 2 is used for quantization and error diffusion. The resultant binary image is stored in the Output data buffer for sending back to host PC. These hardware architecture are too based on four components as edge detection, threshold generation, quantization and error diffusion.

5. SIMULATION RESULT

The proposed algorithm is tested with random number array and the results are displayed.

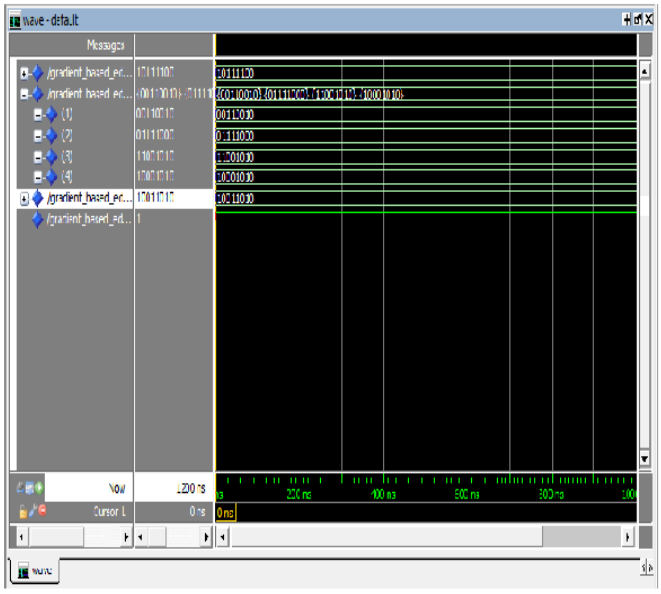

Fig 3 : Simulation result for gradient based edge detection.

Fig 3 shows the output of gradient based edge detection. Input gray level image g(x, y) and unprocessed pixels p(x, y) are first compared and sent to the AS module which performs the n-bit addition or subtraction according to the status of input m being 0 or 1. The operation of the AS module is tabulated in TABLE 1. The comparision result is denoted by the edge flag.

TABLE 1 Operation of AS Module

M	Input Status	s	C
0	x	a+b	Carry
1	a≥b	a-b	1
1	a<b	a-b	0

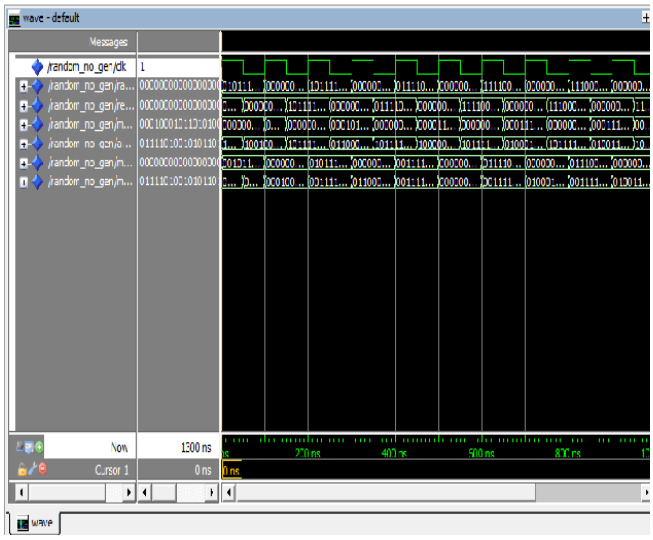

Fig 4 : Simulation result for random number generation.

Fig 4 shows the output of random number generation. 17 bits are used to represent the value of random number Xn and thus the value of the X_n is in the following range.

$$0 \leq X_n \leq \sum_{i=1}^{17} 2^{-i}$$

When X_n is obtained, the chaotic threshold T_2 can be generated from the circuit.

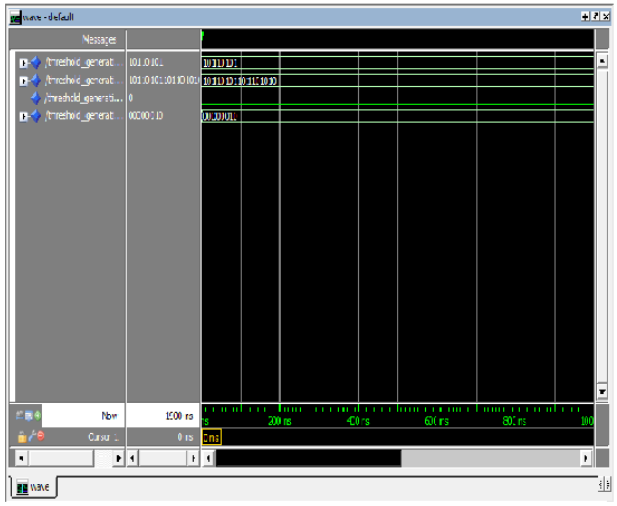

Fig 5 : Simulation result for threshold generation.

Fig 5 shows the output of the chaotic threshold generation. 8 bit input g(x, y), 17-bits generated random number and interval flag are the inputs .If the interval flag is 1, the generated value will be sent to one input of 18 *18 multiplier. Otherwise the value g(x,y) will be sent instead. The flag is set to 1 when the value of g(x,y) is in one of the preceding five intervals. Otherwise it is set to 0. To get the chaotic threshold T_2 we use 128 to add or subtract the extracted number depending on the bit of X_n. Finally the chaotic threshold was generated.

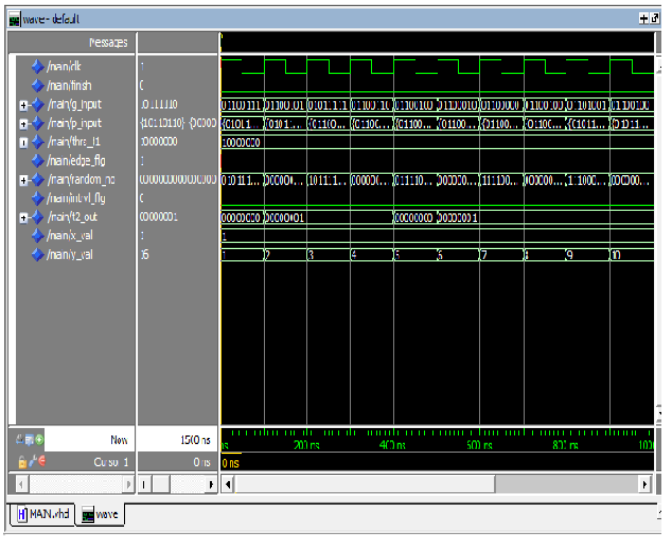

Fig 6 : Simulation result for edge detection,threshold output.

Fig 6 shows the simulation result for the gradient based edge detection and the chaotic threshold generation. The inputs for gradient based edge detection are g(x,y), p(x, y) and 8 bit threshold value. Edge flag will be output. For random number generation clock is the input and X_n random number will be the output. For threshold generation inputs are 17- bit generated random number, 8 bit g(x, y) and interval flag. T_2 will be the output.

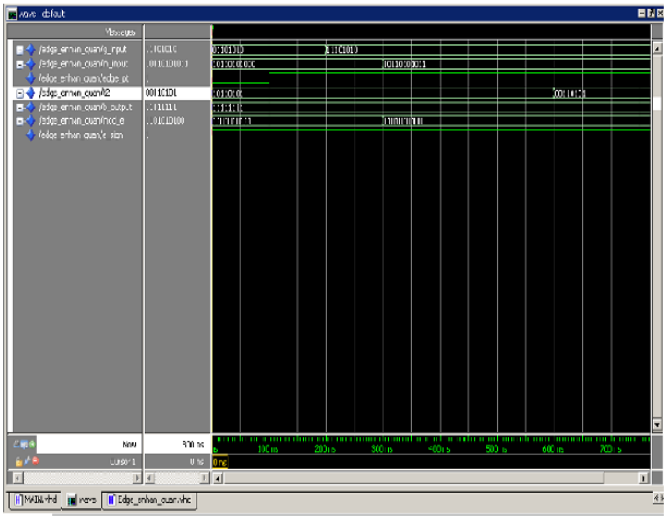

Fig 7 : Simulation result for edge enhanced quantization.

Fig 7 shows the simulation result for edge enhanced quantization. Here the modified value m(x, y) can also included as the input with g(x, y). Both are represented in 2's complement form. Since g(x, y) is always positive , adding three zeros to the left of its MSB does not change its sign and g(x, y) can be extented from 8 to 11. The resultant g(x,y) is added to m(x,y) by the 11-bit adder. The addition result is the sum. If the sum is larger than the quantization threshold and the MSB of the sum is o, then the output b(x, y)will be 255. Otherwise b(x, y) will be 0. The quantization threshold is T_2 or 128 depending on the Edge_point being 0 or 1.

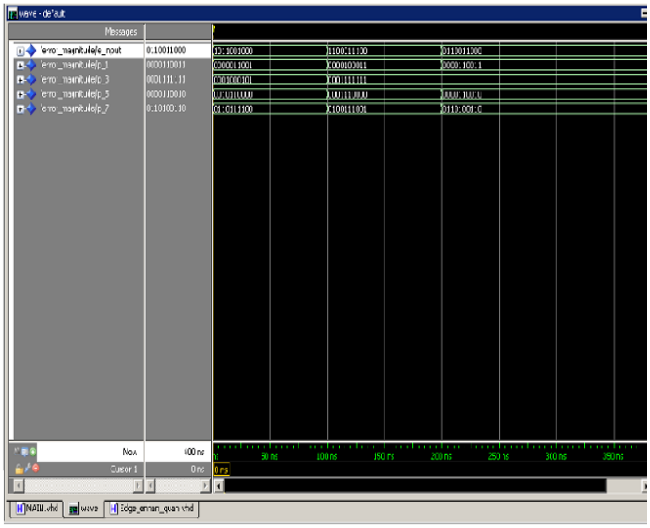

Fig 8 : Simulation result for error diffusion.

Fig 8 shows the simulation result for the error diffusion. To get the negative number –b(x,y), the number of bits are extended and 2's complement are taken. Then this number is added to m(x, y) to get the error e(x, y).

$$e(x, y) = m(x, y) – b(x, y)$$

After generating the error e(x, y) its magnitude bits and its sign bit are seperately extracted for error diffusion and data update. The error e(x, y) should be diffused into four unprocessed pixels with weighting factors : 1/16, 3/16, 5/16 and 7/16. And the diffused four error magnitudes are denoted as P_1, P_3, P_5 and P_7 respectively.

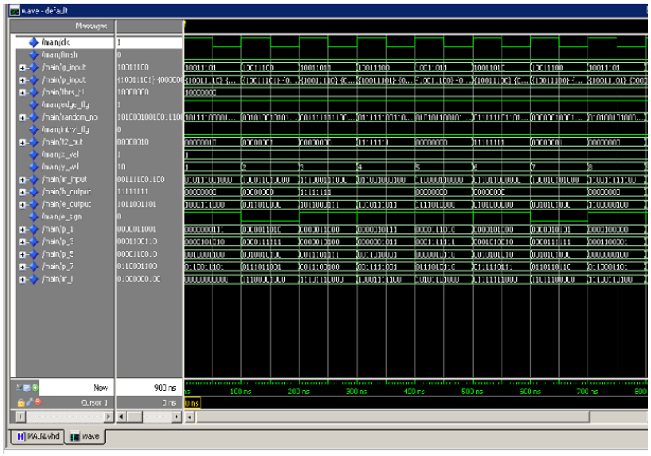

Fig 9 : Simulation result for main output.

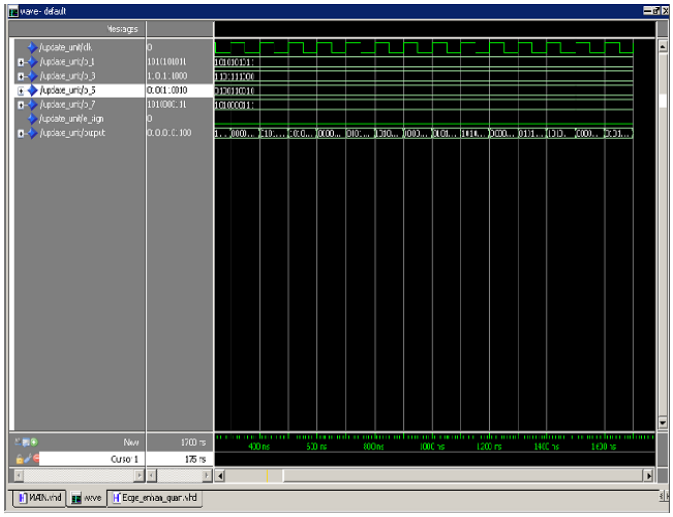

Fig 10 : Simulation result for update unit.

Fig 10 shows the simulaton result for the update unit. After generating the four magnitudes, it is used to update the data of m(x, y) at the corresponding locations. First the magnitude P_ i is combined with the error sign bit to generate the diffused error. Then the error is added to m_i by the 11-bit adder. The sum of the addition is the updated value of m_i and should be delayed by a clock.

6. FUTURE WORK

To obtain the error diffused output, array of pixel values are given as to the input blocks. The processed image will be displayed in Matlab.

7. CONCLUSION

This algorithm is modulized into four main components. Gradient based edge detection, Chaotic threshold generation, Edge enhanced quantization and error diffusion. With these results the algorithm contributes to high quality of images. The processing time needed in the proposed algorithm is quite really short. And also can be extended to process color image as well. This proposed algorithm also reduces the worm like artifacts in the smoother region.

8. ACKNOWLEDGEMENT

I am very greatful to the management, the principal and staff of Kumaraguru College of technology, Coimbatore, TamilNadu, India for encouraging this work.

I express my sincere thanks to Prof.S.Govindaraju for his kind support and valuable guidance in carrying out this research work.

9. REFERENCES

[1] Schumann. T, Susanti,A.R.D. "FPGA design for image processing using a GUI of a web-based VHDL Code Generator," *Proc. of IEEE International Conference on Visual Communications & Image Processing (VCIP)*, 2011.

[2] Swagata Samanta , Soumi Paik, Amlan Chakrabarti, "Design & Implementation of Digital Image Processing using FPGA" *LAP Lambert Academic Publishing*, 2011.

[3] C.-Y. Su and Y.-L. Shiue, "Edge-Adaptive Error Diffusion Using Chaotic Threshold Modulation," *Proc. of IEEE*

International Midwest Symposium on Circuits and Systems, MWSCAS 2009, pp. 718-721, August 2-5, 2009.

[4] W.-C. Kao, J.-A. Ye, and C. Lin, "Image quality improvement for electrophoretic displays by combining contrast enhancement and halftoning techniques," *Proc. of IEEE International Conference on Consumer Electronics*, 11.2-2, 2009.

[5] C.-Y. Hsu, C.-S. Lu, and S.-C. Pei, "Compression of halftone video for electronic paper," *Proc.* of *IEEE International Conference on Image Processing*, pp. 1600-1603, 2008.

[6] J.-M. Guo, "Improved block truncation coding using modified error diffusion," Electronics Letters, vol. 44, no. 7, pp. 462-464, March 2008.

[7] Z.-J. Liu, Z.-H. Liang, and C.-L. Liu, "3-dimensional error diffusion method based on edge detection for digital display devices," *IEEE Trans. Consumer Electronics*, vol. 53, no. 2, pp. 239-242, May 2007.

[8] T. Lin, "Probabilistic error diffusion for image enhancement," *IEEE Trans. on Consumer Electronics*, vol. 53, no. 2, pp. 528-534, May 2007

[9] X. Li, "Edge-directed error diffusion halftoning," *IEEE Signal Processing Letters*, vol. 13, no. 11, pp. 688-690, Nov. 2006.

[10] D. Kacker and J. P. Allebach, "Joint halftoning and watermarking," *IEEE Trans. Image Processing*, vol. 51, no. 4, pp. 1054-1069, 2003.

[11] N. Damera-Venkata and B. L. Evans, "Adaptive threshold modulation for error diffusion halftoning," *IEEE Trans. Image Processing*, vol. 10, no. 2, pp. 104-116, Feb. 2001.

[12] V. Ostromoukhov, R. D. Hersch, and I. Amidror, "Rotated dispersion dither: a new technique for digital halftoning," *Proc. of SIGGRAPH'94*, pp. 123-130, 1994.

[13] K. Knox and R. Eschbach, "Threshold modulation in error diffusion," *J. Electron. Imaging*, vol. 2, pp. 185-192, 1993.

[14] H.-O. Peitgen, H. Jurgens, D. Saupe, *Chaos and Fractals*, Springer- Verlag New York, 1992.

[15] R. Eschbach and K. T. Knox, "Error-diffusion algorithm with edge enhancement," *Journal of Optical Society of America*, vol. 8, no. 12, pp. 1844-1850, Dec. 1991 .

[16] R. Ulichney, "Dithering with blue noise," *Proc. IEEE*, vol. 76, pp. 56-79, Jan. 1988.

[17] R. W. Floyd and I. Steinberg, "An adaptive algorithm for spatial grayscale," *Proc. SID*, vol. 17, no. 2, pp. 75–78, 1976.

[18] R. Ulichney, *Digital Halftoning*. Cambridge, MA: MIT Press, 1987.

[19] P. Stucki, "Mecca—A multiple-error correcting computation algorithm for bi-level image hardcopy reproduction," *Tech. Rep. RZ1060*, IBM Res. Lab., Zurich, Switzerland, 1981.

[20]J. F. Jarvis, C. N. Judice, and W. H. Ninke, "A survey of techniques for the display of continuous-tone pictures on bilevel displays," *Comput.Graph. Image Process*, vol. 5, pp. 13–40, 1976.