Research Article

A Reconfigurable Memory based Fast VLSI Architecture for Histogram Computation

R. Amirtha Varshini[1*], K. Suganya[2]

[1] Research Scholar, Department of ECE, Perunthalaivar Kamarajar Institute of Engineering & Technology, Pondicherry University, South India
[2] Assistant Professor, Department of ECE, Sir Isaac Newton College of Engineering & Technology, Pappakoil, Nagapattinam, South India
*Correspondence: varshini1967@gmail.com

Article History:Received:11 november 2020; Accepted: 27 December 2020; Published online: 5 April 2021

Abstract: - Histogram computation is the crucial task used in processing so many image guided applications like pattern recognition, image segmentation etc. Image registration is one of the fundamental techniques for pre-processing of the images. Registration is the process of overlaying multiple images to geometrically align them. In medical Image processing, the improper registration can have negative impact on the analysis of the image which influences the final diagnosis. The accurate result of image registration is obtained by matching of multimodal images. Mutual Information is one of the commonly used techniques to find the similarity measurement between multi-modal images. Measurement of similarity requires a computation of histogram of individual images and joint histogram between the images. The hardware implementation of histogram computation has advantages in terms of flexible design, low power consumption, high speed, less execution time than the software implementation. This paper proposed a parallel algorithm for histogram computation. A memory based pipeline architecture is designed for implementing the proposed algorithm. The hardware mapping of the algorithm on FPGA is proposed and simulating them using Xilinx software tools.

Keywords: Histogram Generator, Image processing, Mutual Information, parallel algorithm.

1. Introduction

Recently, the integration of useful information from multiple images becomes a primary requisition process for various image guided applications like medical diagnostic and treatment, texture classification and image enhancement problems. The integration process comprises two steps called registration and fusion. Image registration [1-2] is the process of transforming multiple images with different modalities into spatial alignment and fusion is the method of integrating multiple images from different resources into a single image accurately and displays the integrated data. Automatic image registration [3-5] is one of the essential steps in image processing for geometrically align two different images that are obtained at different points of view and sensors. The image registration can be implemented either by Transformation model or by the similarity measurement.

Images can be aligned by applying suitable transformation model; mostly affine transformation model is used. The objective of the transforms [6] is to bring out the different set of data into a single coordinate system. The appropriate model should be selected for mapping the transformation. The deciding of mapping function and estimating the parameters that associated with mapping function are the main issue in the transformation model. Image registration process can be speed up by executing the program in parallel. The Graphics Processing Units (GPU) [7-9] is programmable parallel processor that parallelizes the computation of transformation and joint histogram. GPUs outperform the computation speed of CPU with single processor. Image Alignment between two images can be accomplished either based on the intensity of the images or based on the features of the images or sometimes both of these methods.

In intensity based model [10], intensity patterns of one image is compared with pattern of other images and also the degree of similarity between two intensity patterns is quantified. Mutual information is the similarity measurement techniques for the intensity based multimodal images. Mutual information has advantage of accurately registering multimodality images by measuring alignment of signals where one can guess another. The computation of Mutual information comprises three steps; first it computes the smoothed histogram of individual images and then calculates joint histogram that counts the number of times the gray correspondence occurs between two images. In second step, calculates the marginal probability density and joint probability density function. In third step, the mutual information is calculated by finding the entropy of the individual images and joint entropy between two images.

The histogram is one of the frequently used basic computations in image registration technique, and it has the advantages of fast computation, invariant to rotation and determines the performance of the operations.

[*] Corresponding author: R. Amirtha Varshini
Research Scholar, Department of ECE, Perunthalaivar Kamarajar Institute of Engineering & Technology, Pondicherry University, South India
varshini1967@gmail.com

Image histogram is a graphical representation of digital image that show the frequency of each gray level occurrence. The histogram of a digital image is calculated using the discrete function

$$h(r_l) = n_l \qquad (1)$$

where $r_l$is the l^{th} gray level and n_l is the number of pixels in the image at gray level r_l , l lies in the range [0,L-1]. In general, the joint histogram computation between two images is calculated using the formula

$$h = \begin{bmatrix} h(0,0) & h(0,1) & \cdots & h(0,j-1) \\ h(1,0) & h(1,1) & \cdots & h(1,j-1) \\ \vdots & \vdots & \cdots & \vdots \\ h(i-1,0) & h(i-1,1) & \cdots & h(i-1,j-1) \end{bmatrix} \qquad (2)$$

where $i-1$ denotes the intensity of first image x, $j-1$ denotes the intensity of the second image y , $h(i,j)$ denotes the number of corresponding pairs having intensity value i in the first image and j in the second image.

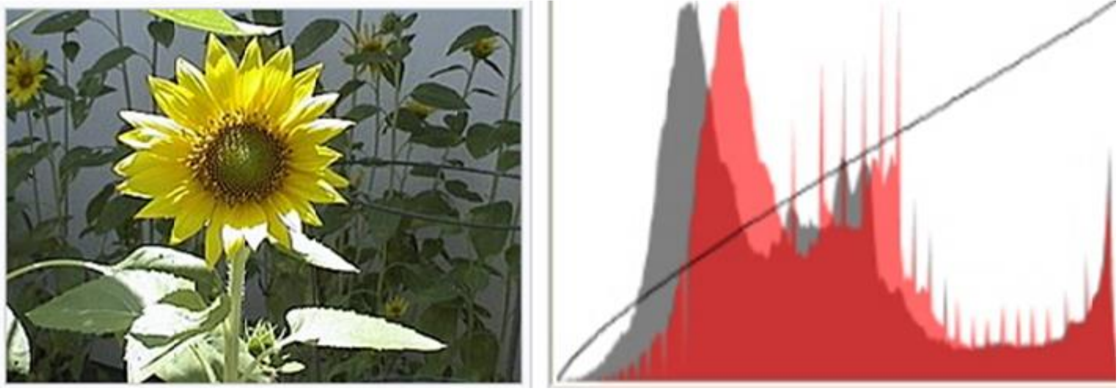


Figure 1. Image and the corresponding histogram

The color image and the corresponding histogram are shown in Figure 1. The histogram graph is a two dimensional coordinate system in which the tonal variation is represented along x axis and the total number of pixels in the particular tone is represented along y axis.

```
Algorithm Histogram_Computation
Begin
        Initialize, histogram [0 to 255]=0
for i=0 to image_height
for j=0 to image_width
Begin
Intensity=gray value of gray image at pixel (i,j)
                        Histogram [intensity] +=1
```

The histogram analysis made based on the gray scale values of two peaks that indicates the foreground and background of the images. After smoothing of the histogram, the foreground is separated from the background by finding the threshold value using either with local maximum and minimum or with statistical methods. The brightness, contrast, and intensity of the images can be processed more efficiently using histogram computation when compared to transformation technique. Most of the applications implements histogram generator as a software program as shown in Figure 2. Histogram computation process can be speedup by applying parallelism in the operation as well as in the hardware architecture designed for histogram generator. This paper proposed a new memory based parallel algorithm for computation of histogram and the implementation of corresponding VLSI architecture for it.

The rest of this paper is organized as follows. Section 2 discusses survey of existing architecture, Section III describes proposed algorithm and discusses the overall architecture design, and Section IV discusses simulation of hardware implementation on FPGA using Xilinx. Section V concludes.

2. Related Works

The software implementation of histogram computation faces the following constraints. A separate software services has to be installed in the system for any small modification in the feature of the histogram. The software program for histogram generation needs compatible processors for executing them. The cost of buying software license has to be considered for mass production. The hardware implementation doesn't have these constraints. For any modification in feature of histogram computation, it is just enough to reconfigure the chips according to the need.

The hardware implementation of histogram computation mechanisms can be categorized as memory based or array of counter based. Scalar Histogram Algorithm (SHA) [11] is the first software program of histogram computation. According to this, the given image split into two sub images based on their pixel locations. The splitting is applied based on odd and even pixel positions. The histogram for the sub images is computed separately using memory locations. At the end of the process, two histograms are joined by adding their corresponding memory locations. This is the first memory based histogram computation that produces good efficiency for small size images, but not suitable for histogram computation of large size images due to the requisition of additional memory and clock cycles for adding. It requires one extra 256×256 size memory for producing joint histogram computation of pixel size of 8 bit.

The first hardware implementation with read-modify-write architecture [12] is proposed. The capability of using the resources efficiently makes this architecture to be explored in many applications. The performance of this architecture depends on the ports of the memory array. The performance can be speedup by introducing dual port memory based read-modify-write architecture, but anyway in memory based model the performance is restricted by the size of the memory. To overcome these limitations, the architecture based on array of counters is proposed. The architecture based on counter have restriction in speed and costly in resource utilization due to the usage of one counter for each histogram bin.

A novel hardware architecture based on array of counters [13] is proposed for computing histogram in parallel manner. The parallelism can be achieved by arranging the array of counters in pipelined fashion working on the stream of two pixels per clock cycle. For four pins, one processing unit is used. The pixel size of input image is 8 bit each. The architecture utilizes 64 processing cell that are interconnected in pipeline fashion. The computation of joint histogram between two images requires 256×64 processing cells. The number of processing cell required increases linearly with the size of the histogram computation, thereby it also increases delay and decreases throughput.

An improvement in the performance of parallel pipelined array of cells via C-slow retiming technique [14] is introduced for real time histogram computation. It takes m-bins as input data items. A stream of 1-bit pixel per clock is employed for histogram computation. C-slow retiming is the technique of introducing an array of register in the path of feedback, thereby it reduces the delay in the feedback and increases the throughput of the pipelined array of cells by 25%. Further performance is improved by arranging the pipeline array to accept 2 bits per clock cycle. Each cell processes 2 bins. The computation of histogram requires $\mathrm{n}/2 + \mathrm{m}/2$ clock cycles, where n denotes the number of input data items and m denotes the number of bins processed at a time. The latency obtained is $\mathrm{m}/2$. Thereby the performance is improved when compared to existing pipelined architecture. The pipeline architecture with array of cells that process one or two histogram bin can generates high latency and also increases the delay. Moreover, if the number of pins to be processed within the single processing cell increases, then the corresponding architecture becomes more complex. This method may not suitable for joint histogram computation

The Parallel Array Histogram Architecture (PAHA) based on the array of register [15] is proposed for histogram computation. The architecture doesn't use pipelining techniques. The M parallel input, each with the size of N-bit if fed into the $\mathrm{N}: 2^{\mathrm{N}}$ decoder. The 1 bit output of 2^{N} decoder is fed into the 2^{N} accumulator. Each accumulator is associated with register bins. The accumulator is used to accumulate 1 bit output coming from the decoder, add them, actually M, 1 bit is added and store it in register. The size of the register is decided in such a way that it can accommodate M parallel input. PAHA with variable number of inputs called Flexible Parallel Array Histogram Architecture (FPAHA) also proposed. The register array based architecture reduces the delay and latency compared to pipeline architecture, so that it produces better throughput and resource utilization efficiency.

An array of counter based hardware architecture for generating the histogram of gray scale image of size (256×256) [16] is proposed. The Field Programmable Gate Array (FPGA) and Programmable SoC (PSoC) based hardware implementation is utilized in this architecture. The architecture performs two operations mainly, computation of histogram for the given input image and displaying the computed histogram. The given input image is stored in ROM. The image size is $(2^{16} \times 8\ )$ bits. The histogram generator consists of (8×2^{8}) decoders and an array of counter. The decoder receives 8 bit pixel value as input and triggers any one of the 256 possible output based on the input. The OR output of the decoder is combined with clock signal to act as input to the AND gate which selects a counter for storing gray level information out of 256 possible array of counters. The pixel value is incremented by the increment block and then storing operation is repeated. The comparison block keeps track of gray levels for histogram display.

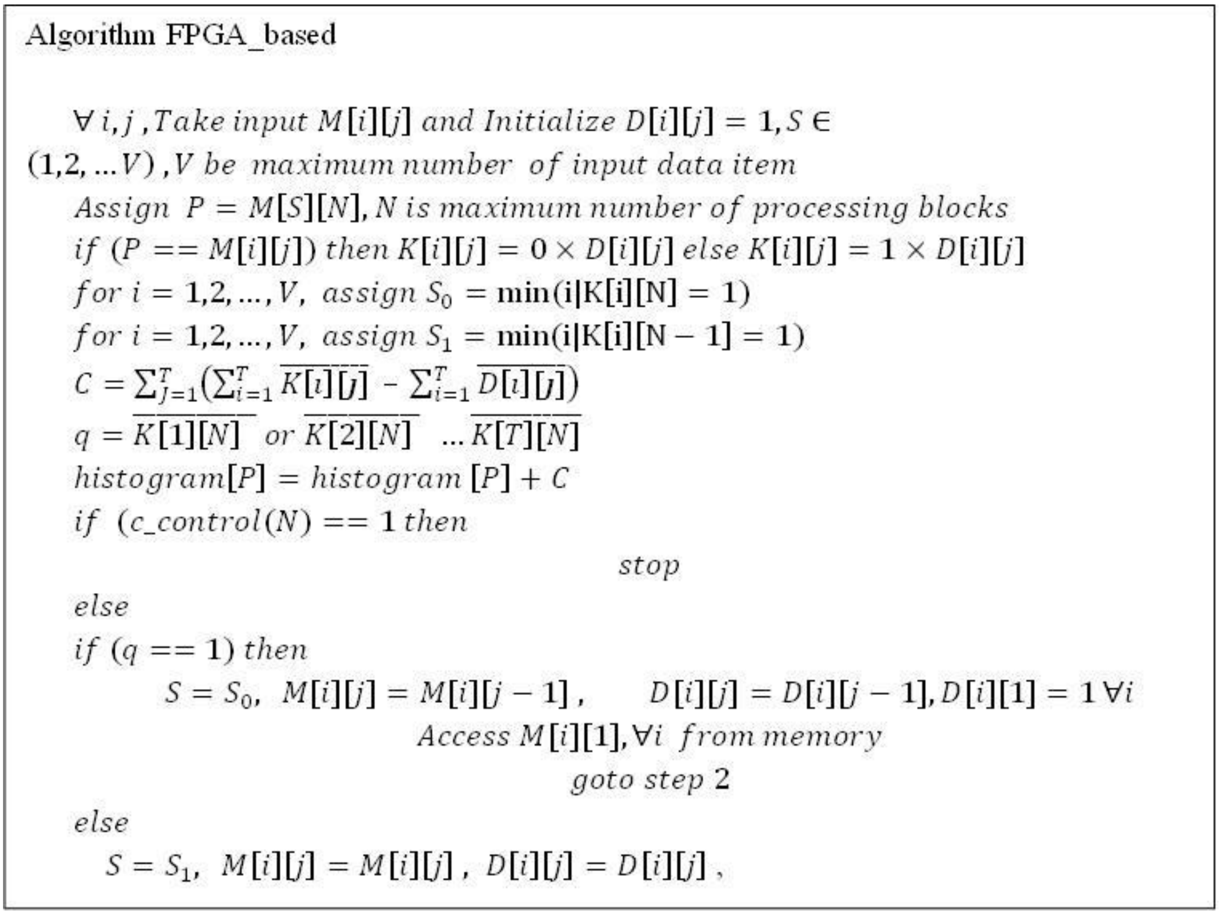
In this paper a new hardware architecture based on memory is developed and parallel algorithm for computing joint histogram is implemented. The corresponding VLSI architecture that make use of less hardware resources is proposed. The architecture produces better performance for computing histogram even when size of the image is larger.

3. Proposed Architecture

The proposed architecture consumed less hardware resources than the existing architecture surveyed in the literature. When the number of processing blocks in the architecture increases, the execution time decreases for joint histogram computation. At the same time, the hardware resources utilization also increases with the processing blocks.

a) Algorithm for FPGA based Architecture

The algorithm based on FPGA implementation for histogram computation is shown in Figure 3.



```
Algorithm FPGA_based

    $\forall i,j$ , Take input $M[i][j]$ and Initialize $D[i][j] = 1, S \in$
$(1,2, \ldots V)$ , $V$ be maximum number of input data item
    Assign $P = M[S][N]$, $N$ is maximum number of processing blocks
    if $(P == M[i][j])$ then $K[i][j] = 0 \times D[i][j]$ else $K[i][j] = 1 \times D[i][j]$
    for $i = 1,2,\ldots,V$, assign $S_0 = \min(i|K[i][N] = 1)$
    for $i = 1,2,\ldots,V$, assign $S_1 = \min(i|K[i][N-1] = 1)$
    $C = \sum_{j=1}^{T}(\sum_{i=1}^{T}\overline{K[i][j]} - \sum_{i=1}^{T}\overline{D[i][j]})$
    $q = \overline{K[1][N]}$ or $\overline{K[2][N]}$ ... $\overline{K[T][N]}$
    $histogram[P] = histogram[P] + C$
    if $(c\_control(N) == 1$ then
                                        stop
    else
    if $(q == 1)$ then
            $S = S_0$, $M[i][j] = M[i][j-1]$ ,     $D[i][j] = D[i][j-1], D[i][1] = 1\,\forall i$
                                Access $M[i][1], \forall i$ from memory
                                        goto step 2
    else
        $S = S_1$, $M[i][j] = M[i][j]$ , $D[i][j] = D[i][j]$ ,
```

Figure 3. FPGA based Algorithm for Histogram Computation

a) Architectural block diagram

The architectures block diagram to implement the algorithm shown in Figure 3. This architecture is applicable for any grayscale image of size (256 x 256)

b) Architectural block diagram

The architectures block diagram to implement the algorithm is shown in Figure 4. This architecture is designed for gray scale image of size (256 x 256). The algorithm for histogram computation is implemented on Field Programmable Gate Array (FPGA). The architectural block consists of array of processing blocks. The processing blocks are interconnected in a pipeline fashion. The leftmost processing block obtains its input from memory directly. The input to the i^{th} processing block taken from $M[k][i]$, where $k = \{1,2, \ldots V\}$. The input of the i^{th} processing block is shifted to the $(i+1)^{th}$ processing block from left to right based on the control signal ‘q’. Initially the signal ‘t’ is set to high. The selection unit is used to select the data ‘P’ from the data that are not processed yet by any processing blocks and fed ‘P’ into all processing blocks. Each processing block compares their input with the data ‘P’ and finds the matching. If the match is found, then the status bit is set with the data item to indicate the data item is counted. When the data items are shifted to the next processing block, the status bit associated with data also passed. The signal *mem_ads* act as a counter. It starts to count only when ‘q’ is high. When the data items reach N^{th} processing block, the timing signal ‘t’ is set to zero. It maintains zero until the histogram is computed. The *mem_ads* signal reset to zero for processing next block of data. The process is repeated until the final block of data processed by the N^{th} processing block. The processing block stops its operation in a linear fashion from left to right.

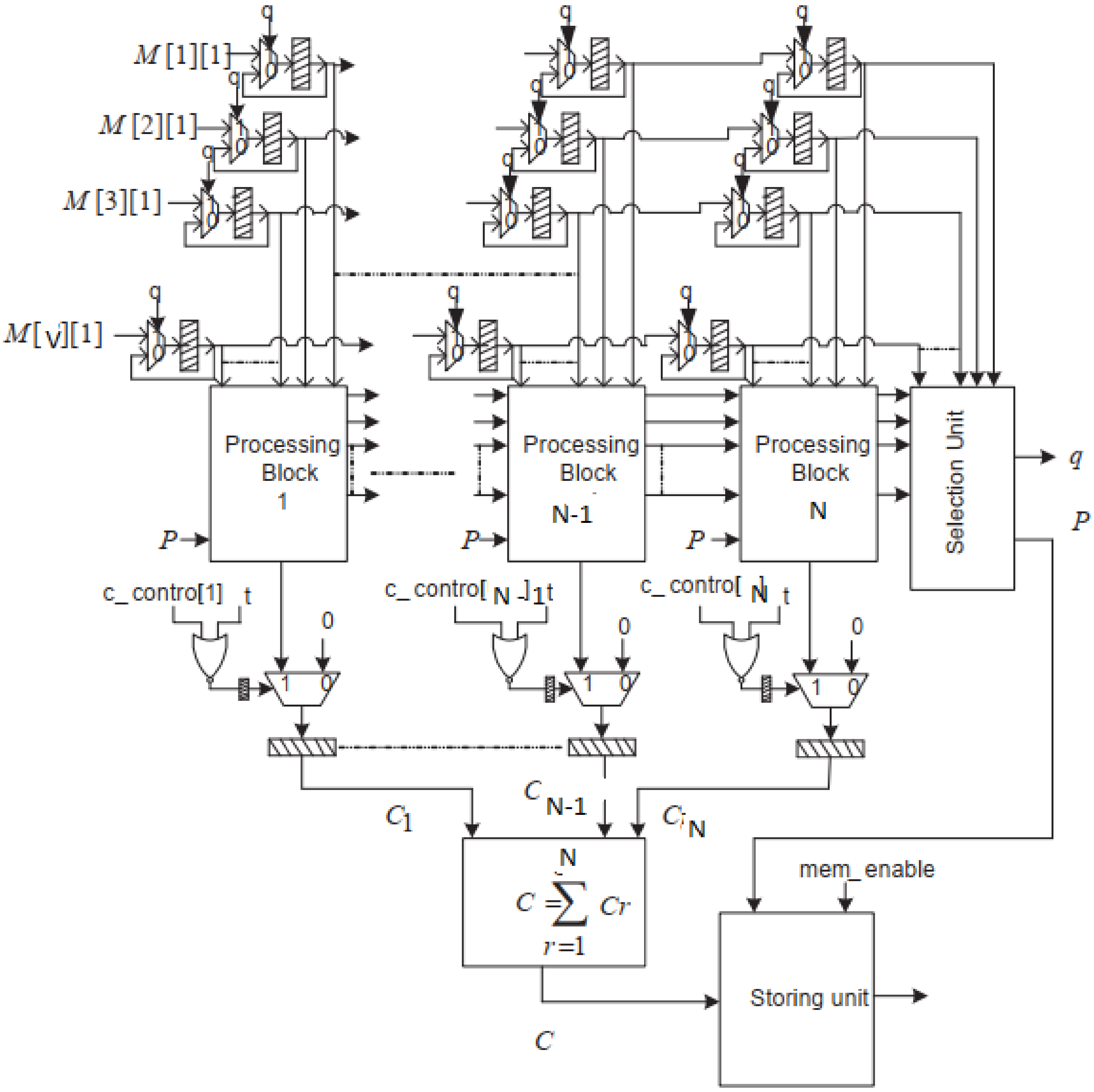


Figure 4. Architectural block diagram for Histogram computation

c) Functions of Processing Block

The hardware design of r^th processing block is shown in Figure 5. The comparator evaluates input data with ‘P’. If the match found, then comparator produces 0 otherwise 1. The output of the comparator and the previous output of the AND gate are taken as two inputs of AND gate. The output of the AND gate is fed into the MUX. The output of MUX is stored in a register and the register maintains the output until the signal ‘q’ is low. If ‘q’ is high, then MUX will take the value from the previous processing block.

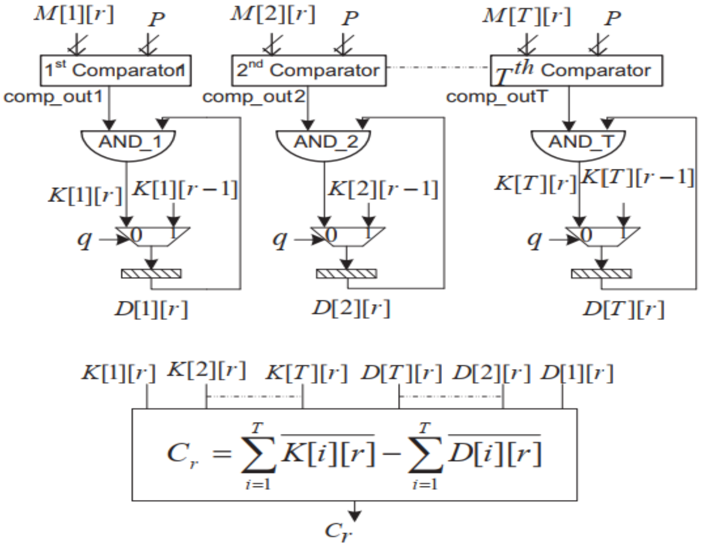


Figure 5. Functions of Processor Block

d) Functions of Selection Unit

The design of selection unit shown in Figure 6 consists of two priority encoders. The control signal ‘q’ is obtained by feeding the data items already counted into the NOR gate. The 2:1 Mux$_0$ is used to select S as either S$_0$ or S$_1$depending on the control signal ‘q’. The value of P is selected from the current processing blocks based on the value of S using T:1 Mux$_1$.

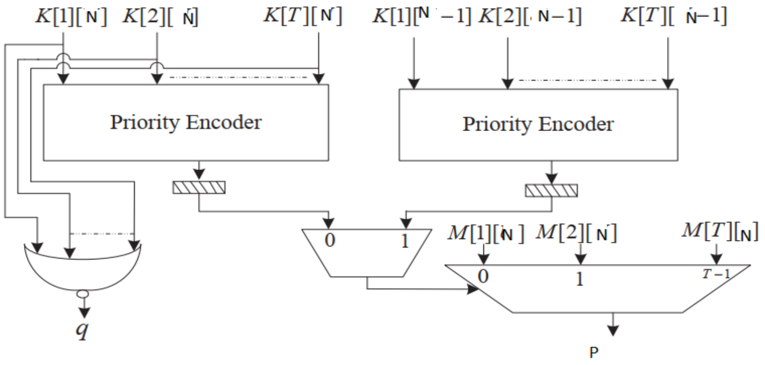


Figure 6. Functions of Selection Unit

e) Functions of Storing Unit

Dual port memory based RAM is shown in Figure 7 used for storing computed histogram. RAM consists of two ports, PortA and PortB. The PortA is used for writing and PortB is used for reading the content. The value of P is taken from the value of C. When the signal *mem_enable* is low, the content of R3 register is allowed through the multiplexer.

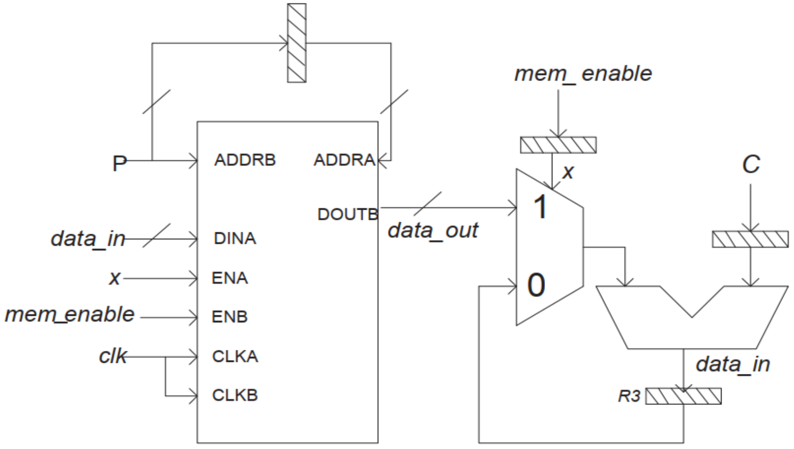
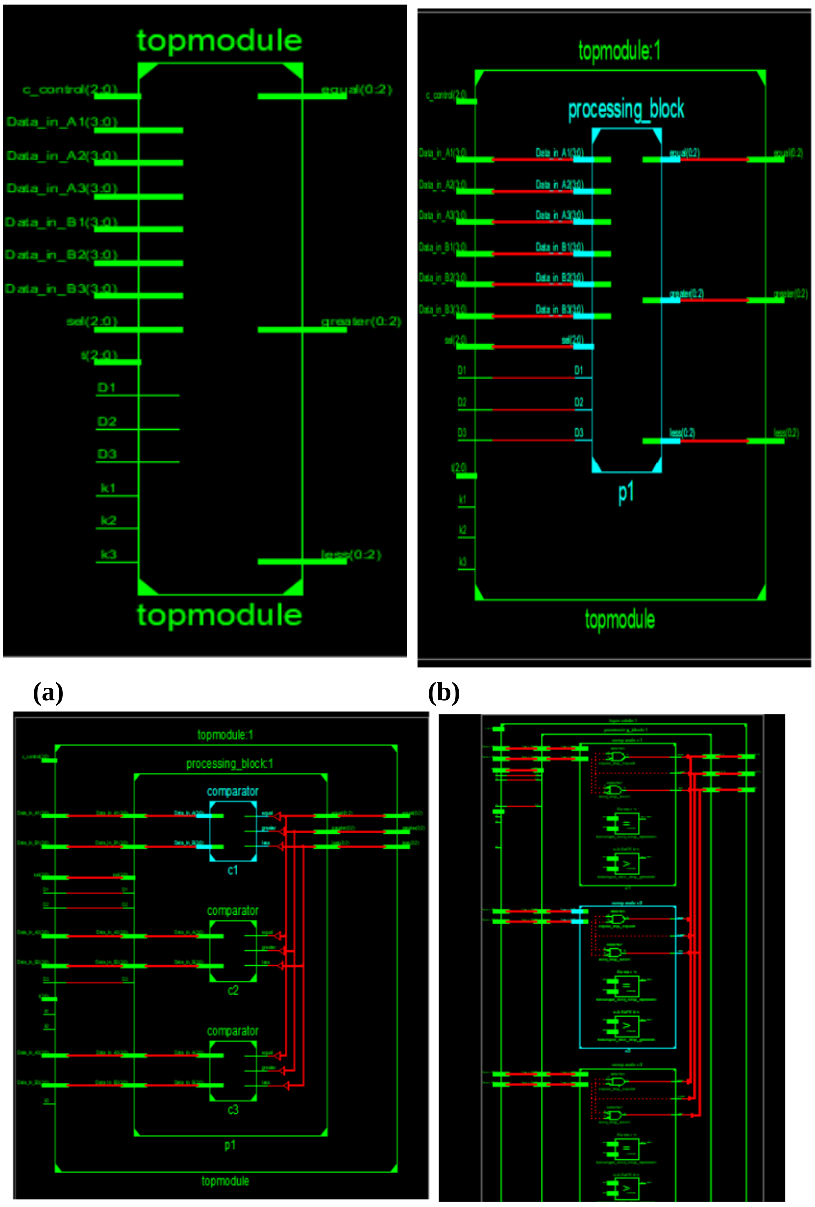


Figure 7. Functions of Storing Unit

4. Implementation and Simulation Results

The use of prototyping tools such as MATLAB-Simulink and Xilinx System Generator becomes increasingly essential for designing hardware architecture. A methodology for implementing histogram generation on a reconfigurable logic platform using Xilinx System Generator (XSG) for MATLAB is developed. The methodology aims to improve the design verification efficiency for complex system. It presents hardware architecture for computing joint histogram using Xilinx System Generator.



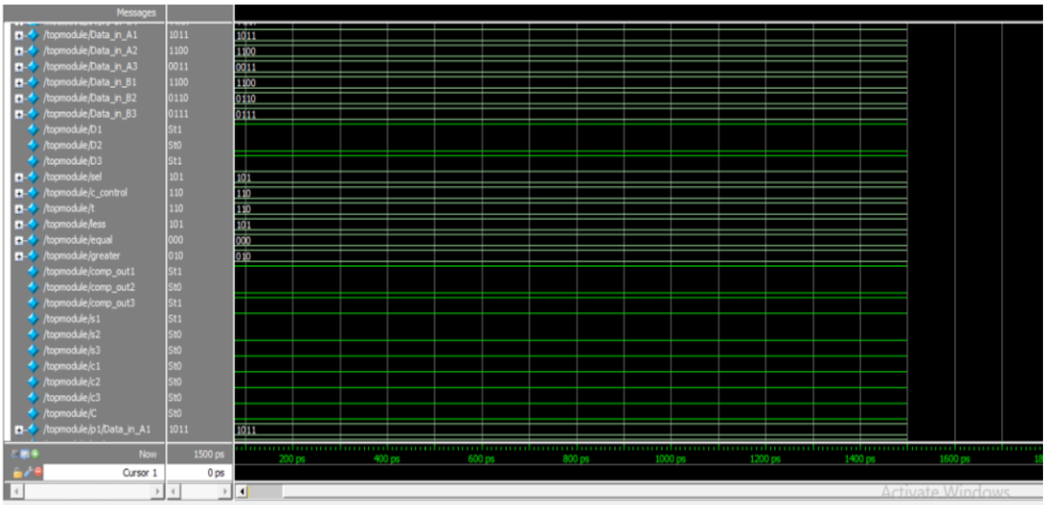
(a) (b)

(c) (d)

Figure 8. Proposed histogram generator a) Block diagram b) Internal Architecture of Processing Block c) Internal Architecture of Comparator d) RTL schematic

The proposed design efficiency is assessed by implementing the algorithm on MATLAB. The logical design has to be synthesized and implemented before checking its correctness. Using Xilinx ISE CAD tool, all the designs are synthesized on FPGA and implemented using Verilog in order to extract the hardware resources. The Register Transfer Level (RTL) used to design the schematic diagram. Register Transfer level concept is employed in a Hardware Description Language (HDL) like VHDL, Verilog. The RTL aims to create high level representation of the circuit. The low level representation of the circuit is obtained from the high level representation that enhances the activity of deriving actual wiring design. RTL has the benefits of acquiring optimized design flow and reduces the complexity of the architectural design for large chips. The verification step is carried out at the initial stage of RTL simulation to verify the logic of the design. The syntax error is removed from VERILOG code at this step and it also reduces the long synthesis time

A hardware simulation block can be generated after verification step and FPGA can be programmed for implementation. Verilog codes are automatically generated from the system generator block sets. Mentor Graphics ModelSim tool is used to describe post simulation steps. The Veilog codes are then synthesized using Xilinx. The architectural block diagram, RTL schematic diagram for processing blocks, comparator, and proposed histogram generator is shown in Figure 8. From the simulation results shown in Figure 9, it is observed that proposed design requires very small hardware resources for implementation compared to the conventional design.



Messages	
/topmodule/Data_in_A1	1011
/topmodule/Data_in_A2	1100
/topmodule/Data_in_A3	0011
/topmodule/Data_in_B1	1100
/topmodule/Data_in_B2	0110
/topmodule/Data_in_B3	0111
/topmodule/D1	St1
/topmodule/D2	St0
/topmodule/D3	St1
/topmodule/sel	101
/topmodule/c_control	110
/topmodule/t	110
/topmodule/less	101
/topmodule/equal	000
/topmodule/greater	010
/topmodule/comp_out1	St1
/topmodule/comp_out2	St0
/topmodule/comp_out3	St1
/topmodule/s1	St1
/topmodule/s2	St0
/topmodule/s3	St0
/topmodule/c1	St0
/topmodule/c2	St0
/topmodule/c3	St0
/topmodule/C	St0
/topmodule/p1/Data_in_A1	1011
Now	1500 ps
Cursor 1	0 ps

Figure 9. Simulation results of the Verilog

5. Conclusion

In this paper, the parallel architecture generates a histogram for gray scale image of size (256 x 256) with each pixel size of 8 bit is proposed. The performance of the proposed memory based parallel algorithm has been evaluated. The proposed algorithm for histogram computation is implemented and simulated on MATLAB using Verilog. The hardware implementation on FPGA works on the basis of logic blocks interconnected by the programmable logic technologies. The FPGA has the characteristics of reprogramming the desired functionality after manufacturing. Programming logic technology with FPGA has the advantages of providing flexibility, cost saving compared to silicon, raising the performance of hardware by parallelism. The performance is increased efficiently with the size of images. There is no significant improvement in the utilization of hardware resources as the image size is increased.

References

Hiba A. Mohammed. The Image Registration Techniques for Medical Imaging (MRI-CT). American Journal of Biomedical Engineering (2016); 6(2): 53-58. doi: 10.5923/j.ajbe.20160602.02

Oliveira, Francisco & Tavares, Joao. Medical image registration: A review. Computer methods in Biomechanics and Biomedical Engineering. (2014);17: 73-93. doi: 10.1080/10255842.2012.670855.

Wahed M, Eltaweel, Ghada, El-karim, Gad, A. Automatic Image Registration Technique of Remote Sensing Images. International Journal of Advanced Computer Science and Applications. (2013); 4 (2). doi: 10.14569/IJACSA.2013.040227.

Tramnitzke, Florian, Rühaak, Jan, König, Lars, Modersitzki, J, Köstler, Harald. GPU Based Affine Linear Image Registration using Normalized Gradient Fields. (2014); doi:10.13140/2.1.3642.7528.

Xu L, Wan JWL. Real-time intensity-based rigid 2D-3D medical image registration using RapidMind Multi-Core Development Platform. in Proc. IEEE 30th Annu. Int. Conf. Eng.Med. Biol. Soc., Vancouver, Canada, (2008); 5382-5385. doi:10.1109/IEMBS.2008.4650431

Salem, Nema & Malik, Hebatullah & Shams, Asmaa. (2019), “Medical image enhancement based on histogram algorithms”, Procedia Computer Science, 163, 300-311. 10.1016/j.procs.2019.12.112.

NVIDIA Corporation: NVIDIA CUDA (2014) C Programming Guide.

Otake Y, Armand M, Armiger RS, Kutzer MD, Basafa E, Kazanzides P, Taylor RH. Intraoperative image-based multiview 2D/3D registration for image-guided orthopaedic surgery: incorporation of fiducial based C-arm tracking and GPU-acceleration. Medical Imaging. IEEE Transactions on Medical Imaging. (2012); 31 (4): 948-962.

Owens JD, Houston M, Luebke D, Green S, Stone JE, Phillips JC. GPU computing. Proceedings of the IEEE, (2008); 96 (5): 879-899

Vidyasaraswathi H, & Hanumantharaju M., “Review of Various Histogram Based Medical Image Enhancement Techniques. (2015); doi: 1-10.1145/2743065.2743113.

Shahbahrami A, Hur JY, Juurlink BHH and Wong S. FPGA implementation of parallel histogram computation. in Proc. 2nd HiPEAC Workshop on Reconfigurable Computing, Gothenburg, Sweden, (2008); 63-72.

Muller S. A new programmable VLSI architecture for histogram and statistics computation in different windows. Proceedings., International Conference on Image Processing, Washington, DC, USA, 1995; 1: 73-76. doi: 10.1109/ICIP.1995.529042.

Cadena J, Sherratt RS, Huerta P, and Kao WC. Parallel pipelined array architectures for real-time histogram computation in consumer devices. IEEE Transactions on Consumer Electronics. (2011); 57 (4): 1460 – 1464. doi: 10.1109/TCE.2011.6131111.

Cadenas Medina J, Sherratt S, Huerta P, Kao, W. and Megson GM. Parallel pipelined histogram architecture via c-slow retiming. In: Proceedings of the IEEE International Conference on Consumer Electronics(ICCE). IEEE, (2013); pp. 230-231. doi: 10.1109/ICCE.2013.

Gan, Qifeng, & Langlois JM, Pierre, Savaria, Yvon. Parallel array histogram architecture for embedded implementations. Electronics Letters, (2013); 49 (2): 99 - 101. 10.1049/el.2012.2701.

Hazra, Sambaran, Ghosh, Sudip & Maity, Santi & Rahaman, Hafizur. A New FPGA and Programmable SoC Based VLSI Architecture for Histogram Generation of Grayscale Images for Image Processing Applications. Procedia Computer Science. (2016); 93: 139 – 145. doi: 10.1016/j.procs.2016.07.193.

Li Yao & Sohail Muhammad. A novel technique for analysing histogram equalized medical images using super pixels. Computer Assisted Surgery. (2019); 24 (1): 53 – 61. doi: 10.1080/24699322.2018.1560100

Dinggang Shen. Image registration by local histogram matching. Pattern Recognition. (2007); 40: 1161 – 1172.