Special Issue Published in International Journal of Trend in Research and Development (IJTRD), ISSN: 2394-9333, www.ijtrd.com

Low cost & High quality Color Demo Saicking Spurious Power Suppression VLSI Design For Real Time Video Applications

[1]A.Indhumathi,[2]K.Shanmuga Priya,[3]P.Anitha
[1,2]Student, Dept of ECE, Ranganathan Engineering College, Coimbatore, India
[3]Assistant Professor, Dept of ECE, Ranganathan Engineering College, Coimbatore, India

Abstract: This project presents a low cost and high quality pipelined color demosaicking design using spurious power suppression technique. The main objective is to improve the quality of reconstructed images; a linear deviation compensation scheme was created to increase the correlation between the interpolated and neighboring pixels. Furthermore, immediately interpolated green color pixels are first to be used in hardware oriented color demosaicking algorithms, which efficiently promoted the quality of the reconstructed image. A boundary detector and a boundary mirror machine were added to improve the quality of pixels located in boundaries. In addition, a hardware sharing technique was used to reduce the hardware costs of three interpolators. By using the spurious power suppression technique the usage of power is minimized. Also the usage of adders and subtractors has been reduced. The gate counts and the core area have been greatly reduced using this technique.

Keywords*: Boundary mirror, Boundary detector, Linear deviation, Color filter array, Demosaicking, Hardware Sharing*

I. INTRODUCTION

The most widely used format of CFAs in modern electronic products is Bayer CFA, as shown in Figure. 2.1 in which each pixels contains one of the red, green, and blue colors. Since two-thirds of color information is missed after using CFA, it is necessary to interpolate missing values back to the CFA image to restructure a full color image. Recently, some high-performance and high-quality color demosaicking and interpolation algorithms have been proposed. An orientation-free edge strength filter is proposed. It utilized edge information to avoid averaging non-correlated color differences and improved demosaicking performance.

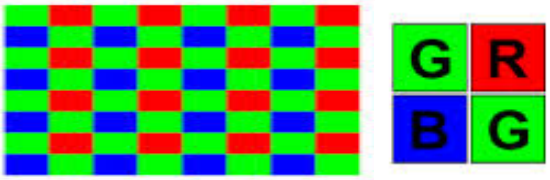


Figure 1.1 Bayer Pattern GRBG

A stochastic estimation approach to adaptive interpolation of CFA was proposed. Gradient edge detection masks and adaptive heterogeneity projection method for demosaicking the CFA was proposed by Chung et al. In order to avoid artifacts such as zipper effect, blur, and color spots, a self-similarity color demosaicking algorithm was exploited to interpolate missed colors, which can be treated more generally as graph-based regularization to the image. In addition, another novel graph-based regularization framework was presented, in which a weight matrix was built to measure similarity and a static Laplacian was used to solve a variation problem.

The artificial effect can be alleviated efficiently. Recently, a voting-based directional demosaicking algorithm has been proposed. It uses a voting-based edge direction detection method and a directional weighted method to improve the quality of reconstructed images. Although the color demosaicking and interpolation algorithms, as aforementioned, achieved high quality, it is hard to realize these algorithms by using the very-large-scale integration

II. REVIEW STAGE

S. C. Hsia, M. H. Chen, and P. S. Tsai, "VLSI implementation of low power high-quality color interpolation processor for CCD camera," IEEE Trans. VLSI Syst., vol. 14, no. 4, pp. 361–369, Apr. 2006 presented a color interpolation techniquefor a single-chip charge-coupled device with color-filter-array format. This proposes an edge-direction weighting and the local gain approach to reconstruct missing color components. With the time-sharing method, the VLSI architecture can interpolate various colors using a common computational kernel, reducing the circuit complexity. The prototype of the color interpolation processor has been successfully verified with a field-programmable gate array device. The chip only uses about 10K gates and two line buffers. In this design, the chip area cannot decrease obviously because the dividers and multipliers were used.

S. L. Chen and E. D. Ma, "VLSI implementation of an adaptive edge-enhanced color interpolation processor for real-time video applications," IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 1991, pp. 1982–345, Nov. 2014. Presented a low complexity color interpolationalgorithm is proposed for the very large scale integration (VLSI) implementation in real-time applications. This novel algorithm consists of an edge detector, an anisotropic weighting model, and a filter based compensator.

The anisotropic weighting model is designed to catch more information in horizontal than vertical directions. The filter based compensation methodology includes a Laplacian and spatial sharpening filters, which are developed to improve the edge information and reduce the blurring effect. In addition, the hardware cost was successfully reduced by hardware sharing and reconfigurable design techniques. The VLSI architecture of the proposed design achieves 200 MHz with 5.2-K gate counts, and its core area is 64 236 μm^2 synthesized by a 0.18-μm CMOS process. In this design the gate counts and core area are increased and it is reduced in proposed work.

Pekkucuksen and Y. Altunbasak, "Edge strength filter based color filter array interpolation," IEEE Trans. Image Process., vol. 21, no. 1, pp. 393–397, Jan. 2012. This presents that for economic reasons, the most digital cameras use color filter arrays instead of beam splitters to capture image data. As a result of this, only one of the required three color samples becomes available at each pixel location and the other two needs to be interpolated.

This process is called Color Filter Array (CFA) interpolation or demosaicing. Edge strength filter output is utilized both to improve the initial green channel interpolation and to apply the constant color difference rule adaptively. This simple edge directed method yields results with low CPSNR. In order to avoid artifacts such as zipper effect, blur, and color spots are produced.

In this a novel cost-efficient and high-performance hardware oriented color demosaicking algorithm the demosaicking equations for different colors have some similar parts, they can be integrated into a shared VLSI architecture by using the hardware sharing technique. Furthermore, in order to meet the demand of real-time video applications, the pipeline scheduling technique was used to realize this design.

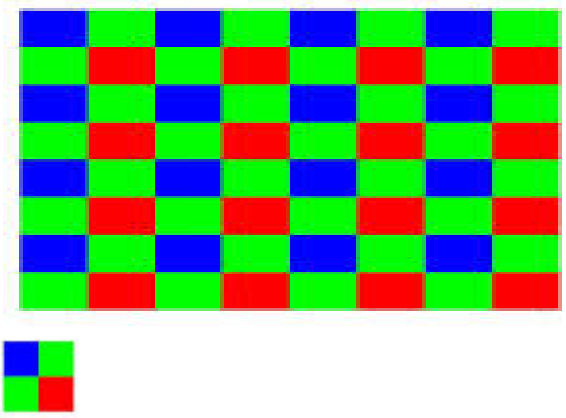


Figure 1 Bayer pattern proposed BGGR

The boundary detector and mirror models were added to detect the boundary information and provide mirror pixels for green and red-blue color interpolations, which can efficiently improve the quality of interpolated pixels located in the boundary. In addition, the linear deviation compensation and immediately interpolated green color pixels G' are used to improve the quality of the interpolated red and blue color pixels.

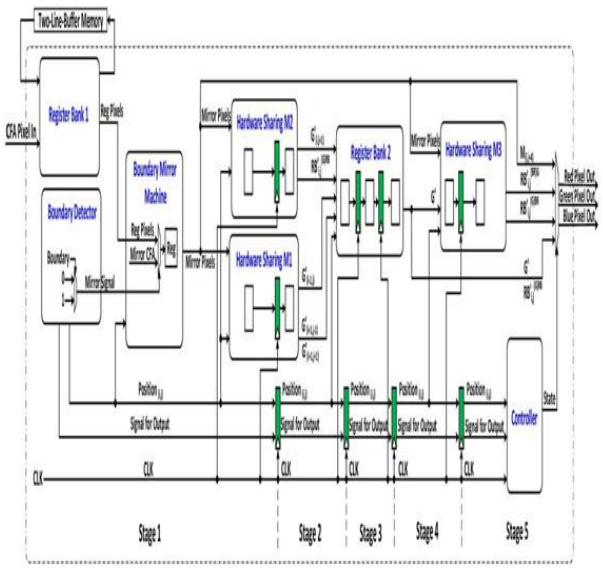


Figure 2 Block diagram of the VLSI architecture for the existing color demosaicking design

Figure.2.3 shows a five-stage pipelined architecture of the proposed color demosaicking algorithm. It consists of register bank 1; a boundary detector; a boundary mirror machine; three hardware sharing M1, M2, and M3; register bank 2; and a controller. All parameters in the proposed algorithm are designed as 2, 4, and 8. Hence, the hardware cost can be greatly reduced by using the shifters rather than dividers and multipliers.

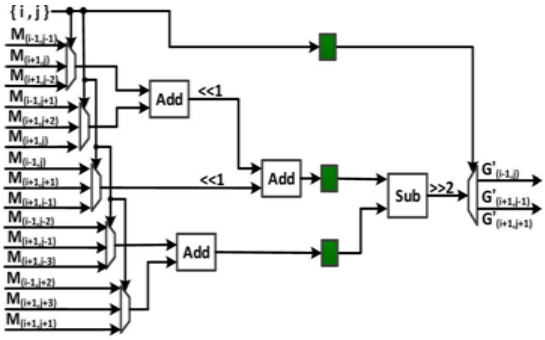


Figure 3 Architecture of hardware sharing M1

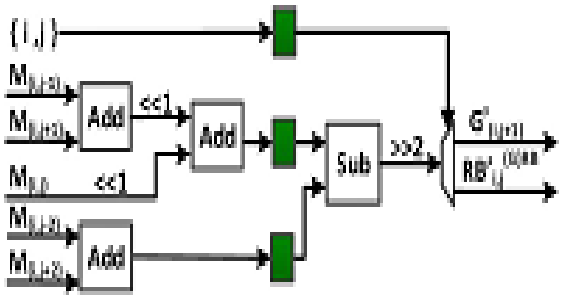


Figure 4 Architecture of hardware sharing M2

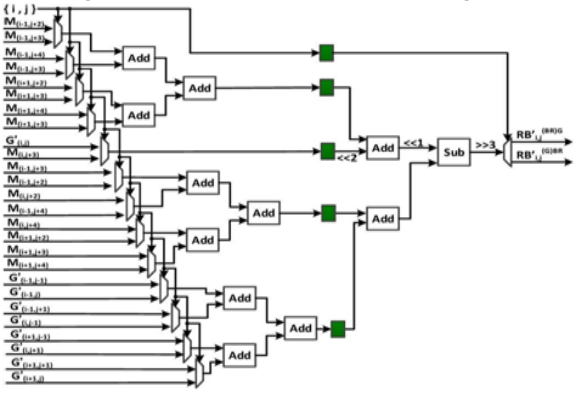


Figure 5 Architecture of hardware sharing M3

To conclude that the existing techniques have drawbacks this has more gate counts and occupies more chip core area. Also the images have low CPSNR value. The proposed technique has been used to avoid artifacts such as zipper effect, blur, and color spots and to improve high CPSNR value and also reduces gate counts and chip area.

III. PROPOSED WORK

The proposed demosaicking algorithm incorporates the residual interpolation into the GBTF algorithm. The GBTF algorithm first interpolates the G pixel values. Then, the R and B pixel values are interpolated by using the color difference interpolation. For the interpolation of the R and B pixel values, we simply replace the color difference interpolation with the proposed residual interpolation as described in the previous section. We use bilinear interpolation for the residual

interpolation. In the following, we propose the interpolation process of the G pixel values based on the GBTF algorithm. The interpolation process of the G pixel values by the GBTF algorithm consists of three steps: (i) The Hamilton and Adams' interpolation formula is applied in the horizontal and vertical directions to estimate the G pixel values at the R and B pixels and the R or B pixel values at the G pixels. As a result, the horizontally and vertically estimated R, G, and B pixel values are generated. (ii) The horizontal and vertical color differences (G-R or G-B) are calculated at each pixel. Then, the horizontal and vertical color differences are smoothed and combined into the final color difference estimate. (iii) The G pixel values at the R and B pixels are interpolated by adding the observed R or B pixel values to the final color difference estimates. The Hamilton and Adams' interpolation formula in the step (i) can be interpreted as a linear color difference interpolation. We replace the linear color difference interpolation with the proposed residual interpolation. To simplify the explanation, in the following, we focus on the estimation of the R pixel values at the G pixels in the horizontal direction. The B pixel values at the G pixels are estimated in the same manner as the R pixel values. And also, we apply the same process in the vertical direction. The Hamilton and Adams' interpolation formula for the R pixel value in the horizontal direction can be expressed as:

$$\hat{R}^H_{i,j} = (R_{i,j-1} + R_{i,j+1})/2 + (2\times G_{i,j} - G_{i,j-2} - G_{i,j+2})/4, \quad -(1)$$

where the suffix i, j represents the target pixel, $\hat{R}^H_{i,j}$ is the horizontally estimated R pixel value at the G pixel. This interpolation formula can be interpreted as the linear color difference interpolation as:

$$\hat{R}^H_{i,j} - G_{i,j} = (R_{i,j-1} - \check{G}^H_{i,j-1})/2 + (R_{i,j+1} - \check{G}^H_{i,j+1})/2, \quad -(2)$$

where $\check{G}^H$ is the horizontally estimated G pixel value at the R pixel calculated as:

$$\check{G}^H_{i,j-1} = (G_{i,j-2} + G_{i,j})/2, \; \check{G}^H_{i,j+1} = (G_{i,j} + G_{i,j+2})/2. \quad -(3)$$

In the proposed algorithm, we use the tentative estimates instead of these estimated G pixel values. In the step (i) of the GBTF, the interpolation process is performed in one dimensional image at every row and column. First, we apply the linear interpolation to the G pixel values to generate the one dimensional guide image. Then, the guided upsampling is applied to the R pixel values to obtain the tentative estimate of the one dimensional R image. After that, the residuals are calculated and the residual interpolation is performed. Finally, we estimate the R pixel values at the G pixels by adding the tentative estimate to the interpolated one dimensional residual image. The resultant R pixel value can be expressed as:

$$\hat{R}^H_{i,j} = (R_{i,j-1} - \check{R}^H_{i,j-1})/2 + (R_{i,j+1} - \check{R}^H_{i,j+1})/2 + \check{R}^H_{i,j}. \quad -(4)$$

where $\check{R}^H$ is the tentative estimate of the R pixel value. The G pixel values at the R pixels are estimated in the same manner, where the tentative estimate of the one dimensional G image is generated. After the above step (i), the step (ii) and (iii) of the GBTF algorithm are applied. Although the proposed residual interpolation also can be incorporated into the step (ii), we will discuss it in future work since the incorporation of the residual interpolation into the step (ii) is not straightforward. In the GBTF algorithm, a simple averaging filter, f = [11111]/5, is used for smoothing the directional color differences in the step (ii). In contrast, we introduce a Gaussian weighted averaging filter instead of the simple averaging filter to improve the performance. We empirically use 1 for the standard deviation of the Gaussian weight.

For the above hardware sharing technique is further improved by reducing adders and subtractors by spurious power suppression technique. The adders and subtractorsis replaced by SPST technique. The SPST is illustrated through a low-power adder/subtractor design example. The adder/subtractor is divided into two parts, i.e., the most significant part (MSP) and the least significant part (LSP). The MSP of the original adder/subtractor is modified to include detection logic circuits, data controlling circuits, denoted as latch-A and latch-B, sign extension circuits, and some glue logics for calculating the carry in and carry out signals.

The 16-bit adder/subtractor is divided into MSP and LSP between the eighth and the ninth bits. Latches implemented by simple AND gates are used to control the input data of the MSP. When the MSP is necessary, the input data of MSP remain unchanged. However, when the MSP is negligible, the input data of the MSP become zeros to avoid glitching power consumption. The two operands of the MSP enter the detection-logic unit, except the adder/subtractor, so that the detection-logic unit can decide whether to turn off the MSP or not.

The detection-logic unit of SPST determines whether the input data of MSP should be latched or not. For the cases like A=11111111 and B=11111111,A=11111111 and B=00000000, A=00000000 and B=11111111, A=00000000 and B=00000000 the detection logic produces close='0' which avoids computation. For all other normal operation close='1' which indicates the normal function.

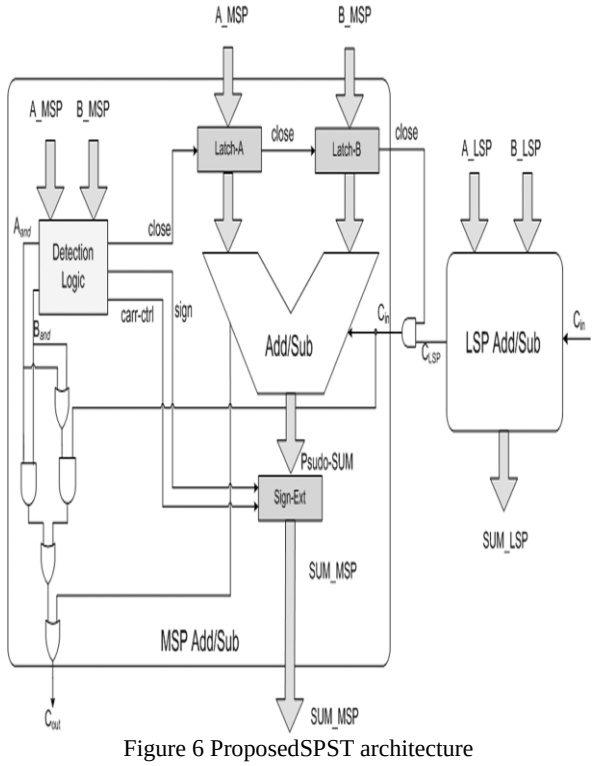


Figure 6 ProposedSPST architecture

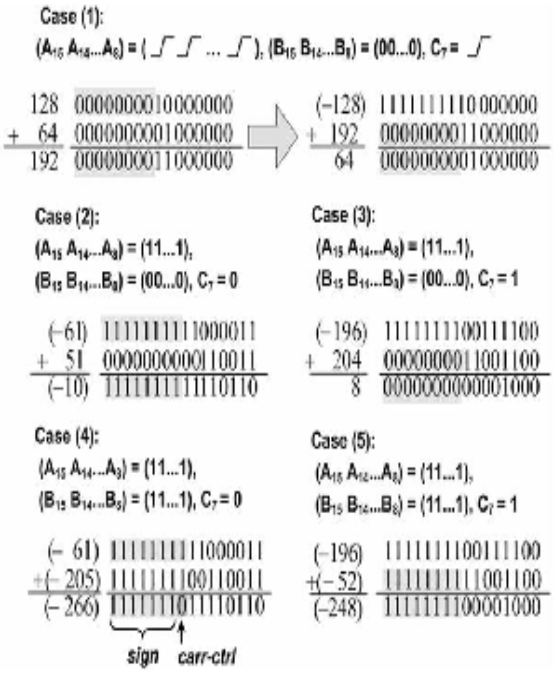


Figure 7 Spurious transitions

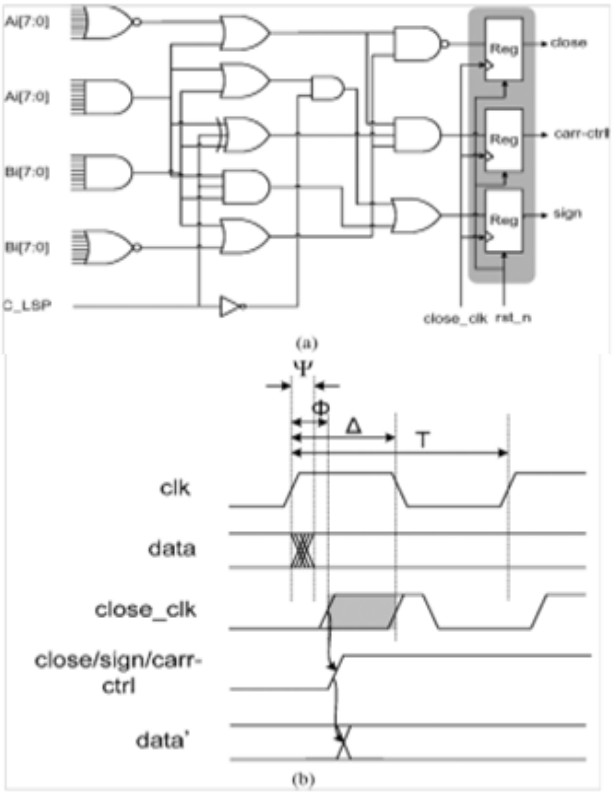


Figure 8 (a) Detection-logic unit and (b) its timing diagram

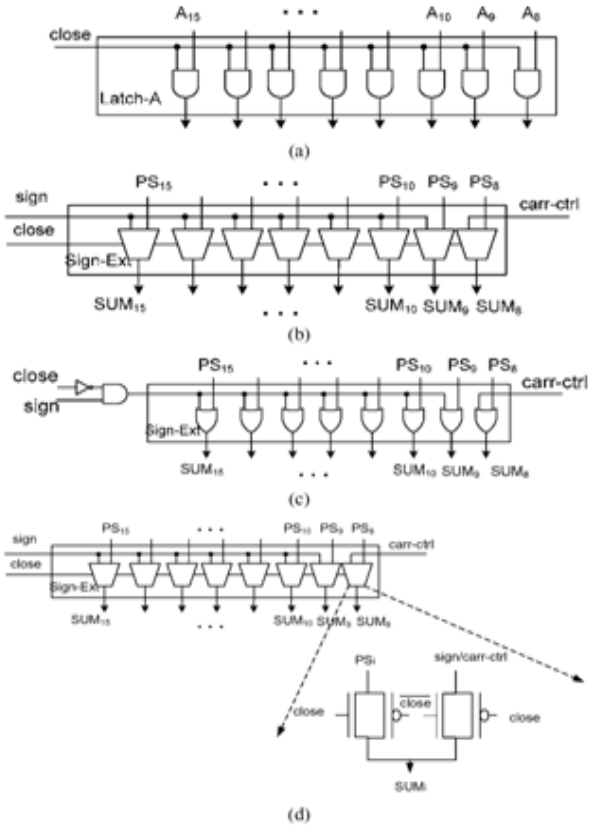


Figure 9 Data-controlling components of the SPST

IV. SIMULATION RESULTS AND COMPARISONS

4.1Existing Results

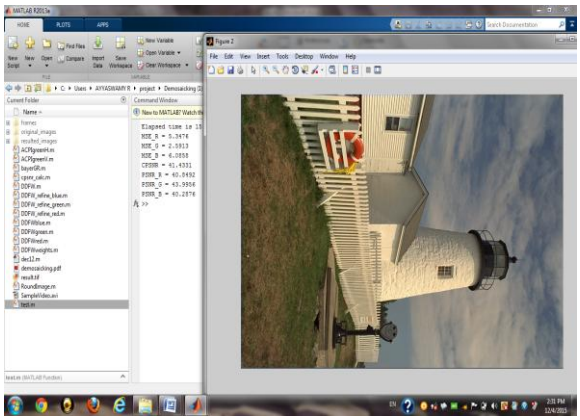


Figure 10 Simulated results for Existing work using MATLAB

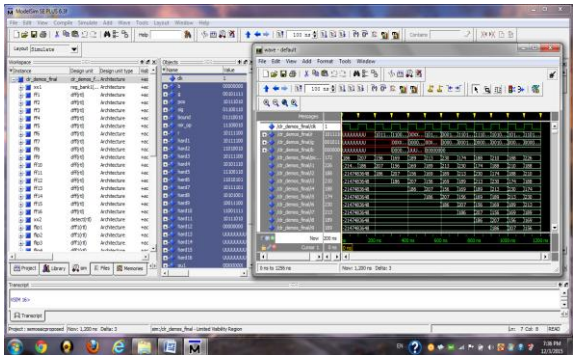
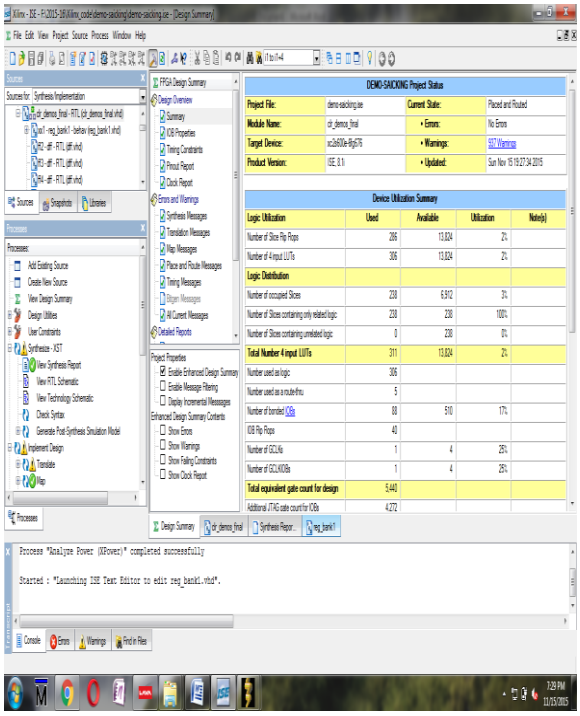


Figure 11 Simulated results for Existing work using Modelsim



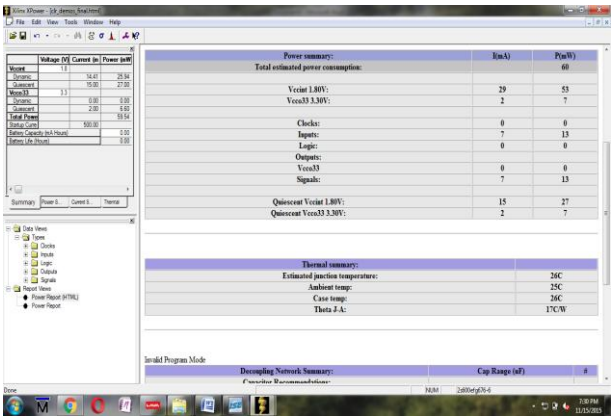


Figure 12 Simulated results for Existing technique using Xilinx

4.2 Proposed Results

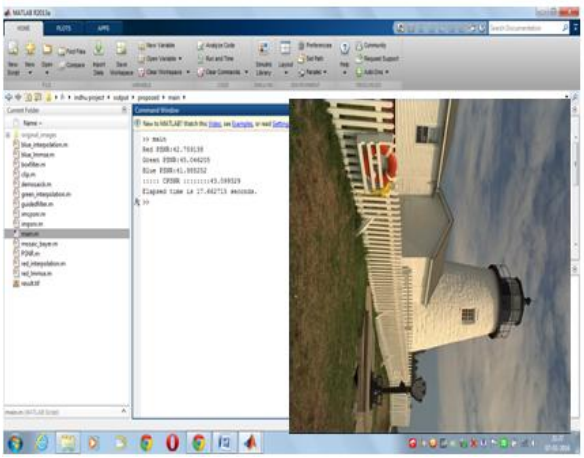


Figure 13 Simulated results for proposed work using MATLAB

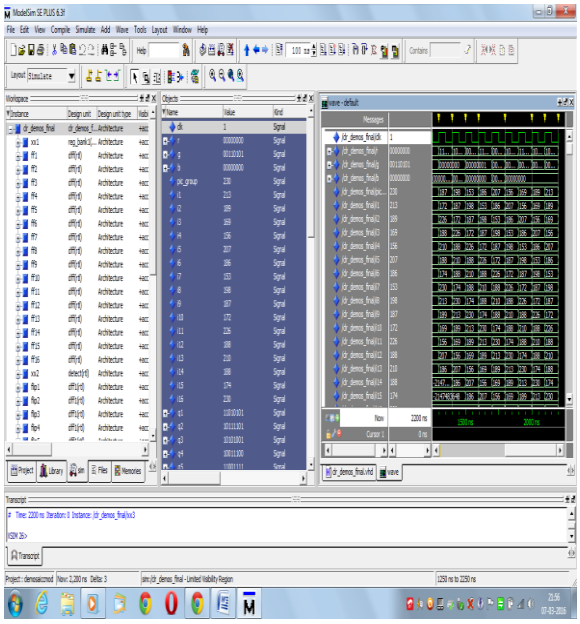
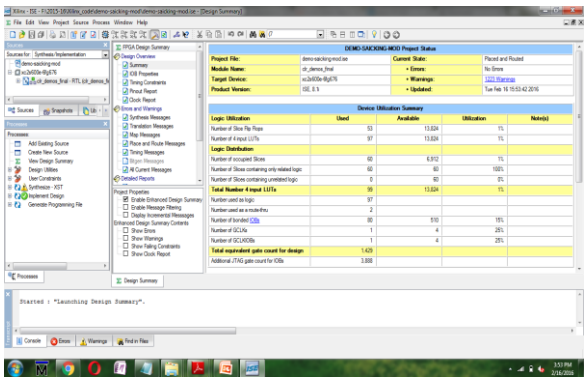


Figure 14 Simulated results for proposed work using Modelsim



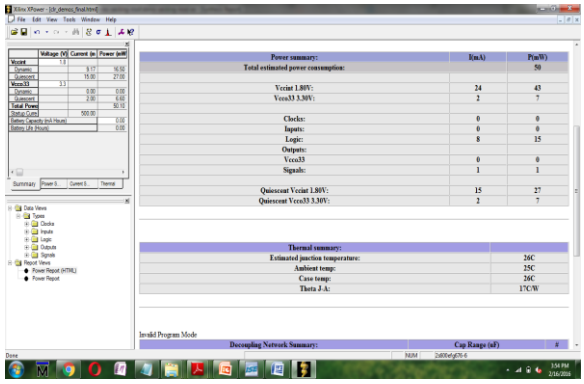


Figure 15 Simulated results for proposed technique usingXilinx

4.3 Comparison Table

Parameters	Existing	Proposed
CPSNR	41.4331	43.0993
PSNR_R	40.8492	42.7591
PSNR_G	43.9956	45.0462
PSNR_B	40.2876	41.9882

Table 4.5 Comparisons of average image quality with existing algorithms and proposed work

Table 4.1 lists the comparison of the four formula values and

computing time for the previous designs with this work. The average CPSNR value in this work is 41.4 dB, which is better than 30.71, 32.21, 32.8, and 33.377 dB in the previous designs. Compared with the previous designs, this work improved the average values of PSNR, CPSNR respectively.

Table 4.5 lists the comparisons of computing resource for the previous designs with this work. The computing resource in this work contains only 17 additions, 3 subtractions, and 9 shifts.

	Existing Method	**Proposed Work**
CPSNR	41.4331	43.0993
Gate Counts	5440	1429
Frequency	13.417ns	12.135ns
Power	60mW	50mW

Table 4.6 Comparisons of Computing Resource for Previous Designs with proposed work

By using the hardware sharing technique, 46.8% adders, 50% subtractors, and 50% shifters were successfully reduced in this work. This work can be realized by adders, subtractors, and shifters only without any dividers and multipliers. The silicon area of a divider or a multiplier is much greater than that of an adder, a subtractor, and a shifter. Moreover, compared with a previous design, this work achieved reduction of 43.3% adders, 70% subtractors, and 40% shifters and without using any absoluter.

This work was implemented by using a hardware description language and Modelsim tool to provide best results since it easily connects matlab to modelsim. By using the Xilinx tool the total gate counts used in this work is only 5440 gate counts and its power consumption is only 60mW. When this work is implemented in an EDA tool of Design Vision (Synopsys) with the library of TSMC 0.18-μm CMOS process the NAND equivalent gate count in this work is only 4.97 K, and its power consumption is 4.76 mW when operates at 200 MHz. The core area is 60229 μm^2, in which the width, length are 243.85 and 246.99-μm, respectively.

Moreover, this work also saved over 50.3%, 80.8%, 11.2%, and 4.4% gate counts than the previous designs LHCI, CDSP, ACDS, and EECP, respectively. The memory requirement in this work is only a two-line-buffer memory device; it is much less than 1 frame. Compared with the previous low-complexity designs, this work not only improved the quality of the interpolated images but also reduced the hardware cost and memory requirement. It provided an efficient color demosaicking VLSI design for real-time video applications. Thus the simulated results and comparison with previous techniques have been demonstrated as mentioned above.

CONCLUSION

The proposed work concludes that a cost efficient and high performance color demosaicking VLSI design based on hardware sharing and pipeline scheduling techniques has been proposed for real time video applications. It uses the residual interpolation method to improve the quality of the reconstructed image. Also it reduces the hardware cost and power by using SPST technique.

References

1. B. E. Bayer, “Color imaging array,” U.S. Patent 3 971 065, Jul. 1976.
2. H. Chen and Y. Cheng, “VLSI implementation of color interpolation in color difference spaces,” in Proc. IEEE ISCAS, May 2012, pp. 1680–1683.
3. H. A. Chang and H. H. Chen, “Stochastic color interpolation for digital cameras,” IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 8, pp. 964–973, Aug. 2007.
4. Kuan-Hung Chen, “A Spurious-Power Suppression Technique for Multimedia/DSP Applications”*Member, IEEE*, and Yuan-Sun Chu, *Member, IEEE*
5. Pekkucuksen and Y. Altunbasak, “Edge strength filter based color filter array interpolation,” IEEE Trans. Image Process., vol. 21, no. 1, pp. 393–397, Jan. 2012.
6. S. C. Hsia, M. H. Chen, and P. S. Tsai, “VLSI implementation of low power high-quality color interpolation processor for CCD camera,” IEEE Trans. VLSI Syst., vol. 14, no. 4, pp. 361–369, Apr. 2006.
7. S. L. Chen and E. D. Ma, “VLSI implementation of an adaptive edge-enhanced color interpolation processor for real-time video applications,” IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 1991, pp. 1982–345, Nov. 2014.
8. S. C. Hsia and P. S. Tsai, “VLSI implementation of camera digital signal processor for document projection system,” in Proc. IEEE ICSPS, Jul. 2010, pp. 657–660.
9. Y. H. Shiau, P. Y. Chen, and C. W. Chang, “An area-efficient color demosaicking scheme for VLSI architecture,” Int. J. Innov. Comput., Inf. Control, vol. 7, no. 4, pp. 1739–1752, Apr. 2011.
10. Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi “Residual Interpolation for Color Image Demosaicking”
11. Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, “Image quality assessment:From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.