A Low-Power Efficient Demosaicing Hardware Design : Project Progress Report

SAGNIK BASU sbasu24@wisc.edu
JAGDISH MOHAPATRA nmohapatra@wisc.edu
LAMSHE RAJA lraja@wisc.edu
RAHIMULLAH SHAIK rshaik@wisc.edu

***Abstract*— Digital cameras use Color Filter Array (CFA) where the sensor detects only a particular spectral band for each pixel. The unmeasured spectral bands are estimated from the neighborhood pixels. Hence to reconstruct a full representation of the color image, an interpolation of the unmeasured spectral bands is required.This is known as demosaicing Though there are many different demosaicing techniques,there is always a trade-off between dynamic power consumed during computation and quality of the reconstructed image.The proposed scheme is based on the fact that edges have strong luminance components. Hence, we are proposing an gradient-corrected technique by taking the intensity values of critical pixels into consideration.The proposed techniques give a better performance in terms of low dynamic power consumption and improved PSNR when compared to some of the available demosaicing techniques such as Nearest Neighbor interpolation, vector median interpolation, and bilinear interpolation.**

I. INTRODUCTION

Image Signal Processors are integral part of all modern smartphones. From [1],it can be summarized that it consists of several blocks to convert the Bayer pattern from CMOS Sensor to a visually perceptive RGB pattern. With more powerful processors which are available to us, the ISP pipelines are becoming more computationally heavy to support more pixel resolution. We propose a novel design for one of the most important blocks of the ISP – which is Image Demosaicing [2].

Our project considers the ISP to be an important part of embedded systems like digital cameras. ISP is necessary for the digital camera to have an efficient imaging pipeline. More research is being done on efficient ISP designs on modern processors [3].The RGB output pixels computed from the demosaicing from them are an important for computer vision algorithms being run on embedded system. Since the output from demosaicing blocks are being fed to noise reduction blocks , we target generating a more accurate demosaicing hardware in order to reduce dynamic power consumption and improve PSNR.By saving dynamic power on the demosaicing block, we are making ISP pipeline more efficient than before.

II. TASK BREAKDOWN AND PROGRESS

A. Hardware

The crucial task from hardware perspective is coming up with novel demosaicing hardware enginer architecture, implement it using verilog HDL.

We are planning to feed sufficient amount of bayer pixel data to the DUT, and compare the RGB output from DUT with the desired software computed RGB data to check our correctness in the design.

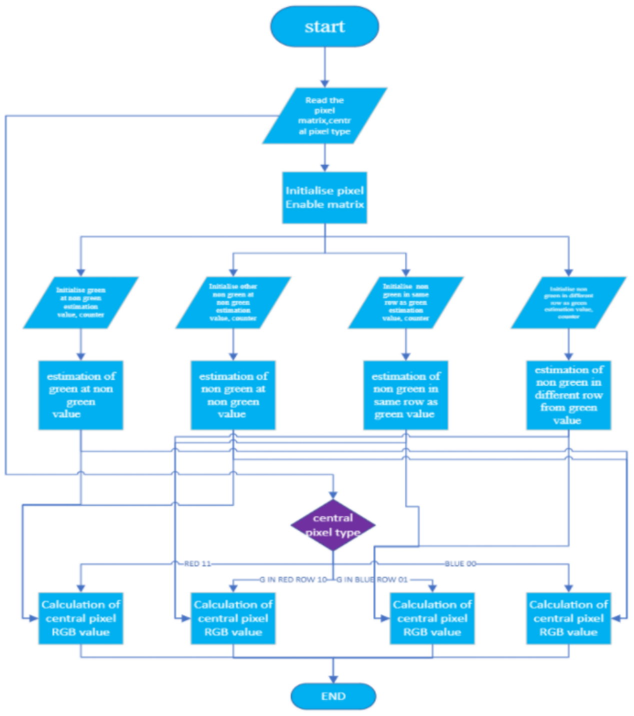
start
Read the pixel matrix,central pixel type
Initialise pixel Enable matrix
Initialise green at non green estimation value, counter
Initialise other non green at non green estimation value, counter
Initialise non green in same row as green estimation value, counter
Initialise non green in different row as green estimation value, counter
estimation of green at non green value
estimation of non green at non green value
estimation of non green in same row as green value
estimation of non green in different row from green value
central pixel type
RED 11
BLUE 00
G IN RED ROW 10
G IN BLUE ROW 01
Calculation of central pixel RGB value
Calculation of central pixel RGB value
Calculation of central pixel RGB value
Calculation of central pixel RGB value
END

Fig. 1. Hardware block diagram

For the calculation of RGB values for central pixel which may be a red, blue, green in red row, green in blue row pixel, we take the pixel matrix and initialize the pixel enable matrix which is used to select the row or column. Then estimation of green pixel value at non green pixel, Estimation of non-green at non green pixel, estimation of non-green in same row as green value, estimation of non-green in different row as green values are carried out as per the proposed algorithm. Hence, we get red, green, blue values for the center pixel from raw data. This ensures that our algorithm is correct and the system uses minimum memory transactions, thus reducing the dynamic power consumption.

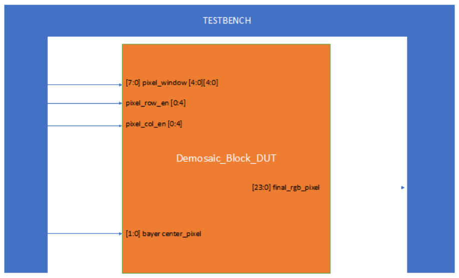

Fig. 2. Hardware Testbench

B. Testbench

We have finished designing with exhaustive test-cases to measure all the feature of the top hardware module before it goes to FPGA and test the code coverage for the RTL. The input pin description of our testbench :

- pixel window : a 5x5 window with each 8bit-width bayer pixel data
- pixel row en : a Single bit signal with depth 5. Used to select the particular row to start computation
- pixel col en : a Single bit signal with depth 5. Used to select the particular column to start computation
- bayer center pixel : a 2-bit depth signal to check the center pixel type

The output ports of our testbench :

- final rgb pixel : à 24-bit pixel data with R,G,B components each contributes 8bit

C. Software

From the software perspective , the first task was to complete generating a set of bayer test vectors and golden reference for the demosaicing block. Since raw bayer format are dependent on the CMOS camera vendor, it is crucial to use a widely know dataset in order to reproduce the correct results. We found the Microsoft demosaicing dataset [4] to be the perfect candidate for our purpose. We are using the OpenCV cvtColor() function for generating the ground truth RGB image from the bayer image. Fig. 3 and 4 shows a Bayer image and the converted RGB image using software demosaicing.

The next planned task is to generate the gain coefficients for interpolation the G values at R location. Also to evaluate the final results , we will be coming with a PSNR measurement tool.

III. Challenges

The most challenging part in hardware design is to implement the mathematical functions in the demosaicing algorithm. Based on the level of computation we are requiring, we have decided the internal signals which are going to be part of the demosaicing module. Since, a huge level of interpolation and filtering is involved with the demosaicing, we are trying every possible way to optimize the design hardware to not consume a lot of area.

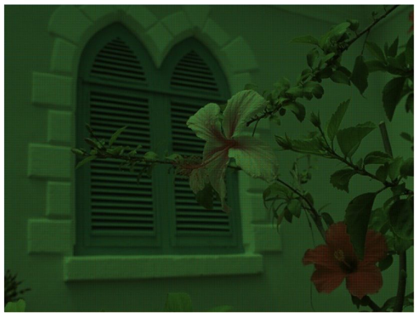

Fig. 3. Bayer image

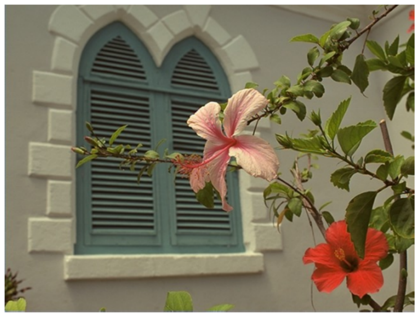

Fig. 4. RGB Image

In the software side, our proposed linear interpolation requires Weiner minimum mean square error interpolation for determining the gain parameters for calculation of the intensity values within the 5x5 pixel window. For this, we will be designing using the algorithm used in [5]. It uses the below equation to find the inter-channel correlation coefficient

$$C_{x,y} = \frac{\sum_{(n_1,n_2)} \left(x(n_1,n_2) - \mu_x\right)\left(y(n_1,n_2) - \mu_y\right)}{\sqrt{\sum_{(n_1,n_2)} \left(x(n_1,n_2) - \mu_x\right)^2}\sqrt{\sum_{(n_1,n_2)} \left(y(n_1,n_2) - \mu_y\right)^2}} \tag{1}$$

Since the original authors in [6] used a simple KODAK data-set to calculate the co-efficient , we expect our calculations to be more accurate than the authors on high resolution images.

References

[1] S.-H. Choi, J. Cho, Y.-M. Tai, and S.-W. Lee, "Implementation of an image signal processor for reconfigurable processors," *2014 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 141–142, 2014.

[2] K. Hirakawa and T. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.

[3] P. Hansen, A. Vilkin, Y. Krustalev, J. Imber, D. Talagala, D. Hanwell, M. Mattina, and P. N. Whatmough, "Isp4ml: The role of image signal processing in efficient deep learning vision systems," *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 2438–2445, jan 2021. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICPR48806.2021.9411985

[4] D. Khashabi, S. Nowozin, J. Jancsary, and A. W. Fitzgibbon, “Joint demosaicing and denoising via learned nonparametric random fields,” *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 4968–4981, 2014.

[5] B. Gunturk, Y. Altunbasak, and R. Mersereau, “Color plane interpolation using alternating projections,” *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 997–1013, 2002.

[6] H. Malvar, L. wei He, and R. Cutler, “High-quality linear interpolation for demosaicing of bayer-patterned color images,” in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, 2004, pp. iii–485.