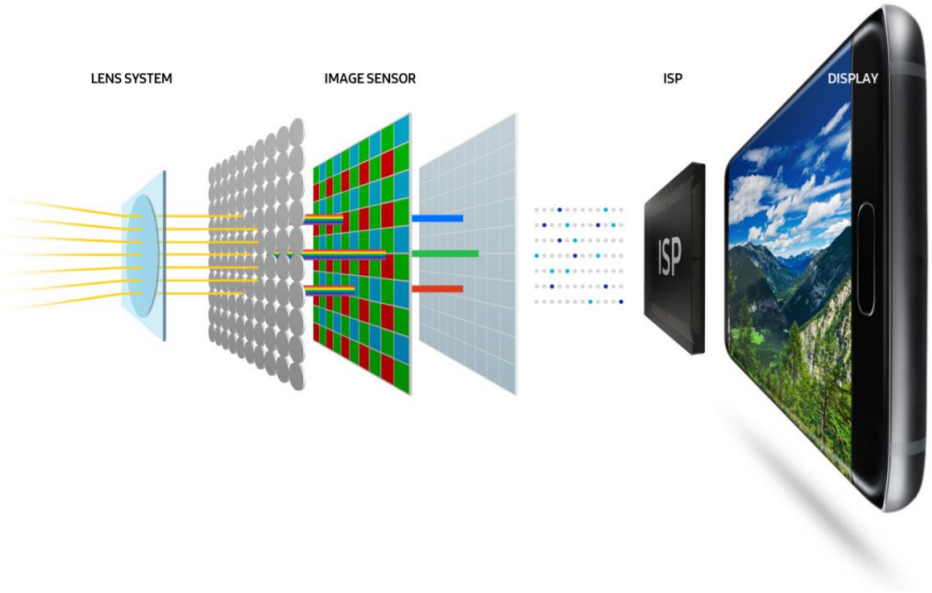
College of Engineering
UNIVERSITY OF WISCONSIN–MADISON

OLYMPUS: High-Quality Linear Interpolation-based Demosaicing Engine

Members : Lamshe Raja, Sagnik Basu, Jagdish Mohapatra, Rahimullah Shaik
Dept of Electrical and Computer Engineering
UNIVERSITY OF WISCONSIN-MADISON

Introduction

Where there is a Camera, there are ISPs

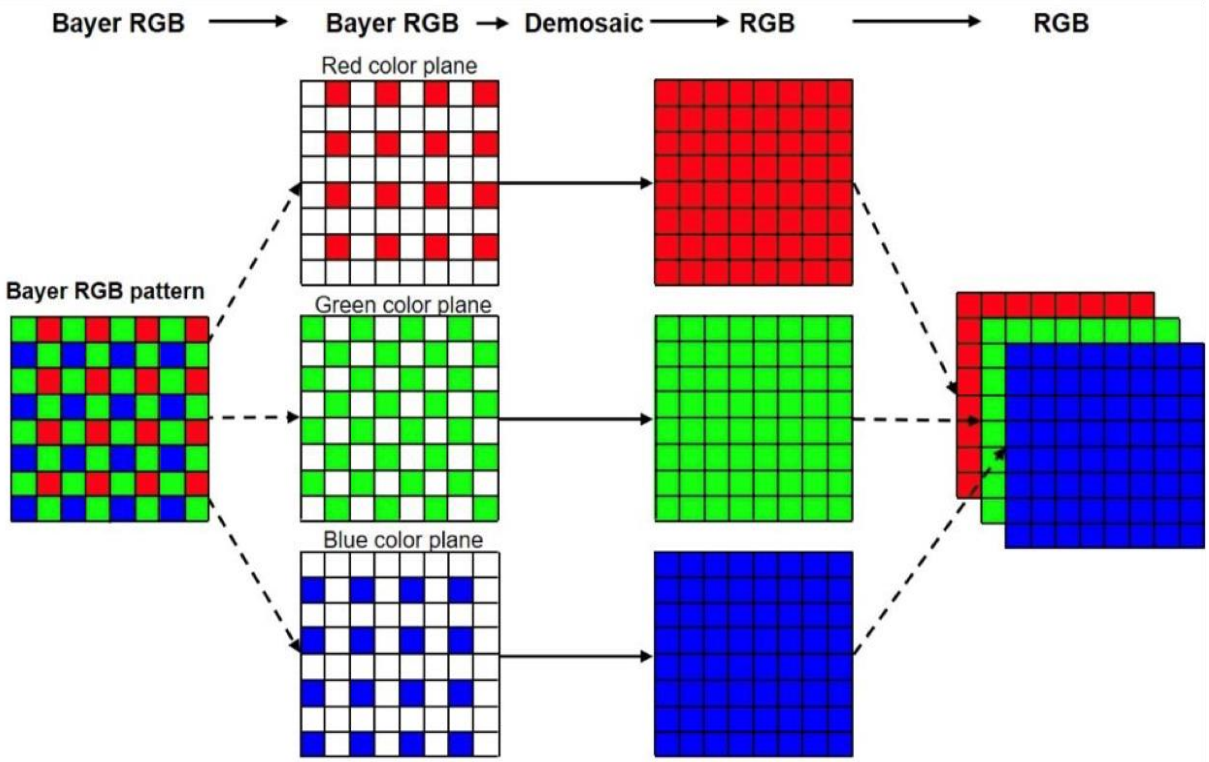


Source : Samsung

Source : Qualcomm

Demosaicing

A digital image process used to reconstruct a full color image from the incomplete color samples output from an image sensor overlaid with a color filter array (CFA).



Source: https://theailearner.com/2018/10/28/bayer-filter/

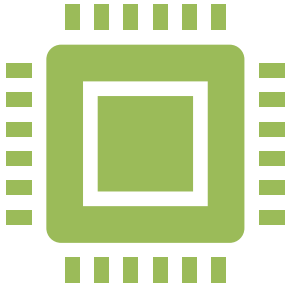
Key Highlights

Enhanced the accuracy of the beginning block in the ISP pipeline (i.e. demosaicing block) compared to the baseline approach Bilinear interpolation.

Reduced the dynamic power of our high-quality linear interpolation based demosaicing hardware engine compared to Bilinear interpolation method.

Design Process

Software



Hardware

Software

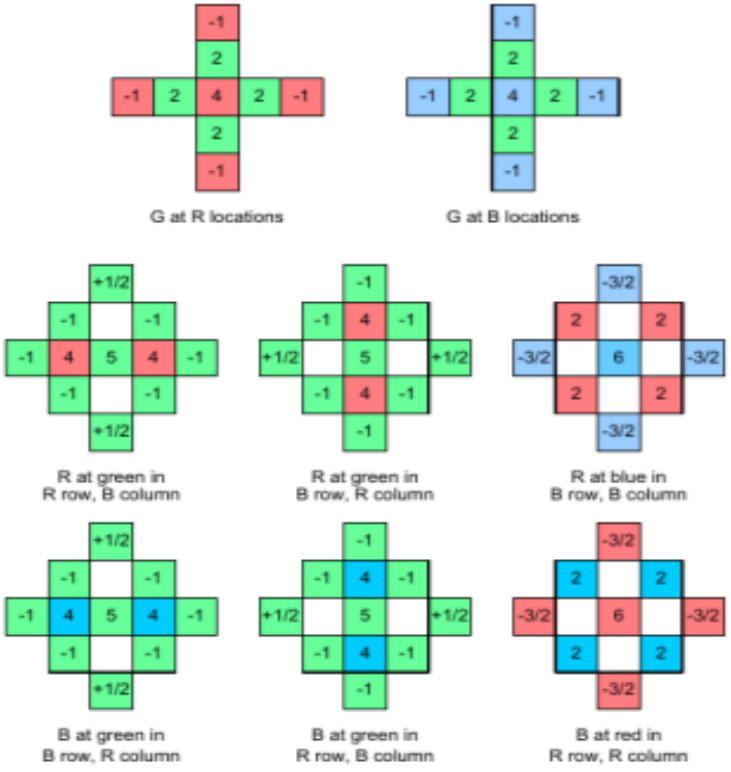
H.S. Malvar et al., “HIGH-QUALITY LINEAR INTERPOLATION FOR DEMOSAICING OF BAYER-PATTERNED COLOR IMAGES “

Estimate gain parameters on MSR Demosaicing dataset.

$\alpha = 1/2$ $\beta = 5/8$ $\gamma = 3/4$

Our Approach

We don't discard the value present at a location like the bilinear interpolation approach, rather, we compare that value to its estimate for a bilinear interpolation for the nearest samples and thus we correct a bilinearly interpolated green/blue/red value by adding a portion of this estimated change.



$$\hat{g}(i,j) = \hat{g}_B(i,j) + \alpha \Delta_R(i,j)$$

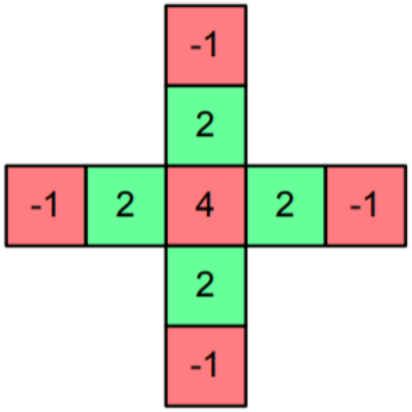
where $\Delta_R(i,j)$ is the gradient of R at that location, computed

$$\Delta_R(i,j) \triangleq r(i,j) - \frac{1}{4} \sum r(i+m, j+n)$$

$$(m,n) = \{(0,-2), (0,2), (-2,0), (2,0)\}$$

Software

- C-Model of interpolation scheme

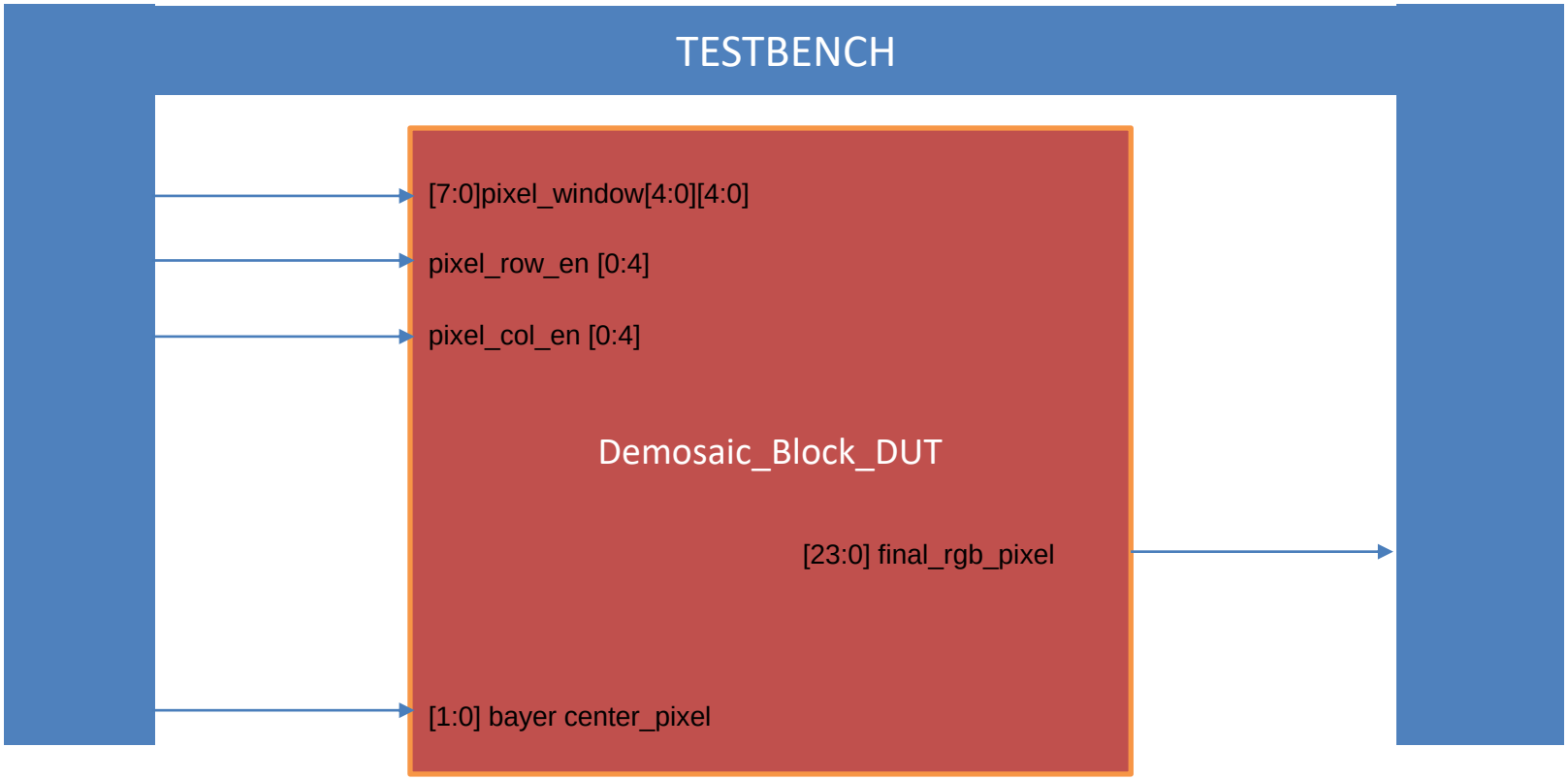



```
if((x & 1) == RedX && (y & 1) == RedY)
{
    /* Center pixel is red */
    OutputRed[i] = Input[i];
    OutputGreen[i] = (2*(Neigh[2][1] + Neigh[1][2]
        + Neigh[3][2] + Neigh[2][3])
        + (NeighPresence[0][2] + NeighPresence[4][2]
        + NeighPresence[2][0] + NeighPresence[2][4])*Neigh[2][2]
        - Neigh[0][2] - Neigh[4][2]
        - Neigh[2][0] - Neigh[2][4])
        / (2*(NeighPresence[2][1] + NeighPresence[1][2]
        + NeighPresence[3][2] + NeighPresence[2][3]));
    OutputBlue[i] = (4*(Neigh[1][1] + Neigh[3][1]
        + Neigh[1][3] + Neigh[3][3]) +
        3*((NeighPresence[0][2] + NeighPresence[4][2]
        + NeighPresence[2][0] + NeighPresence[2][4])*Neigh[2][2]
        - Neigh[0][2] - Neigh[4][2]
        - Neigh[2][0] - Neigh[2][4]))
        / (4*(NeighPresence[1][1] + NeighPresence[3][1]
        + NeighPresence[1][3] + NeighPresence[3][3]));
}
```

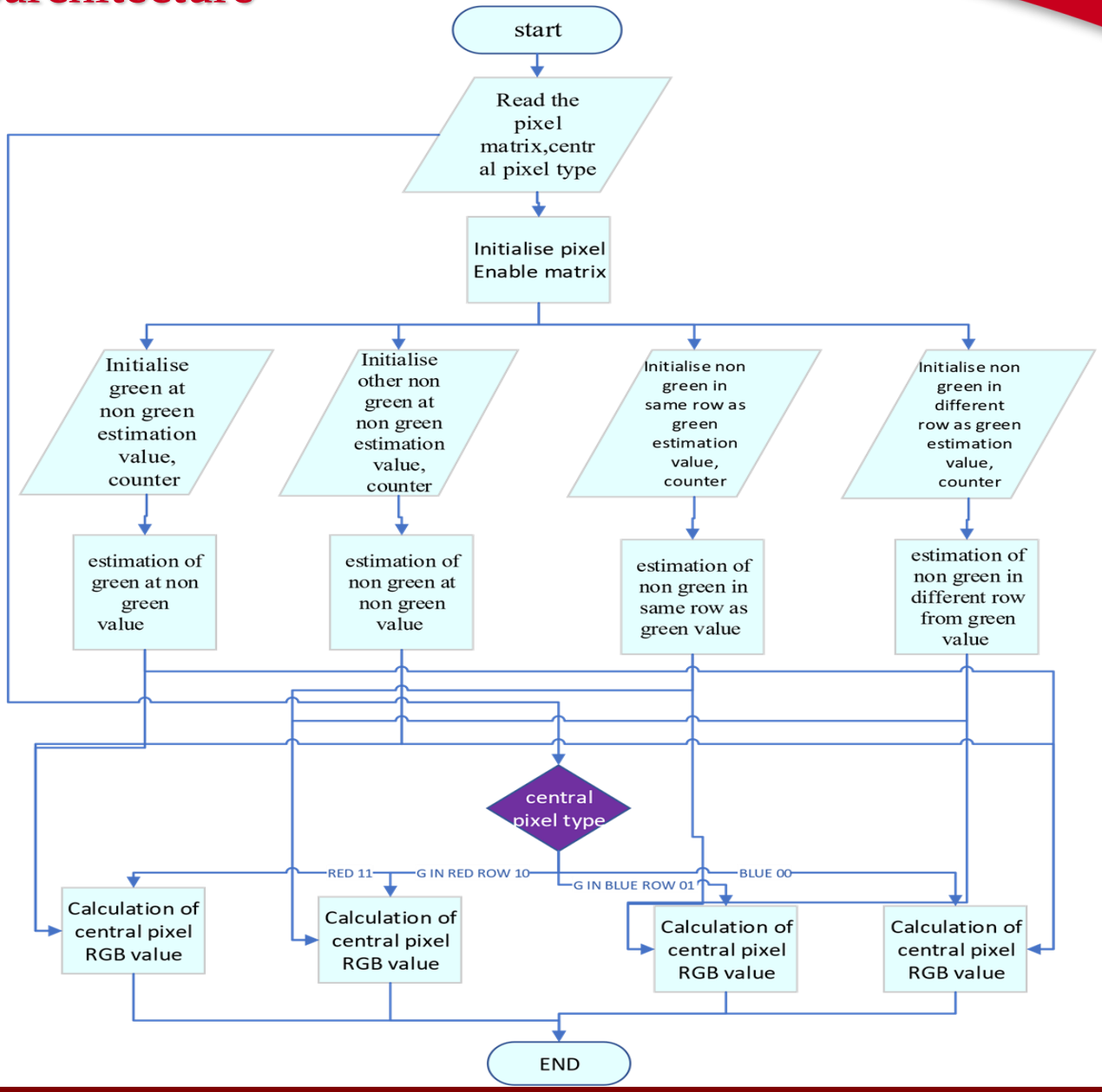
Hardware

- Micro-architected the Demosaicing HW engine.
- Designed using Verilog.
- Simulated using Modelsim.
- Verified the output RGB image with Ground Truth RGB image.

High Level Implementation



Hardware Microarchitecture



Results: Power Consumption

Power analysis was done using Synopsys Design Compiler

	Bilinear (µW)	Olympus: High-Quality Linear (µW)
Dynamic Power	590	302
Leakage Power	2421	2439
Total Power	3011	2741

Total power savings with ref. to bilinear interpolation approach is ~9%.

Results: Area Overhead

	Bilinear (μm^2)	Olympus: High-Quality Linear (μm^2)
No. of cells	1786	1837
Cell Area	5285	5406
Total Area	6408	6542

Area analysis was done using Synopsis Design Compiler

Synopsys 32nm educational cell library was used

PVT parameters set at typical, medium and room temperature

% area increase with ref. to bilinear interpolation method = 2.09%

Results: Area vs Power Tradeoff

With our approach, we are **saving ~ 9% of total power** with a minimal **area increase of 2.09%** when compared to the predecessor bilinear interpolation approach.

Results: PSNR Test

Bilinear	26.27 +\- 0.606
Our approach	29.81 +\- 0.558

Conclusion

We are saving total power by significant amount with small increase in area which will help the ISP pipeline design since every image gets through the Demosaicing block.

We improved the accuracy of RGB pixels with reference to the baseline.

There is a little perfomance penalty involved with our approach which can be avoided with proper computation pipeline.

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 demosaic- ing 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 inter- polation 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.

Bonus Slides

Results : Image Reference

Raw Bayer Image

RGB Image by MATLAB

Actual Camera Pipeline