Design and Analysis of Imaging Chip Using High-Speed AXI-Interface for MPSOC Applications on FPGA Platform

Archana H R (✉ hraarchana@gmail.com)

BMS College of Engineering

C R Byra Reddy

Bangalore Institute of Technology Department of Electronics and Communication Engineering

Research Article

Keywords: Image Enhancement, AXI Protocol, System on Chip (SoC), MPSoC, FPGA, Adaptive median filter, Memory, interface protocol.

Posted Date: March 15th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-259373/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Design and Analysis of Imaging Chip Using High-Speed AXI-Interface for MPSOC Applications on FPGA Platform

Archana H R*, C R Byra Reddy**

*Department of ECE, BMSCE, Bangalore, India.

Email: hraarchana@gmail.com

** Department of ECE, BIT, Bangalore, India

© Springer Science+Business Media New York 2021

Abstract. The recent innovations in real-time video and image enhancements are allowing much advancement in a wide range of diverse applications. These innovations and advancements provide a new hardware architecture which aims to improve the image visualization, processing speed, and complexity reduction in hardware. The imaging chip concept is introduced in this article to support the Multiprocessing system on chip (MPSoC) applications in real-time scenarios on a single chip. The imaging chip model is designed using high-speed interface protocol, which includes different image enhancement algorithms acts as a master model, Advanced Extensible Interface (AXI)-4 as an interface model, and dual-port memory as a slave model. The image enhancement algorithm includes mainly, Brightness control, contrast stretching, Adaptive median filtering (AMF), Edge-detection techniques, image Thresholding, and Image Histogram method. The AXI-4 provides a high-speed interface for communicating master and slave modules. The proposed model work based on the modes of the operation to process the enhanced image output in MPSoC. The design supports multiple masters and multiple slave modules with reconfigurable nature. The imaging chip is a module on the Xilinx ISE environment and implemented on Artix-7 FPGA, along with the performance metrics like chip Area, time, power, and memory utilization are analyzed with improvements. The model offers low latency and high throughput architecture for real-time Multimedia applications.

Keywords. Image Enhancement, AXI Protocol, System on Chip (SoC), MPSoC, FPGA, Adaptive median filter, Memory, interface protocol.

1 Introduction

The real-time image and video enhancements have become essential research domains due to the rapid development of high-resolution quality images in diverse applications like multimedia, computer vision, bio-medical, consumer electronics, robotics, surveillance, and intelligent transportation. High-quality images provide better performance in image processing and analysis. The different types of resolution enhancements like high, low, and super resolutions are used in several applications for high

definition (HD) display competencies. The image quality assessment (IQA) is essential for image quality checking and performance in image processing applications [1-2]. The image enhancement techniques are used in biomedical applications, especially biometric finger-print image recognition. The biometric quality is assured by different image enhancement methods includes histogram equalization, energy normalization, Filtering approaches, Frequency domain modeling, contextual approaches, and composition modeling [3]. The video and image enhancement algorithms are processed with suitable hardware for real-time applications. The field-programmable gate array (FPGA) plays an essential key role by providing reprogramability, scalability, configurability, and concurrency to meet the hardware constraints like throughput, chip area, and power. FPGA has soft-core processors which provide parallelism to achieve the computational speed and throughput [4-6]. The real-time image processing system is the combination of DSP and FPGA architecture to pre-process the camera inputs with high-speed parallel processing capability. In general, the real-time image processing system on FPGA architecture requires Charged Coupled Device (CCD) camera, analog to digital Chip, Video decoding and memory module, Image enhancement modules, configuration interface modules, control module for image data interface with DSP or other Device interface module [7].

The performance enhancement of real-time images requires a memory controller to store the bitstreams, which are processed in MicroBlaze-embedded processor via AXI4 interconnect [8] and contextual image processing operations are achieved by using Zynq-Ultra scale+ MPSoC FPGA devices with built-in AXI-4 stream for real-time 4k-video streaming[9]. The MPSoC supports on-chip communication and high-bandwidth for data processing with increased parallelism. The MPSoC is used many applications on a single FPGA chip which includes artificial vision [10], fibre optics [11], Video processing [12] etc. The traditional buses use only single master and single slave communication at a time in MPSoC, which affects the performance of the complete system. The AXI bus protocol provides high-speed communication between multiple master and various slaves at the time, which improves the system performance. The AXI interface protocols are used in the Network on-chip and Gateway system to improve the scalability of MPSoC [13-14].

This manuscript presents the different image enhancement algorithms as an imaging chip using a high-speed AXI-interface protocol for MPSOC applications on the FPGA platform. The present work offers high speed enhanced image transfer to the real world using low-cost FPGA in less time. The imaging chip design also supports high-quality video frames to process and enhance real-time scenarios on the hardware platform. Section 2 explains the different image enhancement algorithms on software and hardware approaches for different applications and also highlighted the research problem. Section 3 explains the foundations of the proposed work. Section 4 describes the proposed imaging chip framework in detail with architectures. The Results and performance analysis are elaborated in section 5. Finally, and section 6 concludes the overall work with improvements and future scope.

2 Related Work

This section elaborates on the detailed review of different image enhancement algorithms on hardware and software approaches with finding and also discussion of hardware approaches towards MPSoC using image processing applications. Dutta et al. [15] present a smart frame design using a machine learning approach for training and testing the demographic image dataset, which resides in reconfigurable MPSoC architecture. This method improves the execution time and throughput on the FPGA platform. The FPGA based MPSoC integrates with service-oriented architecture on a single chip to offer solutions to programmable interfaces, software chains, and a different module for diverse applications by Wang et al. [16]. These integrated module offers modularity, flexibility, and reconfigurable feature, but lags with scalable issues on the hardware platform. Karim et al. [17] present MPSoC based module for ECG applications on FPGA, which includes master and slave processor module for processing the software-based ECG data and displays the ECG data on FPGA's input-output switches.

The Huang et al. [18] explains the reconfigurable hardware architecture for image contrast enhancement algorithm, which includes statistical computation, Weighting Probability density, and cumulative distribution functions followed by Gamma correction and luminance transformation. The algorithm offered better image quality and failed to improve computational complexity in hardware. The fast algorithm for image enhancement using parallel architecture is designed by Singh et al. [19] with a polynomial based fractional order filter function, and it results in the better-enhanced image for a different order. It is not suitable for hardware-based modeling because of design complexity. The Pugazhenthir et al. [20] presents the automatic multi-histogram equalization as an image enhancement technique for satellite images using Matlab programming, which prevents the brightness and improve the contrast. The image quality results are not so appropriate and not suitable for hardware-based approaches. Mahajan et al. [21] present the image contrast enhancement using a Gaussian mixture model (GMM) and genetic algorithm. The GMM includes modeling, expectation-maximization method, partitioning, and mapping, which is applied for low contrast images to achieve better quality images. The image enhancement techniques for denoising and contrast enlargement on low light images by Li et al. [22] and for surface roughness detection system by Tian et al. [23] are discussed in detail with performance metrics improvements.

The logarithmic image processing models are designed by Zhao et al. [24] for medical image enhancement, which provides the solution to linear enhancement models using the Unsharp masking algorithm. The logarithmic image processing (LIP), generalized, and parameterized -LIP models are analyzed for different images to improve the enhancing edges and reduce noise sensitivity. Khoukhi et al. [25] presents the combination of Matlab -based and HDL modeling for image contrast and reshaping. The work analyzes the performance metrics and execution time, but this type of modeling is not suitable for large multimedia applications. The Fuzzy rule-based image contrast enhancement model is designed by Liviu et al. [26] for real-time application on a hardware platform that offers computational complexity reduction for low contrast images with an illustration of heavy traffic conditions and not suitable for low-cost FPGA implementations. Medical image enhancement using a hardware-based approach is presented by Chiuchisan [27] and which offers edge detection, contrast enhancement,

brightness- adjustment, and sharpen operation on a single module. The real-time reconfiguration is complicated with this approach.

Aranda et al. [28] describe the median filter modeling with error detection for image processing applications. This model offers error detection capabilities during the injection of fault in the system. The pixel-level and image-level average error detection reports are analyzed with redundancy based approaches with noise ratio improvements. Kumar et al. [29] present the hardware modeling of the median filter with a window size of 5X5, and it offers low latency execution compared to existing approaches. This model mainly includes line buffers for sliding window creation and a sorter model for median filter creation. It is designed using a system generator with a co-simulator approach for image visualization. Cadenas et al. [30] present a pipelined median approach for noise reduction using the slice encoding technique. This method is simple and not suitable for multimedia applications. Tidala et al. [31] present the AXI-protocol interface for Network on Chip design on FPGA Platform, which offers on-chip communication quickly for multiple master and slave network and also resolves the real-time routing issues.

Research Gaps: The next level of research work focus will be an electronic generated digital image can be improved regarding its superiority by enhancing it. Numerous mechanisms have been presented in the field of image enhancement with Field Programmable Gate Array (FPGA). Incorporation of Advanced Microcontroller Bus Architecture (AMBA) based Advanced Extensible Bus (AXI) in the mechanism would provide a better interface for communication between the memory and medical image given as an input for enhancement reducing the effect of heterogeneity on MPSoC. The significant research problems of interface interconnection are as follows: i) Most of the existing interface modules includes bus-based architectures, shared bus connection, which are facing scalability and reliability problems. ii) Scarce works towards with AMBA based protocols like AHB, APB, and ASB. Even with AXI-4 protocol, very less work carried with a lot of constraints issues and iii) Complete AXI-4 interface protocol with high-speed architecture is yet to come with optimized constraints. The research problems for real-time image processing are addressed as follows: i) Most of the current research works of image processing algorithms and interface protocols are designed, individual. Could not process and retrieve the image at high speed. ii) Most of the interface protocols for image processing applications are available in the form of application-specific IP-cores, we can't reuse the IP-cores for other image processing applications. iii) The existing systems are failed to provide the standalone hardware architecture for Image processing applications on a single substrate.

3 Foundation

This section describes the theoretical foundation of the proposed Imaging chip model which mainly includes Imaging enhancements algorithms and AXI-Interface module.

3.1 Imaging enhancements algorithms

From the past decade, the growth and demand of low power, low-cost digital image systems are captured from Digital camera, PC camera, and video cam recorder. The CMOS Camera or CMOS Image sensor is used to capture the images, while it provides low-quality images. The enhancement algorithms are crucial in the image processing system, which gives the final definition of an image. The enhancement techniques are used to improve the visual appearance of the image in digital signal processing. The edges in the image are necessary, which gives the exact information for human eye perception. The aim is to the detection of pixel points in an image in which image brightness changes sharply. It is difficult to process the broad data for edge detection, which affects to speed of the operation. The location of intricate edges in standard edge detection is not accurate and perfect. So, Sobel enhancement operation is used to solve this locating intricate edges issue, but in software, it is challenging to meet Real-time requirements. Processing the high-speed images in real-time is a challenging task, which needs the high-speed interface Protocol and which interconnects to the external world. The proposed work considered few of the image enhancements algorithms for the Imaging Chip model and are explained below.

A. Brightness control operation

Brightness control provides brightness by adding or subtracting the constant value 'C' to the input image of each pixel ' $IP(i)$ ' and store the results in $OP(i)$ after brightness control operation. The 'i' represents the input or output image pixel values. The equation (1) shows the brightness control operation as follows,

$$OP(i) = IP(i) + C; \text{if } (IP(i) + C) \leq 255 \\ 255; \text{If } (IP(i) + C) > 255 \quad (1)$$

The constant vales appeared within the range (0 to 255), and if the output pixel values exceed more than 255, the information will be loss of the object image. The brightness control supports dark or low light mode images for processing by adjusting the constant values.

B. Contrast stretching operation

Contrast Stretching is used to strengthen the intensity range of the object image. Before the normalization process, initialize the lower and upper-intensity values 'a' and 'b' respectively. The stretching operation generates the lowest 'c' and highest 'd' pixel values of the current image after the normalization process. The equation (2) shows the contrast stretching operation ' $OP(i)$ ' as follows,

$$OP(i) = (IP(i) - c) * \frac{(b - a)}{(d - c)} + a \quad (2)$$

C. Negative-Image transformation

The Negative-Image transformation operation works based on the gray value in the range of (0, T-1). While increasing the intensity of the input image and decrease the intensity of the output image to generate the negative of the original image from black

to white and vice versa. The transformation of the negative image is formed using equation (3) is as follows,

$$OP(i) = (T - 1) - IP(i) \quad (3)$$

Where ' T ' represents the number of gray values in the image.

D. Gamma correction

Gamma Correction is one of the conventional gray level transformations, and it is also known as power-law Transformation which includes n^{th} power and n^{th} root power transformation are expressed in equation (4) as follows,

$$G = cr^\gamma \quad (4)$$

Where ' c ' and ' r ' are +ve constant values, Gamma (γ) values will be varied for enhancing the images at different display and monitor systems.

E. Mean filter

Mean filter is one of the conventional filtering methods for filtering the images from unwanted impulse noises. The mean filter replaces the center pixel (cp) value in the window matrix with an average (mean) of all the pixels in the window. The smoothing and noise level reduction is achieved by decrease the intensity of one pixel to other pixel values. The convolution matrix is used to calculate the mean by sampling the neighborhood pixels. In this design, the 3x3 window matrix in (5) is used for the calculation of mean value as follows

$$\begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \quad (5)$$

F. Median Filter

The Median filter are widely used for the removal of 'salt and pepper' noises. The median filter mainly used for high-speed image acquisition and video capturing in high quality cameras for the image denoising. The sorting methods like Bubble sort is used for sorting the pixels within window for filtering the image. The conventional median filtering methods are processing slow in real time image processing and difficult to use in dedicated hardware platform. In this work, Median filter is in the adaptive form and 3x3 window size is used for processing the noise input.

G. Edge Detection

Edge detection is used to process the detection of edges using Sobel and Prewitt operators; these operators are discrete differentiation operators used for the calculation of the gradient of image function. The Sobel and Prewitt operators are used for convoluting the neighborhood image pixel with filter coefficient values in horizontal and vertical directions.

The Sobel operator uses two gradients like Horizontal ' G_x ,' and vertical ' G_y ' derivative approximation to form 3X3 sliding window is shown in equation (6) as follows,

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * IP(i) \quad \text{And} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * IP(i) \quad (6)$$

Where $IP(i)$ represents the input image

Similarly, for the Prewitt edge detection, The ' G_x ' and ' G_y ' are represented in equation (7) as follows,

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} * IP(i) \quad \text{And} \quad G_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * IP(i) \quad (7)$$

The magnitude of the gradient of both Sobel and Prewitt edge detection is calculated by integrating both ' G_x ,' and ' G_y ' are represented in equation (8) as follows,

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (8)$$

H. Image Thresholding

Thresholding is a segmentation method used to create binary images in gray level. If the threshold value (T) is higher than the input image, then the output of each pixel is replaced with the minimum '0' value (Black). Similarly, if the Threshold (T) values less than the input image, then the output of each pixel is replaced with the maximum '255' value (White).

I. Histogram of the image

Histogram is used to increase or decrease the contrast of the image. The histogram equalization is performed using cumulative distributive function (CDF) based gray levels and stretches the histogram of the input image from range 0-255 value.

3.2 Interface Protocols

The performance of MPSoC is determined by using on-chip buses for interconnection between multiple cores. These are traditional, can communicate master to slave one at a time which affects the performance of the whole MPSoC system. There are many interconnection bus protocols are available like Wishbone, AHB, and AXI provides interface solutions in MPSoC system. By using FPGA and ASIC prototyping, can improve the execution time and reduce the complexities of the whole system. In the present work, the high speed AXI interface bus is used for communicating the master modules with slave module. The AXI bus supports 16 Masters and 16 Slaves transactions at time.

4 Imaging Chip Model

The imaging chip model has three sub-modules includes the Master module, Interface, and slave modules. The overview of the Imaging chip using the AXI-4 interface is

represented in the **fig.1**. The image input and output sizes are considered as 256x256. The detailed imaging chip sub-modules operation is elaborated in the below section.

Fig.1 Proposed Design of Imaging Chip using AXI Protocol

4.1 Master Module

The master module has a pre-processing image module for neighboring pixel creation followed by Image enhancement Module (IEM) for image processing operations and stores the processed images temporarily in the register module. The imaging chip works based on modes of operation for image enhancement modules and is tabulated in **table 1** and theoretical operations are explained in earlier section 3.1. The complete Hardware module of Image enhancement Algorithms are represented in **fig.2**.

The first step of the imaging chip is a preprocessing module. The image preprocessing receives the input image stored in the temporary ROM-1, followed by the creation of neighboring pixels and stored in ROM-2. The neighbored pixels are extracted using 3X3 matrix generation operation and store it in ROM-3. The ROM-3 outputs are used for the image enhancement module as an input. The image enhancement module (IEM) has 10 different enhancement operations [32] works based on the modes for imaging chip-Master module creation. The IEM operations are explained as follows.

When mode-0000 is activated for the Brightness control operation, the preprocessed output is added with Constant (C) and results are compared with 255 and stored in register for the generation of brightness control Image. When mode-0001 is activated for the Contrast Stretching operation, the preprocessed output image computes the Minimum and maximum values using Counter method and followed by contrast operation using equation 2 and stored the results in register for the generation of Contrast Stretching image. When mode-0010 is activated for the Negative Image transformation operation, the preprocessed output image subtracted from Number of Gray level 'N' and stores the results in register for the generation of Negative image.

When mode-0011 is activated for the Gamma correction operation, the preprocessed output are multiplied by itself ($p * p$) and results are multiplied with Constant (C) and results are right shifted and map the pixel values from range 0 to 255. The mapped values are stored in register for the generation of Gamma correction Image. When mode-0100 is activated for the Mean filtering operation, the preprocessed output values are stored in 3X3 single window mask. Calculate the mean of all the 3x3 window pixel values and updated in Center pixel and the results are stored in register for the generation of mean filtered Image. When mode-0101 is activated for the Adaptive Median Filter (AMF), and it is one of the best filtering methods to suppress unwanted impulse noises.

Table 1 Image enhancement Module (IEM) - Modes of operation

Mode	Image enhancement module	Mode	Image enhancement module
0000	Brightness control	0101	Adaptive Median Filter
0001	Contrast Stretching	0110	Sobel Edge Detection
0010	Negative Image Trans.	0111	Prewitt edge detection
0011	Gamma Correction	1000	Image Thresholding
0100	Mean filter	1001	Image Histogram

Fig.2 Block Diagram of Image enhancement Algorithms

Fig.3 Adaptive Median Filtering Module

The detailed diagram of the AMF module is represented in **fig.3**. AMF mainly has window 3X3 module, 3-input sorter, Adaptive computational block followed by error detection block. The 3X3 window operation module has 9 data flip-flops (DFF's) to form 9- different window values for median computation. The median filter has a sorter module to find the median values using Multiplexors and comparators. The maximum and minimum values are used in error detection to find the error value. The adaptive computation block finds the principal median value using Comparators and basic AND gates along the center window pixel. The adaptive computation adopts the suitable valid output value to find the mean filter value. The error detection block finds the error pixel value using median value along with maximum and minimum values. This error detection gives information about the corrupted and uncorrupted pixel of the filtered output image.

When mode-0110 or 0111 is activated for the Sobel or Prewitt operation, the preprocessed output data are used for the calculation of Gradient Gx and Gy calculation and then apply Sobel or Prewitt operations using equations (6) and (7) respectively and store in results in register for the generation of Edge detection Image. When mode-1000 is activated for the Image Thresholding operation, the preprocessed output data is compared, if it is less than Constant 'T' then replace with Zero value and if it is greater than 'T' then 255 value will be updated and store any of the 0 or 255 values in register for the generation of Threshold Image. When mode-1001 is activated for the Image Histogram operation, the gray level block counts the appearance of the pixel values in the preprocessed output. The CDF block counts the each value in the array with previous one. The multiplier will multiply the CDF array value with previous gray value. Finally map the new gray values into number of pixels and store the mapped values in register for the generation of Histogram image of input Image.

4.2 AXI- Interface Module

The AXI-4 protocol is a high-speed interface bus protocol and which corrects the drawbacks of existing bus interface protocols. The overview of the proposed High-speed AXI-4 Interface is represented in **fig.4**. The model includes AXI-4 Master Module; interconnect module, and AXI4 Slave module. These modules are interconnected, each based on AMBA-AXI-4 Input-outputs and its specifications. The AXI-4 master and slave modules are designed using FSM along with five transactions includes write address, write data, write a response, read address, and read data. The proposed module provides a high-speed interface to the image enhancement module (Main master) and dual-port memory module (slave module). The AXI-4 supports Quality of service signaling, an extension of burst length support, write- response updation and cache signals updation and out of order requirements. The AXI-4 performs the burst based transaction. The address channel provides control and addresses information of every transaction, which offers the type of data is being transferred. The Write data channel provides the data information details from the master to the slave module. The write

response channel offers completion of write addresses and data transactions using a response and acknowledges signals. The read data channel gives the data transaction information from slave to master module along with a response to the slave module.

Fig.4 AXI-4 Master-Slave Interface Module

The AXI-4 Master module is designed using five-channel transactions. The AXI-4 master module is initialized by Write address valid (AWVALID) and Write address ready (AWREADY) signals. The AWVALID will be activated, followed by AWREADY, if the slave module is ready. Then the master module initiates the valid write address (AWADDR) transaction to slave for each transaction. The design considers increment address burst type (AWBURST) along with bufferable and cacheable (AWCACHE) transactions for address processing. The write data transaction will be initiated parallelly along with the write address. When the slave ready signal (WREADY) is ready, the write valid signal (WVALID) will be activated. The Write data (WDATA) contains image data transactions based on the address and performs the write data transaction until the last transaction (WLAST). The Write response valid (BVALID) from the slave, followed by writing ready signal (BREADY) from the master, will be activated and set to low for each data transaction is completed. Finally, the master writes response will set the OKAY signal for the data transaction completion.

The AXI-4 Slave module process is similar to write address, data and response transactions. The only difference is instead of write consider read signals, and master signals act as slave signals and vice-versa along with Reading response build with reading data transaction in the slave module.

The **Slave Module** contains dual-port memory, which is having an 8-bit width, and depth depends upon the image size. In the present work, 65536 memory locations are considered. The memory module stores the image and connects to the external world.

5 Results and Performance Analysis

The Imaging chip Module results are obtained on the Xilinx ISE environment using Verilog-HDL and simulated on Modelsim Simulator. The Imaging chip model is implemented on Artix-7 FPGA and physically verified using the Xilinx Chipscope pro tool. The Imaging Chip model considers 256X256 image size as input for processing the image enhancement as a master module and store the results in memory as a slave module via an AXI-4 interface module. The imaging chip module works based on modes of operations. The 4-bit more input is introduced to select the corresponding image enhancement module (IEM). The input image with each pixel is 8-bit and is considered as an input to the master image enhancement module until the last 65535th pixel, which is followed by the AXI-4 interface module. The interface module establishes the communication between the master and slave modules. The dual-port memory module acts as a slave, which stores the enhanced image output.

The imaging chip hardware results are represented in **fig.5**. Lena.png is considered as an input image for the imaging chip module is in **fig.5 (a)**. The imaging chip outputs are obtained one at a time based on the modes. The brightness control for mode-0000, contrast stretching for mode-0001, negative image transformation for mode-0010, gamma correction for mode-0011, mean filter for mode-0100, Adaptive median filter for mode-0101, Sobel edge detection for mode-0110, Prewitt edge detection for mode-0111, and image Thresholding for mode-1000 are the imaging chip results and are represented in **fig.5(b-j)** respectively.

Fig.5 Imaging Chip- Hardware Results

The AMF filter results for different noise levels are represented in **fig.6**, and the performance metrics of the AMF is analyzed in terms of PSNR and MSE along with the corresponding numeric values are tabulated in **table 2**. The different noise levels, like 1 %, 5%, 10%, and 20% salt and pepper noise, are applied to the input image represented in **fig.6(a-d)**, and corresponding Adaptive median filtered outputs are represented in **fig.6(e-h)** respectively.

Table 2 AMF Filter- Different Noise levels, PSNR and MSE Results

Noise level	PSNR(dB)	MSE
1%	36.73	13.79
5%	35.94	16.54
10%	35.12	19.97
20%	33.34	30.12

Fig.6 AMF- Filter Results for different Noise Levels

The filtered outputs are analyzed using PSNR and MSE ratio, which are used to find the effectiveness of the image quality. The input image is corrupted by different noise levels 1%, 5%, 10% and 20%, the corresponding PSNR (dB) obtained for AMF is 36.73, 35.94, 35.12 and 33.34 respectively. Similarly, the mean square error (MSE) obtained 13.79, 16.54, 19.97 and 30.12 for AMF with different noise-levels 1%, 5%, 10% and 20% respectively. The quality of the image is good even for 20% noise corruption with better PSNR and MSE ratio, which improves the robustness of the AMF module. The input image and its histogram are represented in **fig.7**.

Fig.7 Image Histogram Results

The imaging chip synthesized results are obtained after the place and route operation using Xilinx ISE software. The imaging chip master module hardware constraints include slice LUTs (chip Area), Maximum operating Frequency(MHz), Total power (W), and execution time(ms) are tabulated in **table 3**. The Sobel and Prewitt edge detection consumes more area than other operations because of edge detection operations. For the first five modes of operations, the maximum frequency is not generated because of the pre-processed 3x3 window output as an input to these operations. The AMF works at a maximum frequency of 1384.08MHz with pipelined architecture. The total power is generated using Xilinx –Xpower analyzer, which includes both static and dynamic power. For different image operations, The master module utilizes less power individually. The execution time is calculated based on the simulation results by providing the image as input to the imaging chip test cases. The execution time includes 256X256 image read and store in a temporary memory location, followed by the master module, interface module, and slave module output result till the last pixel. The imaging chip utilizes less execution time on an average of 1.31ms, which is quite suitable for real-time image processing applications.

Table 3 Imaging Chip- IEM (Master Module) Hardware Constraint Results

Mode	Operation	Slice LUTs (Area)	Max.Frequency (MHz)	Total Power (W)	Execution time (ms)
0000	Brightness control	6	NA	0.091	1.3211
0001	Contrast Stretching	206	NA	0.085	1.321
0010	Negative Image Trans.	8	NA	0.086	1.321
0011	Gamma Correction	6	NA	0.084	1.298
0100	Mean filter	19	NA	0.092	1.326
0101	Adaptive Median Filter	103	1384.083	0.088	1.311
0110	Sobel Edge Detection	16331	174.35	0.13	1.326
0111	Prewitt edge detection	14408	174.35	0.128	1.326
1000	Image Thresholding	4	NA	0.087	1.319

100 1	Image Histogram	272	368.026	0.119	1.31
----------	-----------------	-----	---------	-------	------

The Main imaging chip module along with IEM (master) module and AXI module's hardware constraints like area, frequency, power(W), and Memory (KB) results are obtained after the place and route operation, and it is tabulated in **table 4**. Table 4 shows the difference in hardware constraints for the image enhancement module with AXI (Imaging chip) and IEM without the AXI interface. The imaging chip consumes less area (Slice LUTs), operates at a better frequency, and consumes less chip power than IEM without the AXI interface.

The area utilization of imaging chip in terms of Slices Register and LUT improves < 1%, and LUT-FF pairs improve 44% IEM without the AXI interface. The Maximum frequency of imaging chip speeds up 3.2 % than IEM without the AXI interface. The power consumer of imaging chip improves 77% then IEM without the AXI interface. The total memory usage of imaging chip and IEM is 1317284 KB and 1309644 KB, respectively. The individual synthesis results of the AXI-4 Master-slave module is also incorporated in the same **table 4**.

Table 4 Imaging Chip –Hardware Constraint Results

Resources	Imaging Chip using AXI Module	IEM without AXI Module	AXI Master-Slave Interface Module
Area			
Slice Registers	16889	16926	63
Slice LUTs	47962	48010	77
LUT-FF pairs	4559	8191	56
Timing			
Minimum period (ns)	7.19	7.423	1.782
Max. Frequency (MHz)	139.076	134.711	561.073
Power (W)			
Total Power (W)	0.218	0.99	0.085
Dynamic Power (W)	0.136	0.168	0.003
Memory (KB)			
Total memory usage (KB)	1317284	1309644	355204

The Imaging chip offers high-performance computation than IEM without the AXI interface along with reducing the hardware complexity by introducing the AXI interface module in IEM.

Table 5 Performance metrics of recent Image processing applications with Present work

Designs	Interface	Device	Fre- quency (MHz)	LU Ts	DS Ps	BRA Ms	Execu- tion Time (ms)	FP S
[6]	AXI-4	XC7Z020	530	4772 0	120	120	2.98	68
[8]	AXI-4 Lite	XC6VLX2 40T	100	1007 0	NA	33	3.07	46
[33]	PCI	XC6VLX7 60	150	9247 7	191	95	44	328
This Work	AXI-4	XC7A100T	140	4796 2	5	5	1.31	85. 5

The comparison of real-time image processing application with different interface protocols are tabulated in **table 5**. The present design works at 140 MHz frequency along with utilizes less DSPs and BRAMs chip area than the recent existing image processing applications. The imaging chip process at execution time of 1.31ms and 85.5 Frames per second (FPS).

6 Conclusion

In this manuscript, the imaging chip model is designed using the AXI-4 interface protocol and implemented on an FPGA device. The imaging chip offers a high-speed interface for real-time image processing applications for communication. The Imaging chip has an Image enhancement module (IEM) Master, AXI interface protocol, and Memory (Slave) module. Imaging chip works based on the modes of operation for selecting the image enhancement algorithm quickly. The imaging chip is synthesized and implemented on Artix-7 FPGA, and Hardware constraints like Chip area, frequency, and power are encouraging for real-time MPSoC usage. The imaging chip provides better resource utilization than IEM without the AXI interface. The imaging chip using the AXI interface works at 85.58Mbps, which is better than IEM without the AXI interface at 82.8 Mbps. The imaging chip provides better resource utilization and processing time than the existing recent real time image processing applications. The imaging chip model provides scalability, Robustness, and reconfigurable features, which suit for MPSoC applications. In the future, incorporate the security features into the imaging chip for strengthening from attacks.

References

- [1] Walha, Rim, Fadoua Drira, Frank Lebourgeois, Adel M. Alimi, and Christophe Garcia. "Resolution enhancement of textual images: a survey of single image-based methods." *IET Image Processing* 10, no. 4 (2016): 325-337.

- [2] Niu, Yuzhen, Yini Zhong, Wenzhong Guo, Yiqing Shi, and Peikun Chen. "2D and 3D Image Quality Assessment: A Survey of Metrics and Challenges." *IEEE Access* 7 (2018): 782-801.
- [3] Schuch, Patrick, Simon Schulz, and Christoph Busch. "Survey on the impact of fingerprint image enhancement." *IET Biometrics* 7, no. 2 (2017): 102-115.
- [4] Jiang, He, and Jie Yang. "In-place similarity and its applications in image and video detail enhancement." *Electronics Letters* 52, no. 12 (2016): 1022-1024.
- [5] Saponara, Sergio, Giovanni Ramponi, Stefano Marsi, Gerard de Haan, and Erwin Bellers. "Guest editorial: special issue on algorithms and architectures for real-time image and video enhancement." (2013): 1-3.
- [6] Amiri, Moslem, Fahad Manzoor Siddiqui, Colm Kelly, Roger Woods, Karen Rafferty, and Burak Bardak. "FPGA-based soft-core processors for image processing applications." *Journal of Signal Processing Systems* 87, no. 1 (2017): 139-156.
- [7] Gu, Jing, and Yang Huayu. "Real-Time Image Collection and Processing System Design." In *2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, pp. 1649-1652. IEEE, 2015.
- [8] Lifa, Adrian, Petru Eles, and Zebo Peng. "A reconfigurable framework for performance enhancement with dynamic FPGA configuration prefetching." *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35, no. 1 (2015): 100-113.
- [9] Kowalczyk, Marcin, Dominika Przewlocka, and Tomasz Krvjak. "Real-time implementation of contextual image processing operations for 4K video stream in Zynq UltraScale+ MPSoC." In *2018 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pp. 37-42. IEEE, 2018.
- [10] Carrizosa-Corral, Fernando, Alberto Vázquez-Cervantes, Josué-Rafael Montes, Teresa Hernández-Díaz, Julio Cesar Solano Vargas, Leonardo Barriga-Rodríguez, Jorge Alberto Soto-Cajiga, and Hugo Jiménez-Hernández. "FPGA-SoC implementation of an ICA-based background subtraction method." *International Journal of Circuit Theory and Applications* 46, no. 9 (2018): 1703-1722.
- [11] Peesapati, Rangababu, Samrat L. Sabat, and J. Nayak. "FPGA-based embedded platform for fiber optic gyroscope signal denoising." *International Journal of Circuit Theory and Applications* 42, no. 7 (2014): 744-757.
- [12] Kammoun, Manel, Ahmed Ben Atitallah, Rabie Ben Atitallah, and Nouri Masmoudi. "Design exploration of efficient implementation on SoC heterogeneous platform: HEVC intra prediction application." *International Journal of Circuit Theory and Applications* 45, no. 12 (2017): 2243-2259.
- [13] Xiao, Fu-ming, Dong-sheng Li, Gao-ming Du, Yu-kun Song, Duo-li Zhang, and Ming-lun Gao. "Design of AXI bus based MPSoC on FPGA." In *2009 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication*, pp. 560-564. IEEE, 2009.
- [14] Guruprasad S P and Chandrasekar B.S, "Performance Evaluation of Network Gateway Design for NoC based System on FPGA Platform" *International Journal of Advanced Computer Science and Applications (IJACSA)*, 10, no.9, (2019):287:292.
- [15] Dutta, Anandi, and Magdy Bayoumi. "Introducing a Novel Smart Design Framework for a Reconfigurable Multi-Processor Systems-on-Chip (MPSoC) Architecture." In *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 1-3. IEEE, 2016.
- [16] Wang, Chao, Xi Li, Yunji Chen, Youhui Zhang, Oliver Diessel, and Xuehai Zhou. "Service-oriented Architecture on FPGA-based MPSoC." *IEEE Transactions on Parallel and Distributed Systems* 28, no. 10 (2017): 2993-3006.

- [17] Karim, Mohammed, and Mohamed-Yassine Amarouch. "An FPGA-based MPSoC for real-time ECG analysis." In *2015 Third World Conference on Complex Systems (WCCS)*, pp. 1-4. IEEE, 2015.
- [18] Huang, Shih-Chia, and Wen-Chieh Chen. "A new hardware-efficient algorithm and reconfigurable architecture for image contrast enhancement." *IEEE Transactions on Image Processing* 23, no. 10 (2014): 4426-4437.
- [19] Singh, Koushlendra Kumar, Durgesh Kumar, Shubham Chauhan, and Manish Kumar Bajpai. "Parallel architecture based fast algorithm for image enhancement." In *2015 IEEE Bombay Section Symposium (IBSS)*, pp. 1-6. IEEE, 2015.
- [20] Pugazhenth, A., and L. S. Kumar. "Image contrast enhancement by automatic multi-histogram equalization for satellite images." In *2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN)*, pp. 1-4. IEEE, 2017.
- [21] Mahajan, Arushi, and Divya Gupta. "Image contrast enhancement using Gaussian Mixture model and genetic algorithm." In *2017 International Conference On-Smart Technologies For Smart Nation (SmartTechCon)*, pp. 979-983. IEEE, 2017.
- [22] Li, Lin, Ronggang Wang, Wenmin Wang, and Wen Gao. "A low-light image enhancement method for both denoising and contrast enlarging." In *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3730-3734. IEEE, 2015.
- [23] Tian, Jie, and Xijie Yin. "Adaptive image enhancement algorithm based on the model of surface roughness detection system." *EURASIP Journal on Image and Video Processing* 2018, no. 1 (2018): 103.
- [24] Zhao, Zhou, and Yicong Zhou. "Comparative study of logarithmic image processing models for medical image enhancement." In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 001046-001050. IEEE, 2016.
- [25] Khoukhi, Hasnae, and My Abdelouahed Sabri. "Comparative study between HDLs simulation and Matlab for image processing." In *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)*, pp. 1-6. IEEE, 2018.
- [26] Liviu, Tigaeru. "FPGA Implementation of a Fuzzy Rule Based Contrast Enhancement System for Real Time Applications." In *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 117-122. IEEE, 2018.
- [27] Chiuchisan, Iuliana. "An approach to the Verilog-based system for medical image enhancement." In *2015 E-Health and Bioengineering Conference (EHB)*, pp. 1-4. IEEE, 2015.
- [28] Aranda, Luis Alberto, Pedro Reviriego, and Juan Antonio Maestro. "Error detection technique for a median filter." *IEEE Transactions on Nuclear Science* 64, no. 8 (2017): 2219-2226.
- [29] Kumar, Vineet, Abhijit Asati, and Anu Gupta. "Low-latency median filter core for hardware implementation of 5×5 median filterings." *IET Image Processing* 11, no. 10 (2017): 927-934.
- [30] Cadenas, J. "Pipelined median architecture." *Electronics Letters* 51, no. 24 (2015): 1999-2001.
- [31] Tidal, Neethika. "High Performance Network on Chip using AXI4 protocol interface on an FPGA." In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1647-1651. IEEE, 2018.
- [32] Archana, H. R., and KS Vasundara Patel. "A Novel Design and Implementation of Imaging Chip Using AXI Protocol for MPSoC on FPGA." In *Proceedings of the Computational Methods in Systems and Software*, pp. 44-57. Springer, Cham, 2018.

- [33] Ma, Xiaoyin, Walid A. Najjar, and Amit K. Roy-Chowdhury. "Evaluation and acceleration of high-throughput fixed-point object detection on FPGAs." *IEEE Transactions on Circuits and Systems for Video Technology* 25, no. 6 (2014): 1051-1062.

Figures

Figure 1

Proposed Design of Imaging Chip using AXI Protocol

Figure 2

Block Diagram of Image enhancement Algorithms

Figure 3

Adaptive Median Filtering Module

Figure 4

AXI-4 Master-Slave Interface Module

Figure 5

Imaging Chip- Hardware Results

Figure 6

AMF- Filter Results for different Noise Levels

Figure 7

Image Histogram Results