A 4K RESOLUTION-ACCOMPLISHED FPGA IMPLEMENTATION OF SOLITARY IMAGE PARTICLE MAPS

N. SUBASH*, P. SRIKANTH,**

ASSOCIATE PROFESSOR*,PG SCHOLAR,**

DEPARTMENT OF ECE, MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY MAISAMMAGUDA, KOMPALLY, DHULAPALLY, SECUNDERABAD – 500100

ABSTRACT: This paper presents a fast and compact hardware implementation using an efficient haze removal algorithm. The algorithm employs a modified hybrid median filter to estimate the hazy particle map, which is subsequently subtracted from the hazy image to recover the haze-free image. Adaptive tone remapping is also used to improve the narrow dynamic range due to haze removal. The computation error of the proposed hardware architecture is minimized compared with the floating-point algorithm. To ensure real-time hardware operation, the proposed architecture utilizes the modified hybrid median filter using the well-known Batcher's parallel sort. Hardware verification confirmed that high-resolution video standards were processed in real time for haze removal.

KEYWORDS: hardware implementation; haze removal; hazy particle map; modified hybrid median filter; Batcher's parallel sort

INTRODUCTION With the development of self-driving vehicles and intelligent monitoring systems, there is a growing demand for haze removal algorithms [1–3]. Loss of contrast and color fidelity of observed images due to hazy weather can strongly interfere with these applications. Therefore, several image processing algorithms are needed to improve the visibility of observed images. Some of them include haze removal, noise reduction, and wide dynamic range expansion. The haze removal algorithms can be roughly divided into two main categories: multiple image processing [4,5] and single image processing [6–17]. Although multiple image processing is generally superior in performance, it is not attractive due to the high complexity of frame storage memories. Therefore, the recent focus has shifted to single image processing. He et al. proposed the dark channel prior (DCP) [6], which is used to estimate the transmission map based on the assumption that a few pixels with at least one very dark channel are present. However, block artifacts occurred in the estimated transmission due to the pre-assumed constant transmission in local patches. Therefore, the computationally intensive soft matting was used additionally to refine the transmission, resulting in a very slow processing speed of the DCP. Tarel et al. proposed a faster haze removal method with a standard median filter instead of the time-

consuming soft matting approach [7]. Thereby, the processing time was shortened, and the haze was reduced. However, this method resulted in halo artifacts in the corner (or line) areas with depth discontinuities caused by the large median filter for the estimation of the whiteness of the observed image. Nishino et al. proposed a Bayesian probability-based haze removal method utilizing a factorial Markov random field to model the dependence between scene albedo and depth from a single hazy image [8]. The algorithm effectively reduced the haze and color artifacts. However, it was also a time-consuming method requiring manual parametrization by depth prior [3,8]. Zhu et al. found that the difference between the brightness and saturation in a hazy image was generally correlated with the increasing haze concentration along with the scene depth [9]. Therefore, a linear model called color attenuation prior (CAP) was set up to estimate the scene depth. Using the linear model and the guided image filter [19], the processing time and the haze in the observed images were reduced. However, this approach failed to handle grayscale images and dark regions in the images adequately. Bui et al. proposed a color ellipsoid prior (CEP) [15], which was theoretically similar to DCP as proved by Gibson et al. [16]. They also proposed using the fuzzy segmentation process in lieu of soft matting and derived an efficient formula for estimating and refining the transmission map concurrently. This could reduce haze removal time. However, experiments with provided code and parameters have shown that CEP tends to excessively dehaze the input images, causing visible color artifacts in the results. Liu et al., in contrast, proposed a unified strategy to estimate the transmission map and scene radiance simultaneously using total variation-based regularization [17]. Nevertheless, this iterative method required the atmospheric light and transmission map estimated by DCP in its initialization stage. Therefore, it was not computationally friendly. Most importantly, all the algorithms mentioned above require multiple frame memories, which greatly increase the hardware complexity at a high cost in the hardware implementation phase. Another approach proposed by Kim et al. removed haze from hazy images using hazy particle maps, estimated via modified hybrid median filter (mHMF) [11]. It used fewer frame storage memories in the hardware design, and its algorithmic complexity was solely a linear function of input image pixels, as will be discussed in Section 2. In real-time image processing systems, a fast and compact solution for these algorithms is always essential. Park et al. proposed a fast implementation of DCP by lightening the complexity of the atmospheric light estimation and exploiting the high similarity of successive video frames [12]. In contrast, Salazar-Colores et al. shortened the processing

time of DCP by utilizing morphological reconstruction instead of soft matting to refine the transmission map [13]. According to the provided experimental results, Park et al.'s design was able to process a 320 × 240 image in 0.228 s (i.e., 1/0.228 ≈ 4.39 frames per second (fps)), while Salazar-Colores et al.'s design required 0.14 s to handle a 1920 × 1080 image (i.e., 1/0.14 ≈ 7.14 fps). These faster versions of DCP are still inappropriate for real-time applications requiring a processing speed of at least 25 fps. Similarly, Table 1 shows the system complexity based on the CPU times using He et al.'s, Tarel et al.'s, Nishino et al.'s, Zhu et al.'s, Bui et al.'s, and Kim et al.'s algorithms [6–9,11,15]. Nishino et al.'s algorithm was programmed via Anaconda, while the other five algorithms were programmed via MATLAB R2019a. They were all tested on a Core i7-6700 CPU (3.4 GHz) with 16GB RAM. For the smallest 320 × 240 image, the fastest Bui et al.'s algorithm was only able to handle up to 20 fps (= 1/0.05). For the 4K standard of a 4096 × 2160 image, the speed of even the fastest Kim et al.'s algorithm was significantly declined to 0.09 fps (≈1/11.09), and the processing time of He et al.'s algorithm could not be measured due to insufficient RAM. This finding suggests that the real-time processing is completely impossible, and the memory capacity is a key factor in the hardware design for haze removal.

Table 1. Processing time in seconds of haze removal algorithms.

Image Size	He [6]	Tarel [7]	Nishino [8]	Zhu [9]	Bui [15]	Kim [11]
320 × 240	4.11	0.23	29.56	0.11	0.05	0.08
640 × 480	17.83	0.79	96.50	0.47	0.65	0.35
800 × 600	28.05	1.69	129.67	0.72	1.06	0.67
1024 × 768	46.82	2.18	173.86	1.18	1.74	0.96
1920 × 1080	143.86	4.27	583.18	3.14	4.76	2.25
4096 × 2160	-	25.76	2388.85	13.20	20.38	11.09

An alternative might be to implement these algorithms on a graphics processing unit (GPU) with GPU-accelerated libraries included in the Compute Unified Device Architecture toolkit. GPU can be a main means of realizing data-driven haze removal algorithms. For example, a cascaded convolutional neural network for dehazing proposed in Reference [14] was able to handle a 640 × 480 image in 0.1029 s (i.e., 1/0.1029 ≈ 9.75 fps). Unfortunately, this processing speed still does not meet the requirement of real-time processing. In addition, an in-depth study on both classical and deep learning haze removal methods stated that the classical ones tend to generate results favored by human perception [20]. As well as this, researchers may be increasingly interested in designing the hardware using field-programmable gate arrays (FPGA) because of the high cost of the GPU platform in terms of both market price and energy consumption [21] and the inability to handle large image sizes in real time. Several hardware implementations until now including Shiau et al.'s and other designs [1,2,22] are only appropriate for small-size

imaging applications because of their low throughput. Therefore, a strategy for processing a large size image in real time is highly required. This paper presents a 4K-capable hardware design using Kim et al.'s algorithm that meets the real-time processing criteria and does not require frame storage memory for consecutive images (or video frames) with high similarity. The proposed hardware architecture also utilizes the noted Batcher's parallel sorting algorithm [23] to realize the mHMF, consequently reducing the resource utilization and boosting the processing speed significantly.

LITERATURE REVIEW

IN "REFERENCELESS PREDICTION OF PERCEPTUAL FOG DENSITY AND PERCEPTUAL IMAGE DEFOGGING" We propose a referenceless perceptual fog density prediction model based on natural scene statistics (NSS) and fog aware statistical features. The proposed model, called Fog Aware Density Evaluator (FADE), predicts the visibility of a foggy scene from a single image without reference to a corresponding fog-free image, without dependence on salient objects in a scene, without side geographical camera information, without estimating a depth-dependent transmission map, and without training on human-rated judgments. FADE only makes use of measurable deviations from statistical regularities observed in natural foggy and fog-free images. Fog aware statistical features that define the perceptual fog density index derive from a space domain NSS model and the observed characteristics of foggy images

IN "CHROMATIC FRAMEWORK FOR VISION IN BAD WEATHER" Conventional vision systems are designed to perform in clear weather. However, any outdoor vision system is incomplete without mechanisms that guarantee satisfactory performance under poor weather conditions. It is known that the atmosphere can significantly alter light energy reaching an observer. Therefore, atmospheric scattering models must be used to make vision systems robust in bad weather. In this paper, we develop a geometric framework for analyzing the chromatic effects of atmospheric scattering. First, we study a simple color model for atmospheric scattering and verify it for fog and haze. Then, based on the physics of scattering, we derive several geometric constraints on scene color changes, caused by varying atmospheric conditions. Finally, using these constraints we develop algorithms for computing fog or haze color, depth segmentation, extracting three dimensional structure, and recovering "true" scene colors, from two or more images taken under different but unknown weather conditions.

IN "VISIBILITY IN BAD WEATHER FROM A SINGLE IMAGE" Bad weather, such as fog and haze, can significantly degrade the visibility of a scene. Optically, this is due

to the substantial presence of particles in the atmosphere that absorb and scatter light. In computer vision, the absorption and scattering processes are commonly modeled by a linear combination of the direct attenuation and the airlight. Based on this model, a few methods have been proposed, and most of them require multiple input images of a scene, which have either different degrees of polarization or different atmospheric conditions. This requirement is the main drawback of these methods, since in many situations, it is difficult to be fulfilled. To resolve the problem, we introduce an automated method that only requires a single input image. This method is based on two basic observations: first, images with enhanced visibility (or clear-day images) have more contrast than images plagued by bad weather; second, airlight whose variation mainly depends on the distance of objects to the viewer, tends to be smooth. Relying on these two observations, we develop a cost function in the framework of Markov random fields, which can be efficiently optimized by various techniques, such as graph-cuts or belief propagation. The method does not require the geometrical information of the input image, and is applicable for both color and gray images.

IN "SINGLE IMAGE DEHAZING" we present a new method for estimating the optical transmission in hazy scenes given a single input image. Based on this estimation, the scattered light is eliminated to increase scene visibility and recover haze-free scene contrasts. In this new approach we formulate a refined image formation model that accounts for surface shading in addition to the transmission function. This allows us to resolve ambiguities in the data by searching for a solution in which the resulting shading and transmission functions are locally statistically uncorrelated. A similar principle is used to estimate the color of the haze. Results demonstrate the new method abilities to remove the haze layer as well as provide a reliable transmission estimate which can be used for additional applications such as image refocusing and novel view synthesis In almost every practical scenario the light reflected from a surface is scattered in the atmosphere before it reaches the camera. This is due to the presence of aerosols such as dust, mist, and fumes which deflect light from its original course of propagation. In long distance photography or foggy scenes, this process has a substantial effect on the image in which contrasts are reduced and surface colors become faint. Such degraded photographs often lack visual vividness and appeal, and moreover, they offer a poor visibility of the scene contents. This effect may be an annoyance to amateur, commercial, and artistic photographers as well as undermine the quality of underwater and aerial photography.

IN "AN IMPROVED DARK-OBJECT SUBTRACTION TECHNIQUE FOR ATMOSPHERIC SCATTERING CORRECTION OF MULTISPECTRAL DATA" Digital analysis of remotely sensed data has become an important component of many earth-science studies. These data are often processed through a set of preprocessing or "clean-up" routines that includes a correction for atmospheric scattering, often called haze. Various methods to correct or remove the additive haze component have been developed, including tho "- ",~ely used dark-object subtraction technique. A problem with most o1 these methods is that the haze values for each spectral band are selected independently. This can create problems because atmospheric scattering is highly wavelength-dependent in the visible part of the electromagnetic spectrum and the scattering values are correlated with each other. Therefore, multispectral data such as from the Landsat Thematic Mapper and Multispectral Scanner must be corrected with haze values that are spectral band dependent. An improved dark-object subtraction technique is demonstrated that allows the user to select a relative atmospheric scattering model to predict the haze values for all the spectral bands from a selected starting band haze value. The improved method normalizes the predicted haze values for the different gain and offset parameters used by the imaging system

HAZE REMOVAL

Haze is traditionally an atmospheric phenomenon in which dust, smoke, and other dry particulates obscure the clarity of the sky. The World Meteorological Organization manual of codes includes a classification of horizontal obscuration into categories of fog, ice fog, steam fog, mist, haze, smoke, volcanic ash, dust, sand, and snow.[1] Sources for haze particles include farming (ploughing in dry weather), traffic, industry, and wildfires. Seen from afar (e.g. an approaching airplane) and depending on the direction of view with respect to the Sun, haze may appear brownish or bluish, while mist tends to be bluish grey. Whereas haze often is thought of as a phenomenon of dry air, mist formation is a phenomenon of humid air. However, haze particles may act as condensation nuclei for the subsequent formation of mist droplets; such forms of haze are known as "wet haze." Haze also occurs when there is too much pollution in the air while there is also dust In meteorological literature, the word haze is generally used to denote visibility-reducing aerosols of the wet type. Such aerosols commonly arise from complex chemical reactions that occur as sulfur dioxide gases emitted during combustion are converted into small droplets of sulfuric acid. The reactions are enhanced in the presence of sunlight, high

relative humidity, and stagnant air flow. A small component of wet-haze aerosols appear to be derived from compounds released by trees, such as terpenes. For all these reasons, wet haze tends to be primarily a warm-season phenomenon. Large areas of haze covering many thousands of kilometers may be produced under favorable conditions each summer Haze often occurs when dust and smoke particles accumulate in relatively dry air. When weather conditions block the dispersal of smoke and other pollutants they concentrate and form a usually low-hanging shroud that impairs visibility and may become a respiratory health threat. Industrial pollution can result in dense haze, which is known as smog. Since 1991, haze has been a particularly acute problem in Southeast Asia. The main source of the haze has been fires occurring in Sumatra and Borneo. In response to the 1997 Southeast Asian haze, the ASEAN countries agreed on a Regional Haze Action Plan (1997). In 2002, all ASEAN countries signed the Agreement on Transboundary Haze Pollution, but the pollution is still a problem today. Under the agreement, the ASEAN secretariat hosts a co-ordination and support unit.[2] During the 2013 Southeast Asian haze, Singapore experienced a record high pollution level, with the 3-hour Pollution Standards Index reaching a record high of 401.[3] In the United States, the Interagency Monitoring of Protected Visual Environments (IMPROVE) program was developed as a collaborative effort between the US EPA and the National Park Service in order to establish the chemical composition of haze in National Parks and establish air pollution control measures in order to restore the visibility to pre-industrial levels.[4] Additionally, the Clean Air Act requires that any current visibility problems be remedied, and future visibility problems be prevented, in 156 Class I Federal areas located throughout the United States. A full list of these areas is available on EPA's website.[5]

IMPLEMENTATION

Super-resolution has generated a wide spectrum of studies since the seminal work [12]. And we refer readers to for a comprehensive literature review. The most straightforward methods for super-resolution are those based on interpolations, including nearest-neighbor, bilinear, bicubic, and Lanczos algorithms. These methods usually run fast and are easy to implement, but inevitably produce excessively blurry results. Model-based methods aim to restore high resolution scenes according to the observation model in Figure 1 and with priors (regularizations). Most of the works (e.g., [4]) assume known blur kernels and noise levels, but in reality they can be arbitrary [14]. Example-based approaches learn the mapping between low- and high-resolution patches. These approaches either exploit internal similarity of the same image, or learn the

mapping function from external exemplar pairs. It is worth noticing that deep learning techniques have been successfully applied in super-resolution and often achieve state-of-the-art restoration quality

FPGA-based Neural Network Accelerators

FPGA-based accelerators for neural networks are gaining popularity because of its higher energy-efficiency comparing to GPUs and shorter development cycles comparing to ASICs. Since convolution operations often take up a large proportion of the total operations in neural networks, most of the previous works focus on optimizing convolutions. Many accelerators focus on improving the computational efficiency. They explore parallelism, computing sequences (pipelines), and computation-communication balance by loop optimization techniques like loop unrolling and loop tiling. These techniques are analyzed in depth in.

Real-time super-resolution systems based on the iterative back projection algorithm are presented. It combines and slightly modifies a model-based super-resolution algorithm that assumes identical blur between frames (for computational efficiency), and an iterative one that uses L1-norm minimization (for robustness). Fixed-point precision is used, and a highly pipelined architecture is proposed for the real-time purpose. Szydzik et al.reduces logic occupation when implementing the Non-Uniform Grid Projection Algorithm. In a learning-based super-resolution system is presented. It implements the A+ algorithm using only a few line buffers (and without a frame buffer). The system consists of three pipelined stages, which are an interpolation stage for generating low-frequency parts, a mapping stage to select high-frequency patches by pre-trained regression functions, and a construction stage that enhances and overlaps the low frequency image patches with high-frequency information. Noticing that the second stage handles massive computations and introduces long latency, the operation period in the second stage is doubled, and the system is designed with multiple clock domains. In a convolutional neural network for super-resolution based on FRCNN [8] is implemented on FPGA. Instead of enlarging the input beforehand, it applies horizontal and/or vertical flips to the network input images. This flip prevents the information decrease which occurs in the pre-enlargement process, enabling the network to utilize the most of its input image size

PROPOSED SYSTEM

To process higher resolution videos, it is necessary to redesign a higher performance hardware board system. As shown in Fig. 1, the proposed implementation is suitable for nine paths real-time video data processing. To processing multi-channel videos, previous designed video processing boards also can

seamless merged two High Definition (HD) video channels, and send the merged videos to new designed higher performance hardware system. The new designed system is powerful enough to process the higher resolution videos, and construct 4K real-time videos, which is the core of the implementation.

The new video processing board storage data throughput rate is up to 25.6Gbits/s, which is faster enough to the 4K video processing. The core video processor, Xilinx FPGA XC6SLX150, contains about 147K logic cells, which is enough for the video processing. Moreover, the new system supports hardware core based video processing, such as High Definition Multimedia Interface (HDMI) signal decoding and encoding, noise reduction, video smooth and picture enhancement. ARM Cortex A9 can supports a powerful operation system to analyze the video states and manage the whole implementation. The details about the new board system can be described in two parts: hardware implementation architecture and Hardware Description Language (HDL) implementation architecture.

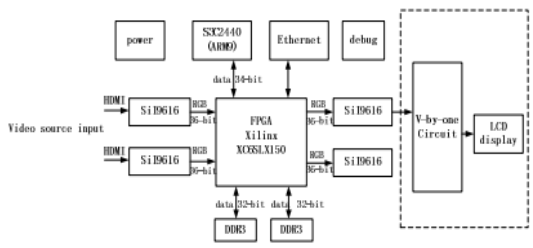


Fig 1 : The processing board architecture

FPGA is the core processor in the whole system, which can drive the DDR3 memory at 25.6Gbits/s and implements the video data processing algorithm. Moreover, ARM 9 is the core controller of the whole system. So, the ARM 9 will configure the SiI9616 and communicate with the FPGA processor. The Ethernet is a useful port, which can communicate with the system and achieve online controlling. The structure of the processing board is shown in Fig. HDMI video input module can receive a group of differential signals to SiI9616 processor. The HDMI signals have a good anti-interference ability. And the HDMI interface has smaller volume than the traditional DVI digital interface. The HDMI video data is organized in standard format. Therefore, to simplify the FPGA processing works and make sure the HDMI videos can be decoded with low time delay, HDMI format data will be converted to parallel data. The parallel data can be converted to 12-bits true color by the SiI9616 processor. The hard process core can convert the format efficiently.

HDMI video output module is similar to the input module. FPGA video processor directly drive the SiI9616 processor by transmitting 36-bit parallel video output. Due to the requirement that the implementation support 4K resolution videos, the speed of parallel data is about 268MHZ. For a normal 1080p FHD video, the speed is only 148.5MHZ. Therefore,

the timing of 4K video should pay more attention. At last, the encoded HDMI 4K resolution video will be transformed to the screen. Xilinx FPGA XC6SLX150 core processor is the core part in the whole system, which will implement the video processing algorithm, and manage multi-channel video data streams.

SiI9616 video processors feature a digital processing core that performs real-time video format conversion and image improvement. The format is supported from any input format to 4K resolutions. The video processor supports multicolor space conversion, mosquito noise reduction, video smoothing, detail enhancement and so on. All of the processes are based on the hard core designed. Therefore, the delay of the videos will be extremely reduced. ARM 9 Coprocessor is the controller of the whole system. To make the implementation more convenient to control, the image analysis and top-level operating platform should be constructed. The Coprocessor will configure the SiI9616 processor; write video information to FPGA and analysis the key image and FPGA state

Ethernet ports can be used to receive control commands. So, we can control the multi-channel videos display through Wireless Fidelity (WIFI). DDR3 memory will be used to store the plenty of video data. In the design, there are 4Gbits space for video data storage. The outside video data is coming into the system in an independent clock. But their speed is much lower than the DDR3 operation speed. Therefore, First Input First Outputs (FIFOs) will be introduced into the system to match the DDR3 operation speed. Also, the DDR3 memory is driven by FPGA core processor.

HDL implementation architecture

The video processing algorithm and simulation is conducted on the ISE software. The video data timing and storage strategy need to be carefully designed and simulated. After all the prepare works, the ISE program will generate the bit-stream file. The next step is downloading the file into the board and beginning to debug the system on board. The HDL implementation architecture is shown in Fig. 3. The architecture included seven kinds of modules: input controller and optimizing module, FIFO buffer module, configure module, video processing arithmetic module, DDR3 controller module, DDR3 driver module and output controller module.

Fig 2 : HDL Implementation Architecture

The video processing board can receive and transmit 2 channels video data at the same time. For the input channels, the videos are totally independent. Therefore, the timing of the two channel videos are different and the pixel clock are not synchronous. The input controller and optimizing module receive and preprocess the video data, which can enhance the color and image quality and reduce the noises. FIFO is an ideal way to deal with the multi-clock problem, which is part and parcel of multi-channel video processing. Configure module is in charge of convert the control words to the detailed parameters of each modules. DDR3 state control module controls both the videos data write and read processes. The DDR3 driver module is controlled by DDR3 state control module and directly drives the DDR3 IP-core. The output controller module generates 4K frames by state machines. To processing a stable frame, the timing need to carefully construct. Video processing arithmetic module is the core part of the HDL implementation. The detailed flow of which is shown in Fig. 3.

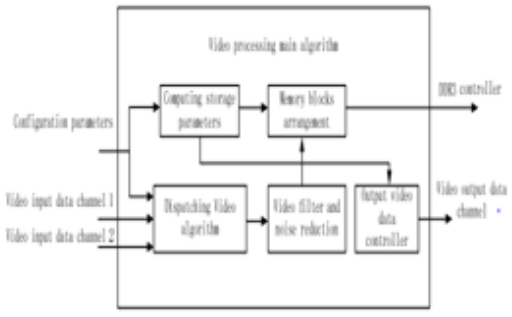


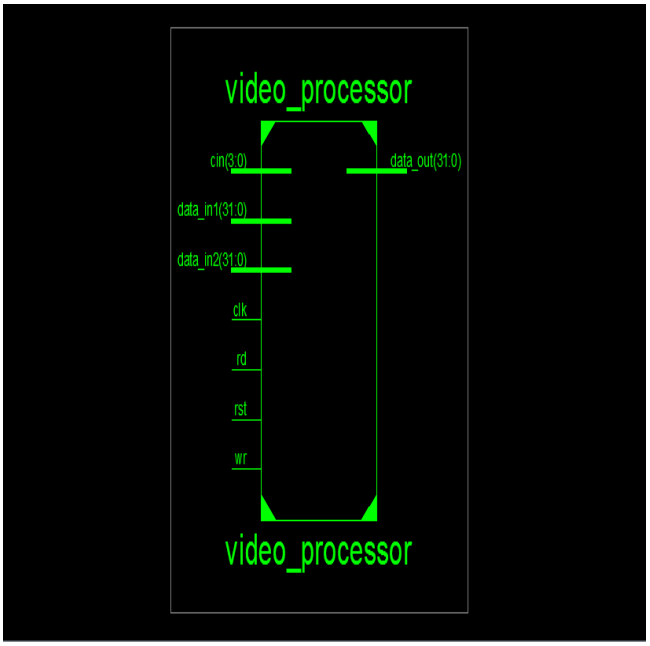
Fig 3 : A Block Diagram Video Processing Arithmetic Model

To processing two path videos, the most important task is controlling the timing, which is to decide the imaging processing sequence and storage blocks arrangement. According to the configuration parameters, the computing storage parameters module will calculate the video storage addresses, video read numbers, video write numbers, and frame packaging parameters. The video storage map will be settling down. The time significant of dispatching correspond video can be get from configuration parameters. After dispatched videos, each frame need filtering and noise reducing. Then, the video data will dispatch to the corresponded storage space. The output data controller can generated 4K resolution videos, according to the system state what the configuration parameters described.

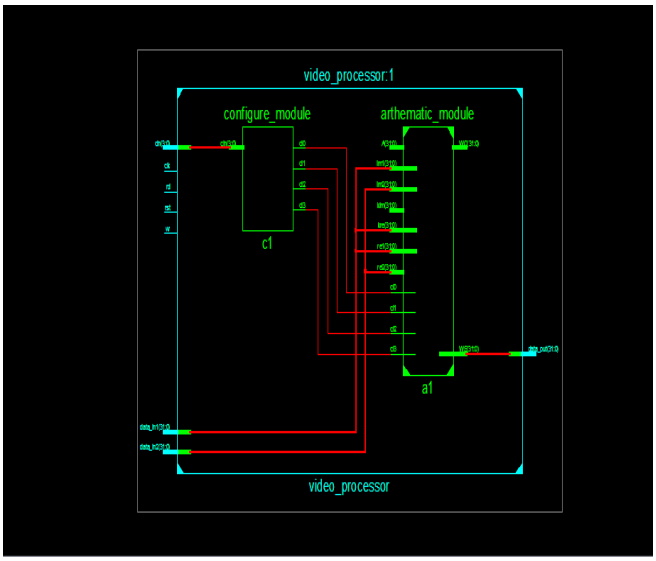
SIMULATION RESULTS

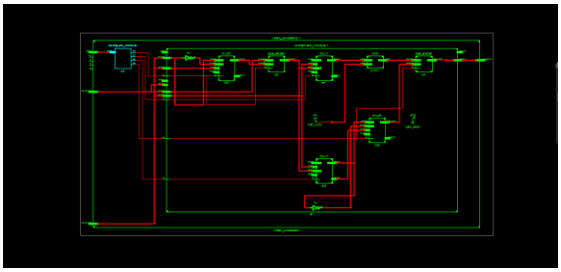
RTL SCHEMATIC

The below is the Top most Schematic of the Video Processor.

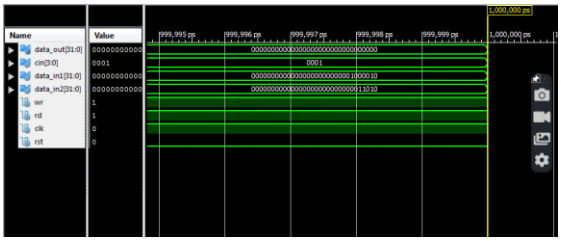


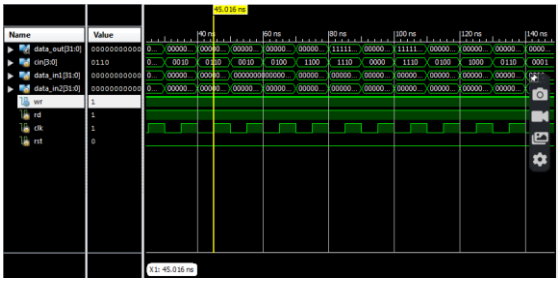
Internal Schematic of Video Processor :





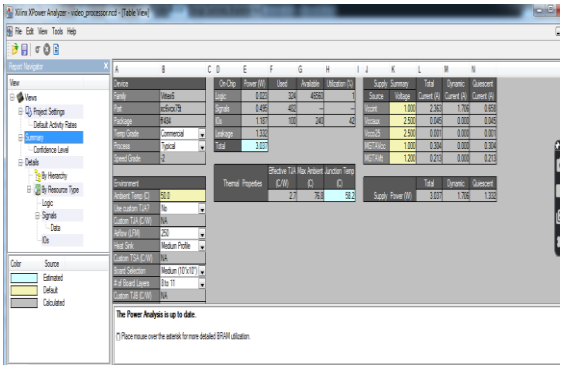
The below are the Simulation output waves:





```
Device utilization summary:
---------------------------

Selected Device : 6vcx75tff484-2


Slice Logic Utilization:
 Number of Slice LUTs:                 1752  out of  46560     3%
    Number used as Logic:              1752  out of  46560     3%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:   1752
   Number with an unused Flip Flop:    1752  out of   1752   100%
   Number with an unused LUT:             0  out of   1752     0%
   Number of fully used LUT-FF pairs:     0  out of   1752     0%
   Number of unique control sets:         0

IO Utilization:
 Number of IOs:                         104
 Number of bonded IOBs:                 100  out of    240    41%
```

```
Timing Details:
---------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default path analysis
  Total number of paths / destination ports: 206266531285 / 32
-------------------------------------------------------------------------
Delay:               16.660ns (Levels of Logic = 43)
  Source:            data_in2<1> (PAD)
  Destination:       data_out<31> (PAD)
```



CONCLUSION:

A fast and efficient hardware architecture for haze removal was presented in this paper. The hardware was compact due to the novel Batcher's sort-based modified hybrid median filter implementation and provided high throughput for real-time video processing based on pipeline architecture. In order to guarantee the high performance of the proposed architecture, the hardware was also designed to minimize the computation error to less than ±0.5 LSB compared with the floating-point algorithm. The maximum operating frequency of the designed hardware was 236.29 MHz, which was fast enough to handle 4K video standards in real time at 26.7 fps. Finally, the designed hardware was verified on the SoC board.

REFERENCES

[1] P. Tang, W. Jin and J. Liu, "Railway inspection oriented foreground objects detection and occlusion reasoning for locomotive-mounted camera video," 2016 35th Chinese Control Conference (CCC), Chengdu, 2016, pp. 10144-10149.

[2] C. H. Chen, T. Y. Chen, D. Y. Huang and K. W. Feng, "Front Vehicle Detection and Distance Estimation Using Single-Lens Video Camera," 2015 Third International Conference on Robot, Vision and Signal Processing (RVSP), Kaohsiung, 2015, pp. 14-17.

[3] V. Chandrasekaran, S. Dantu, P. Kadiyala, R. Dantu and S. Phithakkitnukoon, "Socio-technical aspects of video phones," 2010 Second International Conference on Communication Systems and Networks (COMSNETS 2010), Bangalore, 2010, pp. 1-7.

[4] D. R. Marković, A. M. Gavrovska and I. S. Reljin, "4K video traffic analysis using seasonal autoregressive model for traffic prediction," 2016 24th Telecommunications Forum (TELFOR), Belgrade, 2016, pp. 1-4.

[5] B. C. Sunny, Ramesh R, A. Varghese and V. Vazhayil, "Map-Reduce based framework for instrument detection in large-scale surgical videos," 2015 International Conference on Control Communication & Computing India (ICCC), Trivandrum, 2015, pp. 606-611.

[6] R. Peng, R. J. Sclabassi, Q. Liu, G. Justin and M. Sun, "Synthesizing Multi-View Video Frames for Coding Patient Monitoring Video," 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, New York, NY, 2006, pp. 5157-5160.

[7] M. Tahir, Z. Ul-Abdin and M. A. Qadir, "Enhancing the HEVC video analyzer for medical diagnostic videos," 2015 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET), Islamabad, 2015, pp. 1-5.

[8] P. P. Shete, D. M. Sarode and S. K. Bose, "Real-time panorama composition for video surveillance using GPU," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Jaipur, 2016, pp. 137-143.

[9] J. C. T. Hai, O. C. Pun and T. W. Haw, "Accelerating video and image processing design for FPGA using HDL coder and simulink," 2015 IEEE Conference on Sustainable Utilization And Development In Engineering and Technology (CSUDET), Selangor, 2015, pp. 1-5.

[10] S. Chen, S. Yu, J. Lu, G. Chen and J. He, "Design and FPGA-Based Realization of a Chaotic Secure Video Communication System," in IEEE Transactions on Circuits and Systems for Video Technology, vol. PP, no. 99, pp. 1-1.

[11] A. Schwenk, L. Thieling, G. Hartung and G. Büchel, "FPGA accelerated video-compression for cloud-based vision sensors," 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, 2015, pp. 1674-1679.

[12] W. Zhou et al., "Real-time implementation of panoramic mosaic camera based on FPGA," 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR), Angkor Wat, 2016, pp. 204-209.

[13] P. Sun, A. Achim, I. Hasler, P. Hill and J. Nunez-Yanez, "Energy efficient video fusion with heterogeneous CPU-FPGA devices," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2016, pp. 1399-1404.

[14] L. Araneda and M. Figueroa, "Real-Time Digital Video Stabilization on an FPGA," 2014 17th Euromicro Conference on Digital System Design, Verona, 2014, pp. 90-97.

[15] M. Hosseini and V. Swaminathan, "Adaptive 360 VR Video Streaming: Divide and Conquer," 2016 IEEE International Symposium on Multimedia (ISM), San Jose, CA, 2016, pp. 107-110.

[16] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, 2007, pp. 225-234.

[17] S. Friston, A. Steed, S. Tilbury and G. Gaydadjiev, "Construction and Evaluation of an Ultra Low Latency Frameless Renderer for VR," in IEEE Transactions on Visualization and Computer Graphics, vol. 22, no. 4, pp. 1377-1386, April 21 2016.