Implementation of Dynamic Frequency Controlled Parallel-Pixel Processing System

T. Roja latha, A.Anitha

Department of ECE, V R Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India

***Abstract*— *The main objective of this work is to develop an effective hardware system that respond to a run-time power constraint. These are handled on FPGAs by Dynamic Frequency Control (DFC) for the management of digital image and video processing architectures. In proposed design, the DFC is handled by utilising minimum resources. The pixel-processor architecture designed here is based on the implementation of single-pixel gamma correction operation. Here, the power and performance in-terms of throughput are constraints of digital image depend on the frequency of operations and number of pixel processing cores. The dynamic frequency controlled parallel-pixel processor is implemented on Virtex-6 FPGA's and parallel-pixel processor architecture is verified by using System Generato*r.

Keywords*— *Dynamic Frequency Control (DFC), Field Programmable Gate Array (FPGA), Parallel-Pixel Processing Architecture based on look-up-tabels(LUTs).

I. INTRODUCTION

Now-a-days most of the image processing applications are provided to meet real-time solutions. The common way of approaching performance on FPGA's is by implementing the parallel architectures. The cases where real-time image processing must also suitable for low power consumption portable applications are vehicle identification, remote sensing applications, mobile phones etc.

1.1 Image Processing Algorithm (Gamma Correction)

The development of the real-time digital processing basically depends upon the image processing algorithms. DIP (Digital image processing) mainly avoids the problem of signal distortion and build-up noise during image processing and DIP is easy for allowing wide-range of algorithms on input images when compared with analog image processing. The image processing algorithms mainly include with gamma correction, histogram equalization, contrast enhancement, thresholding, Huffman table encoding, histogram shaping, and quantization.

Gamma correction: The Gamma correction function is a change in encoded luminance values of an image. Here, the correction of image luminance is done. Gamma correction function is represented by Vout = Vin^ (gamma). If gamma value is >1 gamma expansion and if gamma<1 gamma compression. Gamma correction mainly supports for feature enhancement, intensity correction and thersholding applications. In-corrected gamma causes low contrast, low colour balance and delivery of light levels for an image is improper.

Gamma correction in FPGA can be implemented by using gamma correction logic IPcore which is based on BlockRAMs here the contents can vary on- demand, LUT based approach and other method is using piecewise linear approximation based. In piecewise method, due to precision overflow, the luminance displayed on the device is not exact output luminance. Therefore, to simplify this problem LUT based method is better. In LUT based approach the output pixel values are stored in look-up tables. In proposed system, the distributed memory is used for storing the gamma values which is based on LUT approach. Therefore, the accuracy of the operation increases by increasing the number of input and output bits and the memory space is not the major constraint in LUT based approach.

This paper presents an effective way for varying the frequency of the system at runtime is managed by considering MMCM. The MMCMs inside FPGAs provides a wide range of clock management by adjusting frequency and phase on virtex-6 FPGA. The system for dynamic frequency is evaluated in terms of power, performance in-terms of throughput and resource consumption. Previously, the author in [1] used the frequency control core connected to the Power PC processor which provides the clock for pixel-processor. The Power PC processor peripherals are operated at 100MHz. So, at runtime the frequency can vary up to 100MHz only. The additional processor also increases the resource consumption of the system. Therefore, the present system, for frequency control core provides the clock to the pixel processor by using SM chart for sequential control of the pixel system.

The present work explains the Dynamic frequency control of pixel-processing system which is designed and implemented on virtex-6 FPGA's. The present system provides less utilization of recourses when compared with previous system. The present document as follows Section II describes the literature survey on previous

https://dx.doi.org/10.22161/ijaers *ISSN: 2349-6495(P) | 2456-1908(O)*

system and pixel processor architecture. Section III explains existing system architecture for dynamic frequency control. Section IV explains the implementation of parallel-pixel processing system- using System generator. Section V explains the modified Dynamic frequency controlled parallel-pixel processing system. Section VI discusses on simulation results and comparison of distributed Block ROM LUT based approach and LUT based approach of previous design of pixel processor on power and slices utilised. Section VII has conclusion.

II. LITERARY SURVEY

In earlier literature, the design of pixel processing architecture is based on custom hardware design and LUT-based method. The custom hardware such as precise gamma correction presented in[2] , a contrast enhancement algorithm referred in[3] , an approach for histogram equalization referred in[4] and the architecture for image enhancement by using successive mean quantization transform which is described in[5]. All these architectures utilises less resources for implementation of large input pixel bit-widths and these designs lack the versatility of the LUT based approaches.

Later, the pixel architecture is designed by using gamma correction IP logic core[6] which is based on BlockRAMs and these contents can vary on-demand. In existing system [7] the author implements the pixel processor architecture based on LUT based approach. The pixel output values are stored in look-up tables. The author describes about the pixel processor core implementation and mapping of the LUTs onto Virtex-4 CLB primitives. Here, the pixel system is designed by using FSL and PLB buses. The number of bits for pixel processor is limited to 32bit as the bus width for interfacing is only 32bit width wide. So, the limitation in the pixel architecture is designed for performing parallel pixel operations for four single pixels at time. In proposed system, the distributed ROM LUT based approach is used for implementation of pixel system on Virtex-6 FPGAs. As, this method reduces the number of resources utilised when compared with LUT based approach.

In the study of dynamic frequency control on image processing system, [8] earlier there is architecture DFLAP (dynamic frequency linear array processor) which is used to vary the frequency dynamically from 400MHz to 50MHz. The architecture provides parallelism at array level by using N processing elements (PEs). Here, N represents the size of the image and each PE contains the arithmetic/logic unit, multiplier, shifter, a bidirectional neighbour communication unit, dual port SRAM and a DCU (dynamic clocking unit). The DCU is used for switching the frequency dynamically on each PE. The main drawback is design architecture is complex which contains many PE blocks which are based on image size.

Later [1], the author proposed a dynamic frequency control system which is implemented on pixel processing system. Here, the dynamic reconfiguration and dynamic frequency control is controlled by using a processor. The processor operates the peripherals at 100MHz. So, the maximum frequency that can allow the designer to change dynamically is below 100MHz only this is limitation of the pixel system. Therefore, the proposed system is designed which can utilise less resources by not using any additional processor like soft-core or hardcore. Due to the proposed system the frequency is not limited.

III. Existing System Architecture for Dynamic Control Frequency

The existing system mainly used for controlling the frequency of operation dynamically i.e., at runtime. In this system the FSL and PLB bus interfaces are used for interfacing with processor and parallel-pixel processing system. Here, the DCR (device control register) bus is used for interfacing the processor with frequency control core which provides clock for pixel processor. The processor used here can be either soft-core (Microblaze) or hardcore (Power PC). Here, the internal configuration access port (ICAP) which is used for effective change in input and output bits, number of cores at run-time. The memory is used for storing the input and output bit streams. The main drawbacks of processor are it utilises lot of resources and maximum frequency used by the processor for operating peripherals is 100MHz, so the dynamic frequency is limited to 100MHz or less than processor frequency.

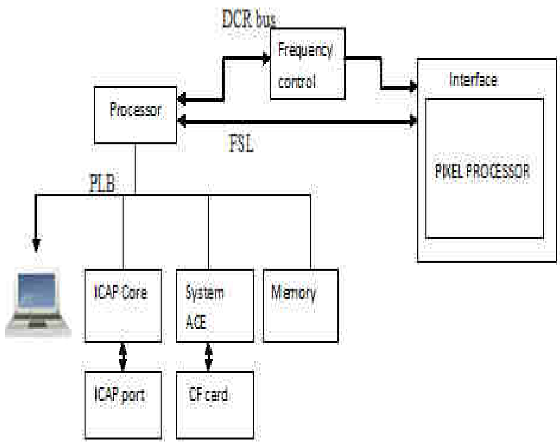
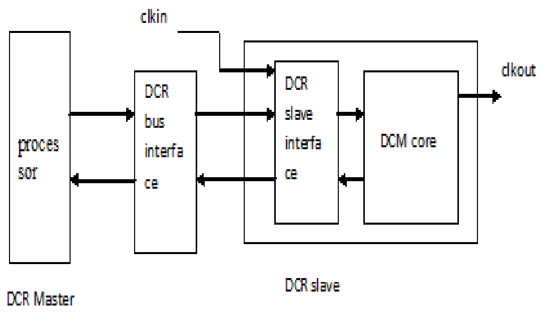


Fig 1: Existing System for Dynamic Frequency Control

3.1 Dynamic frequency control hardware:

In frequency control hardware, the Xilinx virtex-4 DCM is connected to DCR bus via DCR slave interface. Here, the frequency and phase can adjust at runtime without allocating a new bitstream to the FPGA via dynamic reconfiguration port which is register-based. The DCR slave architecture varies for different FPGA families which use different method for loading M and D values. The frequency control can mainly depend upon the modifying the ratio of M to D for providing wide ranges of clock management.



Fig,2: Dynamic Frequency Control Hardware via the DCR Bus Interface

IV. IMPLEMENTATION OF PARALLEL PIXEL PROCESSING SYSTEM- USING SYSTEM GENERATOR

To meet the real-time applications, image processing should be implemented on hardware. System generator is used for performing the hardware implementation by using two software tools to be configured. One is Matlab/Simulink and other is Xilinx ISE. The System Generator is part of the ISE design suite which is used for verification of parallel-pixel processing by using Xilinx blocks.

The design flow of hardware implementation of parallel-pixel processing system is shown in Fig[3]. In the parallel-pixel processing system, image source and image viewer are used as input and output for an image. Here, image source is used for sending the image as input and image viewer is used for viewing the output image. Image pre-processing and image post-processing are used for providing the input and output for Xilinx block sets which are designed by using Simulink blocksets. The Xilinx black box consists of parallel-pixel processing system design which has to implemented by using Xilinx System Generator.

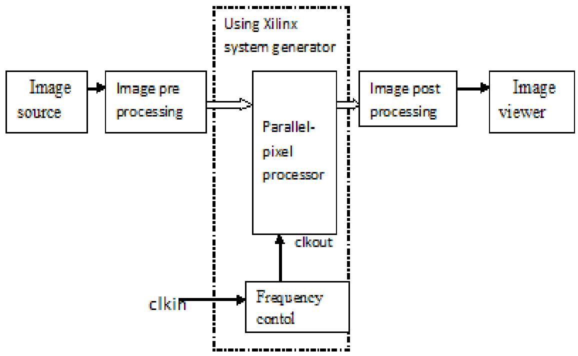


Fig.3: Implementation of Parallel-Pixel Processor-Using System Generator

4.1 Design of Parallel- Pixel Processor Architecture in System Generator:

The parallel-pixel processor architecture is show in Fig[4]. In system generator there are no existing simulink block sets for processing the pixels inputs parallel. So, pixel processor used here is connected with shifters black box which is designed by using Xilinx block sets. The Xilinx black box is used for implementation of shifter and parallel-pixel processor code. The shifters are used for sending and receiving the pixel inputs and outputs parallel. Here, shifters and parallel pixel-processor are operated at different frequencies.

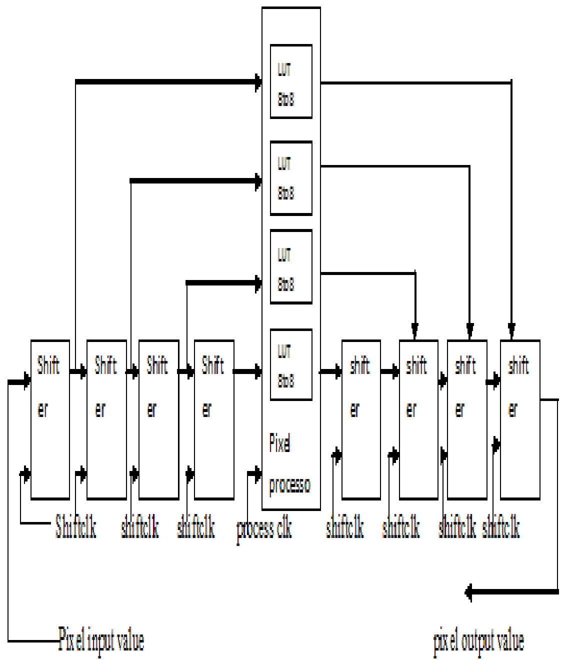


Fig.4: In System Generator Parallel-Pixel Processor Architecture

The shifter frequency depends upon the parallel-pixel processing distributed ROM LUT cores. For example, the

https://dx.doi.org/10.22161/ijaers *ISSN: 2349-6495(P) | 2456-1908(O)*

number of cores of parallel-pixel processing system is four than the frequency of the shifter is multiplied by four times of the parallel-pixel processor clock frequency. In, this way the clock is provided to the shifters. If N cores are presented than shifters clock frequency is parallel-pixel processing frequency multiplied by N.

V. MODIFIED DYNAMIC FREQUENCY CONTROL PARALLEL-PIXEL PROCESSING SYSTEM

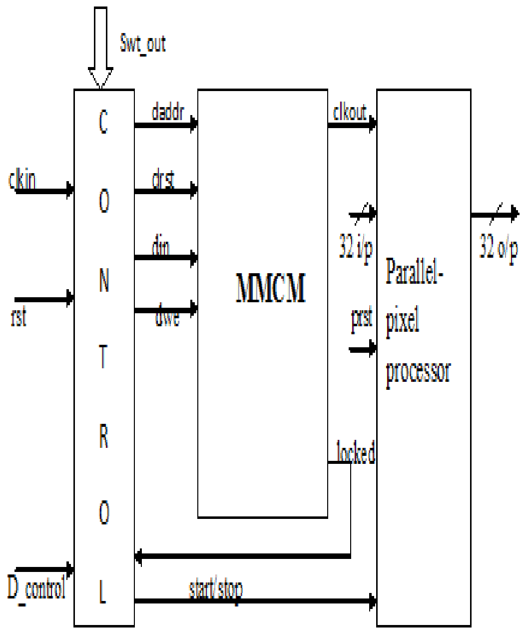


Fig.5: Modified Dynamic Frequency Control Parallel-Pixel Processing System

In this section, the overall system architecture is seen in figure [5]. The modified Dynamic frequency control Parallel-pixel processing system consists of MMCM(Mixed Mode Clock Manager for virtex-6) for dynamic frequency control and parallel-pixel processing system which are controlled by using a controller. The inputs of controller are clock-in, reset, dcontrol is for dynamic control for MMCM, swt_out is used for assigning the particular frequency to vary dynamically. The outputs of controller are daddr, drst, din and dwe. These are the signals which are used for changing the frequency at runtime. The locked signal from MMCM acts as input for controller for activating the parallel-pixel processor to work at particular frequency. The state diagram for dynamic frequency control parallel-pixel processing system for understanding the sequential operation of the system.

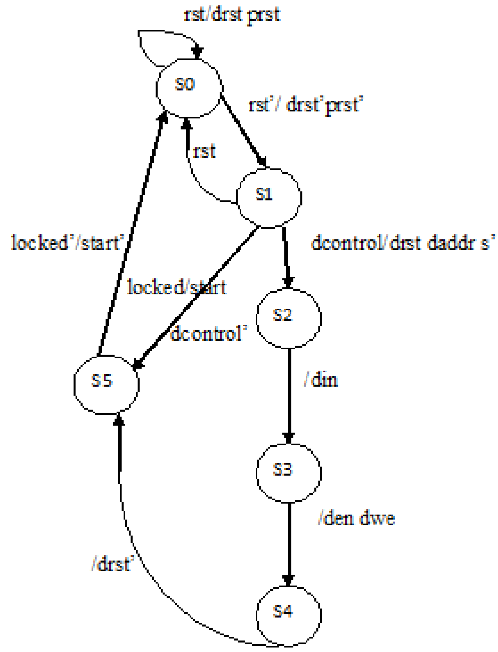


Fig.6: State Machine for Modified Dynamic Frequency Control Parallel-Pixel Processing System

5.1 The Operation of Dynamic Frequency Control Parallel-Pixel Processing System:

When dcontrol signal is made high, the frequency is dynamically controlled varying the ratio of the M, D and O parameters which are assigned by using switch out signal. The relationship of clock frequency is Fclkout= M/(D*O) *Fclkin. The value of M corresponds to the CLKFBOUT_MULT_F setting, the value of D to the DIVCLK_DIVIDE, and O to the CLKOUT_DIVIDE, Fclkin is input clock frequency. These values are given to MMCM for varying the frequency at that time pixel processor start signal set low. Once the locked signal is set high, than pixel processor start signal is made high and the parallel-pixel operation is performed.

5.2 Pixel Processor Core Architecture:

The architecture is implemented by using the virtex-6 LUTs which is used for mapping on CLBs efficiently. Here, the higher number of bits on LUTs is designed by combining the LUT primitives and multiplexers. The 8 input bits are used for implementing the pixel operation of each output bit. The distributed ROM LUT based approach is used for designing the pixel processor because the utility of resources are less when compared with LUT based approach.

https://dx.doi.org/10.22161/ijaers *ISSN: 2349-6495(P) | 2456-1908(O)*

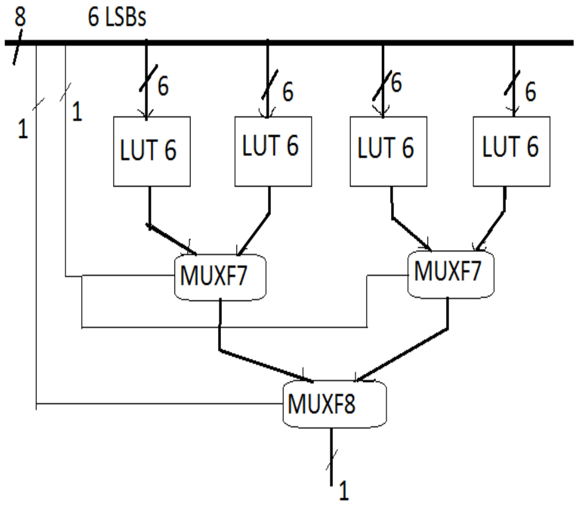


Fig.7: Pixel Processor Architecture For Implementation of Virtex-6 Lut8-To-1

VI. SIMULATION RESULTS

In this section, the dynamic frequency control parallel-pixel processing is implemented and tested by using Xilinx ISE simulator and by using Xilinx System Generator.

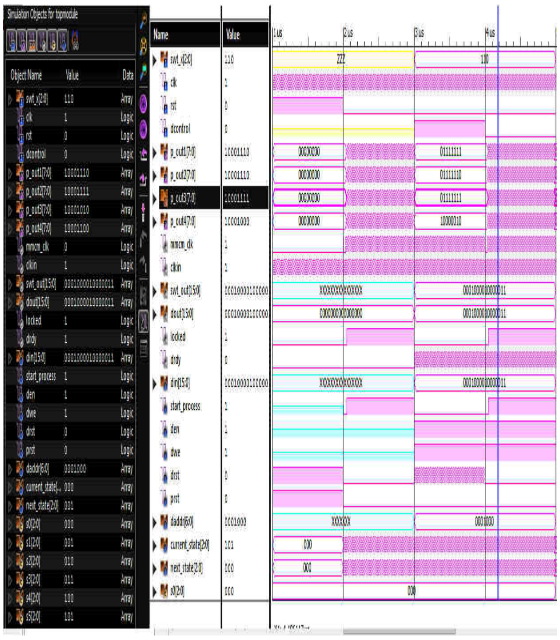


Fig.9: Simulation Result of Dynamic Frequency Control Parallel-Pixel Processing System

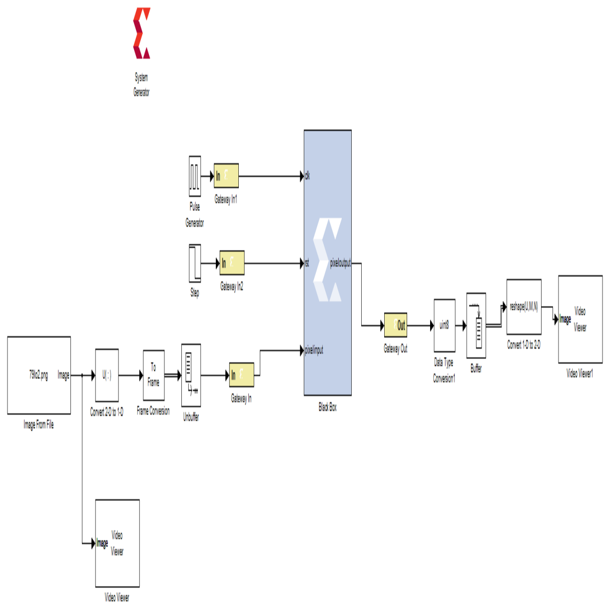


Fig.8: Parallel-Pixel Processor Architecture Simulation Block Diagram

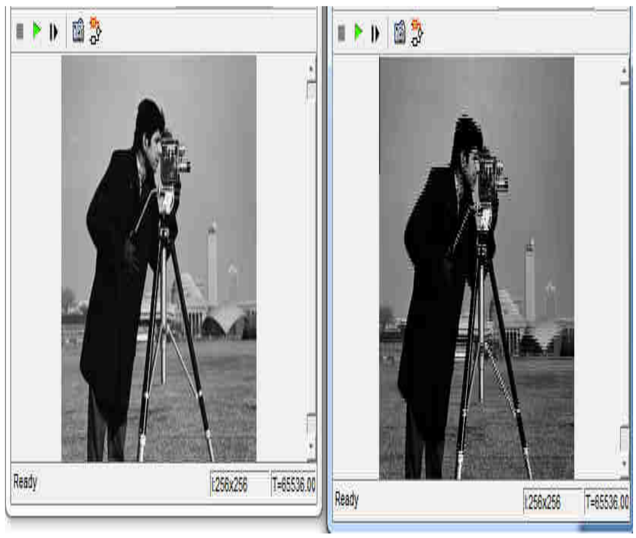


Fig.9: Simulation Results of Input and Output of Parallel-Pixel Processor Architecture

The power consumption and resources utility of parallel-pixel processor architecture is less for Distributed ROM LUT based approach when compared with LUT based approach in [1] of previous system architecture. These are shown in given table.

	LUT based approach	Distributed ROM LUT based approach
No. of slices	320	40
No. of LUTs	576	115

Power(mW)	24	1.05
Maximum Frequency control on the system(MHz)	100	600
Number of cores supports by the system	4	-

The table shown below provides the power consumed and resource utilized by the dynamic frequency control Parallel-pixel processing system.

	No. of slices	No. of LUTs	Power (mW)	Maximum Frequency control on the system(MHz)
Dynamic frequency control parallel-pixel processing system	81	127	68.5	600

VII. CONCLUSION

An effective implementation of dynamic frequency control parallel-pixel processing system is presented. This paper presents an effective parallel-pixel processor for reducing power consumption and can vary the frequency at runtime by utilising fewer resources. This architecture is implemented by using Xilinx Virtex-6 FPGAs and verified by using System Generator.

REFERENCES

[1] D. Llamocca, M. Pattichis, “A dynamically reconfigurable pixel processing system based on power/energy-performance-accuracy optimization,” in Circuits and systems for video technology, IEEE Transactions vol.23 No. 3, March 2013, pp. 488-502.

[2] D.-U. Lee, R. C. C. Cheung, and J. D. Villasenor, “A flexible architecture for precise gamma correction,” IEEE Trans. Very Large Scale Integr. Syst., vol. 15, no. 4, pp. 474–478, Apr. 2007.

[3] W. Bing-Jian, L. Shang-Qian, L. Qing, and Z. Hui-Xin, “A real-time contrast enhancement algorithm for infrared images based on plateau histogram,” Infrared Phys. Technol., vol. 48, no. 1, pp. 77–82, Jul.2005.

[4] A. M. Alsuwailem and S. A. Alshebeili, “A new approach for real-time histogram equalization using FPGA,” in Proc. Int. Symp. Intell. Signal Process. Commun. Syst., Dec. 2005, pp. 397–400.

[5] M. Chandrashekar, U. N. Kumar, K. S. Readdy, and K. N. Raju, “FPGA implementation of high speed infrared image enhancement,” Int. J. Electron. Eng. Res., vol. 1, no. 3, pp. 1480–1485, 2002.

[6] LogiCORE IP Gamma Correction (DS719), v3.0 ed., Xilinx, Inc., San Jose, CA, Sep. 2010.

[7] D. Llamocca, M. Pattichis, and A. Vera, “A dynamically reconfigurable parallel pixel processing system,” in Proc. Int. Conf. Field Programmable Logic Applicat., Sep. 2009, pp. 462–466.

[8] N. Ranganathan Naveen, B havanishankar, N. Vijaykrishnan “A Dynamic Frequency Linear Array Processor for Image Processing,” Pattern Recognition, 1996., Proceedings of the 13th International Conference, August 2002.