

General Structure Design for Fast Image Processing Algorithms Based upon FPGA DSP Slice

Prof. Pranali R. Dandekar
Assistant Professor,
Dept. of Computer Application
RCOEM, Nagpur
Pranalimeshram@gmail.com

Prof. Punam Khandar
Assistant Professor,
Dept. of Computer Application
RCOEM, Nagpur
punam.khandar@gmail.com

Prof. Mohini Upasani
Assistant Professor,
Dept. of Computer Application
RCOEM, Nagpur
upasanims@gmail.com

Abstract— Increasing the speed and accuracy for a fast image processing algorithms during computing the image intensity for low level 3x3 algorithms with different kernel but having the same parallel calculation method is our target to achieve in this paper. FPGA is one of the fastest embedded systems that can be used for implementing the fast image processing image algorithms by using DSP slice module inside the FPGA. we aimed to get the advantage of the DSP slice as a faster, accurate, higher number of bits in calculations and different calculated equation manoeuvre capabilities. Using a higher number of bits during algorithm calculations will lead to a higher accuracy compared with using the same image algorithm calculations with less number of bits, also reducing FPGA resources as minimum as we can and according to algorithm calculations needs is a very important goal to achieve. So in the recommended design we used as minimum DSP slice as we can and as a benefit of using DSP slice is higher calculations accuracy as the DSP capabilities of having 48 bit accuracy in addition and 18 x 18 bit accuracy in multiplication. For proofing the design, Gaussian filter and Sobelx edge detector image processing algorithms have been chosen to be implemented. Also we made a comparison with another design for proofing the improvements of the accuracy and speed of calculations, the other design as will be mentioned later on this paper is using maximum 12 bit accuracy in adding or multiplying calculations.

Keywords— Fast Image processing; DSP slice; Embedded systems

I. INTRODUCTION

Utilization an embedded system for a computer vision application, there is currently the choice between either using Digital Signal Processors (DSPs) or Field Programmable Gate Arrays different vendors. The design considerations for FPGA multiplication units, numerous logic elements, parallel hardware structures, handling of high data rates and the reconfigure-ability of FPGAs. Compared to high end DSPs, FPGAs are more expensive,

the computations is slower than on DSPs because of the higher clock frequency of DSPs. [10] DSP is a class of hardware devices that fall somewhere between an ASIC and a PC in terms of the performance and the design complexity. FPGA is mainly used for computationally demanding functions like convolution filters, motion estimators, two-dimensional Discrete Cosine Transforms (2D DCTs) and Fast Fourier Transforms (FFTs) all are better optimized when targeted on FPGAs [6,7].

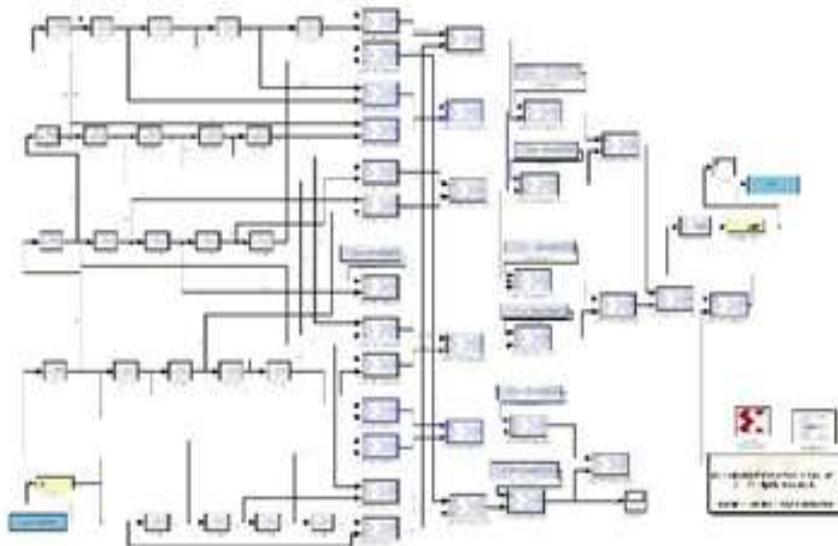


Figure 1. Proposed structure for Gaussian 5x5 filter using DSP48 slice in Xilinx Vertex 4 FPGA

In an application that requires real-time processing, like video or television signal processing or real-time trajectory

generation of a robotic manipulator, the specifications are very strict and are better met when implemented in hardware

[3-5]. Features like embedded hardware multipliers, increased number of memory blocks and system-on-a-chip integration enable video applications in FPGAs that can

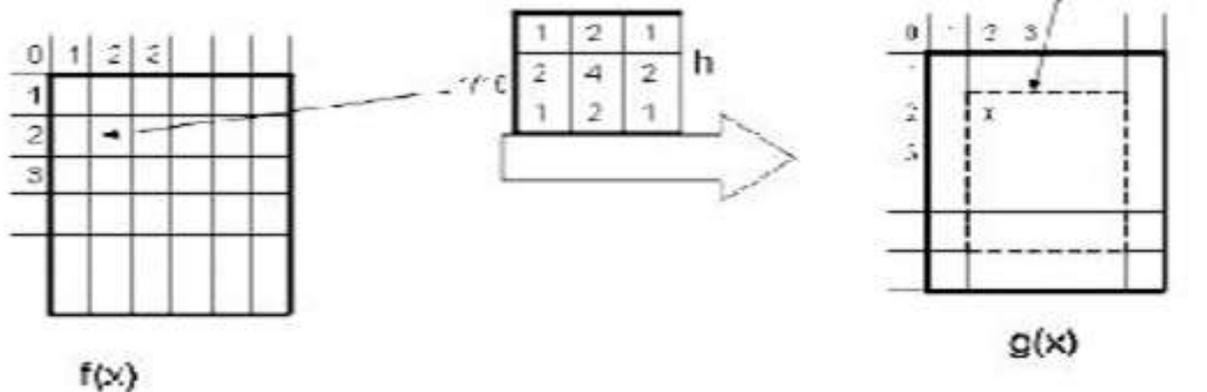
II. IMAGE PROCESSING ALGORITHMS

Low-level Vision 3x3 Gaussian pyramid Algorithm [4,9]: two dimensional low-pass filters, such as the Gaussian low-pass filter, work with a filter kernel, calculate an average

outperform conventional DSP designs [2,8]. Concluding all of best features in a design is not an easy job.

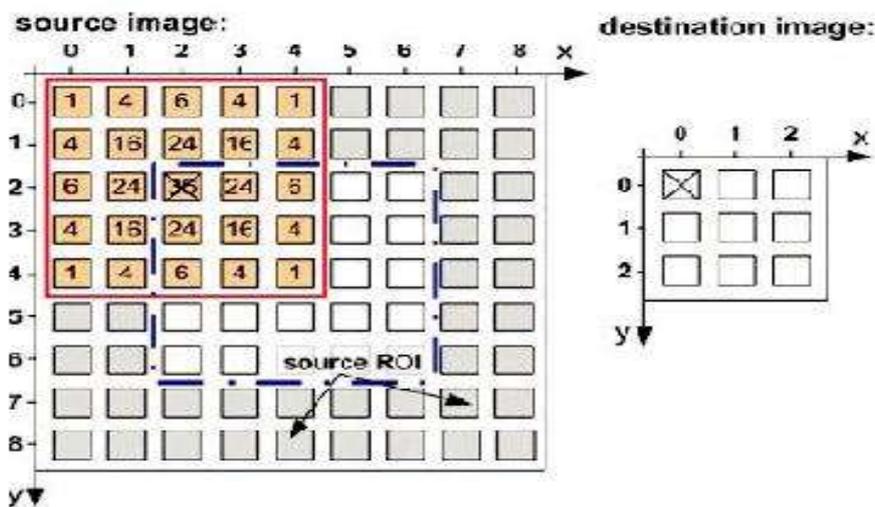
value for a destination pixel using a number of neighboring source pixels. The two dimensional Gaussian filter is shown in following figure 2

Gaussian Filter 3x3 Low level algorithm



When dealing with digital images integer weighting factors are used. A typical 3x3 Gaussian filter matrix and the decimation of the pixels are shown in Figure 2. The anchor point of the Gaussian filter kernel is marked with an "X". Obviously for every calculated kernel pixel one neighboring pixels in both dimensions are required. Therefore, this function uses a Region of Interest (ROI). With every Gaussian pyramid level the number of pixels in x- and y-coordinates is reduced by a factor of 1. According to Gaussian calculation method for using the neighbor pixels window 3x3 or 5x5 to get the new pixel data, taking into consideration that their differences in their kernel's size. Taking into consideration that; expressing the kernel operation in each frame process can be decomposed into block processing mode, and this block has the same

processing function and the number and size determine degree of algorithm parallelism. The degree of parallelism for one (256x256) image frame is 256x3 or 256x5 delay element blocks respectively with. Computing time is determined by number of machine cycle to get the first frame pixel plus number of a the whole frame pixels calculated in the pipe line, total number of machine cycles = $9 + 256*256 = 65,545$ machine cycle. Working with frequency 100MHz (10nSec) we found that time = $10nSec * 65,545 = 0.6554$ mSec almost 0.66 mSec for single frame. so from that; we defined a new general structure design of fast image processing as we will explain later. This process and technique for Gaussian 3x3 and Gaussian 5x5 will have the same speed of calculations versus using a little more of FPGA utilization resources.



Design General Structure Techniques

III. USE STANDARD COMPONENTS

Map designs to Mult and AddSub blocks or use higher level IP such as the MACFIR filter generator blocks. This

IV. USE SYNTHESIZABLE BLOCKS

Structure the design to map onto the DSP48's internal architecture and compose the design from synthesizable Mult, AddSub, Mux and Delay blocks. This approach relies on logic synthesis to infer DSP48 blocks where appropriate. This approach gives the compiler the most freedom and can often achieve full-rate performance.

approach is useful if the design needs to be compatible with V2P or S3 devices or uses a lower-speed clock and the mapping to DSP48s is not required.

V. USE DSP48 BLOCKS

Use System Generator's DSP48 and DSP48 Macro blocks to directly implement DSP48-based designs. This is the highest performance design technique. Be aware however that obtaining maximum performance and minimum area for designs using DSP48s may require careful mapping of the target algorithm to the DSP48's internal architecture, as well as the physical planning of the design [15].

VI. DELAY LINE

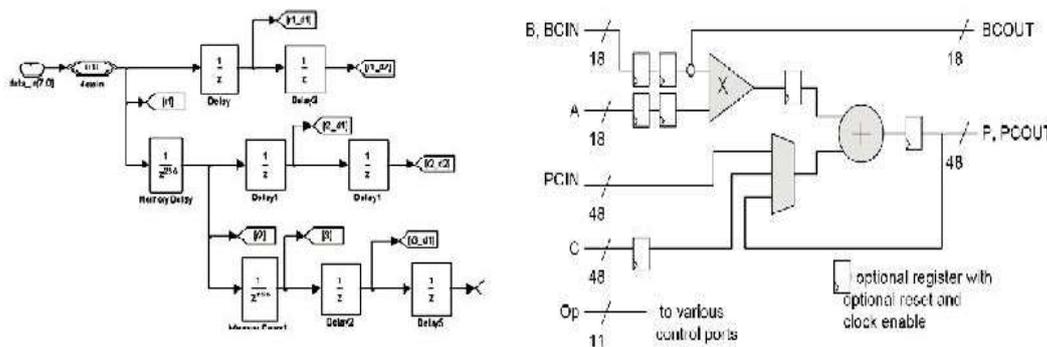


Figure4.Line buffer delay line for image size 256x256

Incoming pixels are processed by means of a 2D filter convolution kernel, working on the grayscale intensities of each pixel's neighbors in a 3x3 or 5x5 regions. Image lines are buffered through delay-lines producing primitive 256x3 or 256x5 cells where the filter kernel size applies. The line buffering principle is shown in Figure 4. A z⁻¹ delay block produces a neighboring pixel in the same scan line, while a z⁻²⁵⁶ delay blocks produces the neighboring pixel in the previous image scan line. We assume image size of 256x256 pixels.

If a change in frame size is required we need to redesign the structure, the number of delay blocks depends on the size of the convolution kernel, while delay line depth depends on the number of pixels in each line. Each incoming pixel is at the center of the mask and the line buffers produce the neighboring pixels in adjacent rows and columns [18].

This block is listed in Xilinx Blockset libraries; The System Generator DSP48 Macro block provides the blocks DSP48, DSP48A, and DSP48E. Using this block instead of using a technology-specific DSP slice helps makes the design more portable between Xilinx technologies. Depending on the target technology specified at compile time, the block wraps one DSP48/DSP48A/DSP48E block along with reinterpret and convert blocks for data type alignment, multiplexers to handle multiple opmodes and inputs, and registers [15].

The simple model using DSP48 Macro which has three inputs defined as X_o, Y_o, and Z_o. Because more than one

Instruction opmode can be specified in block dialog box, the Sel input port is automatically added:

The general case is : $P = C + A * B$

When this design is compiled, if the target technology is Virtex-4, then a DSP48 slice will be netlisted. If Virtex-5 is specified, then a DSP48E slice will be netlisted, and if the Spartan-3A DSP technology is specified, then a DSP48A slice will be used in the implementation [17].

In our design we used DSP slice which has a 48 bit Adder and 18 bit by 18 bit Multiplier accuracy, higher number of bits gives more accurate result in calculations as a it is associated with higher number of bits used. Also each module can work as adder or multiplier or adder and multiplier at the same time. The great benefit from using macro DSP48 that pre-implemented equations can be saved in each module and can be selected accordingly to select which equation will be activated at certain pin selected.

VII. CALCULATION PROCEDURES

For calculating one pixel data value in the Gaussian 3x3 filter it needs 9 inputs to get one output, similar for Gaussian filter 5x5 it needs 25 inputs to get one output. The signal flow are divided into levels of calculations, these levels defined as each electronic element will cause delay in the signal calculation flow will be consider as a level, the delay line elements are an obligation for certain frame size and it will be taking into consideration one time for calculating the first pixel data as an overall one time delay, then data will flows consequently. By using image size 256x256 and

100MHz frequency (10nsec /per machine cycle) the calculation time will be calculated as the following:

Time for frame = (number of first in first out machine cycle + total number of machine cycle per frame) multiply by time for a single machine cycle.

$$T_{\text{frame}(3 \times 3)} = (9 + 256 \times 256) \times 10 \text{ n Sec} \\ = 0.65545 \text{ m Sec} \approx 0.66 \text{ m Sec}$$

$$\text{For calculating the } 5 \times 5 \text{ Gaussian kernel } T_{\text{frame}(5 \times 5)} = (11 + 256 \times 256) \times 10 \text{ n Sec} \\ = 0.65547 \text{ m Sec} \approx 0.66 \text{ m Sec}$$

The advantage of our design is that both calculations take almost similar short time. Comparing with calculation method using C language for DSP implementation we can find that, Calculating 3x3 Gaussian filter by identifying how many variables should be added together and how many weighs values need to be multiplied as well:

3x3 gives has 9 inputs variables need to be added and 3 different weights (1,2,4) need to be multiplied, so we calculate the output pixel brightness as follows:

Time for Eight addition process plus time for 3 multiplication processes plus time taken for the first

calculated pixel. 5x5 gives has 25 inputs variables need to be added and 6 different weights (1,4,6,16,24,36) need to be multiplied, so we calculate the output pixel brightness as follows:

Time for twenty four addition process plus time for six multiplication processes plus time taken for the first calculated pixel System Performance Evaluation Integrated Software Environment (ISE 10.1) software for FPGA and Matlab Math-works (R2007a) software both were used for design, validate and simulate our general structure fast image design, the FPGA embedded system used was from Xilinx Company product Virtex4, XC4VSX55-12FF1148.

FPGA resources consummated due to our proposed design in Xilinx Virtex 4 number XC4VSX55-12FF1148, that DSP48 slices is 2% in Gaussian kernel size 3x3 and 5% in Gaussian kernel size 5x5. Also umber of number of Digital Clock Manager (DCM) and other resources almost the same. Finally The main target is to see the both low level different kernel algorithm deals with real image with the size of 256x256 in figure 6-a and the output image result from applying Gaussian filter 3x3 shown in figure 6-b and Gaussian kernel 5x5 result image in 6-c.

A Original image b- Gaussian Filter 3x3 c- Gaussian Filter 5x5
Figure6.tput images from the proposed Gaussian general structure filter 3x3 and 5x5 design respectively



CONCLUSION

The great advantage in our design implementation on FPGA is that whatever the kernel size and more usage of DSP48 slice the time difference is slightly change not the same as if we were using C language Programming technique, designing fast image algorithm main concerns is accuracy and reducing the time as minimum as possible which we applied the high accuracy by using DSP slice 18bit by 18 bit in multiplication and 48 bit in addition, in the other hand by using the implementation target is an FPGA instead of DSP we reduce the total image frame processing time.

REFERENCE

- [1] D. V. Rao, S. Patil, N.A. Babu, V. Muthukumar, Implementation and evaluation of image processing algorithms on reconfigurable architecture using C-based Hardware Descriptive Languages, International Journal of Theoretical and Applied Computer Sciences, Vol. 1 No. 1, P .9-34, 2006
- [2] R.J. Petersen, B.L. Hutchings, An assessment of the suitability of FPGA-based systems for use in digital signal processing,

- 5th International Workshop on Field-Programmable Logic and Applications, Oxford, England, P. 293-302, August 1995.
- [3] B.A. Draper, J.R. Beveridge, A.P.W. Bohm, Ch. Ross, M. Chawath, Accelerated image processing on FPGAs, IEEE Transactions on Image Processing , December 2003.
- [4] R. C. Gonzalez and R. E. Woods. Digital Image Processing, Second Edition. Pearson Education International, 2002.
- [5] M. Leeser, S. Miller, H. Yu, Smart camera based on reconfigurable hardware enables diverse real time applications,
- [6] M. Rogers, M. Won, A. Soohoo, Altera FPGA co-processors accelerate the performance of 3-D stereo image processing, Altera Corporation, News & Views Spring/Summer 2005.
- [7] D. Baumgartner, P. Rossler, W. Kubinger, Performance Benchmark of DSP and FPGA Implementations of low level vision algorithms, IEEE, 2007
- [8] I.S. Koc. Design considerations for real-time systems with dsp and risc architectures. 13th European Signal Processing tware FPGA based system for fast image processing, Microprocessors and Microsystems, P. 95-106, 2008.