

An Efficient VLSI Architecture for 2-D Convolution with Quadrant Symmetric Kernels

Ming Z. Zhang, Hau T. Ngo, Adam R. Livingston and Vijayan K. Asari
*Department of Electrical and Computer Engineering
 Old Dominion University, Norfolk, VA 23529
 {mzhan002, hngo001, alivi001, vasari}@odu.edu*

Abstract

A high performance digital architecture for computing 2-D convolution utilizing the quadrant symmetry of the kernels is proposed in this paper. Pixels in the four quadrants of the kernel region with respect to an image pixel are considered simultaneously for computing the partial products of the convolution sum. A novel data handling strategy to identify the pixels to be fed to different processing elements helps reducing the data storage requirements in the circuitry. The new design results in 75% reduction in multipliers and 50% reduction in adders when compared with the conventional systolic architecture. The proposed architecture design is capable of performing convolution operations with 14×14 kernel at a rate of 57 1024×1024 frames per second in a Xilinx's Virtex 2v2000ff896-4 FPGA.

Keywords: 2-D convolution, symmetric kernel, pipelined architecture, systolic architecture.

1. Introduction

Two dimensional convolution is one of the most frequently used operations in many image and video processing applications. It is necessary to have optimal design to reduce hardware resources, power consumption and to increase high speed operation for performing 2-D convolution effectively. 2-D convolution is defined as:

$$O(m,n) = \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} W(i,j) \cdot I\left(m+i-\frac{K}{2}, n+j-\frac{L}{2}+1\right) \quad (1)$$

where $W(i, j)$ are the weights of a $K \times L$ kernel, $I(m, n)$, $O(m, n)$ are the $M \times N$ input and output images, respectively. Some existing designs [1-7] make use of arbitrary coefficients for the kernels; however, we take advantage of popular kernels with symmetric structure and optimize to greater extend compared to conventional approaches. In this paper, we present a high performance systolic architecture with pipelined delay lines to achieve high throughput rate.

2. Real Time 2-D Convolution Design with Quadrant Symmetry

Figure 1 illustrates a specialized architecture with quadrant symmetric property such as Gaussian filters. The architecture design is based on decomposition of (1) into (2) and (3) respectively for odd and even dimensional kernels.

$$O(m,n) = \sum_{i=0}^{\frac{K-1}{2}} \sum_{j=0}^{\frac{L-1}{2}} W(i,j) \cdot I\left(m+i+\frac{K}{2}, n+j+\frac{L}{2}\right) + W\left(\frac{K}{2}, \frac{L}{2}\right) \cdot I(m,n) \quad (2)$$

$$O(m,n) = \sum_{i=0}^{\frac{K-1}{2}} \sum_{j=0}^{\frac{L-1}{2}} W(i,j) \left[I\left(m+i-\frac{K}{2}+1, n+j-\frac{L}{2}+1\right) + I\left(m-i+\frac{K}{2}, n+j-\frac{L}{2}+1\right) + I\left(m+i-\frac{K}{2}+1, n-j+\frac{L}{2}\right) + I\left(m-i+\frac{K}{2}, n-j+\frac{L}{2}\right) \right] \quad (3)$$

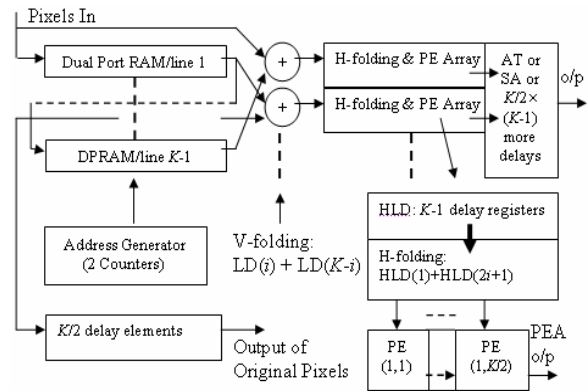


Figure 1: Block diagram of the architecture for 2-D convolution with quadrant symmetric kernels

The line delays (LDs), which are implemented with Dual Port RAMs (DPRAMs) to allow dynamic change of image sizes, are at the front end of

processing elements (PEs) to optimize storage requirements of delay elements. Each PE computes partial result and passes this result to next PE where it is summed with the partial result computed in this current PE. As the pixels come in raster scan fashion through the LDs, vertical folding takes place by adding pixels on vertical symmetry. This folding strategy reduces the number of adders and multipliers by half. The results from vertical folding are then fed into $K/2$ horizontal delay lines of $K-1$ delay elements. The architecture takes account of appropriate delays so the results from horizontal folding are passed to consecutive PEs correctly. Horizontal folding further reduces the multipliers in half. At the end of PE arrays (PEAs), the partial results are summed with adder tree to produce 2-D convolution output at the rate of one per cycle.

The original pixel values are also available with $K/2$ more delays from the center node of the LDs. It is very helpful that the original pixel be available in synchronization with the output from 2-D convolution of quadrant symmetric kernels. This synchronization permits the subsequent processing elements to perform their operation at run time. It also eliminates the bottleneck of memory accesses where the pixels can be read in without buffering the entire frame.

3. Implementation/Hardware Utilization

The simulation of the proposed architecture design is performed using Xilinx's ISE software. The FPGA chip targeted for our implementation is a Virtex 2 2v2000ff896-4 FPGA. Gaussian filter with standard deviation of five, in conjunction with 24×24 images is used in verification of our design with the data produced from MATLAB software. Table 1 shows the hardware utilization for the quadrant symmetric architecture.

Table 1. Hardware utilization

Kernel	CLB	FF	BRAM	Clock Rate
10×10	695	990	9	60 MHz
12×12	975	1407	11	60 MHz
14×14	1303	1900	13	60 MHz
*24×24	3664	5507	23	59 MHz

*Compiled on XC2V8000-4

For 14×14 square kernel, it consumes 12% of Configurable Logic Blocks (CLBs), 8% of flip-flops (FFs), 9 embedded Block RAM modules (BRAMs). This kernel with maximum operating frequency, which is mainly limited by speed of 18-bit multipliers, gives

maximum throughput of 57 frames per second for 1024×1024 video frames. The architecture presented in this paper achieves significant hardware efficiency by reducing 75% of multipliers and nearly 50% of adders.

4. Conclusion

In this paper, we presented a flexible and very high performance architecture for 2-D convolution of quadrant symmetric filters with significant reduction of storage requirements compared to conventional designs. The proposed design reduces 75% of the multipliers and nearly 50% of the adders required to perform 2-D convolution operations. The proposed design also provides a minimum set of delay elements to pass the original pixel to the output in synchronization with the convolution results. The architecture also allows dynamic change of image size with the constraint of maximum image size. With a XC2V2000-4 FPGA, this design can achieve maximum of 57 1024×1024 frames per second.

5. References

- [1] H. T. Kung and S. W. Song, "A systolic 2-D Convolution Chip," in *Proc. IEEE Comput. Soc. Workshop Comput. Architecture for Pattern Anal. And Image Database Management*, pp. 159-160, 1981.
- [2] H. T. Kung, L. M. Ruane, and D. W. L. Yen, "A two-level pipelined systolic array for multidimensional convolution," *Image and Vision Computing*, vol. 1, no. 1, pp. 30-36, Feb. 1983.
- [3] A. Wong, "A New Scalable Systolic Array Processor Architecture For Discrete Convolution", *Master Thesis*, University of Kentucky, 2003.
- [4] H. M. Chang and M. H. Sunwoo, "An Efficient Programmable 2-D Convolver Chip," in *Proc. Of the 1998 IEEE International Symposium on Circuits and Systems, ISCAS, Part 2*, pp. 429-432, May 1998.
- [5] V. Hecht, K. Ronner and P. Pirsch, "An Advanced Programmable 2-D Convolution for Real Time Image Processing," in *Proc. Of IEEE International Symposium on Circuits and Systems*, pp. 1897-1900, 1991.
- [6] J. H. Kim and W. E. Alexander, "A Multiprocessor Architecture for 2-D Digital Filters," *IEEE Trans. On Comput.*, vol. C-36, pp. 876-884, July 1987.
- [7] A. E. Nelson, "Implementation of Image Processing Algorithms On FPGA Hardware", *Master Thesis*, University of Tennessee, May 2000.
- [8] B. Bosi and G. Bois, "Reconfigurable Pipelined 2-D Convolver for fast Digital Signal Processing," *IEEE Trans. On Very Large Scale Systems*, vol 7, no. 3, pp. 299-308, Sept. 1999.