

International Journal of Engineering Sciences & Research Technology

(A Peer Reviewed Online Journal)
Impact Factor: 5.164



Chief Editor
Dr. J.B. Helonde

Executive Editor
Mr. Somil Mayur Shah

ABSTRACT

The Digital Image Processing convolution is core block for Convolution Neural Networks (CNN) which is used in Deep CNNs and is used for advanced applications of feature extraction, image recognition etc. This paper introduces 2 novel hardware architectures for convolution process one of them is hardware Quad parallelized architecture. Performance comparison is done and I proved that parallelization is highly useful. This parallel one can speed up the process of convolution filtered image data transmission through telemedicine communication network etc. Implementation is done for 64 by 64 size matrix, 3 by 3 Kernel using Xilinx Vivado 2015.2 tool on Xilinx Artix-7 FPGAs using Verilog HDL.

KEYWORDS: Convolution; FPGAs; Kernel, Quad parallelized architecture.

1. INTRODUCTION

In digital image processing Convolution is one of the spatial domain techniques in which manipulating pixels is done between image and a 2D kernel of 3 by 3 or 5 by 5 size, which results in de-noised image, edge detected image, blurred image etc. the values in the kernel can be integers or can be integer kernel multiplied with a common fraction outside the kernel and can be customized depending on the requirement of the application.

In one of my research topics, I found that convolution filter is performing better than all the filters when it is going to higher levels of noise to the images [1]. It is found to be efficient by itself and also when combined with Translation Invariance (TI) algorithm [1]. So, I have got motivated to continue working on Digital Image Processing Convolution. The further motivation is due to its applications in Deep Convolution Neural Networks in the current booming era of Artificial Intelligence. So I have got motivated in developing its hardware.

In this research I have developed 2 different novel hardware architectures for convolution process, nowhere in the literature, and first of their kind [2]. This research is an extension to my research in [2] and to verify for *higher important and standard size of 64*. The first one is completely sequential process of generating convolution results, and the second one is for improving the speed of the same architecture by parallelizing the process [2]. This hardware Quad parallelized architecture improved the speed of the convolution process. The programming is done using Verilog HDL and implemented on Xilinx Artix-7 FPGAs using Vivado 2015.2 Tool.

In section 2 architectures are explained, the section 3 discusses the implementation results and section 5 concludes the research, useful applications are noted in section 4.

2. NOVEL ARCHITECTURES

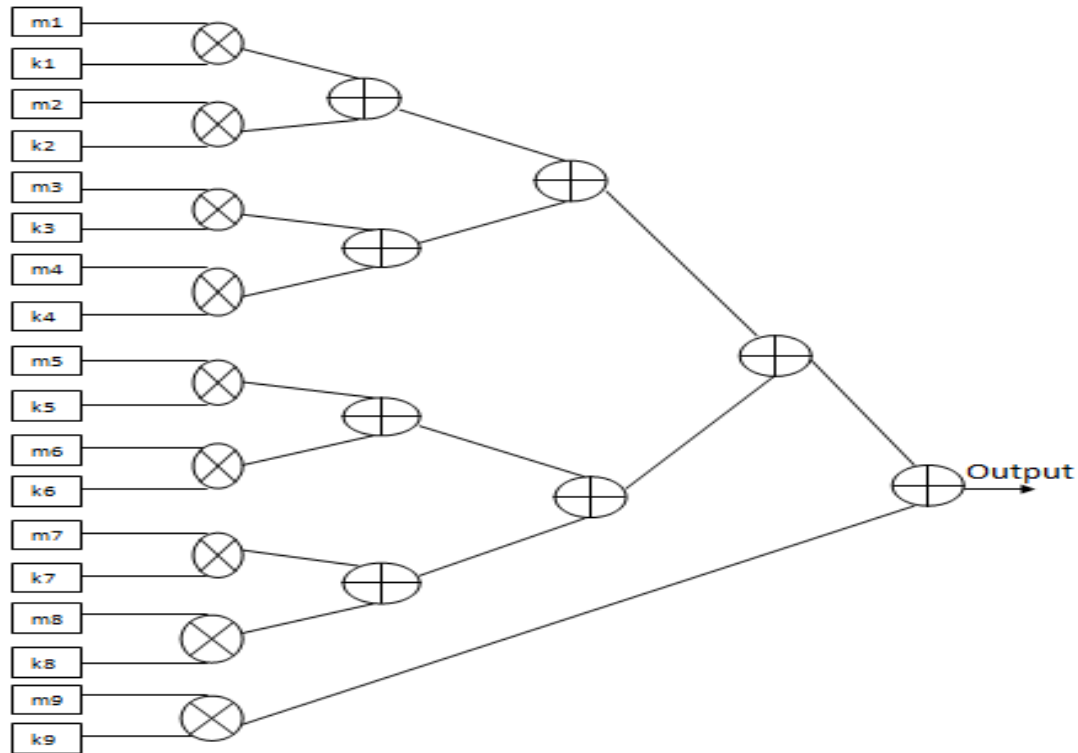


Figure 1 Convolution Result Hardware (CRH) Structure

The main unit in this architecture is the hardware which generates Convolution Result and I named it as Convolution Result Hardware (CRH). Its tree like structure is given as in Figure 1. The inputs m_1 and so on are part of the image pixels on which Kernel is overlapped, and k_1 and so on are the kernel. So It can be seen CRH unit as 9 paired inputs and one output black box. In this \otimes symbol is for multiplication and $+$ symbol is for addition required for convolution process. I have designed hardware for Booth's algorithm for multiplication [3], Ripple Carry Adder (RCA) and Carry Look Ahead adder (CLA) for addition processes. This CRH is the critical role in the architectures.

Here I am describing the two implemented Novel architectures (1) All sequential architecture (2) Quad Parallelized Hardware Architecture. The image size is 64 by 64 and kernel size is 3 by 3.

2.1 All sequential Architecture

Let us consider The 64 by 64 size image pixel data is coming serially from a communication network one pixel after the other in the order that first row first and then second row and then third row and so on, from an external world/from a different unit in a single circuit to the convolution engine, and are loaded sequentially in a array memory block of size 4096 locations. A counter is used to load the data into the memory block. The property of the memory is that accessing memory for writing is serial and for reading is in serial/parallel for various words at a time. Whenever there is a convolution result generated then it is written into the output 4096 size array memory, where all the convolution results are written. This memory is having the property that accessing the memory for writing in parallel/serial and reading serial. Here also there is a counter keeping track of the location where the result is to be written basing on the location of the image pixel for which the convolution result is generated. The architecture of this design is provided in the Figure 2.

The input memory and output memory, all control signals, counters are cleared to zero; and the kernel memory of 9 locations is loaded with proper flipped kernel values directly on reset signal. There is a controller which

enables a signal to start loading the data into the input memory of the stream of multi byte serial data of image pixels. There is a counter which tracks how many pixels are loaded and through controller a finish signal is generated when all the pixels are loaded. Then the controller generates a start signal and there is a state machine which consists of 4096 states, goes into state 1 (also on reset). There is a single CRH is utilized to generate all convolution results one after the other. When the state machine is in first state S1, CRH generates the first row first column convolution result, by taking the corresponding pixels from the input memory and the Kernel values as its inputs. Then the state machine generates a finish signal, by taking a complete signal from CRH, for a clock period of time when the convolution result is ready. There is a counter that takes care of the location where the most recently generated convolution result to be written into the output memory. Once the finish signal is generated a write signal is generated for a clock cycle in the same state to write the value into the corresponding location of the output memory. Then the output memory counter also incremented by one. Then the controller generates one more signal, so that the state machine goes into second state S2, and the first row second column convolution result is generated by the CRH by taking the corresponding values of image pixels from the input memory and the kernel and this process goes on until all the convolution results are generated and written into the output memory. In this architecture the CRH is called 64 times $64 = 4096$ times.

2.2 Quad Parallelized Hardware Architecture

This is described in Figure 3. In this architecture the 64 by 64 image matrix is divided into 4 equal parts like a Cartesian plane 4 equal quadrants (imaginary quadrants for understanding purposes, as all the pixels are loaded into a single contiguous memory). Each quadrant is having its own state machine of 1024 states, and own CRH. Then 4 different CRH units are used for generating the Convolution results, one per quadrant. Imagine that the image matrix is divided into 4 equal quadrants, means each of 32 by 32 size = 1024 image pixels. For generating the convolution results of each quadrant the CRH of that quadrant is called 1024 times. There are 4 1024 state machines to generate each convolution result in all 4 quadrants, and as in the architecture – 1 when the result is generated 4 finish signals are generated simultaneously. So at the same time 4 concurrent results are generated by the 4 CRH units. This time a small counter is sufficient to generate the addresses, because it has to generate just 1024 addresses, as 4 results are simultaneously written into the output memory into the right locations on write signal (signals) generated by the 4 state machines as the memory is multiple write at a time and single read property, as discussed before. Here the output memory is accessed from multiple locations simultaneously for writing as there are 4 results are generated at a time one per quadrant by using 4 different CRHs in parallel by taking the corresponding image pixels from input memory and kernel values for each convolution result. No memory location is written twice, so no problem of simultaneous access.

3. RESULTS

Both the above developed sequential and parallel architectures for Digital Image Processing convolution are designed and implemented on Artix-II FPGAs, Xilinx Vivado 2015.2 tool programming using Verilog HDL. These are implemented for 64 by 64 size image. These two architectures implementations are compared in terms of clock speed, path delay, resource utilization and total power consumption. The results are tabulated in Table 1, Table 2 and analyzed in Table 3. The results are graphically represented in Figure 4. From Table 3 and Figure 4 it is clear that Architecture – 2 works much faster than Architecture – 1, due to the Quad Parallelization of Architecture – 1 but at the expense of more utilization of resources and power.

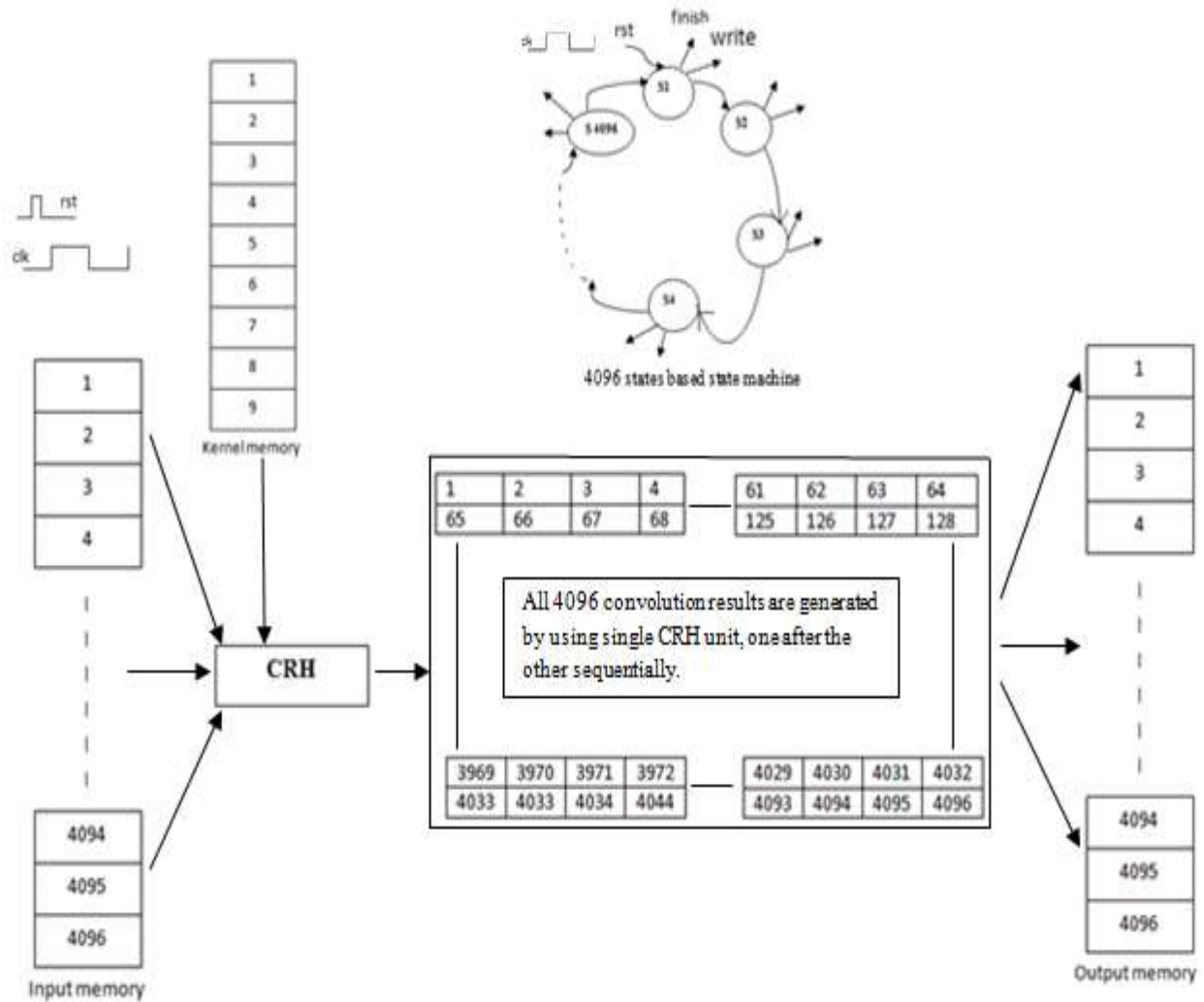


Figure 2 All Sequential Architecture: In this all 4096 convolution results are generated one after the other by using single CRH unit. State-machine with states from S1 to S4096. Memory blocks names are "input memory", "kernel memory", "output memory"

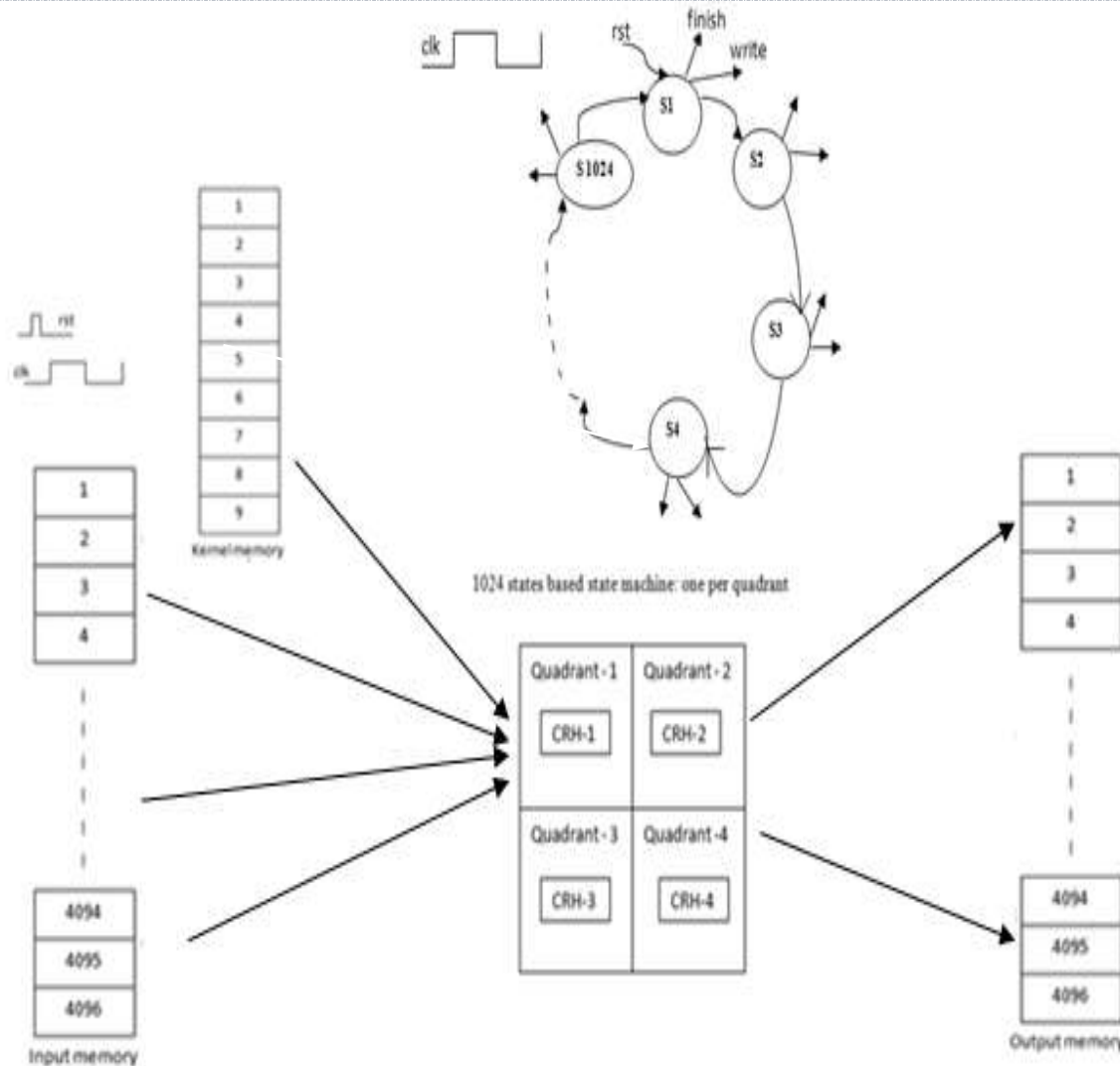


Figure 3 Quad parallelized architecture by dividing image into 4 imaginary quadrants and 4 CRH units working in parallel, one for each quadrant and 4 1024 state machines one per quadrant all working in parallel. 1024 convolution results per quadrant generated sequentially one after the other, and 4096 are generated 4 at a time in parallel from 4 quadrants. Memory blocks names are “input memory”, “kernel memory”, “output memory”.

Table 1. Results for Architecture 1

FFs(267600)	LUT(133800)	BUFG(32)	On chip power(w)	Total Delay(ps)	Clock Speed(GHz)	Time Period(ps)
3335	1279	2	23.237	292912	675	1.48

Table 2. Results for Architecture 2

FFs(267600)	LUT(133800)	BUFG(32)	On chip power(w)	Total Delay(ps)	Clock Speed(GHz)	Time Period(ps)
4636	1677	2	32.766	76280	698	1.43

Table 3 % Comparison of Architecture – 2 over Architecture – 1

Parameter	Architecture 1	Architecture 2	Percentage (%)
Total on chip resources	4616	6315	36.81 % more
Total on chip power(w)	23.237	32.766	41.00 % more
Total Delay (ps)	292912	76280	284% less time (3.84 times faster)

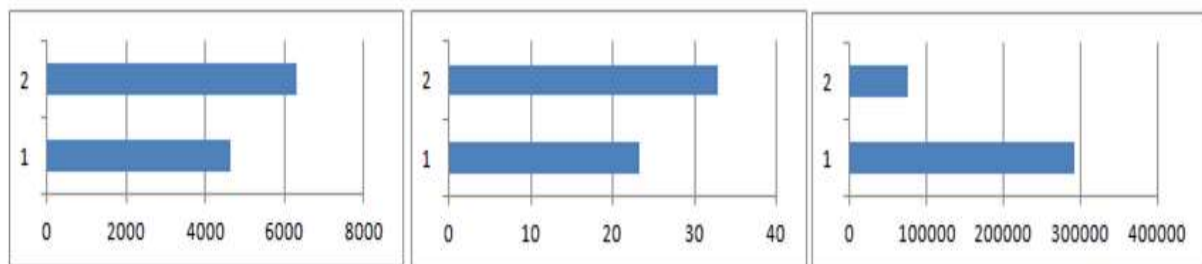


Figure 4 The graphical representation of the comparative parameters between Architecture 1(1 in graph), and Architecture 2(2 in graph) in terms of (from left to right) resource utilization, power utilization and delay for the total process to be finished.

4. APPLICATIONS

The implemented hardware for digital image processing Convolution process is highly useful in finding convolution of 64 by 64 size images, which is a standard size in many image processing applications. If in any image of higher size such as 1024 by 1024 image if part of it is to be find convolution results of 64 by 64 size of any part of the whole image then this implemented hardware is readily useful. In telemedicine if needed to communicate the convolution filtered image, second architecture is highly useful in transmitting faster than the first architecture, as speed of transmission is most important in medical image data transmissions. Also applicable in hardware of DCNN convolution stages.

5. CONCLUSIONS

In this research I have implemented two hardware architectures for 2D convolution for a 64 by 64 size digital image, 3 by 3 size kernel. I have parallelized the Architecture – 1 into Quad parallelized architecture. This Architecture – 2 has reached the theoretical expectations of 4 times speed: 3.84 times faster than first one at the expense of more resources.

Conflict of interest

The author declares that there is no conflict of interest in this paper

REFERENCES

- [1] Ranganadh Narayanam, “Translation Invariant (TI) based Novel Approach for better De-noising of Digital Images”, IRJET, vol 4, Issue 3, March 2017. Google Scholar.
- [2] Ranganadh Narayanam, “FPGA implementation of Novel architectures for digital image processing convolution filter: development of Novel quartile division architectures”, IJESRT, 8(2), Feb 2019. ResearcherID.
- [3] Ranganadh Narayanam, SSSP Rao “Implementation of a highly efficient novel frequency domain SNR hardware using Xilinx FPGAs”, IJESRT, 6(12), December 2017. ResearcherID.
- [4] Rafael C Gonzalez and Richard E Woods, “Digital Image Processing”, third edition, Pearson.