CSD Based VLSI Architecture for NN Based Image Compression

Lakshmi kiran Mukkara, K. Venkata Ramanaiah

Abstract: Images take huge amount of transmission bandwidth and memory storage space. Image compression can handle these issues properly. Especially, neural network (NN) structures can be used in many decades for this purpose because of its reliability, programmability, parallel computing capability, robustness and so on which are not possible with conventional techniques. VLSI implementation of NN structures is limited due to its complexity and power consumption. In this paper a novel approach is proposed for low loss image compression VLSI architecture for efficient power and speed by using pipelined CSD algorithm. The proposed low loss image compression can be implemented on FPGA by using Xilinx Vivado.

Index Terms: Canonic Signed Digit, Neural Network, FPGA Structures, VLSI Design.

I. INTRODUCTION

Communication or storage of images is a lit bit costlier process. Fortunately, image compression techniques are developed which reduce the cost and allow transfer or store the more amounts of image data. One can adopt neural network structures to compress and decompress the images in noisy environments to get fast and better image compression. There are many approaches of software and hardware algorithms are available for constructing NN structures. Transistor scaling has led to the development of low power-low area hardware for realizing complex parallel structures. It is very important for web or internet based applications to control image data across many computer networks.

As the field of Digital Image processing is growing because of image compression, so the field of VLSI will be successful with low power-high speed design. Hardware design with VLSI contains multiplier as a major basic block. So Multiplier design affects the entire digital system design & implementation of DIP algorithms. CSD algorithm is proved to be an efficient multiplication algorithm when compared with standard multiplication algorithms such as Booth algorithm and Vedic algorithm. In literature some works are carried out on CSD based VLSI architecture for NN based Image Compression. Though it [3] produces power efficient and speed efficient results but failed to provide accurate results due to truncation of fraction part. In this paper a novel approach is proposed to have accurate results along with power efficiency and speed efficiency.

Section 2 shows about NN Image compression by considering various Multi-Layer Perceptron (MLP) NN structures. Section 3 presents CSD based VLSI architecture for NN based Image compression. Section 4 shows results & discussion of this work. Section 5 presents conclusion and future scope of this work.

II. NN BASED IMAGE COMPRESSION

NN based image compression is considered to be coding based image compression. Neural network contains input layer, output layer and hidden layer. Image of appropriate size is given as input to the input layer after reshaping the pixels in it. Target is also set as input and the neural network is trained by adjusting its weights and bias values to provide desired output with less MSE. If hidden layer is also considered in the neural network with lesser number of nodes in it than in input side and in output side, then the input image is said to be compressed.

Consider an input image of size 64x64, and divide it into sub images of size 8x8 each. Weight matrix is taken at the size of 4x8 and multiplying this with 8x8 sub images in sequence produces compressed sub images of size 4x8 each. Merging of all compressed sub images provide us compressed image of size 4x8x64=32x64 which 50% of input image. In this work, fit-net is the NN-training algorithm. With different number of input, hidden and output nodes are considered for compressing the images as shown in Fig 1. Performance comparison of NN structures are shown in Table 1. In Table 1 S.No 1-3 are single hidden layer results and S.No 4-6 are two hidden layer results.

Among these 8-4-8-8 and 16-16-16 structures are shown in the Fig 1. The structure 8-4-8-8 is taken into considered for implementing on FPGA as it has good PSNR value and 50% compression.

Table 1: Performance comparison of neural network structures

<table>
<thead>
<tr>
<th>S.no</th>
<th>MLP NN structure</th>
<th>r(correlation factor)</th>
<th>MSE</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16-4-16</td>
<td>0.9998</td>
<td>2.25</td>
<td>44.6430</td>
</tr>
<tr>
<td>2</td>
<td>16-8-16</td>
<td>0.9998</td>
<td>2.25</td>
<td>44.6430</td>
</tr>
<tr>
<td>3</td>
<td>16-16-16</td>
<td>1.000</td>
<td>2.1176</td>
<td>44.9063</td>
</tr>
<tr>
<td>4</td>
<td>16-4-16-16</td>
<td>0.9996</td>
<td>2</td>
<td>45.1545</td>
</tr>
<tr>
<td>5</td>
<td>16-4-8-16</td>
<td>0.9995</td>
<td>2.1176</td>
<td>44.9063</td>
</tr>
<tr>
<td>6</td>
<td>8-4-8-8</td>
<td>0.9996</td>
<td>2</td>
<td>45.1545</td>
</tr>
</tbody>
</table>
III. CSD BASED VLSI ARCHITECTURE FOR NN BASED IMAGE COMPRESSION

NN based image compression contains many number of multipliers and adders. Replacing convention multipliers and adders with the effective ones will give good VLSI architecture. One such multiplier is the Radix-2 CSD multiplier also known as multiplier less algorithm requires lesser number of shift and add operations as it converts multiplier with lesser number of non-zeros approximately 33% lesser non-zeros than in actual number.

CSD multiplication algorithm with pipelined concept:

\[ \hat{a}_{i-1} = 0 \]
\[ \gamma - 1 = 0 \]
\[ \hat{a}_w = \hat{a}_{w-1} \]
\[ \text{for } (i = 0 \text{ to } W - 1) \]
\[ \theta_i = \hat{a}_i \oplus \hat{a}_{i-1} \]
\[ \gamma_i = \gamma - 1 \theta_i \]
\[ a_j = (1 - 2\hat{a}_{i+1})\gamma_i \]

CSD conversion algorithm:

\[ \text{X} \]
\[ \text{h} \]
\[ \text{W} \]
\[ \text{X} \]

Fig 3: Block diagram of pipelined CSD multiplication algorithm [4].

IV. RESULTS & DISCUSSION

The proposed neural network architecture contains 8 input neuron is designed by using Verilog HDL. The network supports numbers between 0 and 1 that can be acquired with BCSD algorithm. The proposed design is verified functionally with the test bench, synthesized on Spartan device to know the complexity of hardware for efficient FPGA implementation on Nexys board and Zed board. Re-configurability of FPGA leads to get better compression by changing input matrix and weight matrix. This work is based on multiplication of 2 matrices, first one is input image matrix and second one is weight matrix which is obtained after NN training tool. The multiplied output is send through activation function for getting compressed output, and is either stored (in compressed format) or transmitted. The compressed image is multiplied again with the weight matrix to recover the original image at the receiver side. Quality of the recovered image is based upon weight matrix. The 8x8 sub image is multiplied with 4x8 weight matrix to get a 4x8 compressed image. This compressed image is multiplied with 8x4 weight matrix to get back 8x8 original sub image.

MATLAB results and Simulation results are as shown in the following paragraphs. Some works carried out in literature regarding this work. But their results are not perfectly with the MATLAB results. They discarded the fractional part in the output of matrix multiplication, thus it may not be suitable for the application requiring no loss of data such as medical image processing applications.
In the proposed work, FPGA implementation results are almost matched with the MATLAB results so that it may be useful for low loss image compression applications. Furthermore additional bits can be taken for the multiplier operant to get exact MATLAB results.

A. MATLAB Results:
\[
A = \begin{bmatrix}
226 & 228 & 226 & 226 & 226 & 226 & 228 & 228 \\
226 & 226 & 226 & 226 & 226 & 226 & 228 & 228 \\
226 & 228 & 226 & 226 & 226 & 226 & 228 & 228 \\
226 & 226 & 226 & 226 & 226 & 226 & 228 & 228 \\
226 & 228 & 226 & 226 & 226 & 226 & 228 & 228 \\
226 & 226 & 226 & 226 & 226 & 226 & 228 & 228 \\
226 & 228 & 226 & 226 & 226 & 226 & 228 & 228 \\
226 & 226 & 226 & 226 & 226 & 226 & 228 & 228 \\
\end{bmatrix}
\]
\[
B = \begin{bmatrix}
0.6670 & 0.4805 & 0.1365 & -0.1067 & 0.4565 & 0.3721 \\
0.2078 & 0.1189 & 0.4004 & -0.0576 & -0.3003 & -0.0237 \\
-0.0930 & -0.1882 & -0.2195 & 0.9998 & 0.9998 & 0.6091 & 0.2117 \\
0.6836 & 0.6758 & 0.5840 & 0.5327 & 0.0247 & 0.0989 & 0.1956 \\
0.2620 & 0.0022 & 0.0105 & 0.0342 & 0.0547 & & & \\
\end{bmatrix}
\]
\[
C = A \times B = \begin{bmatrix}
527.125 & 530.81 & 527.125 & 527.125 & 527.125 \\
528.562 & 531.6 & -527 & 526 & 527 & 527 \\
-535 & -527 & -530 & 1197 & 1204.1 & 1197 & 1197 & 1197 \\
1200.2 & 1206.5 & 1206.4 & 154.3 & 154.8 & 154.3 & 154.3 \\
154.3 & 154.9 & 155.6 & 155.3 & & & & \\
\end{bmatrix}
\]

B. Simulink Results
Simulation results of the proposed work is as shown in below 4 figures (Fig 4), each figure contains one row of a resultant matrix. These Simulation results are exactly match with the MATLAB results shown above. These results are corresponding to compressed image of size 4x8 with initial size of 8x8. Similar work is to be done for getting decompressed image. i.e multiplying the compressed image with weight matrix produced by NN tool to get back the decompressed image. Table 2 shows FPGA Post implementation results in the boards: Nexys board and Zed board.

V. CONCLUSION & FUTURE WORK
A novel approach of image compression and its FPGA implementation is successfully carried out in this work by considering Multi-Layer NN structures for image compression and Pipelined CSD multiplication algorithm for efficient FPGA implementation. Results of the proposed work are almost exactly matched with the MATLAB results. i.e. output values didn’t undergo truncation of decimal part unlike in existing works.
Thus this work can be extended to make it useful in critical applications such as medical image compression which require low-loss or lossless image compression.

REFERENCES

9. V.Sandhiya, Mr.S.Karthick², Dr.M.Valarmathy “A Survey of New Reconfigurable Architectures For Implementing FIR Filters With Low Complexity” International Conference on Computer Communication and Informatics (ICCCI -2014), Jun. 03 –05, 2014; Coimbatore, INDIA