Systolic Array based Multiply Accumulation Unit for IoT Edge Accelerators

Lahari P.L, Siva Sankar Yellampalli, *IEEE Senior Member,* and, Ramesh Vaddi, *IEEE Senior Member*
Department of Electronics and Communication Engineering, SRM University, Andhra Pradesh, India
Email: lahari_p@srmap.edu.in, Sivasankar.y@srmap.edu.in, Ramesh.v@srmap.edu.in

***Abstract*—Accelerator is a hardware that runs along with the processor and executes the key functions much faster than the processor. The Main purpose of the Accelerator is to increase speed. Deep Neural Networks has achieved wide results in the various Machine Learning Applications Such as image, video, text classification and language translation. The purpose of DNN Accelerators is to speed up the most complex Computation i.e., matrix multiplication. Systolic array Based Accelerator seems like multiply Accumulate unit with Systolic Array based multiplication followed by Adder and accumulator. Multiply Accumulate Unit comprises multiplier, adder and Accumulator. Multiplier is designed used systolic array and that output is given as one of the inputs to the adder followed by Accumulator. In this paper general Matrix based Multiply Accumulate Unit is compared with systolic array based Multiply Accumulate Unit using Xilinx ISE 14.5, various parameters like area, delay and speed are compared. Systolic Array based Multiply Accumulate Unit consumes less area of 49%, less delay of 35% and in turn provides high speed when compared with general matrix multiplier-based multiplier Accumulate unit.**

Keywords— Systolic Array, Accelerator, Multiply Accumulate Unit, Processor

I. INTRODUCTION:

Deep Neural Network is a special type of Artificial Neural Network which contains many layers between input and the output layer. DNN operates the most complex operation that is matrix multiplication [1]. Here present layer multiplies with the previous layer considering matrix of weights. Systolic Arrays [3] uses the concept of systolic processing. In the systolic arrays we have many simple processors technically called as processing elements. By seeing the architecture inside the processing element [4][5],. Systolic architecture is nothing but the network of processing elements that is used for passing data throughout the system rhythmically.

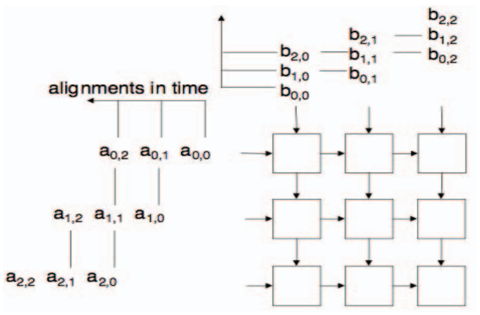


Fig 1: Architecture of PE array

Processing Elements [6] are arranged in such way to form an array. This is explained clearly with the help of matrix. Here 3*3 matrices are taken which are given as inputs named as a & b. Suppose the input matrices a, b be

$$A = \begin{bmatrix} a0,0 & a0,1 & a0,2 \\ a1,0 & a1,1 & a1,2 \\ a2,0 & a2,1 & a2,2 \end{bmatrix} \quad [1]$$

$$B = \begin{bmatrix} b0,0 & b0,1 & b0,2 \\ b1,0 & b1,1 & b1,2 \\ b2,0 & b2,1 & b2,2 \end{bmatrix} \quad [2]$$

The matrix multiplication is done in such a way that

$$A * B =$$

$$\begin{bmatrix} a0,0*b0,0+a0,1*b1,0+a0,2*b2,0 & a0,0*b0,1+a0,1*b1,1+a0,2*b2,1 & a0,0*b0,2+a0,1*b1,2+a0,2*b2,2 \\ a1,0*b0,0+a1,1*b1,0+a1,2*b2,0 & a1,0*b0,1+a1,1*b1,1+a1,2*b2,1 & a1,0*b0,2+a1,1*b1,2+a1,2*b2,2 \\ a2,0*b0,0+a2,1*b1,0+a2,2*b2,0 & a2,0*b0,1+a2,1*b1,1+a2,2*b2,1 & a2,0*b0,2+a2,1*b1,2+a2,2*b2,2 \end{bmatrix} \quad [3]$$

But when comes to the systolic array [8]. For example, here 3*3 matrices are considered. So that the output consists of 9 elements. When systolic array[9] is considered, we require 9 processing elements to form an array.

II. DESIGN METHODOLOGIES

A. Existing designs:

i. Matrix Multiplication:

For example A, B be the input matrices taken as given below in the process explained in the equation 3.

$$\textbf{input A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad [4]$$

$$\textbf{input B} = \begin{bmatrix} 10 & 11 & 12 \\ 1 & 2 & 3 \\ 13 & 14 & 15 \end{bmatrix} \quad [5]$$

$$\textbf{A*B} = \begin{bmatrix} 51 & 57 & 63 \\ 123 & 138 & 153 \\ 195 & 219 & 243 \end{bmatrix} \quad [6]$$

ii. Systolic Matrix multiplication:

From the same above example, it is represented as processing elements already discussed.

PE1=51	PE2=57	PE3=63
PE4=123	PE5=138	PE6=153
PE7=195	PE8=219	PE9=243

B.Proposed Designs:

i. Matrix based Multiply Accumulate unit:

To increase the speed of overall processor, multiply accumulate unit is required which in turn consists multiplier, adder and accumulator.

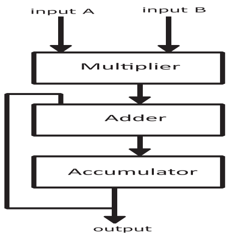


Fig 2: multiply Accumulate unit

The output that is obtained from the multiplier is given as one of the inputs to the adder followed by accumulator also called as normal matrix multiplier-based MAC[2].

ii.Systolic Array based Multiply Accumulate Unit/systolic array based DNN Accelerator:

This block comprises of systolic Array based matrix multiplier followed by Adder and accumulator. Here we can say the same architecture as Systolic array based DNN Accelerator. This is also designed in the similar way of normal matrix multiplier-based MAC [10]but instead of normal matrix multiplier systolic array is incorporated. The adder block[11] adds the output that is coming from the systolic array and one more from the accumulator.

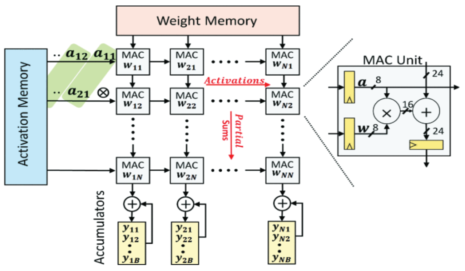


Fig 3: systolic array based DNN Accelerator.

This Architecture consists of systolic array-based multiplier followed by adder and accumulator resembles MAC

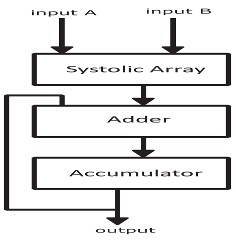


Fig 4 : systolic array-based MAC

Systolic array-based MAC also consumes less area and provides high speed.

III. RESULTS AND ANALYSIS

Matrix multiplication

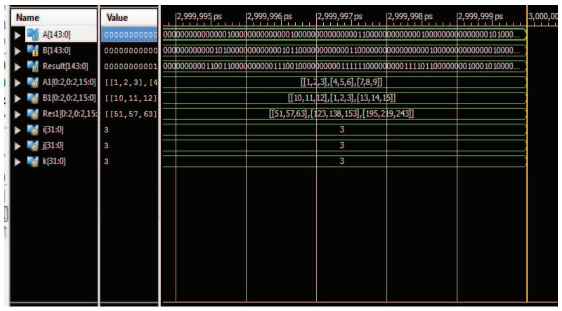
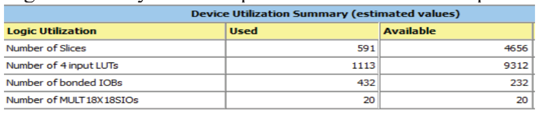


Fig 5: simulation results for matrix multiplication

```
 ------------------------------------------------
 Total                      20.266ns (15.869ns logic, 4.397ns route)
                                     (78.3% logic, 21.7% route)

==========================================================================
```

Fig 6: Delay consumption of matrix multiplication



Device Utilization Summary (estimated values)		
Logic Utilization	**Used**	**Available**
Number of Slices	591	4656
Number of 4 input LUTs	1113	9312
Number of bonded IOBs	432	232
Number of MULT18X18SIOs	20	20

Fig 7:Area consumption of matrix multiplication

Systolic Array

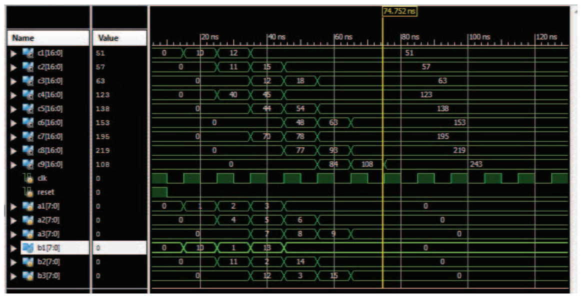


Fig 8: simulation results for systolic array based 3*3 multiplier

```
 ------------------------------------------------
 Total                      9.425ns (8.395ns logic, 1.030ns route)
                                    (89.1% logic, 10.9% route)

=====================================================================
```

Fig 9: Delay Consumption for systolic array based 3*3 multiplier.

Device Utilization Summary (estimated values)		
Logic Utilization	**Used**	**Available**
Number of Slices	109	4656
Number of Slice Flip Flops	201	9312
Number of 4 input LUTs	153	9312
Number of bonded IOBs	203	232
Number of MULT18X18SIOs	9	20
Number of GCLKs	1	24

Fig 10: area comparison for Systolic Array based 3*3 multiplier

General Matrix multiplication-based MAC

$$input\ A = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \qquad [7]$$

$$input\ B = \begin{bmatrix} 2 & 2 & 2 \\ 2 & 2 & 2 \\ 2 & 2 & 2 \end{bmatrix} \qquad [8]$$

$$A * B = \begin{bmatrix} 12 & 12 & 12 \\ 12 & 12 & 12 \\ 12 & 12 & 12 \end{bmatrix} \qquad [9]$$

Complete structure operates in two cases.

Case 1: if no accumulation is done,

$$Z = \begin{bmatrix} 12 & 12 & 12 \\ 12 & 12 & 12 \\ 12 & 12 & 12 \end{bmatrix} \qquad [10]$$

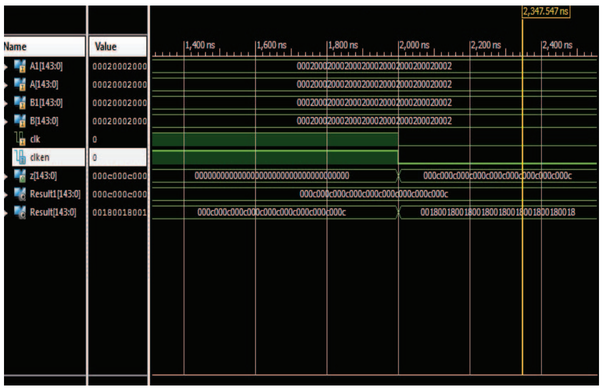
Case 2: if accumulation done

From case1

$$\mathrm{Z} = \begin{bmatrix} 12 & 12 & 12 \\ 12 & 12 & 12 \\ 12 & 12 & 12 \end{bmatrix} + \text{ multiplier output gets added and provides final Z.} \qquad [11]$$

$$\textbf{Final}\ z = \begin{bmatrix} 12 & 12 & 12 \\ 12 & 12 & 12 \\ 12 & 12 & 12 \end{bmatrix} + \left(A * B = \begin{bmatrix} 12 & 12 & 12 \\ 12 & 12 & 12 \\ 12 & 12 & 12 \end{bmatrix} \right)$$

$$= \begin{bmatrix} 24 & 24 & 24 \\ 24 & 24 & 24 \\ 24 & 24 & 24 \end{bmatrix} \qquad [12]$$



```
Name            Value
A1[143:0]       00020002000   0002000200020002000200020002000200020002
A[143:0]        00020002000   0002000200020002000200020002000200020002
B1[143:0]       00020002000   0002000200020002000200020002000200020002
B[143:0]        00020002000   0002000200020002000200020002000200020002
clk             0
clken           0
z[143:0]        000c000c000   000000000000000000000000000000000000   000c000c000c000c000c000c000c000c000c
Result1[143:0]  000c000c000   000c000c000c000c000c000c000c000c000c
Result[143:0]   00180018001   000c000c000c000c000c000c000c000c000c   001800180018001800180018001800180018
```

Fig 11: simulation results for 3*3 based MAC

```
  ----------------------------------------
  Total                      644.968ns (456.852ns logic, 188.116ns route)
                                       (70.8% logic, 29.2% route)

==========================================================================
```

Fig 12: Delay comparison for 3*3 based MAC

Device Utilization Summary (estimated values)		
Logic Utilization	**Used**	**Available**
Number of Slices	818	4656
Number of 4 input LUTs	1545	9312
Number of bonded IOBs	433	232
Number of MULT18X18SIOs	20	20

Fig 13: RTL, Simulation Results, delay and area consumptions of general matrix-based MAC.

This design consumes the area of 818 slices out off 4656 slices and delay of 644.968 ns.

Systolic array based 3*3 DNN Accelerator/Systolic array-based MAC

$$\text{Input A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \qquad [13]$$

$$\text{input B} = \begin{bmatrix} 1 & 3 & 3 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \end{bmatrix} \qquad [14]$$

$$A * B = \begin{bmatrix} 8 & 10 & 10 \\ 20 & 28 & 28 \\ 32 & 46 & 46 \end{bmatrix} \qquad [15]$$

Since 3*3 matrix is considered 9 PE's are required.

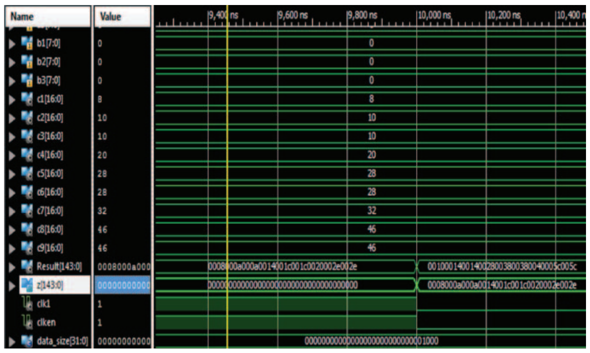
PE1 =8	PE2=10	PE3=10
PE4=20	PE5=28	PE6=28
PE7=32	PE8=46	PE9=46

Case 1: if no accumulation is done,

$$Z = \begin{bmatrix} 8 & 10 & 10 \\ 20 & 28 & 28 \\ 32 & 46 & 46 \end{bmatrix} \qquad [16]$$

Case 2: if accumulation is done

$$\text{Final}\ \ Z = \begin{bmatrix} 8 & 10 & 10 \\ 20 & 28 & 28 \\ 32 & 46 & 46 \end{bmatrix} + \left(A * B = \begin{bmatrix} 8 & 10 & 10 \\ 20 & 28 & 28 \\ 32 & 46 & 46 \end{bmatrix} \right) = \begin{bmatrix} 16 & 20 & 20 \\ 40 & 56 & 56 \\ 64 & 92 & 92 \end{bmatrix} \ [17]$$



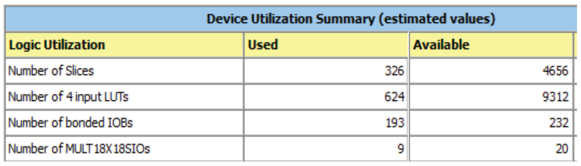
```
Name             Value
b1[7:0]          0             0
b2[7:0]          0             0
b3[7:0]          0             0
c1[16:0]         8             8
c2[16:0]         10            10
c3[16:0]         10            10
c4[16:0]         20            20
c5[16:0]         28            28
c6[16:0]         28            28
c7[16:0]         32            32
c8[16:0]         46            46
c9[16:0]         46            46
Result[143:0]    0008000a000   0008000a000a0014001c001c0020002e002e   0010001400140028003800380040005c005c
z[143:0]         00000000000   000000000000000000000000000000000000   0008000a000a0014001c001c0020002e002e
clk1             1
clken            1
data_size[31:0]  00000000000   00000000000000000000000000001000
```

Fig 14: Simulation Results for 3*3 systolic array based DNN Accelerator

```
  ----------------------------------------
  Total                      290.909ns (222.456ns logic, 68.452ns route)
                                       (76.5% logic, 23.5% route)

==========================================================================
```

Fig 15: delay consumption for 3*3 systolic array based DNN Accelerator

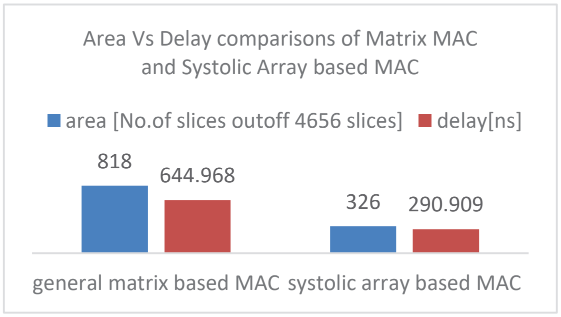


Device Utilization Summary (estimated values)		
Logic Utilization	**Used**	**Available**
Number of Slices	326	4656
Number of 4 input LUTs	624	9312
Number of bonded IOBs	193	232
Number of MULT18X18SIOs	9	20

Fig 16: Area consumptions of 3*3 systolic array based DNN Accelerator

This design consumes the area of 326 slices outoff 4656 slices and delay of 290.909 ns.

Comparison Chart : area and delay comparison for the normal MAC vs systolic array based MAC



From this systolic array-based MAC occupies less area of 326 slices outoff 4656 and inturn provides high speed compared to normal MAC.

IV. CONCLUSION:

Convolution, parallel integration, matrix multiplication or data sorting, multiply Accumulate unit etc., tasks will be performed with the help of the matrix. For performing data sorting tasks systolic array algorithm is preferred. So general matrix based Multiply Accumulate unit is designed and compared with systolic array based multiply Accumulate unit in Xilinx ISE 14.5 software. From the simulation results, it is clear that systolic array-based MAC consumes less area of 49% and less delay of 35% compared with matrix multiplication-based MAC. Since delay is inversely proportional to speed, systolic array-based MAC is speed efficient.

REFERENCES

[1]. J. Zhang, K. Rangineni, Z. Ghodsi and S. Garg, "ThUnderVolt: Enabling Aggressive Voltage Underscaling and Timing Error Resilience for Energy Efficient Deep Learning Accelerators," 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), 2018.

[2]. M. Bharathi, Y. J. M. Shirur and P. L. Lahari, "Performance evaluation of Distributed Arithmetic based MAC Structures for DSP Applications," 2020 7th International Conference on Smart Structures and Systems (ICSSS), 2020.

[3]. H. Snopce, A. Aliu and A. Luma, "Mapping Signal Processing Algorithms Into The Systolic Arrays," 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), 2020.

[4]. N. Sutisna, G. Jonatan, I. Syafalni, R. Mulyawan and T. Adiono, "Polynomial Multiplication Systolic Array for Homomorphic Encryption in Secure Network Communications," 2020 IEEE International Conference on Communication, Networks and Satellite (Comnetsat), 2020.

[5]. E. Yago, P. Castelló, S. Petit, M. E. Gómez and J. Sahuquillo, "Impact of the Array Shape and Memory Bandwidth on the Execution Time of CNN Systolic Arrays," 2020 23rd Euromicro Conference on Digital System Design (DSD), 2020.

[6]. I. Ullah, K. Inayat, J. -S. Yang and J. Chung, "Factored Radix-8 Systolic Array for Tensor Processing," 2020 57th ACM/IEEE Design Automation Conference (DAC), 2020.

[7]. G. Shomron, T. Horowitz and U. Weiser, "SMT-SA: Simultaneous Multithreading in Systolic Arrays," in IEEE Computer Architecture Letters, vol. 18, no. 2, pp. 99-102, 1 July-Dec. 2019.

[8]. L. Jia, L. Lu, X. Wei and Y. Liang, "Generating Systolic Array Accelerators With Reusable Blocks," in IEEE Micro, vol. 40, no. 4, pp. 85-92, 1 July-Aug. 2020.

[9]. H. Waris, C. Wang, W. Liu and F. Lombardi, "Design and Evaluation of a Power-Efficient Approximate Systolic Array Architecture for Matrix Multiplication," 2019 IEEE International Workshop on Signal Processing Systems (SiPS), 2019.

[10]. J. Zhang, Z. Ghodsi, S. Garg and K. Rangineni, "Enabling Timing Error Resilience for Low-Power Systolic-Array Based Deep Learning Accelerators," in IEEE Design & Test, vol. 37, no. 2, pp. 93-102, April 2020.

[11]. S. -Y. Lin, K. -H. Lin, C. -K. Tsai and P. -H. Tseng, "Reconfigurable MAC Systolic Array Architecture Design for Three-Dimensional Convolution Neural Network," 2020 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-Taiwan), 2020.