International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 5 | ISSN : 2456-3307

High Performance Scalable Deep Learning Accelerator Unit on FPGA

V. Jhansi[1], Dr. Chandrasekhar[2]
[1]M.Tech Student, Department of ECE, S.V. College of Engineering, Tirupati, India
[2]Professor, Department of ECE, S.V. College of Engineering,Tirupati, India

ABSTRACT

The deep learning network the size of the networks becomes increasingly large scale due to the demands of the practical applications .The DLAU architecture can be configured to operate different sizes of tile data to leverage the trade-offs between speedup and hardware costs. Consequently the FPGA based accelerator is more scalable to accommodate different machines. The DLAU includes three pipelined processing units, which can be reused for large scale neural networks. High performance implementation of deep learning neural network is maintain the low power cost and less delay .in this work is we are done the samples of network signals which attain data optimization, upgraded the speed.

Keywords : DLAU, Neural network ,Deep learning, Accelerator.

I. INTRODUCTION

Neural networks are a set of algorithms, modeled loosely after the human brain, that are designed to recognize patterns. They interpret sensory data through a kind of machine perception, labeling or clustering raw input. The patterns they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text or time series, must be translated.

Machine Learning offers many innovative applications in the IoT devices, such as face recognition, smart security and object detection. State-of-the-art machine-learning computation mostly relies on the cloud servers. Benefiting from the graph processing unit (GPU)'s powerful computation ability, the cloud can process high throughput video data coming from the devices and use CNN to achieve unprecedented accuracy on most applications. However, this approach has its own drawbacks. Since the network connectivity is necessary for cloud-based applications, those applications cannot run in the areas where there is no network coverage. In addition, data transfer through network induces significant latency, which is not acceptable for real-time applications such as security system. Finally, most of the IoT applications have a tough power and cost budget which could tolerate neither local GPU solutions nor transmitting massive amounts of image and audio data to data center servers.

Deep learning is networks composed of several layers .The layers are made of *nodes.* A node is just a place where computation happens, loosely patterned on a neuron in the human brain, which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights, that either amplify or dampen that input, thereby assigning significance to inputs for the task the algorithm is trying to learn.

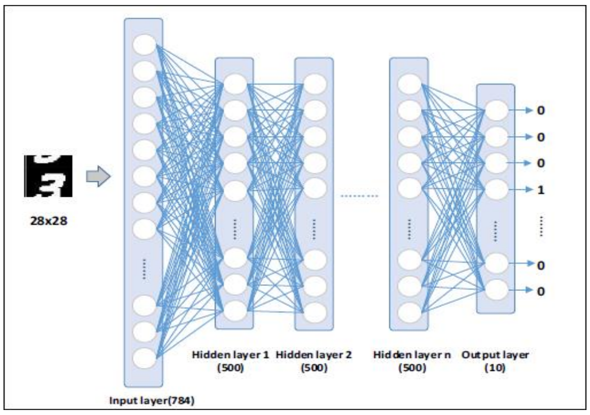


Figure 1. Deep learning neural network

Largescale deep learning neural network models. So far, the state of- the-art means for accelerating deep learning algorithms are Field-Programmable Gate Array (FPGA), Application Specific Integrated Circuit (ASIC), and Graphic Processing Unit (GPU). Compared with GPU acceleration, hardware accelerators like FPGA and ASIC can achieve at least moderate performance with lower power consumption. However, both FPGA and ASIC have relatively limited computing resources, memory, and I/O bandwidths, therefore it is challenging to develop complex and massive deep neural networks using hardware accelerators.

DEEP LEARNING ACCELERATOR UNIT(DLAU):

The DLAU system architecture which contains an embedded processor, a DDR3 memory controller, a DMA module, and the DLAU accelerator. The embedded processor is responsible for providing programming interface to the users and communicating with DLAU via JTAG-UART. In particular it transfers the input data and the weight matrix to internal BRAM blocks, activates the DLAU accelerator, and returns the results to the user after execution.

The DLAU is integrated as a standalone unit which is flexible and adaptive to accommodate different applications with configurations. The DLAU consists of 3 processing units organized in a pipeline manner: Tiled Matrix Multiplication Unit (TMMU), Part Sum Accumulation Unit (PSAU), and Activation Function Acceleration Unit (AFAU). For execution, DLAU reads the tiled data from the memory by DMA, computes with all the three processing units in turn, and then writes the results back to the memory.

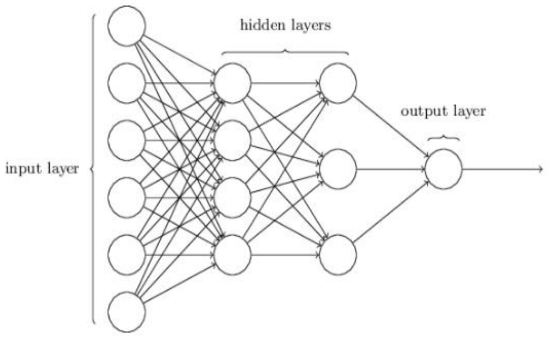


Figure 2: Neural network and deep learning

Machine Learning has been used for classification on images and text for decades, but it struggled to cross the threshold – there's a baseline accuracy that algorithms need to have to work in business settings. Deep Learning is finally enabling us to cross that line in places we weren't able to before.

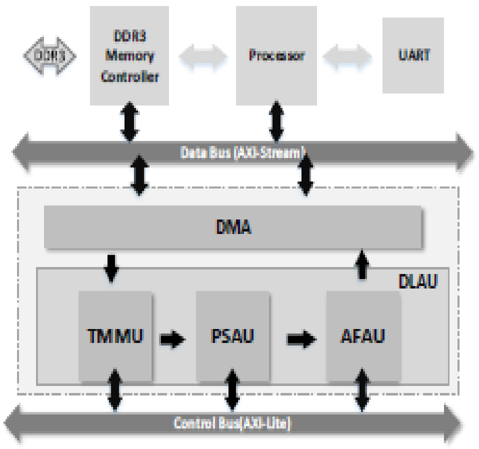


Figure 3:TheDLAU accelerator architecture

TMMU architecture:

We are now in a position to describe matrix multiplication. The product of these two matrices is the matrix showing the number of paths of length between each pair of vertices.discover its row and column within the matrix.

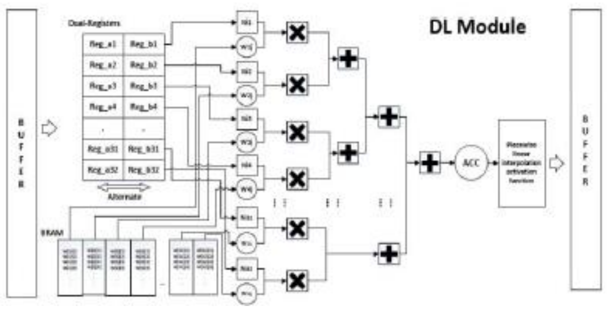


Figure 4: TMMU architecture

It is a good idea to make sure you understand those two lines before moving on. The if statement in the next line terminates the thread if its row or column place it outside the bounds of the product matrix. This will happen only in those blocks that overhang either the right or bottom side of the matrix.

We use the BRAM resources to cache the weight coefficients between two adjacent layers. The accelerator reads the matrix of weight coefficients data from input buffer,and loops to save into different BRAMs in 32 by the row number of the weight matrix (*n=i%32* *n* refers to the number of BRAM, and *i*indicates the row number of weight matrix). So, the accelerator can read 32 weight values in parallel.

To reduce the impact on performance of the data access time, we design two registers set to read the required data for next iteration computation or be used in current iteration alternately. In our test, the time to cache 32 input values is much less than the time of the computation of 32 values. So, the iteration computing will start without waiting except the first iteration.

PSAU:

Part Sum Accumulation Unit (PSAU) is responsible for the accumulation operation. presents the PSAU architecture, the TMMU is produced the accumulation parts . Then part sum is take that value s and add that values using PSAU.will write the value to output buffer and send results to AFAU in a pipeline manner.

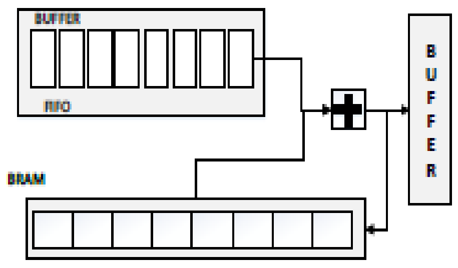


Figure 5 : PSAU architecture

PSAU can accumulate one Part Sum every clock cycle, therefore the throughput of PSAU accumulation matches the generation of the Part Sum in TMMU.

AFAU:

Activation Function Acceleration Unit (AFAU) implementsThe computation of DNNs does not need high precision; Piecewise linear interpolation can achieve better performance than other methods, such as binomial expansion. The piecewise linear interpolation (y=ai*x+bi, x2[x1,xi+1)). can implement any activation function with negligible loss of accuracy when the interval *k* between xi and xi+1 is insignificant. is enough small.

$$f(x) = \begin{cases} 0 & \text{if } x \le -8 \\ 1 + a[\lfloor \frac{-x}{k} \rfloor]x - b[\lfloor \frac{-x}{k} \rfloor] & \text{if } -8 < x \le 0 \\ a[\lfloor \frac{x}{k} \rfloor]x + b[\lfloor \frac{x}{k} \rfloor] & \text{if } 0 < x \le 8 \\ 1 & \text{if } x > 8 \end{cases} \tag{1}$$

implementation of sigmoid function. This implementation takes advantage of symmetry and bounded range. For x>8 and x_-8, the results are sufficiently close to the bounds of 1 and 0, respectively. For the cases in -8<x_0 and 0<x_8, different functions are configured. In total we divide the sigmoid function into four segments.

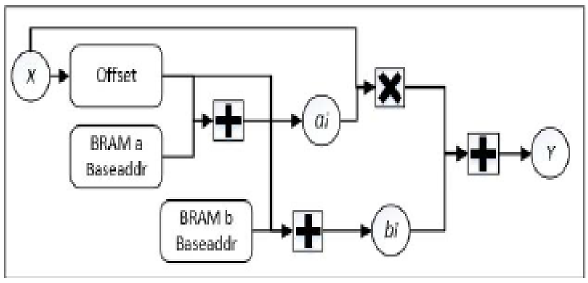


Figure 6:AFAU architecture

Using *BRAMs* to store the values of *a* set and *b* set. The *a*, *b* and *k* are fixed .So find the corresponding *a* value and *b* value according the value *x*, and get *y* after multiplication and addition. The computation process is pipelined and we can get a value every clock cycle.

SAMPLING TECHNIQUE

How many samples are necessary to ensure we are preserving the information contained in the signal. If the signal contains high frequency components, we will need to sample at a higher rate to avoid losing information that is in the signal.

In general, to preserve the full information in the signal, it is necessary to sample at twice the maximum frequency of the signal. The Sampling Theorem states that a signal can be exactly reproduced if it is sampled at a frequency F, where F is greater than twice the maximum frequency in the signal.

In this system sample the 4bit down samples so the delay of the system is reduced .and the speed of the accelerator is increased .

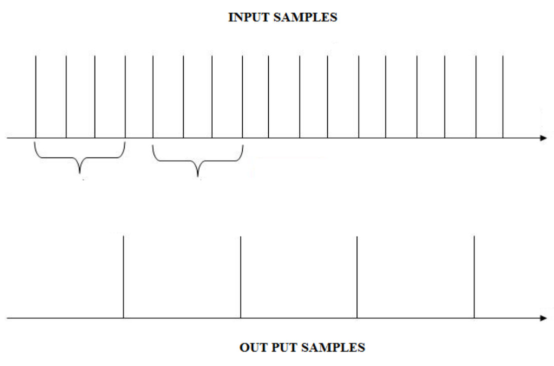


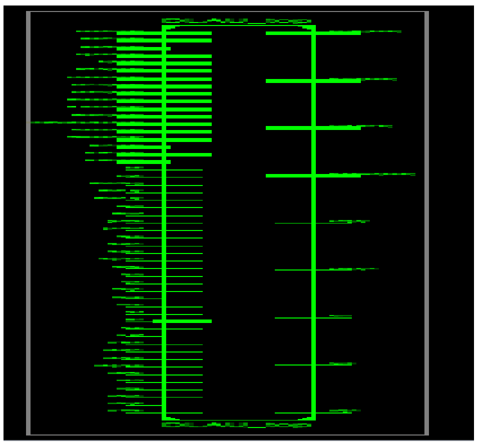
Figure 7: sample technique in DLAU

In this system the input will be separating in the form of samples then that samples are applied to the design according to this processor the input pattern will be reduced. we are Applying all the input pattern in this place we are Appyling the samples only, in these samples the input pattern will be covered.

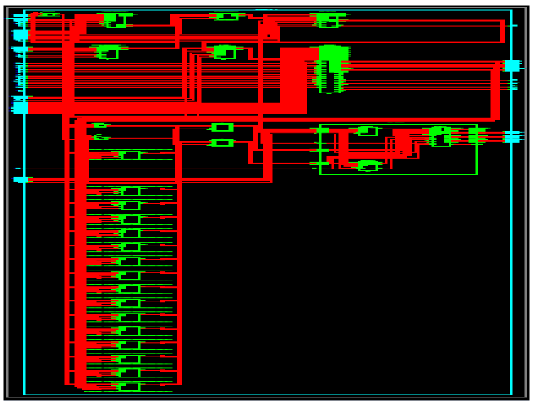
The accelerator will produce the output with less delay why because we are applying the samples of input pattern so the time delay of applying input to the system is very less than the previous system so the accelerator will work very faster.

RESULTS:

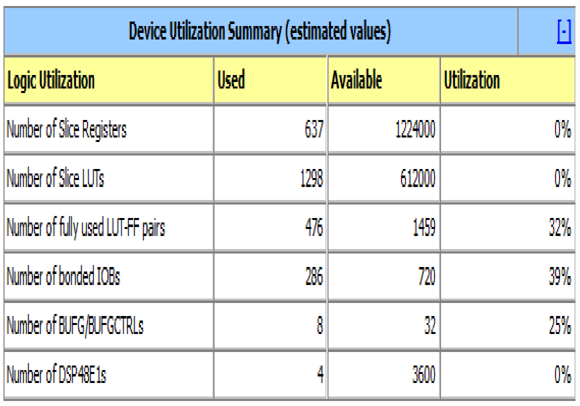
Block diagram:



RTL schematic:

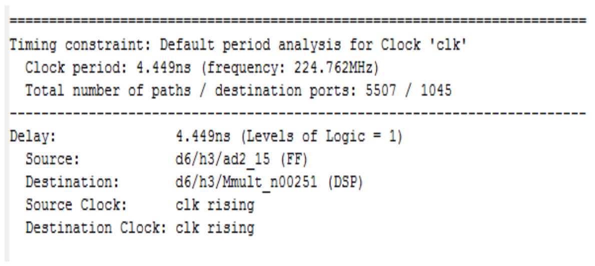


AREA



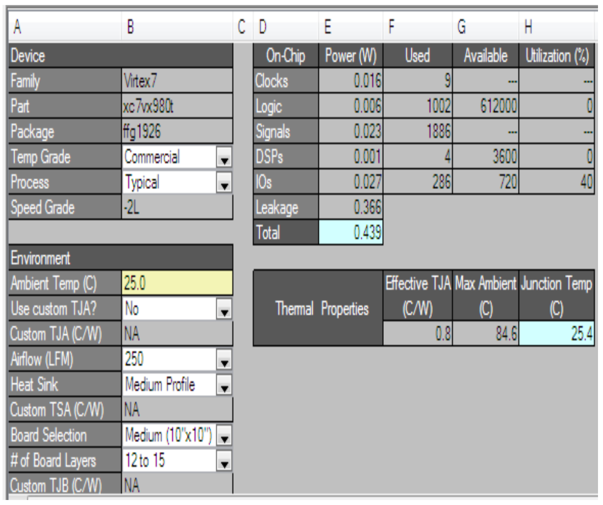
Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	637	1224000	0%
Number of Slice LUTs	1298	612000	0%
Number of fully used LUT-FF pairs	476	1459	32%
Number of bonded IOBs	286	720	39%
Number of BUFG/BUFGCTRLs	8	32	25%
Number of DSP48E1s	4	3600	0%

DELAY



```
================================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 4.449ns (frequency: 224.762MHz)
  Total number of paths / destination ports: 5507 / 1045
--------------------------------------------------------------------------------
Delay:               4.449ns (Levels of Logic = 1)
  Source:            d6/h3/ad2_15 (FF)
  Destination:       d6/h3/Mmult_n00251 (DSP)
  Source Clock:      clk rising
  Destination Clock: clk rising
```

POWER



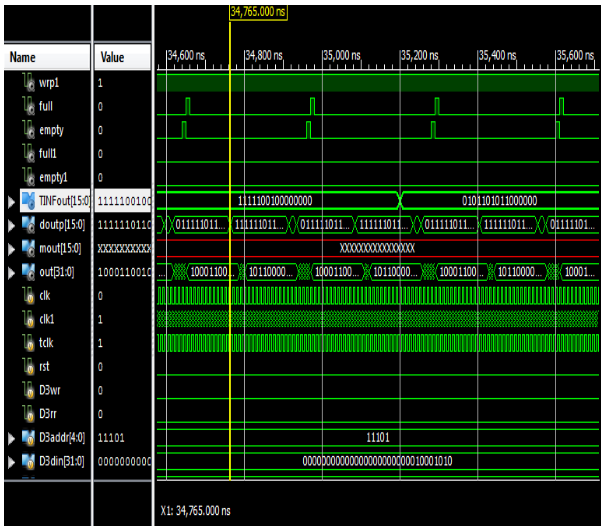
Device	
Family	Virtex7
Part	xc7vx980t
Package	ffg1926
Temp Grade	Commercial
Process	Typical
Speed Grade	-2L

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	12 to 15
Custom TJB (C/W)	NA

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.016	9	---	---
Logic	0.006	1002	612000	0
Signals	0.023	1886	---	---
DSPs	0.001	4	3600	0
IOs	0.027	286	720	40
Leakage	0.366			
Total	0.439			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	0.8	84.6	25.4

Simulation:



Name	Value
wrp1	1
full	0
empty	0
full1	0
empty1	0
TINFout[15:0]	1111100100
doutp[15:0]	1111110110
mout[15:0]	XXXXXXXXXX
out[31:0]	1000110010
clk	0
clk1	1
tclk	1
rst	0
D3wr	0
D3rr	0
D3addr[4:0]	11101
D3din[31:0]	0000000000

Comparison table

systems	area	power	delay
proposed	1242(LUTs)	0.384	9.530
extension	1298	0.439	4.449

II. CONCLUTION

In this proposed system DLAU is performed with the fast accelerator using deep neural network with the deep learning algorithm. This is increases the speed of accelerator 32 times more than the existing one. This proposed work is simple then the previous work so the accelerator using in timing based projects. The future work is optimize the memory or tiled matrix will increases the speed then the proposed work.

III. REFERENCES

1. LeCun, Y., Y. Bengio, and G. Hinton, Deep learning. Nature, 2015. 521: p. 436-444.
2. Hauswald, J., et al. DjiNN and Tonic: DNN as a service and its implications for future warehouse scale computers. in ISCA 2015.
3. Zhang, C., et al. Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks. in FPGA 2015.
4. Thibodeau, P. Data centers are the new polluters. 2014 [cited 2015].
5. Ly, D.L. and P. Chow, A high-performance FPGA architecture for restricted boltzmann machines, in FPGA 2009.
6. Chen, T., et al., DianNao: a small-footprint high-throughput accelerator for ubiquitous machine-learning, in ASPLOS 2014. p. 269-284.
7. Kim, S.K., et al. A highly scalable restricted Boltzmann machine FPGA implementation. in FPL 2009.