DESIGN AND IMPLEMENTATION OF LOW AREA AND HIGH SPEED MODIFIED DLAU ARCHITECTURE ON FPGA

B. Srividya*, y. David solomon raju**

Pg scholar*, associate professor & hod**

Department of ece, holy mary institute of technology and science bogaram(v), keesara(m), medchal district, 501301

ABSTRACT:

At present days, artificial intelligence plays an prominent role in the digital world. Machine learning and deep learning are used to solve the complex problems which are facing in the artificial intelligence. In artificial intelligence neural networks are the basic building blocks to operate any operation. Hence high speed and energy efficient deep learning neural networks are needed. To achieve this scalable deep learning accelerator unit (DLAU) is implemented for large scale architectures. The proposed DLAU used in the carry save adder for the calculation process and to verify the performance analysis. The experimental results shows that the proposed design achieves high speed and the performance is high compared to the other architectures.

I. INTRODUCTION

Deep Learning Accelerator (DLA) is a free, open architecture that encourages with its modular architecture a conventional way of designing deep learning inference accelerator. Machine learning has recently been commonly Used in cloud services and applications such as image search, face identification, speech recognition, etc. A subset of artificial neural networks has emerged since 2006 compared to traditional state-of - the-art algorithms to obtain higher accuracy and good results across a broad spectrum of machine learning applications. Deep learning is an effective multi-layer neural network computing and intensive memory. However, with Examples include the Google cat recognition scheme (1 trillion neuronal links) and the Baidu Brain scheme (100 trillion neuronal links) to increase the precision demands and the complexity of practical apps. The high performance application of large-scale deep-learning neural networks is therefore particularly important.

Convolutional neural network (CNN), a famous deep learning architecture extended from artificial neural network, has been considerably adopted in various programs, which encompass video surveillance, cellular robot vision, photograph search engine in information centers, etc. Inspired by using the conduct of optic nerves in living creatures a CNN design tactics statistics with a couple of layers of neuron connections to obtain excessive accuracy in image reputation. Recently, rapid increase of current programs primarily based on deep gaining knowledge of algorithms has in addition stepped forward research on deep convolutional neural network. Due to the specific computation sample of CNN, preferred cause processors aren't green for CNN implementation and may hardly ever meet the overall performance requirement. Thus, numerous accelerators primarily based on FPGA, GPU, and even ASIC design has been proposed currently to enhance overall performance of CNN designs. Among these strategies, FPGA primarily based accelerators have attracted an increasing number of interest of researchers because they have advantages of excellent overall performance, excessive strength efficiency, rapid improvement spherical, and functionality of reconfiguration.

As a main means to accelerate deep learning algorithms, FPGA (Field Programmable Gate Array) has high performance and low power consumption. It poses significant challenges to implement high performance deep learning networks with low power cost, especially for large-scale 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.

To tackle these problems, a scalable deep learning accelerator unit named DLAU to speed up the kernel computational parts of deep learning algorithms is presented. In particular, we utilize the tile techniques, FIFO buffers, and pipelines to minimize memory transfer operations, and reuse the computing units to implement the large size neural networks. This approach distinguishes itself from previous literatures with following contributions. The DLAU accelerator is composed of three fully pipelined processing units, including tiled matrix multiplication unit (TMMU), part sum accumulation unit (PSAU), and activation function acceleration unit (AFAU). Different network topologies such as CNN, DNN, or even emerging neural networks can be composed from these basic modules. Consequently, the scalability of FPGA-based accelerator is higher than ASIC-based accelerator.

II. RELATED WORK

Convolutional neural community (CNN) has been widely employed for image reputation due to the fact it may achieve excessive accuracy by means of emulating behavior of optic nerves in residing creatures. Recently, rapid growth of present day packages based totally on deep mastering algorithms has in addition stepped forward studies and implementations. Especially, numerous accelerators for deep CNN were proposed based totally on FPGA platform as it has advantages of excessive performance, reconfigurability, and rapid improvement spherical, etc. Although contemporary FPGA accelerators have established better overall performance over everyday processors, the accelerator layout space has now not been nicely exploited. One critical hassle is that the computation throughput may not nicely match the reminiscence bandwidth provided an FPGA platform. Consequently, existing approaches can not obtain great overall performance because of underutilization of either common sense resource or memory bandwidth. At the same time, the increasing complexity and scalability of deep gaining knowledge of programs aggravate this hassle. In order to conquer this trouble, C. Zhang et al proposed an analytical layout scheme using the roofline model. For any answer of a CNN layout, they quantitatively analyze its computing throughput and required memory bandwidth the usage of numerous optimization strategies, such as loop tiling and transformation.

Machine learning has become omnipresent in multiple areas of research and commercial apps over the previous few years and has produced adequate products. The advent of deep learning has accelerated the growth of artificial intelligence and machine learning. As a result, deep learning in study organizations has become a hot spot for studies. Deep learning generally utilizes a multi-layer neural high-level Features that combine low-level abstractions to identify distributed information functions to address complex machine learning issues. Deep Neural Networks (DNNs) and Convolution Neural Networks (CNNs) are currently the most commonly used neural profound learning models., Excellent ability to solve image recognition, voice recognition and other complex machine learning assignments has been demonstrated.

III. PROPOSED SYSTEM

DLAU architecture consists of a DDR3 memory controller, DMA module and DLAU accelerator, which are embedded in it. To communicate with the DLAU a programming interface is used with the users. In practical it transfers the data in the matrix format to the internal RAM blocks, which activate the DLAU accelerator and the output is obtained with the execution. It consists of three processing units those are

1. TMMU (Tiled Matrix Multiplication Unit)
2. PSAU (Part Sum Accumulation Unit)
3. AFAU

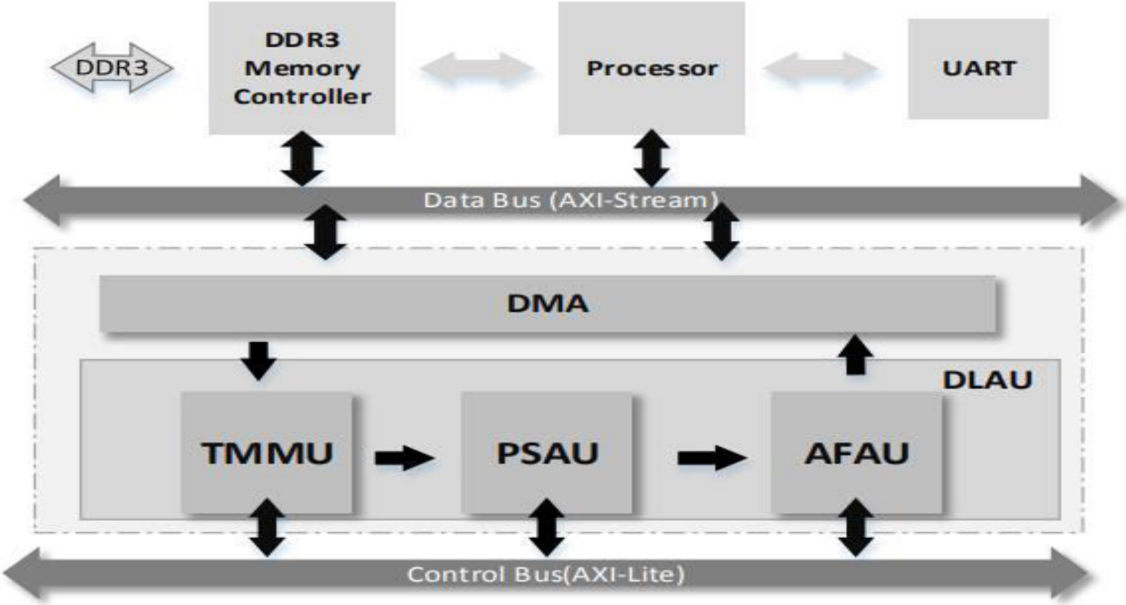


Figure 1: DLAU Accelerator Architecture

3.1 TMMU (Tiled Matrix Multiplication Unit) Architecture

In this TMMU block the multiplication of two inputs are done in the matrix format and the horizontal and vertical input lines are implemented according to the inputs. The rows and columns are placed outside the product matrix, these building blocks are either matrix top or bottom corner. The accelerator unit selects the weighted coefficients from the entry buffer and loops to save the amount of RAM. To reduce the data access time two separate registers are used to store the row and column data and the cache time of the data access is also gets reduced. Except for the first iteration the computation will start without any further delay. Hence with this TMMU architecture the delay of the data access time is gets reduced.

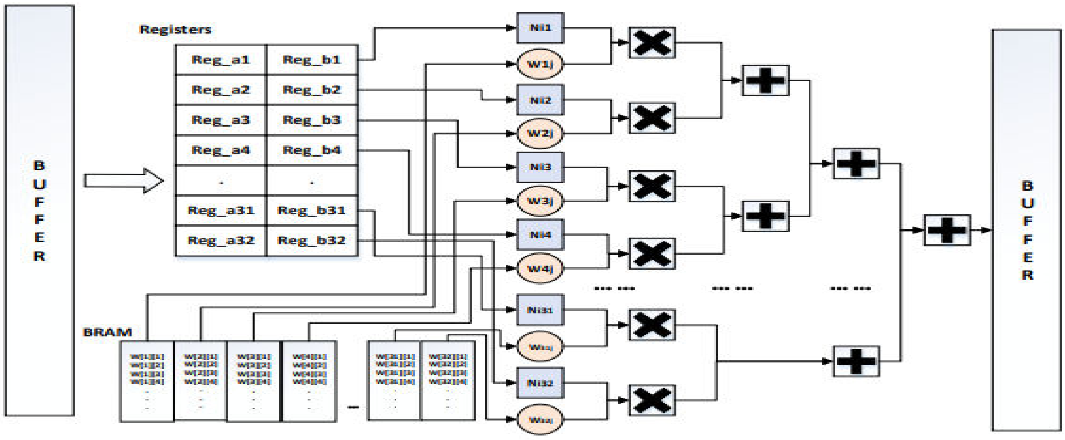


Figure 2: TMMU Schematic diagram

3.2 PSAU (Part Sum Accumulation Unit)

The data accessed and processed from the TMMU is passed to the PSAU for the accumulation of the inputs. In this block the data is accumulated using the high speed adders and the delay of the carry propagation is reduced by using the multiple designs. Hence, the PSAU architecture accumulates the complete sum and the data is drawn with the matched using PSAU.

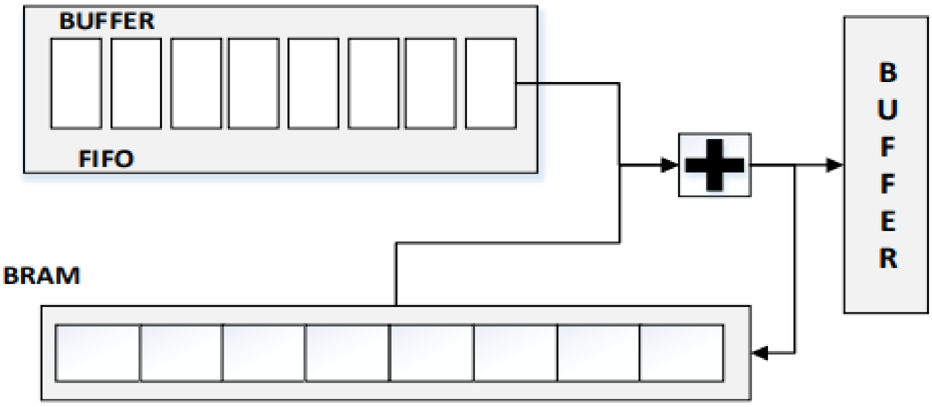


Figure 3: PSAU Schematic diagram

3.3 AFAU Architecture

The activation function is enabling when the data has to process and the activation functions with negligible accuracy data loss is achieved by the AFAU architecture. The data is passed in high speed architecture and the complete function is implemented in the sigmoid functions.

3.4 Modified PSAU

The addition process is replaced with the modified carry save adder in which three operands are added at the same time. Hence the propagation delay and power consumption of the adders are reduced and the speed of the design is increased.

The general structure of the carry save adder consists of two stages in which the sum and carry are generated separately and in the second stage the generated carry and sum are added by using high speed adders. Hence this can be extended to multi operand addition also in which the addition stages are increased with the input operands. The number of addition stages is equal to the input operands minus one, due to this addition stages the propagation delay is reduced. Hence the modified PSAU is replaced in the DLAU architecture and the performance results gives the best performance in terms of delay are area (in terms of LUT) while comparing with the previous architectures.

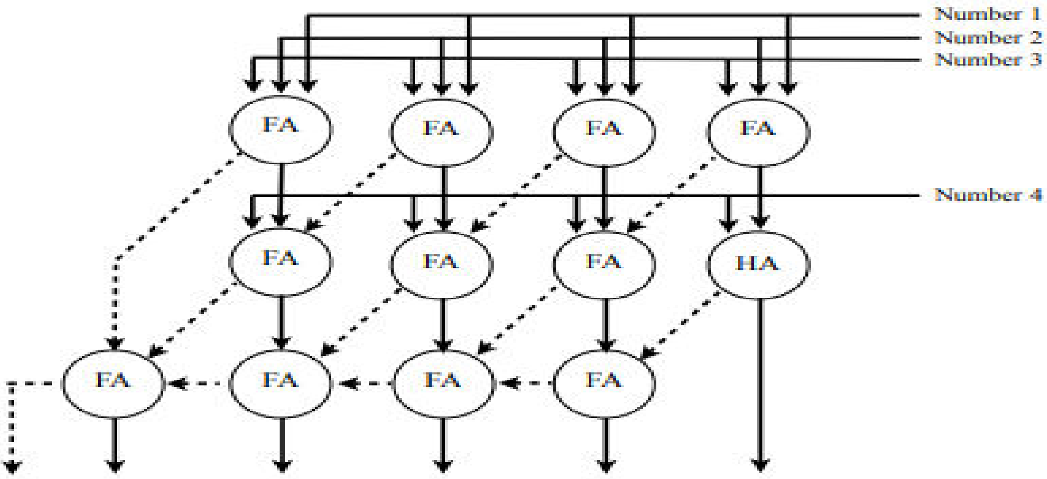
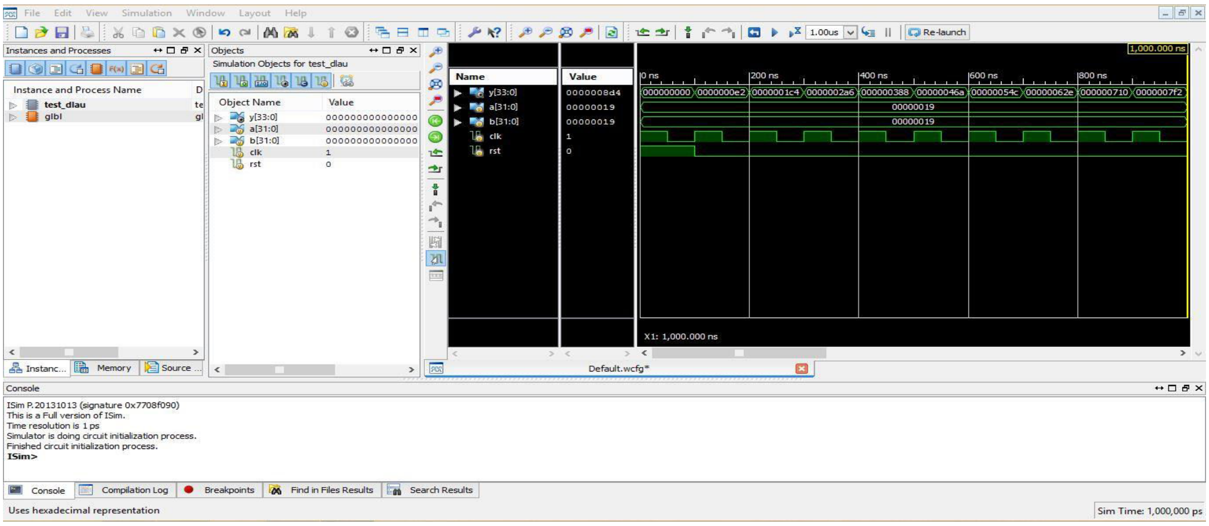


Figure 4: Modified PSAU architecture

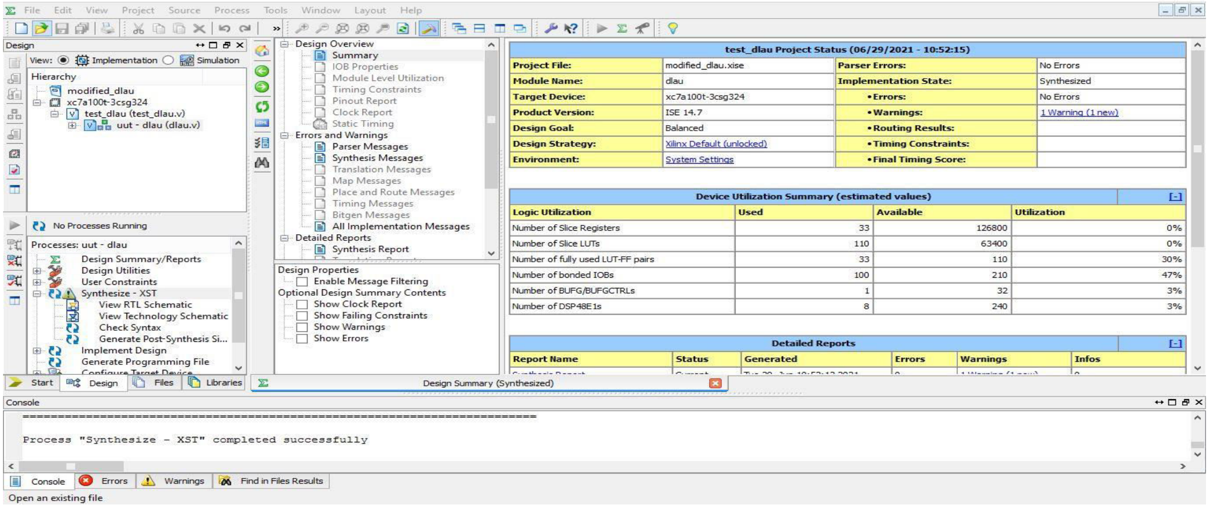
IV. RESULTS



```
ISim P.20131013 (signature 0x7708f090)
This is a Full version of ISim.
Time resolution is 1 ps
Simulator is doing circuit initialization process.
Finished circuit initialization process.
ISim>
```

Uses hexadecimal representation | Sim Time: 1,000,000 ps

Figure 5: Simulation result of the modified DLAU architecture



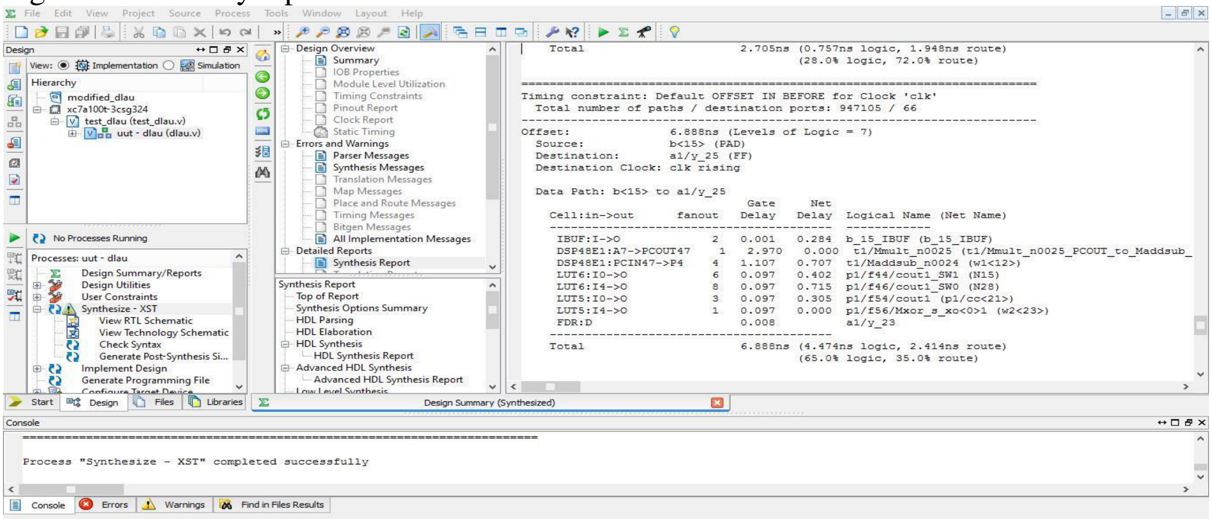
test_dlau Project Status (06/29/2021 - 10:52:15)			
Project File:	modified_dlau.xise	**Parser Errors:**	No Errors
Module Name:	dlau	**Implementation State:**	Synthesized
Target Device:	xc7a100t-3csg324	• **Errors:**	No Errors
Product Version:	ISE 14.7	• **Warnings:**	1 Warning (1 new)
Design Goal:	Balanced	• **Routing Results:**	
Design Strategy:	Xilinx Default (unlocked)	• **Timing Constraints:**	
Environment:	System Settings	• **Final Timing Score:**	

Device Utilization Summary (estimated values)			
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice Registers	33	126800	0%
Number of Slice LUTs	110	63400	0%
Number of fully used LUT-FF pairs	33	110	30%
Number of bonded IOBs	100	210	47%
Number of BUFG/BUFGCTRLs	1	32	3%
Number of DSP48E1s	8	240	3%

Report Name	Status	Generated	Errors	Warnings	Infos

```
Process "Synthesize - XST" completed successfully
```

Figure 6: Summary report of the modified DLAU architecture



```
   Total                      2.705ns (0.757ns logic, 1.948ns route)
                                      (28.0% logic, 72.0% route)

=========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 947105 / 66
-------------------------------------------------------------------------
Offset:              6.888ns (Levels of Logic = 7)
  Source:            b<15> (PAD)
  Destination:       a1/y_25 (FF)
  Destination Clock: clk rising

  Data Path: b<15> to a1/y_25
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O             2   0.001   0.284  b_15_IBUF (b_15_IBUF)
     DSP48E1:A7->PCOUT47    1   2.970   0.000  t1/Mmult_n0025 (t1/Mmult_n0025_PCOUT_to_Maddsub_
     DSP48E1:PCIN47->P4    4   1.107   0.707  t1/Maddsub_n0024 (w1<12>)
     LUT6:I0->O            6   0.097   0.402  p1/f44/cout1_SW1 (N15)
     LUT6:I4->O            8   0.097   0.715  p1/f46/cout1_SW0 (N28)
     LUT5:I0->O            3   0.097   0.305  p1/f54/cout1 (p1/cc<21>)
     LUT5:I4->O            1   0.097   0.000  p1/f56/Mxor_s_xo<0>1 (w2<23>)
     FDR:D                     0.008          a1/y_23
    ----------------------------------------
    Total                      6.888ns (4.474ns logic, 2.414ns route)
                                       (65.0% logic, 35.0% route)
```

```
Process "Synthesize - XST" completed successfully
```

Figure 7: Delay report of the modified DLAU architecture

Table 1: comparison table of the DLAU and modified DLAU architecture

	Number of LUT's Utilized	Delay in ns
DLAU Architecture	173	7.376 ns
Modified DLAU	110	6.888 ns

Architecture		

V. CONCLUSION

The proposed DLAU has the scalable and flexible deep learning accelerator based on FPGA. It consists of three pipelined units which are reused to improve the efficiency and reduce the power consumption. The proposed DLAU used carry save adder in the computation process and to improve the speed of the architecture. Experimental results show that the proposed DLAU achieves better performance in terms of delay compared to the previous architectures.

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.

[8] Qi Yu, et al. A Deep Learning Prediction Process Accelerator Based FPGA. CCGRID 2015: 1159-1162 [9] Jiantao Qiu, et al. Going Deeper with Embedded F