# FPGA Based, Supervised Learning Incorporation in Arithmetic and Logic Unit

Shubham Mishra , Shailendra Kumar Dhakad

**Abstract :** In this paper incorporation of machine learning(supervised learning) in arithmetic and logic unit has been described .This unit was targeted for VIRTEX -6(xc6vlx240t)FPGA using VHDL. The unit is capable of learning various arithmetic operations (trigonometric, polynomial, exponential mixed functions in one variable).The unit also has a capability to learn a series of operations from the learning sets. The unit was successful in achieving a learning rate of more than 90% for the tests (for at least 5 training sets) .Technique used for interpolation consumes ($2n^2$ – 4n ) number of multiplier blocks and (n) number of divider blocks where 'n' is the number of training sets.

.

**Index Terms**: Machine Learning, FPGA, Learning Set, Interpolation, Arithmetic Logic Unit.

.

## 1 Introduction

Supervised learning is the machine learning [1, 2,4] task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances.

In computing, an arithmetic and logic unit (ALU) is a digital circuit that performs integer arithmetic and logical operations. The ALU is a fundamental building block of the central processing unit of a computer, and even the simplest microprocessors contain one for purposes such as maintaining timers.

Incorporation of machine learning [1,2,4] at hardware level[11,12] can help in fast operation, low memory requirement and gives the ability to perform several operation without developing a separate circuitry for each.

## 2 Training Sets

In artificial intelligence or machine learning [1,2,4], a training set consists of an input vector and an answer vector, and is used together with a supervised learning method to train a knowledge database (e.g. a neural net or a naive bayes classifier) used by an AI machine. In statistical modeling, a training set is used to fit a model that can be used to predict a "response value" from one or more "predictors". The fitting can include both variable selection and parameter estimation. In order to learn a particular function, a training set will be required for building up the stretch. The accuracy of the function for further operation depends strictly on the number of feedings done though the accuracy of the device after a particular number of input data becomes stable i.e. after reaching that point providing more data sets will not improve the accuracy. This occurs when an accuracy of 100 percent is achieved .This point will be different for different function.

## 3 Interpolation

The mathematical field of numerical analysis, interpolation is a method of constructing new data points within the range of a discrete set of known data points. The data is obtained by sampling or experimentation, which represent the values of a function for a limited number of values of the independent variable. In the case of our device, data set is obtained from training sets and these sets are defined by the user.

There can be several methods of interpolation: linear interpolation, polynomial interpolation, Gaussian process, spline interpolation, Lagranges interpolation [3, 5, 10] etc. For our unit we have used Lagrange's interpolation [3, 5, 10]. The general form of Lagrange's interpolation [3, 5, 10] can be described as follows:

- *Shubham mishra is currently pursuing bachelors degree program in electronics and instumentation in BITS PILANI University, India E-mail: shubhammishra015@gmail.com .*
*Shailendra Kumar Dhakad is currently Lecturer (EEE/EnI) ,BITS PILANI University ,India shaildhakad09@gmail.com .*

If $x_0, x_1, . .. , x_n$ are n + 1 distinct numbers and f is a function whose values are given at these numbers, then a unique polynomial P(x) of degree at most n exists with $f(x_k) = P(x_k)$, for each k = 0, 1,. .. , n.
This polynomial is given by

$$P(x)=f(x0)\ Ln,0(x)+\ldots\ldots+f(xn)\ L\ n,n(x) = \sum_{k=0}^{n} (f(xk)Ln,k(x))$$

where,for each k =0,1,...,n,Ln,k(x)is de .ned as follows:

$$Ln,k(x)=\frac{\{(x-X_0)(x-X_1)\ldots(x-x_{k-1})(x-xk+1)\ldots(x-X_n)\}}{\{(X_k-X_0)(X_k-X_1)\ldots(X_k-Xk_{-1})(X_k-Xk+1)\ldots(X_k-X_n)\}}$$

$$= \prod_{\substack{i=0\\ i\neq k}}^{n} (X-X_i)/(X_k-X_i)$$

## 4. MODIFICATION IN THE ALGORITHM FOR HARDWARE IMPLEMENTATION

Since we are using supervised learning for our Arithmetic and Logic unit, so all points in the sample data set or the learning set will lie on a particular curve, in simpler terms they will form a function. Interpolating [3, 5,10] the data sets to the point where the value is to be calculated.

In order to implement this unit, the unit will store the training sets in a dual-port RAM. The data, from this RAM will be accessed for he calculation of Lagrangian polynomials, in a cyclic manner.This cycle will be controlled by a black box which will make surethe following aspects :

1) There will be 'n-1' such cycles where n is the number of points in the training set.

2) Every $n^{th\ cycle}$ will retrieve data from the RAM in a cyclic manner ,but will exclude the data from the $n^{th}$ memory address of RAM(*address start from 0 instead of 1).
3) Data retrieval from each memory location in every clock pulse.

The general form of Lagrangian polynomial will provide a better insight in the above mentioned functioning of black box.

$$L_n (x) = (X-D_k)/(D_n - D_k)$$

n k n k

Where $L_n(x)$ is the nth Lagrangian polynomial[3,5,10], Dk and Dn represent the data or the value stored in the RAM at kth and nth memory locations respectively.
In Fig.1 the graph describes the address locations (vertical axis) and time (horizontal axis).The implementation shown in the graph is for a sample space with 5 training data sets, thereby 5 Lagranges polynomials[3,5,10].The senders unit could be better understood by the following flow diagram :

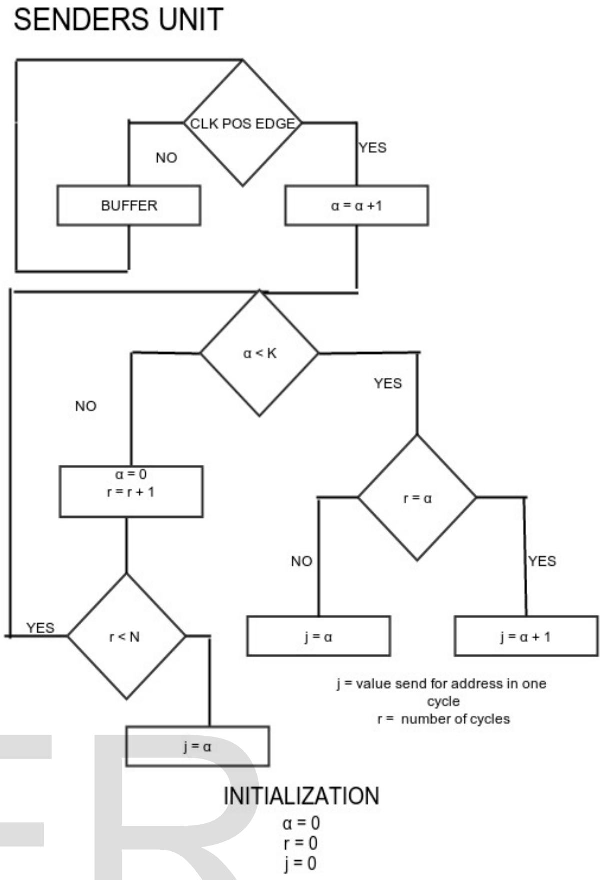
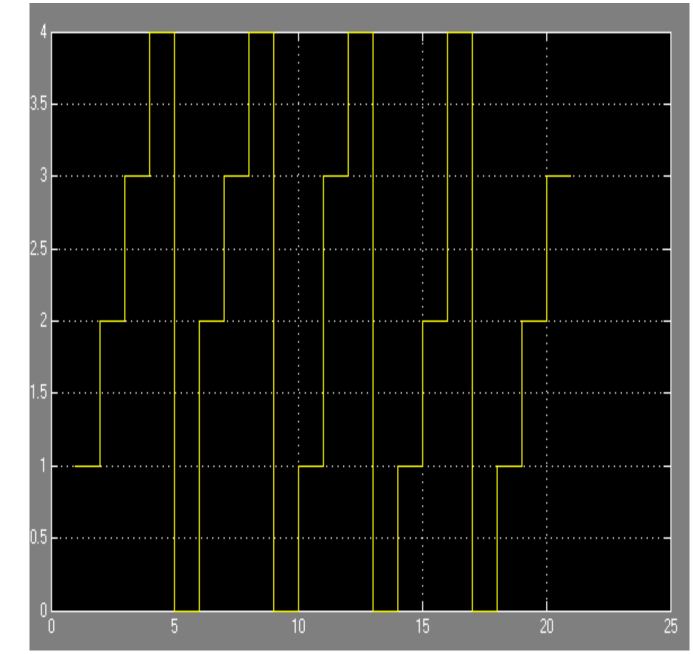


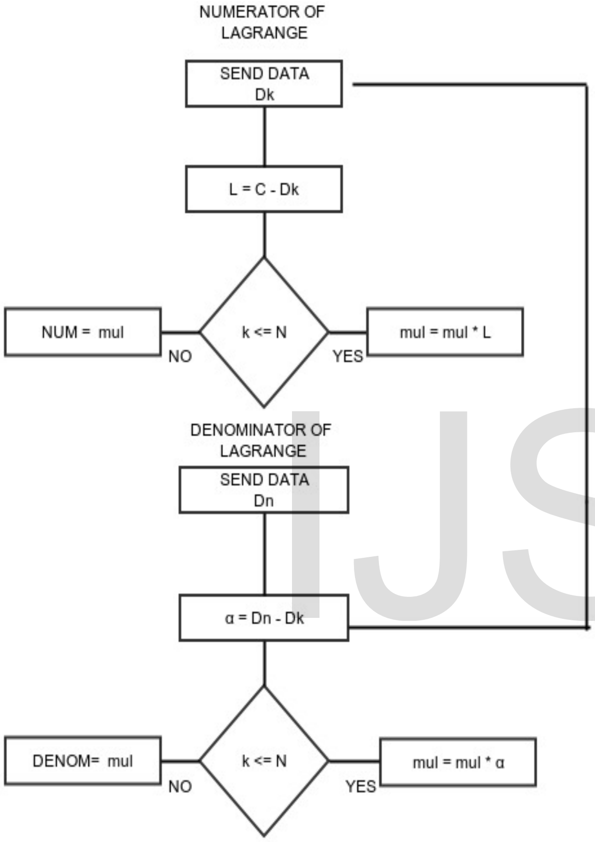
Figure 2.

These data sets once called are then taken up by the next unit



which calculates the values of the numerator and the denominators for the Lagrangian polynomial [3,5,10]. For the calculation purpose, a loop accumulation method is used. The loop starts with a constant value ('1') for one port of the multiplier (for both numerator and denominator) and the other port directly gets the value from the RAM out port. The other port with initial constant value is updated after one loop to the previous result of the multiplier. This cycle, after every n cycle, sends values further to the division unit whose output is the desired Lagrangian polynomial. This could be understood from fig 3. Which is the design structure of the Lagrangian polynomial [3,5,10] calculation unit.

Figure 3.



## 5. RESULTS AND IMPLEMENTATION

The unit was designed using VHDL, Xilinx ISE 12.3, Simulink (2013) and System Generator (14.3). The unit was targeted on xc6vlx240t-1ff1156. The unit was tested to 4 different functions of different nature with different number of training sets. The implementation results came satisfactory when in terms of learning ability.

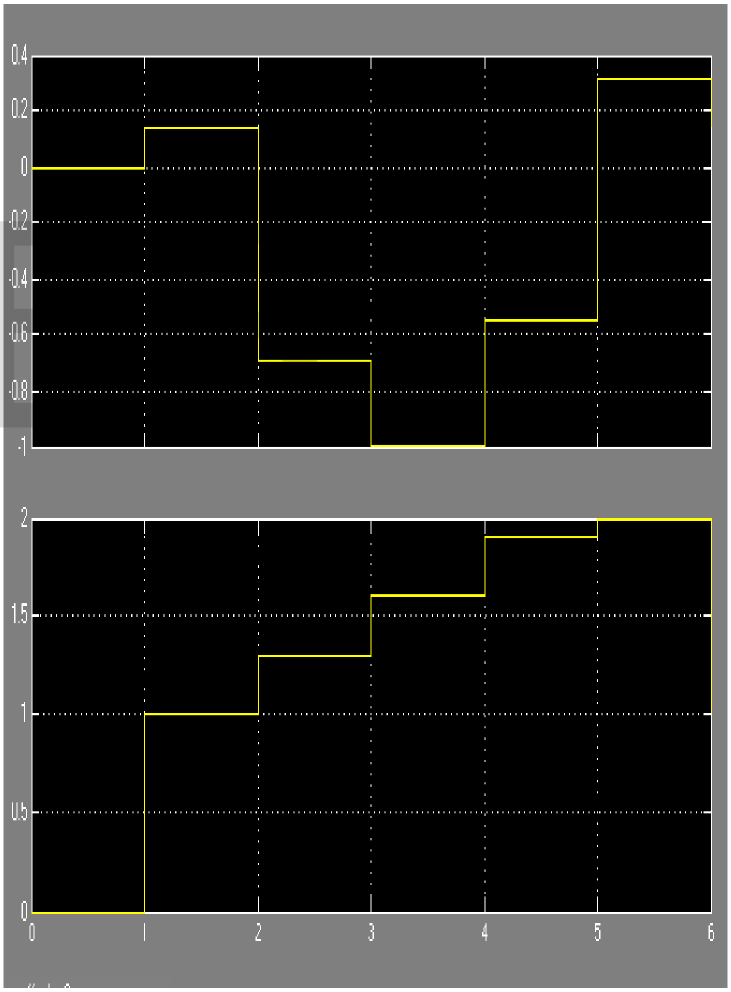
Table 1. Shows the implementation results of few trigonometric functions, polynomials, exponential functions and mixed functions. The learning rates came best for polynomials when compared to other functions.

Table 1

| Function | Number Of Training Sets (% error ) | | | | |
|---|---|---|---|---|---|
| | 4 | 5 | 6 | 7 | 8 |
| sin(3x) | 94.2% | 98.9% | 99.10% | 99.12% | 99.16% |
| $6x^6+x^5+78+x^4+91x^3+44+x+1$ | 46% | 58% | 89% | 100% | 100% |
| $e^x$ | 75% | 96.1% | 96.8% | 97.1% | 97.12% |
| $12xe^x+49$ | 75% | 89% | 94.1% | 98.2% | 98.4% |

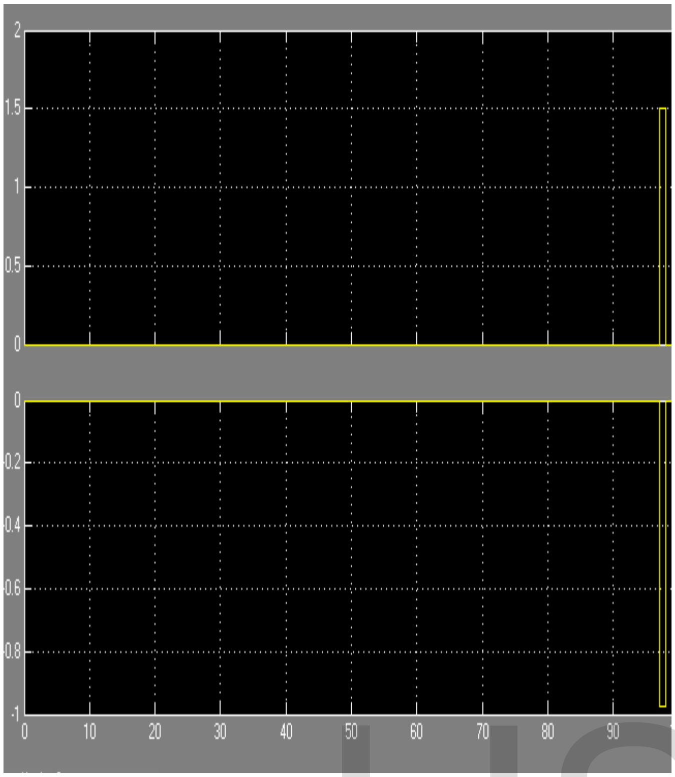
The simulation results of the unit to learn sin(3x) is shown. The results were obtained using System Generator and the simulation graphs using Matlab. Fig.4 shows the training 5 training sets for function sin(3x). The first graph is the values of the function and second graph represents the value of x at which the above graph shows the values.

Figure 4.



The output at the required x = 1.5 is shown in Fig 5 .The value predicted error within E ≈ 2 × $10^{-4}$. The first graph shows the point at which value is to be predicted and the second is for the predicted value.

Figure 5.



The number of look up tables(LUTs) utilized ,number of slices and F&F's used (it depends upon the number of training sets) are summarized in the Table 2. The number of LUTS, F&F`s and Slices increases with the increase in the number of trainining sets due to increase in the depth of the RAM, increase in the multiplication and division units due to the increase in the number of Lagrangian Polynomials[3,5,10].

Table 2.

| PARME-TRE | NUMBER OF TRAINING SETS | | |
|---|---|---|---|
| | 4 | 5 | 6 |
| LUT's | 568 | 591 | 623 |
| F&F's | 515 | 535 | 568 |
| Slices | 561 | 581 | 603 |

Table 3. Describes the description of the increase in the multiplication and division process with the increase in the number of Training sets. It also shows the generalization for $n^{th}$ training set.

Table 3.

| Unit | TRAINING SETS | | | | |
|---|---|---|---|---|---|
| | 4 | 5 | 6 | 7 | n |
| Multiplication | 16 | 30 | 48 | 70 | $2n^2 - 4n$ |
| Division | 4 | 5 | 6 | 7 | n |

## 6 CONCLUSION AND FURTHUR RESEARCH

The unit was successful in learning various types of functions which included trigonometric, polynomial, exponential and mixed functions. The device even showed satisfactory learning rates which open the scope for further advancements for explicit functions, implicit in more than one variables etc. The unit with sufficient training sets, can be used to save large implementation spaces. What earlier was to be done for every function individually can be done by a single unit by varying the training sets.

## REFERENCES

[1] Kulkarni, P. "Introduction to Reinforcement and Systemic Machine Learning" ,Wiley-IEEE Press ,1 - 21

[2] Machine Learning May 2000,SPRINGER,Volume 39,Issue 2-3,pp 169-202 ,"Machine Learning for Information Extraction in Informal Domains" Dayne Freitag.

[3] "Lagrange interpolation on Leja points" by Rodney Taylor.

[4] L. Hubert and P. Arabie, "Comparing Partitions," *J. Classification,* vol. 2, no. 4, pp. 193-218, Apr. 1985. (Journal or magazine citation) "Introduction to Machine Learning" by Alex Smola and S.V.N. Vishwanathan

[5] "Engineering Mathematics", Volume 2E. ,by Rukmangadachari.

[6] Simulink® Getting Started Guide R2013b

[7] System Generator for DSP User Guide UG640 (v 14.3) October 16, 2012

[8] ISE In-Depth Tutorial UG695 (v 12.3) September 21, 2010.

[9] "VHDL (Computer hardware description language) "by Douglas L.Perry

[10] "Interpolation Processes: Basic Theory and Applications "By Giuseppe Mastroianni, Gradimir Milovanovic.

[11] Perkowski, M.Jozwiak, L,Lewis, T . "Learning in hardware: architecture and implementation of an FPGA-based rough set machine" Proc. of IEEE EUROMICRO Conference ,326 - 334 vol.1 .

[12] del Campo, S.M.Albertsson, K.;Nilsson, J.;Eliasson, J.;Sandin, F. "FPGA prototype of machine learning analog-to-feature converter for event-based succinct representation of signals" IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2013 ,Page(s):1 - 6 .