UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Faculty of Engineering, Built Environment and Information Technology

EES 424

Research project

Initials & Surname	Student Number	Signature
Maaz Harris Khan	14346011	MHKhan

Implementation of The Sigmoid Function on a FPGA

I. Introduction

With The Turn of the 21st century artificial intelligence (AI) and artificial neural networks (ANN) are becoming quickly an integral part in our everyday lives from Tesla's self-driving cars to simple smart home IOT appliance. With this surge come along the need for rapid prototyping AI and ANN. One of the core components of these systems is the activation which is represented by the sigmoid function. ANNs form part of a filed in AI that designed to model the behaviour of the human brain which raises the concept of deep learning[1], as well as the multi-layer perception these include but are not limited to speech recognition and image identification and calcification[2]. These systems can be implemented on a software level however research has shown software based neural networks suffer from slower execution time compare to there hardware level counterparts in real time applications[1].

One of the key changes engineers face when synthesizing artificial neural network based on FPGA is the approximation of the activation function which in this case is the sigmoind as sated prior[3]. There have been multiple studies with regards to the optimum implementation of the sigmoid function. The A-Law approach[4] is a pairwise linear approximation used to implement a pulse code modulation which the authors suggest can be implemented with a using a small number of logic gates. An other approach to the activation is proposed by C.Alippi approximation is base on selecting an integer set breakpoints and setting the y-values of these integers as the power of 2 integers[5][6]. The piecewise linear approximation of a nonlinear function (PLAN) approximation proposed by Amin, Curtis and Hayes–Gill. deploys digital gates to directly transform values from x to y[7]. This approximation uses the x as the in put an y the approximated sigmoidal output.

A piecewise second order approximation was proposed to to approximate the sigmoid function, with this method comes greater performance and lower implementation cost. The method require one multiplication and no lookup table as suggested by the author[8]. The second order approximation scheme substantially improves approximation of higher order piece wise functions. The scheme requires no memory and has significantly low delay and error rate.

This literature study will investigate the methods to implement these methods on a FPGA as well as take a look it the advantages and trade offs of each of the motioned approaches, as well the logical element cost of each of the approaches.

II. Literature Review

Sigmoid activation functions are used in feed forward neutral netorks. The sigmoid function is monotonic growing and differentiated.[9]. The sigmoidal function is defined by Eq.1

$$f(x,k,b,T,c) = k + \frac{c}{1+be^{Tx}}, \forall x \in R. \tag{1}$$

$$Where\ k \in R, b \in R^{+}, T, c \in R \backslash \{0\}$$

classical sigmoid function ($k = 0, c = b = 1, T = 1$):

$$f(x) = \frac{1}{1+e^{-x}}. \tag{2}$$

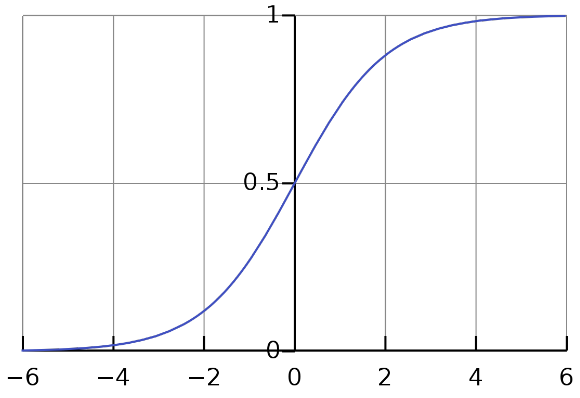

Fig. 1: The graphical representation of Eq.1

A-Law Approach

With this proposed method of implementation to achieve the modified sigmoid function each linear segment is implied to be the power of 2 this new curve has only seven segments[3]. Splitting the curve this way allows to replace multipliers with shift registers. The fig.2 below is the hardware inflammation proposed by [10]. It can be seen in fig.2 require 4 shift registers, 2 multiplexers and a single sum block. The MCode block is for compare functions. It is also stated by the author that "All the used blocks are part of the System Generator library Xilinx Integrated Software Environment". The results obtained by [10] where as follows, the maximum error introduced by A-law approximation was found be 5.63% and the mean error is 0.6335%.

TABLE I: Break points of the A LAW approximation

x	-8.0	-4.0	-2.0	-1.0	1.0	2.0	4.0	8.0
y	0.0	0.0625	0.12	0.25	0.75	0.87	0.937	1.0

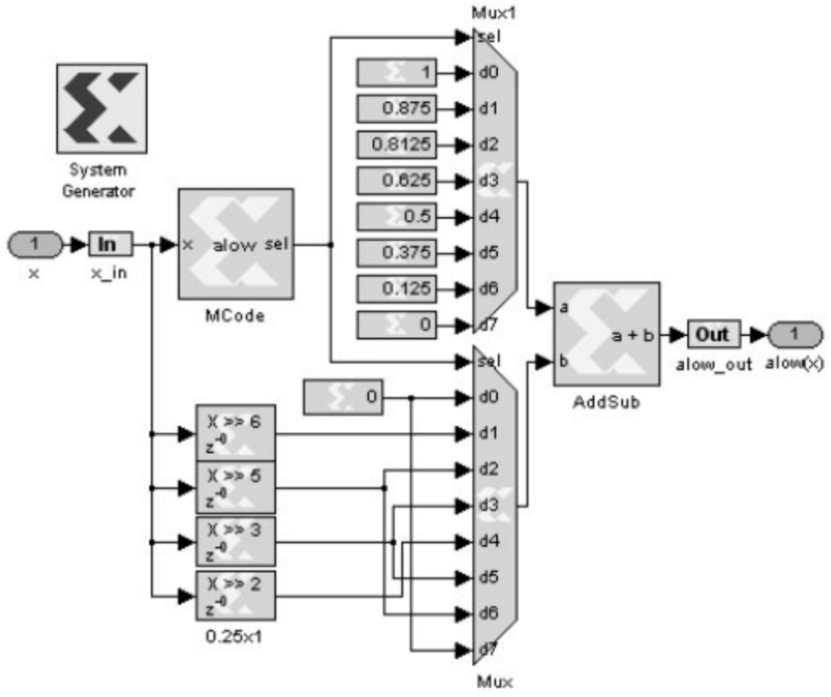

Fig. 2: Hardware architecture of the A Law approximation[10]

Alippi and Storti-Gajani model

This approximation of this scheme depends on selecting the set of breakpoints of the first derivative and setting up the function as the sums of powers similar to the A-Law approach. Hence the sigmoid function will be symmetrical at point $x = 0, y = 0.5$, therefore only half the pair need to be determined thanks to the symmetrical nature[5].

$$y_{x>0} = 1 - y_{x\leq 0}. \tag{3}$$

Considering only the negative half of the function on the negative x-axis we can define the integer part of x, n[6] can be written as the following function:

$$n = |(x)| + 1. \tag{4}$$

The decimal part of x can express as the equation below

$$\hat{x} = x + |(x)|. \tag{5}$$

The C.Alippi and G.Storti-Gajani scheme can be implemented with 8 shift registers for dividing $\hat{x}$ into 4 signals for shifting with the number of x bits. 4 Multiplexers, subtraction blocks, 2 slice blocks and 1 cooperator block[10] this implementation can be seen in fig.3. The author stated "The numerical representation used for the analysis is 32 bits with 16 bits binary point representation induces a maximum of 1.89 $\%$" and a mean error of 1.11$\%$[10]

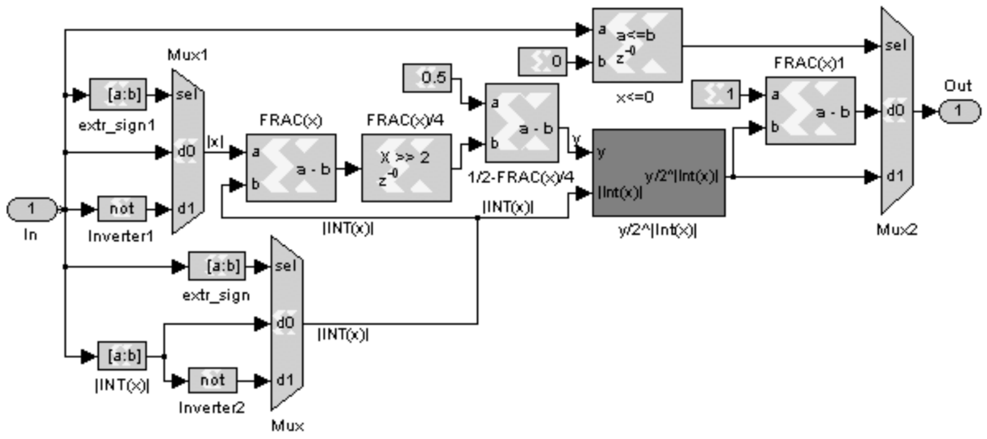

Fig. 3: Hardware architecture of the Alippi function approximation[10]

It should made know to the reader in the above figure $\hat{x} = FRAC(X)$ and $|x| = |INT(x)|$

Piecewise Linear Approximation of a Nonlinear function (PLAN)

The PLAN is a efficient piecewise approximation that can be implemented using digital gate to transform from X to Y. This approximation is used in the outputs of neural networks to perform approximation[7]. Due to the nature of the PLAN calculation are only need to be performed on the absolute value of the input x[5]. The one main advantage of the scheme is that instead of multiplication we use shift operation[10]. This due to the PLAN efficiently choosing the gradient of the line to eliminate the need for multipliers[7]. The representation induces a maximum error of 1.89 $\%$ and a mean error of 0.63$\%$.

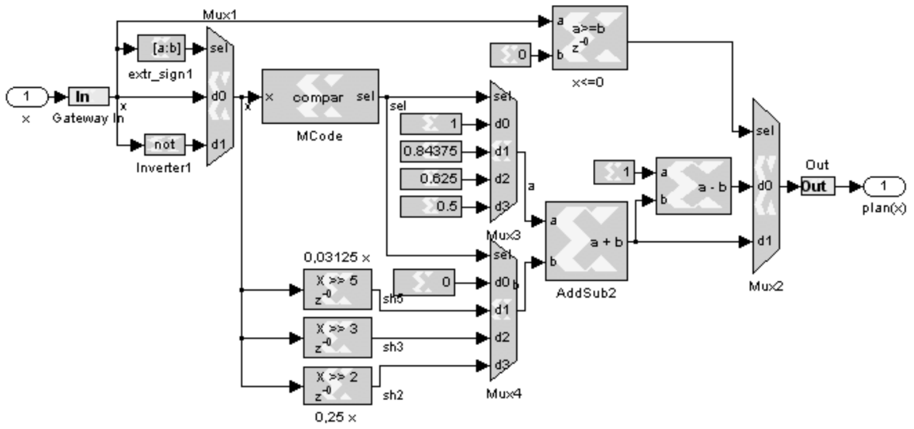

Fig. 4: Hardware architecture of the PLAN function approximation[10]

Piecewise second-order approximation

The sigmoid can also be realised with a oiece wise second order approximation, the follow is formula for this approximation[8].

$$y(x) = c_0 + c_1 * x + c_2 * x^2. \tag{6}$$

When we take a closer look at the above function we can see that on clear drawback the need for multiplication[5]. Zhang suggests scheme requiring one multiplier, in the range [-4,4] that sigmoid is computes as follows:

$$y = \begin{cases} 2^{-1} * \left(1 - |2^{-2} * x|^2\right) & -4 < x < 0 \\ 1 - 2^{-1} * \left(1 - |2^{-2} * x|^2\right) & 0 \leq x < 4 \end{cases} \tag{7}$$

From simplification of the above piecewise function, it can be implemeted with one multiplier, two shift registers and two XOR gates this can be seen in fig.5. It has been reported in [10] that there are there way to implement the second order approximation however, these require upto tree time more hardware resources[8].

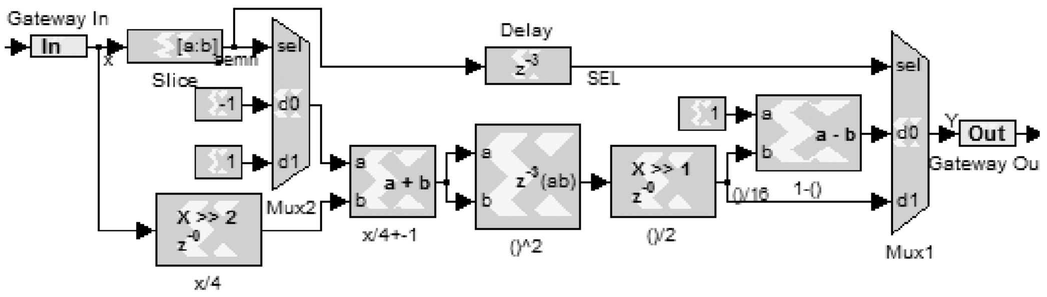

Fig. 5: Hardware architecture of the Zhang function approximation[10]

TABLE II: Erros and resources utilization of each method of approximation [10]

Approximation function	Maximum error (%)	Mean error (%)	Total equivalent gates count for design
A-law	5.63	0.63	411
Allipi	1.89	1.11	877
Plan	1.89	0.63	351
Zhang	2.16	1.10	314

Interpreting the results in the above table we can see that the Plan method has the lowest mean at a rate, followed by A Law which also has the same mean at a rate however has a significantly larger maximum average. Allipi and Plan have the same maximum and rate however plan has a lower mean at a rate and also requires less total gates for the design. Zhang falls in the middle of these results not having a terrible maximum at a rate and also not having good mean at a rate however, Zhang requires the least number of equivalent gates.If we disregard the maximum error percentage for A Law it presents the simplest hardware implementation method from the study. However, if the maximum error is of concern than the second-best implementation from simplicity standpoint would be Zhang approach.

References

[1] Z. Li, Y. Huang, and W. Lin. Fpga implementation of neuron block for artificial neural network. In *2017 International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, pages 1–2, 2017.

[2] P. W. Zaki, A. M. Hashem, E. A. Fahim, M. A. Masnour, S. M. ElGenk, M. Mashaly, and S. M. Ismail. A novel sigmoid function approximation suitable for neural networks on fpga. In *2019 15th International Computer Engineering Conference (ICENCO)*, pages 95–99, 2019.

[3] Z. Xie. A non-linear approximation of the sigmoid function based on fpga. In *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI)*, pages 221–223, 2012.

[4] D. J. Myers and R. A. Hutchinson. Efficient implementation of piecewise linear activation function for digital vlsi neural networks. *Electronics Letters*, 25(24), 1989.

[5] M. T. Tommiska. Efficient digital implementation of the sigmoid function for reprogrammable logic. *IEE Proceedings - Computers and Digital Techniques*, 150(6):403–411, 2003.

[6] Cesare Alippi and Giancarlo Gajani. Simple approximation of sigmoidal functions: realistic design of digital neural networks capable of learning. pages 1505 – 1508 vol.3, 07 1991.

[7] H. Amin, K. M. Curtis, and B. R. Hayes-Gill. Piecewise linear approximation applied to nonlinear function of a neural network. *IEE Proceedings - Circuits, Devices and Systems*, 144(6):313–317, 1997.

[8] Ming Zhang, S. Vassiliadis, and J. G. Delgado-Frias. Sigmoid generators for neural computing using piecewise approximations. *IEEE Transactions on Computers*, 45(9):1045–1049, 1996.

[9] I. Tsmots, O. Skorokhoda, and V. Rabyk. Hardware implementation of sigmoid activation functions using fpga. In *2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM)*, pages 34–38, 2019.

[10] Alin Tisan, Stefan Oniga, Daniel Mic, and Buchman Attila. Digital implementation of the sigmoid function for fpga circuits. *ACTA TECHNICA NAPOCENSIS Electronics and Telecommunications*, 50, 01 2009.