SEVENTH FRAMEWORK PROGRAMME
FP7-ICT-2011-C

Grant agreement for: Collaborative project

Deliverable D3.1

Low-power algorithm and hardware architecture for classification of external stimuli from electrical response of plants

Project acronym: PLEASED
Project full title: " PLants Employed As SEnsor Devices "
Grant agreement no: 296582

Responsible Partner: University of Southampton
Version date: 20/05/2015

Contents

1 Introduction

In this report we will present the VLSI implementation of a linear classifier on FPGA which is one of the major outcomes from the theoretical works carried out with single plant as described in D2.1. The focus of this report is primarily on the hardware architecture for linear discriminant analysis (LDA) based binary classifiers. To be noted that the final results described in D2.1 show that while LDA performs very well for classification of the external stimuli, the best classification performance could be achieved by Mahalanobis distance based minimum distant classifier. However we arrived at that result only very recently. Since the initial results indicated that LDA shows good classification performance, we concentrated on implementation of LDA based classification scheme which is described here. The classification between two external stimuli – H2SO4 and O3, based on the features extracted from raw electrical signals from plants [4], were used as a template for this implementation. Furthermore, we would present, in this report, an analysis of the optimal datapath wordlength selection (total number of bits associated with a fixed point representation) to minimise the effects of errors accumulated during the arithmetic processing steps in the linear classifier. To this end we have carried out theoretical analysis to investigate the effects of finite precision fixed point representation as well as truncation error (resulting from mathematical operations) on classification performance. Using this analysis we have found the aforementioned optimal wordlength representation for the LDA classifier. Although our analysis is based on plant signal classification, our results could be potentially generalized for any linear classifier targeting a resource constrained hardware platform.

1.1 Motivation

The main idea of PLEASED is to deploy a device in the field that can monitor the electrophysiological response of the plants and classify the external stimuli that may have caused it. The notion of long-term monitoring automatically introduces the constraint of low-power operation of the system in resource constrained sensor node. The implementation of a system on a resource constrained hardware platform must take into account several important factors including power consumption and complexity of the system regarding processing times [1], [2]. In such a scenario, a fixed point representation of all the variables involved has its own merits over floating point representation – operators, memories and buses require lesser area along with lower power consumption [1], [2]. However with fixed point representation, an arithmetic error gets introduced in the computation at every step which gets accumulated. If this error is large, it has possibility to alter the result significantly. Therefore, an optimal wordlength representation has been an area of analysis for all systems which needs to be implemented on hardware [1], [2]. Since the error propagated due to fixed precision of wordlength varies from system to system (or data to data), there may not be one solution to fit all. However, if a detailed mathematical analysis for a generalized wordlength representation for a particular system (algorithm) type is explored, then such analysis could be used to extend the solution to other similar systems. Implementation of an LDA based classification system on hardware has been reported in [1], [2] where the authors have discussed fixing the word length during computation of the weights (of the corresponding features) while training. However, training the LDA classifier online requires

much more power and computation time.

The work presented in this report revolves around training the classifier *offline* - on a standalone PC using MATLAB [3], to find out optimum features giving best classification accuracy (between two classes) and obtain their corresponding weights. Thereafter we have explored the optimum fixed word length of the corresponding weights which are thus obtained. By training the classifier offline, we reduce the complexity of computations *online* on hardware platform. However the question arises on the classification accuracy when dealing with unknown test data (prospective study) using fixed word length representation for feature and weights computed offline. This is explored in this report to validate the method of finding the weights of optimal features offline and then fixing their word length representation. Thus the motivation for this analysis of the optimum fixed word length for implementation of an LDA classifier is to realize a plant electrical signal based external stimuli sensing platform. However such a fully functional platform will need to operate as an individual standalone device and hence must have low power consumption during data acquisition, feature extraction and classification. Therefore it becomes necessary to carry out all required operations in fixed point. The analysis of fixed point computations during classification part is presented in the subsequent sections.

1.2 Features for Analysis

[4] reports the experimental setup used to extract the electrical signal from a variety of plants after application of different external stimuli. It also reports the 11 statistical features which were extracted from the raw plant electrical signal for classification. We see that using just Hjorth complexity along with LDA was the most discriminating (between all pairs of stimuli) individual feature. It also provided a classification accuracy of 66.9% between H_2SO_4 and O_3. Although, using Mahalanobis distance based classifier improves this classification accuracy to around 81%, we focused on analyzing the implementation of LDA based classification since using a feature pair of Interquartile range and variance with LDA provided a classification accuracy of around 85% (between H_2SO_4 and O_3). Additionally the complexity of LDA is much lower than a Mahalanobis classifier [6] which makes it easier to implement an LDA on hardware.

Analysis of the optimum word length in this report was carried out using all the 11 features mentioned in [4] which provided a classification accuracy of over 95% (when using floating point computation in MATLAB).

2 Theoretical analysis

Let n and f be the number of bits in the integer and fractional parts of a binary number representation respectively.

If we consider the sign bit, then the total number of bits in the integer part becomes $n-1$.

Thus the decimal representation is given by the following equation

$$X_{10} = -a_{i=n-1} \cdot 2^{n-1} + \sum_{i=n-2}^{0} a_i \cdot 2^i + \sum_{i=-1}^{-f} a_i \cdot 2^f \qquad (0.1)$$

in which the first term represents the sign bit, the second term represents the integer part and the third term represents the fractional bit.

2.1 Error Bounds for fixed point representation

Maximum value which can be represented

The maximum decimal number that can be represented in fixed point using n number of bits for the integer part and f number of bits for the fractional part (having accuracy of representation as 2^{f}) is given by

$$X_{10\ max} = (2^{n+f-1} + 1) / 2^{f} = 2^{n-1} - 2^{-f} \quad (0.2)$$

Minimum value which can represented

Correspondingly, the minimum represented number with the aforementioned fixed point representation is

$$X_{10\ min} = -2^{n+f-1} / 2^{f} = -2^{n-1} \quad (0.3)$$

Representation error

If we are representing a decimal number in binary form with f fractional bits, then the error we are introducing during the binary representation is

$$\varepsilon = 2^{-f} \quad (0.4)$$

Therefore the difference in the original value and quantized value is given as

$$X_{rep} = Q\ X\ = X - \varepsilon \quad (0.5)$$

where $Q\ X$ is denoting the quantized form of x .

Let us now see how this affects the results when computing Linear Discriminant Analysis (LDA) for classification of two groups.

2.2 Error propagation during Linear Discriminant Analysis (LDA) computation

The general expression for LDA is given by the following equation [5], [6]

$$y = \sum_{i=1}^{k} W^{T}_{i} \cdot x_{i} \quad (0.6)$$

Where, y is the projected single dimension scalar value whose distance from the user defined threshold predicts the class to which the feature (independent variables) values, x_i belong to. W_i is the optimal weight vector, whose elements corresponds to each of the features, providing best separation between the two classes.

Let us now look at two cases of feature values – Standardized / Z score (0.7) and Normalized (0.8)

$$x_{standardized} = \frac{x - \mu}{\sigma} \tag{0.7}$$

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{0.8}$$

where x_{min} and x_{max}, μ and σ are the minimum and maximum values, mean and standard deviation of the features respectively.

Standardized feature values can both be negative and positive, whereas normalized values are always positive and lie between 0 and 1.

Let us now proceed further with the case for normalized feature values. Here two situations may occur which are discussed below.

Normalized feature values, Case 1: All W_i and x_k are at their maximum values. In this case we consider $W_i \leq \infty$

$\therefore$ From (0.6) we get

$$y = W_1 \cdot x_1 + W_2 \cdot x_2 + W_3 \cdot x_3 + \ldots + W_k \cdot x_k \tag{0.9}$$

The equation (0.9) has k multiplications and $k - 1$ additions.

Let $k = 2$, then (0.9) becomes

$$y = W_1 \cdot x_1 + W_2 \cdot x_2 \tag{0.10}$$

By introducing representation error for each feature and corresponding weights, we get

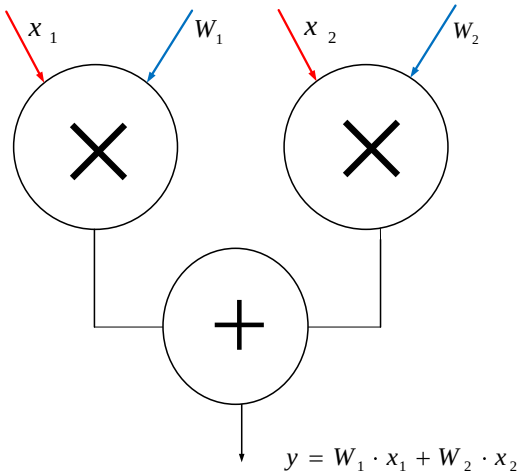
$$y' = Q\ y = W_1 - \varepsilon \cdot x_1 - \varepsilon + W_2 - \varepsilon \cdot x_2 - \varepsilon \tag{0.11}$$

$\therefore$ Total error, ignoring the higher order terms in ε is given by,

$$E_{total} = y - Q\ y = W_1 \cdot x_1 + W_2 \cdot x_2 - W_1 - \varepsilon \cdot x_1 - \varepsilon + W_2 - \varepsilon \cdot x_2 - \varepsilon \tag{0.12}$$

or,

$$E_{total} = \varepsilon\ x_1 + x_2 + W_1 + W_2 + 2\varepsilon_{truncation} \tag{0.13}$$



where $\varepsilon_{truncation}$ is the error introduced in the products, during each multiplication, due to representation with reduced word lengths.

Therefore generalizing from (0.13) for k features, we get

$$E_{total^+} = \sum_k x_k \varepsilon + \sum_k W_k \varepsilon + k \cdot \varepsilon_{truncation} \tag{0.14}$$

Now, substituting for W its maximum representation $x_{10\ max}$ and since the features always lie between $[0,1]$ we can further substitute $x_k = 1$ in (0.14). Furthermore we define the representation as well as truncation error as $\varepsilon = \varepsilon_{truncation} = 2^{-f}$ in (0.14). Thus now with all these substitution we have the total error E_{total^+} for maximum weight and feature values given by

$$E_{total^+} = k\,2^{-f}\ \ 2 + 2^{n-1} - 2^{-f} \tag{0.15}$$

Normalized feature values, Case 2: All W_i and x_k are at their minimum values. In this case we consider $W_i \geq -\infty$

Again referring to (0.10), for $k = 2$, we get

$$y = \ -W_1 \ \cdot x_1 + \ -W_2 \ \cdot x_2 \tag{0.16}$$

and,

$$Q\ \ y \ = \ -1\ \ W_1 - \varepsilon \ \ \cdot \ x_1 - \varepsilon \ + \ -1\ \ W_2 - \varepsilon \ \ \cdot \ x_2 - \varepsilon \tag{0.17}$$

or,

$$Q\ \ y \ = - \ \ x_1 \cdot W_1 + x_2 \cdot W_2 \ + \varepsilon \ \ x_1 + x_2 + W_1 + W_2 \ - 2\varepsilon^2 - 2\varepsilon \tag{0.18}$$

For a small value for ε^2, it can be ignored. Therefore, the expression for the total error in this case is given by

$$E_{total^-} = y - Q\ \ y \ = \ -x_1 \cdot W_1 - x_2 \cdot W_2 \ - Q\ \ y \tag{0.19}$$

or,

$$E_{total^-} = -\varepsilon \quad x_1 + x_2 + W_1 + W_2 \quad - 2\varepsilon_{truncation} \tag{0.20}$$

Therefore generalizing from (0.20) to k features, we get

$$E_{total^+} = -\sum_k x_k \varepsilon - \sum_k W_k \varepsilon - k \cdot \varepsilon_{truncation} \tag{0.21}$$

Now, substituting for W its minimum representation $x_{10\ \min}$ and since the features always lie between $[0,1]$ we can further substitute $x_k = -1$ (corresponding to the minimum value of 0 and considering the sign bit as -1) in (0.21). Furthermore we define the representation as well as truncation error as $\varepsilon = \varepsilon_{truncation} = 2^{-f}$ in (0.21). Thus now with all these substitution we have the total error E_{total^-} for maximum weight values given by

$$E_{total^-} = -k\,2^{-f}(2 + 2^{n-1}) \tag{0.22}$$

So far we have found out the two boundaries of the total fixed point representation and truncation errors depicting two possible cases of normalized features where the all weights could either are positive or negative. Now these are the two corner cases and generally the intermediate error should lie between these two values. Thus we have the intermediate error $E_{total\ intermediate}$

$$E_{total^+} \geq E_{total\ intermediate} \geq E_{total^-} \tag{0.23}$$

2.3 Error during binary classification

So far, we have calculated the possible error while computing the discriminant function y with fixed point representation. Let us now investigate how this error affects the classification accuracy and how to limit the mis-classification rate.

Considering a binary classification problem with $k = 2$.

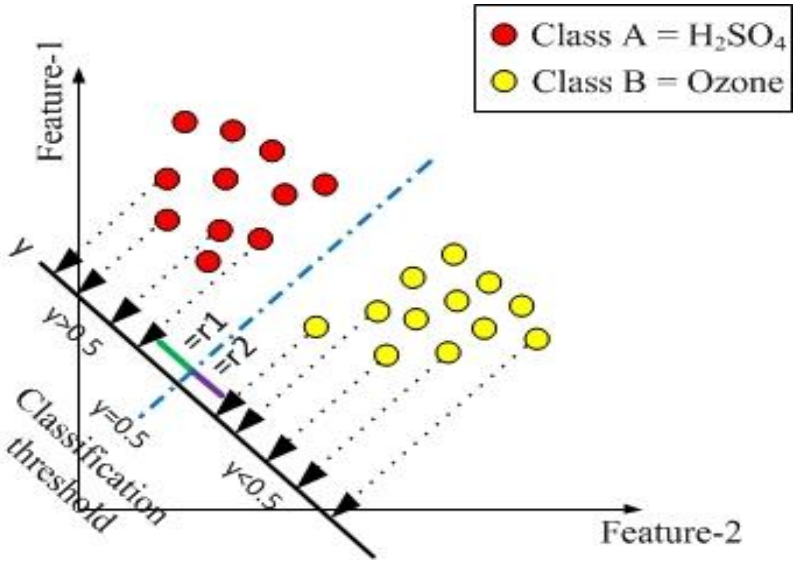


Figure 1: Two feature binary classification problem

As we can see from Figure 1, the mis-classification can occur if the error propagation exceeds the distance $r1$ (or $r2$) which are the distance of the members of class A (and class B) closest to the user defined decision boundary on the lower dimensional projected plane. These distances $r1$ and $r2$ will depend on the data. To understand the error influence consider the data points belonging to class A. In this case an intermediate error $E_{total\,intermediate} \approx E_{total^-}$ will pull the data points in the feature space towards the decision boundary, while for class B such an intermediate error will push the data points away from the decision boundary. For an intermediate error $E_{total\,intermediate} \approx E_{total^+}$ the influence is reversed.

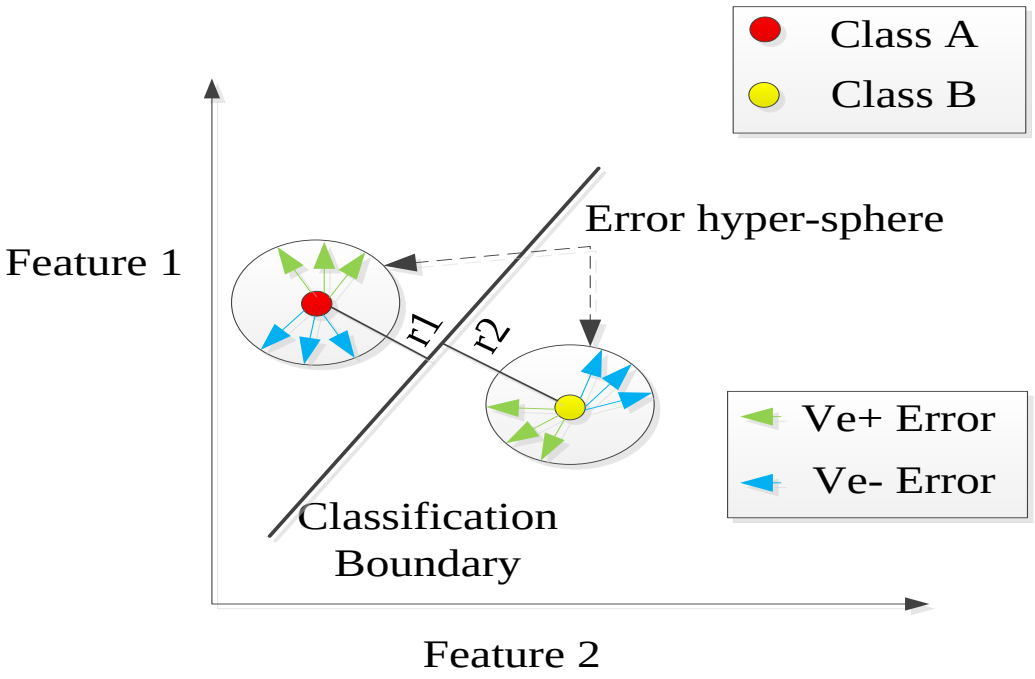


Figure 2: Error influence on classification

Figure 2 depicts these error influences. In Figure 2, one data point of each class has been shown in the feature space. Notice here that the green arrow depicting the negative error $E_{total\,intermediate} \approx E_{total^-}$ influence pulls the data towards the origin for class B while the positive error $E_{total\,intermediate} \approx E_{total^+}$ (depicted by the blue arrow) shows opposite behavior. Mis-classifcation will occur when these error hyper-sheres will touch the decision/classification boundary. Thus in order to counter these errors we need to use fixed point representation with the following criteria:

$$E_{total^-} - r1 \leq 0 \quad (0.24)$$

and,

$$E_{total^+} - r2 \leq 0 \quad (0.25)$$

Adhering to these above conditions we obtain two guard bits GB_1 and GB_2 settings needed to be added with the integer bits as:

$$GB_1 = \log_2\left[-k2^{-f}(2+2^{n-1}) - r1\right] \quad (0.26)$$

and,

$$GB_2 = \log_2\left[k2^{-f}\ \ 2+2^{n-1}-2^{-f}\ \ -r2\right] \quad (0.27)$$

In practice the guard bits GB can be chosen as the maximum of GB_1 and GB_2. With the number of guard bits thus obtained, and considering 1 sign bit (S), N and F bit to represent integer and fractional parts with accuracy 2^F, we can finally produce an optimal fixed point representation as shown in Figure 3 that will prevent mis-classification due to error propagation. Note that to represent the integer part now we use a total of $GB + N$ number of bits.

Figure 3: Optimal fixed point representation

3 Hardware architecture

With the cognizance of the optimal wordlength from our theoretical analysis, we proceeded towards implementing the LDA algorithm in FPGA. The architecture for the classifier is shown in Figure 4.

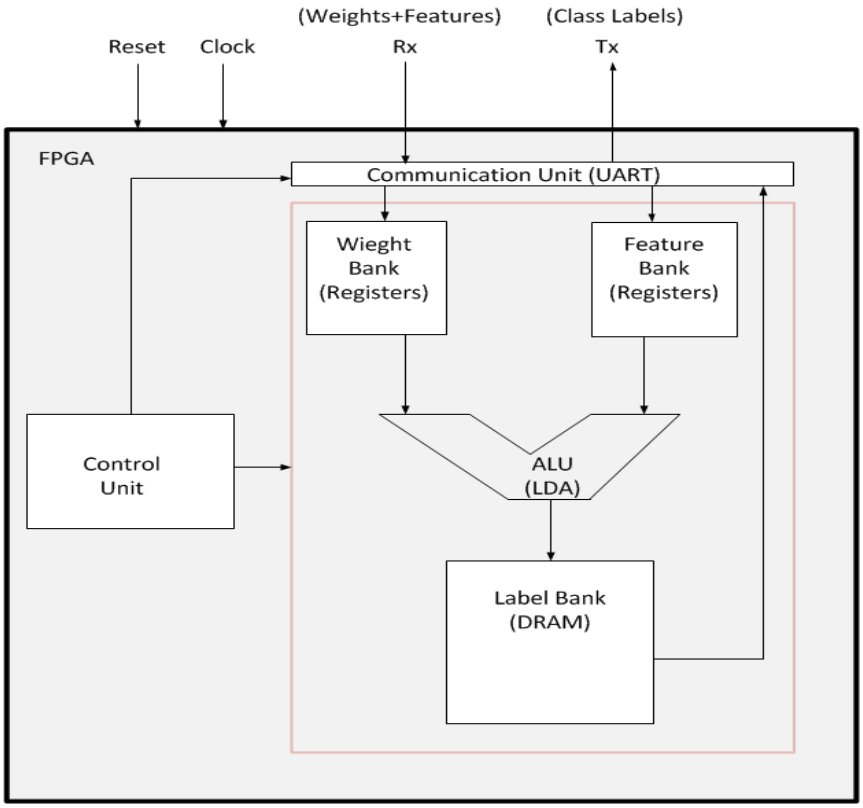


Figure 4: Hardware architecture of LDA classifier

The architecture consists of a register bank each for both the weights and the features. The

features and weights are received at real time through the UART unit. A multiply and accumulator unit is used to compute the LDA output and then the prediction for classes are produced through a comparator and subsequently saved in the Label Bank (instantiating a DRAM). The control unit synchronizes the data-flow through the aforementioned units with the UART. We purposefully avoided the implementation of feature calculation in hardware to maintain coherence with our analysis in the previous sections, as our wordlength calculation does not concern with the feature computation arithmetic.

In order to validate the design, classification between H_2SO_4 and O_3, was carried out using all the 11 features which were extracted from the raw plant electrical signals [4]. We updated the register bank containing the weights using the values learnt in the training phase. The classifier (hardware) was then tested using an independent feature vector of length 240 containing both the aforementioned classes. Table 2 reports the classifier prediction errors for the mentioned architecture with 64, 32, 24, 16 and 12-bit wordlength respectively. The error values are accumulated after simulating the RTL of the architecture in Modelsim. From Table 2 it is evident that wordlength of 12-bit and below introduces mis-classification and thus 16-bit is considered as an optimal wordlength for this architecture.

Table 2: Accuracy vs Wordlength

Wordlength	Error rate	Accuracy rate
32	0.833%	99.167%
24	0.833%	99.167%
16	0.833%	99.167%
12	12.917%	87.083%
8	50%	50%

3.1 FPGA details

We have used System Verilog for RTL development and used Altera QUARTUS II application for synthesis and placement in FPGA. Altera DE 2 [7] development board having a Cyclone IV FPGA [8] was chosen as the target platform for the classifier implementation. We chose the maximum available clock speed of 50 MHz for our design and to simulate a real time processing environment we used the MATLAB serial communication interface to stream feature values directly to the FPGA from MATLAB workspace using RS 232 connection as shown in Figure 5.

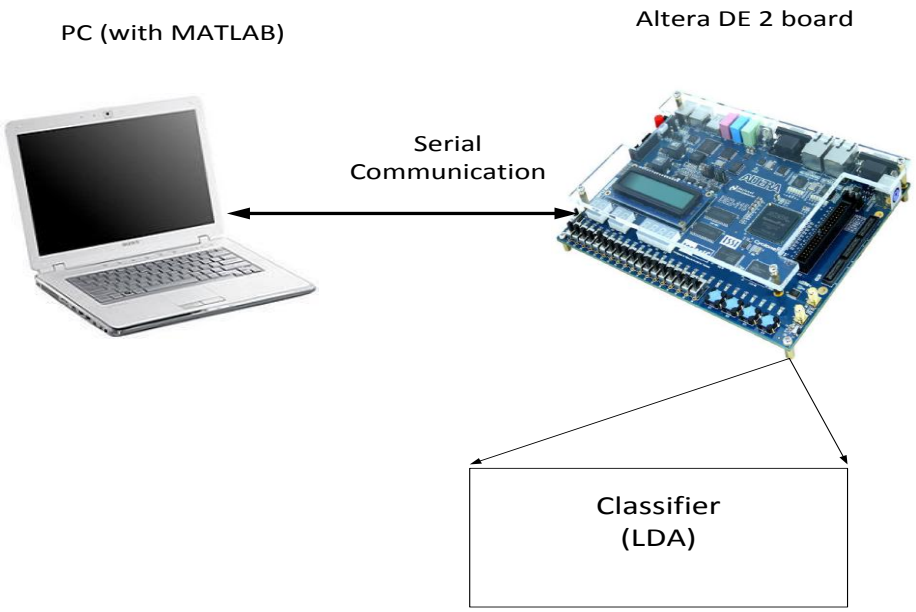


Figure 5: The test setup for real time classification

The classification results were validated by probing the Label Bank DRAM using a JTAG interface.

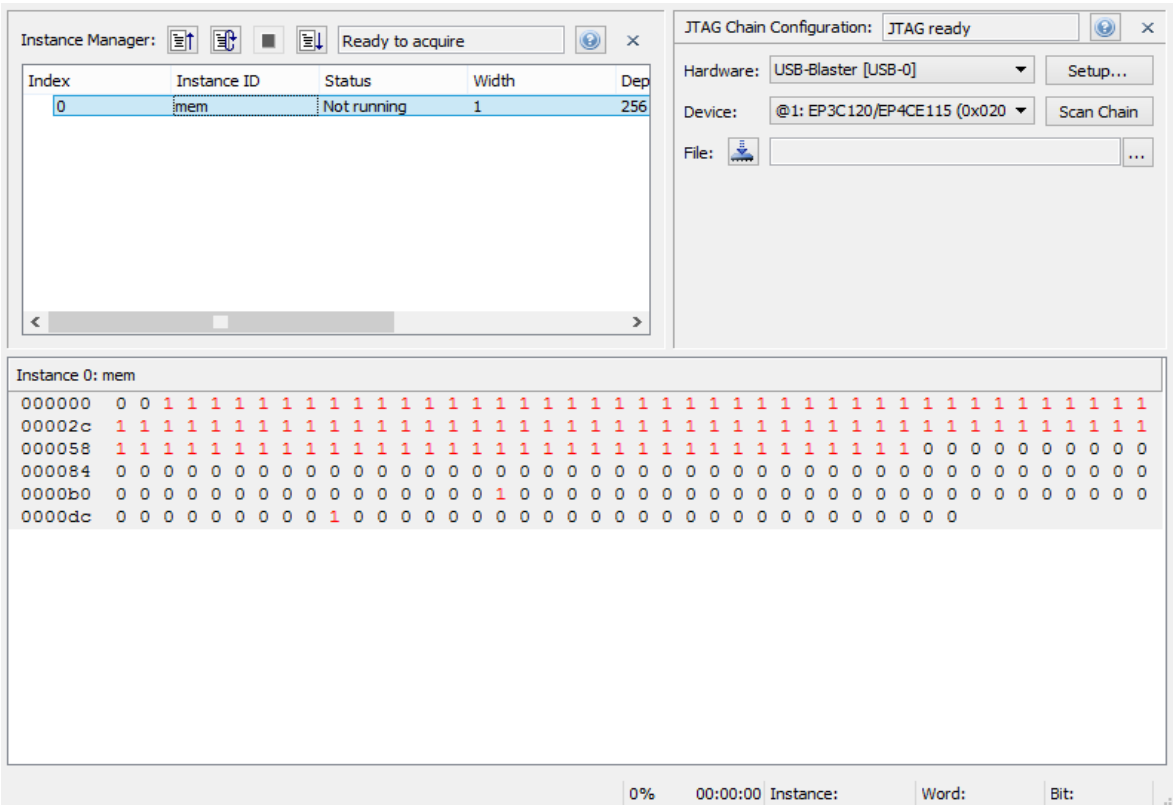


Figure 6: Memory view of Label Bank (DRAM)

Figure 6 shows the internal DRAM view of the Label Bank probed through JTAG interface. The memory values corresponding to '1' and '0' annotate the predictions as belonging to the H_2SO_4 and O3 class respectively. Figure 7 Summarizes the synthesis results.

```
+-------------------------------------------------------------------------------------+
; Flow Summary                                                                       ;
+-----------------------------------+-------------------------------------------------+
; Flow Status                       ; Successful - Tue Aug 26 23:09:41 2014           ;
; Quartus II 64-Bit Version         ; 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition ;
; Revision Name                     ; HanYutong                                       ;
; Top-level Entity Name             ; top                                             ;
; Family                            ; Cyclone IV E                                    ;
; Device                            ; EP4CE115F29C7                                   ;
; Timing Models                     ; Final                                           ;
; Total logic elements              ; 825 / 114,480 ( < 1 % )                         ;
;     Total combinational functions ; 688 / 114,480 ( < 1 % )                         ;
;     Dedicated logic registers     ; 369 / 114,480 ( < 1 % )                         ;
; Total registers                   ; 369                                             ;
; Total pins                        ; 26 / 529 ( 5 % )                                ;
; Total virtual pins                ; 0                                               ;
; Total memory bits                 ; 256 / 3,981,312 ( < 1 % )                       ;
; Embedded Multiplier 9-bit elements ; 22 / 532 ( 4 % )                               ;
; Total PLLs                        ; 0 / 4 ( 0 % )                                   ;
+-----------------------------------+-------------------------------------------------+
```

Figure 7: FPGA synthesis summary

3.2 ASIC Implementation

As apparent from Figure 5 the number of logic elements used is considered as a low value here, as they occupy less than 1% of all the available logic resources. With fewer logic resources, synthesized area would be relatively small and power dissipation would consequently be kept to a minimum. These factors are important for a possible ASIC implementation in the future. Thus we synthesized the entire design using for a possible ASIC implementation using the ST 130 nm library with a clock frequency of 50 MHz. Table 3 shows the total dynamic power consumption of the synthesized ASIC at 1.2 V and area of the synthesized design.

Table 2: Synthesis Results with ST 130 nm library

AREA (μm^2)	Total Dynamic Power (μW)
35809.334986	10.8077

4 Conclusion

This document presents a low-power implementation of the LDA classifier that shows good accuracy in classifying the two types of external stimuli under consideration. We present a complete real-time Altera Cyclone IV FPGA implementation which is running and need only to be interfaced with the plant. The generalized theory for fixed-point representation of the problem presented here allows optimal selection of wordlength to minimize the arithmetic error accumulated during the datapath operation. This analysis can be extended easily towards

accommodating non-linear classifiers as well as different datasets, where resource constraining is of utmost importance.

References

[1] E. D. Kusuma, R. Yusof, and M. F. Othman, "Fixed-point Implementation of Classifier for Tropical Wood Recognition System," *The 2014 International Conference on Communications, Signal Processing and Computers*.

[2] E. D. Kusuma, R. Yusof, and M. F. Othman, "Floating-point to Fixed-point Conversion of Tropical Wood Recognition System Classifier," *International Journal of Circuits, Systems and Signal Processing*, vol. 8, 2014.

[3] The mathworks Inc., "MATLAB Statistics Toolbox ," *Natick, Massachusetts* , vol. b.

[4] S. K. Chatterjee, S. Das, K. Maharatna, E. Masi, L. Santopolo, S. Mancuso, and A. Vitaletti, "Exploring strategies for classification of external stimuli using statistical features of the plant electrical response," *Journal of The Royal Society Interface*, vol. 12, no. 104, p. 20141225, 2015.

[5] C. M. Bishop and others, *Pattern recognition and machine learning*, vol. 1. Springer New York, 2006.

[6] S. Theodoridis and K. Koutroumbas, *Pattern recognition*, 4th ed. Academic Press, 2009.

[7] Altera, "DE2 Development and Education Board", http://wl.altera.com/education/univ/materials/boards/de2/unv-de2-board.html

[8] Altera, "Cyclone IV FPGA Device Family Overview", https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyclone-iv/cyiv-51001.pdf