



  
**RECIBE**  
Revista electrónica  
DE COMPUTACIÓN, INFORMÁTICA, ENERGÍA Y ELECTRÓNICA



# Índice

## Computación e Informática

The machine learning in the prediction of elections.....II

M.S.C. José A. León-Borges

M.A. Roger-Ismael-Noh-Balam

Lic. Lino Rangel Gómez

Br. Michael Philip Strand

## Electrónica

Implementación de un circuito custom DSP en FPGAs para cálculo del determinante 3x3, y matriz inversa de matrices ortogonales 3x3.....I

Francisco Plascencia Jáuregui

J.J. Raygoza Panduro

Susana Ortega C.

Edwin Becerra



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.

## The machine learning in the prediction of elections

M.S.C. José A. León-Borges  
Universidad Politécnica de Quintana Roo  
jleon@upqroo.edu.mx

M.A. Roger-Ismael-Noh-Balam  
Instituto Tecnológico de Chetumal  
ismael\_balam@hotmail.com

Lic. Lino Rangel Gómez  
Instituto Tecnológico de Chetumal  
linoqroo@gmail.com

Br. Michael Philip Strand  
Instituto Tecnológico de Chetumal,  
Estudiante de Ingeniería en Sistemas Computacionales

**Abstract:** This research article, presents an analysis and a comparison of three different algorithms: A.- Grouping method K-means, B.-Expectation a convergence criteria, EM and C.- Methodology for classification LAMDA, using two software of classification Weka and SALSA, as an aid for the prediction of future elections in the state of Quintana Roo. When working with electoral data, these are classified in a qualitative and quantitative way, by such virtue at the end of this article you will have the elements necessary to decide, which software, has better performance for such learning of classification.

The main reason for the development of this work, is to demonstrate the efficiency of algorithms, with different data types. At the end, it may be decided, the algorithm with the better performance in data management.

**Keywords:** Automatic Learning, fuzzy logic, grouping, Weka, SALSA, LAMDA, state elections, prediction.

## **El aprendizaje automático en la predicción de las elecciones**

**Resúmen:** Este artículo de investigación presenta el análisis y comparación de tres algoritmos diferentes: A.- método de agrupamiento K-media, B.- expectativa de criterios de convergencia y C.- metodología de clasificación LAMDA usando dos softwares de clasificación, Weka y SALSA, como auxiliares para la predicción de las futuras elecciones en el estado de Quintana Roo. Cuando se trabaja con datos electorales, éstos son clasificados en forma cualitativa y cuantitativa, de tal virtud que al final de este artículo tendrá los elementos necesarios para decidir que software tiene un mejor desempeño para el aprendizaje de dicha clasificación. La principal razón para hacer este trabajo es demostrar la eficiencia de los algoritmos con diferentes tipos de datos. Al final se podrá decidir sobre el algoritmo con mejor desempeño para el manejo de información.

**Palabras clave:** aprendizaje automático, lógica fuzzy, agrupamiento, Weka, SALSA, LAMDA, elecciones estatales, predicción.

## **1. Introduction**

The fascination for predict the future, is one of the intents and desires that man still insists on getting. Much effort have individuals and companies to predict various aspects, for example, climate and product prices in the market. For example: Toro (2006) Forecast stock market using intelligent techniques, Matamoros (2006) Methodology for predicting oil prices based on fractal

dynamics and Weron (2007) Modeling and forecasting electricity loads and prices: A statistical approach. Other works are to: Matamoros (2005) Methodology oil price prediction based on fractal dynamics; calculated logarithmic returns, tracing methods, average values in time series to generate probabilistic scenarios.

There are many works prediction Data mining, a variant is research on the prediction and treatment of disease, alcohol use in adolescents, Vega (2009) Data Mining Applied to Prediction and Treatment of Diseases. Another perspective is: Jungherr (2012) Why the pirate party won the German election of 2009 or the trouble with predictions. A response to Tumasjan, A, Sprenger, TO, Sander, PG, & Welpe, IM predicting elections with Twitter : What reveal 140 characters reveal political sentiment.

There are studies on predicting elections that have been made in countries such as:

Spain, where Dellte (2012) Conducted prediction political tendency Twitter: Andalusian elections 2012; For Holland, Tumasjan (2010), Predicting Elections with Twitter. What about 140 Characters Reveal Political Sentiment; Germany, Sang (2012) Predicting the 2011 senate election results with Dutch twitter. In Proceedings of the Workshop on Semantic Analysis in Social Media. Jungherr (2012) Why the pirate party won the German election of 2009 or the trouble with predictions: A response to Tumasjan, A, Sprenger, TO, Sander, PG, & Welpe, IM "predicting elections with Twitter. What reveal 140 characters about political sentiment "; and Canada, where Sidorov (2013) did an Empirical study of machine learning based approach for mining review in tweets. In Advances in Artificial Intelligence.

According to (Becerra-Fernandez, 2003), the discovery of knowledge in databases has made the computational procedures in automatic programming are increasingly sophisticated. Say Berry (1997), the data mining aims to uncover patterns and relationships to make predictions.

First, the classification of the data, by a process of unsupervised learning as the grouping clustering, brings the find of groups that are different but the individuals are equal among themselves as noted (Vega, 2012) In the methodology of data mining.

We select the use of the data mining software called Weka, because it is an easy tool, and where different jobs are chose: like, Empirical studies of machine learning based approach for opinion mining in tweets, (Sidorov, 2013), The comparison of different classification techniques using Weka for breast cancer (Bin, 2007) and the compared in Data Mining Applied to Prediction and Treatment of Diseases (Vega, 2012) and the different software products for data mining.

Also we chose a hybrid model, see table 1, as clustering techniques for better results, some work related are, for example: The Data Mining applied to prediction and treatment of diseases (Arango, 2013) and Methods for predicting stock market indices. Echoes of Economics, (García, 2013).

**Table 1.**Description of prediction models (Garcia, 2013) and (Arango, 2013).

<b>Technique</b>	<b>Model type</b>
Multiple Regression	Linear
Neural networks (Radial Basis Function, RBF and Backpropagation)	Nonlinear
Methods K-nearest neighbor	Nonlínear
Probabilistic neural network (PNN)	Nonlínear
Genetic algorithm	Nonlinear
Neuro-fuzzy networks	Nonlinear e Hybrid
Neural networks MPL	Nonlinear

**Table 1.** Description of prediction models (Garcia, 2013) and (Arango, 2013).

<b>Technique</b>	<b>Model type</b>
SVM support vector machine	Nonlinear

With comparative purposes, this work shows the results of the classification performed into two applications of software: Weka and SALSA, with different techniques of clustering. Also are shown and detailed the experiment on the preparation of the data. In the first part it explains our intention of predicting state elections with qualitative and quantitative data as well as briefly describing of the three techniques of clustering: Expectation-Maximization, k-means and describe the methodology of classification LAMDA. Was made, a comparison of the techniques and software used with the results obtained, finally, we show the performance of each software tool.

## **2. Background**

### **2.1 References**

To analyze the decision-making of the citizens, it is necessary to have instruments of measurement with respect to its electoral behavior, such as surveys and projections. In Mexico in the first part there is something written, but on the second there is very little.

The literature on electoral projections is basic, because studies should nourish it, as in the case of statistical analysis, which are scarce (Torres, 2000). The lack of specialized bibliography, is due to the fact that since 1993, are disseminated by the Federal Electoral Institute IFE, and the state electoral bodies, the overall results and with some levels of disaggregation; what has meant is that there are no historical series of voting, nor criteria to build units of comparison.

With the practice, of presenting the basic statistics disaggregated to the level of electoral section and even casilla, have been ameliorated some of the described

shortcomings, however, there remains the need to analyze and interpret the data; setting criteria for the construction and use of statistical aggregates; and finally, to make attempts of predictions.

The literature on individual electoral behavior, has underlined the existence of stable predispositions to vote, sustained in the long term, on the basis of which the decision be finalized, except that they act on the individual circumstances of a particular choice: candidates, issues, and so on, all forces of short-term.

## **2.2 Prediction of state elections**

The prediction of the 2009 elections in Germany, using the techniques referenced, as shown in (Jungherr, 2012) Why the party won the German election of 2009 or the trouble with predictions: was done by taking into account the frequency of mentions and to get the total mentions, replication of mentions and percentages of mentions. The sample is less than a month, and takes representative days. It also takes into account the progression of the followers. The analysis of the results is quantitative.

On the other hand, the classification in the Empirical study of machine learning is based in approach for opinion mining in tweets. Sidorov (2013), through automatic learning, shows the possible classifications of classes and Support Vector Machine SVM as the best binder driving records 3393 as the best data set of training.

In this case, we take the elections in the state of Quintana Roo in the years 1998, 2004 and 2010.

# **3. Theoretical framework**

## **3.1 Automatic learning**

Today's Data Mining can be defined in terms of three easy concepts:



Statistics with emphasis on exploratory data analysis proper, big data and machine learning.

Samuel (1959) say, the definition of machine learning, was given to the field of study that assigns to the computers the ability to learn without being explicitly programmed. In other words, machine learning investigates ways in which the computer can acquire knowledge directly from data and thus learn to solve problems. The automatic learning is the acquisition of new knowledge, the development of a motor and cognitive skills through instruction or practice, the organization of new knowledge, effective representation and discovery of new facts and theories through observation and experimentation. The types of knowledge acquired are parameters in algebraic expressions, decision trees, formal grammar, production rules, logic-based formal expressions, graphs and networks, frameworks and diagrams and other procedural encodings and computer programs.

This learning is applied to many areas such as chemistry, education, computer programming, expert systems, video games, mathematics, music, natural language processing, robotics, speech recognition and image, and sequences of prediction example: An overview of machine learning, in Machine learning, by (Carbonell, 1983) among others.

### **3.2 Techniques of grouping or clustering**

The Clustering techniques, is referred in English as grouping, are procedures which are used to group a series of items. Clustering is used in statistics and science. The methods to deal with are: the hierarchical method because it is an exploratory tool designed to reveal the natural groups within a set of data that would otherwise is not clear. It is useful when you want to group a small number of objects, may be cases or variables, depending on if you want to classify cases or examine relationships between the variables.

A review of clustering algorithms, can be referenced by (Xu, 2005) and discuss different algorithm. The hierarchical method is built by a cluster of trees or hierarchical clusters. Each node contains clusters children. Categorized in agglomerative and divisive. The first begins with a cluster and after two or more similar clusters. The second begins with a cluster containing all the data points and recursively divides the group most appropriate. The process continues and stops until the criterion is improved.

**Table 2.**Classification of clustering algorithms (Berkhin, 2006).

<b>Method</b>	<b>Category</b>
Hierarchical method	Agglomerative algorithms y divisive algorithms
Method partition and relocation	Clustering probabilistic, K-medoids y K-means.
Partitioning method based on density	Clustering connectivity clustering based on density and density functions.
Network-based method	
Based on co-occurrence of categorical data Method	
Other clustering techniques	Constraint-based clustering, graph-partitioning, clustering algorithms with supervised learning and clustering algorithms with machine learning
Scalable clustering algorithms	
Algorithms for high dimensional data	Subspace clustering and co-clustering techniques

### **3.1.1 K-means**

K-means (MacQueen, 1967) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. k-means is one of the

simplest unsupervised learning algorithms that solve the well known clustering problem. This algorithm is classified, as a method of partitioning, and relocation of each of its clusters, represents the average of their points centroid. The advantage of using this is by the quick view graph and statistics. The target function, is the sum of the errors between the centroid and their points, i.e. the total variance within the cluster.

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster centres. The k-means algorithm can be run multiple times to reduce this effect. K-means is a simple algorithm that has been adapted to many problem domains, it is a good candidate for extension to work with fuzzy feature vectors.

Finally, this algorithm aims at minimizing an objective function, in this case a squared error function.

### **3.1.2 Expectation Maximization EM**

The algorithm of Expectation Maximization, EM, belongs to the family of finite mixture models, used for segmenting multivariable data. It is a probabilistic clustering algorithm, where it seeks to know the target function of the unknown probability to which the data set belongs. Each cluster is defined by a normal distribution, the problem is, that the distribution is not known to which the data and parameters corresponds. Specifically, the algorithm is divided into two stages: expectation and a convergence criteria.

The first with unknown underlying variables, using parameter estimation up to the observation. The second provides a new estimation of the parameters, as shown in the papers: The algorithm of expectation-maximization (Moon, 1996) and unsupervised learning of finite mixture models (Figueiredo, 2002). Iterate both, until converge. In other words, the calculation of the probabilities or the expected

values is performed and the value of the parameters of the distributions, is calculated for maximizing the probability of distributions of the data. The estimation of the parameters is considered to be the probability that the data belongs or does not belong to a cluster.

### 3.1.3 Fuzzy Logic

The fuzzy logic deals when imprecise or subjective terms is handled, where an element can belong to multiple sets of partial form. Fuzzy logic was defined in this work by membership functions of campaign by the functions mean and standard deviation.

The characteristic function  $\mu_S$  commonly used to define if a member belongs or not to a member, or a set of members (S),  $\mu_S(x) = 1$ , if x is a member of S and  $\mu_S(x) = 0$ , if x is not a member of S.

$$\mu_S(x) \begin{cases} 1, & x \in S \\ 0, & x \notin S \end{cases}$$

**Formula 1.** Feature.

The operations to be performed on fuzzy sets are shown in table 3.

**Table 3.** Operations on fuzzy sets.

Operation
Inclusion or subassembly
Union. Zadeh logical OR operator (max)
Intersection. Zadeh logical AND operator (min)
Denial or supplement
Cartesian product

**Table 3.**Operations on fuzzy sets.

Operation
Cartesian co-product

### 3.1.4 LAMDA classification methodology

The LAMDA algorithm, Learning Algorithm for Multivariate Data Analysis, is a classification method developed by N. Piera and J. Aguilar (Aguilar, 1982), (Desroches, 1987), (Piera, 1989). It is created by a principle to categorize data, where there may be many variables, however, classified in both numeric and symbolic, quantitative data and qualitative data.

LAMDA enters the classification within the theory of networks of radial basis function, which is a method to improve the generalization of new data. The learning of radial basis can be given supervised or not supervised. Supervised, when it seeks to minimize the error between the output value of the network and the desired output value, using least squares. Non-supervised, where it allows to divide the space of input patterns into classes. The number of classes, is the number of hidden neurons in a radial basis network.

An object X, is represented by a vector that contains a set of features, in this case it can be the votes of party1, party 2 or party 3, called descriptors. In the classification, each object X is assigned to a class. A class is defined as the universe of descriptors that is characterized as a set of objects.

LAMDA, performs classification according to criteria of similarity in two stages: first, the similarity criteria of each descriptor object corresponding, to the descriptor of a given class measured, this is also known as the obtaining or calculation of the degree of marginal adequacy MAD (Marginal Adequation Degree). Second, when the measurement of the relevance of an object is added to a particular class, this is also known, as the obtaining or calculation of the degree of overall adequacy GAD (Global Adequation Degree).

The calculation of GAD, deals with the concept of diffuse connective, which is explained by the aggregation function as linear interpolation of t-norm with t-conorma. The total diffuse intersection and the total diffuse union respectively. These aggregation operators get the GAD an individual data added to a class.

$$GAD = \beta T(MAD) + (1 - \beta) S(MAD) \quad (1)$$

$$GAD(x|C) = \alpha \cdot \gamma[MAD(x_1|C), \dots, MAD(x_P|C)]$$

$$+ (1 - \alpha) \cdot \beta[MAD(x_1|C), \dots, MAD(x_P|C)]$$

The connectors used in GAD are, the intersection, t-norm and union, t-conorma. The role of aggregation (Piera and Aguilar, 1991) is a linear interpolation between t-norm ( $\alpha$ ) and t-conorma ( $\beta$ ).

$$\gamma(a, b) = a.b(t\text{-norm}) \text{ and } \beta(a, b)$$

$$= a + b - a.b(t\text{-conorm})$$

$$\gamma(a, b) = \min(a, b)(t\text{-norm}) \text{ and } \beta(a, b)$$

$$= \max(a, b)(t\text{-conorm})$$

Finally the global maximum of similarity of an object in a class allows the definition of a class that best describes the object.

In other words MAD is a related term of how similar a descriptor object is, to the same descriptor of a given class and GAD is defined as the degree of relevance of an object at a given class given, as a function of relevance diffuse.

$$MAD_{c,d} = \rho_{c,d}^{X_{i,d}} (1 - \rho_{c,d})^{1-X_{i,d}} \quad (1)$$

Where:

$\rho_{(c,d)}$ =Learning Parameter (Rho) for class c and the descriptor d.

$X_{(i,d)}$ =The Descriptor d of object i.

The implementation of LAMDA, includes a probability function for estimating the distribution of descriptors based on fuzzification, the process of converting an element in a value of each class to which it belongs.

The main features and that makes difference of LAMDA are the NIC, Not Information Classified, which allows you to perform supervised classifications and non-supervised, learning functions are based on arithmetic, they can modify the parameters representative of each class. NIC accepts all objects contained in the universe of description with the same degree of appreciation GAD. For more information about the learning algorithm for the analysis of multivariate data, refer (Aguado-Chao, 1998), a mixed qualitative quantitative self-learning classification technique applied to situation assessment in industrial process control.

## **4. Methodology**

It is important to study the relationship between the historical trend of the vote and the electoral results of a specific process; it is important because it allows us to make predictions, which can, in good measure, sensitizing political actors and citizens about the possible results of the electoral process.

It is pertinent to note, that the research was carried out, by ordering the results of the processes of local governor 1998, 2004 and 2010 of the state of Quintana Roo, to develop historical series of votes, which were necessary to carry out the projections.

The election results are not entirely accidental events, fully relieved by past events, and that much of what happens in local processes allows us to contemplate the possible scenarios of the local process.

Thus, in the case of the local executive, it incorporates data from the last three elections for governor 1998, 2004 and 2010, data were analyzed from municipal presidents and local deputies, the above are for every 3 years; due to the

difficulties to normalize the data and the lack of the data themselves; it was determined to use the data for governorship.

The historical evolutions that have had the political parties in the State of Quintana Roo, clearly shows how diversity have emerged from these political actors , but with the passage of time it have been terminated. The political parties which, with the passage of time have subsisted alone or coalitions are: the PAN, PRI and the PRD, for the case study.

To obtain the data already standardized, was made a historical analysis of the evolution of political parties and their coalitions; it concluded: in the case of the state of Quintana Roo, in all elections for governor all three major political parties in Mexico were present or were coalitions.

First, and for not having bias or trend, it was taken, in the order they appear registered in the State electoral unit, in such a way that they appear in the following way: PAN, PRI and PRD or their respective coalitions. In that sense we started to take the year of election 1998, 2004 and 2010 as data, being the data that was obtained from State Electoral Body and taking into account that the major election is performed every 6 years.

The data are then classified by electoral district of the years: 1998, 2004 and 2010 for these years there have been 15 districts; in such a way that the division was carried out by electoral district and for each electoral district it disaggregated by casilla, for every casilla there was a need to standardize the information; for each casilla is divided by type of casilla, in such a way that it was the unbundling of the most basic data form.

The record is as follows: year of the election, the electoral district, casilla, political 1, political 2 and 3. Leaving 2 types of data: qualitative data and the others quantitative.

## **5. Experiments**



## 5.1 Salsa

The generated file of the data, must include the header, a, according to the format that handles the tool for this case SALSA, subsequently the data is standardized; file, b, is saved. It is appropriate to perform the loading of the data through the file in text format, c.

Once it has been proceeded to load the data in SALSA, ii proceeds to process them d.

### 5.1.1 File Header

The format of the header of the file that will be used to process the data in the tool is the one shown in table 4, the tool asks that at the beginning of the file exists & and the other columns should be separated by a Tab.

**Table 4.**File header.

<b>&amp;ANIO</b>	<b>DIST</b>	<b>CAS</b>	<b>PAN</b>	<b>PRI</b>	<b>PRD</b>
					.

### 5.1.2 Standardized Data year 1998- 2010

The data were grouped by years 1998, 2004 and 2010, the electoral district to which corresponds I. XV, the number and type of polling station Basic, contiguous, special or extraordinary and finally the vote for the party.

**Table 5.**SALSA dataset file.

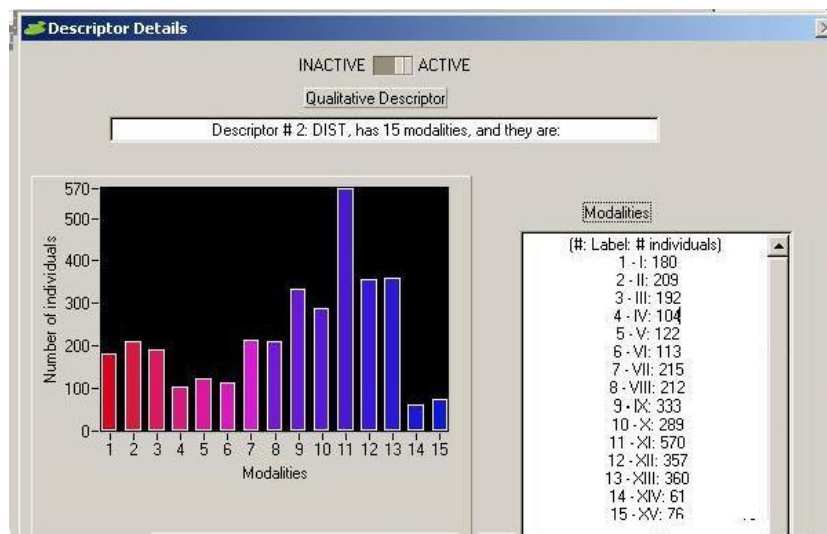
<b>&amp;ANIO</b>	<b>DIST</b>	<b>CAS</b>	<b>PAN</b>	<b>PRI</b>	<b>PRD</b>
1998	I	300B	83	149	45
...					
1998	XV	297B	11	235	236

**Table 5.**SALSA dataset file.

&ANIO	DIST	CAS	PAN	PRI	PRD
...					
2004	I	300B	206	161	20
...					
2004	XV	297B	3	35	127
2010	I	300B	47	153	58
...					
2010	XV	297B	73	137	79

### 5.1.3 Data loaded

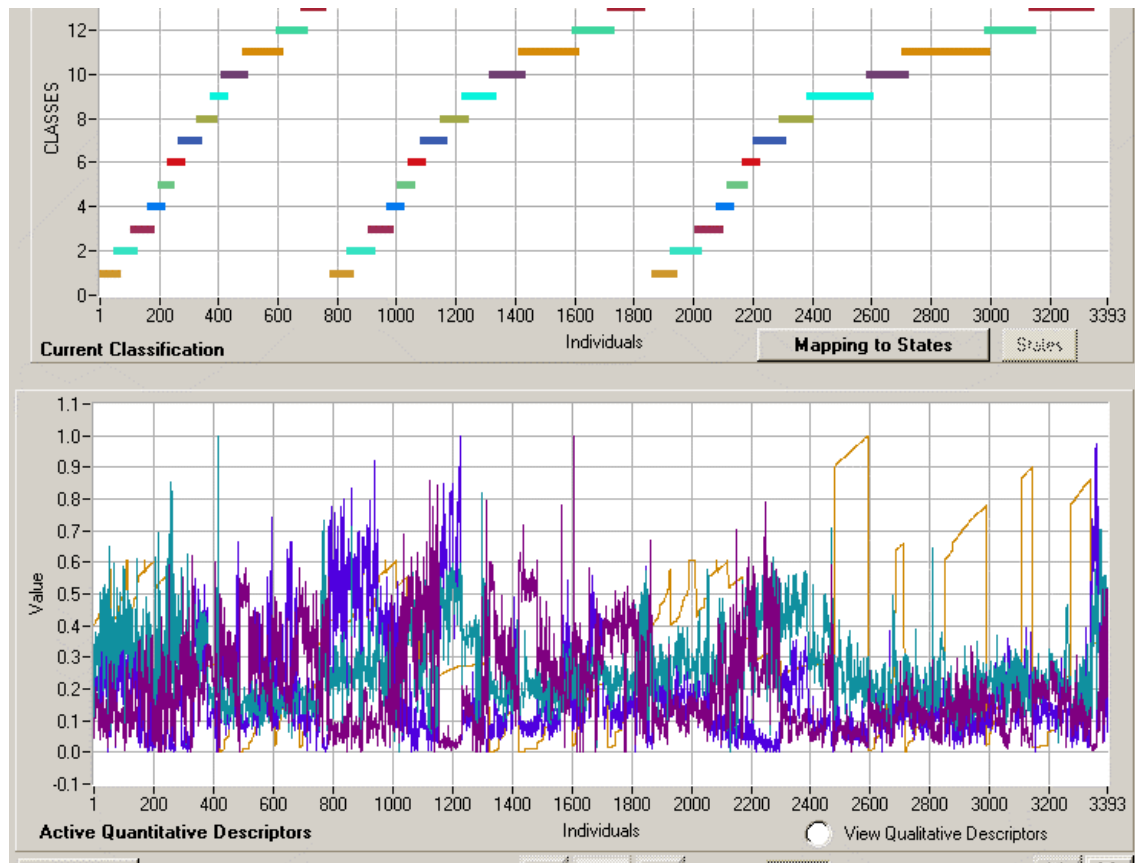
In figure 1, it is seen as if the tool has already grouped and sorted all 100 % the data, in a quantitative and qualitative manner, if we are to observe the previous point, the file is a set of unsorted and unclassified data, where there are numbers and alphanumeric.



**Figura 1.** Details of descriptors.

## 5.1.4 Processed data

The historical data processed contain 3390 samples and 15 descriptors or physical variables obtained by the normalization done to the election data. Through a standardization of data, it shows the representation of the behavior of the variables. The minimum and maximum values that are used in each descriptor helps homogenize the influence of its dimensions.



*Figura 2. Profiles.*

As a last step the profile of the class e is obtained.

## 5.1.5 Descriptors

This context has 6 descriptors with the current state of the active descriptor:

**Table 6.** Descriptors of the dataset.

No.	Nombre	Tipo	Max	Min
1	Anio	CUANTITATIVO	2010	1998
2	DIST	CUALITATIVO	Label no. 1 .... Label no. 15	Valor I .... Valor XV
3	CASILLA	CUANTITATIVO	741	1
4	PAN	CUANTITATIVO	248	0
5	PRI	CUANTITATIVO	491	0
6	PRD	CUANTITATIVO	491	0

## 5.2 WEKA

A procedure is performed similar to that made with *salsa*. As a first step it generates a data file, the file generated from the data, must include the header a, according to the format the tool handles, for this case *Weka*, after such data is standardized, the file is saved. It is appropriate to perform the loading of the data through the file in text format b.

### 5.2.1 File Header

In the case of the EM algorithms and K-means using the software *Weka*, the attribute *polling station* had to be removed, due to both *grouping* classifiers does not allow qualitative data.

@relation 'datos'

@attribute anio numeric

@attribute distrito {I, II, III, IV, V, VI, VII, VIII, IX, X, XI, XII, XIII, XIV, XV}

@attribute pan numeric

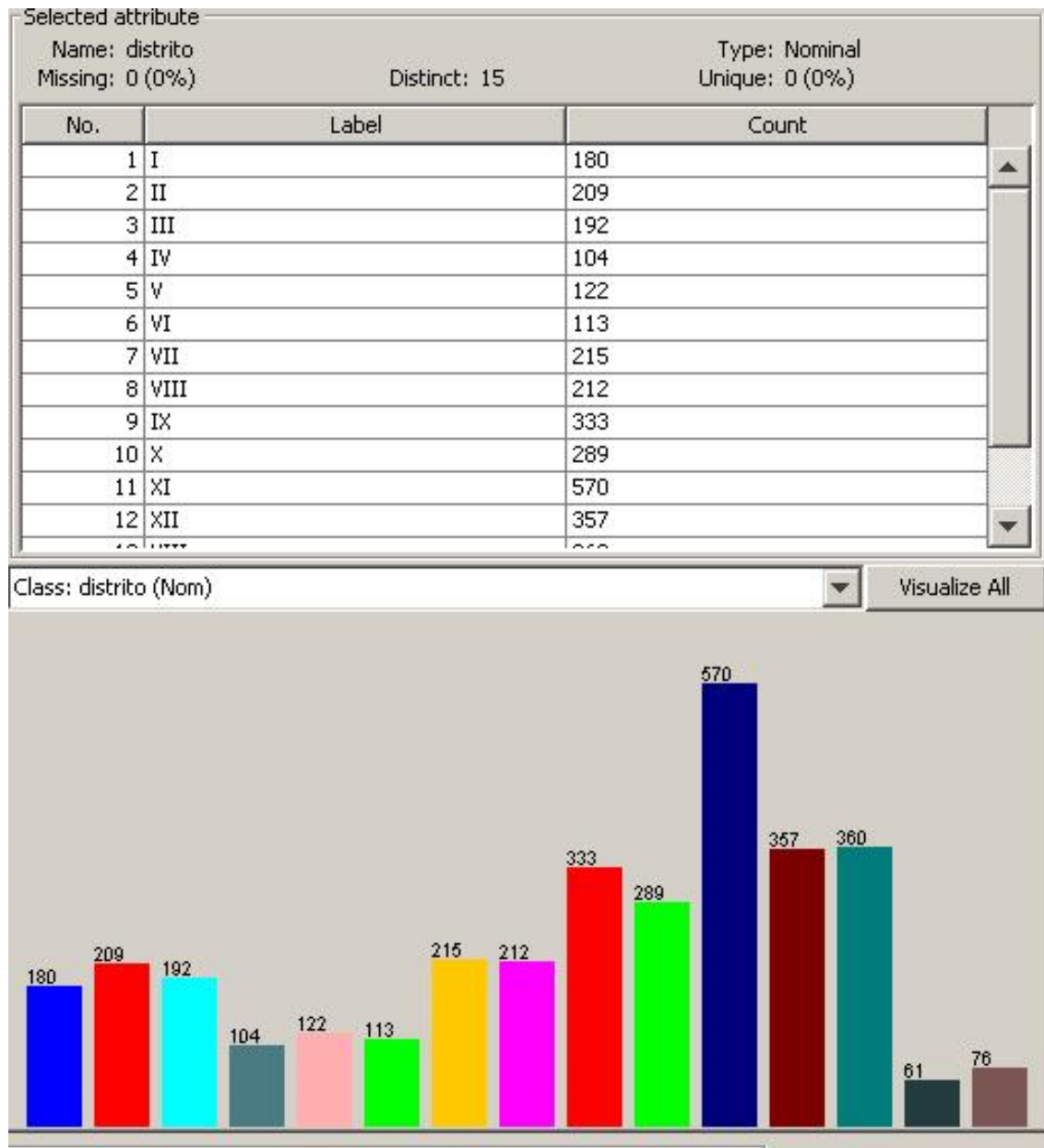
@attribute pri numeric

@attribute prd numeric

*Figura 3. File Weka Dataset.*

### **5.2.2 Data loaded**

It is noted how Weka does the classification in the form of a table and in the form of bars, in the case of the table it makes a classification by district and the result of the grouping of the data for each district. For the graphs it only shows its concentration and one would have to deduce that each bar is an electoral district.



*Figura 4. Classification.*

### 5.2.3 Data processed by K-means

As how it can be observed in the circles in Figure 3, the tool generates two Cluster Centroids turns the display XI and XIII and it handles them as if they were the most distant and from that first the adjustment is made.

```

Clusterer output

Clusterer Model

kMeans
*****

Number of iterations: 22
Within cluster sum of squared errors: 2635.438902787142
Missing values globally replaced with mean/mode

Cluster centroids:

Attribute      Full Data      Cluster#
              (3393)        0          1
-----
distrito      XI            XIII       XI
pen           51.3118      81.3363    29.1205
pri          129.104      148.432    114.8186
prd          88.0899      65.9341    104.4654

Time taken to build model (full training data) : 0.36 seconds

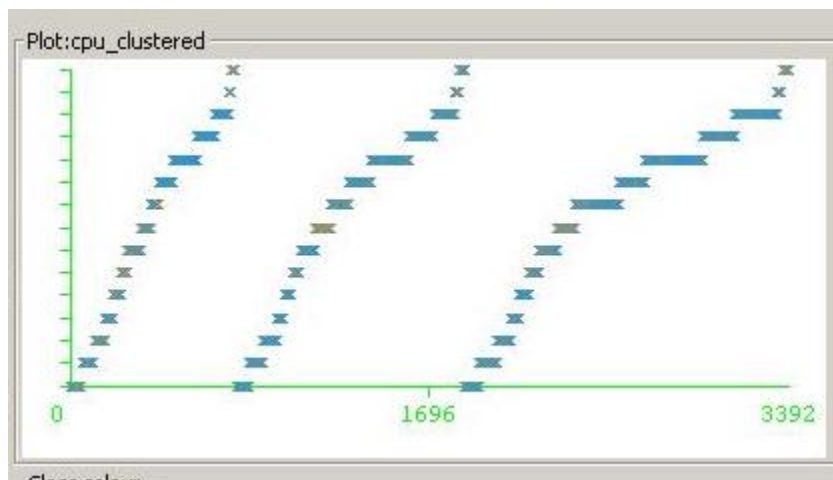
--- Model and evaluation on training set ---

Clustered Instances

0      1442 ( 42%)
1      1951 ( 58%)

```

*Figura 5. Cluster Model - K-means.*



*Figura 6. Cluster Plot.*

Generation of results through the Weka methods.

## **6. Advantage**

Proper use of data classification algorithms, has several advantages. With this work, we have concluded that there are the following advantages:

In an investigation of qualitative nature, these algorithms serve to validate the data, because they are based on mathematical models. Once the system has been trained, within the domain that is done, the manipulation of data, interpretation of the results is often simple. Because, using the mathematical methods, experiments can be repeated and provide the same results for verification.

The clustering techniques, have significant advantages in terms of time. The quality of the results depends on the similarity measure used. Further the quality is measured by its ability to discover hidden patterns.

The cost of learning is null. No need to make any assumptions about the concepts to learn. You learn complex concepts using simple functions such as local approximations. You can extend the mechanism to predict continuous values.

## **7. Results and future work**

Although Weka is a software that allows you to make automated classifications by different methods and forms, in this analysis only the methods proposed at the beginning of the article were used, in such a way that it shows, the results that were more adjusted or approximated the classifications long awaited by the expert. This is due to the fact that there are qualitative data, not all of the tools have the ability to properly classify and on the other hand the behavior cannot be explained under some statistical method. The software that best made the classifications was SALSA, see section 5.1, C and D, due to the fact that it included in the right way, the qualitative data along with the quantitative data.



With the data that has already been processed in both Weka and in SALSA, an analysis and comparison was carried out. The analysis was performed with respect to its performance to the clustering of quantitative and qualitative data, on the other hand, is made with respect to the efficiency when performing such data. In this point, it can be seen that the 3393 data 100 %, were classified properly in SALSA, unlike Weka which could not reach its classification in this same percentage.

The application of these algorithms in state elections is very efficient. The algorithms to predict the future, can be used in real time, like an option for the process electoral. It would be very useful and efficient to determine trends in state election by electoral districts in real time. For this case it is very useful to use the larger districts. With the bases shown in this work, on the future, a forecast can be done for any mathematical methodology as well as the correlations between the data of the districts and the casillas, and completing the corresponding analyzes by experts; for the work or the purposes to be used. The application of these algorithms in state elections is very efficient. The algorithms to predict the future, can be used in real time, like an option for the process electoral. It would be very useful and efficient to determine trends in state election by electoral districts in real time. For this case it is very useful to use the larger districts.

Based on the article and with the base shown in this work, in the future, you can make projections for a forecast, to be classified with new data and clustering algorithms; and we will make an analysis, a correlations between the data of the districts and the casillas. Our work will be done, for the purposes that the political groups consider relevant.

## **8. Conclusions**

This article, find the current state of the art, related to the prediction of elections, using automatic programming. In particular of three algorithms: A.- Grouping method K-Means, B.- EM and C.- Methodology for classification LAMDA, through

the use and application of the Weka and SALSA, as support tools for the processing of standardized set of data, to make electoral predictions, for the election of governor in the state of Quintana Roo. Due to the set of data that were analyzed qualitative and quantitative as a whole, it was determined that the algorithm of LAMDA that used implicitly the SALSA software is the one that best returned the results, and is adjusted to the criteria of a more accurate form.

## Referencias

Aguado-Chao, J. C. (1998). *A mixed qualitative quantitative self-learning classification technique applied to situation assessment in industrial process control*. Scotland: Associate European Lab, on Intelligent Systems and Advanced Control (LEA-SICA).

Aguado-Chao, J. C. (1998). *A mixed Qualitative-Quantitative self-learning Classification Technique Applied to Situation Assessment in Industrial Process Control*. Ph. D. Thesis Universitat Politecnica de Catalunya, Catalunya, España.

Aguilar-Martín, J., & López De Mantaras R. (1982), R. The process of classification and learning the meaning of linguistic descriptors of concepts. *Approximate reasoning in decision analysis*. 1982, pp. 165–175.

Arango, A., Velásquez, J. D., & Franco, C. J. (2013). *Técnicas de Lógica Difusa en la predicción de índices de mercados de valores: una revisión de literatura*. *Revista Ingenierías Universidad de Medellín*, 12(22), 117-126.

Becerra-Fernández, I.; Zanakis, S.H. & Steven Walczak (2002). *Knowledge discovery techniques for predicting country investment risk*. *Computers and Industrial Engineering*, 43(4) 787-800.

Berkhin, P. (2006). *A survey of clustering data mining techniques*. In *Grouping multidimensional data*. (pp. 25-71). Springer Berlin Heidelberg.

Berry, M. & Linoff, G.(1997). *Data Mining Techniques for Marketing, Sales, and Customer Support*, John Wiley & Sons Inc.

Bin-Othman, M. F., & Yau, T. M. S. (2007). *Comparison of different classification techniques using WEKA for breast cancer*. In *3rd Kuala Lumpur International Conference on Biomedical Engineering 2006* (pp. 520-523). Springer Berlin Heidelberg.

Carbonell, J. G., Michalski, R. S., & Mitchell, T. M. (1983). An overview of machine learning. In *Machine learning* (pp. 3-23). Springer Berlin Heidelberg.

De Ariza, M. G., & Aguilar-Martin, J. (2004). Clasificación de la personalidad y sus trastornos, con la herramienta LAMDA de Inteligencia Artificial en una muestra de personas de origen hispano que viven en Toulouse-Francia. *Revista de Estudios Sociales*, (18), 99-110.

Dellte, L., Osteso, J., M., & Claes, F (2013). Predicción de tendencia política por Twitter: elecciones Andaluzas 2012. *Ámbitos. Revista Internacional de Comunicación*, 22(1).

Desroches, P.(1987). *Syclare: France de Classification avec Apprentissage et Reconnaissance de Formes. Manuel d'utilisation. Rapport de recherche, Centre d'estudis avançats de Blanes, Spagne.*

Figueiredo, M. A., & Jain, A. K. (2002). Unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence. IEEE Transactions on*, 24(3), 381-396.

García, E. G., López, R. J., Moreno, J. J. M., Abad, A. S., Blasco, B. C., & Pol, A. P. (2009). La metodología del Data Mining. Una aplicación al consumo de alcohol en adolescentes. *Adicciones: Revista de sociodrogalcohol*. 21(1), 65-80.

García, M. C., Jalal, A. M., Garzón, L. A., & López, J. M. (2013). Métodos para predecir índices bursátiles. *Ecós de Economía*, 17(37), 51-82.

Garre, M., Cuadrado, J. J., Sicilia, M. A., Rodríguez, D., & Rejas, R. (2007). Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software. *Revista Española de Innovación, Calidad e Ingeniería del Software*, 3(1), 6-22.

Jungherr, A., Jürgens, P., & Schoen, H. (2012). Why the pirate party won the german election of 2009 or the trouble with predictions: A response to Tumasjan, A., Sprenger, to, Sander, PG, & Welpe, IM "Predicting elections with twitter: What 140 characters reveal about political sentiment". *Social Science Computer Review*, 30(2), 229-234.

J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*", Berkeley, University of California Press, 1:281-297

Makazhanov, A., & Rafiei, D. (2013). Predicting political preference of Twitter users. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 298-305). ACM

Matamoros, O. M., Balankin, A., & Simón, L. M. H. (2005). Metodología de predicción de precios del petróleo basada en dinámica fractal. *Científica*, 9(1), 3-11.

Torres Medina, Luis Eduardo (2000) Proyecciones electorales. *Estudios Políticos*, (26).

Moon, T. K. (1996). The expectation-maximization algorithm. *Signal processing magazine, IEEE*, 13(6), 47-60.

Piera, N., Deroches, P. and Aguilar-Martin (1989). J. LAMDA: An Incremental Conceptual Clustering Method. LAAS-CNRS, report (89420), Toulouse, France.

Xu R., & Wunsch D. (2005). Survey of clustering Algorithms. *Neural Networks, IEEE Transactions on*, 16(3), 645-678.

Ross, Timothy J (2009). *Fuzzy logic with engineering applications*. John Wiley & Sons.

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. II - Recent progress. *IBM Journal of research and development*, 11(6), 601-607.

Sang, E. T. K., & Bos, J. (2012). Predicting the 2011 dutch senate election results with twitter. In *Proceedings of the Workshop on Semantic Analysis in Social Media*, 53-60.

Sidorov, G., Miranda-Jiménez, S., Viveros-Jiménez, F., Gelbukh, A., Castro-Sánchez, N., Velásquez, F., & Gordon, J. (2013). Empirical study of machine learning based approach for opinion mining in tweets. In *Advances in Artificial Intelligence* (pp. 1-14). Springer Berlin Heidelberg.

Toro Ocampo, E. M., Molina Cabrera, A., & Garcés Ruiz, A. (2006). Pronóstico de bolsa de valores empleando técnicas inteligentes. *Revista Tecnura*, 9(18), 57-66.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., & Welp, I. M. (2010). Predicting Elections with Twitter. What 140 Characters Reveal about Political Sentiment. *ICWSM*, 10, 178-185.

Vega, C. A., Rosano, G., López, J. M., Cendejas, J. L., & Ferreira, H. (2010). Data Mining Aplicado a la Predicción y Tratamiento de Enfermedades. *CISCI (Conferencia Iberoamericana en Sistemas, Cibernética e Informática) 2012, Aplicaciones de Informática y Cibernética en Ciencia y Tecnología*. Recuperado el 24 de Marzo de 2015, de [http://www.iiis.org/CDs2012/CD2012SCI/CISCI\\_2012/PapersPdf/CA732OV.pdf](http://www.iiis.org/CDs2012/CD2012SCI/CISCI_2012/PapersPdf/CA732OV.pdf)

Weron, R. (2007). *Modeling and forecasting electricity loads and prices: A statistical approach* (Vol. 403). John Wiley & Sons.

Zadeh, L. A. (1994). *Fuzzy logic, neural networks, and soft computing*. *Communications of the ACM*. 37(3), 77-84.

## Notas biográficas:

**José Antonio León Borges** 2014: Candidato a Doctor en Sistemas Computacionales: Universidad del Sur, plantel CanCùn 2012. Maestría en Ciencias de la Informática. Universidad Villa Rica. Boca del Río, Veracruz. 2010. Especialidad en Base de Datos. Universidad Villa Rica. Boca del Río, Veracruz. 2009. Licenciatura en Computación y Sistemas. Universidad Villa Rica. Boca del Río, Veracruz. Inglés: 453 puntos en TOEFL Perfil Deseable Dic 2014-Dic 2017

**Roger Ismael Noh Balam** 2014: Candidato a Doctor en Sistemas Computacionales: Universidad del Sur, plantel CanCùn 2000. Maestría en Administración. Instituto de Estudios Universitarios, plantel CanCùn 1994. Licenciatura en Informática. Instituto Tecnológico de Chetumal 1994-2003: Profesor de Tiempo Completo Instituto Tecnológico de Chetumal 2003-2012: Titular de la Unidad Técnica de Informática y Estadística Instituto Electoral de Quintana Roo 2012: Profesor de Tiempo Completo Instituto Tecnológico de Chetumal

**Lino Rangel Gómez** 2008: Maestría en Tecnología Educativa Universidad Da Vinci 1993: Licenciado en Computación Universidad Autónoma del Estado de Hidalgo. 2002:Subdirector de Planeación del Instituto Tecnológico de Chetumal

**Micheal Philip Strand** 2015: Instituto Tecnológico de Chetumal, Estudiante de la Ingeniería en Sistemas Computacionales 2012: University of Belize, Architectural Technology, Associates degree in Applied Science



Esta obra está bajo una licencia de Creative Commons  
Reconocimiento-NoComercial-CompartirIgual 2.5 México.

## Implementación de un circuito custom DSP en FPGAs para cálculo del determinante 3x3, y matriz inversa de matrices ortogonales 3x3

Francisco Plascencia Jáuregui  
Centro Universitario de Ciencias Exactas e Ingenierías,  
CUCEI, Universidad de Guadalajara  
fplascenciaj@hotmail.com

J.J. Raygoza Panduro  
Centro Universitario de Ciencias Exactas e Ingenierías,  
CUCEI, Universidad de Guadalajara  
juan.raygoza@cucei.udg.mx

Susana Ortega C.  
Centro Universitario de Ciencias Exactas e Ingenierías,  
CUCEI, Universidad de Guadalajara  
susana.ortega@gdl.cinvestav.mx

Edwin Becerra  
Centro Universitario de Ciencias Exactas e Ingenierías,  
CUCEI, Universidad de Guadalajara  
edwincbecerra@gmail.com

**Resumen:** En este artículo se presenta el diseño e implementación de un circuito digital a medida para el cálculo de determinantes de orden 3x3 y matriz inversa de matrices ortogonales 3x3. Se analizan los resultados de la implementación de los circuitos en dos plataformas de familias de dispositivos reconfigurables,

estas son Artix 7 y Spartan 6 Low-Power, en los que se comparan la ocupación y los tiempos de respuesta. La descripción del circuito se realizó en Lenguaje de Descripción de Hardware (HDL).

**Palabras clave:** Determinante, FPGA, Matriz inversa.

## **Implementation of an orthogonal custom DSP FPGA circuit for calculating the determinant 3x3 and 3x3 matrix inverse**

**Abstract:** In this paper are presented the design and implementation of a digital circuit suited for the calculus of 3X3 determinants and inverse matrix of orthogonal 3X3 matrixes. The circuits' implementation results are analyzed in two platforms of the family of reconfigurable devices: Artix 7 and Spartan 6 Low-Power, for which occupation and respond answer were compared. The circuit description was made in hardware description language (HDL).

**Keywords:** Desterminants, FPGA, inverse matrix.

# **1. Introducción**

Para la resolución de diversos tipos de problemas en el ámbito de la ingeniería, ciencias exactas e investigación se usan diversas herramientas, una de ellas es el cálculo de matrices, que ha formado parte de estos trabajos desde mediados del siglo XIX, dicho tema sigue siendo actual hasta la fecha, esto gracias a que las innovaciones tecnológicas nos proporcionan herramientas que nos permiten realizar cálculos cada vez más complejos, con mayor precisión y en cantidades de tiempo muy reducidas.

Las aplicaciones en que se puede hacer uso de las matrices son tan amplias que van desde el procesamiento de imágenes digitales hasta el control de motores y



robots, allí radica la importancia de desarrollar nuevas herramientas que nos permitan obtener resultados de forma clara, rápida y sencilla.

En nuestros días existen dispositivos que combinan gran flexibilidad de uso con capacidad de procesamiento de datos, estos son los dispositivos reconfigurables FPGA, que les permite implementar hardware mediante el uso del Lenguaje de Descripción de Hardware (HDL), lo que facilita a su diseñador desarrollar circuitos complejos de acuerdo a sus necesidades.

## **2. Trabajos previos**

Para realizar estudios en las matrices de tamaño  $n \times n$  Wayne Eberly usó la aritmética entera, y calculó el determinante de la matriz completa mediante su forma Smith (Eberly, 2000). Tres años después, Xiaofang Wang optimizó el desempeño de un multiprocesador de operaciones matriciales, basado en un dispositivo reconfigurable FPGA (Wang, 2003). En 2007, Hongyan Yang presentó el procesamiento de vectores para operaciones de matrices mediante FPGA (Yang, 2007).

Más recientemente, en 2011 B. Holanda dió a conocer un acelerador para multiplicaciones matriciales con punto flotante en FPGA (Holanda, 2011), de igual forma lo hizo Z. Jovanovic al presentar otra estrategia de multiplicación con matrices de punto flotante (Jovanovic, 2012). Ese mismo año Yi-Gang Tai publicó sobre la aceleración de las operaciones matriciales con algoritmos pipeline y la reducción de vectores (Tai, 2012). En 2013, Sami Almaki presentó un nuevo algoritmo paralelo para la resolución de determinantes de orden  $n \times n$  (Almaki, 2013). Por último, el año pasado fue publicado por Xinyu Lei, el artículo sobre el paradigma del cómputo en el servicio de la nube, enfocándose en el caso de calcular determinantes de grandes matrices (Lei, 2014).

## **3. Matrices y determinantes**

Una matriz se define como un arreglo bidimensional de datos. Se les nombra con letras mayúsculas, mientras que sus elementos se enumeran con letras minúsculas. Tal como se observa en la figura 1.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

**Figura 1.** Matriz de 2x2.

Los subíndices nos indican la posición del elemento respecto a las filas y las columnas respectivamente, así pues, el elemento  $a_{21}$  se encuentra en la fila 2 y columna 1.

Mientras que el determinante se refiere a la expresión matemática que está intrínsecamente relacionado con la matriz cuadrada que le da origen, y posee varias propiedades, como la de establecer la singularidad de dicha matriz, es decir, si a partir de esa matriz es posible obtener un determinante mediante ciertas operaciones, se dice que la matriz es no singular.

El determinante ha acompañado a los estudiosos de las matemáticas muchos años antes de que hicieran aparición las matrices en el siglo XIX, y su análisis cobró mayor relevancia al momento de relacionárseles gracias a James Joseph Sylvester.

Desde entonces se han desarrollado diferentes algoritmos para el cálculo de determinantes, dependiendo del tamaño de la matriz y la complejidad de sus elementos, algunas de las formas de indicar que se calculará el determinante de una matriz se representan en la figura 2.

$$\det A = \det \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

Figura 2. Formas de expresar el determinante de una matriz 3x3.

## 4. Algoritmos para calcular determinantes de matrices 3x3

Uno de los algoritmos más usados es el de la regla de Sarrus, debido a su sencillez para recordarlo, esto gracias a la disposición de los elementos dentro de la matriz, sin embargo, no es el único, como se aborda más adelante.

En la figura 3 podemos observar la visualización de dicha regla:

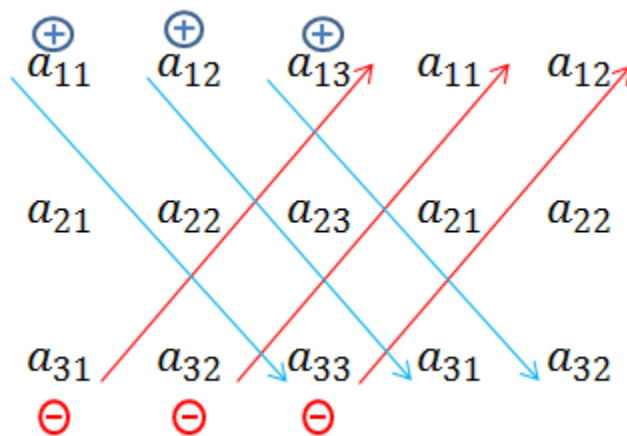


Figura 3. Regla de Sarrus para determinantes de matrices 3x3.

Este algoritmo establece que, para obtener el determinante debemos obtener seis términos a partir de multiplicar ciertos elementos, los primeros tres términos son:

$$(a_{11}a_{22}a_{33} + a_{21}a_{32}a_{13} + a_{31}a_{12}a_{23}) \quad (1)$$

En seguida, obtenemos los siguientes, con las estas multiplicaciones:

$$(a_{13}a_{22}a_{31} + a_{23}a_{32}a_{11} + a_{33}a_{12}a_{21}) \quad (2)$$

Por último, restamos a la ecuación (1) el resultado de la ecuación (2), de esta forma tenemos un resultado final.

Para obtener un determinante también podemos emplear el Teorema de Laplace, por el que podemos encontrar un determinante a partir del uso de menores y cofactores.

$$|A| = \sum_{i=1}^n \sum_{j=1}^n (-1)^{i+j} a_{ij} c_{ij} \quad (3)$$

La fórmula general planteada en la ecuación (3) nos dice que debemos seleccionar una fila o columna, para en seguida obtener los menores de cada uno de los elementos de esa fila o columna. Los menores son las matrices reducidas que se obtienen eliminando los elementos de la fila y columna del elemento elegido. Esto se observa en la figura 4.

$$\begin{vmatrix} a_{11} & \mathbf{a_{12}} & a_{13} \\ \mathbf{a_{21}} & a_{22} & \mathbf{a_{23}} \\ \mathbf{a_{31}} & a_{32} & \mathbf{a_{33}} \end{vmatrix} \rightarrow a_{12} = \begin{pmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{pmatrix}$$

**Figura 4.** Para obtener el menor del elemento  $a_{12}$  se ignora a  $a_{11}, a_{13}, a_{22},$  y  $a_{32}$ ; en cambio se considera a  $a_{21}, a_{23}, a_{31},$  y  $a_{33}$  para formar el menor.

Una vez que se tiene el menor se obtiene su determinante, que se conoce como cofactor. Ahora, este cofactor se multiplica con el elemento seleccionado, y su signo depende de la posición del elemento, es decir, si la suma de sus subíndices es par, el resultado será positivo, mientras que si la suma es impar su signo es negativo.

En la figura 5 se muestra, en términos generales, el planteamiento del determinante de una matriz 3x3 por la tercera columna, aplicando el Teorema de Laplace.

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = +a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} - a_{23} \begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} + a_{33} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

**Figura 5.** Aplicación de Teorema de Laplace para calcular un determinante 3x3 mediante su tercera columna.

Tal como se observa, el método de Sarrus implica doce multiplicaciones, cuatro sumas y una resta, mientras que el Teorema de Laplace requiere nueve multiplicaciones, cuatro restas y dos sumas, sin embargo, por su fácil implementación se elige trabajar con la regla de Sarrus.

## 5. Inversa de una matriz

Se define como la inversa de una matriz a aquella matriz que al multiplicarla por la matriz original da como resultado la matriz identidad (Grossman, 1996), es decir:

$$A \cdot A^{-1} = A^{-1} \cdot A = I \quad (4)$$

Esta matriz inversa se puede obtener mediante la siguiente fórmula:

$$A^{-1} = \frac{1}{|A|} adj(A) \quad (5)$$

Y a su vez, se define a la matriz adjunta como la matriz transpuesta de los cofactores:

$$adj(A) = cof(A)^T \quad (6)$$

Dónde cada cofactor es el determinante de la matriz disminuida, de orden (n-1), obtenida a partir de la eliminación de la fila y columna del elemento para el que se está realizando la operación. Esto se expresa como:

$$d_{ij} = (-1)^{i+j} \det \hat{A}(i, j) \quad (7)$$

Todos estos conceptos se resumen en la siguiente expresión:

$$A^{-1} = \begin{pmatrix} a_0 & a_1 & a_2 \\ a_3 & a_4 & a_5 \\ a_6 & a_7 & a_8 \end{pmatrix} = \frac{1}{\det(A)} \begin{pmatrix} a_4 a_8 - a_7 a_5 & a_2 a_7 - a_1 a_8 & -a_2 a_4 + a_1 a_5 \\ -a_3 a_8 + a_6 a_5 & -a_2 a_6 + a_0 a_8 & a_2 a_3 - a_0 a_5 \\ a_3 a_7 - a_6 a_4 & a_1 a_6 - a_0 a_7 & a_0 a_4 - a_1 a_3 \end{pmatrix} \quad (8)$$

## 6. Matrices ortogonales

Se le llama matriz ortogonal a aquella matriz cuadrada cuya matriz inversa es la misma que su matriz transpuesta (Grossman, 1996).

$$A^{-1} = A^t \quad (9)$$

Al contar con esta particularidad se deriva que el determinante de estas matrices es  $\pm 1$ , como se muestra a continuación.

$$\begin{aligned} \text{Si } A^{-1} = A^t &\rightarrow A^t A = I_{(10)} \\ \det(A^t A) &= \det A^t \cdot \det A = \det A \cdot \det A = (\det A)^2 \quad (11) \\ 1 &= \det I = \det A^t \cdot A = (\det A)^2 \quad (12) \\ \therefore \det A &= \pm 1 \quad (13) \end{aligned}$$

Este atributo nos ayuda a reducir el número de operaciones en la ecuación (8), facilitando así la obtención de la matriz inversa.

Algunos ejemplos de matrices con determinante igual a +1 son los siguientes:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} 2 & 5 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 2 \\ 2 & 4 & 3 \\ 4 & 11 & 5 \end{pmatrix} \begin{pmatrix} 2 & 0 & 1 \\ 1 & 1 & 5 \\ 3 & 0 & 2 \end{pmatrix}$$

# 7. Expansión de las fórmulas a matrices de mayor tamaño

Para resolver matrices de mayor tamaño se debe plantear las ecuaciones necesarias para resolver su determinante y matriz inversa. Esta expansión crece de forma muy significativa. Un ejemplo de de una matriz de 4x4 se muestra en las siguientes ecuaciones.

## a) Matrices 4x4

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} = \begin{vmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} \\ a_{12} & a_{13} & a_{14} & a_{15} \end{vmatrix}$$

### a. Determinante:

$$+a_0 \begin{vmatrix} a_5 & a_6 & a_7 \\ a_9 & a_{10} & a_{11} \\ a_{13} & a_{14} & a_{15} \end{vmatrix} - a_4 \begin{vmatrix} a_1 & a_2 & a_3 \\ a_9 & a_{10} & a_{11} \\ a_{13} & a_{14} & a_{15} \end{vmatrix} + a_8 \begin{vmatrix} a_1 & a_2 & a_3 \\ a_5 & a_6 & a_7 \\ a_{13} & a_{14} & a_{15} \end{vmatrix} - a_{12} \begin{vmatrix} a_1 & a_2 & a_3 \\ a_5 & a_6 & a_7 \\ a_9 & a_{10} & a_{11} \end{vmatrix}$$

### b. Matriz inversa:

$$\frac{1}{\det(A)} \left( \begin{array}{cccc} + \begin{vmatrix} a_5 & a_6 & a_7 \\ a_9 & a_{10} & a_{11} \\ a_{13} & a_{14} & a_{15} \end{vmatrix} & - \begin{vmatrix} a_1 & a_2 & a_3 \\ a_9 & a_{10} & a_{11} \\ a_{13} & a_{14} & a_{15} \end{vmatrix} & + \begin{vmatrix} a_1 & a_2 & a_3 \\ a_5 & a_6 & a_7 \\ a_{13} & a_{14} & a_{15} \end{vmatrix} & - \begin{vmatrix} a_1 & a_2 & a_3 \\ a_5 & a_6 & a_7 \\ a_9 & a_{10} & a_{11} \end{vmatrix} \\ - \begin{vmatrix} a_4 & a_6 & a_7 \\ a_8 & a_{10} & a_{11} \\ a_{12} & a_{14} & a_{15} \end{vmatrix} & + \begin{vmatrix} a_0 & a_2 & a_3 \\ a_8 & a_{10} & a_{11} \\ a_{12} & a_{14} & a_{15} \end{vmatrix} & - \begin{vmatrix} a_0 & a_2 & a_3 \\ a_4 & a_6 & a_7 \\ a_{12} & a_{14} & a_{15} \end{vmatrix} & + \begin{vmatrix} a_0 & a_2 & a_3 \\ a_4 & a_6 & a_7 \\ a_8 & a_{10} & a_{11} \end{vmatrix} \\ + \begin{vmatrix} a_4 & a_5 & a_7 \\ a_8 & a_9 & a_{11} \\ a_{12} & a_{13} & a_{15} \end{vmatrix} & - \begin{vmatrix} a_0 & a_1 & a_3 \\ a_8 & a_9 & a_{11} \\ a_{12} & a_{13} & a_{15} \end{vmatrix} & + \begin{vmatrix} a_0 & a_1 & a_3 \\ a_4 & a_5 & a_7 \\ a_{12} & a_{13} & a_{15} \end{vmatrix} & - \begin{vmatrix} a_0 & a_1 & a_3 \\ a_4 & a_5 & a_7 \\ a_8 & a_9 & a_{11} \end{vmatrix} \\ - \begin{vmatrix} a_4 & a_5 & a_6 \\ a_8 & a_9 & a_{10} \\ a_{12} & a_{13} & a_{14} \end{vmatrix} & + \begin{vmatrix} a_0 & a_1 & a_2 \\ a_8 & a_9 & a_{10} \\ a_{12} & a_{13} & a_{14} \end{vmatrix} & - \begin{vmatrix} a_0 & a_1 & a_2 \\ a_4 & a_5 & a_6 \\ a_{12} & a_{13} & a_{14} \end{vmatrix} & + \begin{vmatrix} a_0 & a_1 & a_2 \\ a_4 & a_5 & a_6 \\ a_8 & a_9 & a_{10} \end{vmatrix} \end{array} \right)$$

## b) Matrices 5x5





conjunto de unos y ceros se separan para representar los valores individuales de la matriz, y así poder realizar las operaciones necesarias y obtener el determinante de dicha matriz, en la figura 6 se muestran las variables que se declaran para recibir estos valores.

$$\begin{pmatrix} a_0 & a_1 & a_2 \\ a_3 & a_4 & a_5 \\ a_6 & a_7 & a_8 \end{pmatrix}$$

**Figura 6.** Variables utilizadas para la distribución de los valores contenidos en el vector de entrada.

A cada una de estas variables se le asignan dos bits de la cadena inicial, como se indica en la figura 7.

```
a0<=A(1 downto 0);
a1<=A(3 downto 2);
a2<=A(5 downto 4);
a3<=A(7 downto 6);
a4<=A(9 downto 8);
a5<=A(11 downto 10);
a6<=A(13 downto 12);
a7<=A(15 downto 14);
a8<=A(17 downto 16);
```

**Figura 7.** Asignación de los bits de la cadena de entrada a las variables declaradas.

En la figura 8 se muestra la expresión en VHDL utilizada para la obtención del determinante.

```
det=((a0*a4*a8)+(a1*a5*a6)+(a2*a3*a7))-((a2*a4*a6)+(a5*a7*a0)+(a1*a3*a8));
```

**Figura 8.** Sintaxis de la instrucción para calcular el determinante.

Respecto a la matriz inversa, ésta se calcula con la ayuda de señales auxiliares, y su cadena final de bits se obtiene a partir de la unión de resultados individuales de los elementos que la componen, mismos que se muestran en la figura 9.

$$\begin{aligned}b_0 &\leq a_4 * a_8 - a_7 * a_5; \\b_1 &\leq a_2 * a_7 - a_1 * a_8; \\b_2 &\leq a_1 * a_5 - a_2 * a_4; \\b_3 &\leq a_5 * a_6 - a_3 * a_8; \\b_4 &\leq a_0 * a_8 - a_2 * a_6; \\b_5 &\leq a_2 * a_3 - a_0 * a_5; \\b_6 &\leq a_3 * a_7 - a_6 * a_4; \\b_7 &\leq a_1 * a_6 - a_0 * a_7; \\b_8 &\leq a_4 * a_0 - a_1 * a_3;\end{aligned}$$

**Figura 9.** Señales auxiliares para recibir los resultados individuales para la matriz inversa.

Estos resultados individuales se asignan a nueve señales de salida para observar su comportamiento, dicha asignación se muestra en la figura 10.

$$\begin{aligned}c_0 &\leq b_0; \\c_1 &\leq b_1; \\c_2 &\leq b_2; \\c_3 &\leq b_3; \\c_4 &\leq b_4; \\c_5 &\leq b_5; \\c_6 &\leq b_6; \\c_7 &\leq b_7; \\c_8 &\leq b_8;\end{aligned}$$

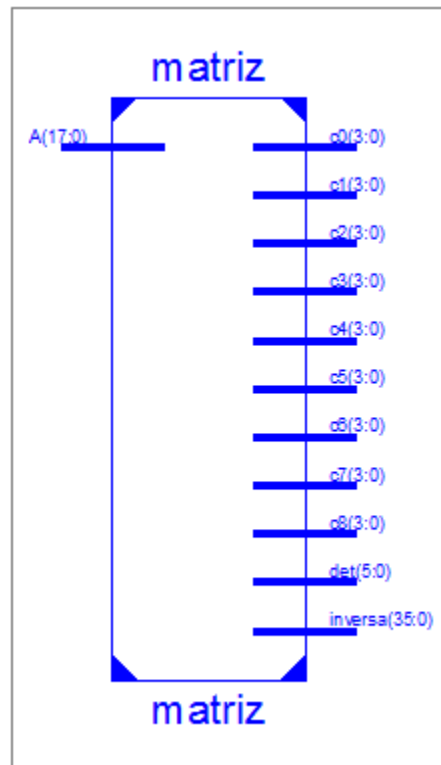
**Figura 10.** Asignación de los resultados individuales a las señales de salida, para observar su comportamiento en la simulación.

En la figura 11, observamos que los resultados individuales son concatenados, generando así la cadena final de bits.

```
inversa<= b8 & b7 & b6 & b5 & b4 & b3 & b2 & b1 & b0;
```

**Figura 11.** Concatenación de los resultados individuales.

A continuación se muestra en la figura 12 la vista esquemática generada por el software ISE de Xilinx, donde observamos que el sistema tiene una entrada de 18 bits, una salida del determinante calculado de 5 bits, mientras que la salida de la matriz inversa es de 35 bits. Observamos también la salida de las señales con los resultados individuales.



**Figura 12.** Diagrama de entradas y salidas del bloque general.

Con el fin de observar los tiempos de retardo y ocupaciones de dos FPGA's diferentes, se selecciona a la XC7A100T en su empaquetado CSG324 de la familia Artix 7 y a la XC6SLX4L en el empaquetado TQG144 de la familia Spartan 6 Low-Power.

*Spartan* Despliegue completo del esquemático RTL.

*Artix* Despliegue completo del esquemático RTL.

## 9. Resultados de calcular el determinante de una matriz

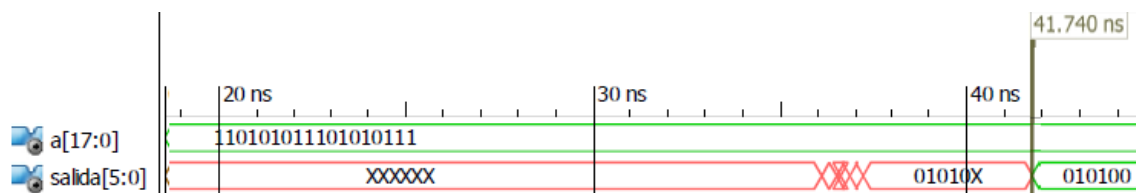
Al declarar el vector de entrada se representa a la matriz utilizada para poner a prueba el desempeño de los dispositivos reconfigurables seleccionados, como se muestra en la figura 13, cabe mencionar que el determinante de dicha matriz es 20.

$$\begin{vmatrix} 3 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 3 \end{vmatrix} \rightarrow "1101010111101010111"$$

*Figura 13.* Representación de la matriz como vector de entrada como para las FPGA's.

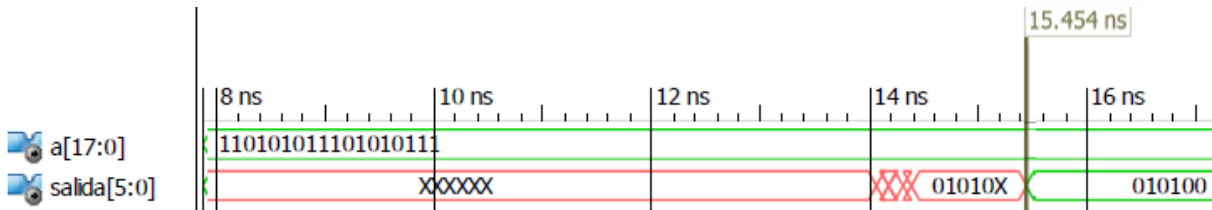
Los resultados que se obtienen al momento de calcular el determinante de la matriz planteada anteriormente, se observan en las figuras 14 y 15.

En el caso de la Spartan 6 Low-Power, observamos en la figura 14 la simulación que nos muestra el resultado fijo a partir de los 41.740 nanosegundos.



**Figura 14.** Determinante calculado en 41.740ns.

En cambio, la FPGA de la familia Artix 7 nos proporciona el resultado desde los 15.454 nanosegundos, tal como se muestra en la figura 15.



**Figura 15.** Simulación del determinante calculado en 15.454ns.

Estos resultados se resumen en la tabla 1.

**Tabla 1.** Comparación de los tiempos de desempeño de las dos FPGA para el cálculo del determinante.

	Determinante
<b>Artix 7</b>	15.454 ns
<b>Spartan 6 Low-Power</b>	41.740 ns

Los recursos utilizados en el circuito para el cálculo del determinante en las FPGAs Artix 7 y Spartan 6 se muestran en la tabla 2, en ésta se puede observar que el número de slices utilizados en ambas familias de FPGAs es similar, pero debido a la disponibilidad de recursos en la familia Artix 7 se considera despreciable la ocupación del circuito, en tanto en la FPGA Spartan 6 alcanzo a ser considerado como el 1% de utilización de los recursos del dispositivo. De manera similar los DSP embebidos utilizados en ambas familias de FPGAs se puede observar que en la Spartan 6 la ocupación de éstos fue del 87% quedando solo uno disponible.

**Tabla 2.**Comparación del uso de las FPGA en el cálculo del determinante.

<b>Uso del dispositivo</b>			
<b>ARTIX 7</b>			
Utilización Lógica	<b>Usado</b>	<b>Disponible</b>	<b>Utilizado</b>
Número de registros Slice	0	126,800	0%
Número de Slices LUT's	12	63,400	0%
Número de DSP48E1s	25	240	10%
<b>SPARTAN 6 LOW-POWER</b>			
Utilización Lógica	<b>Usado</b>	<b>Disponible</b>	<b>Utilizado</b>
Número de registros Slice	0	4,800	0%
Número de Slices LUT's	12	2,400	1%
Número de DSP48A1s	7	8	87%

## 10. Resultados de calcular una matriz inversa

De manera similar que en el punto anterior, la matriz que se invertirá se representa mediante la declaración de la cadena de bits mostrada en la figura 16. El conjunto de bit que se interpreta como la matriz resultante se observa en la figura 17; cabe mencionar que el determinante de la matriz original es 1.

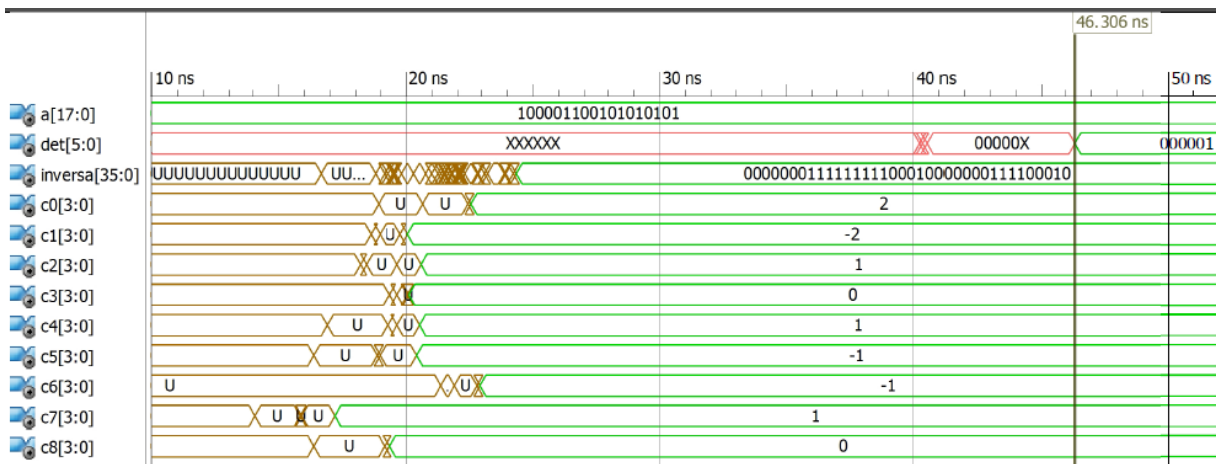
$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix} \rightarrow \text{"100001100101010101"}$$

**Figura 16.** Matriz original y su representación en cadena de bits.

$$\begin{pmatrix} 2 & -2 & 1 \\ 0 & 1 & -1 \\ -1 & 1 & 0 \end{pmatrix} \rightarrow \text{"000000011111111100010000000111100010"}$$

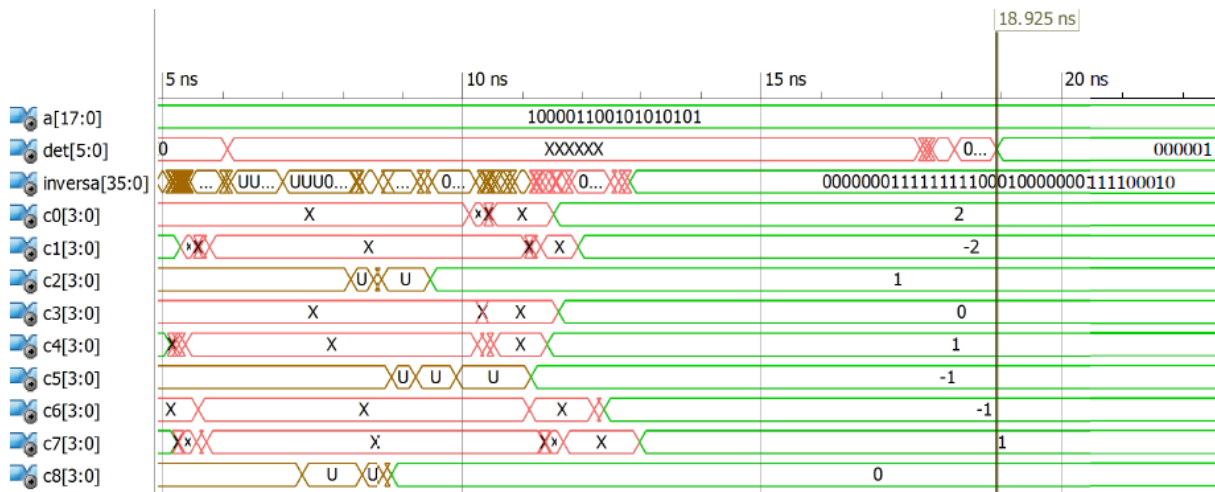
**Figura 17.** Matriz inversa y su representación en unos y ceros.

Para el caso de la FPGA Spartan 6 Low-Power podemos observar que los valores de la matriz inversa se muestran tanto individual como colectivamente, donde la cadena completa se calculó en un total de 24.330 ns, (representada en la figura 18 con "inversa[35:0]"), mientras que el determinante ("det[5:0]") se calcula en 46.306 ns. Como se observa se calcula más rápidamente la matriz inversa que el determinante, a razón de una diferencia de 21.976 ns.



**Figura 18.** Obtención de los elementos de la matriz inversa, el último valor se obtiene a los 23ns.

En el caso de la familia Artix 7, notamos que la cadena que representa a la matriz inversa se obtiene en 12.803 ns, mientras que la determinante se calculó en 18.925, la diferencia es de 6.122 ns, esto se observa en la figura 19.



**Figura 19.** Obtención de los elementos de la matriz inversa, el último valor se obtiene a los 13ns.

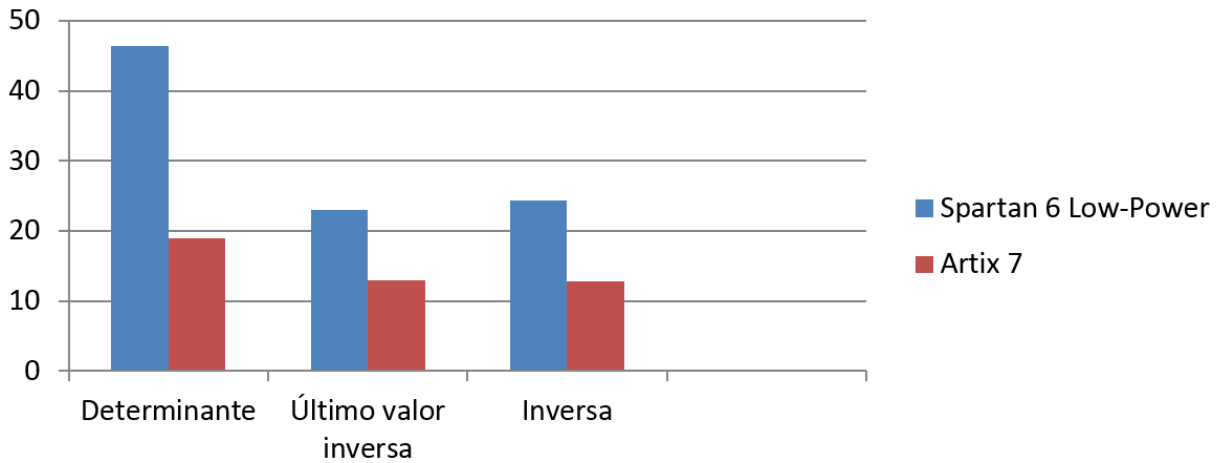
A partir de los resultados obtenidos de las figuras 18 y 19 podemos formar la tabla 3 para observar los datos arrojados por las simulaciones:

**Tabla 3.** Comparación de tiempos promedio de desempeño de ambas FPGA.

	Determinante	Último valor de la Inversa	Inversa
<b>Artix 7</b>	18.925 ns	12.969 ns	12.803 ns
<b>Spartan 6 Low-Power</b>	46.306 ns	22.926 ns	24.330 ns

A partir de las figuras 18, 19 y de la tabla 3 podemos construir la gráfica 1, donde se visualiza la comparación de los distintos tiempos, medidos en nanosegundos de ambas FPGAs.





**Gráfica 1.** Comparación de los distintos tiempos, medidos en nanosegundos.

Los recursos utilizados en el circuito para el cálculo del determinante más la matriz inversa en las FPGAs Artix 7 y Spartan 6 se muestran en la tabla 4, en ésta se puede observar que el número de slices utilizados en ambas familias de FPGAs, Los recursos en la familia Artix 7 se considera del 1% de ocupación del circuito, en tanto en la FPGA Spartan 6 es de 3% de utilización de los recursos del dispositivo.

**Tabla 4.** Comparación del uso de las FPGA en el cálculo del determinante más la matriz inversa.

Uso del dispositivo			
ARTIX 7			
Utilización Lógica	Usado	Disponible	Utilizado
Número de registros Slice	0	126,800	0%
Número de Slices LUT's	25	63,400	1%
SPARTAN 6 LOW-POWER			
Utilización Lógica	Usado	Disponible	Utilizado
Número de registros Slice	0	4,800	0%

**Tabla 4.** Comparación del uso de las FPGA en el cálculo del determinante más la matriz inversa.

<b>Uso del dispositivo</b>			
Número de Slices LUT's	77	2,400	3%

## 11. Conclusiones

El cálculo de matrices computacionalmente hablando son labores que implican grandes cantidades de operaciones, por lo que son costosas la mayoría de las veces, tal como observamos en la realización del circuito de este trabajo. Una propuesta de solución es tratar de resolver el cálculo de las operaciones en forma paralela, esto contribuye de forma directa en el tiempo de procesamiento, y una herramienta como son las FPGAs permiten separar y paralelizar este tipo operaciones.

Como resultados de la implementación de los circuitos podemos concluir que la familia Spartan 6 Low-Power registra tiempos de procesamiento mayores, compensando así su bajo consumo de energía, en comparación con el buen desempeño que tiene la familia Artix 7, quienes registran los tiempos más cortos, tanto en el caso del determinante como en el de la matriz inversa.

De igual forma se resalta que en ambas familias registran un menor tiempo en el cálculo de la matriz inversa en comparación con la obtención del determinante.

## Referencias

Almalki, S. (2013). *New parallel algorithms for finding determinants of NxN matrices. Computer and Information Technology (WCCIT), 2013 World Congress on. Sousse.*

Eberly, W. (2000). *On Computing the Determinant and Smith Form of an Integer Matrix. Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on. Redondo Beach, CA.*

Grossman, S. I. (1996). *Álgebra lineal (5a. ed.). México: McGraw-Hill.*

Holanda, B. (2011). An FPGA-Based Accelerator to Speed-Up Matrix Multiplication of Floating Point Operations. *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on* (págs. 306-309). Shangai: IEEE.

Jovanovic, Z. (2012). FPGA accelerator for floating-point matrix multiplication. *Computers & Digital Techniques, IET, 6(4), 249-256.*

Lei, X. (2014). Cloud Computing Service: the Case of Large Matrix Determinant Computation. *Services Computing, IEEE Transactions on, PP(99), 1-.*

Tai, Y.-G. (2012). Accelerating Matrix Operations with Improved Deeply Pipelined Vector Reduction. *Parallel and Distributed Systems, IEEE Transactions on, 23(2), 202-210.*

Wang, X. (2003). Performance Optimization of an FPGA-Based configurable multiprocessor for matrix operations. *Field-Programmable Technology (FPT), 2003. Proceedings. 2003 IEEE International Conference on* (págs. 303-306). IEEE.

Xilinx. (19 de Noviembre de 2014). Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics. Obtenido de [http://www.xilinx.com/support/documentation/data\\_sheets/ds181\\_Artix\\_7\\_Data\\_Sheet.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds181_Artix_7_Data_Sheet.pdf)

Xilinx. (30 de Enero de 2015). Spartan-6 FPGA Data Sheet: DC and Switching Characteristics. Obtenido de [http://www.xilinx.com/support/documentation/data\\_sheets/ds162.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds162.pdf)

Yang, H. (2007). FPGA-based Vector Processing for Matrix Operations. *Information Technology, 2007. ITNG '07. Fourth International Conference on. Las Vegas, NV.*

## Notas biográficas:



**Francisco Javier Plascencia Jauregui** Recibió el grado de Ingeniero en Computación con orientación a Sistemas Digitales de la Universidad de Guadalajara, México en 2012. Actualmente es estudiante de la Maestría en Ciencias en Electrónica y Computación en el Centro Universitario de Ciencias Exactas e Ingenierías de la

Universidad de Guadalajara. Su área de investigación es el diseño de circuitos electrónicos.



**Juan José Raygoza Panduro** Estudió la licenciatura en Ingeniería en Comunicaciones y Electrónica en la Universidad de Guadalajara, recibió el Grado de Maestro en Ciencias en el Centro de Investigación y Estudios Avanzados del IPN, Zacatenco, México. Sus estudios Doctorado los realizó en Informática y Telecomunicaciones en la Escuela Politécnica Superior de la Universidad Autónoma Universidad de Madrid, España. También trabajó en IBM, Participó en la transferencia tecnológica de la planta de Fabricación de Discos Duros de IBM, en San José California a Planta GDL. Sus áreas de investigación son arquitecturas de microprocesadores, neuroprocesadores, System On Chip y estructuras digitales basadas en FPGAs, así como Sistemas Electrónicos Aplicados a la Biomedicina, Control Digital, y sistemas embebidos. Actualmente es Profesor Investigador del departamento de Electrónica, del CUCEI, Universidad de Guadalajara.



**Susana Ortega Cisneros** Ingeniero en Comunicaciones y Electrónica egresado de la Universidad de Guadalajara, México, su maestría la realizó en el Centro de Investigación y Estudios Avanzados Estudios del IPN, Zacatenco México. Susana Ortega recibió su grado de Doctor en la Escuela Politécnica Superior de la Universidad Autónoma de Madrid, España, en la especialidad de Informática y Telecomunicaciones. Ella se especializa en el diseño digital y basado en FPGAs, y DSPs. Las principales líneas de investigación son Control Digital, Self-Timed, Sistemas Embebidos, Diseño de Microprocesadores, Aceleradores de Cálculo y MEMs. Actualmente es Investigadora del Centro de Investigación y Estudios Avanzados Estudios del IPN Unidad Guadalajara.



**Edwin Christian Becerra Álvarez** Recibió el grado de Ingeniero en Comunicaciones y Electrónica de la Universidad de Guadalajara, México en 2004, el grado de Maestro en Ciencias en

Ingeniería Eléctrica con Especialidad en Diseño Electrónico del CINVESTAV, México en 2006, diploma de estudios avanzados o suficiencia investigadora en Microelectrónica de la Universidad de Sevilla, España en 2008 y el grado de Doctor en Microelectrónica de la Universidad de Sevilla, España en 2010. Miembro SNI Candidato y Perfil Deseable PROMEP. Desde 2010 ha estado trabajando en la Universidad de Guadalajara, donde sus líneas de investigación son el diseño de Circuitos Integrados analógicos, de señal mezclada, digitales, radio frecuencia.



Esta obra está bajo una licencia de Creative Commons Reconocimiento-NoComercial-CompartirIgual 2.5 México.