

## 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:** Destermnants, 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:

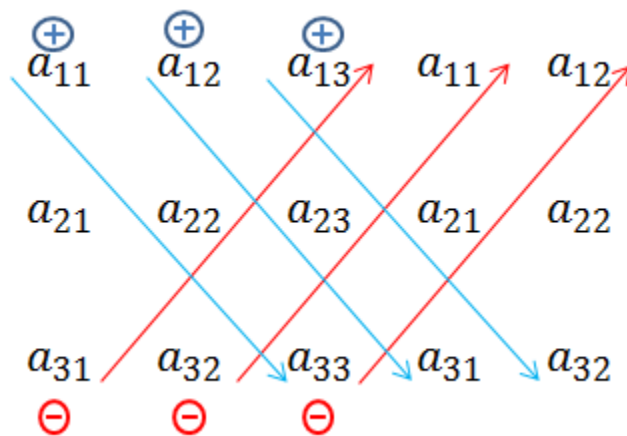


Figura 3. Regla de Sarrus para determinantes de matrices 3x3.

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 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.

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

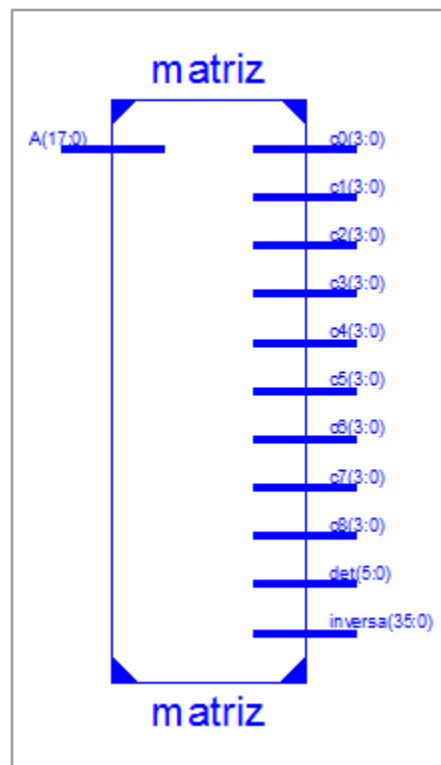
**Figura 10.** Asignación de los resultados individuales a las señales de salida, para observar su comportamiento en la simulación.

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

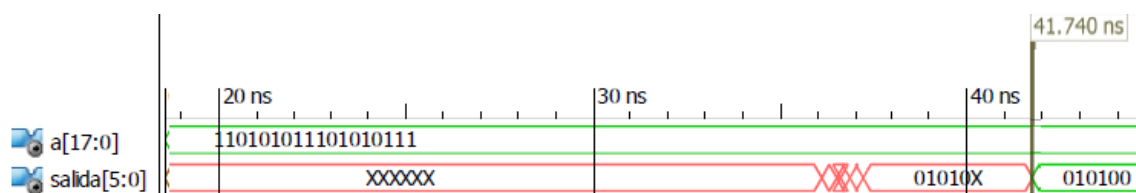
Al declarar el vector de entrada se representa a la matriz utilizada para poner a prueba el desempeño de los dispositivos reconfigurables seleccionados, como se muestra en la figura 13, cabe mencionar que el determinante de dicha matriz es 20.

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

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

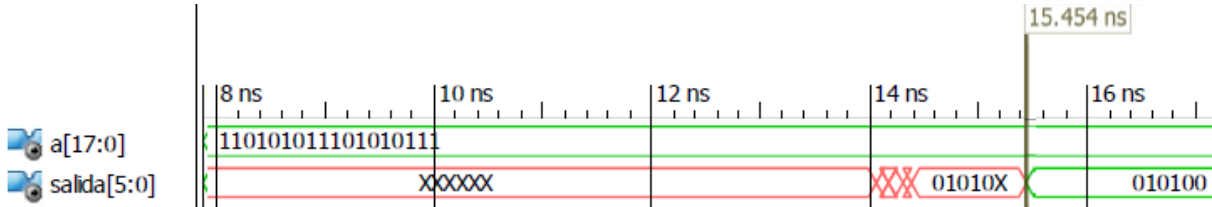
Los resultados que se obtienen al momento de calcular el determinante de la matriz planteada anteriormente, se observan en las figuras 14 y 15.

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.

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

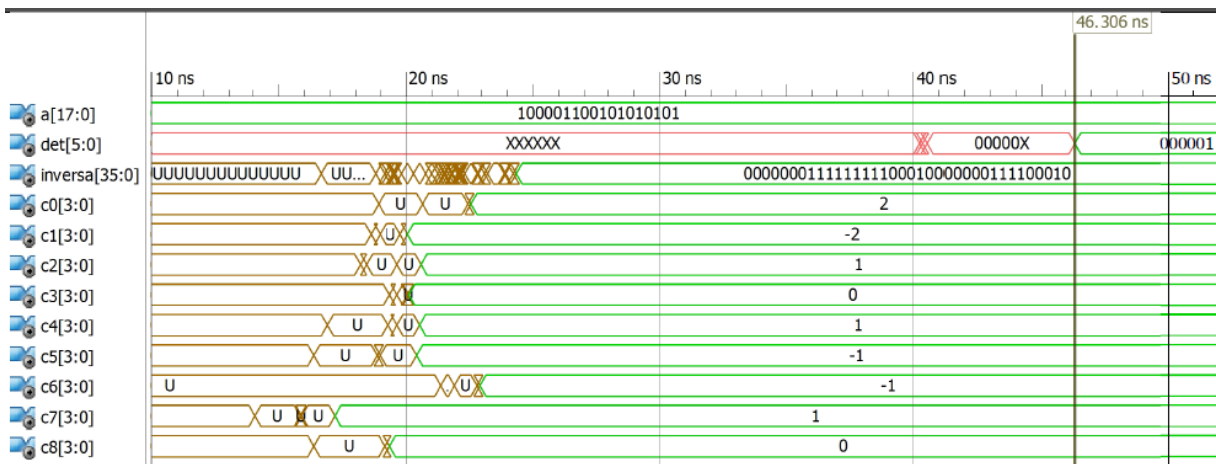
**Figura 16.** Matriz original y su representación en cadena de bits.



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

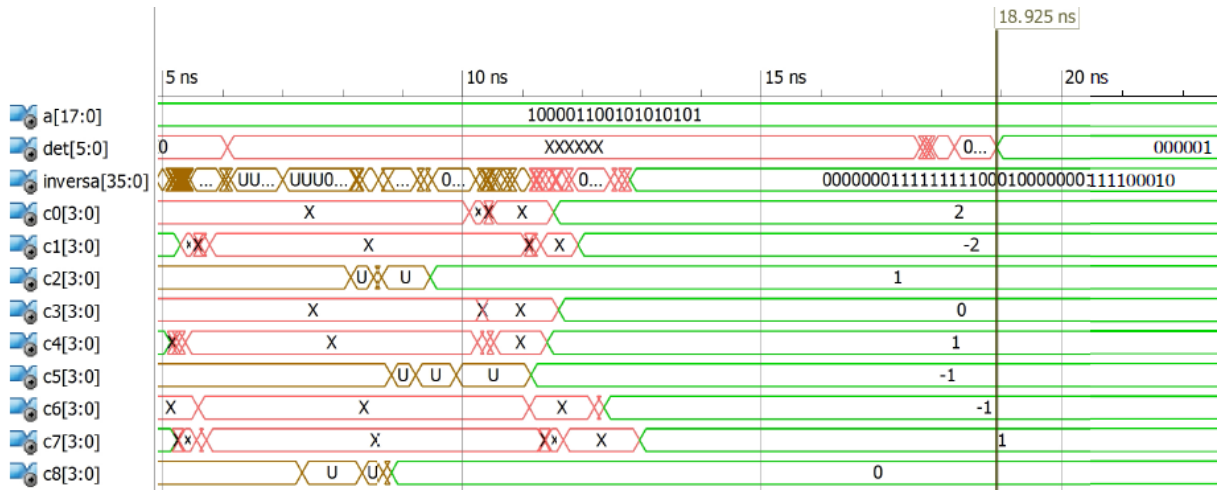
**Figura 17.** Matriz inversa y su representación en unos y ceros.

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.



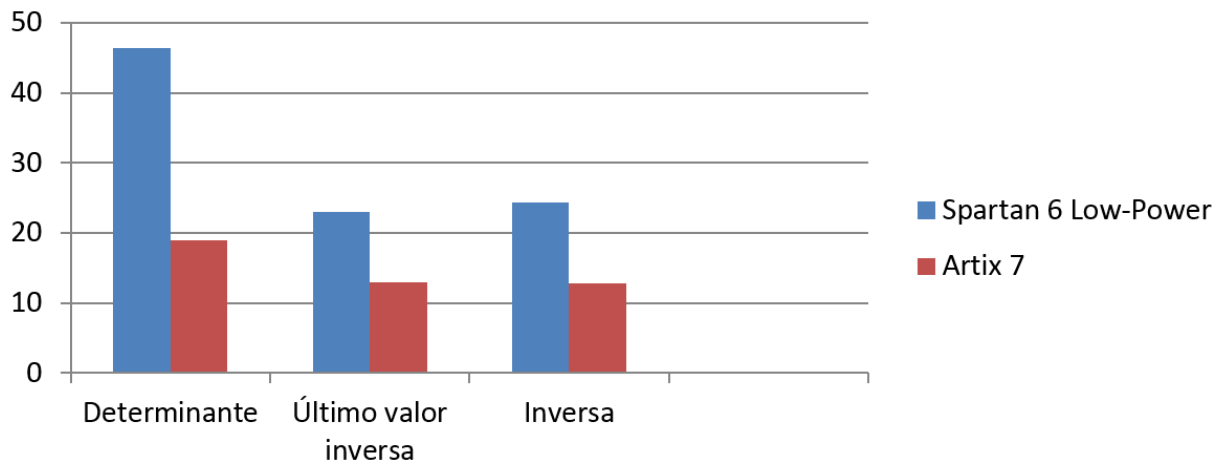
**Figura 19.** Obtención de los elementos de la matriz inversa, el último valor se obtiene a los 13ns.

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 FGAs.



**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.

Uso del dispositivo			
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.