

# Implementación de una Unidad de Punto Flotante basado en el algoritmo BKM

## **Tesista**

Ignacio Lesser, *Padrón Nro. 90.942*  
[ignacio.lesser@gmail.com](mailto:ignacio.lesser@gmail.com)

## **Director**

Ing. Nicolas Alvarez, *Profesor Adjunto*  
[nalvarez2001@yahoo.com.ar](mailto:nalvarez2001@yahoo.com.ar)

## **Co-director**

Ing. Octavio Alpago, *JTP. Interino*  
[ocalpago@gmail.com](mailto:ocalpago@gmail.com)

5 de junio de 2016

## 1. Objeto y Área de la Tesis

El objetivo principal de este trabajo consiste en diseñar e implementar en hardware digital una unidad de punto flotante basada en el algoritmo BKM. Dicho diseño e implementación será realizado a través del uso del lenguaje de descripción de hardware Verilog, y su síntesis en un dispositivo FPGA <sup>1</sup>. El interés en este desarrollo proviene de la necesidad de crear una unidad de punto flotante eficiente y con gran capacidad de cálculo para utilizarse como periférico de un microprocesador diseñado en la misma facultad. Con este proyecto no solamente se logra cubrir las operaciones básicas de una FPU sino que se va un paso más lejos y se agregan una serie de operaciones extras que expanden la funcionalidad de la unidad así como también la posibilidad de trabajar en el campo de los números complejos. Estas características surgen naturalmente al utilizar el algoritmo BKM. La principal ventaja de utilizar una FPU es la capacidad de implementar operaciones matemáticas complejas en tiempo real.

El área profesional de relevancia de la presente tesis es el diseño de sistemas digitales y su aplicación en el ámbito de los microprocesadores, aunque no se limita sólo a ese ámbito. Una vez comprendido el funcionamiento del algoritmo y analizado sus características se podrá utilizar el conocimiento en la aplicación práctica en hardware de algoritmos de procesamiento de señales en general.

---

<sup>1</sup>Field Programmable Gate Array: dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada ‘in situ’ mediante un lenguaje de descripción especializado.

## 2. Introducción. Antecedentes

### 2.1. Unidad de punto flotante

Hablar sobre FPU???? hace falta explicar xq se necesita una? o sobre su historia?

### 2.2. Algoritmo BKM

El algoritmo BKM [1] fue desarrollado en 1994 por Jean-Claude Bajard, Sylvanus Kla y Jean-Michel Muller (con las primeras letras de cada apellido se puede formar la sigla BKM) y es parte de la familia de los denominados algoritmos de desplazamiento y suma o *Shift and add algorithms* en inglés. Estos algoritmos intentan explotar características particulares de un problema para poder reducirlo en operaciones simples como sumas y desplazamientos. Estas operaciones pueden ser implementadas muy facilmente en un circuito digital lo que permite reducir considerablemente la complejidad del circuito a utilizar.

El algoritmo BKM está intimamente relacionado con el método de Briggs para el cálculo del logaritmo (Henry Briggs, 1600 apróx.) y el algoritmo CORDIC [2] creado por Jack E. Volder en 1956. De manera simplista se lo puede pensar como una mejora del último, ya que resuelve un problema muy similar pero con numerosas mejoras y/o ventajas (ACA TENGO QUE PONER ALGO SOBRE LAS VENTAJAS?? O LO DEJO PARA LA TESIS?).

El algoritmo de Briggs para el cálculo de  $\ln(x)$  dice que si se encuentra una secuencia  $d_k$  tal que la productoria de  $x$  con  $(1 + d_k 2^{-k})$  es cercana a 1 entonces vale que:

$$\begin{aligned} x \prod_{k=1}^n (1 + d_k 2^{-k}) &\approx 1 \\ \ln(x) &\approx - \sum_{k=1}^n \ln(1 + d_k 2^{-k}) \end{aligned} \quad (2)$$

El algoritmo CORDIC realizaba rotaciones en coordenadas circulares. Partiendo de esa base es fácil extender su funcionamiento para que realice rotaciones en coordenadas hiperbólicas y lineales [3]. Para lograrlo se agrega una variables que modifica las ecuaciones y además se eligen diferentes angulos para el acumulador de angule (variable  $z$  del algoritmo). Las ecuaciones del CORDIC completo son:

$$\begin{aligned} x_{n+1} &= x_n - m \cdot d_n \cdot 2^{-n} \\ y_{n+1} &= y_n + d_n \cdot 2^{-n} \\ z_{n+1} &= z_n - d_n \cdot \alpha_{m,n} \end{aligned} \quad (4)$$

Donde  $N$  representa la cantidad de pasos del algoritmo y se cumple que  $n = 0, 1, 2, \dots, N - 1$ .  $\alpha_{m,n}$  representa los angulos rotados para las diferentes coordenadas.  $d_n$  es una variable de control que maneja los sumadores/restadores.

Consideremos el paso básico del algoritmo CORDIC en modo trigonométrico (con  $m=1$ ). Si definimos el número complejo  $E_n = x_n + j y_n$  con  $j = \sqrt{-1}$ , obtenemos  $E_{n+1} = E_n(1 + j d_n 2^{-n})$ , esta relación es similar al paso básico del algoritmo de Briggs. Esta similitud nos lleva a una generalización de ese algoritmo: podríamos realizar multiplicaciones por terminos  $(1 + d_n 2^{-n})$ , donde los  $d_n$ s son números complejos elejidos de tal manera que la multiplicación por  $d_n$  pueda reducirse a unas pocas sumas. Entonces se define el algoritmo BKM de la siguiente manera:

$$\begin{cases} E_{n+1} = E_n \cdot (1 + d_n 2^{-n}) \\ L_{n+1} = L_n - \ln(1 + d_n 2^{-n}) \end{cases} \quad (5)$$

con  $d_n = d_n^r + j d_n^i$  y  $d_n^r, d_n^i \in \{0, \pm 1\}$  y  $\ln z = t$  es el número complejo  $t$  tal que  $\exp t = z$  y cuya parte imaginaria está entre  $-\pi$  y  $\pi$ . Este algoritmo nos permitirá calcular de manera natural logaritmos y exponenciales en el campo de los números complejos.

## 3. Desarrollo previsto de la Tesis

### 3.1. Teoría, enfoque y métodos a utilizar

El enfoque de la tesis se basará en una parte teórica y una experimental. Se analizarán los modelos y desarrollos algorítmicos para la implementación del algoritmo BKM y luego se realizará un

análisis para el diseño de un IP core que ponga en práctica uno de ellos. Una vez codificado el hardware, se procederá a realizar su síntesis en un dispositivo FPGA utilizando un kit de desarrollo, se realizarán los bancos de prueba correspondientes y se analizarán los resultados obtenidos.

### 3.2. Estudios conexos

Asignaturas y otros estudios previstos que son relevantes al desarrollo de la Tesis.

- **Sistemas Digitales:** Asignatura que abarca la teoría de técnicas de diseño de hardware digital y codificación de sistemas digitales, así como también su síntesis y medición.
- **Procesamiento de Señales:** Asignatura que abarca los conceptos respecto a la implementación de los algoritmos.

### 3.3. Alcance proyectado para la tesis

Como resultados a obtener de la presente tesis se tienen los siguientes:

- Código fuente completamente sintetizable del diseño y todos sus subbloques.
- Ejemplos de instanciación.
- Scripts de simulación.
- Bancos de prueba, incorporando chequeos automáticos del tipo PASA/FALLA.
- Resultado de mediciones pertinentes al diseño del IP.
- Hoja de datos.
- Detalle de la arquitectura usada.
- Análisis comparativo de procesamiento entre el IP desarrollado y desarrollos de terceros.
- Proposición de trabajos futuros y/o mejoras.

Para asegurar que el proyecto de tesis incluya todos los trabajos requeridos, los procesos a completar se describen en un plan de trabajo en la siguiente sección.

### 3.4. Plan de trabajo

La duración total del trabajo se estima en un año y se considera que la misma estará compuesta por las siguientes etapas:

- **Investigación Bibliográfica:** Recolección de Libros, Papers, Trabajos de Tesis, y fuentes de investigación con el objetivo de obtener el entendimiento teórico requerido y conocer el estado del arte en el tema a trabajar.
- **Introducción al trabajo de tesis:** Comprender la teoría de una unidad de una FPU y del algoritmo BKM. Describir el contenido teórico requerido para exponer los conceptos de funcionamiento del hardware a desarrollar.
- **Análisis de Arquitecturas:** Analizar las diferentes arquitecturas digitales propuestas para implementar el sistema. Elegir en base a un criterio una de ellas para realizar el desarrollo.
- **Estudio de mejoras:** Estudiar la factibilidad de realizar una mejora a la arquitectura antes mencionadas.
- **Implementación:** Implementar la arquitectura seleccionada en Verilog y desarrollar los bancos de prueba de simulación para verificar su correcta funcionalidad. Se generará un IP core en RTL para implementar el sistema. Dicho RTL cumplirá con ciertas condiciones de portabilidad y legibilidad del código para que el mismo sea efectivamente un IP core.
- **Verificación y validación:** Experimentar el IP core en un ambiente de simulación y en campo. Realizar la síntesis de la misma para distintos dispositivos FPGA, medir recursos utilizados, máxima frecuencia de operación y potencia consumida.
- **Conclusiones y Trabajos a Futuro:** Se extraerán las conclusiones pertinentes sobre los resultados obtenidos y se propondrán futuras mejoras de la arquitecturas (si correspondiese) a partir de los resultados obtenidos.
- **Preparación del Informe Final:** Se consolidará la documentación con la memoria de la tesis la cual contendrá el resultado de todo el trabajo realizado. Se revisará la tesis por el director y por los pares antes de enviarla al jurado.
- **Preparación de la presentación y defensa de la Tesis:** Se preparará la presentación (diapositivas) con los objetivos, alcance, introducción, desarrollo de la arquitectura, resultados obtenidos, conclusiones y trabajos a futuro. La misma será posteriormente utilizada para la defensa.

En la figura 3.4 se encuentra el plan propuesto para la formación del tesista y el cumplimiento de los objetivos.

Actividad	Mes											
	1	2	3	4	5	6	7	8	9	10	11	12
Investigación Bibliográfica	■											
Introducción al trabajo de tesis		■	■									
Análisis de Arquitecturas				■	■							
Estudio de mejoras					■	■	■	■	■			
Implementación						■	■	■	■	■		
Verificación y validación									■	■	■	
Conclusiones y Trabajos a Futuro											■	■
Preparación del Informe Final		■	■	■	■	■	■	■	■	■	■	
Preparación de la presentación y defensa de la Tesis												■

Figura 1: Diagrama de las actividades a realizar por el tesista.

## 4. Bibliografía

### Referencias

- [1] J.M. Muller, J.C. Bajard y S. Kla, “BKM: a new hardware algorithm for complem elementary functions”, IEEE Transactions on Computers, 1994.
- [2] J. Volder, “The cordic computing technique”, IRE Trans. Elect. Comput., 1959. Reimpreso en Computer Arithmetic, vol. 1, E.E. Swartzlander, Ed. CA: IEEE Computer Society Press Tutorial, 1990.
- [3] J. Walther, “A unified algortithm for elemetary functions”, Joint Comput. Conf. Proc., 1971. Reimpreso en Computer Arithmetic, vol. 1, E.E. Swartzlander, Ed. CA: IEEE Computer Society Press Tutorial, 1990.

Buenos Aires, 5 de junio de 2016

---

Sr. Ignacio Lesser  
Tesista

---

Ing. Nicolás Alvarez  
Prof. Adjunto, Director

---

Ing. Octavio Alpago  
JTP. Interino, Co-director