

# Proyecto Integrador

---



Autor

Gomez, Gomez - Lovaisa Michelini Valeria

Tema

**Implementación de un sistema SoC con microprocesado  
OpenRisc con soporte Linux**

—

---

# ÍNDICE GENERAL

<b>1. INTRODUCCIÓN</b>	<b>3</b>
1.1. Descripción General . . . . .	3
1.2. Objetivos . . . . .	3
1.2.1. Objetivo General . . . . .	3
1.2.2. Objetivo Específico . . . . .	3
1.3. Motivación . . . . .	4
1.4. Metodología . . . . .	5
1.5. Importancia del Problema . . . . .	5
1.6. Alcance de Estudio . . . . .	5
1.7. Modelo de Desarrollo . . . . .	5
1.8. Metodología . . . . .	5
1.9. Organización del Proyecto Integrador . . . . .	5

---

# ÍNDICE DE FIGURAS

---

---

# CAPÍTULO 1

---

## INTRODUCCIÓN

### 1.1. Descripción General

### 1.2. Objetivos

#### 1.2.1. Objetivo General

Implementar un system on chip OpenSource con un microprocesador embebido Soft-core que soporte un sistema operativo libre , con la finalidad de entregar un sistema integral FPGA-SoC-Sistema Operativo completamente funcional y bajo licencia GPL v2.

#### 1.2.2. Objetivo Específico

- Evaluar, seleccionar y validar las prestaciones de los Kit de desarrollos con FPGA disponibles en el área de trabajo.
- Obtener un System on Chip completamente funcional sobre un kit de desarrollo XILINX XtremeDSP Starter Platform Spartan 3A DSP 1800.
- Implementar un Sistema Operativo en el Kit de desarrollo XILINX XtremeDSP Starter Platform Spartan 3A DSP 1800.
- Probar el adecuado funcionamiento de el sistema global que tenga las prestaciones funcional tradicionales de diseño

### 1.3. Motivación

Desde el comienzo de la tecnología de semiconductores, los sistemas electrónicos han experimentando un constante crecimiento en complejidad, en consecuencia las prestaciones que tiene que ofrecer una aplicación concreta son cada vez mayores y de naturalezas más diversas. Esto ha dado lugar a que un sistema tenga a la vez restricciones aparentemente incompatibles, como pueden ser de trabajo en tiempo real, de tolerancia a fallos o de procesamiento de grandes flujos de datos. Este incremento de complejidad y diversidad en las restricciones ha motivado al desarrollo de diseños híbridos en los que interactúan elementos hardware y software, dichos elementos pueden complementarse para resolver problemas de distinta naturaleza.

Entre las alternativas para diseñar e implementar un hardware específico se encuentran FPGAs (Fieldprogrammable gate array) circuitos de alta densidad programables por el usuario en un tiempo reducido y sin la necesidad de verificación de sus componentes, tarea ya realizada por el fabricante al tratarse de un producto estándar.

Los lenguajes que se utilizan para diseñar e implementar un sistema digital sobre una FPGA proporcionan gran versatilidad para el desarrollo de hardware, permitiendo especificar, diseñar, simular y verificar sistemas digitales complejos, mediante el apoyo de un universo de herramientas EDA (Electronic Design Automation) Las arquitecturas reconfigurables de las FPGA combinan parte de la flexibilidad del software con la gran performance del hardware.

Los a veces requieren nuevas características de los cores existentes. El proveedor del núcleo puede hacer estas modificaciones (solución comercial) con un incremento sustancial del coste del núcleo. Otra posibilidad (Solución Ad-hoc) es el uso de cores de código abierto con el fin de crear un núcleo de desarrollo adaptable. El enfoque de código abierto tiene varias ventajas: el núcleo posee un costo muy bajo e inclusive cero, los usuarios puede tener acceso al código fuente y hay un grupo de desarrolladores que proporcionan conocimientos para mantener y mejorar el núcleo.

Los microprocesadores “softcore”(núcleo software) son aquellos cuyo hardware está íntegramente implementado en un lenguaje de descripción de hardware. Posibilitan un desarrollo confiable con facilidad de ajustar el mismo a las necesidades de aplicación que deben satisfacerse e incluso el desarrollo de microprocesadores doble o múltiple núcleo. Entre los microprocesadores softcore encontramos los Open Source (Código Abierto) uno de ellos OpenRISC de Open Cores. El procesador OpenRISC de 32 bits se comunica con sus periféricos por medio de un bus tipo Wishbone , también de código abierto. OpenRISC puede ser implementado fácilmente en chips de cualquier fabricante (Xilinx , Altera , Actel) razón por la cual es el elemento central de algunos proyectos SoC de código abierto como MinSoC y de OpenCores u ORPSoC de OpenCores. Toda la implementación del proyecto posee una licencia LGPL lo que otorga al programador la capacidad de modificar el código a necesidad. Un conjunto de herramientas de desarrollo de software (compilador, ensamblador, debugger) también de código abierto ayudan al desarrollo de aplicaciones para esta arquitectura y son provistas en el proyecto OpenRISC.

Si bien la placa seleccionada, ZedBoard, soporta tanto Android como GNU/Linux nos motivó utilizar este último por lo siguiente: Es el SO que por defecto viene instalado en la placa. Es un SO abierto con vasta documentación detallada y una gran comunidad. La mayoría de los proyectos documentados con la utilización de esta placa han sido con GNU/Linux. Todas las guías provistas por el fabricante se basan en este SO.

Que el sistema operativo sea multitarea. 2 Que el sistema operativo disponga de mecanismos de sincronización. 3 Que el sistema operativo soporte arquitecturas de sistemas embebidos. Por

todo esto, y puesto que se cumplen los requerimientos enunciados en el apartado 8.2, es que decidimos desarrollar este proyecto sobre GNU/Linux.

### **1.4. Metodología**

### **1.5. Importancia del Problema**

El softcore OpenRisc que se encuentra en el SoC OrpSoc y MinSoc se tiene que implementar en una Spartan 3A de Xilinx. Tenemos como fin montar un Linux para validar y verificar el sistema global entregando un sistema funcional bajo licencia libre. Actualmente las FPGAs nos brindan la posibilidad de implementar estos proyectos, donde el Hardware y el Software son una misma entidad. Este nuevo enfoque nos permite aprovechar la facilidad de implementar soluciones por Hardware.

### **1.6. Alcance de Estudio**

### **1.7. Modelo de Desarrollo**

### **1.8. Metodología**

### **1.9. Organización del Proyecto Integrador**

Una vez detalladas las motivaciones y expuestas las ventajas que un receptor coherente puede aportar a las comunicaciones ópticas, el presente proyecto tiene como principal objetivo el estudio, diseño y la simulación de los diferentes métodos de recuperación de portadora de un receptor digital coherente para lo cual se organiza su contenido de la siguiente manera:

---

# BIBLIOGRAFÍA

- [1] Claude Herard and Alain Lacourt, 1991. *New multiplexing technique using polarization of light.*
- [2] Wikipedia, the free encyclopedia, Free and open source software. <http://en.wikipedia.org>