

# 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 . . . . .	4
1.2.1. Objetivo General . . . . .	4
1.2.2. Objetivo Específico . . . . .	5
1.3. Motivación . . . . .	5
1.4. Metodología . . . . .	6
1.5. Importancia del Problema . . . . .	6
1.6. Alcance de Estudio . . . . .	6
1.7. Modelo de Desarrollo . . . . .	6
1.8. Metodología . . . . .	6
1.9. Organización del Proyecto Integrador . . . . .	6

---

# ÍNDICE DE FIGURAS

1.1. OpenRISC 1200 - Diagrama de Modulos . . . . .	4
--	---

---

# CAPÍTULO 1

---

## INTRODUCCIÓN

### 1.1. Descripción General

Desde los principios de la tecnología de semiconductores, los sistemas electrónicos han venido experimentando un constante crecimiento en complejidad, debido a que 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 la aparición de diseños híbridos en los que interactúan elementos hardware y software, ya que dichos elementos pueden complementarse para resolver problemas de distinta naturaleza.

Entre las alternativas para diseñar e implementar un hardware específico encontramos ASICs (Application-specific integrated circuit), FPGAs (Field-programmable gate array), CPLDs (Complex Programmable Logic Device) entre otros. El uso de ASICs posibilita desarrollos con producción a gran escala a bajo costo y es de masiva utilización en este tipo de aplicaciones. Los CPLD y las FPGA son 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. El procesamiento digital de señales, prototipado de ASICs, tratamiento de imágenes, reconocimiento de voz, glue logic son algunas de las aplicaciones de este tipo de dispositivos. Existen diferentes formas de llevar adelante el diseño e implementación de un sistema digital para FPGA, entre ellas tenemos la realización de un diseño esquemático, herramientas específicas (provistas por el fabricante) y la utilización de un lenguaje de descripción de hardware HDL (Hardware description language) entre los que se encuentran lenguajes como Verilog y VHDL, ambos de gran aceptación en los ambientes industrial y académico. Estos lenguajes proporcionan gran versatilidad para el desarrollo de hardware, permitiendo especificar, diseñar, simular y verificar sistemas digitales complejos,

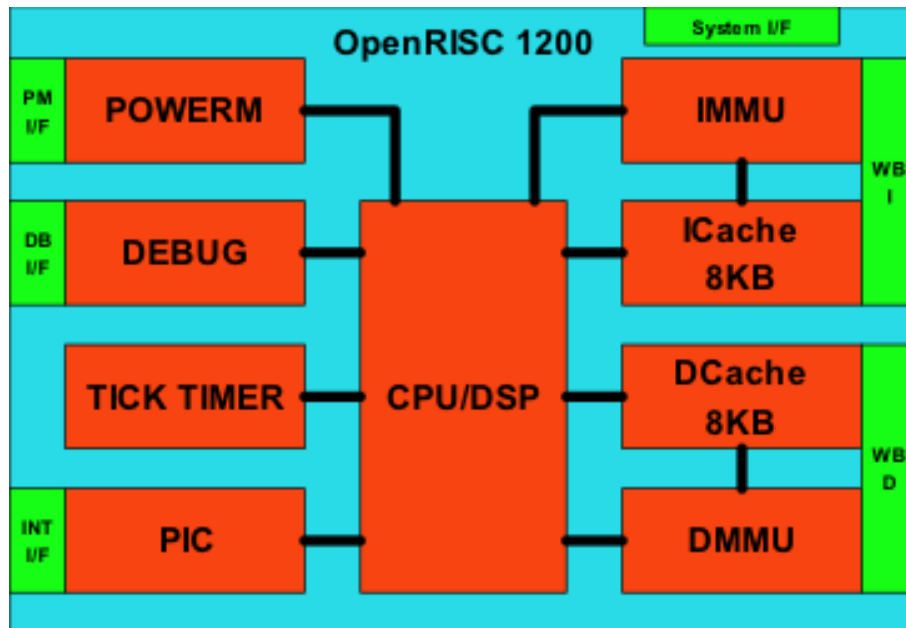


Figura 1.1: OpenRISC 1200 - Diagrama de Módulos

mediante el apoyo de un universo de herramientas EDA (Electronic Design Automation).

Actualmente las FPGA cuentan con una gran cantidad de recursos disponibles (Compuertas lógicas, Bloques de RAM) para implementar diseños digitales complejos. Las FPGA pueden ser usadas para implementar cualquier función lógica que un ASIC pueda realizar. Una de las grandes ventajas del uso de FPGA en la etapa de prototipado es su capacidad de reconfigurar el diseño parcial o totalmente para su actualización o corrección de errores con un costo relativamente bajo a diferencia del prototipado sobre ASICs. Durante la etapa de producción los ASIC resultan de muy bajo costo respecto de la producción de FPGA y esto se traduce en una gran ventaja para desarrollos que deben ser producidos a gran escala.

El desarrollo de aplicaciones de software se ve limitado a los recursos disponibles en los microprocesadores comerciales. El software necesita del soporte de un procesador para su ejecución, así la elección de este elemento conlleva algunas dependencias respecto de las herramientas a utilizar, algunas de estas son: compiladores, ensambladores, depuradores y herramientas de

## 1.2. Objetivos

### 1.2.1. Objetivo General

Implementar un system on chip Open Source con un Softcore OpenRisc que soporte Linux, con la finalidad de entregar un sistema integral FPGA-SoC-Sistema Operativo completamente funcional.

### 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 encontramos 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 combinan parte de la flexibilidad del software con la gran performance del hardware utilizando chips reconfigurables como FPGAs

documentacion y licencias para usar OpenRisc

En caso de que algún usuario o programador encontrase algún error en la aplicación, lo único que podía hacer era darlo a conocer a la empresa desarrolladora para que ésta lo solucionara. Aunque el programador estuviese capacitado para solucionar el problema y lo deseara hacer sin pedir nada a cambio, el contrato le impedía que modificase el software.

Puesto que el software libre permite el libre uso, modificación y redistribución, a menudo encuentra un hogar entre usuarios para los cuales el coste del software no libre es a veces prohibitivo, o como alternativa a la piratería. También es sencillo modificarlo localmente, lo que permite que sean posibles los esfuerzos de traducción a idiomas que no son necesariamente rentables comercialmente.

Las universidades suelen optar por el uso de software libre en vez de utilizar software privativo, porque satisface de una mejor manera sus necesidades de cómputo, dada su naturaleza de

apertura del código y la libertad de compartir los resultados obtenidos. De forma colateral, no se tienen gastos adicionales derivados del pago de licenciamientos.

El software libre no se limita a ser gratuito o de muy bajo coste, porque también tiene un valor social fundamental, puesto que la única restricción que tiene es la de conservarse libre, lo cual quiere decir que puede ser explorado, verificado, reproducido, y extendido, en todas sus capacidades, para beneficio de todos, de forma muy similar a la naturaleza de la producción de la ciencia.

Los nuevos proyectos 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.

### **1.4. Metodología**

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

### **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] [www.wikipedia.com.ar](http://www.wikipedia.com.ar)