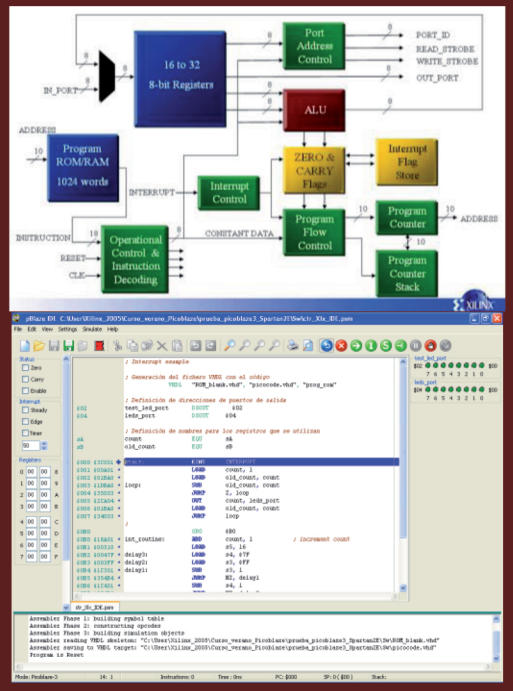
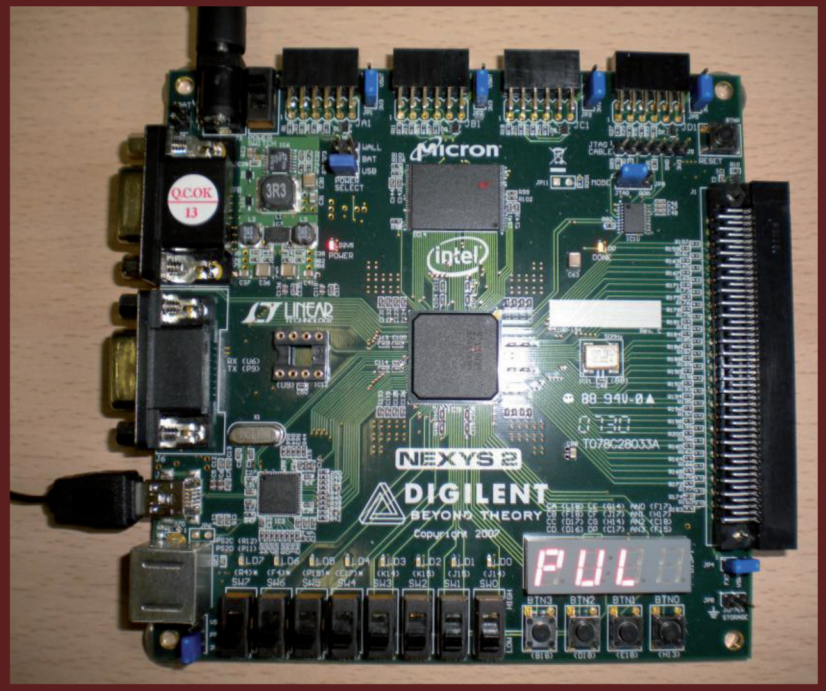
DISEÑO DE SISTEMAS EMPOTRADOS DE 8 BITS EN FPGAs CON XILINX ISE Y PICOBLAZE





Francisco Poza González
Luis Jacobo Álvarez Ruiz de Ojeda

Vision Ebooks

Reservados todos los derechos. Queda rigurosamente prohibida, sin la autorización escrita de los titulares del Copyright bajo las sanciones establecidas de las leyes, la reproducción parcial o total de esta obra por cualquier medio o procedimiento incluidos la reprografía y el tratamiento informático.

© 2012 Diseño de sistemas empotrados de 8 bits en FPGAs con Xilinx ISE y PicoBlaze

© Autores:
Luis Jacobo Álvarez Ruiz de Ojeda
Francisco Poza González

© 2012 Editorial: Vision Ebooks
C/ San Benito, 15, Local 28029 Madrid
Tel: 0034 91 3117696 Web: www.visionebooks.es
Visión Ebooks es marca registrada de Vision Netware S.L.

ISBN: 978-84-940472-0-6

Disponible en ebook
www.visionebooks.com

Pedidos a:
pedidos@visionnet.es

Si quiere recibir información periódica sobre las novedades de nuestro grupo
editor envíe un correo electrónico a:
subscripción@visionnet.es

Dedicatoria

A los alumnos de nuestros cursos de Posgrado, sin los cuales este libro no hubiera sido posible.

ÍNDICE

Página

Introducción 21

Agradecimientos 25

PRESENTACIONES DE TEORÍA 27

Tema 1. Introducción a los Sistemas en un Circuito (SOCs) 29

Introducción al diseño de sistemas digitales complejos.

Método “software”.

Método “hardware”.

Sistema en un Circuito (S.O.C.).

Sistema en un Circuito Programable (P.S.O.C.).

Microprocesadores “hardware”.

Microprocesadores “software”.

Tema 2. Codiseño Hardware / Sotware 71

Introducción.

Codiseño hardware/software.

Fases detalladas del flujo de codiseño.

Particionado hardware/software.

Tema 3. Arquitectura de las FPGAs. Familia Spartan 3E 83

Introducción.

Arquitectura de la familia Virtex 2 de Xilinx.

CLBs. “Slices”.

Memorias.

Multiplicadores “hardware”.

Circuitos de reloj.

Bloques de E/S. Tecnologías de E/S.

Virtex 2 Pro frente a Virtex 2.

Virtex 4 frente a Virtex 2.

Virtex 5 frente a Virtex 4.

Spartan 3 frente a Virtex 2.

Normas de síntesis.

Tema 4. Arquitectura interna Picoblaze 143

Introducción.

Versiones de Picoblaze.

Características generales Picoblaze 3.

Arquitectura interna Picoblaze 3.

Diagrama de bloques.

Resultados de implementación.

Comparación versiones Picoblaze.

Tema 5. Juego de instrucciones Picoblaze 155

Introducción.

Versiones de Picoblaze.

Resumen instrucciones Picoblaze 3.

Otras operaciones.

Tabla instrucciones Picoblaze 3.

Ejercicios.

Tema 6. Arquitectura externa Picoblaze 173

Introducción.

Interfaz de E/S del Picoblaze.

Instrucciones de E/S.

Conexión de periféricos de entrada.

Conexión de periféricos de salida.

Puesta en estado inicial.

Interrupciones externas.

Memoria de programa.

Tema 7. Codiseño HW/SW para sistemas empotrados basados en Picoblaze ... 233

Introducción.

Proceso de diseño de aplicaciones empotradas con Picoblaze.

Proceso de diseño detallado.

Tema 8. Aplicaciones Picoblaze 269

Introducción.

Aplicaciones del Picoblaze.

Tema 9. Recursos web de Xilinx sobre microprocesadores empotrados 277

Introducción.

Página principal de Xilinx.

Página sobre procesadores empotrados de Xilinx.

Páginas sobre Picoblaze 3.

Picoblaze Forum.

Picoblaze IDE.

Tema 10. Análisis de la implementación del Picoblaze 287

Introducción.

Memorias de programa y datos.

Contador de programa.

Memoria pila.

ALU.

Tema 11. Juego de instrucciones detallado del Picoblaze 303

Introducción.

Juego instrucciones comunes.

Instrucciones específicas Picoblaze 3.

Codificación instrucciones Picoblaze 3.

PRESENTACIONES DE LABORATORIO 331

Tema 1. Introducción a la herramienta Foundation ISE de Xilinx 333

Introducción.

Descripción.

Compilación. Síntesis.

Simulación funcional.

Implementación.

Simulación temporal.

Descripción (continuación).

Prueba en la placa.

Placa Nexys 2 de Digilent.

Programación del circuito.

iMPACT.

Adept.

Tema 2. Desarrollo “software” del microprocesador Picoblaze con KCPSM 427

Introducción.

Sintaxis ensamblador.

Programa ensamblador KCPSM de Xilinx.

Directivas KCPSM.

Ensamblador KCPSM.

Ejemplo programa ensamblador.

Ejercicios.

Tema 3. Desarrollo “software” del microprocesador Picoblaze con pBlazeIDE 449

Introducción.

Sintaxis ensamblador.

Programa ensamblador y simulador pBlazeIDE.

Directivas pBlazeIDE.

Menús pBlazeIDE.

Ejemplo programa ensamblador.

Simulación con pBlazeIDE.

Diferencias entre pBlazeIDE y KCPSM.

Ejercicios.

Tema 4. Realización de circuitos de acoplamiento para periféricos de Picoblaze 483

Introducción.

Circuitos de selección de periféricos de entrada.

Circuitos de selección de periféricos de salida.

Circuitos de gestión de interrupciones.

Tema 5. Realización de periféricos para Picoblaze 533

Introducción.

Periféricos básicos.

Interruptores.

Detector de flancos.

Pulsadores.

Registro simple.

Visualizador dinámico.

Memoria RAM dedicada.

Memoria RAM distribuida.

Sincronizador reloj rápido a reloj lento.

Circuitos periféricos estándar.

Temporizador / Contador.

Modulador PWM.

UART RS232.

Propuestas.

Tema 6. Diseño de sistemas empotrados basados en Picoblaze 607

Introducción.

Proceso de diseño de aplicaciones empotradas con Picoblaze.

Ejemplo detallado del proceso de diseño.

Especificaciones.

Particionado HW/SW.

Descripción “hardware”.

Simulación “hardware”.

Depuración “hardware”.

Desarrollo “software”.

Compilación y enlazado.

Simulación “software”.

Depuración “software”.

Combinación HW/SW.

Síntesis e implementación.

Cosimulación HW/SW.

Codepuración HW/SW.

Proceso de diseño alternativo para modificaciones sólo de “software”.

Ejercicios.

Conclusiones.

Tema 7. Depuración de sistemas empotrados con “Chipscope” 669

Introducción.

“Core Chipscope”.

Parámetros “Chipscope”.

Implementación.

Análisis con “Chipscope”.

Conclusiones.

EJEMPLOS DE DISEÑO DE SISTEMAS EMPOTRADOS 703

Ejemplo reloj digital con alarma Spartan 3 705

Enunciado en formato Powerpoint 705

Especificaciones.

Manejo de la aplicación.

Observaciones.

Solución:

Circuitos necesarios.

Programa en ensamblador.

Asignación de puertos de E/S.

Simulación de los circuitos.

Prueba de la aplicación.

Guión de laboratorio 737

Ejemplo sistema de gestión basado en un transmisor/receptor asíncrono RS232 Spartan 3 739

Enunciado en formato Powerpoint 739

Especificaciones.

Manejo de la aplicación.

Observaciones

Puerto serie RS232.

UART RS232.

Solución:

Circuitos necesarios.

Programa en ensamblador.

Asignación de puertos de E/S.

Simulación de los circuitos.

Prueba de la aplicación.

Guión de laboratorio 793

Ejemplo de control de visualización de mensajes a través de un visualizador LCD 797

Enunciado en formato Powerpoint 797

Especificaciones.

Manejo de la aplicación.

Observaciones.

Visualizador LCD.

Solución:

Circuitos necesarios.

Programa en ensamblador.

Asignación de puertos de E/S.

Simulación de los circuitos.

Prueba de la aplicación.

Guión de laboratorio 845

MATERIAL DE CONSULTA 849

Comparación entre las versiones del microprocesador Picoblaze 851

Etapas del proceso de diseño mediante Picoblaze 853

Hojas de planificación de uso de registros para el diseño de sistemas empotrados .. 859

BIBLIOGRAFÍA 869

LOS AUTORES 873

INTRODUCCIÓN

Este libro está dirigido a todos aquellos profesionales de la Electrónica, tanto docentes como técnicos, que deseen introducirse en el diseño de sistemas empotrados de 8 bits en FPGAs ("Field Programmable Gate Arrays"), es decir, sistemas basados en microprocesadores embebidos ("embedded") en el interior de FPGAs.

En este libro se estudian el microprocesador de 8 bits Picoblaze de Xilinx, para su implementación mediante FPGAs de las familias Spartan y Virtex de Xilinx. Para poder trabajar con estos microprocesadores, es necesario aprender el manejo de la herramienta ISE de Xilinx [XILINX].

Este libro tiene su origen en la documentación elaborada para la impartición de varios cursos de posgrado sobre sistemas empotrados de 8 bits de Xilinx para el diseño de aplicaciones empotradas en FPGAs, que se celebraron en el Departamento de Tecnología Electrónica de la Universidad de Vigo en los años 2005 a 2008 (más información en http://www.dte.uvigo.es/logica_programable/cursos.htm).

Este libro tiene un nivel básico y para seguirlo, el lector debe disponer de "software" gratuito de Xilinx [XILINX]. Para el máximo aprovechamiento de este libro, los lectores deben poseer los siguientes conocimientos previos:

- Nivel básico del lenguaje VHDL.
- Manejo de la herramienta ISE de Xilinx para el diseño con FPGAs.
- Conocimientos de microprocesadores. Arquitectura general. Buses.
- Nivel básico de lenguaje ensamblador.

- Manejo básico del sistema operativo Windows.

Si el lector desea un libro avanzado sobre el diseño de sistemas empotrados de 32 bits con “software” que no es gratuito, puede consultar el libro [ÁLVAREZ 11].

El libro se estructura en los siguientes apartados:

- Teoría. Se incluyen las presentaciones en “Powerpoint” utilizadas para explicar los siguientes temas:
  - Introducción al diseño de Sistemas en un Circuito (S.O.C.).
  - arquitectura de las FPGAs de la familia Spartan de Xilinx.
  - Microprocesador “Picoblaze" de Xilinx. Arquitectura interna. Juego de instrucciones. Arquitectura externa.
- Laboratorio. Se incluyen las presentaciones en “Powerpoint” utilizadas para explicar los siguientes temas:
  - Herramienta ISE para el diseño de sistemas digitales con FPGAs de Xilinx.
  - Herramientas “software” del microprocesador Picoblaze de Xilinx.
  - Realización de circuitos de acoplamiento de periféricos para el microprocesador Picoblaze.
  - Realización de circuitos periféricos para el microprocesador Picoblaze.
  - Diseño de sistemas digitales basados en el microprocesador Picoblaze.
- Práctica. Se incluyen las presentaciones en “Powerpoint” utilizadas para explicar los ejemplos y los guiones de laboratorio necesarios para realizar los siguientes ejemplos:
  - Ejemplo básico con interrupción.
  - Reloj digital.
  - Sistema de gestión basado en un transmisor/receptor asíncrono RS232.

- Sistema de visualización de mensajes a través de un visualizador LCD.

Las referencias al CD-ROM incluidas en el texto de este libro se encuentran accesibles para el lector en el enlace http://www.dte.uvigo.es/logica_programable/libros.htm [DTE2].

AGRADECIMIENTOS

Los autores desean agradecer su aportación a las siguientes personas y empresas:

- Xilinx, por su donación al Departamento de Tecnología Electrónica de la Universidad de Vigo de licencias de sus herramientas “software” ISE y EDK, así como de diferentes placas de desarrollo con FPGAs. Estas placas han sido utilizadas para la realización de los diferentes ejemplos. También debemos agradecer su cortesía al ceder las figuras de sus hojas de características y manuales para los distintos capítulos de este libro.

- Silica, por sus gestiones con su representada Xilinx para la obtención de las licencias y placas mencionadas anteriormente, y por su apoyo a la celebración de los cursos que han dado lugar a este libro.

PRESENTACIONES DE TEORÍA

Sistemas en un circuito (S.O.C.)

Índice

- **Introducción al diseño de sistemas digitales complejos.**
  - **Método “software”.**
  - **Método “hardware”.**
- **Sistema en un Circuito (S.O.C.).**
- **Sistema en un Circuito Programable (S.O.P.C.).**
  - **Microprocesadores “hardware”.**
  - **Microprocesadores “software”.**

Introducción (I).

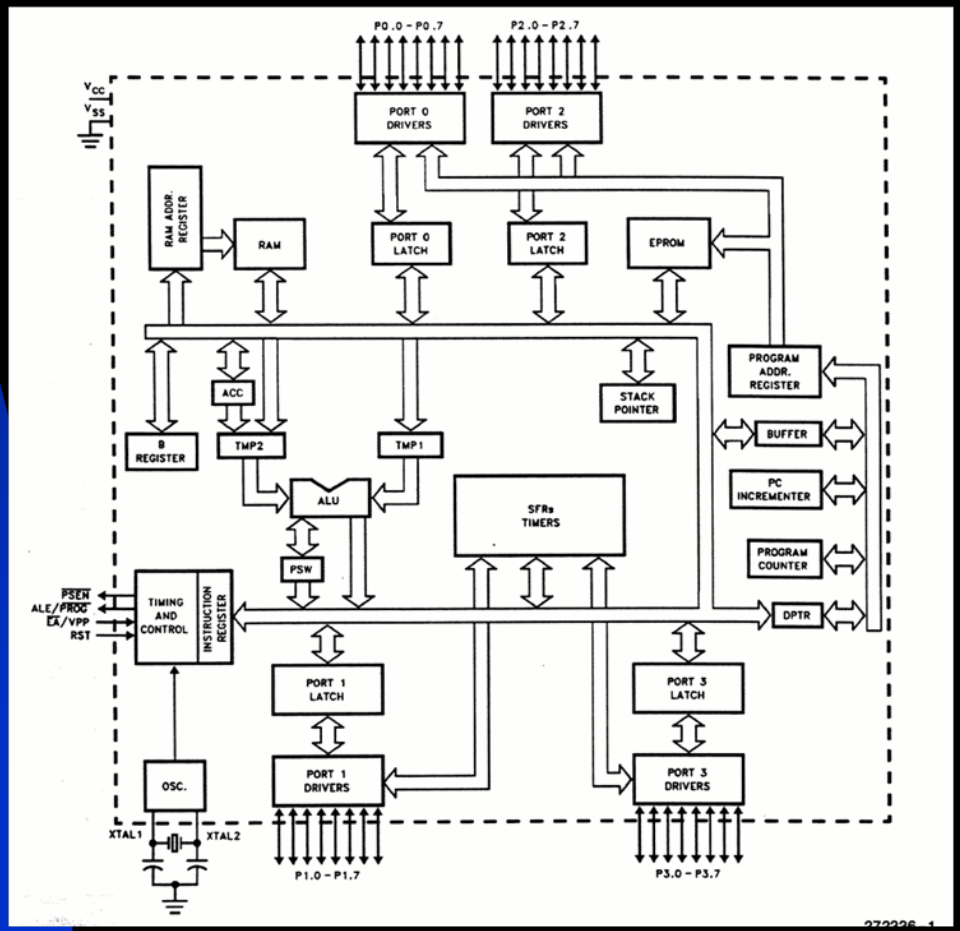
- **El diseño de un sistema digital complejo que permita resolver un determinado problema puede realizarse mediante dos aproximaciones diferentes:**
  - **Método “software”.**
  - **Método “hardware”.**

Método “software” (I).

- **Se utilizan circuitos procesadores estándar, ya sean de uso general (microprocesadores) o de uso específico (DSPs).**
- **Estos circuitos presentan una serie de funciones predefinidas, asociadas a un juego de instrucciones fijo.**
- **El diseñador debe realizar el programa (secuencia de instrucciones) que debe ejecutar el procesador para realizar las funciones requeridas.**

Sistema digital realizado mediante un procesador

- **La arquitectura de los procesadores se basa en:**
  - **Unidad operativa (unidad aritmético-lógica + memoria de datos).**
  - **Unidad de control (búsqueda → decodificación → ejecución).**
  - **Memoria de programa.**



Método “software” (II).

- **La capacidad de los procesadores estándar para realizar sistemas complejos se ve limitada por:**
  - **El procesado secuencial de las instrucciones.**
  - **La arquitectura concreta del procesador utilizado.**

Método “hardware” (I).

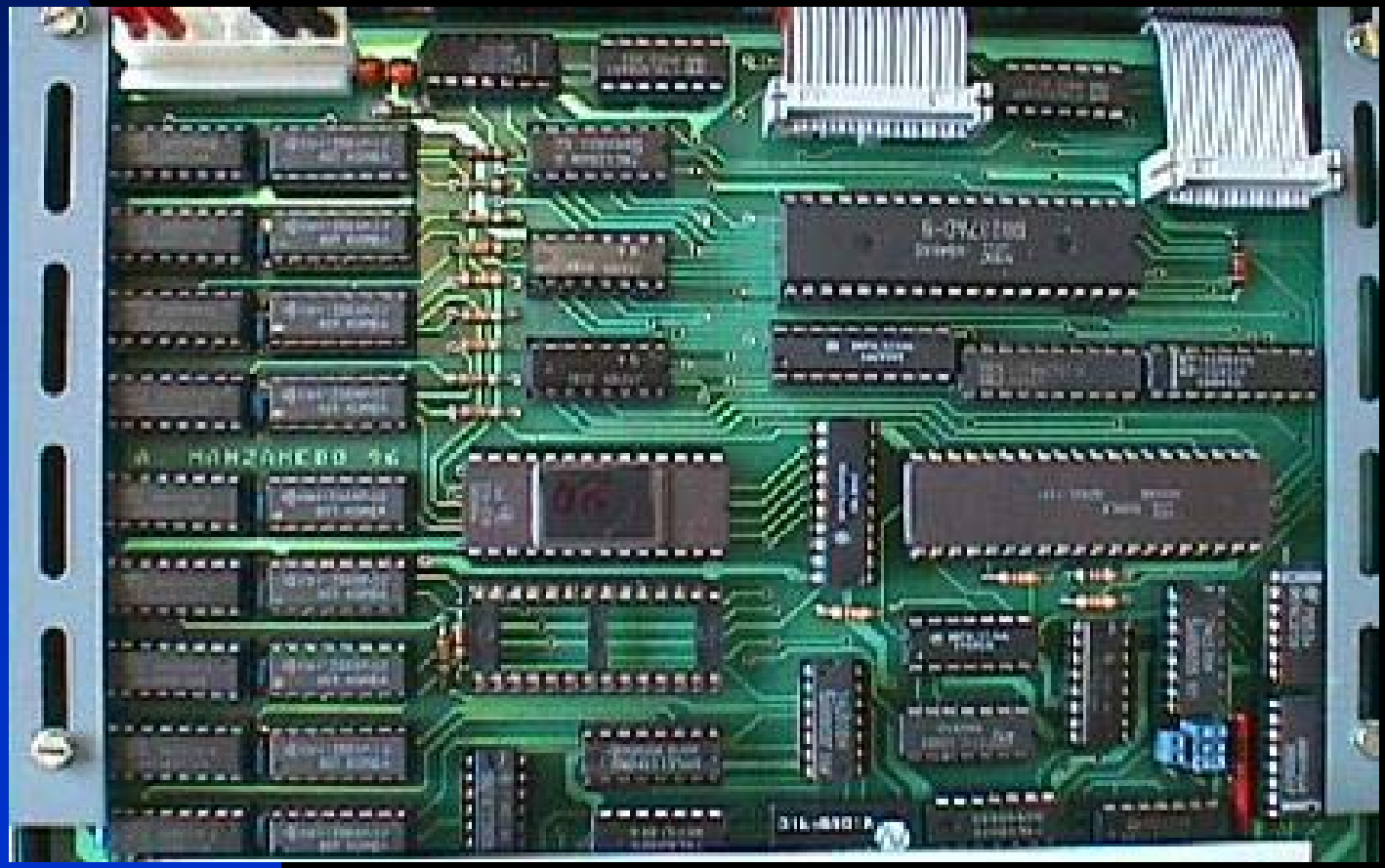
- **Se utilizan circuitos digitales estándar, ya sean de función fija o definible por el usuario.**
- **El diseñador debe definir el sistema digital necesario para realizar las funciones requeridas.**
- **El sistema digital resultante debe implementarse mediante los circuitos elegidos.**

Método “hardware” (II).

- **Tres aproximaciones:**
  - **Circuitos estándar de función fija (método clásico).**
  - **Circuitos programables por el usuario (Lógica Programable: PLDs y FPGAs).**
  - **Circuitos de aplicación específica (ASIC):**
    - **Semimedida (“semi-custom”).**
    - **Totalmente a medida (“full-custom”).**

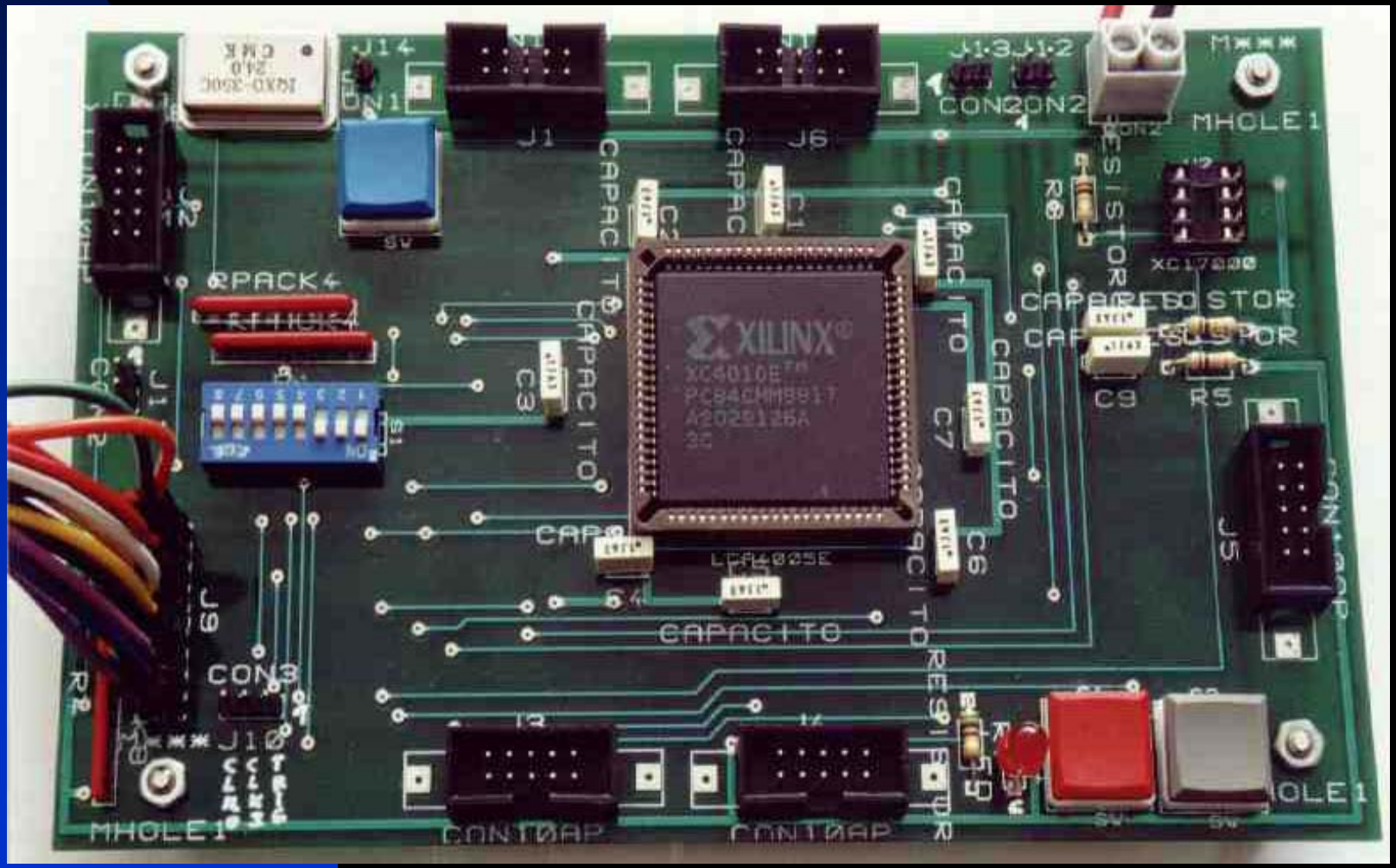
Sistema digital realizado mediante circuitos estándar de función fija

- **Cuando los sistemas digitales son complejos, la utilización de este tipo de circuitos da lugar a una placa con un gran número de circuitos integrados y de interconexiones.**
- **Por ello, actualmente no se utilizan este tipo de circuitos para el diseño de nuevos sistemas digitales.**



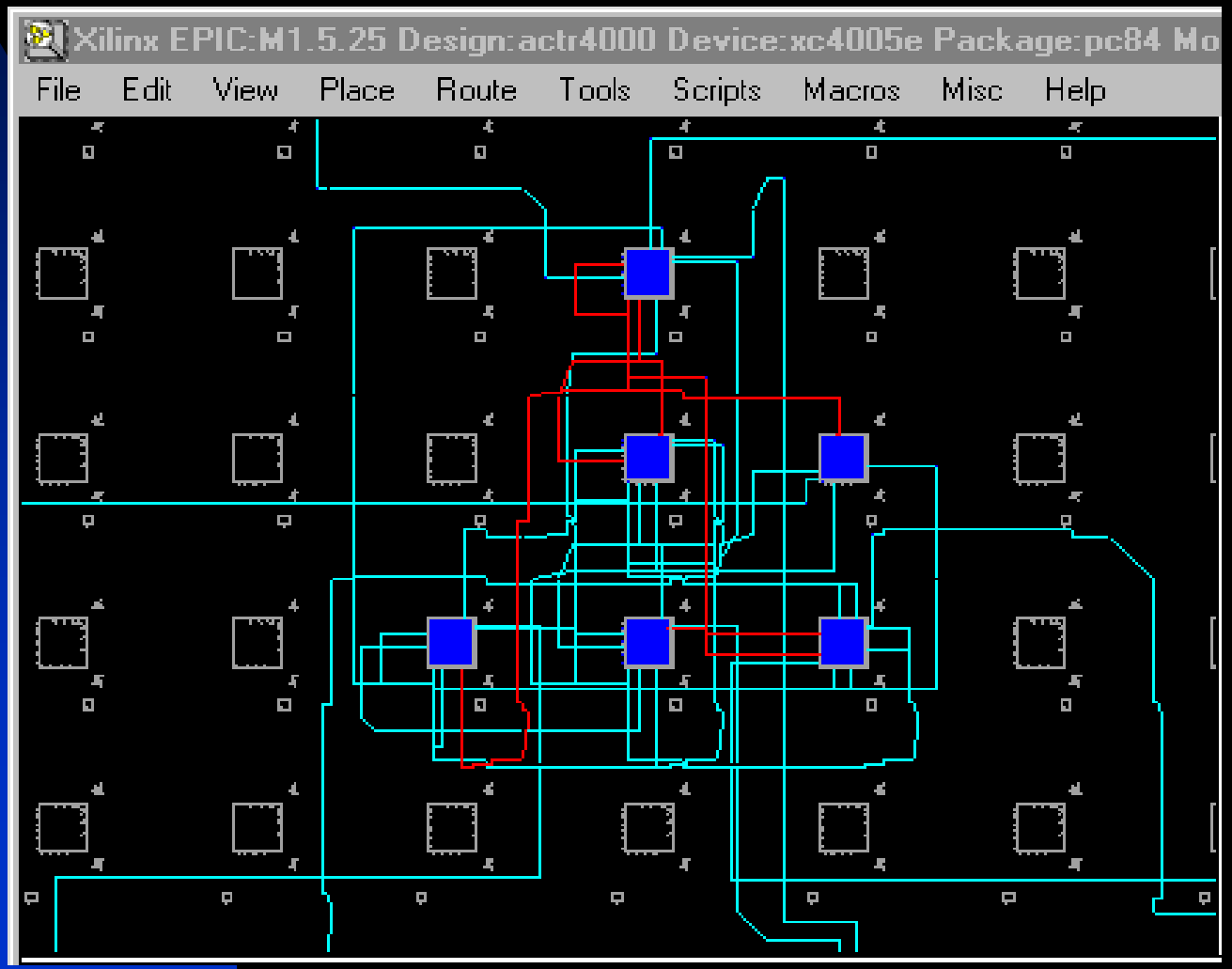
Sistema digital realizado mediante circuitos programables por el usuario (I)

- **La capacidad lógica de estos circuitos permite realizar un sistema digital complejo en un único circuito integrado.**



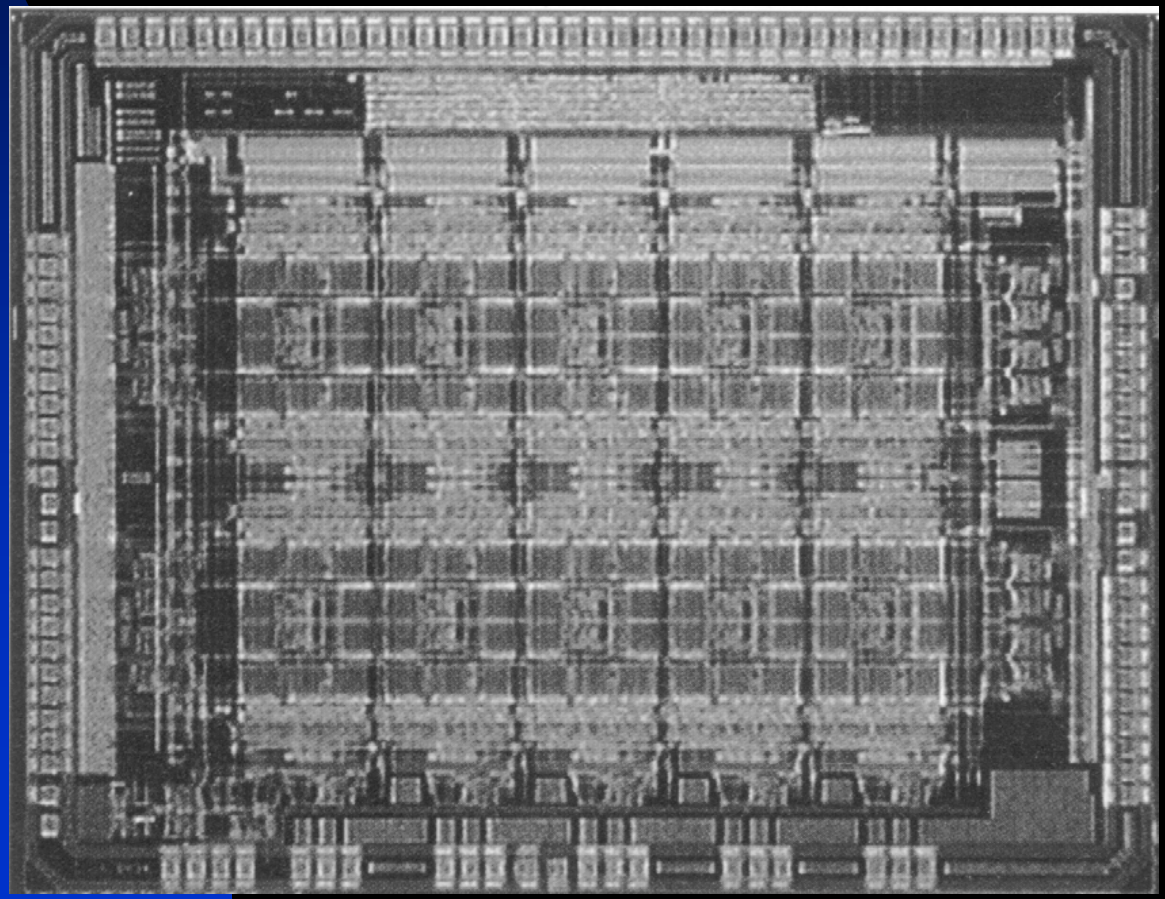
Sistema digital realizado mediante circuitos programables por el usuario (II)

- **Diagrama de bloques interno de un circuito programable del tipo FPGA, en el que se ha implementado un sistema digital.**



Sistema digital realizado mediante un circuito de aplicación específica

- **Dificultad del proceso de diseño.**
- **Necesidad de realizar la fabricación del circuito posteriormente => coste y tiempo de desarrollo elevados.**
- **Circuito totalmente a medida diseñado por el usuario.**



Sistema en un Circuito (SOC) (I)

- **Actualmente, existe una opción intermedia entre las dos aproximaciones, “hardware” y “software”, que se denomina:**

“System On Chip”

o

S.O.C.

Sistema en un Circuito (SOC) (II)

- **Consiste en implementar en un único circuito integrado los siguientes circuitos:**
  - **Un microprocesador.**
  - **Una o varias unidades de memoria (SRAM, FLASH-EEPROM, etc.).**
  - **Circuitos de interfaz estándar.**
  - **Periféricos de entrada y salida.**
  - **Otros sistemas digitales de distinta complejidad, que pueden ser independientes o interactuar con el microprocesador.**

Sistema en un Circuito (SOC) (III)

- **Del microprocesador incluido en un Sistema en un Circuito se suele decir que está empotrado (“embedded”), término que hace referencia a que no es directamente accesible desde el exterior del circuito integrado.**

- **De la misma forma, al diseño de Sistemas en un Circuito que incluyen un microprocesador se le suele denominar Diseño de Sistemas Empotrados (“Embedded System Design”).**

Sistema en un Circuito Programable (PSOC) (I)

- **Las dos opciones que existen actualmente para realizar un SOC son las siguientes:**
  - **Mediante un ASIC que suele incluir una parte configurable por el usuario.**
  - **Mediante una FPGA de gran capacidad lógica.**

- **En este último caso, los SOCs se suelen denominar "System On a Programmable Chip" (SOPC o PSOC).**

Sistema en un Circuito Programable (PSOC) (II)

- **Actualmente es posible implementar en un mismo Circuito Configurable del tipo FPGA:**
  - **Uno o varios microprocesadores.**
  - **Una o varias unidades de memoria (SRAM, FLASH-EEPROM, etc.).**
  - **Los periféricos de entrada y salida de los microprocesadores.**
  - **Otros sistemas digitales complejos, que pueden ser independientes o interactuar con los microprocesadores.**

Sistema en un Circuito Programable (PSOC) (III)

- **Dos posibles implementaciones:**
  - **Bloque microprocesador “hardware”, ya integrado en el Silicio, junto con un circuito configurable del tipo FPGA.**
  - **Bloque microprocesador “software”, prediseñado para su implementación como un circuito más, en una FPGA genérica de gran capacidad.**

Microprocesador “hardware” (I).

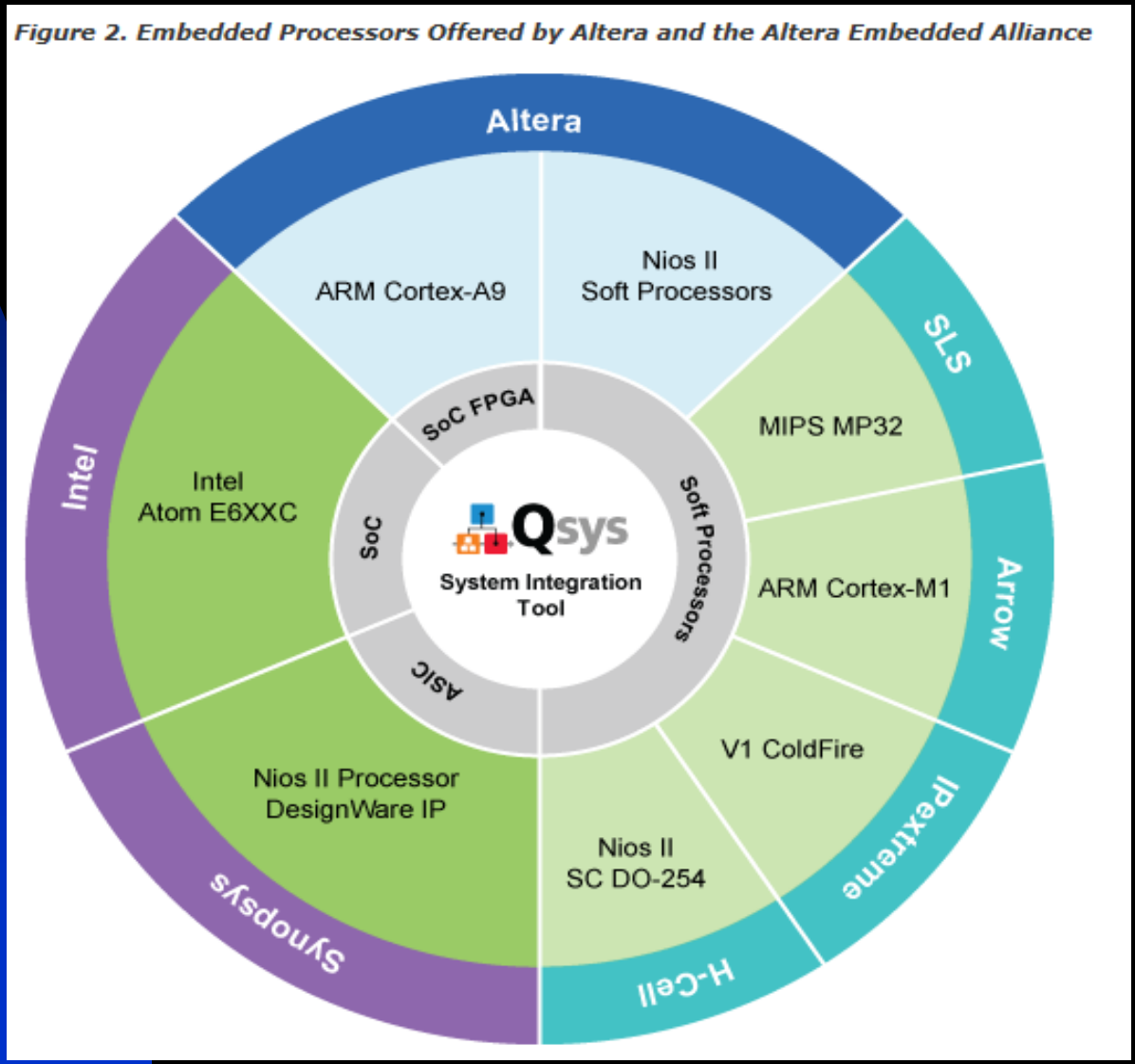
- **SOCs de fabricantes de FPGAs:**
  - **Altera.**
  - **Actel.**
  - **Cypress.**
  - **Xilinx.**



Altera (I)

- **Procesadores empotrados de Altera y empresas asociadas (1 de 2).**

Figure 2. Embedded Processors Offered by Altera and the Altera Embedded Alliance



Altera (II)

- **Procesadores empotrados de Altera y empresas asociadas (2 de 2).**



Accelerate Your Time to Market with the Altera Embedded Alliance

Altera has partnered with industry-leading embedded suppliers to provide a broad range of processors, tools, IP cores, and operating system support to accelerate your time to market. Our highly integrated, customizable solutions target FPGAs, HardCopy® ASICs, and SoC FPGAs.

Altera Embedded Alliance

ARM® Cortex™-A9 MPCore™ Processor

Altera's ARM-based SoC FPGAs let you reduce system power, system cost, and board space by integrating the performance of a dual-core ARM Cortex-A9 MPCore processor with the flexibility of our power-optimized 28-nm low-power FPGA fabric.

Nios II Processor

Altera's Nios® II processor is the world's most versatile processor. This soft processor delivers unprecedented flexibility for your cost-sensitive, real-time, safety-critical (DO-254), ASIC-optimized, and applications processing needs. According to Gartner Research, the Nios II processor is the most widely used soft processor in the FPGA industry.

Cortex-M1 Processor

The ARM Cortex-M1 processor, combined with Altera's cost- and power-optimized FPGA devices and suite of embedded IP cores, delivers an ARM-based microcontroller-class solution with an exact-fit peripheral set and familiar ARM tool suite.

E6xxC Processor

The Intel® Atom processor E6x5C series provides a customizable Intel Atom processor by combining the E6x5C processor with an Altera® FPGA on a multichip package. Altera's best-in-class SoC system integration tool, Qsys, allows you to easily adapt to changing interface standards and peripherals, and is an ideal platform for the development of netbooks, mobile internet devices, and automotive infotainment solutions.

MP32 Processor

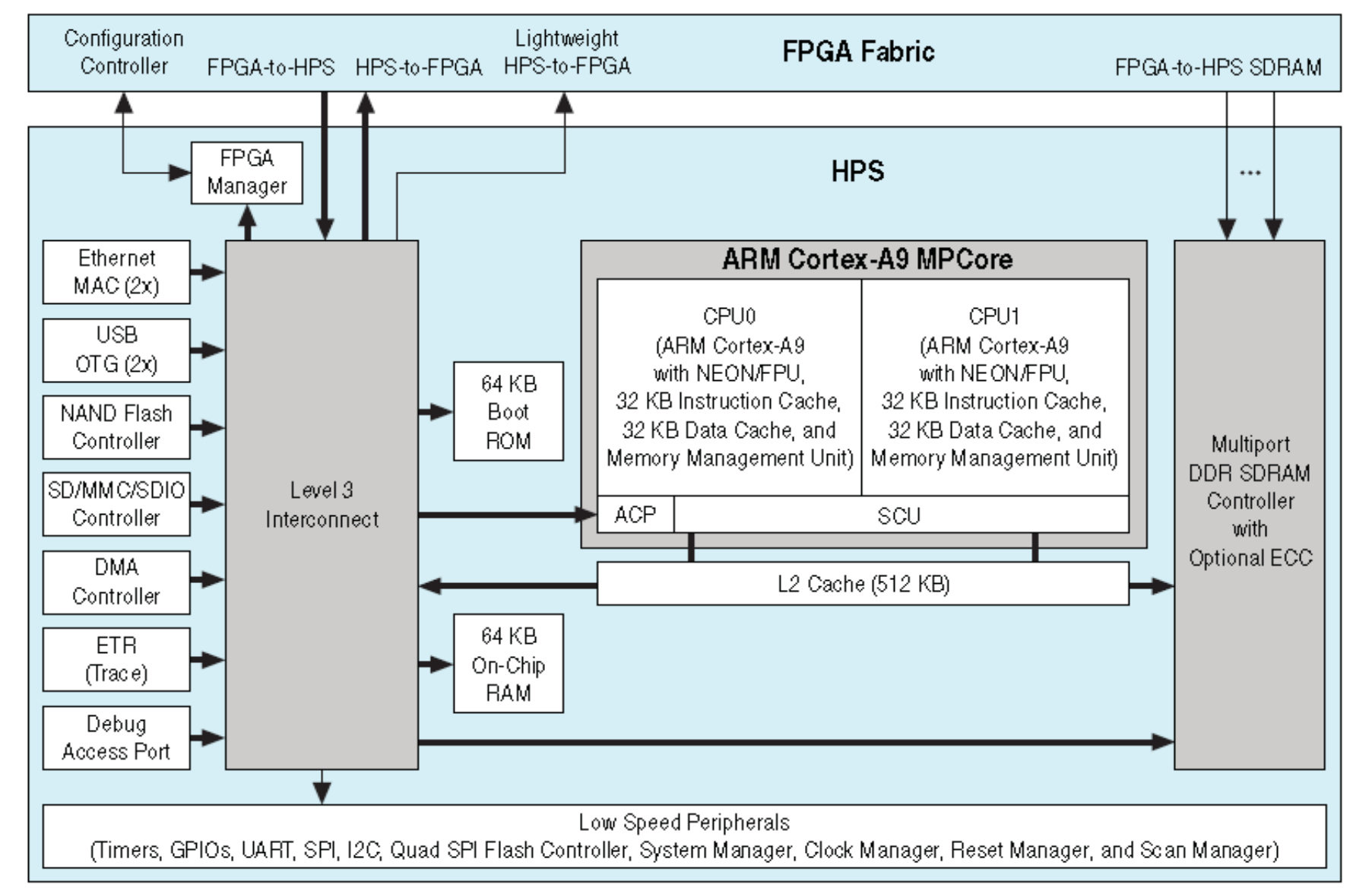
The MP32 processor brings the vast MIPS® ecosystem to Altera's FPGAs, allowing you to reuse software code and IP cores to build customizable MIPS-based solutions for video, digital signal processing, and networking.

V1 ColdFire Processor

The small-footprint V1 ColdFire core is designed for entry-level 32-bit low-power consumer and industrial applications. Combined with Altera's cost- and power-optimized FPGA devices and suite of embedded IP cores, the V1 ColdFire core delivers a custom microcontroller solution that is both extensible and obsolescence proof.

Altera (III)

- **Familia de FPGAs Arria V SoC de Altera.**



Altera (IV)

- **Procesador ARM Cortex-A9. Disponible en las FPGAs Arria V SoC y Cyclone V SoC de Altera.**

