

Sistemas Embebidos: Teoría, Implementación y
Metodologías de Diseño

UNIVERSIDAD NACIONAL DE COLOMBIA

Carlos Iván Camargo Bareño

28 de octubre de 2011

INFORME FINAL

Sistemas Embebidos: Teoría, Implementación y Metodologías de Diseño

AUTHOR: C. Camargo

E-MAIL: cicamargoba@unal.edu.co

Copyright ©2004, 2005 Universidad Nacional de Colombia

<http://www.unal.edu.co>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, version 1.2, with no invariant sections, no front-cover texts, and no back-cover texts. A copy of the license is included in the end.

This document is distributed in the hope that it will be useful, but without any warranty; without even the implied warranty of merchantability or fitness for a particular purpose.

Published by the Universidad Nacional de Colombia

Índice general

1. Sistema en un Chip (SoC)	3
1.1. Arquitectura	3
2010/8/12 Oscar Javier Fajardo Ruiz ¡ojfajardor@unal.edu.co! Nuestra idea es el ecualizador	

Capítulo 1

Sistema en un Chip (SoC)

En esta sección estudiaremos la arquitectura básica de un SoC moderno, componentes, funcionamiento, programación y operación. Como se mencionó anteriormente, la tendencia actual de la industria de los semiconductores es integrar en un solo dispositivo las funcionalidades necesarias para la implementación de dispositivos digitales modernos. Esto es posible gracias a los grandes avances en las técnicas de fabricación de circuitos integrados y a la demanda de nuevas características exigidas por los fabricantes de dispositivos digitales de consumo masivo como teléfonos celulares, PDAs, consolas de juegos y reproductores multimedia. Para utilizar estos avances tecnológicos es necesario conocer su arquitectura, principio de funcionamiento, programación e implementación, para esto, se estudiarán dos proyectos abiertos que implementan un SoC en un PLD utilizando lenguaje de descripción de hardware y herramientas GNU; proporcionan el código fuente, lo que permite un estudio profundo de su arquitectura. El proyecto *Plasma* [1] y el proyecto Mico32[2].

1.1. Arquitectura

Un SoC, integra un conjunto de periféricos, memorias y una o varias unidades de procesamiento (CPUs) en un solo chip, lo cual facilita el desarrollo de aplicaciones. Comercialmente encontramos una gran variedad de configuraciones CPU - periféricos, dependiendo de la aplicación; dentro de los más comunes se encuentran: controladores de memorias externas (NOR, NAND, SDRAM, DDR), puertos de comunicación (I2C, SPI), puerto de depuración (UART), timers, reloj de tiempo real, codecs de audio, controladores de LCD, controladores de red, controlador de puerto USB host, controlador para sensores de imágenes, etc. La figura 1.1 muestra una arquitectura de un SoC sencillo con los componentes necesarios para implementar tareas simples.

Unidad de Procesamiento Central (CPU)

La unidad de procesamiento central (CPU), como su nombre lo indica, está encargada de centralizar las tareas del sistema coordinando las acciones de los periféricos. Puede ser vista como una máquina de estados programable que controla un camino de datos compuesto por bloques aritméticos, lógicos y un banco de registros. Cada CPU es capaz de realizar una serie de operaciones sobre variables almacenadas en el banco de registros, estas operaciones reciben el nombre de *conjunto de instrucciones*. El programador utiliza estas instrucciones para hacer que la CPU realice una función específica, indicándole paso a paso

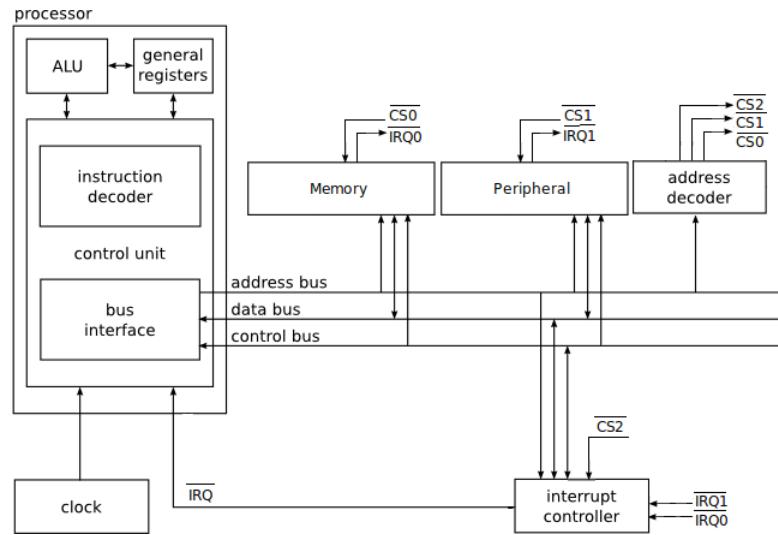


Figura 1.1: Arquitectura mínima de un SoC

donde debe leer la información, como procesarla y como entregar el resultado. A Esta función se le conoce con el nombre de programa o tarea Software.

Periféricos

Los periféricos proporcionan la comunicación con el exterior, permiten el ingreso, almacenamiento y procesamiento de la información

Bibliografía

- [1] S. Rhoads. Plasma - most MIPS I(TM) opcodes. URL: <http://opencores.org/project,plasma>, 2008.
- [2] Lattice Semiconductor Corporation. LatticeMico32 Open, Free 32-Bit Soft Processor. URL: <http://www.latticesemi.com/products/intellectualproperty/ipcores/mico32/index.cfm>, 2008.