Diseño Automático de Sistemas Fiables

System On Chip (SoC)

UNIVERSIDAD NEBRIJA

Objetivos Clase

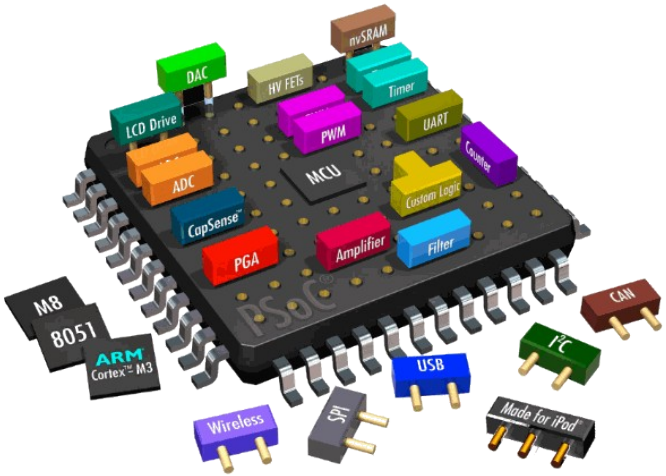
- Conocer lo que es un System on Chip (SoC).
- Entender las diferencias con una CPU y un MCU.
- Coprocesadores y su rol actual en los productos de consumo.
- Entender el por qué puede ser necesario usar una CPU en un SoC junto a una FPGA
- Entender que tareas deben ir a una CPU (software) y cuales a una FPGA (hardware)

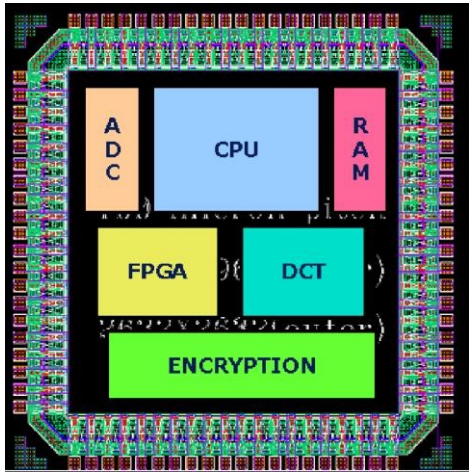
Contenidos

1. Introducción a los SoCs
2. Comparación entre una CPU, un CPU y un SoC
3. Ejemplos de diferentes diseños digitales
4. Ventajas de un SoC
5. Desventajas de un SoC
6. Ejemplos de SoCs con FPGAs
7. Codiseño ¿Cómo, Cuando?
8. ¿Cómo elegir una CPU para nuestro diseño?

Introducción a los SoCs

- Un System On Chip (SoC) consiste en integrar en un mismo chip diferentes elementos con propósitos diferentes (CPUs, GPUs, procesado de señal (DSPs), módems de comunicación, aceleradores de redes neuronales, etc…).





Introducción a los SoCs

Problemas presentes en los diseños digitales actuales:

- Procesado muy especifico como puede ser análisis de señales, ej: medidor de pulso de una smartband.
- Limites de consumo muy bajo: un portátil o un móvil.
- Necesidades de flexibilidad no disponibles en una FPGA

CPU vs MCU vs SoC

Procesador (CPU)

- Es una unidad de procesado que requiere de memoria y periféricos externos. El ejemplo más claro es el procesador de un ordenador de escritorio. Tanto la memoria como las interfaces (USB, PCIe, ethernet, etc..) son externas al chip, por eso es necesario una placa base que tiene dichos elementos.

Microcontrolador (MCU)

- Es una unidad de procesado que tiene todos los periféricos necesarios para su funcionamiento. Habitualmente solo es necesario proporcionar una fuente de alimentación constante y en algunos casos un oscilador. Un ejemplo de MCU serían los micros que utilizáis en sistemas en tiempo real.

System on Chip (SoC)

- En un SoC existen diferentes unidades de procesado orientados a propósitos diferentes. No requiere de tantos elementos externos como una CPU. Un ejemplo sería el chip que llevan los móviles.

Ejemplos: CPU vs MCU vs SoC

Procesador (CPU)

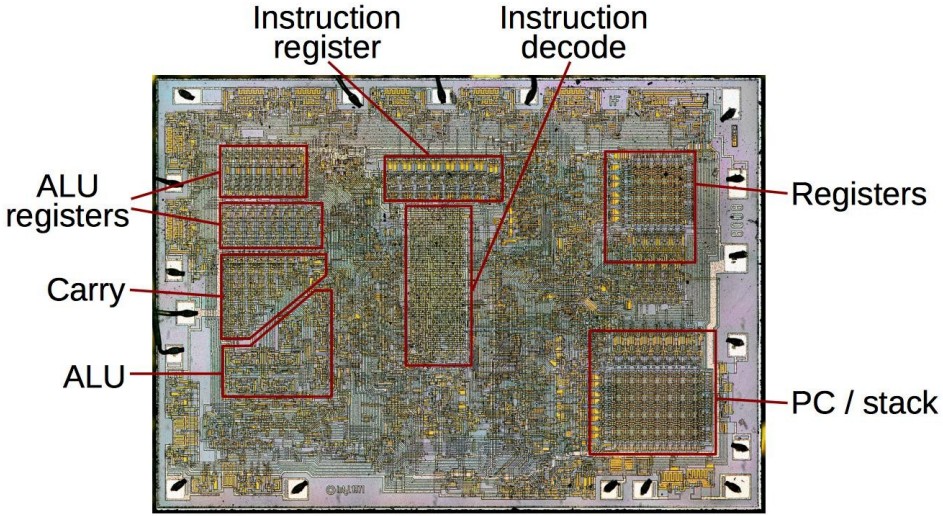
Microcontrolador (MCU)

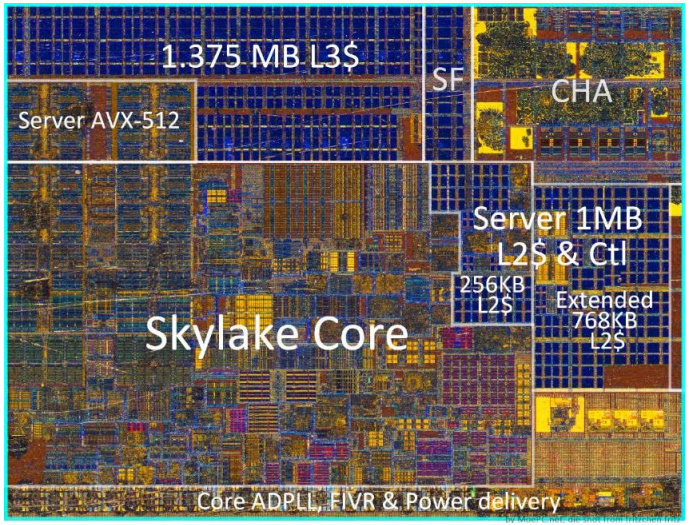
System on Chip (SoC)

Ejemplos: Die CPU

Intel 8080 (1974) (6000 nm)

Procesador completo, solo tenia un core de 8 bits.





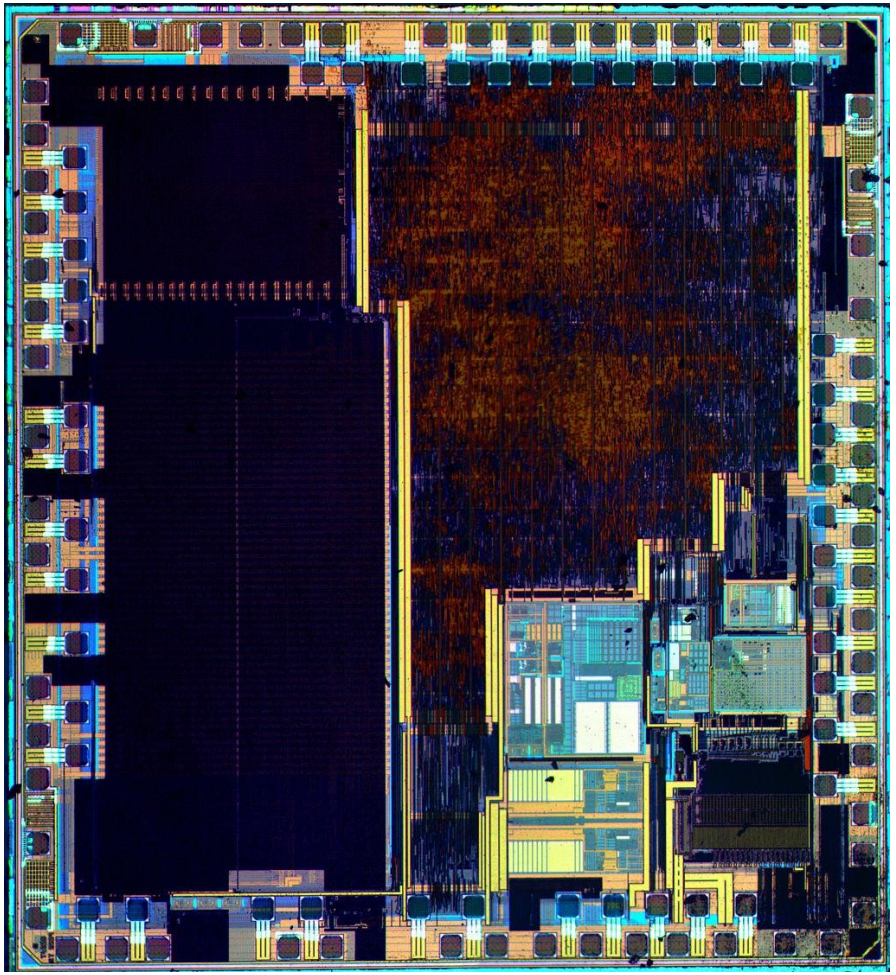
Intel Skylake (2017) (14 nm)

Un core de 64 bit de los 6 – 28 que puede tener un procesador completo

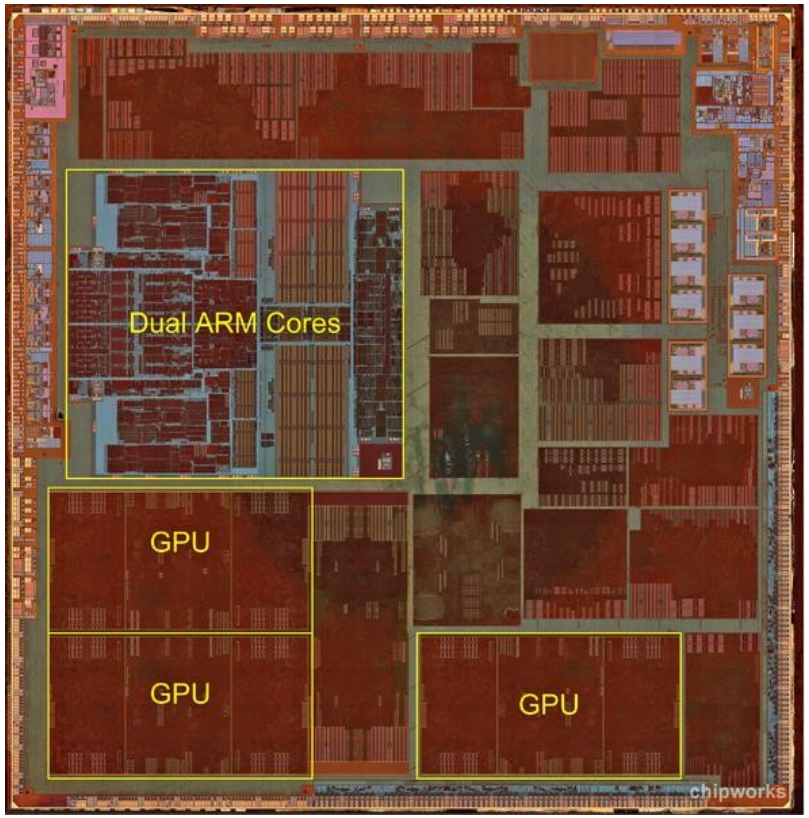
Ejemplos: Die MCU

STM32F100C4T6B (2009) (90 nm)

Se trata de un ARM Cortex M3 con 16kB de flash y 4kB RAM. Un core de 32 bits.



Ejemplos: Die SoC



Apple A6 (2012) (32 nm)

Se trata de un ARM Cortex A15 custom de Apple y 2 GPUs.1Gb de RAM

Utilizado en el iPhone 5

Ventajas de un SoC

- Alto rendimiento
- Mayor eficiencia
- Diseño más compacto
- Mayor fiabilidad, no requiere rutado ni tener en cuenta componentes externos
- Menor coste de implementación
- Menor tiempo de diseño al integrar sistemas

Desventajas de un SoC

- Coste muy elevado de diseño propio
- Puede tener una serie de características que no necesitamos
- No adaptado a nuestra solución
- Dependencia de las herramientas de diseño, es necesario trabajar a un nivel más alto

Ejemplos de SoC con FPGAs

Los SoC más habituales en FPGAs consisten en incluir una CPU, normalmente un ARM.

Altera: Altera SoC (dual/single ARM Cortex A9)

Xilinx: Zynq (dual ARM Cortex A9)

Microchip: SmartFusion2 (ARM Cortex M3) y PolarFire (RISC V)

Hardware vs Software

Cualquier algoritmo se puede implementar en un circuito hardware o en una aplicación de software.

Nosotros como ingenieros debemos decidir cuando elegir una implementación u otra.

Software: Flexibilidad, Facilidad de desarrollo, Trabajo a alto nivel, velocidad en aplicaciones generales (funciones que disponga el procesador).

Hardware: Velocidad en aplicaciones especificas, paralelizar, control total sobre el funcionamiento.

Codiseño HW/SW

Combinar lo ~~peor~~ mejor de ambos mundos.

- Del Hardware se aprovecha la posibilidad de soluciones especificas y la paralelización inherente.
- Del Software la capacidad de realizar tareas de control de forma sencilla y realizar operaciones generales a muy alta velocidad.

Codiseño HW/SW

Consiste en diseñar un sistema en el que cooperen elementos HW y SW en armonía.

Podemos hablar de elementos fijos (HW) y flexibles (SW)

Muy importante la planificación y el diseño del sistema.

Codiseño HW/SW

Son dos tipos de diseño muy diferentes que tienen características distintas.

	Hardware	Software
Paradigma	División especial	Division temporal
Coste de recursos	Área (LUTs)	Tiempo
Flexibilidad	Baja	Alta
Paralelismo	Alto	Bajo
Reutilización	Complejo	Sencillo

Codiseño HW/SW: ¿Cuándo?

Criterios generales:

paralelizable -> FPGA **secuencial** -> CPU

Otros motivos por los que usar una CPU.

Poco impacto en el rendimiento final ej: Comunicaciones externas.

Tarea con una complejidad elevada ej: Reprogramación de la FPGA durante su uso.

Necesidad de actualizar el funcionamiento ej: El control del sistema completo.

Codiseño HW/SW: Características CPU

Es importante elegir correctamente el procesador que vamos a usar, características a tener en cuenta:

- Facilidad de uso
- Disponibilidad de debugger
- Sencillez de añadir periféricos
- Rendimiento por Hz
- Área (en el caso de los procesadores hard no se aplica)
- Frecuencia de funcionamiento
- Licencia

Codiseño HW/SW: ¿Cómo?

Hay dos opciones usar procesadores *hard* (SoC) o usar procesadores *soft* (implementados en la propia FPGA).

- Softcores: El propio fabricante nos proporciona herramientas, en el caso de Xilinx, Microblaze. Podemos usar también diseños abiertos o propietarios de terceros, ej: Cortex M0 de ARM, LEON3

Resumen

- Un SoC consiste en integrar más de un procesador con finalidades diferentes en el mismo chip.
- El uso de SoC está muy extendido tanto en las FPGAs como en el resto de sistemas electrónicos actuales debido a las ventajas en tamaño, eficiencia y fiabilidad.
- El codiseño consiste en utilizar circuitos de HW y aplicaciones de SW para lograr un compromiso entre flexibilidad y eficiencia.