

Sistemas Empotrados

Hipólito Guzmán Miranda
Departamento de Ingeniería Electrónica
Universidad de Sevilla
hguzman@us.es

Objetivos de aprendizaje

- Revisar las posibilidades existentes para diseño de sistemas embebidos/empotrados
- Encajar los dispositivos FPGA en dichas posibilidades
- Conocer qué es una FPGA
- Motivar el aprendizaje de diseño digital usando FPGAs

Contenido

- Definición
- Soluciones para sistemas embebidos
- ¿Qué es una FPGA y por qué aprender a usarlas?
- Niveles de abstracción
- Flujo de diseño
- Programa vs diseño

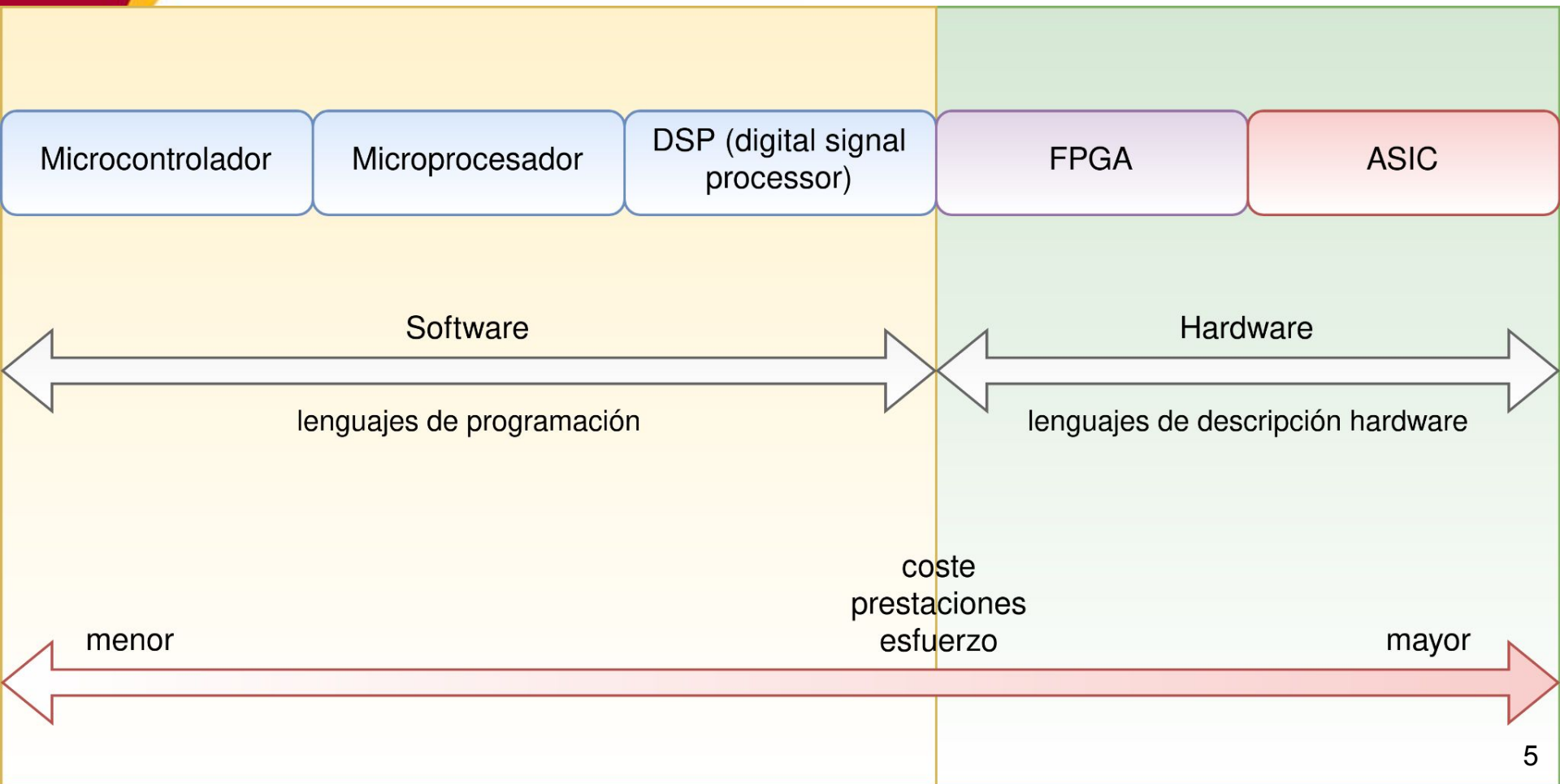
Definición

“A combination of computer hardware and software, and perhaps additional mechanical or other parts, designed to perform a dedicated function. In some cases, embedded systems are part of a larger system or product, as in the case of an antilock braking system in a car.”

– Michael Barr, [“Embedded Systems Glossary”](#)

También podemos pensar en un sistema electrónico que implementa la inteligencia (toma de decisiones) dentro de otro sistema mayor. Normalmente con un énfasis en tener un tamaño limitado, es decir, usamos un sistema empotrado (o embebido) donde un PC de sobremesa es demasiado grande. También suele ocurrir que, aunque el sistema electrónico pueda ser de propósito general, se configura/programa para resolver un problema concreto en un futuro despliegue.

Múltiples opciones para sistemas 'embedded'



De los uC a los ASIC

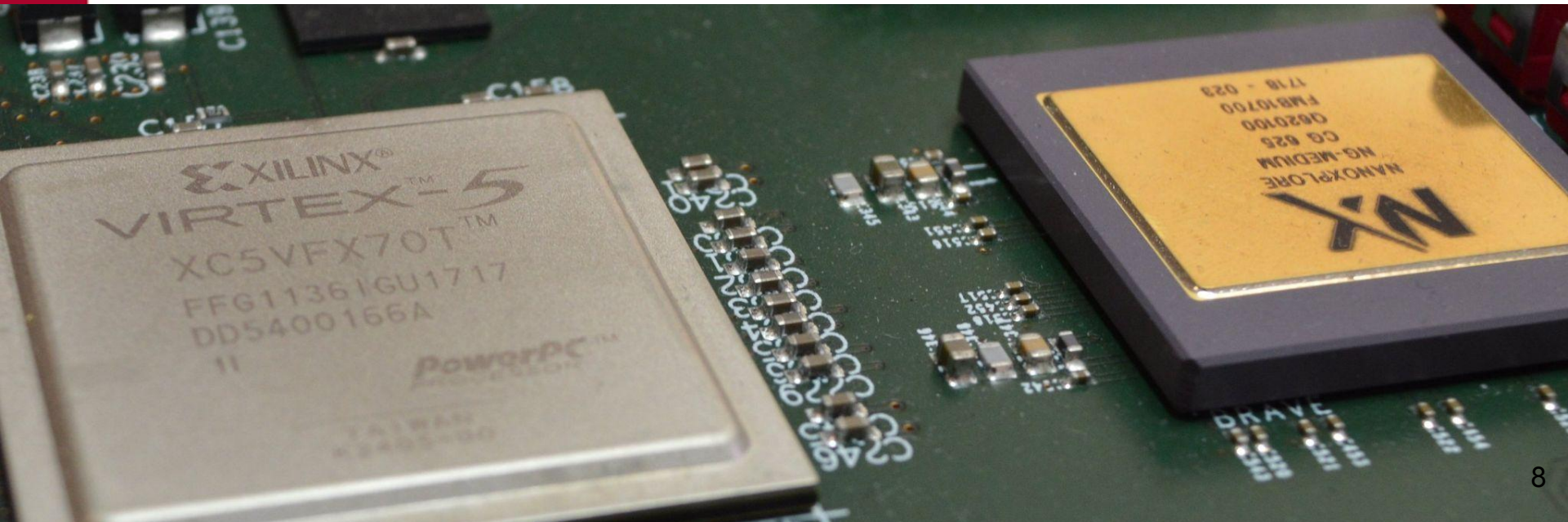
- **Microcontrolador**
 - Menores prestaciones (e.g., 8/16/32 bit), normalmente más periféricos integrados
- **Microprocesador**
 - Mayores prestaciones (e.g., 32/64 bit)
- **DSP**
 - Digital Signal Processor: Microprocesador con módulos específicos para procesamiento de señal
- **FPGA**
 - Field Programmable Gate Array: circuito digital que puede configurarse para implementar la arquitectura que queramos
- **ASIC**
 - Application-Specific Integrated Circuit: circuito integrado que se diseña y fabrica para resolver un problema específico

De los uC a los ASIC

- Microcontrolador
 - Microprocesador
 - DSP
 - FPGA
 - ASIC
- Ya sabéis de microprocesadores
 - Profundizaremos un poco también
 - Para FPGAs y ASIC:
 - Lenguajes de descripción hardware
 - Consideraciones físicas en el caso de ASIC
 - No lo veremos en la asignatura

¿Qué es una FPGA?

- FPGA = Field Programmable Gate Array
- Es un circuito integrado cuya arquitectura interna puede ser configurada por el cliente (diseñador)
- Normalmente se configura utilizando un Lenguaje de Descripción Hardware (HDL), y herramientas asociadas para la síntesis e implementación del diseño



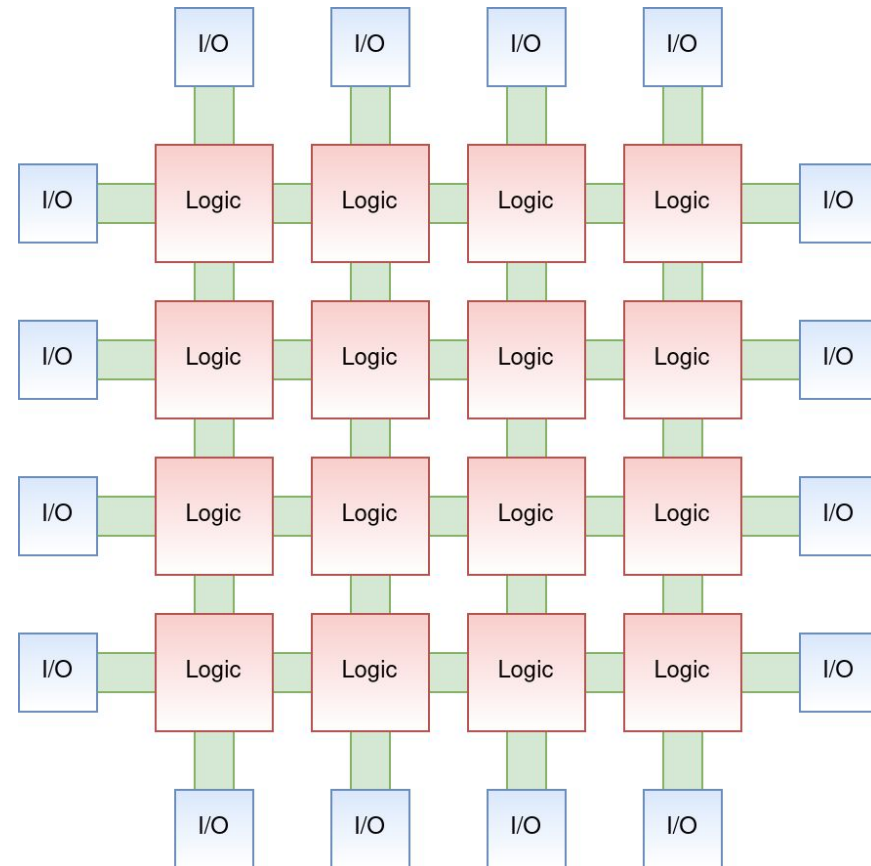
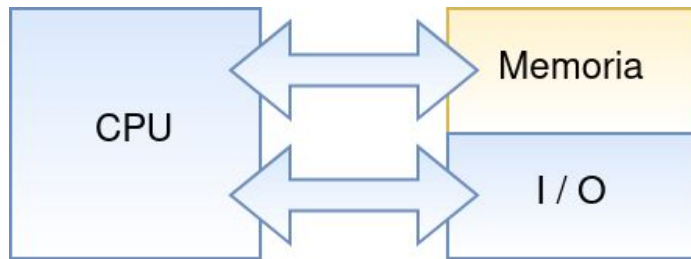
Programabilidad

FPGA vs microprocesador

“Pero... un microprocesador también es programable, no?”

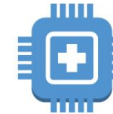
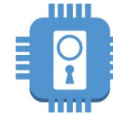
La **separación entre CPU y memoria** es lo que hace que el microprocesador sea programable: la CPU siempre es la misma, y el ingeniero puede cambiar el **programa** almacenado en la memoria

Microprocesador vs FPGA

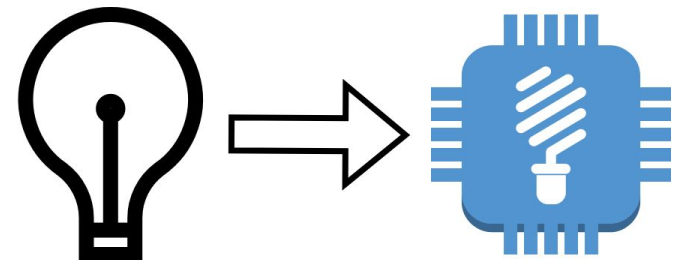


¿Por qué aprender a usar FPGAs?

Prototipado rápido de circuitos digitales a medida



- No es necesario fabricar un ASIC
 - Menor esfuerzo
 - Reducción de costes para tiradas pequeñas
- Reducción del 'Time to Market'
- I+D+i
- Confidencialidad



¿Por qué aprender a usar FPGAs?

Rendimiento / prestaciones

- Procesamiento en paralelo

Aplicaciones en múltiples industrias:

- Astronomía
- Física de partículas
- Aeronáutica
- Espacio
- Banca / inversiones

Aplicaciones

- Astronomía
 - BEE2 ([Berkeley Emulation Engine 2](#))
 - Usado en el [Allen Telescope Array](#)
 - PetaOp/second
- Física de partículas
 - [Detector del CMS](#) del LHC en CERN
- Aeronáutica
 - Sistemas de [RADAR](#) y comunicaciones
- Espacio
 - [Solar Orbiter](#), Misiones [OSIRIS](#), Misión [Rosetta](#) y muchas otras...
- Banca / inversiones
 - [High Frequency Trading](#)



¿Qué es una FPGA y por qué aprender a usarlas?

Oportunidad laboral

FPGA jobs

Sort by: **relevance** - date

3,008 jobs ?

FPGA Engineer

Quantum Dimension
Huntington Beach, CA

We are seeking a talented FPGA design engineer to join our design & development team at our corporate office in Huntington Beach, CA. This FPGA position...

Posted 30+ days ago · More...

ASIC-FPGA Design Engineer

Intellibee
Phoenix, AZ

Description: Analyze, design, simulate, and implement algorithms in hardware descriptor languages, HDL (VHDL, Verilog, System Verilog), based on customer...

Posted 30+ days ago · More...

PWB Designer - Staff

Qualcomm
San Diego, CA

Company: Qualcomm Technologies, Inc. Job Area: Engineering Services Group, Engineering Services Group > PWB Design General Summary: The successful...

Posted 6 days ago · More...

Senior FPGA Design Engineer

302 VHDL Jobs in Spain (21 new)



FPGA ENGINEER

onhunters

Madrid, Community of Madrid, Spain

☒ Be an early applicant

2 weeks ago



Power Electronics Engineer

Applus+ IDIADA

la Bisbal del Penedès, Catalonia, Spain

☒ Be an early applicant

6 days ago



FPGA's Design Engineer

Airbus

Madrid, Community of Madrid, Spain

☒ Be an early applicant

2 hours ago



FPGA Engineer

Gradient

Vigo, Galicia, Spain

3 weeks ago



Electronic Hardware Designer

SET Europa

San Sebastián, Basque Country, Spain

☒ Be an early applicant

2 months ago



Fpga Engineer Luxquanta, Spain

LuxQuanta

Madrid, Community of Madrid, Spain

☒ Be an early applicant

2 weeks ago

<- indeed.com

(FPGA, United States)

linkedin.com ->

(FPGA, Spain)

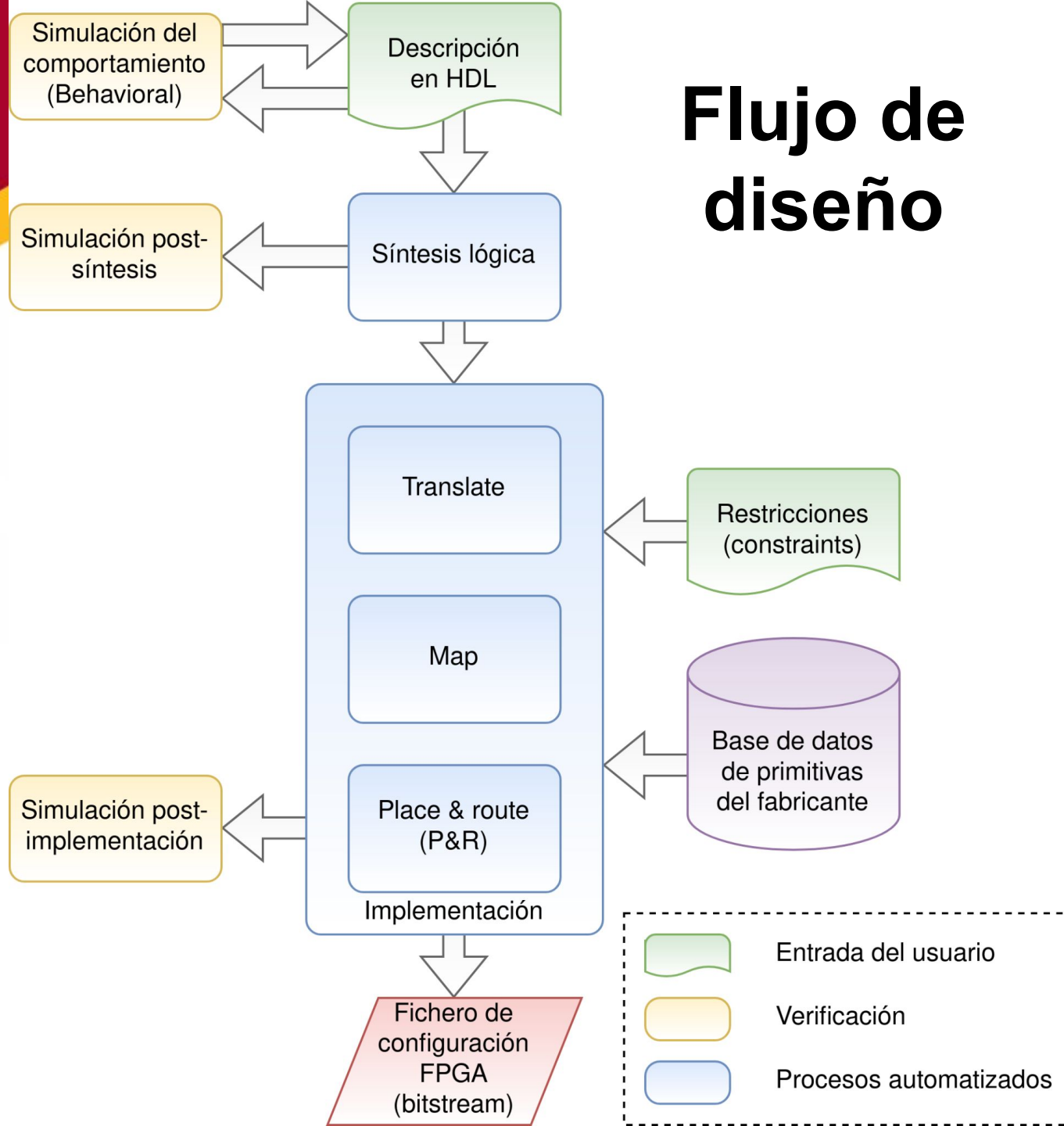
De abstracto a específico

Nivel	Descripción
System Level	Visión general (diagrama de bloques)
High Level	Implementación funcional en lenguajes de programación de alto nivel (C, C++, SystemC, Matlab, Python, ...)
Behavioral Level	Descripción 'cycle-accurate' del circuito en lenguaje de descripción hardware (Verilog, VHDL, ...)
Register-Transfer Level (RTL)	Lista de registros y funciones lógicas. Típicamente una netlist, pero también puede ser un código en HDL (Verilog, VHDL, ...)
Logic Gate Level	Netlist de biestables y puertas lógicas básicas (and, or, not, ...)
Physical Gate Level	Netlist de puertas o primitivas disponibles en la arquitectura objetivo (ya sea FPGA o ASIC)
Switch Level	Netlist de transistores individuales

Uso típico en diseño HW

Nivel	Uso típico
System Level	Planteamiento a nivel general del sistema y división en bloques funcionales
High Level	Creación de un modelo de referencia (golden model) para verificación. En algunos casos, síntesis de alto nivel utilizando herramientas tipo Vitis HLS
Behavioral Level	Diseño del hardware en sí utilizando VHDL o Verilog. Simulación funcional (behavioral simulation)
Register-Transfer Level (RTL)	Diseño del hardware en sí utilizando VHDL o Verilog. Simulación funcional (behavioral simulation)
Logic Gate Level	Diseño mapeado a la tecnología FPGA concreta. Simulación post-síntesis
Physical Gate Level	Diseño rutado. Simulación post-implementación
Switch Level	Configuración del dispositivo FPGA

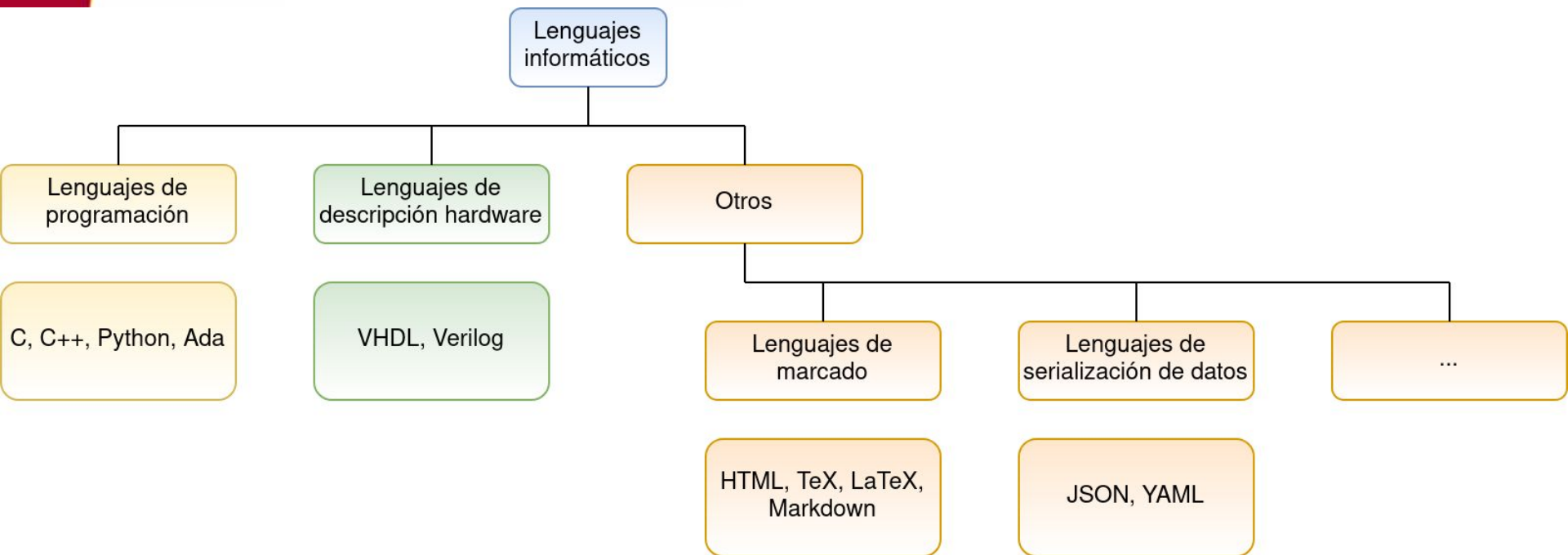
Flujo de diseño



Síntesis e Implementación

- **Síntesis:** convierte de Behavioral a RTL
- **Translate:** parte de todo el RTL y las 'constraints' (pines, timing, etc) y los une en un único conjunto
- **Map:** describe el diseño utilizando únicamente las primitivas que tenemos disponibles en la FPGA (Look-up-Tables, Flip-flops, Block RAMs, etc)
- **Place and Route:** decide qué recursos exactos va a utilizar dentro del silicio y configura las conexiones entre estos

Tipos de lenguajes informáticos



Programación vs Descripción Hardware

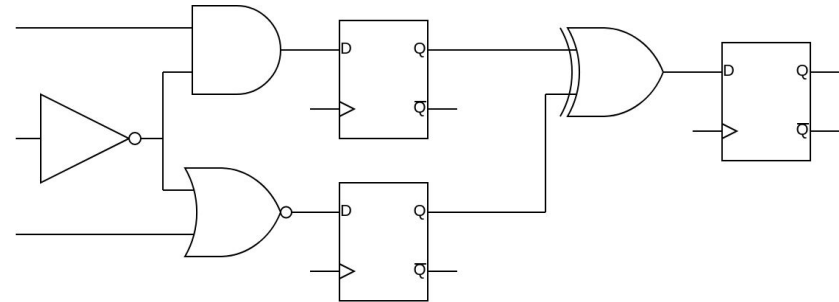
Lenguaje de programación

- Describe una secuencia de instrucciones
- Secuencialidad
 - El orden es importante!

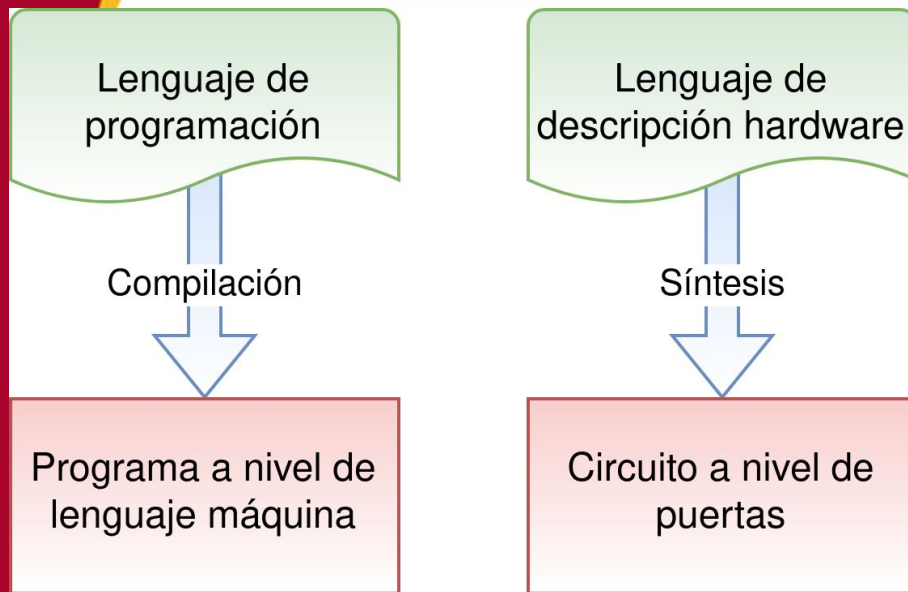
```
load 3, R0
load 4, R1
mul R0, 3
add R1, 3
mul R0, R1
jnz R2 ...
```

Lenguaje de descripción hardware

- Describe un circuito
- Concurrencia
 - Todo ocurre a la vez!



HDL: Programa vs diseño



- La sintaxis es muy similar
 - VHDL se parece a Ada
 - Verilog se parece a C
- En VHDL no se programa, se **describe**
- Hay que pensar siempre que la descripción se corresponde con **circuitos** funcionando en paralelo

Conclusiones

- Los **sistemas empotrados** permiten **implementar la inteligencia** (toma de decisiones) de los 'ingenios' que diseñemos como futuros ingenieros
- Existe toda una **panorámica de soluciones** a las que podemos recurrir
 - Un ingeniero utilizará la más **sencilla y económica** que resuelva su problema
- Las **FPGAs** son **dispositivos programables** con **múltiples aplicaciones** en ingeniería
- VHDL es un **lenguaje de descripción hardware** que puede utilizarse para trabajar con FPGAs

Resultados de aprendizaje

- Conocer, de manera descriptiva, qué son los sistemas empotrados o embebidos
- Conocer qué opciones tenemos para implementar sistemas empotrados
- Conocer, de manera descriptiva, qué es una FPGA, en qué se diferencia de un microprocesador y qué posibles aplicaciones puede tener
- Comprender las diferencias existentes entre un lenguaje de programación y un lenguaje de descripción hardware
- Conocer, de manera descriptiva, en qué consisten los procesos de síntesis e implementación