Sistemas Electrónicos Digitales

Tema #4

Diseño mediante Lenguajes de Descripción Hardware

Parte 4.1

Índice

HDL: definición

- Un **lenguaje de descripción de hardware** (HDL, *Hardware Description Language*) es un lenguaje de programación especializado que se utiliza para definir la estructura, diseño y operación de circuitos electrónicos, y más comúnmente, de circuitos electrónicos digitales

- El acrónimo VHDL proviene de la combinación de dos acrónimos: VHSIC (Very High Speed Integrated Circuit) y HDL (Hardware Description Language)

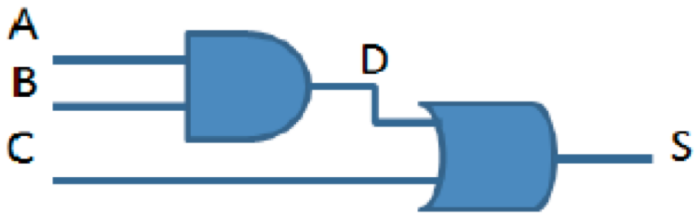
- Hacen posible una descripción formal de un circuito electrónico, y posibilitan su análisis automático y su simulación.

- ¿Dónde se usan?
  - FPGAs, CPLDs

Más sobre VHDL

- Desarrollado por el departamento de defensa de los Estados Unidos a inicios de los años 80 para simular circuitos eléctricos digitales.
- Posteriormente se desarrollaron herramientas de síntesis e implementación
- VHDL **no es un lenguaje de programación**
- VHDL **es un lenguaje de descripción de hardware**
  - Pensar en puertas y biestables, no en variables ni funciones.
  - Evitar bucles combinacionales y relojes condicionados.
  - Saber qué parte del circuito es combinacional y cuál secuencial.

¿Por qué VHDL?



Prog 1
D = A and B;
S = D or C;

Prog 2
S = D or C;
D = A and B

Simulación Hardware		
t = 0ns	t = 5ns	t = 10ns
A = 0	A = 0	A = 1
B = 0	B = 1	B = 1
C = 0	C = 0	C = 0

¿S?

Un lenguaje de descripción HW nos debe dar el mismo resultado en simulación para los dos programas del ejemplo

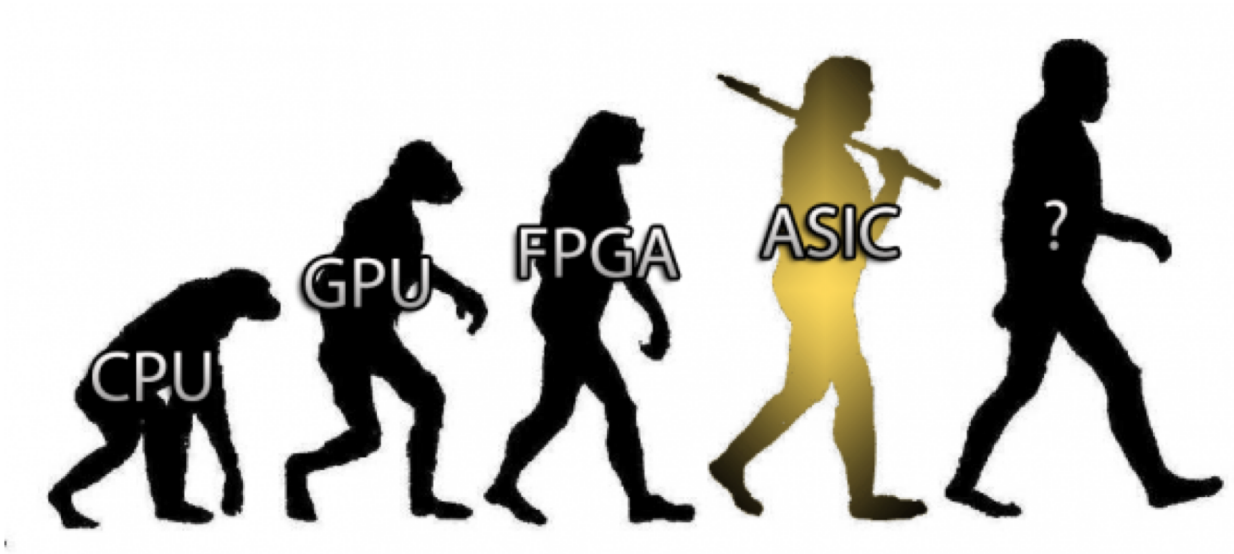
Evolución de la tecnología

- En 1965 Gordon E. Moore, cofundador de Intel enunció la que se conoce como “Ley de Moore”.
  - **Ley de Moore: el nº de transistores por chip se duplica cada dos años.**
- Se trata de una ley empírica, pero no ha dejado de cumplirse desde entonces.
- Se pueden hacer enunciados similares para:
  - Frecuencia de reloj
  - Capacidad de las memorias.
  - Prestaciones.
  - Tamaño de las programas.

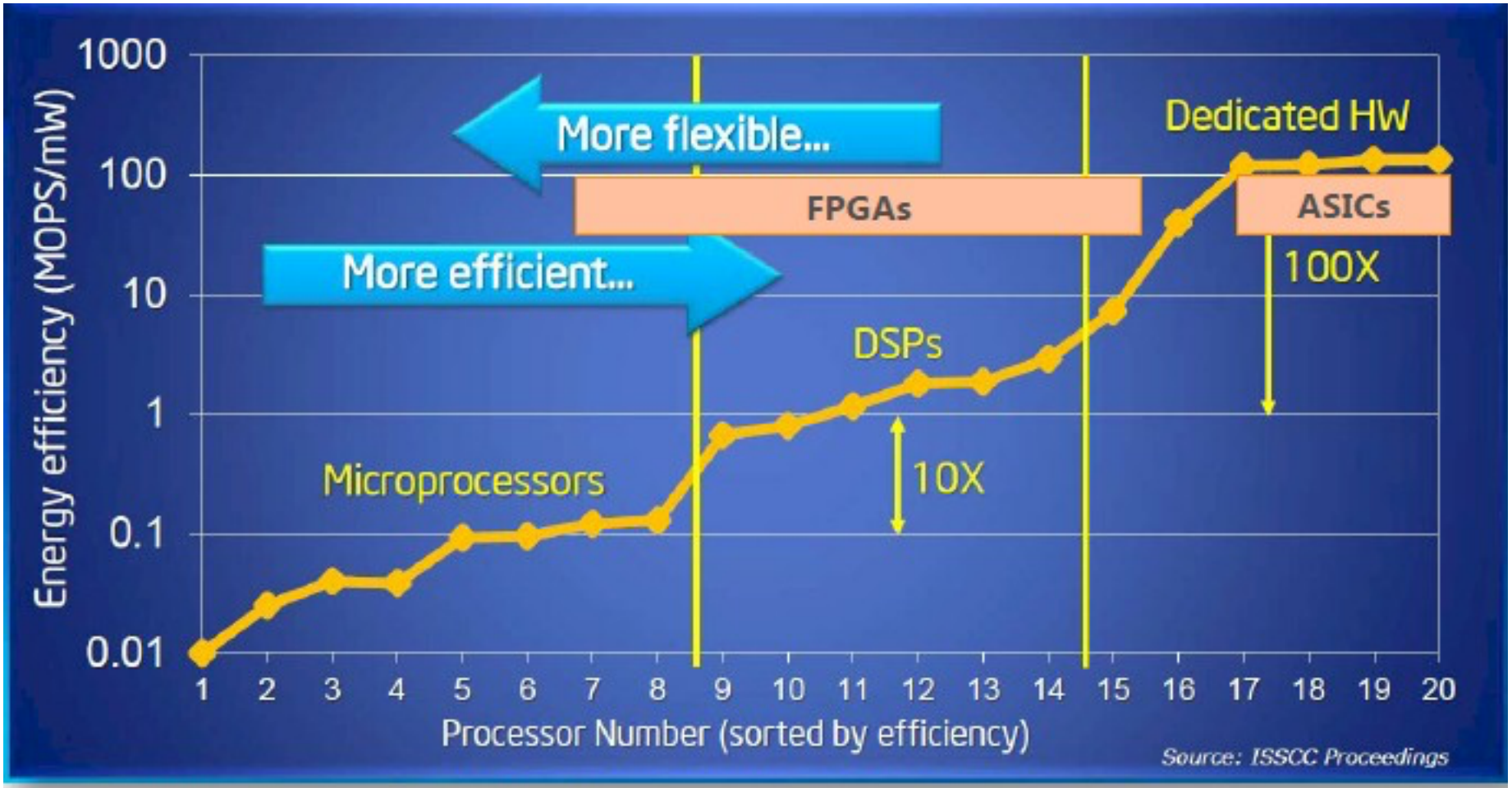
SoC: Sistemas en chip

- **ASIC**: Application-Specific Integrated Circuit (ASIC)
  - IC desarrollado para un uso específico
- En la actualidad es posible integrar sistemas completos en un chip:
  - Microprocesador, DSP.
  - Memoria.
  - Controladores de bus (memoria, PCI, USB, ...)
  - E/S (paralelo, Ethernet, RocketIO...)
  - Sensores y electrónica de acondicionamiento.
  - Etc.
- Es lo que se conoce como **SoC: *System on Chip*.**

SoC: Sistemas en chip



SoC: Sistemas en chip



Source: Bob Broderson, Berkeley Wireless group

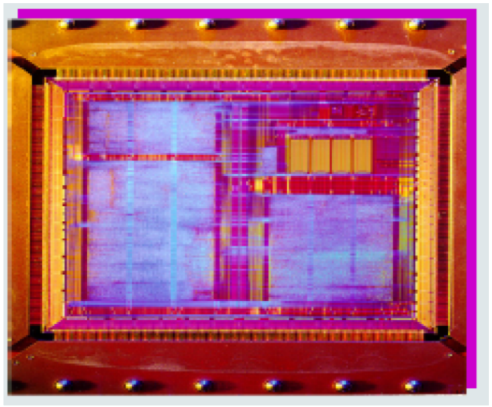
Ejemplo SoC

HW

SW

Microprocesador

Memoria incorporada



ASIC

Circuitos analógicos

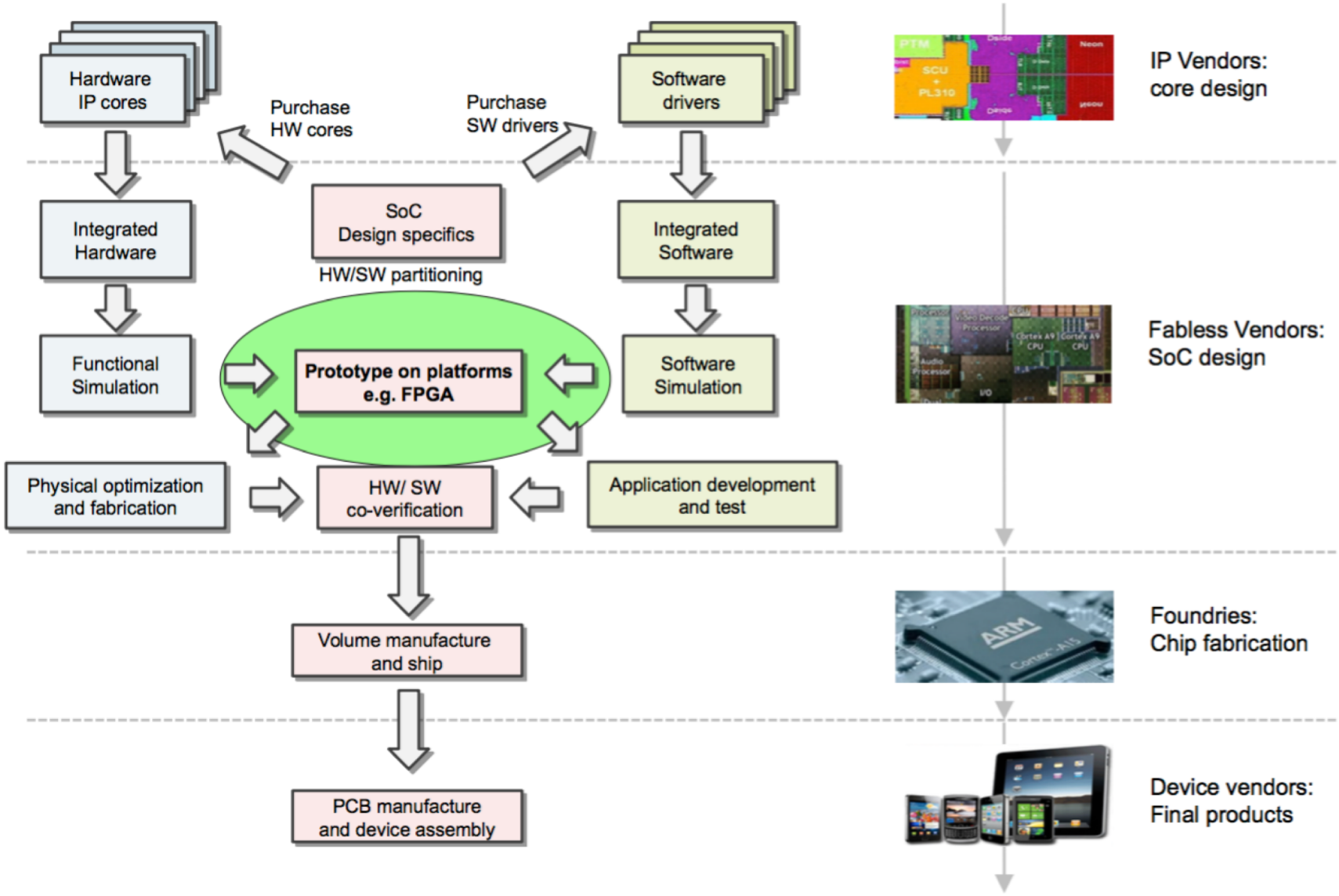
DSP

Sensores

Electrónica de alta velocidad

Interfaz de red

Flujo de diseño SoC



Ventajas del SoC

- Mayor rendimiento
  - Menor ”propagation delay” y “gate delay”
- Mayor eficiencia energética
  - Menor voltaje requerido
- Menor “footprint”: peso y tamaño
- Mayor fiabilidad
  - Encapsulado en un solo chip => menor interferencia externa
- Menor coste?
  - Un solo chip fabricado en serie

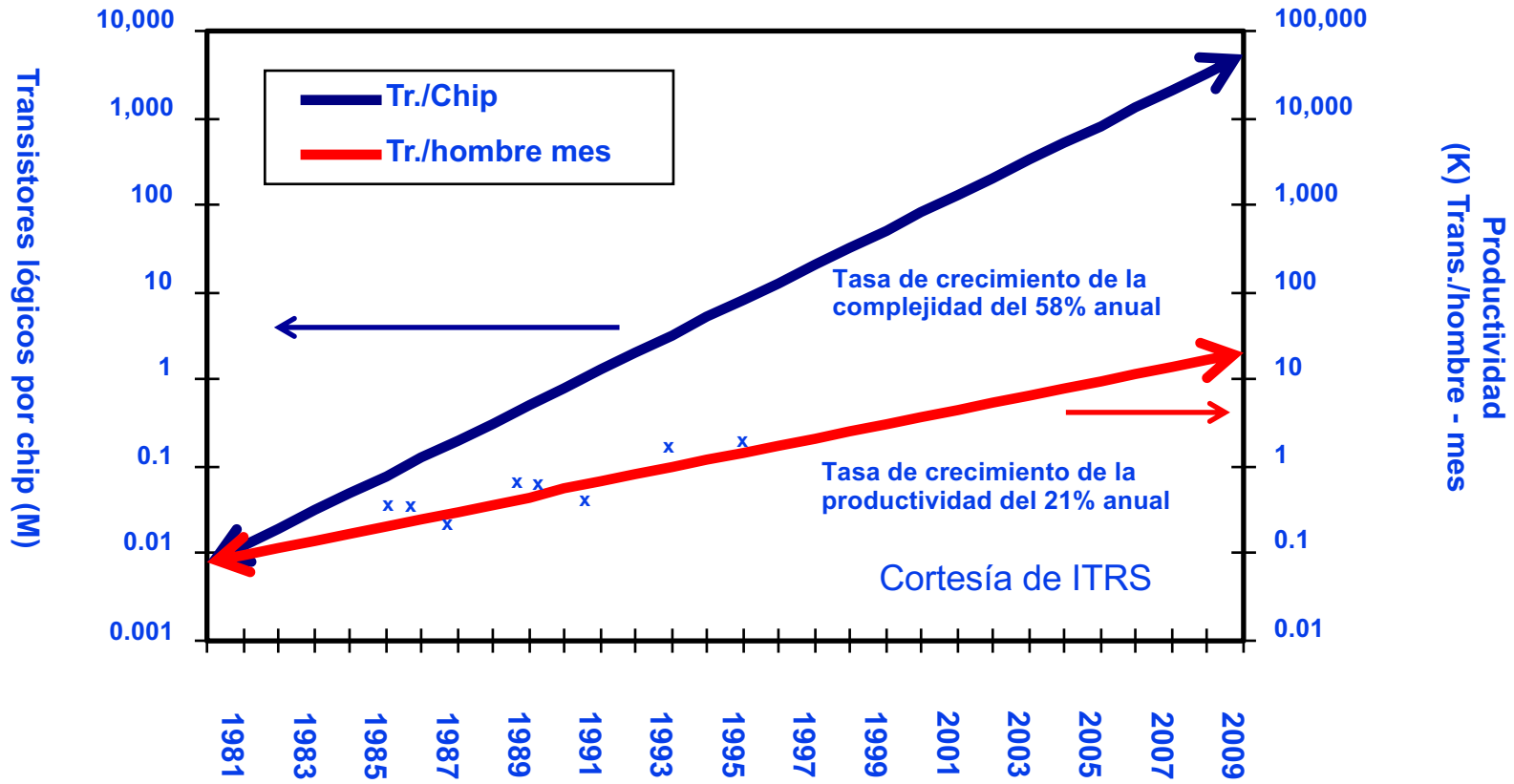
Limitaciones del SoC

- Menor flexibilidad
  - No se pueden modificar sus componetes, i.e. RAM
- Complejidad

Complejidad

- La tecnología actual permite la fabricación de **sistemas** muy **complejos**.
- La tendencia es que la complejidad aumente aún más.
- Desgraciadamente no ocurre lo mismo con la **productividad** de los diseñadores.
- Según aumenta la **complejidad** el **coste** de desarrollo se dispara.

Productividad



Tiempo de desarrollo

- Con la productividad de 1981, un equipo de 1000 personas habría tardado más de 35 años en desarrollar un Pentium5 (42MT).
- En la actualidad la mayoría de los circuitos quedan obsoletos en menos de dos años.
- Si un nuevo desarrollo no sale el primero al mercado es posible que nunca resulte rentable.

Coste de diseño

- El aumento de complejidad también afecta al coste de desarrollo.

  Con los siguientes parámetros:

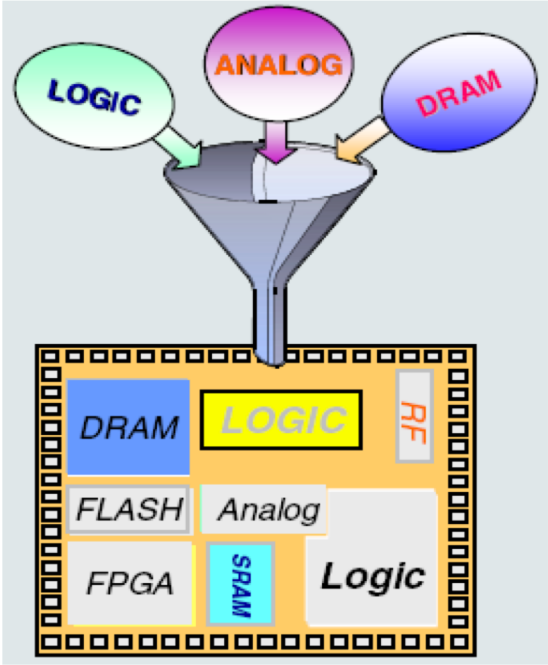
  - Productividad de 15000 puertas/hombre-año.
  - Salario típico de $150.000.
  - Un sistema de 12 millones de puertas supondría:
    - 800 diseñadores durante un año.
    - $120 millones de coste.

Desafíos del diseño

- Tecnológicos:
  - Sistemas más complejos.
  - Mayor densidad de integración.
  - Mejores prestaciones.
  - Menor consumo.
- Comerciales:
  - Ciclos de desarrollo cada vez más cortos.

Diseño multidisplinar

- Antes lo habitual era que en cada chip se emplease una sola tecnología.
- Con los SoC esto ha cambiado: incorporan lógica, circuitos analógicos, memoria...
- También se incorporan bloques de tecnología avanzada: FPGA, memoria Flash, RF/microondas.
- O más allá de la Electrónica: MEMS (Micro Electro Mechanical Systems), optoelectrónica.

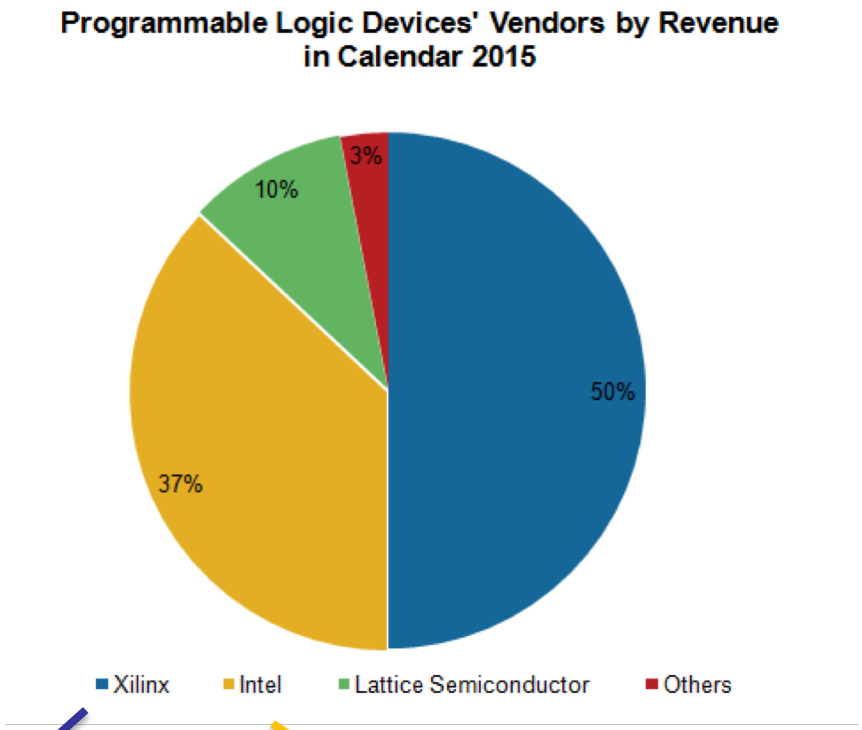


¿Cómo enfrentar la complejidad?

- Metodologías nuevas y más potentes de diseño.
- Uso de abstracciones de más alto nivel.
- Automatización del mayor número de tareas posible con herramientas avanzadas.
- Arquitecturas avanzadas basadas en la modularidad.
- Reutilización extensiva de los diseños.

Mercado FPGAs

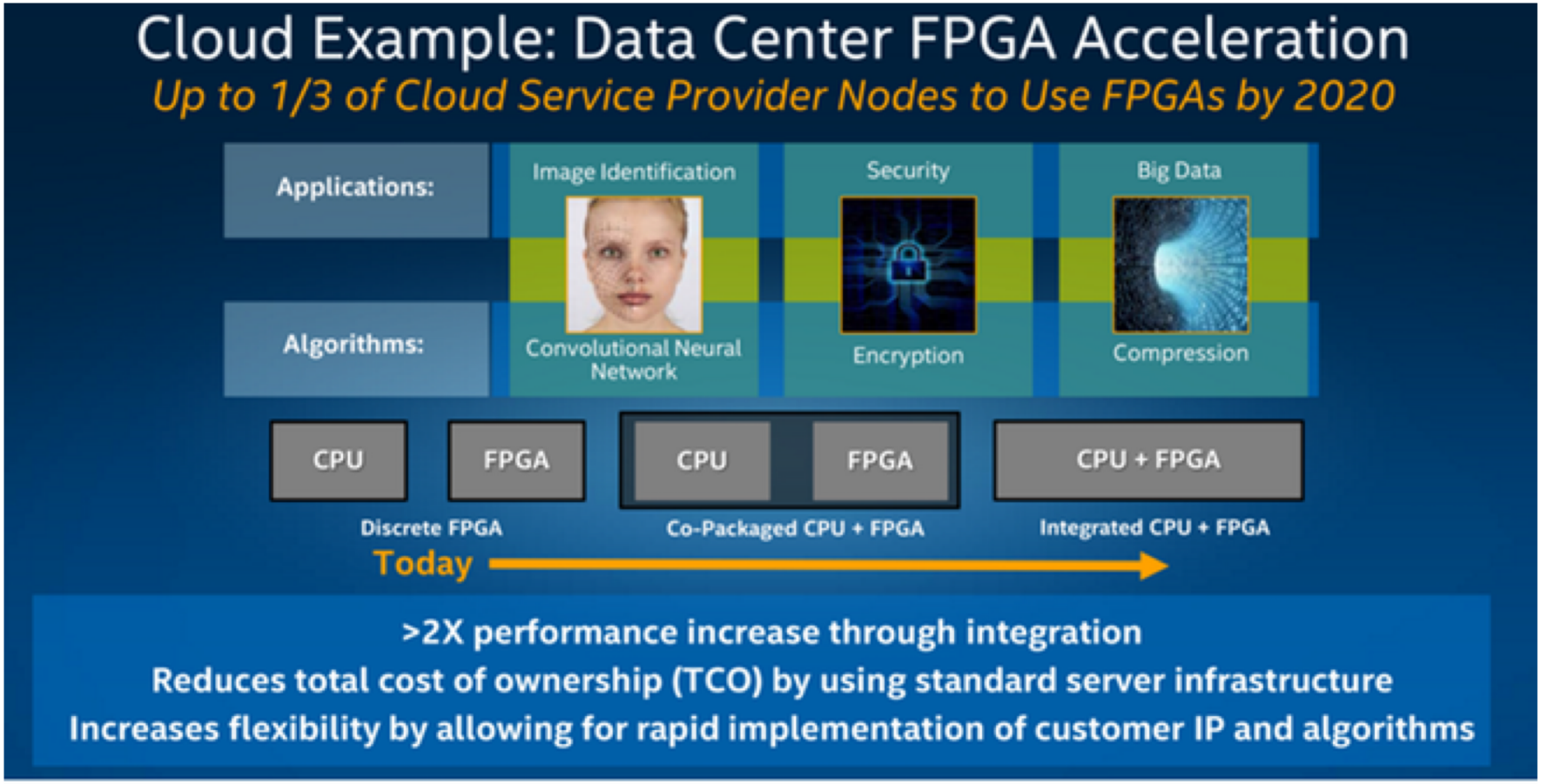
	2015		2016		
Vendor	FPGA Total	Market share	FPGA Total	Market share	Growth CY15-CY16
Xilinx	$2,044	53%	$2,167	53%	6%
Intel (Altera)	$1,389	36%	$1,486	36%	7%
Microsemi	$301	8%	$297	7%	-1%
Lattice	$124	3%	$144	3%	16%
QuickLogic	$19	0%	$11	0%	-40%
Others	$2	0%	$2	0%	0%
TOTAL	$3,879	100%	$4,112	100%	6%



VHDL

Verilog

Futuro?


Cloud Example: Data Center FPGA Acceleration
Up to 1/3 of Cloud Service Provider Nodes to Use FPGAs by 2020
Applications:
Image Identification
Security
Big Data
Algorithms:
Convolutional Neural Network
Encryption
Compression
CPU
FPGA
CPU
FPGA
CPU + FPGA
Discrete FPGA
Co-Packaged CPU + FPGA
Integrated CPU + FPGA
Today
>2X performance increase through integration
Reduces total cost of ownership (TCO) by using standard server infrastructure
Increases flexibility by allowing for rapid implementation of customer IP and algorithms
Market Realist
Source: Intel's Presentation

FIN