

Microcontroladores y Microprocesadores

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

Microcontroladores y Microprocesadores (y FPGAs)

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

Objetivos

- Conocer las diferencias entre microcontroladores y microprocesadores
- Familiarizarse con las generalidades de las arquitecturas microprocesador, así como algunas técnicas de diseño y elementos constructivos relevantes
- Conocer cómo es la arquitectura interna de una FPGA

Contenido

- Microcontroladores y microprocesadores
- Revisión de arquitecturas microprocesador
- Concepto de pipeline
- Memorias caché
- Buses para microprocesadores empujados
- Arquitectura de FPGAs

Contenido

- **Microcontroladores y microprocesadores**
- Revisión de arquitecturas microprocesador
- Concepto de pipeline
- Memorias caché
- Buses para microprocesadores empujados
- Arquitectura de FPGAs

Microcontroladores y microprocesadores

Microcontroladores

- MCU: Microcontroller Unit
- Circuitos integrados que combinan CPU, memoria y periféricos en un único chip
- Suelen tener menos prestaciones que los microprocesadores

Microprocesadores

- MPU: Microprocessor Unit
- Circuitos integrados que únicamente contienen una CPU, y requieren de componentes externos para la memoria y periféricos de entrada y salida

Microcontroladores y microprocesadores

- Al necesitar menos hardware externo, los microcontroladores suelen ser buena opción para sistemas empuotrados
- Con la mejora de las tecnologías CMOS y la capacidad de integración, hoy día los hay muy potentes (32/64 bits, cachés de datos e instrucciones, etc)
- A veces se abusa un poco del lenguaje y se le llama a todo “microprocesador”, aunque sea un MCU

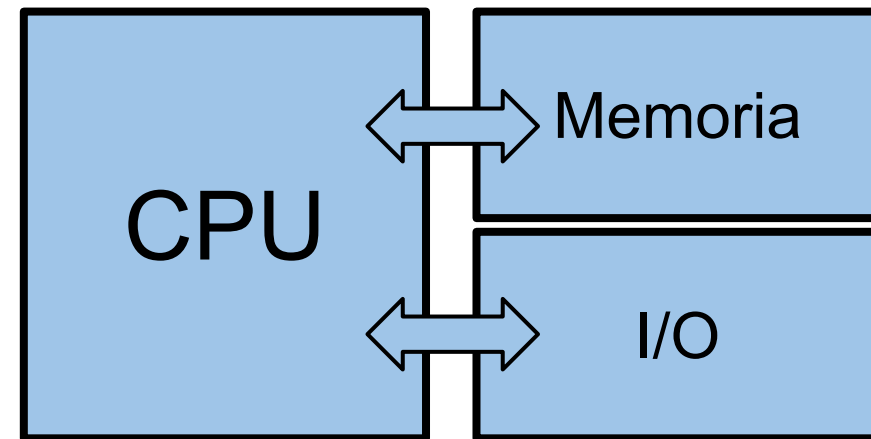
Contenido

- Microcontroladores y microprocesadores
- Revisión de arquitecturas microprocesador
- Concepto de pipeline
- Memorias caché
- Buses para microprocesadores empujados
- Arquitectura de FPGAs

Revisión arquitecturas clásicas

La separación entre la CPU y la memoria hace que el computador sea programable

- CPU (Unidad Central de Proceso)
- Memoria
- Perif. Entrada/Salida



Von Neumann vs Harvard

Von Neumann

La misma memoria
contiene datos e
instrucciones

Único bus
datos/direcciones
entre CPU y memoria

Von Neumann bottleneck!

Harvard

Memorias separadas
para datos e
instrucciones

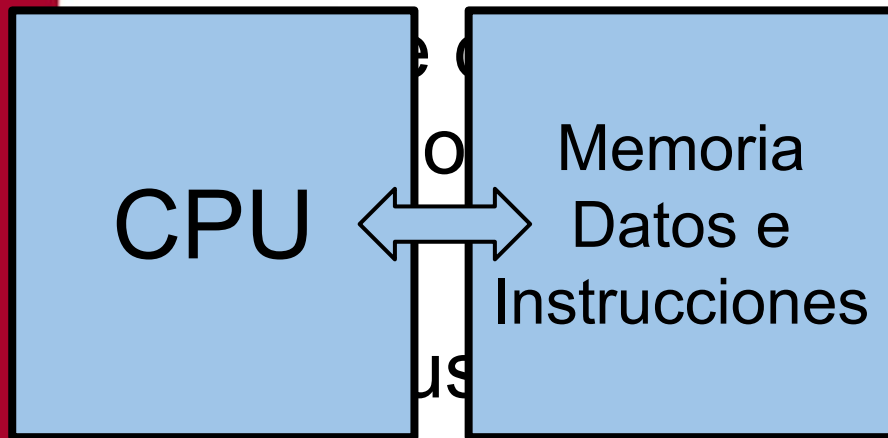
Dos buses
datos/direcciones
entre CPU y mem(s)

Mayor uso de recursos (\$)

Von Neumann vs Harvard

Von Neumann

La misma memoria

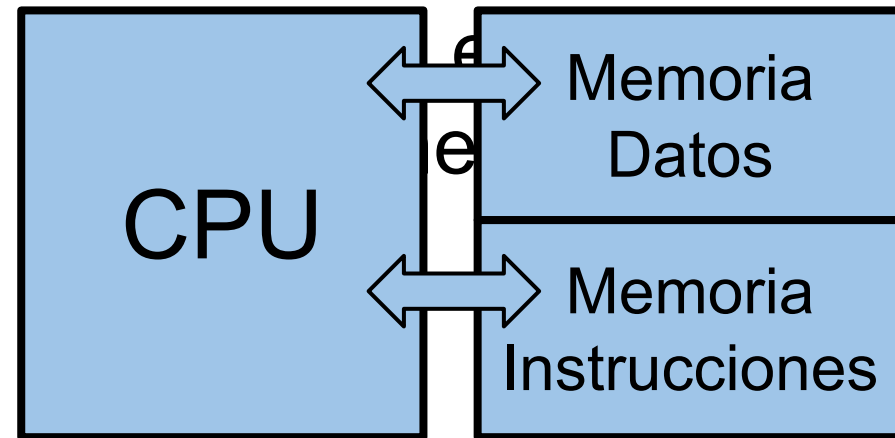


datos/direcciones
entre CPU y memoria

Von Neumann bottleneck!

Harvard

Memorias separadas



datos/direcciones
entre CPU y mem(s)

Mayor uso de recursos (\$)

RISC vs CISC

Reduced Instruction Set Computer

Instrucciones compactas y uniformes

Facilita el pipelining

Mayor uso de memoria

Permiten mejores optimizaciones al compilador

Complex Instruction Set Computer

Instrucciones anchas

Múltiples modos de direccionamiento

Uso de memoria más eficiente (alta densidad de código)

Posible necesidad de optimizaciones manuales (sist. empotrados)

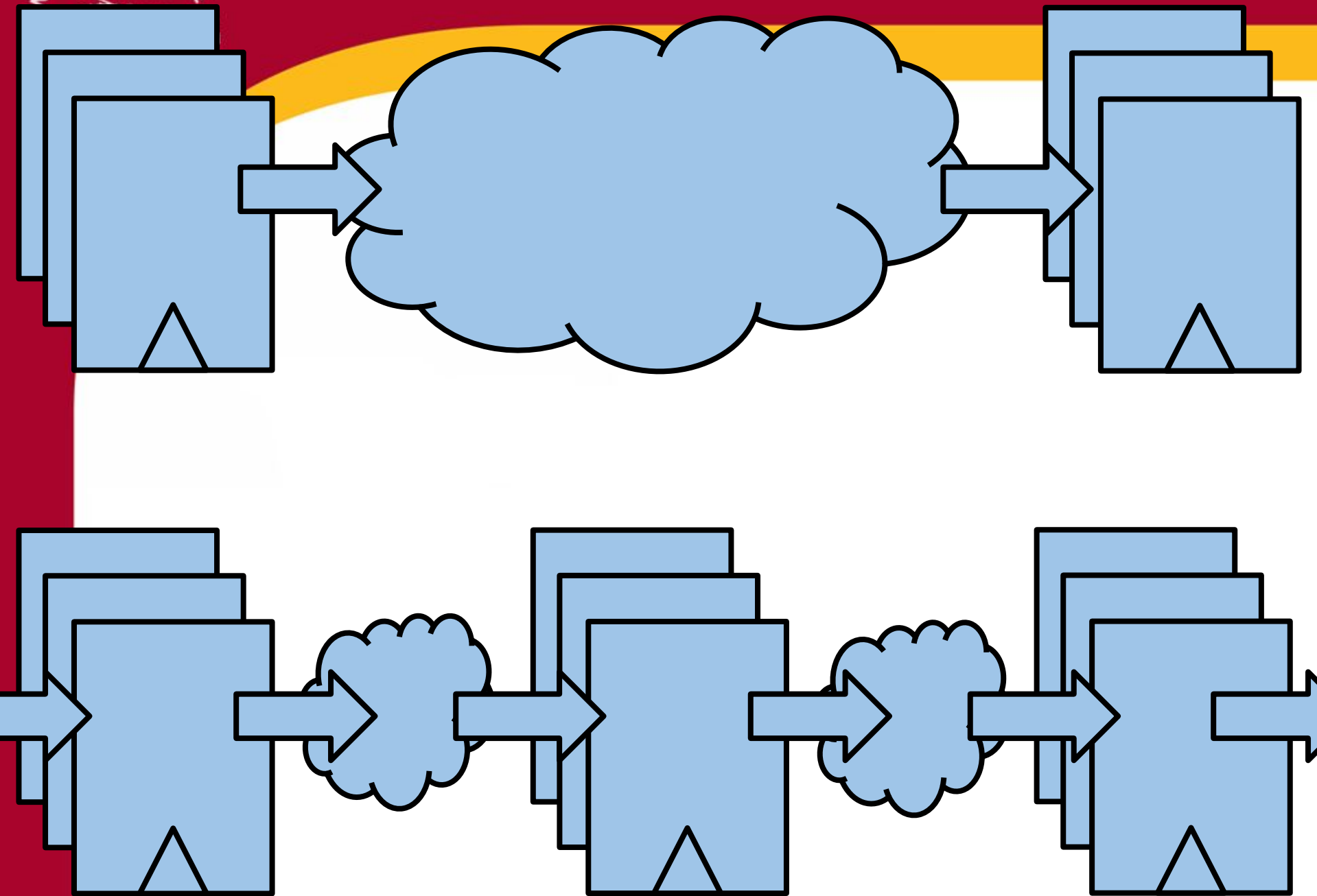
Algunos ejemplos

	Von Neumann	Harvard
RISC	MSP430 ARM7	MicroBlaze Leon4 ARM9
CISC	Pentium	8051 SHARC DSP DSPs en general

Contenido

- Microcontroladores y microprocesadores
- Revisión de arquitecturas microprocesador
- **Concepto de pipeline**
- Memorias caché
- Buses para microprocesadores empujados
- Arquitectura de FPGAs

Pipeline



Latencia y Throughput

Latencia: tiempo entre que entra el dato y sale procesado

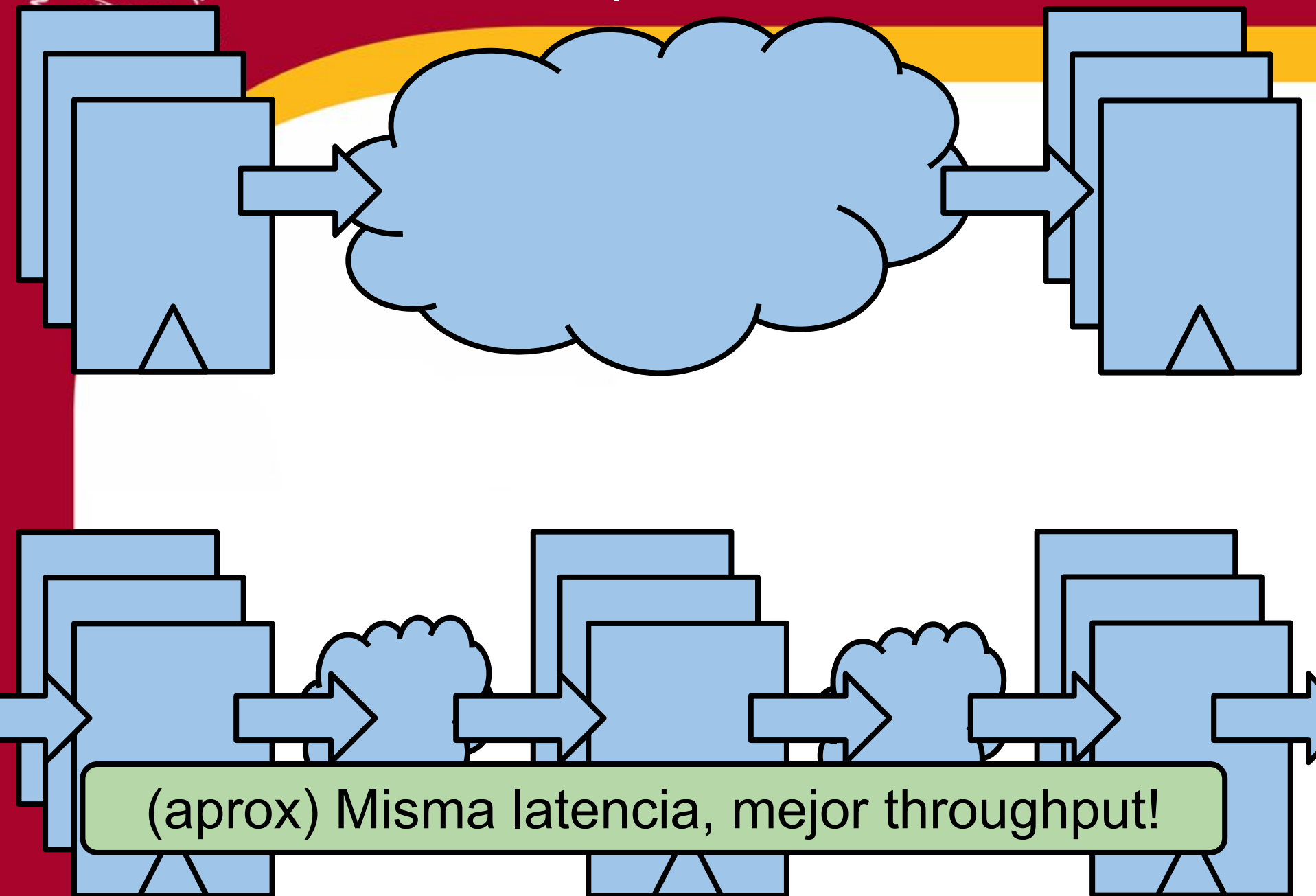
- Tarda M ciclos ($\times X$ ns/ciclo) en cruzar el pipeline completo

Throughput: cada cuántos ciclos sale un dato nuevo

- Un dato cada N ciclos ($\times X$ ns/ciclo)

¡Importante la frecuencia de reloj (ns/ciclo) a la hora de comparar diferentes implementaciones!

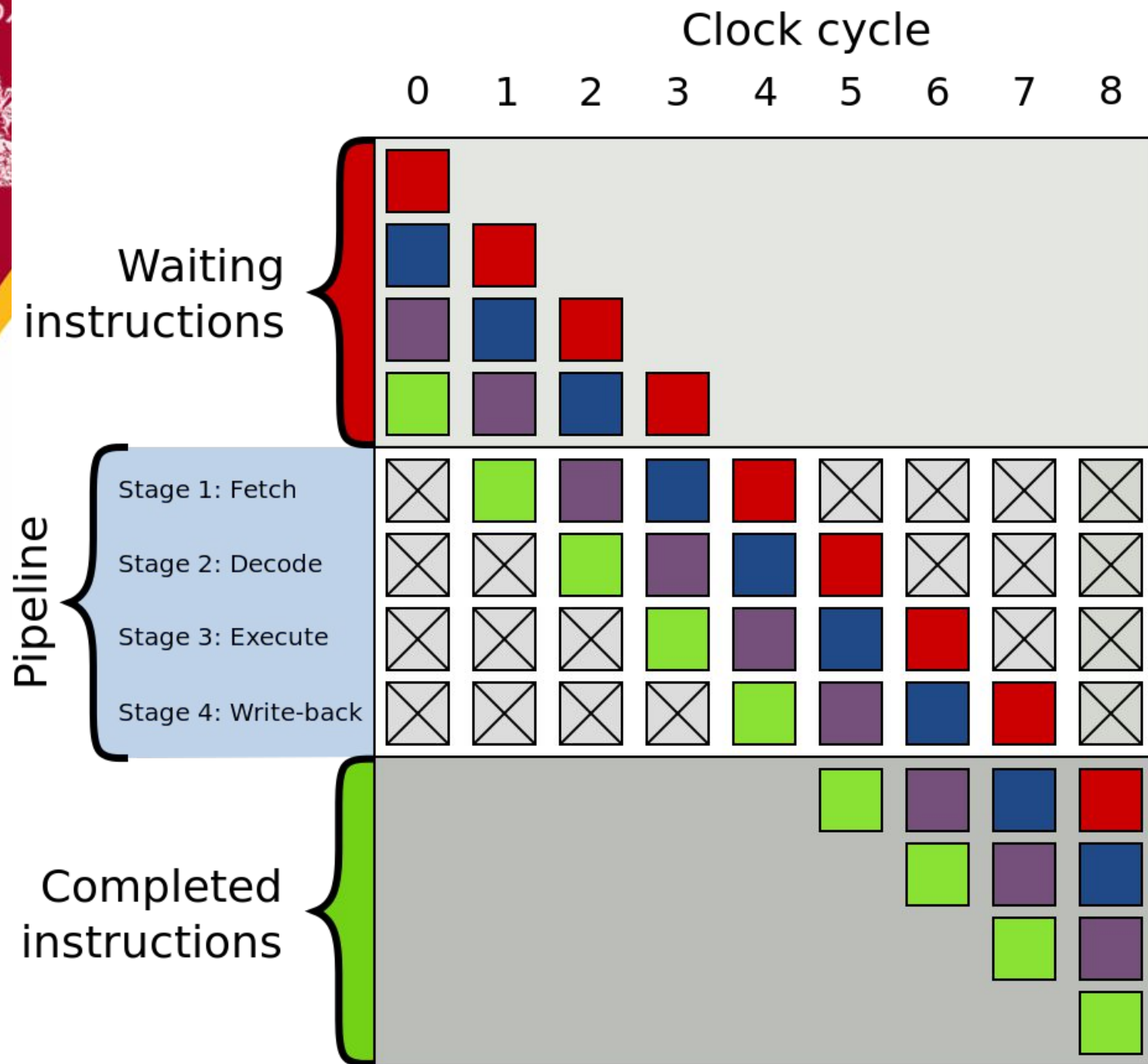
Pipeline



Técnica muy utilizada en microprocesadores

Si para cada instrucción tenemos que hacer:

- Fetch
- Decode
- Execute
- Write-back

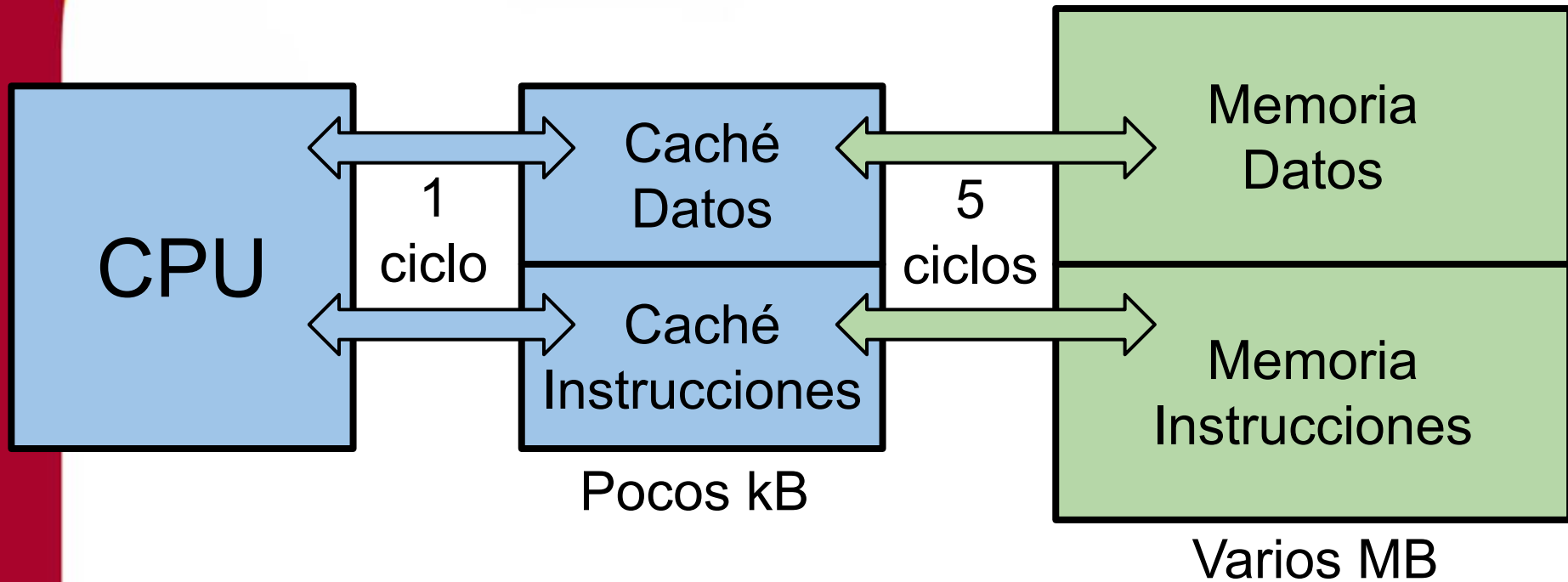


Fuente: [Wikipedia](https://es.wikipedia.org/wiki/Procesador_de_datos)

Contenido

- Microcontroladores y microprocesadores
- Revisión de arquitecturas microprocesador
- Concepto de pipeline
- **Memorias caché**
- Buses para microprocesadores empujados
- Arquitectura de FPGAs

Memoria auxiliar de acceso más rápido



Funcionamiento

La caché tiene menos capacidad que la memoria externa, pero es más rápida.

La caché guarda 'paginas' de memoria completas

Es 'transparente' para el procesador: el micro habla con la caché y la caché lee la memoria externa si necesita datos

Funcionamiento

Cache hit: la dirección (address) solicitada por el microprocesador está en alguna de las páginas almacenadas en la caché (más rápido).

Cache miss: la dirección solicitada no está almacenada en la caché. La caché debe realizar el acceso externo (más tiempo).

Buses para microprocesadores empotrados

Periféricos específicos tienen interfaces con buses específicos.

Un microprocesador concreto soportará uno o varios buses:

- AXI4 (ARM A9 de la Zynq-7000)
- PLB, AXI4 (MicroBlaze)
- Wishbone (estándar en Opencores)
- AMBA (Leon)

Buses para microprocesadores empotrados

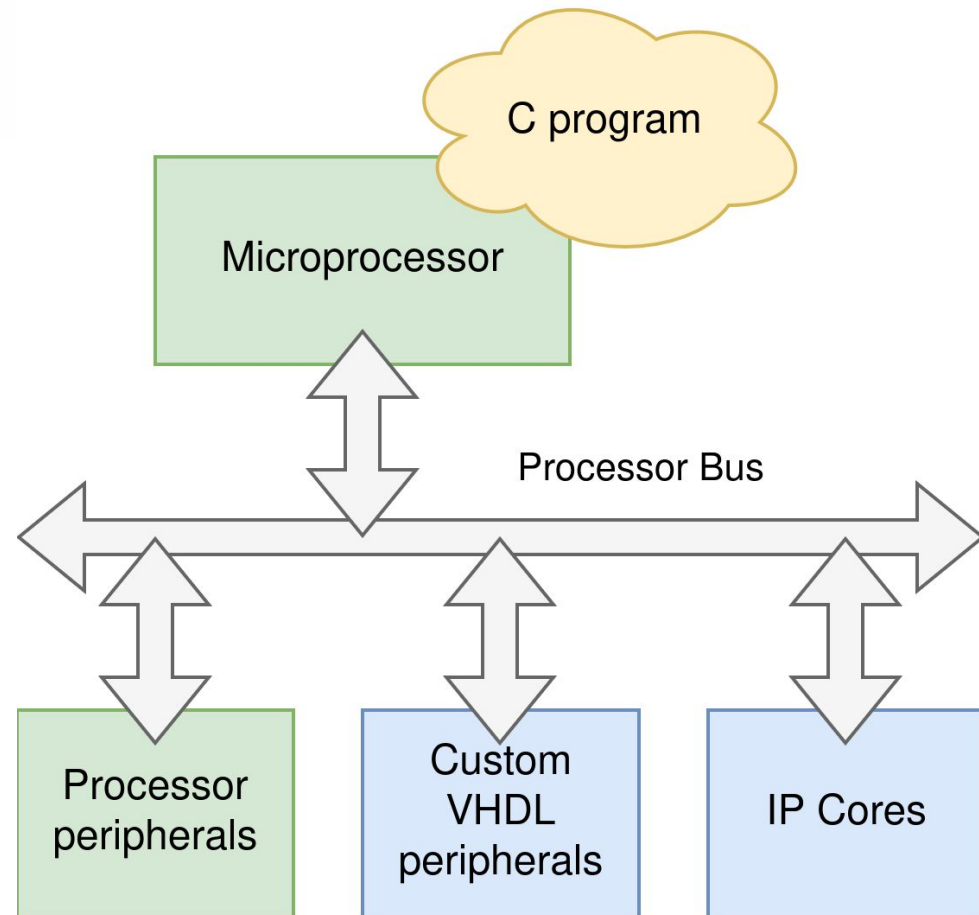
Los IP cores de terceros que integremos y los que desarrollaremos nosotros deberían tener un interfaz con el bus elegido

Aunque si no necesitamos velocidad podremos utilizar una conexión por GPIO

De esta forma: diseño de System-On-Chip complejos utilizando IP cores basados en estándares

¿System-on-Chip?

- Muchas FPGAs modernas incluyen un microcontrolador (MCU) potente integrado en el silicio.
- En particular, la Zynq-7020 que utilizamos en la asignatura tiene un ARM Cortex-A9 dual-core
- El micro se puede comunicar con el 'FPGA fabric' (donde se implementa nuestro VHDL) a través del bus AXI
- Más información en el tema de System-on-Chip



Contenido

- Microcontroladores y microprocesadores
- Revisión de arquitecturas microprocesador
- Concepto de pipeline
- Memorias caché
- Buses para microprocesadores empujados
- **Arquitectura de FPGAs**

Arquitectura interna FPGAs

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

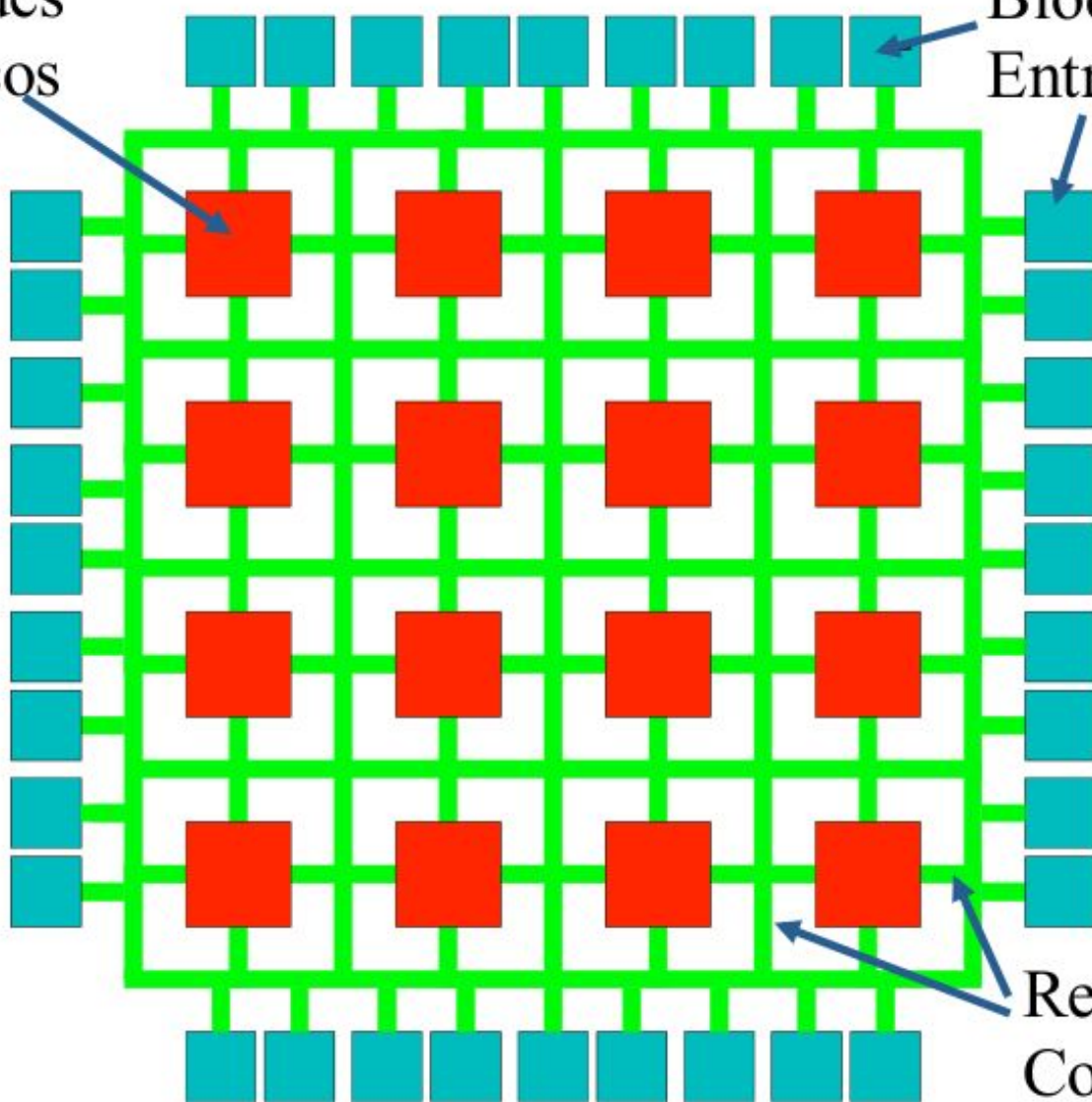
Arquitectura interna de una FPGA

- IOBs: In/Out Blocks (Bloques de Entrada/Salida)
- CLBs: Configurable Logic Blocks (Bloques Lógicos Configurables)
- Routing Resources (Recursos de Conexión)
- (Re)Programabilidad

Arquitectura de una FPGA

Bloques Lógicos

Bloques de Entrada/Salida



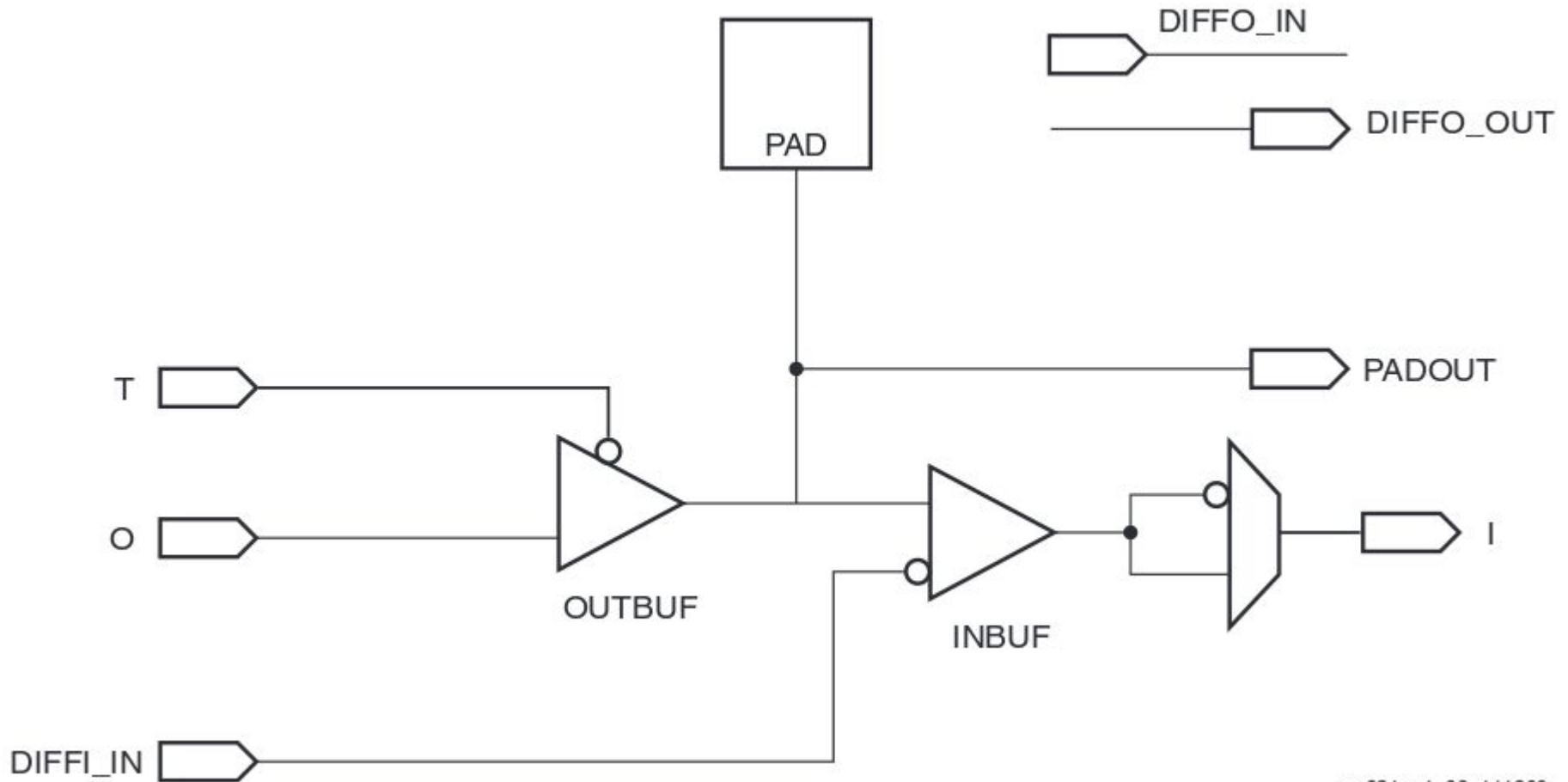
Programabilidad

Recursos de Conexión

In-Out Blocks (IOBs)

- PAD (conexión al exterior)
- Buffer de entrada
- Buffer de salida (triestado)
- Soporte para entradas/salidas diferenciales (dependiendo de la familia)

IOB Spartan-6



ug381_c1_02_111309

Configurable Logic Blocks (CLBs)

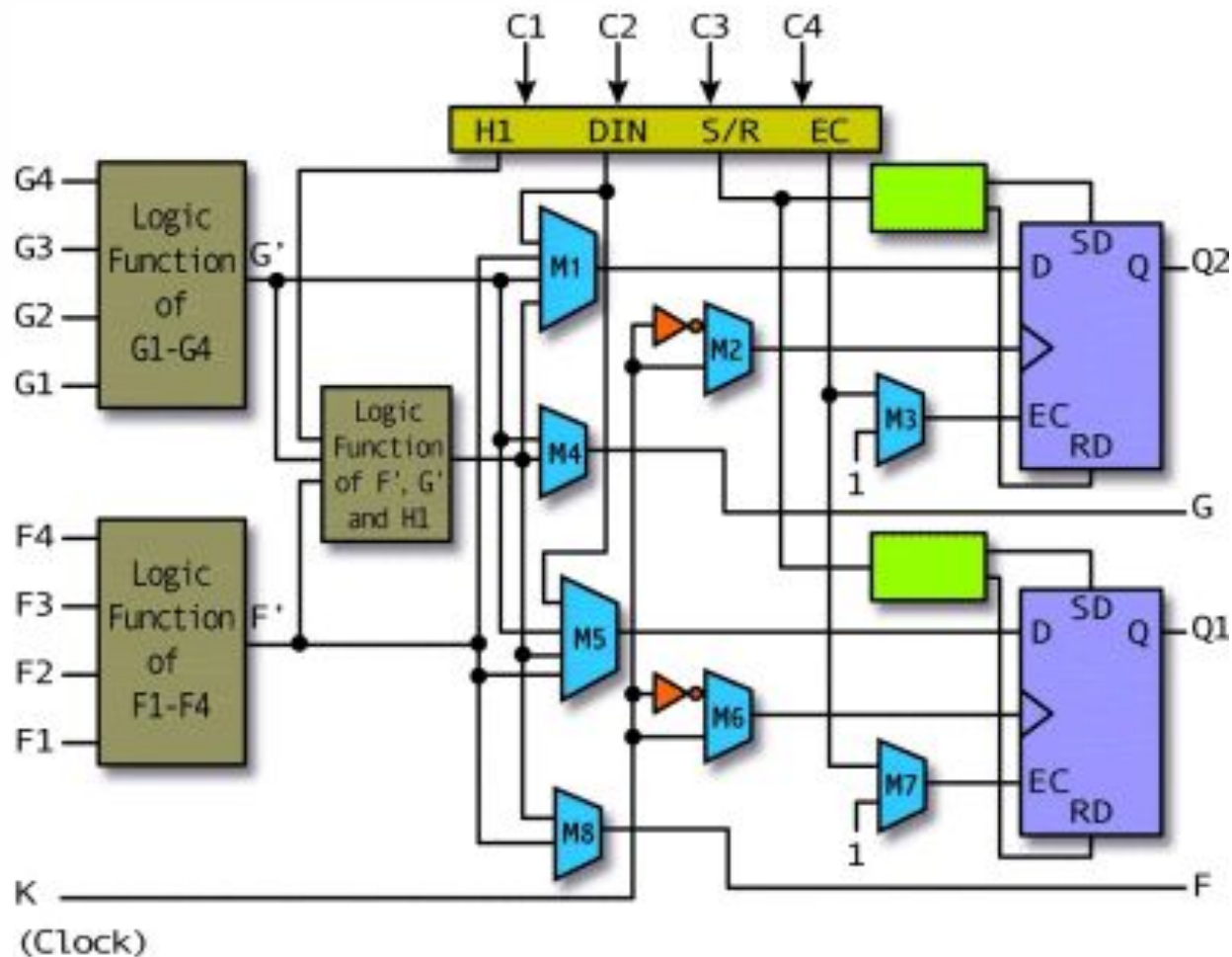
Compuestas por una o varias parejas de:

- N-input LUT (Look-Up Tables de N entradas)
- Flip-flops configurables

Tecnologías modernas tienen más parejas de LUT+FF

N crece también en tecnologías modernas (6 en Spartan-6)

Configurable Logic Blocks (CLBs)

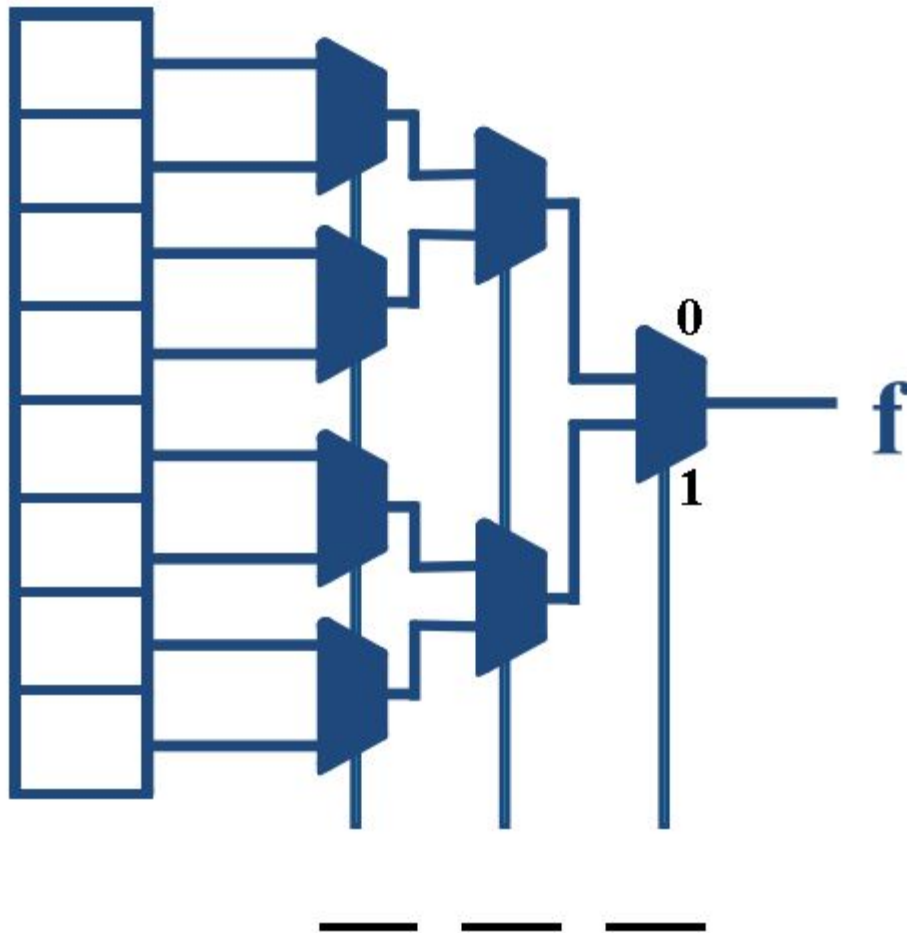


Look-Up Tables (LUTs)

En lugar de implementar las funciones lógicas con puertas lógicas, en FPGA se implementan con tablas de verdad

- Ej: Una LUT de 4 entradas ('4-LUT') puede implementar cualquier función lógica de 4 entradas

SRAM



3-LUT

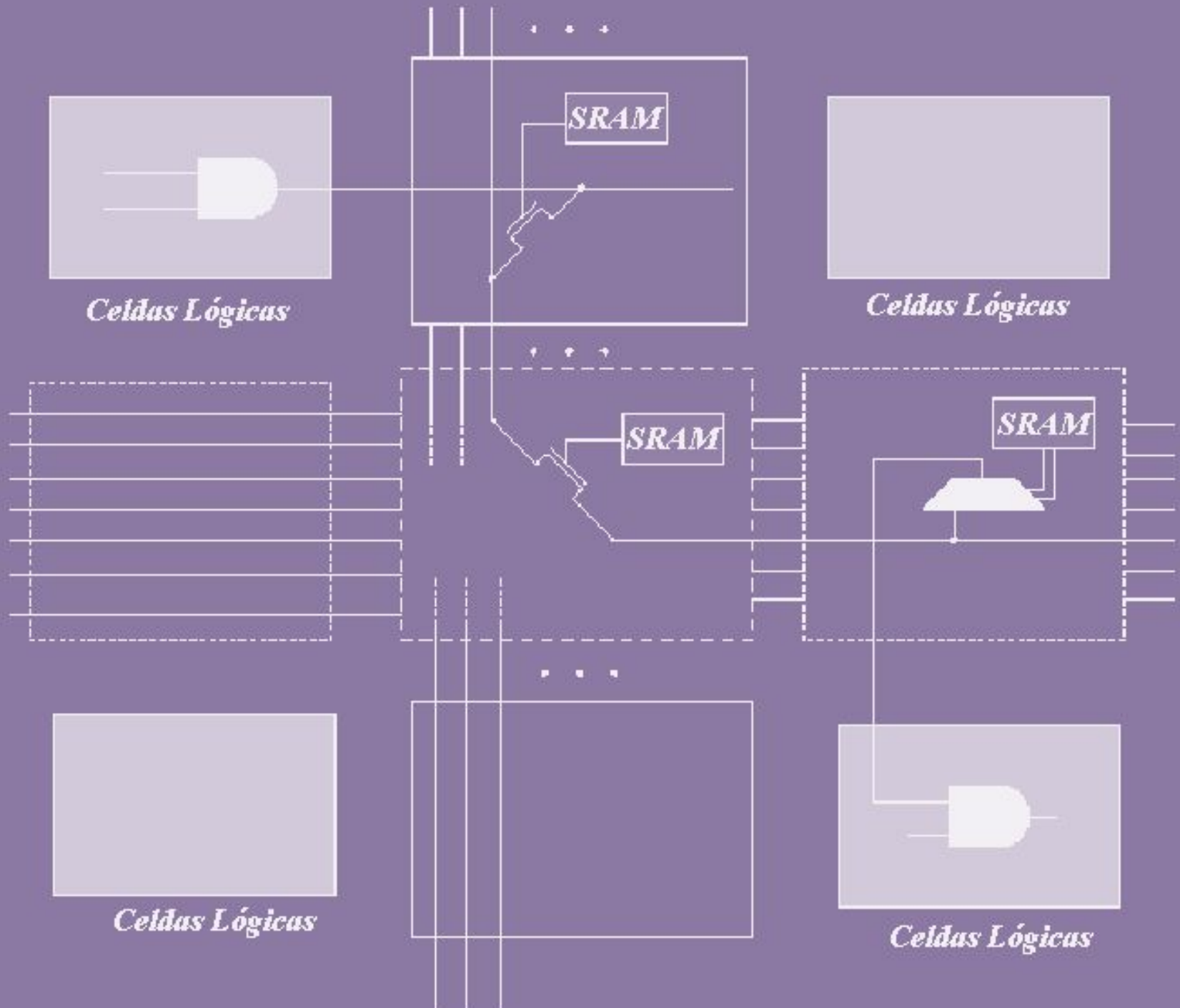
Ejercicio

Configura la LUT de forma que implemente la función

$$F = ABC + ABC$$

Recursos de Rutado

- PIP: Programmable Interconnection Points (Puntos de Interconexión Programable)
- Líneas cortas y largas
- Recursos dedicados para relojes (ej: BUFG)



Configurabilidad y Reconfigurabilidad

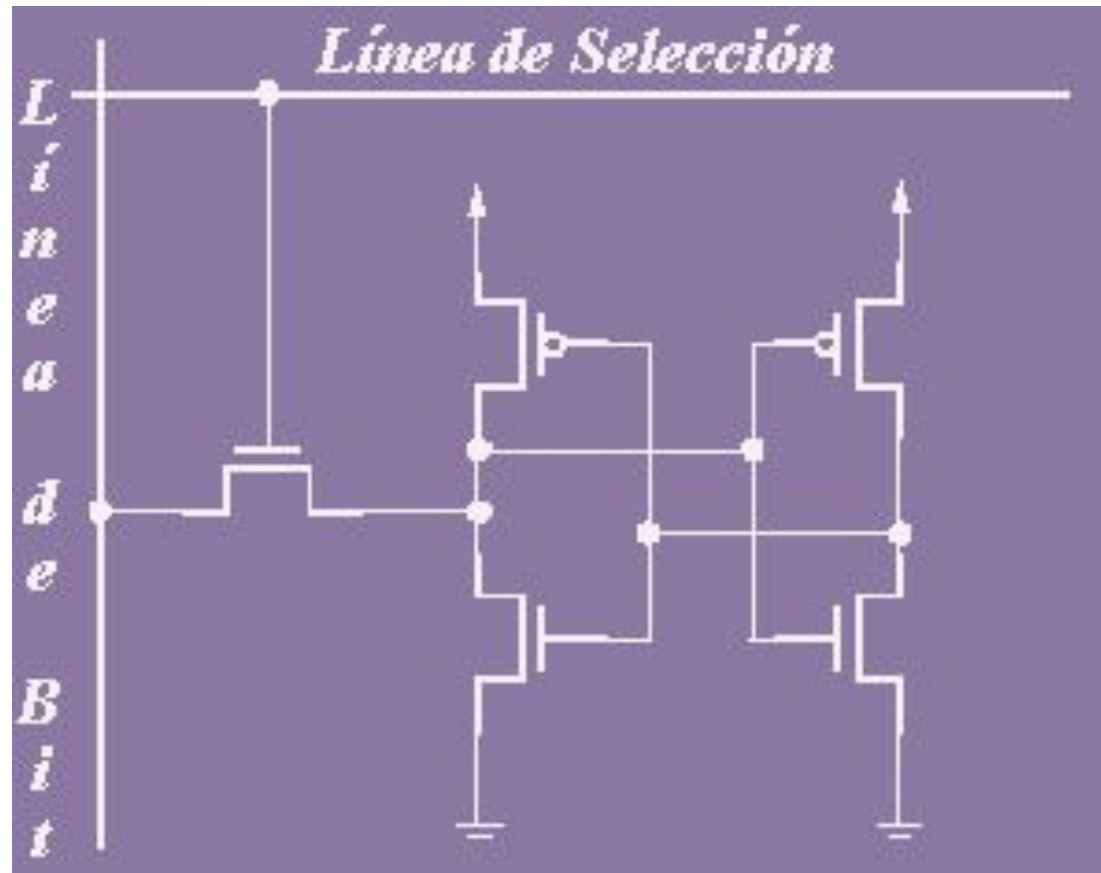
Existen varias tecnologías:

- SRAM: reconfigurable, volátil, muy extendida, aprovecha proceso CMOS estándar
- Flash: reconfigurable, no volátil, proceso no estándar
- Antifusible: no reconfigurable, proceso no estándar

Celda SRAM

Son dos
inversores
realimentados

4 transistores,
pero CMOS
estándar



Tecnologías Flash

Se basan en el uso de FGMOS (Floating-Gate MOS)

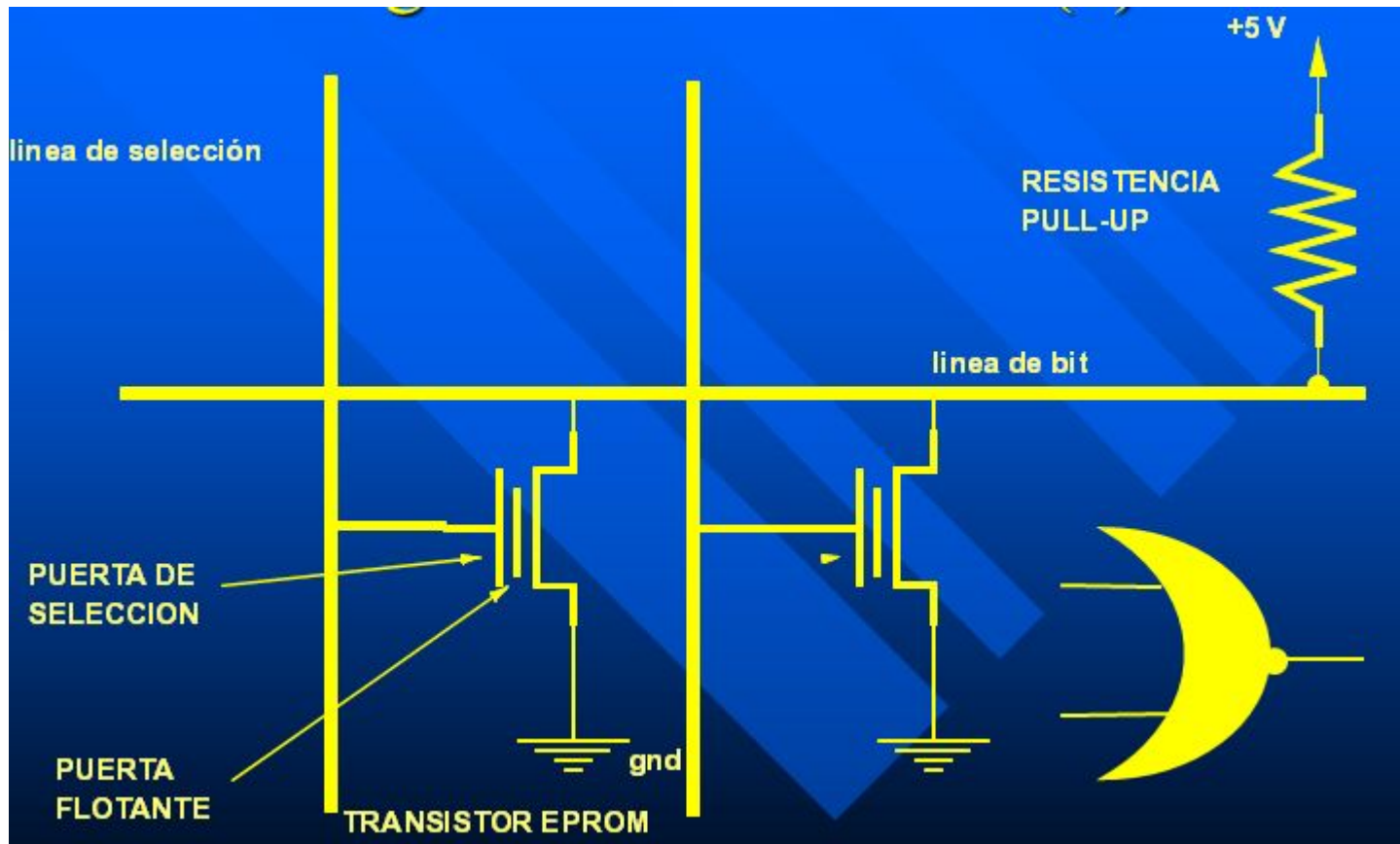
- Por lo que requieren de tecnologías con 2 niveles de polisilicio

Si la puerta flotante está cargada:

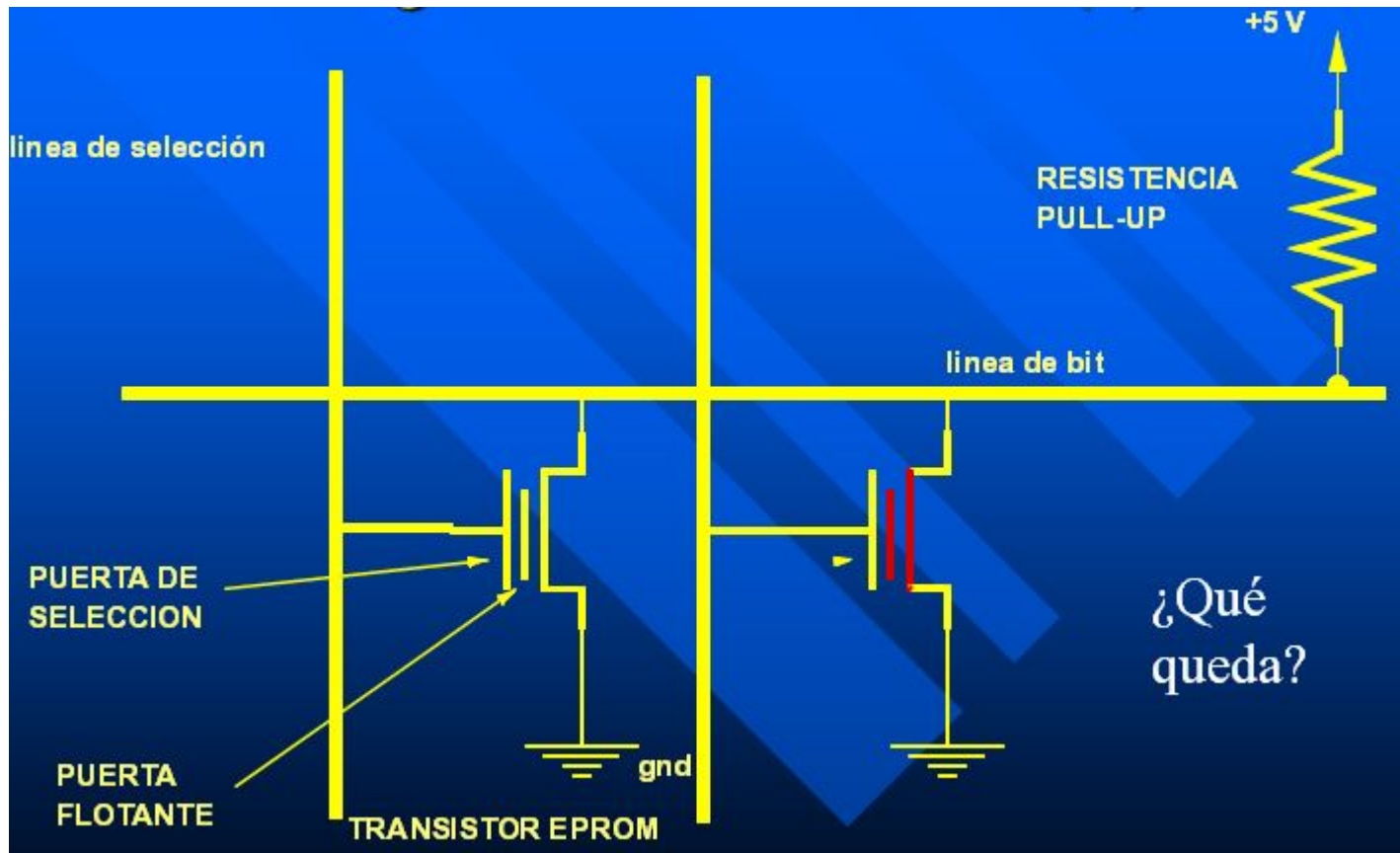
-> Incremento de la V_t

-> Transistor no puede encenderse ni con V_{dd} en la puerta

Ejemplo configurabilidad Flash

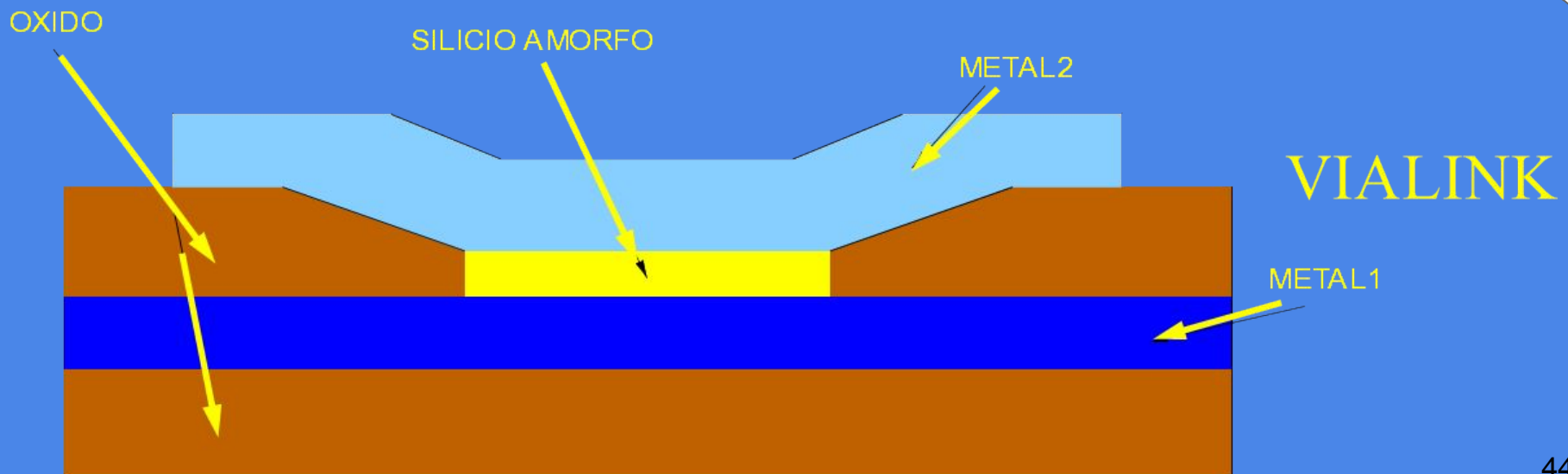


Ejemplo configurabilidad Flash



Tecnologías Antifusible

- OTP: One Time Programmable
- Requieren un proceso específico (no CMOS estándar)



Conclusiones

- Microcontroladores y microprocesadores no son exactamente lo mismo, pero sus arquitecturas tienen muchísimas similitudes
- Podemos hablar de 'arquitectura microprocesador' en general, para la parte que es igual
- La técnica de diseño 'pipeline' es muy útil para mejorar las prestaciones de los circuitos digitales, entre ellos los microprocesadores
- Las memorias caché aceleran la ejecución del software y son indispensable en MCU/MPU modernos

Conclusiones

- Las memorias caché aceleran la ejecución del software y son indispensable en MCU/MPU modernos
- La arquitectura interna de las FPGAs es totalmente diferente a las de los MCU/MPU, y permite implementar circuitos que funcionan realmente en paralelo

Resultados de aprendizaje

- Conocer las diferencias entre microcontroladores y microprocesadores
- Saber qué es la técnica del pipeline y cómo se utiliza para incrementar las prestaciones de circuitos digitales en general y en arquitecturas microprocesador en particular
- Entender qué es una memoria caché y cómo se utiliza para mejorar las prestaciones de un microcontrolador/microprocesador

Resultados de aprendizaje

- Entender por qué es necesario que una FPGA disponga de IOBs/CLBs/Recursos de Rutado/Bits de Configuración
- Entender cómo se implementan las funciones lógicas en una FPGA utilizando Look-Up Tables
- Conocer las distintas tecnologías de configuración de FPGA, entendiendo cuáles son volátiles (y cuáles no) y cuáles pueden configurarse múltiples veces (y cuáles no)