

Tema 14: Comunicaciones

Arquitectura de Computadoras

Ing. Nicolás Majorel Padilla (npadilla@herrera.unt.edu.ar)

<http://microprocesadores.unt.edu.ar/arqcom/>

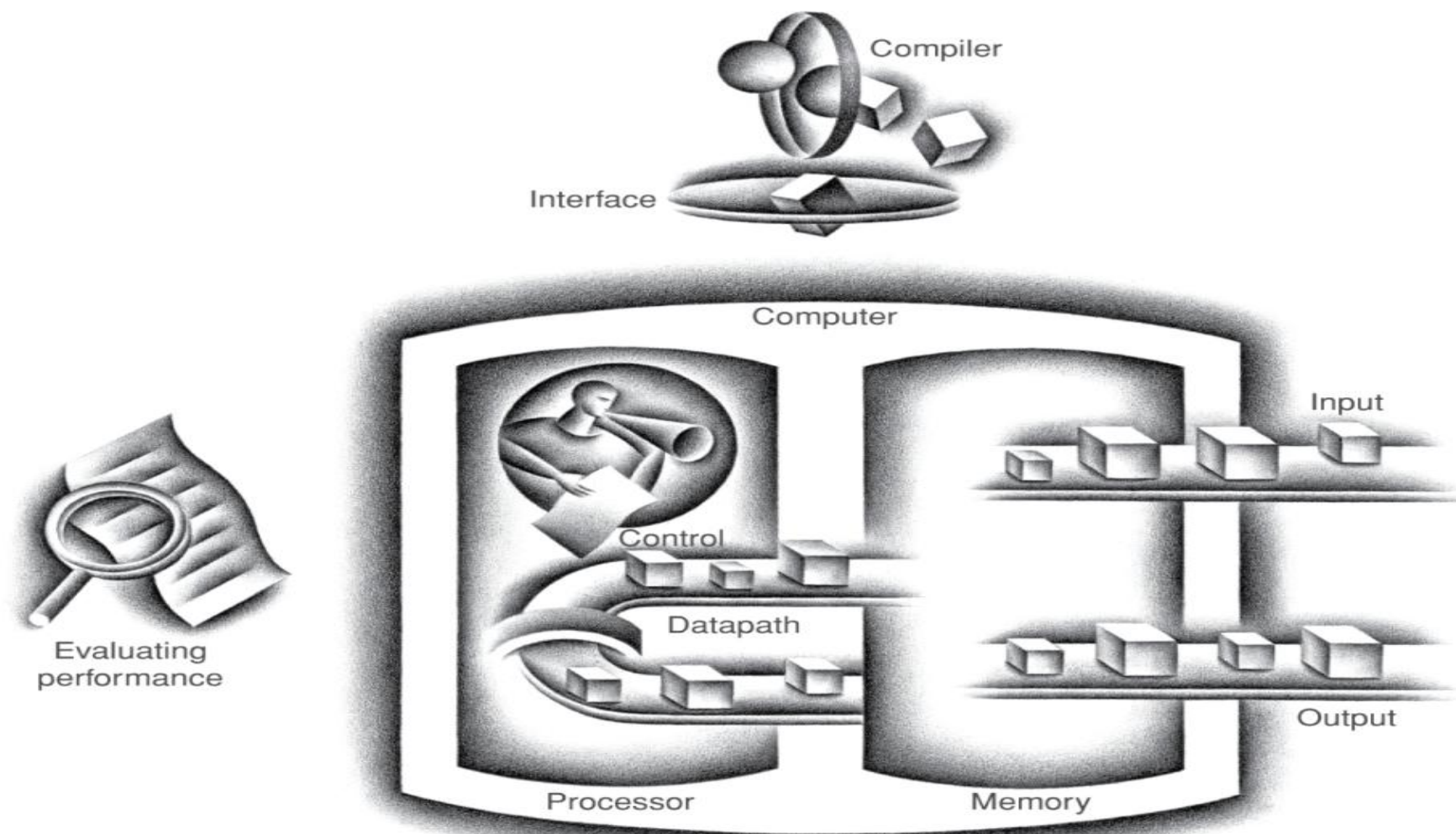
Temas que veremos

- ▶ Bus: concepto, características esenciales, clasificación y topologías más comunes.
- ▶ Conexiones *off-chip*: buses paralelos y buses seriales.
- ▶ Evolución de esquemas de interconexión: desde lo básico hasta el SoC.
- ▶ Conexiones *on-chip*.
- ▶ Conexiones de gran escala.
- ▶ Tendencias actuales.

Lectura recomendada

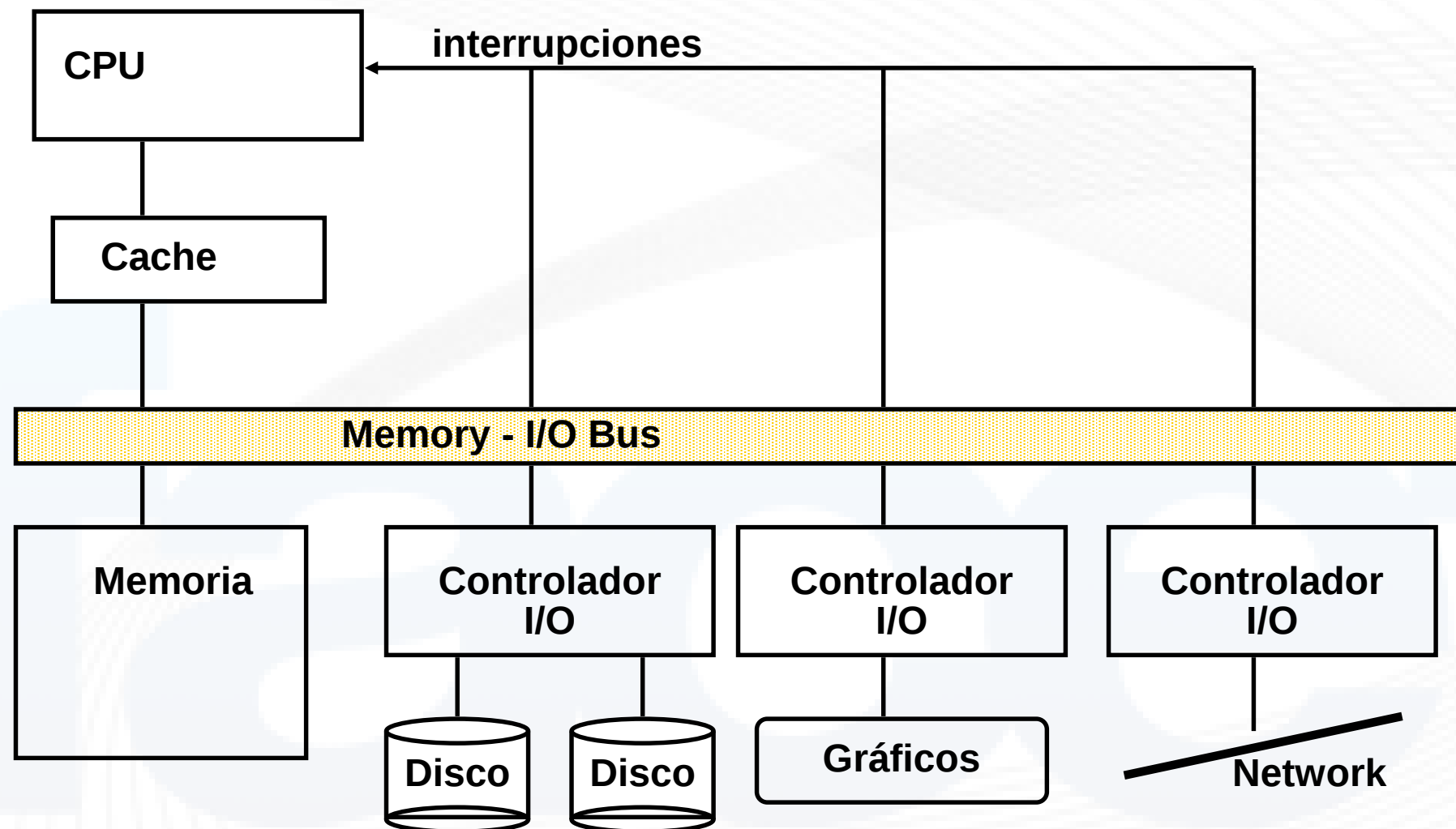
- ▶ Computer Organization and Design, RISC-V Edition (2da ed, 2021)
 - ▶ Sección 6.9: *Introduction to Multiprocessor Network Topologies*
 - ▶ Sección 6.10: *Communicating to the Outside World*

Repaso: Partes de una computadora



- ▶ Ya vimos sobre Procesadores, sus caminos de datos y su control, sobre el sistema de Memoria y el Sistema de I/O.
- ▶ Ahora veremos cómo se conectan todos esos componentes.

Repaso: Esquema básico del Sistema de I/O



- ▶ Existencia de **múltiples dispositivos**, con características y comportamientos **muy diferentes**, conectados a uno o más buses.
- ▶ Ya vimos aspectos globales de I/O y ejemplos de dispositivos. Ahora veremos cómo se realizan las transferencias de información.

Evolución tecnológica

- ▶ Inicialmente, los procesadores se conectaban a los dispositivos mediante buses.
 - ▶ La mayoría de estos buses estaban ubicados físicamente en la motherboard.
 - ▶ Los controladores estaban en los mismos dispositivos, o en la motherboard.
 - ▶ El SO debe tener el driver para poder reconocer y manejar cada dispositivo.
- ▶ Con el avance de la Ley de Moore, tanto los buses como los controladores van cambiando de lugar.
 - ▶ Actualmente, es muy común encontrarlos dentro del mismo procesador (en los de alta performance).
 - ▶ En cierto modo, los chips de los procesadores actuales son tan complejos que podría decirse que tienen una motherboard incorporada.
- ▶ Pero empecemos por el principio...

¿Qué es un Bus?

- ▶ Un Bus es un conjunto de “cables” que se usa para conectar múltiples dispositivos.
- ▶ Es una herramienta sistemática de la arquitectura para armar sistemas complejos.
- ▶ Definido por distintas capas:
 - ▶ **Protocolo de Transacciones.**
 - ▶ Especificaciones de señales y tiempo.
 - ▶ Paquete de cables.
 - ▶ Especificaciones eléctricas.
 - ▶ Características físicas y mecánicas (conectores).

Características esenciales de buses

- ▶ Basados en protocolos **estándar**.
 - ▶ **Compatibilidad:** Aseguran que dispositivos de terceras partes puedan funcionar sin inconvenientes.
 - ▶ Sin necesidad de nuevos drivers ni nuevos modelos de programación.
 - ▶ **Efectividad:** Aseguran que las transacciones se realicen correctamente.
 - ▶ **Eficiencia:** ya sea como productividad (ancho de banda) o latencia (tiempo de respuesta).
- ▶ Organización **jerárquica**.
 - ▶ Posibilidad de conectar diversos estándares, mediante **bridges**.
 - ▶ Mejorar la eficiencia: buses más rápidos para dispositivos más rápidos.
 - ▶ Permite transferencias simultáneas.
 - ▶ Simplifica la lógica de los dispositivos lentos.

Características esenciales de buses

- ▶ Típicamente poseen tres tipos de conjuntos de **señales**: direcciones, datos y control.
- ▶ Requieren alguna especie de **arbitraje**.
 - ▶ Para manejar las **contenciones** (más de un dispositivo pretende usar el bus).
 - ▶ Usualmente, algún esquema del tipo Master/Slave o similar.
 - ▶ Puede ser centralizado o distribuido, con diversos esquemas de manejo de prioridades.
- ▶ Si se utilizan para conectar dispositivos de memoria, necesitan manejar **coherencia**.
 - ▶ Permiten actualizaciones automáticas de datos almacenados localmente que son modificados por otros dispositivos conectados.
 - ▶ Simplifican la sincronización de datos.

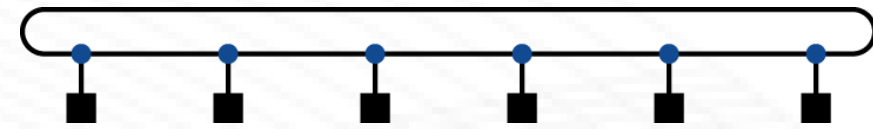
Características esenciales de buses

- ▶ Métricas usadas
 - ▶ Principales, las de siempre: **Ancho de banda** (Bytes/s) y **Latencia** (s).
 - ▶ Suele usarse el ancho de banda de bisección.
 - ▶ También es bastante usada Transferencias por segundo (T/s).
 - ▶ Frecuencia (Hz) no muy utilizada, porque en general pertenecen a dominios diferentes e independientes.
 - ▶ Ej: procesador de 3 GHz y un bus de 1 GHz, entonces 1 ciclo de bus equivale a 3 ciclos del procesador.
 - ▶ Tolerancia a fallos.
 - ▶ Eficiencia energética.

Clasificaciones de buses

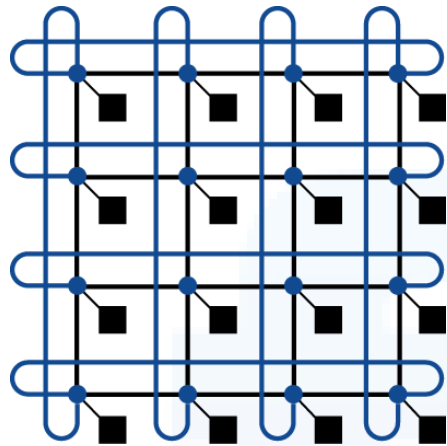
- ▶ Por la cantidad de líneas: Paralelos o Seriales.
- ▶ Por la cantidad de dispositivos: Punto a Punto o Múltiples.
- ▶ Por tipo de sincronización de la información:
 - ▶ Sincrónicos: la señal de reloj va implícita en la señal o en un cable aparte.
 - ▶ Asincrónicos: mediante un protocolo de *handshaking*.
- ▶ Por el tipo de comunicación: *simplex*, *half duplex* o *full duplex*.
- ▶ Por ancho de banda fijo o variable:
 - ▶ Pueden tener múltiples “canales” de comunicación, y usar sólo los necesarios.
- ▶ Por su ubicación dentro de una computadora:
 - ▶ *System bus* o *Memory bus*: tradicionalmente el bus que conecta al procesador con la memoria principal.
 - ▶ *Peripheral bus*: bus usado por los dispositivos más lentos del sistema.
 - ▶ *On-chip bus*: conectan los distintos núcleos y componentes dentro de un mismo chip.

Topologías de redes

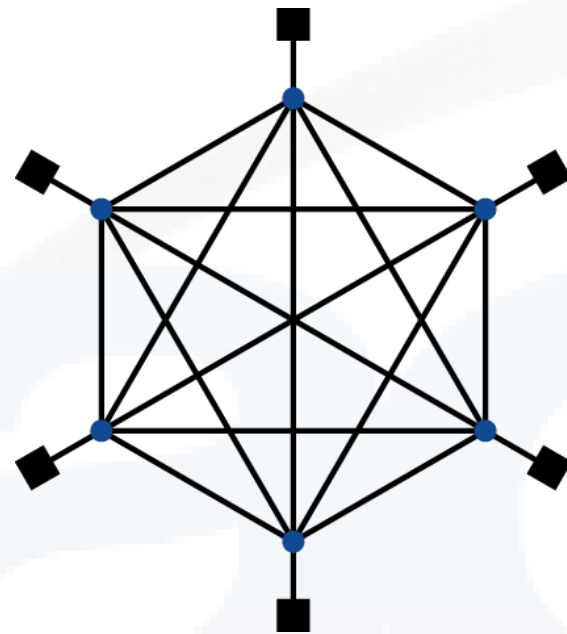


Bus

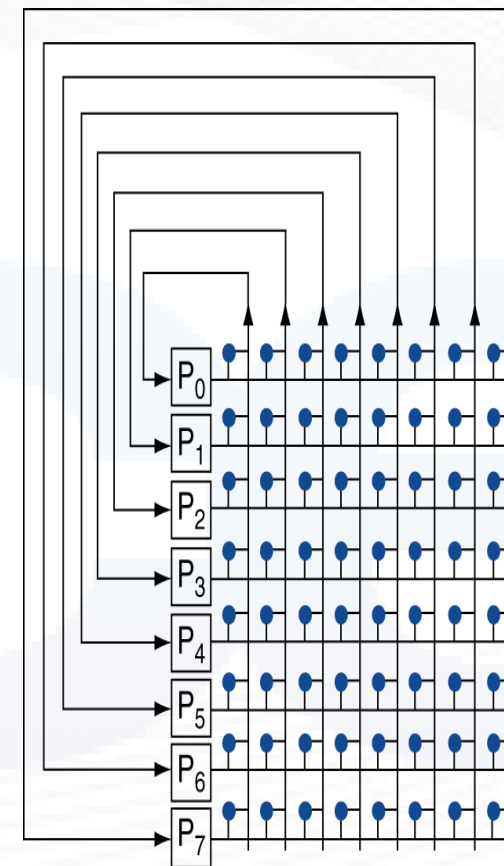
Ring



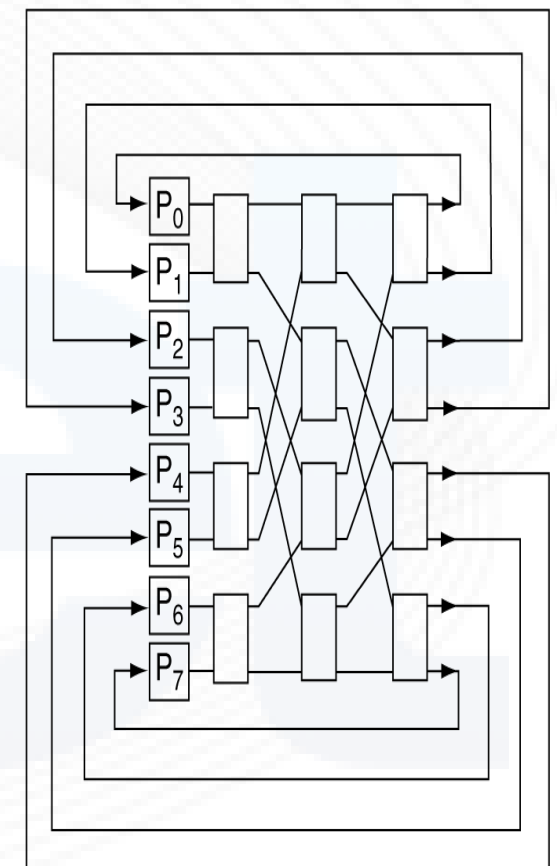
2D Mesh



Fully
connected



a. Crossbar

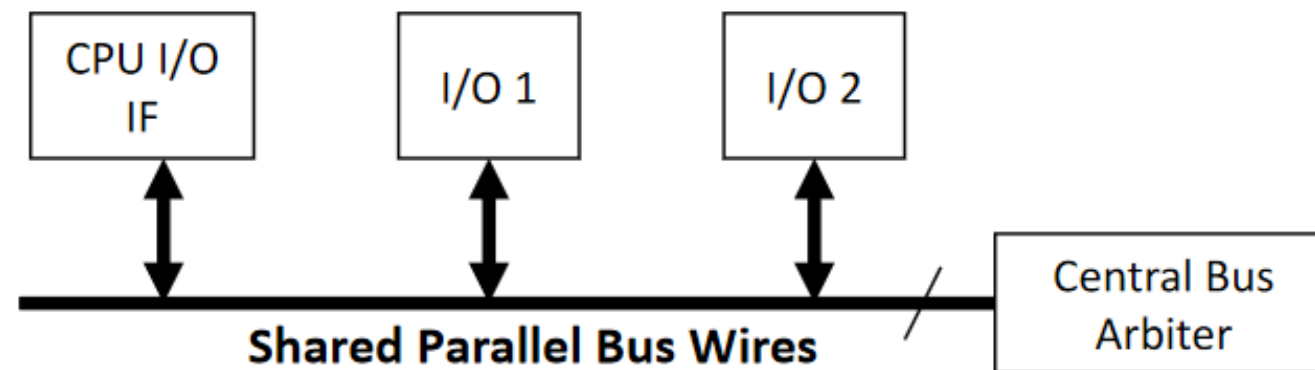


b. Omega network

N-cube ($N = 3$)

- ▶ Actualmente están ganando terreno las “**fábricas**”: una red dinámica de dispositivos, reconfigurable (por sw), sin topología específica.

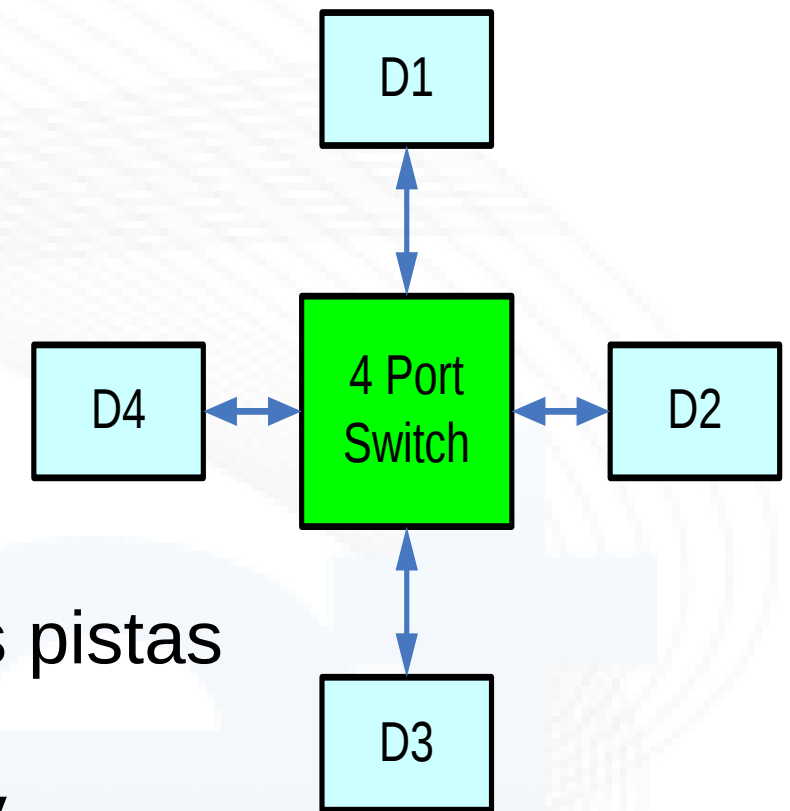
Conexiones off-chip: buses paralelos



- ▶ Usados en un tiempo muy, muy lejano ☺
- ▶ Frecuencia limitada por la propagación de la señal a través del bus (~100 MHz).
- ▶ Cantidad de dispositivos limitada.
 - ▶ Problemas de interferencia entre cables (*crosstalk*) y de retardos de propagación (*skew*).
- ▶ Necesitan mucha energía para transmitir datos por muchos cables paralelos.
- ▶ El arbitraje centralizado añade latencia, limitando el ancho de banda.
- ▶ Los conectores costosos, e iguales para todos los dispositivos.
- ▶ Ejemplos: ISA, PCI, SCSI, IDE.

Conexiones off-chip: buses seriales

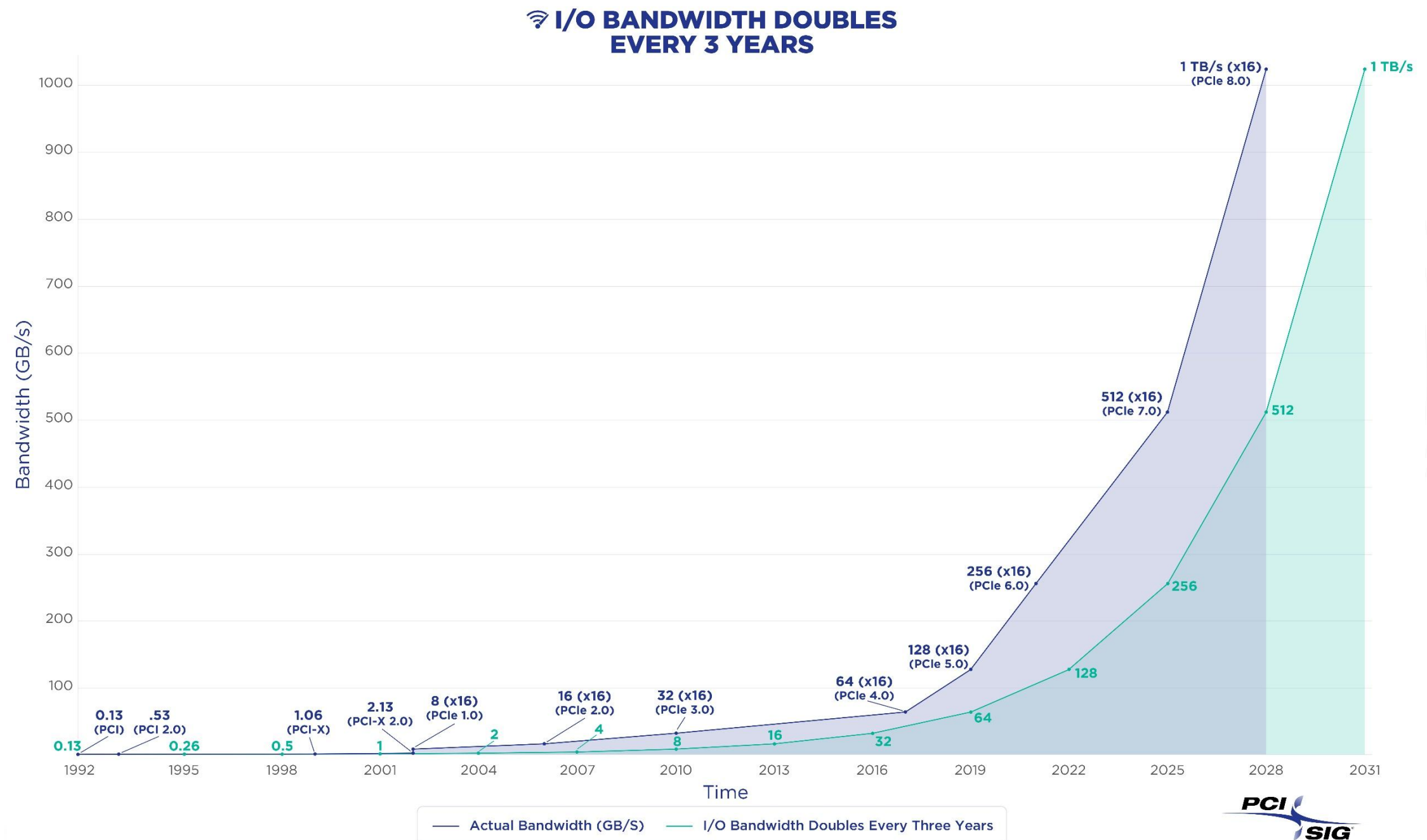
- ▶ Los más usados en la actualidad.
- ▶ Conexiones punto a punto dedicadas para cada dispositivo.
- ▶ Admite frecuencias más altas (~10 GHz).
- ▶ Múltiples transferencias simultáneas.
- ▶ Múltiples pistas independientes (*lanes*).
- ▶ Ancho de banda variable, pudiendo asignar más pistas a dispositivos más rápidos.
- ▶ Menos pins por chip, haciéndolos más simples y baratos.
- ▶ Menor consumo de energía.
- ▶ Baja latencia entre enlaces.
- ▶ Es modular, pudiendo conectar los que uno desee.
- ▶ Ejemplos: Ethernet, PCI Express, SATA, USB, Firewire.



Ejemplo de bus serial: PCIe

- ▶ **Estándar** mantenido por PCI-SIG, una organización sin fines de lucro, formado por más de 900 compañías diferentes.
 - ▶ Intel, AMD, ARM, IBM, NVIDIA, Qualcomm, etc.
- ▶ Objetivo: ser la red de interconexión universal para todo I/O.
- ▶ Totalmente **compatible hacia atrás** con generaciones anteriores.
- ▶ 1.0 anunciado en 2003, **7.0** lanzado en jun-25, **8.0** anunciado en ago-25 (para ser lanzado en 2028).
 - ▶ El estándar va varios años por delante de la realidad de mercado.
 - ▶ ~30 meses desde la especificación hasta el lanzamiento de los primeros productos.
- ▶ Serial, full duplex, punto a punto.
 - ▶ Cada pista (*lane*) necesita 4 cables.

Ejemplo de bus serial: PCIe



- Impulsado por nuevas demandas y por la Ley de Moore.

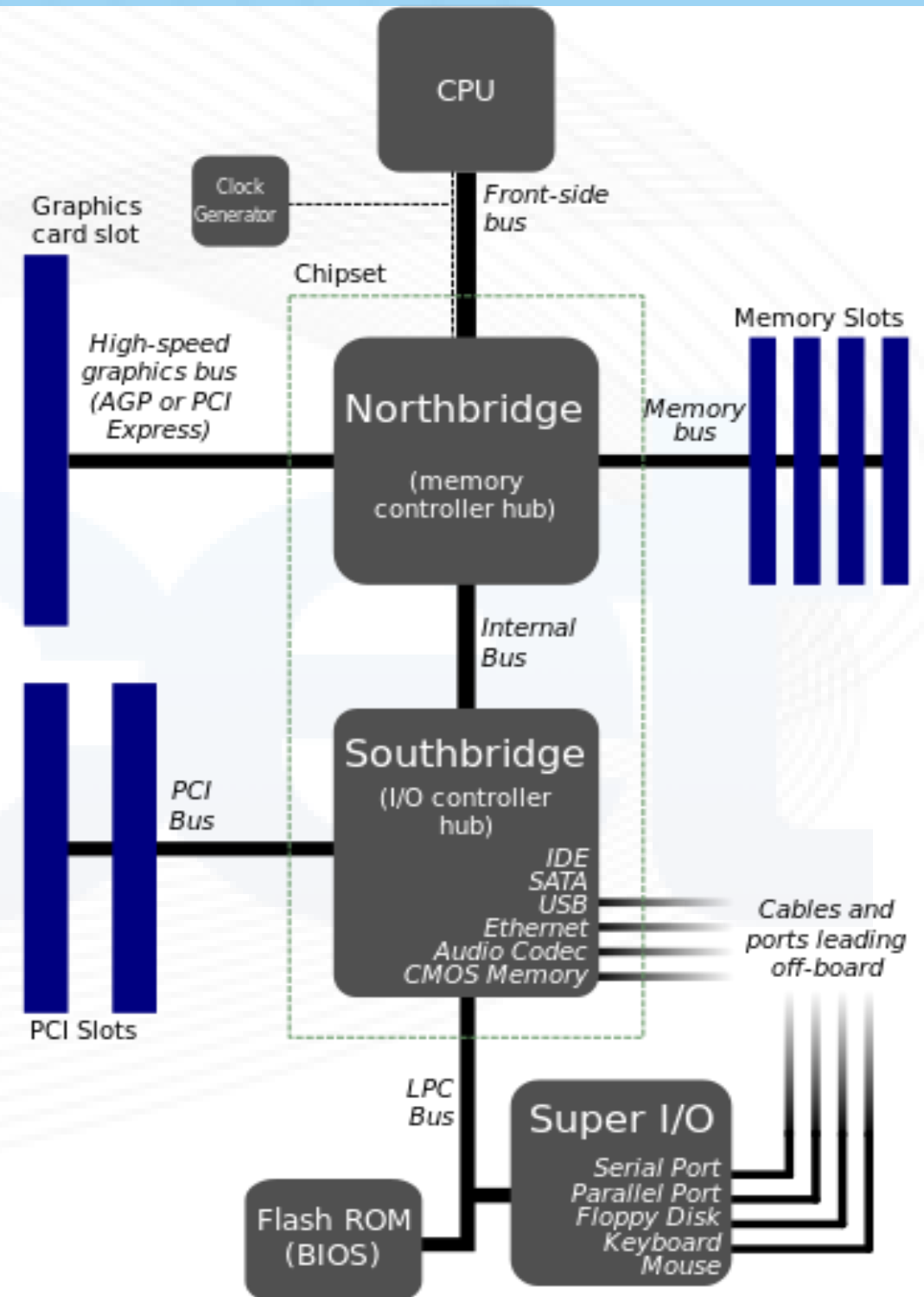
Ejemplo de bus serial: PCIe

- ▶ Gran flexibilidad
 - ▶ Cada dispositivo puede seleccionar cuántas pistas necesita, entre 1 y 16.
 - ▶ Sumado a las 8 versiones compatibles, ofrece 12 anchos de banda entre 40 combinaciones posibles.

Specifications	Lanes				
	x1	x2	x4	x8	x16
2.5 GT/s (PCIe 1.x +)	500 MB/S	1 GB/S	2 GB/S	4 GB/S	8 GB/S
5.0 GT/s (PCIe 2.x +)	1 GB/S	2 GB/S	4 GB/S	8 GB/S	16 GB/S
8.0 GT/s (PCIe 3.x +)	2 GB/S	4 GB/S	8 GB/S	16 GB/S	32 GB/S
16.0 GT/s (PCIe 4.x +)	4 GB/S	8 GB/S	16 GB/S	32 GB/S	64 GB/S
32.0 GT/s (PCIe 5.x +)	8 GB/S	16 GB/S	32 GB/S	64 GB/S	128 GB/S
64.0 GT/s (PCIe 6.x +)	16 GB/S	32 GB/S	64 GB/S	128 GB/S	256 GB/S
128.0 GT/s (PCIe 7.x +)	32 GB/S	64 GB/S	128 GB/S	256 GB/S	512 GB/S
256.0 GT/s (PCIe 8.x +)	64 GB/S	128 GB/S	256 GB/S	512 GB/S	1 TB/S

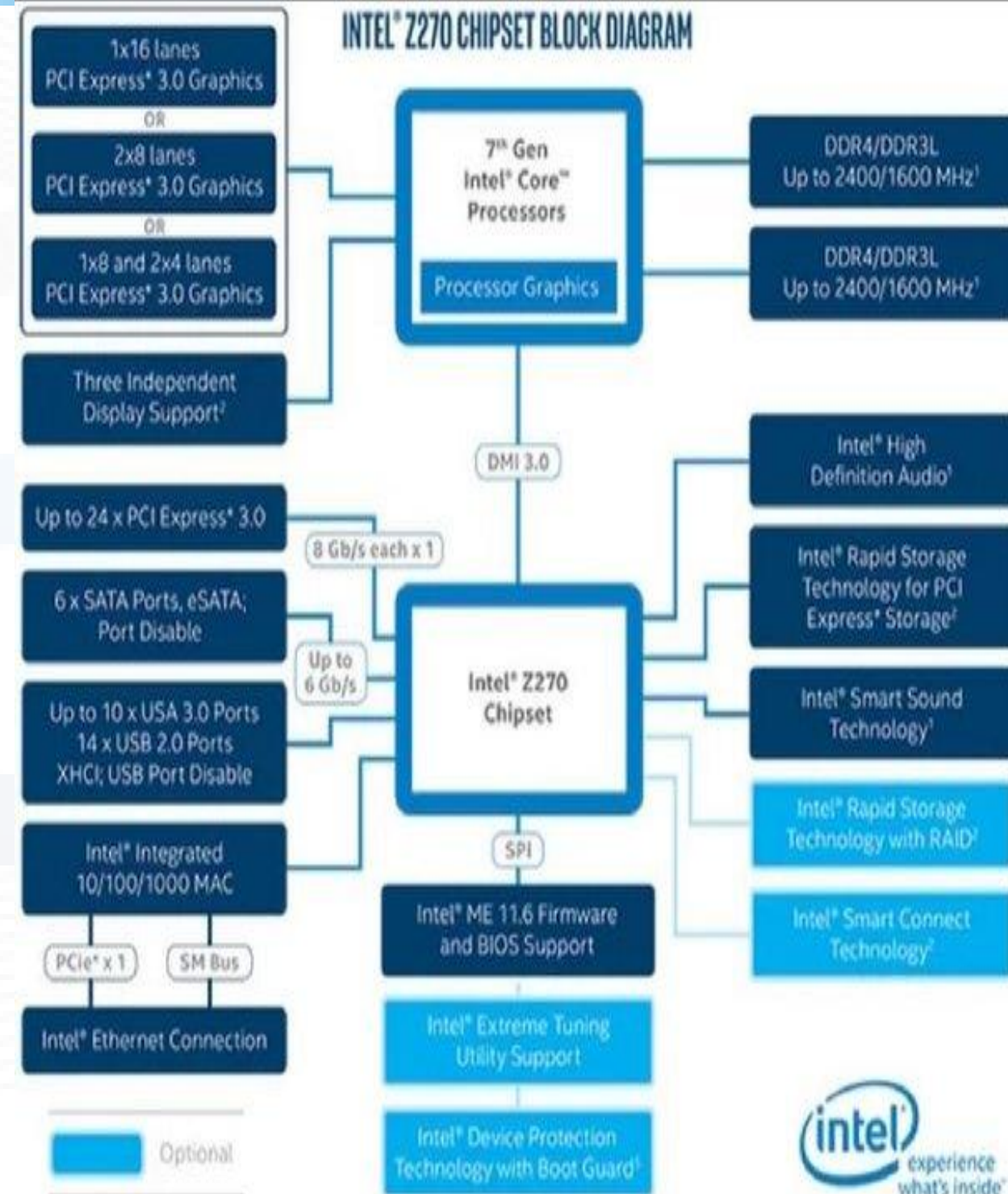
Esquema “típico” de un Sistema de I/O

- ▶ Desde 1998 hasta 2008.
- ▶ **Arquitectura de 3 buses.**
 - ▶ Con distintas frecuencias, y se conectan entre sí mediante “puentes” (*bridges*).
- ▶ Chipset compuesto por un par de chips: **Northbridge** y **Southbridge**.
- ▶ Velocidades (ancho de banda):
 - ▶ FSB de 10,5 GB/s (DDR2).
 - ▶ Internal Bus de 2 GB/s.
 - ▶ SATA de 300 MB/s.
 - ▶ IDE de 100 MB/s.
 - ▶ USB de 60 MB/s.
 - ▶ Teclado/Mouse de 1 MB/s.

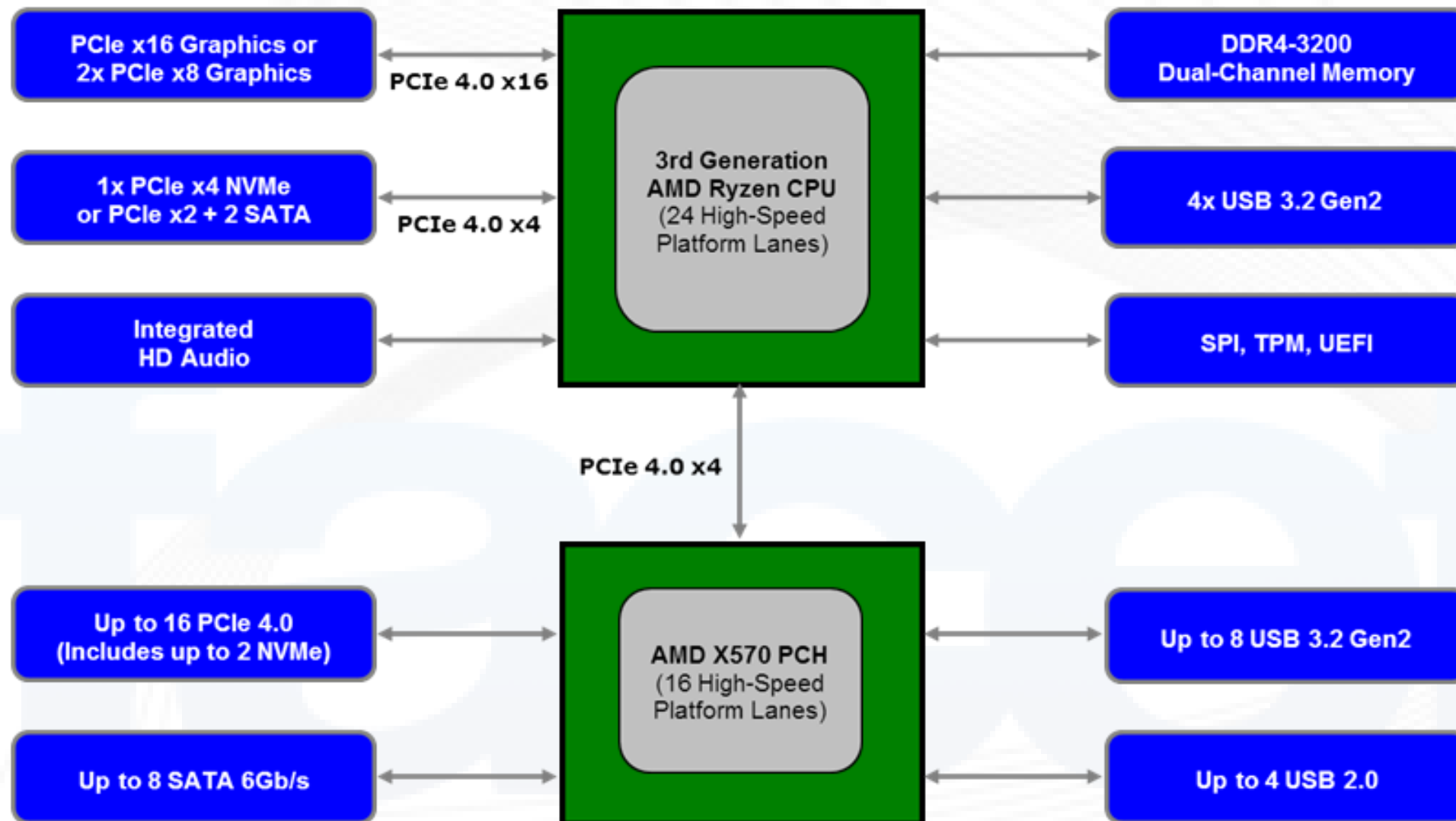


Esquema moderno de un Sistema de I/O

- ▶ Desde 2008, los procesadores de Intel integraron dentro del mismo chip el controlador de memoria y de PCIe.
- ▶ Aumenta la complejidad del procesador (Moore una vez más).
- ▶ Aumenta el ancho de banda y sobre todo la escalabilidad.
- ▶ Ya no hubo necesidad del Northbridge.
- ▶ Reemplazaron el Southbridge por un Platform Controller Hub (PCH).
- ▶ Reemplazaron el FSB por un Direct Memory Interface (DMI).

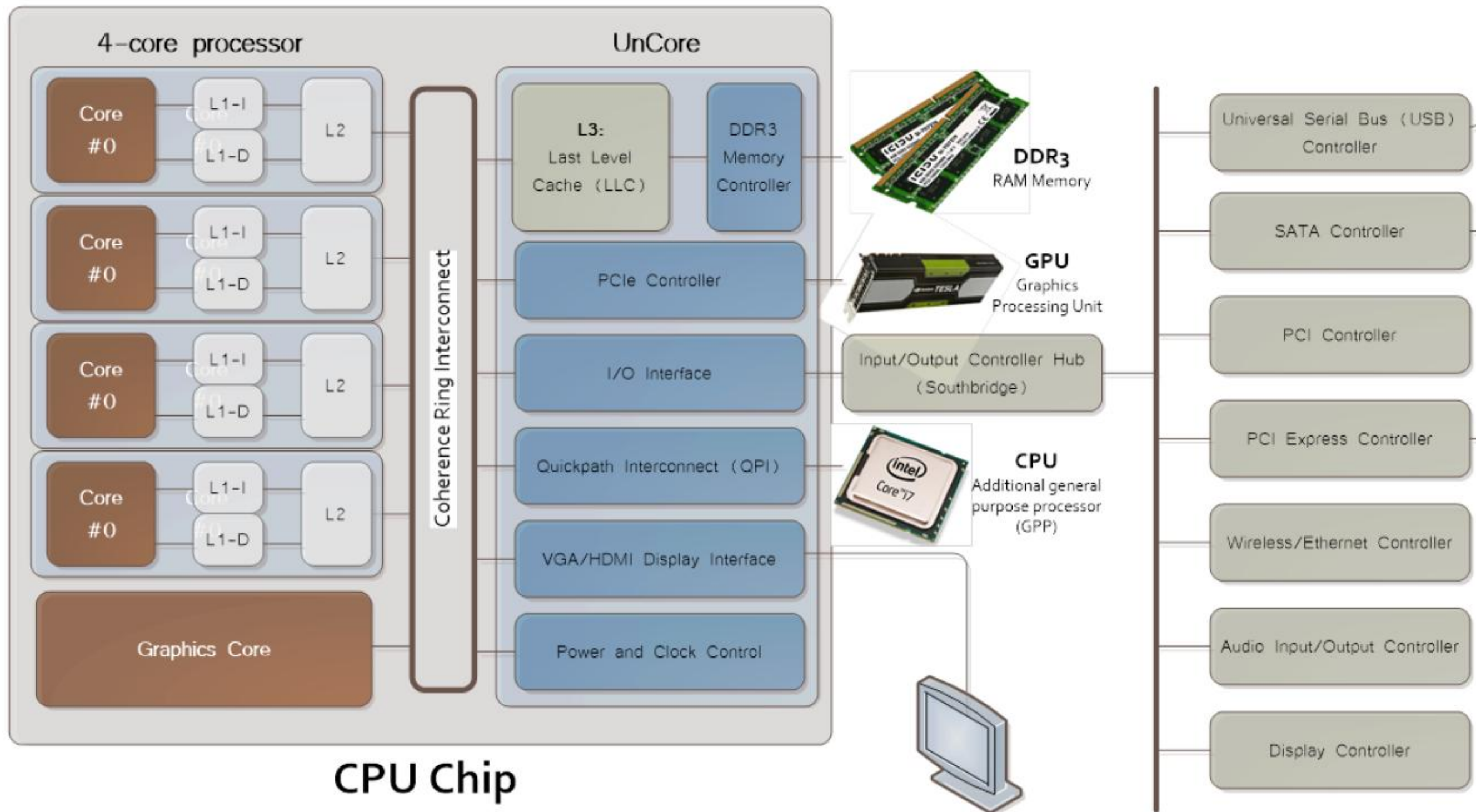


Esquema moderno de un Sistema de I/O



- ▶ La variante de AMD de la diapositiva anterior, también con PCH.
 - ▶ La principal diferencia es el soporte de PCIe 4.0 (en vez de 3.0), inclusive para la conexión entre el CPU y el PCH (en vez de DMI).
 - ▶ Además integra nuevas funciones al CPU (Audio, USB, etc).

Esquema de un *System On a Chip* (SOC)

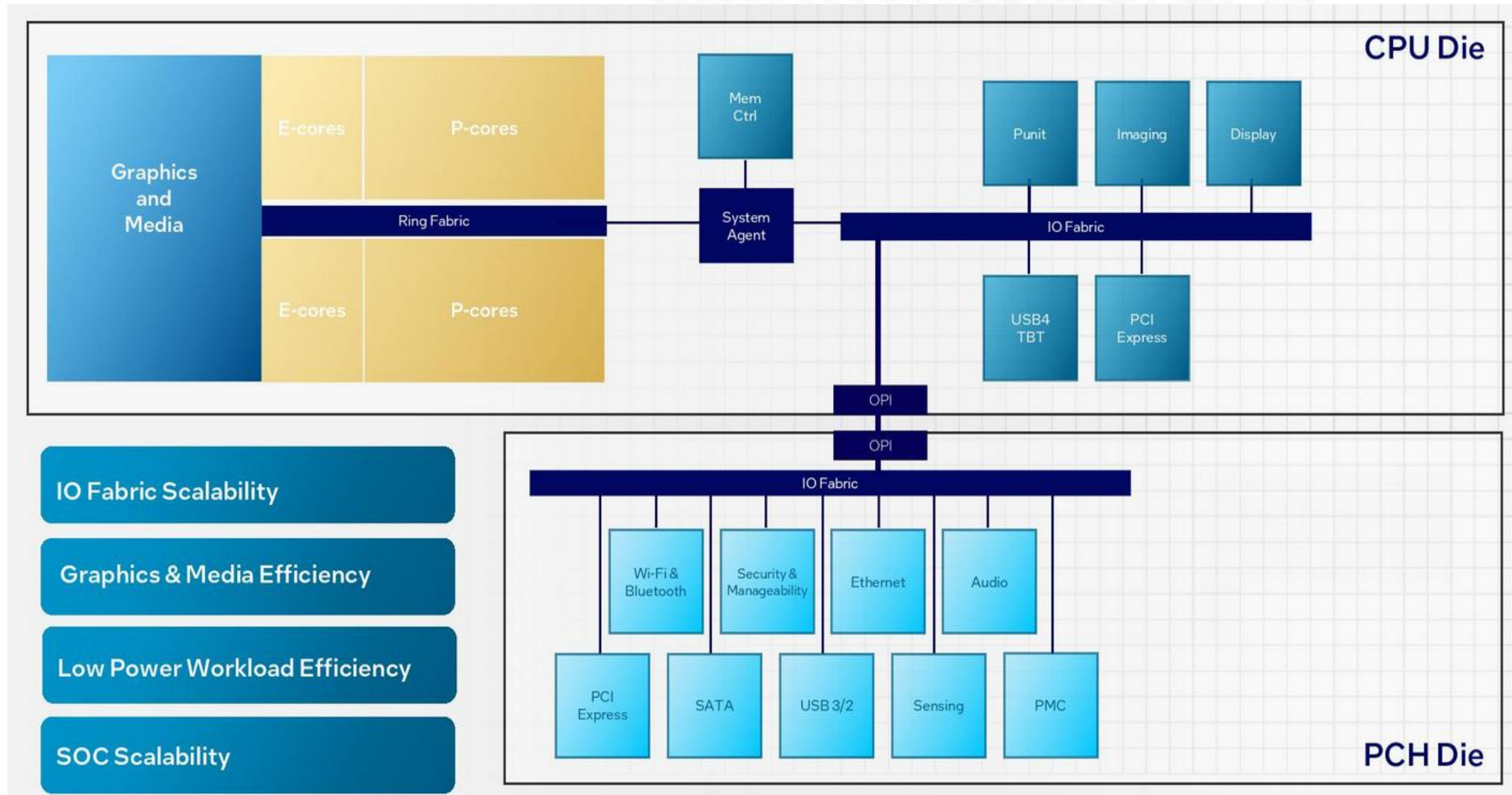


Esquema de un *System On a Chip* (SOC)

- ▶ En la diapositiva anterior podemos destacar:
 - ▶ Varios núcleos, cada uno con sus L1 y su L2.
 - ▶ Además un L3 común, compartido entre todos los núcleos (UMA).
 - ▶ Un núcleo adicional para manejo de gráficos, y muchos controladores.
 - ▶ Todos juntos conforman lo que se conoce como **uncore**: componentes de un procesador que no son núcleos.
 - ▶ Todo lo anterior conectado mediante un bus interno.
 - ▶ En el dibujo, una topología de anillo.
 - ▶ Todo en un solo chip (**System On a Chip**).
 - ▶ Una única interfaz genérica con la mayoría de dispositivos (el *Southbridge*), y algunas interfaces dedicadas.

Esquema de un *System On a Chip (SOC)*

► Intel 13ra generación:



Conexiones *on-chip*

- ▶ Muy populares en los procesadores más nuevos.
 - ▶ Poseen varios núcleos y otros componentes, todos interconectados entre sí (*System On a Chip*).
 - ▶ También conocidas como conexiones *chip-to-chip*, o *die-to-die*.
- ▶ Definen los controladores y capa física (PHY).
 - ▶ ¿Se acuerdan del Modelo OSI de Fundamentos de Redes?
- ▶ Suelen ser **paralelos**, con una topología que conecte todos contra todos.
 - ▶ Se busca minimizar la latencia (~10 ns).
- ▶ En general, son tecnologías propietarias, que dependen de cada fabricante.
 - ▶ Se genera un gran problema de compatibilidad.
 - ▶ AMD tiene su ***Infinity Fabric (IF)***, presentada en 2017.
 - ▶ Intel tiene su ***Advanced Interface Bus (AIB)***, y lo hizo gratuito.
 - ▶ ARM hizo su ***Advanced Microcontroller Bus Architecture (AMBA)***.

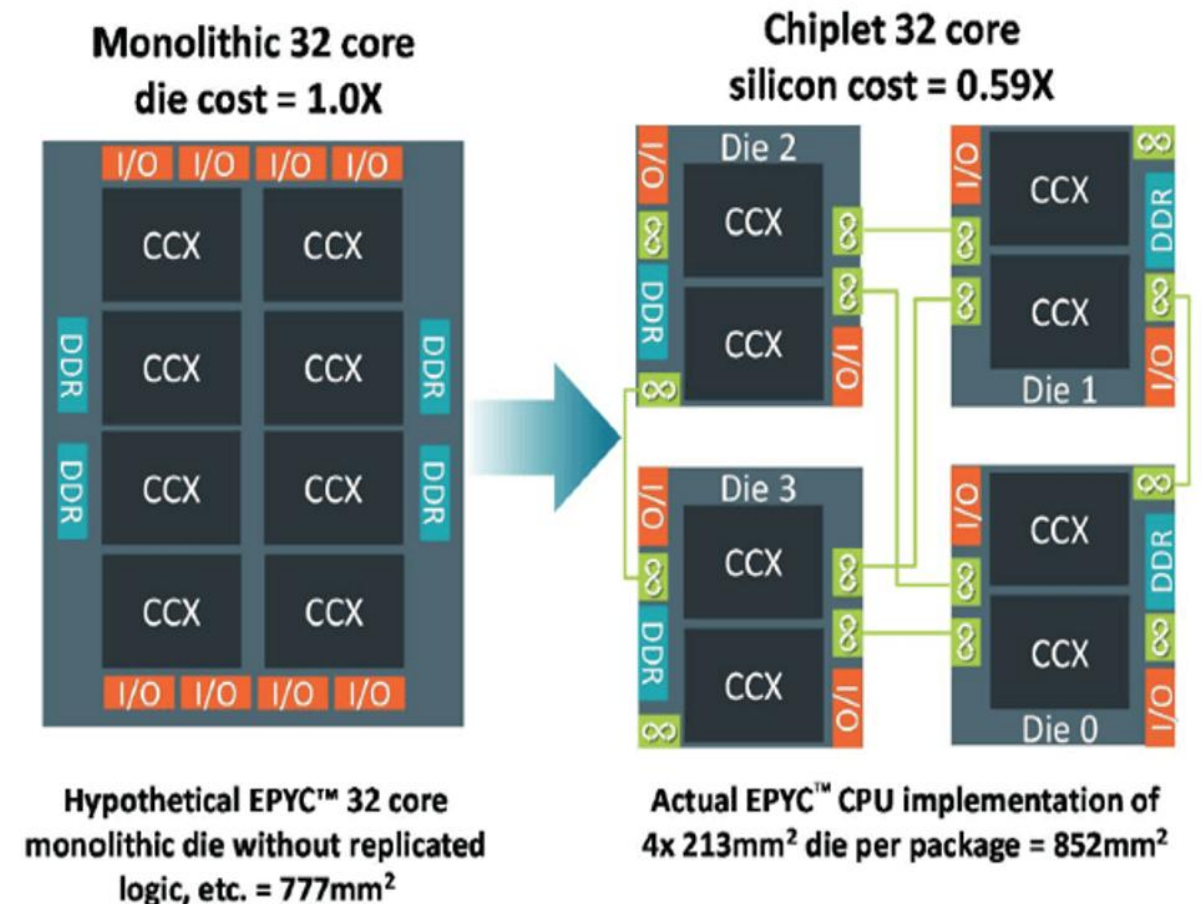
Repaso Tema 2: *Chipllets*

► Ventajas:

- **Mejoran el yield**, y la cantidad de chips por oblea, lo cual hace que **baje el costo**.
- Permiten reutilización.
- **Modularidad** (Tema 16).
- Mejoran la performance.
- **Mejoran el consumo de energía** (Tema 15).

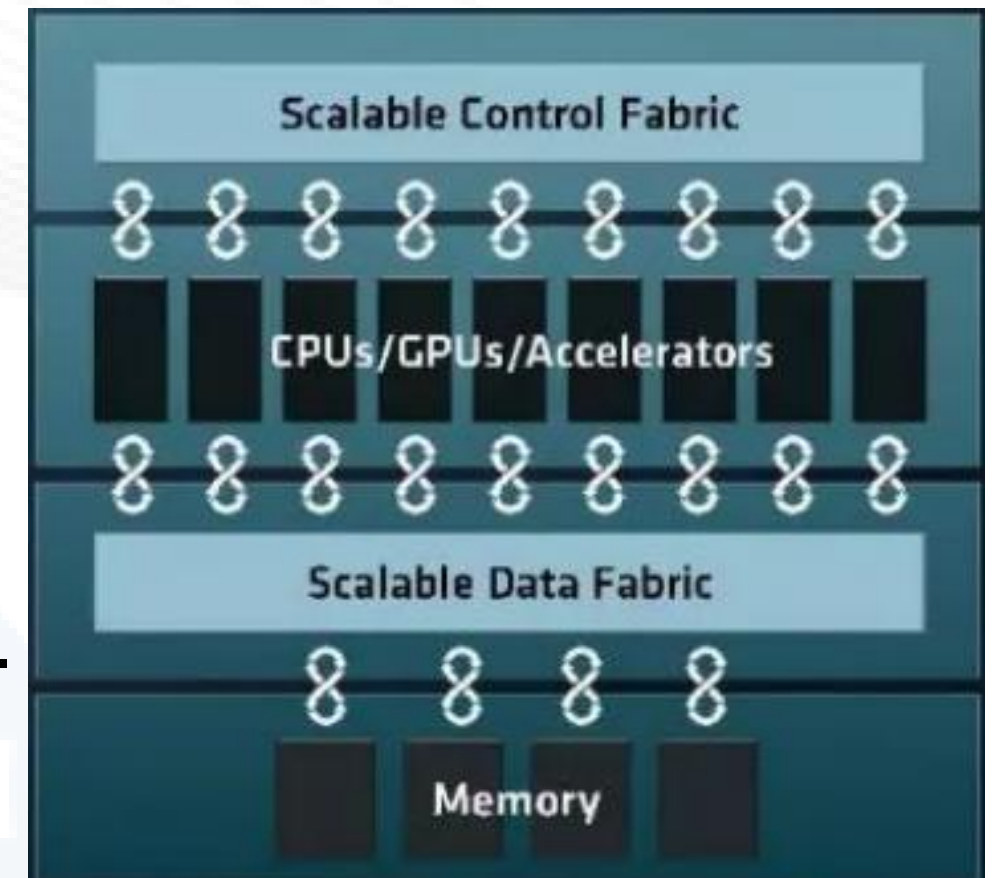
► Desventajas:

- Las interconexiones son propietarias (Tema 14).
- **Aumentan el área total**.
- Añaden más complejidad.



Infinity Fabric de AMD

- ▶ Criterio fundamental: **escalabilidad**.
 - ▶ No tiene limitación en cuanto a cantidad o tipo de dispositivos conectados.
 - ▶ No tiene limitaciones en cuanto a la topología de red.
 - ▶ Punto a punto, como bus o en una mesh.
- ▶ Sincrónico: posee su propio clk (fclk), independiente de la frecuencia del core.
 - ▶ Suele ser igual al clk de la memoria instalada (mclk).
- ▶ Dos planos de comunicación: SDF (Datos) y SCF (Control).
 - ▶ SDF define cómo fluyen los datos.
 - ▶ Interconecta los controladores de memoria, dispositivos PCIe, USB, SATA, Ethernet y otros periféricos (Southbridge).
 - ▶ SCF define el flujo de información para controlar el consumo de energía, aspectos de seguridad, tests, etc.
- ▶ Provee **coherencia** entre todos los dispositivos conectados.



AMBA de ARM

- ▶ **Estándar** abierto, introducido en 1996.
 - ▶ Con sucesivas versiones, hasta AMBA 5 del 2013.
- ▶ Es un conjunto de varios protocolos:
 - ▶ APB (*Advanced Peripheral Bus*): usado para dispositivos más lentos y/o simples.
 - ▶ AHB (*Advanced High performance Bus*): es usado en los diseños más pequeños.
 - ▶ **AXI (*Advanced eXtensible Interface*)**: usado en los de mayor performance.
 - ▶ Paralelo, síncronico, muy versátil.
 - ▶ Además están ASB, ATB, ACE, CHI, entre otros.
 - ▶ Algunos protocolos poseen una versión *Lite*, que simplifica algunas características.

CXL (*Compute eXpress Link*)

- ▶ **Estándar** abierto, del 2019, impulsado por la industria, diseñado principalmente para ser usado en datacenters.
- ▶ Está recibiendo muy buena aceptación en la industria, con nuevas versiones completamente compatibles: 2.0 (2020) y 3.0 (2022, ya sobre PCIe 6.0).
- ▶ Agrega una capa de funciones sobre la capa física de PCIe 5.0.
- ▶ Conecta múltiples CPU con múltiples dispositivos heterogéneos, todos bajo un mismo espacio de memoria unificado y coherente.
 - ▶ Por eso es que vimos algo adelantado en el tema anterior.
 - ▶ Maneja conflictos que podrían ocurrir cuando múltiples dispositivos realizan múltiples peticiones al bus a múltiples memorias distribuidas por todo el sistema.
 - ▶ Simplifica los accesos a memoria en sistemas heterogéneos.

CXL (*Compute eXpress Link*)

- ▶ Soporta tres protocolos diferentes: CXL.io (idéntico a PCIe), CXL.cache y CXL.mem.
 - ▶ CXL.mem es el que define el pool de memoria.
 - ▶ CXL.cache implementa MESI como protocolo de coherencia.
- ▶ Define tres tipos de dispositivos:
 - ▶ Tipo 1: aceleradores y dispositivos estándar PCIe sin memoria; soporta CXL.cache únicamente.
 - ▶ Tipo 3: dispositivos de memoria, para ampliación de capacidad y/o ancho de banda; soporta CXL.io y CXL.mem
 - ▶ Tipo 2: aceleradores con memoria (GPUs); soporta los tres protocolos.
 - ▶ Los dispositivos Tipo 2 y Tipo 3 exponen su memoria como un Nodo NUMA adicional.

NVLink

- ▶ Exclusivo de NVIDIA, anunciado en 2014.
- ▶ No es estándar sino propietario.
- ▶ 6ta generación (2026), tiene 3.6 TB/s para conectar sus GPUs.
 - ▶ 14x más rápido que PCIe 6.0.
 - ▶ 22.5x más rápido que la versión 1.0, en 12 años.
- ▶ Serial, de múltiples *lanes*, usa una topología mesh.
- ▶ Posee una extensión C2C para proveer **coherencia** entre los dispositivos conectados.

Universal Chiplet Interconnect Express (UCIe)

- ▶ Nuevo **estándar** para conexiones die-to-die.
- ▶ Propuesto por líderes de la industria.
 - ▶ Intel, AMD, ARM, NVIDIA, Samsung, Qualcomm, IBM, TSMC, Marvell, Google, Microsoft, Meta, entre otros 140 miembros.
 - ▶ Curiosamente, sin Apple ni RISC-V.
- ▶ Versión 1.0 en 2022, 2.0 en 2024 y 3.0 en 2025.
- ▶ *Open source*, para disminuir costos.
- ▶ Basado en buses existentes: PCIe y CXL (no reinventar la rueda).
 - ▶ Define capa física y capa de protocolo.
- ▶ Objetivos:
 - ▶ Ampliar el ecosistema de *chiplets* disponibles, brindando **compatibilidad**.
 - ▶ Los primeros *chiplets* compatibles salieron a fines de 2025.
 - ▶ Convertirse en universal, como USB o PCIe.
 - ▶ Presentar una **latencia menor que 2 ns**.
 - ▶ Ser muy personalizable y adaptable. También para conexiones *off-chip*.

Conexiones de gran escala

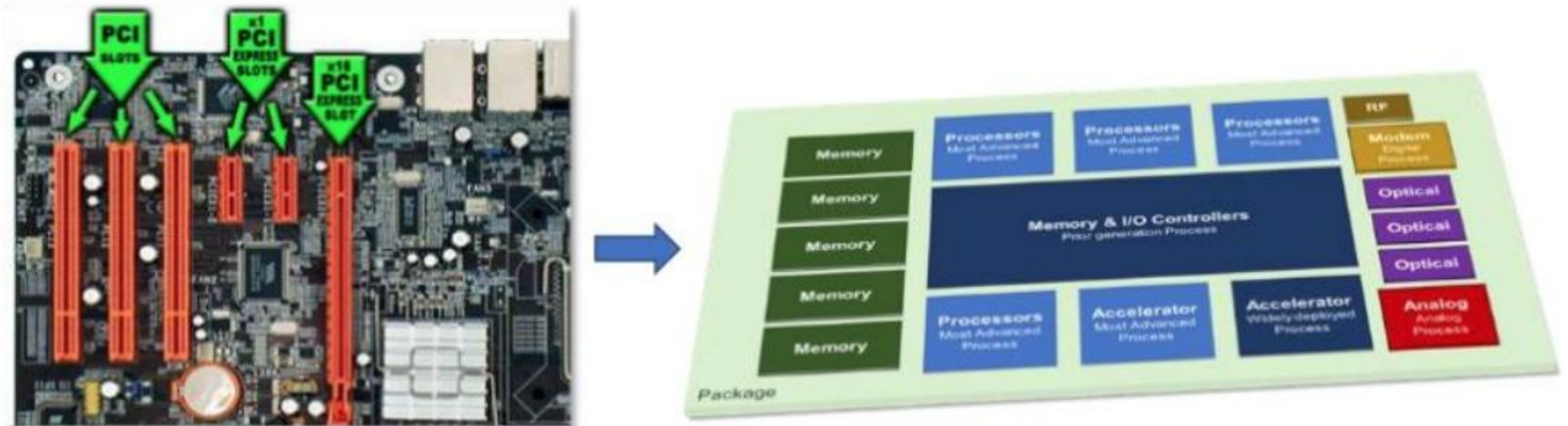
- ▶ Las supercomputadoras o los datacenters de gran escala necesitan además otro tipo de conexiones.
 - ▶ Los dispositivos que conectan están inherentemente más distanciados.
- ▶ Necesitan baja latencia y mucha escalabilidad.
 - ▶ *¿Las anteriores no eran de baja latencia?*
 - ▶ Usan además otras métricas de calidad de servicio.
- ▶ Basadas en topologías tipo *crossbar*, que evolucionan en complejos toroides 3D.
- ▶ Hay varios tipos de interconexión, siendo la más popular **Infiniband**.
 - ▶ Suelen ser específicamente diseñadas para cada instalación.
- ▶ Aunque cada vez más se va imponiendo la simpleza de **Ethernet** (porque va mejorando su ancho de banda).
 - ▶ Top500: 48% Ethernet + 34% Infiniband.
 - ▶ Top10: 0% Ethernet + 70% Infiniband.
 - ▶ Además, Infiniband es 3-4x más costoso que Ethernet.
- ▶ Mucha investigación actual en esta área, porque es un diferenciador entre grandes empresas (Google, Amazon, Facebook, etc).

Tendencias actuales sobre buses

- ▶ A medida que aumentan los requerimientos de performance sobre los buses (mayor ancho de banda y/o menor latencia), van apareciendo nuevas versiones de buses existentes o nuevos buses.
- ▶ Nuevos protocolos van agregando funciones adicionales a los protocolos existentes, para ampliar sus capacidades.
 - ▶ Sin tener que reinventar lo que está probado que funciona.
- ▶ Sin embargo, las comunicaciones van evolucionando más lentamente que el poder de cómputo.
 - ▶ Se están convirtiendo en un cuello de botella.
 - ▶ Limitando la escalabilidad y la performance del sistema.

Tendencias actuales sobre buses

- ▶ A medida que aumenta la capacidad de integración de los chips, más componentes se van moviendo desde fuera hacia dentro del chip (*uncore*).
 - ▶ Y ello implica que las conexiones también se van moviendo dentro del chip.
 - ▶ Se pasa de sistemas compuestos por chips a sistemas dentro de un chip (SoC).



Tendencias actuales sobre buses

- ▶ Se están integrando conexiones ópticas, tanto dentro del *package* como *near-package*
 - ▶ Para superar las limitaciones propias de las conexiones por cobre.
 - ▶ Alcance, energía/bit transmitido, densidad de conexiones.
 - ▶ También conocido como *Co-Package Optics* (CPO).
- ▶ Se propone que en el futuro las conexiones sean:
 - ▶ UCle sobre PCI-Express conectando todos los chiplets dentro de un mismo chip.
 - ▶ CXL sobre PCI-Express conectando CPUs, aceleradores, memoria y flash dentro de un mismo nodo.
 - ▶ PCI-Express conectando nodos dentro un rack.
 - ▶ Ethernet o Infiniband conectando los racks con el mundo exterior.

Resumen de buses vistos (y extras)

Tipo	Interconexión	Bandwidth	Tipo	Interconexión	Bandwidth
Chip-to chip	NVLink 6.0	3600 GB/s	Chip-to chip	CXL 1.0	64 GB/s
Chip-to chip	NVLink 5.0	1800 GB/s	Chip-to chip	PCIe 4.0 x16	64 GB/s
On-chip	HBM4	1600 GB/s	Chip-to chip	400 Gb Ethernet	50 GB/s
On-chip	HBM3e	1024 GB/s	Off-chip	DDR5-3200	51,2 GB/s
Chip-to chip	NVLink 4.0	900 GB/s	Off-chip	DDR4-1600	25,6 GB/s
On-chip	HBM3	819 GB/s	Off-chip (I/O)	HDMI 2.0	2,25 GB/s
Chip-to chip	NVLink 3.0	600 GB/s	Off-chip (I/O)	USB 3.1	1,21 GB/s
On-chip	Infinity Fabric (máx)	512 GB/s	Off-chip (I/O)	SATA 3.0	600 MB/s
Chip-to chip	PCIe 7.0 x16	512 GB/s	Off-chip (I/O)	SDHC (UHS-II)	156 MB/s
On-chip	HBM2e	460 GB/s	Off-chip (I/O)	IEEE 802.11n (Wi-Fi 4)	75 MB/s
Chip-to chip	NVLink 2.0	300 GB/s	Off-chip (I/O)	Bluetooth 5.0	6,25 MB/s
Chip-to chip	PCIe 5.0 x16	256 GB/s			
Chip-to chip	NVLink 1.0	160 GB/s			
Chip-to chip	PCIe 5.0 x16	128 GB/s			
Chip-to chip	CXL 3.0	128 GB/s			
Chip-to chip	Infiniband HDR 12x	75 GB/s			

Conclusiones generales

- ▶ *“Anyone can build a fast CPU. The trick is to build a fast system.”*
Seymour Cray
- ▶ Un sistema de computación está compuesto por diferentes subsistemas.
 - ▶ Procesador, sistema de memoria, sistema de I/O, y software (programa de usuario, drivers del SO) deberían estar orquestados para maximizar performance.
 - ▶ Un universo de protocolos e interconexiones para conectarlos, cumpliendo requerimientos y limitados por “las tres P”.
- ▶ Es crítico que la configuración general del sistema **valide todos los requerimientos del problema que se pretende resolver.**
 - ▶ Latencias mínimas, tiempos de respuesta, cantidad de dispositivos, etc.
- ▶ Es crítico **analizar la existencia de cuellos de botellas** entre los distintos subsistemas.
 - ▶ El cuello de botella nos va a determinar la máxima productividad de todo el sistema.

Resumen final

- ▶ Los buses son el medio a través del cual se conectan los distintos componentes de una computadora.
 - ▶ No son solamente un conjunto de cables. Definidos por varias capas.
 - ▶ Basados en estándares, que aseguran compatibilidad, efectividad y eficiencia.
 - ▶ Organizados jerárquicamente, conectados entre sí mediante *bridges*.
 - ▶ Requieren alguna especie de arbitraje.
- ▶ Métricas más importantes: ancho de banda y latencia.
 - ▶ También escalabilidad, tolerancia a fallos y eficiencia energética.
- ▶ Diversos tipos, diversas clasificaciones, diversas topologías.
- ▶ Conexiones off-chip: se imponen los buses seriales, sincrónicos, punto a punto.
- ▶ Los componentes y los buses se van moviendo hacia dentro del chip (SoC).
- ▶ Conexiones on-chip: paralelas, muy baja latencia, propietarias.
- ▶ Importante considerar el sistema como un conjunto completo.
 - ▶ Validar requerimientos mínimos, encontrar posibles cuellos de botella.