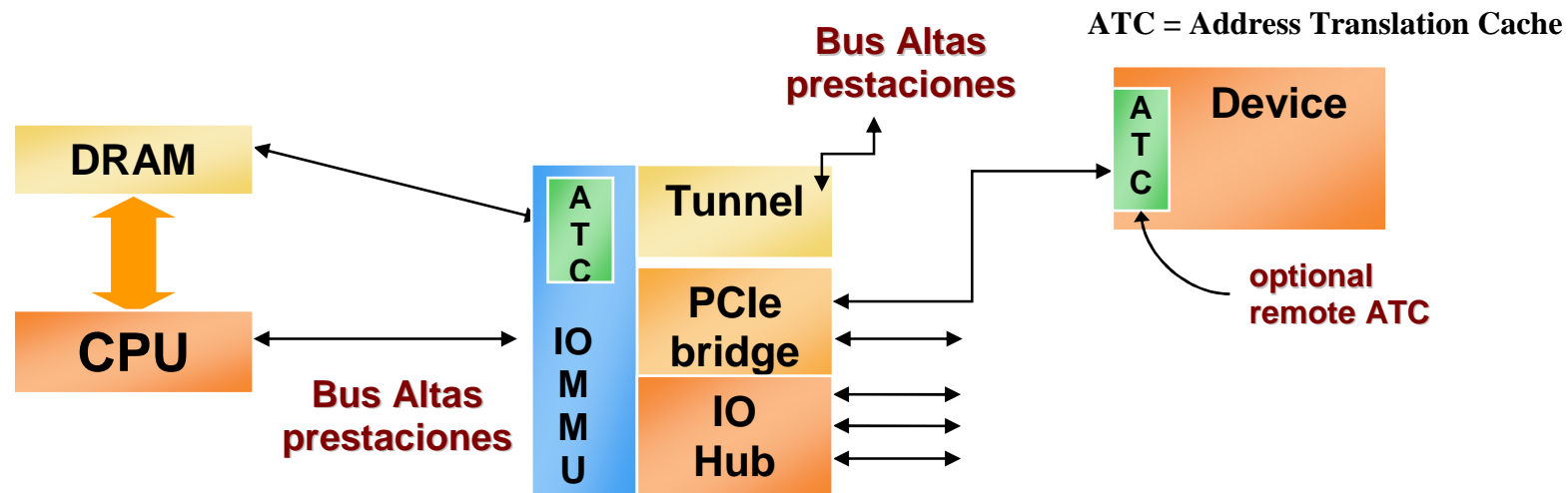


## TEMA 3: Hardware de gestión de memoria y E/S

### Contenidos

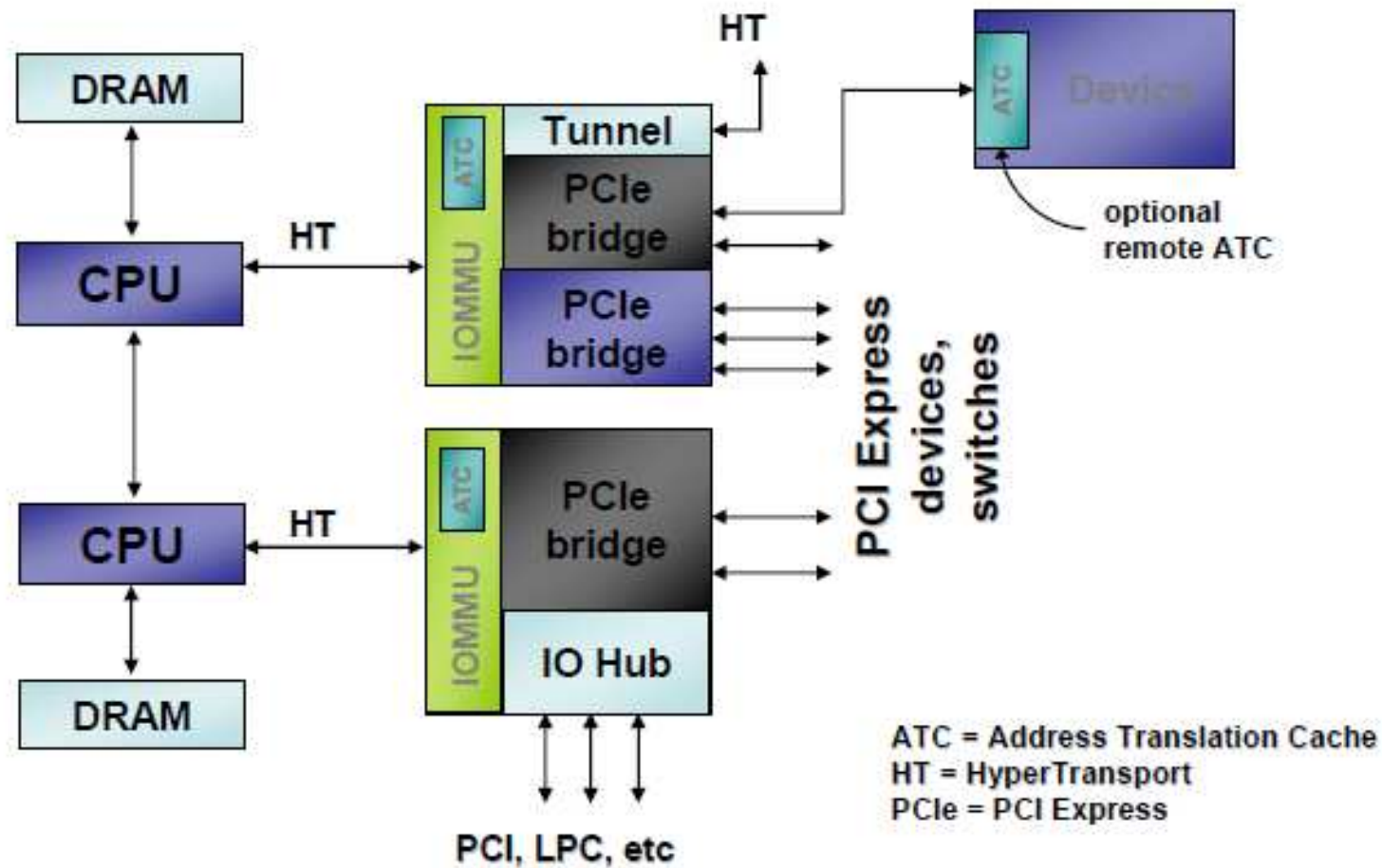
1. Gestión de memoria. Introducción
2. Memoria Virtual. Paginación, Segmentación, Seg con paginación.
3. Gestión de E/S. DMA.
4. DMA. Gestión de tareas
5. DMA. Estructura Hardware
6. DMA. Secuencia de funcionamiento
7. IOMMU. Funcionamiento, Características, Traducción de estructuras, Interface con el Software

## 7. IOMMU: Unidad de Manejo de Memoria y entrada-Salida



- Traduce las solicitudes que vienen de todos dispositivos independientemente de quien las realiza
- Hablar de IOMMU es hablar del corazón de los Chipsets

- Aplica los derechos de acceso de los dispositivos al espacio de direcciones del sistema
  - Protección granular página
  - Atiende a los derechos de acceso de lectura y escritura
- Mantiene la memoria caché de las traducciones (**ATC**). Similar a la paginación y segmentación de la memoria
- Actúa como Raíz del árbol de jerarquía de distribuida de cachés para la traducción de direcciones



Ejemplo AMD

IOMMU conecta las líneas de comunicación externa del sistema. Permiten una identificación completa de la procedencia/destino del dato de E/S

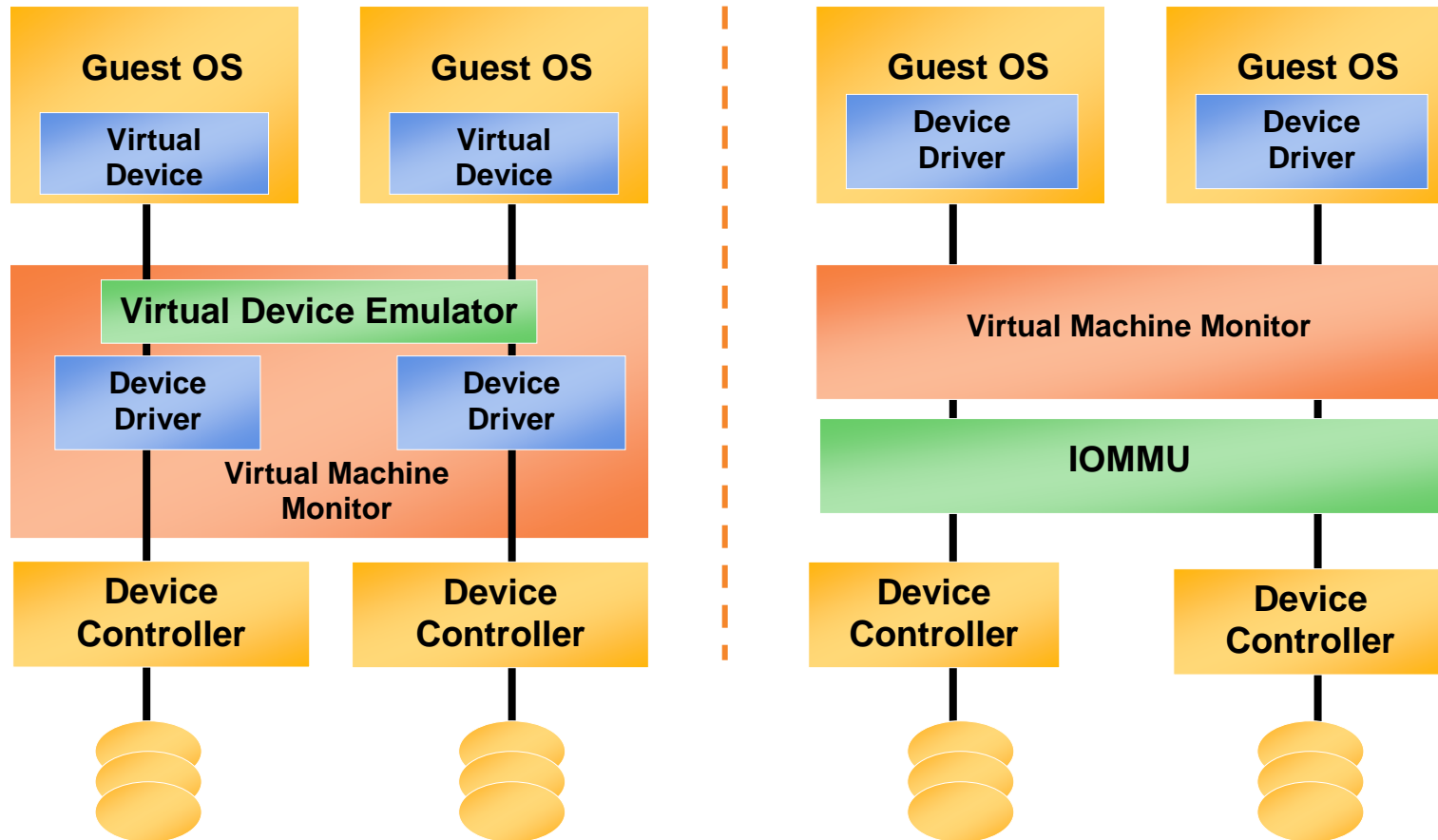
IOMMU permiten configuraciones flexibles e independientes. Crean estructuras escalables de almacenamiento en caché.

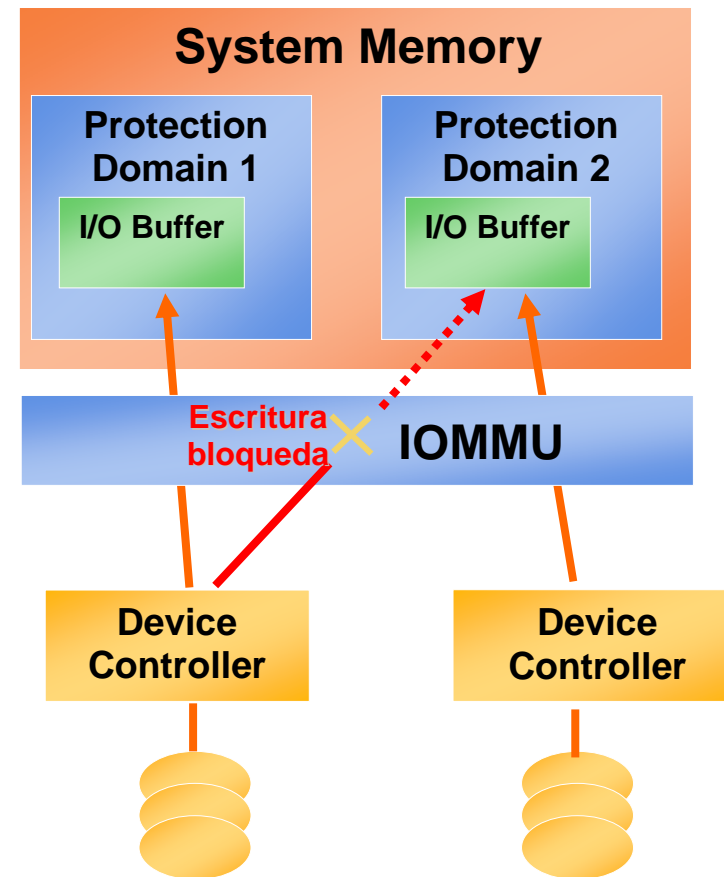
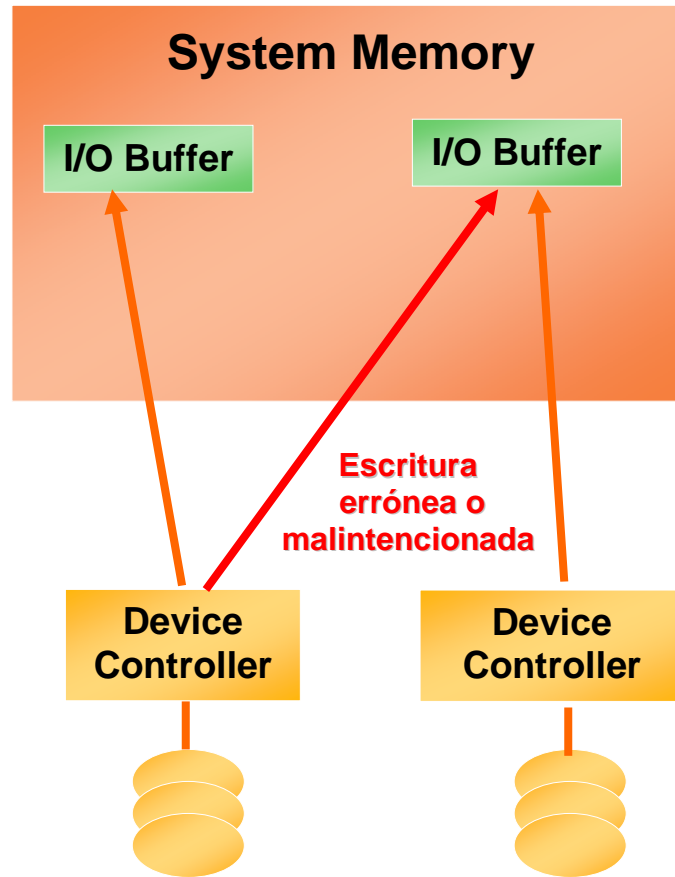
IOMMU soporta direccionamiento remoto con almacenamiento en caché de traducción. Permiten ajustar la jerarquía de almacenamiento en cachés

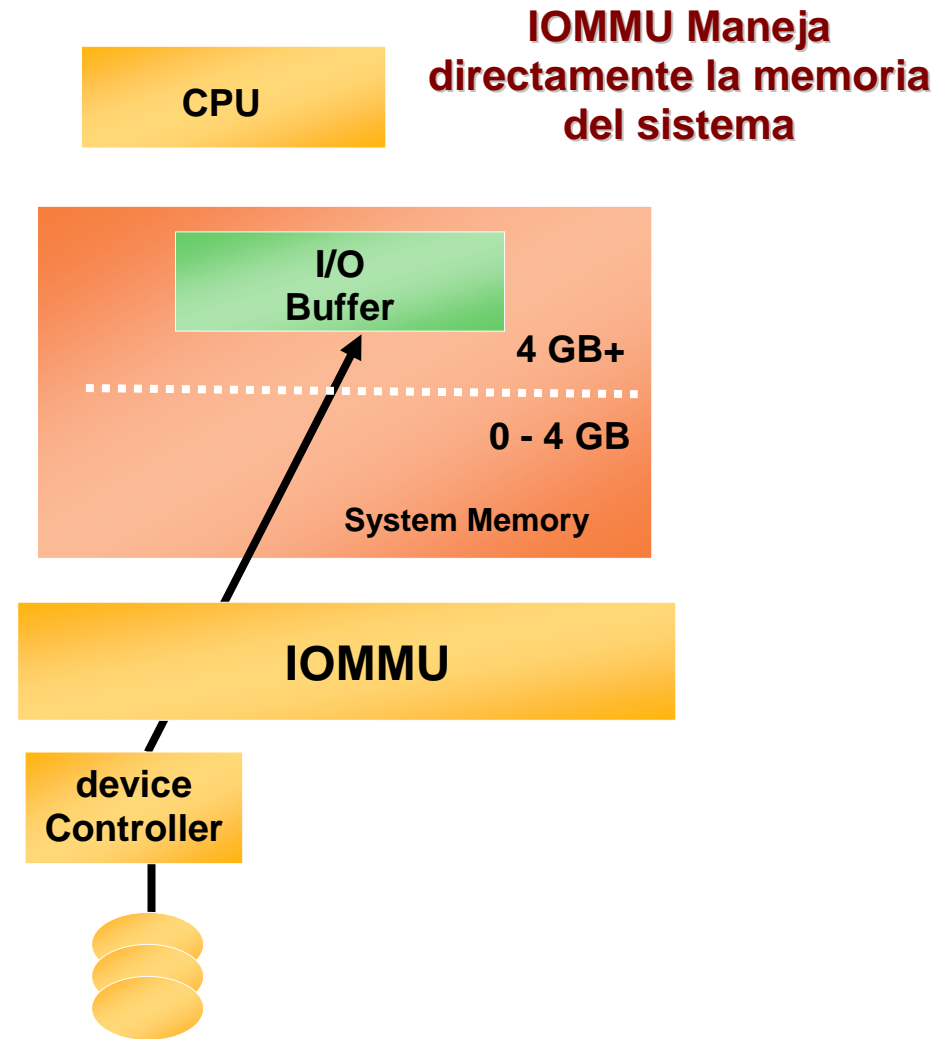
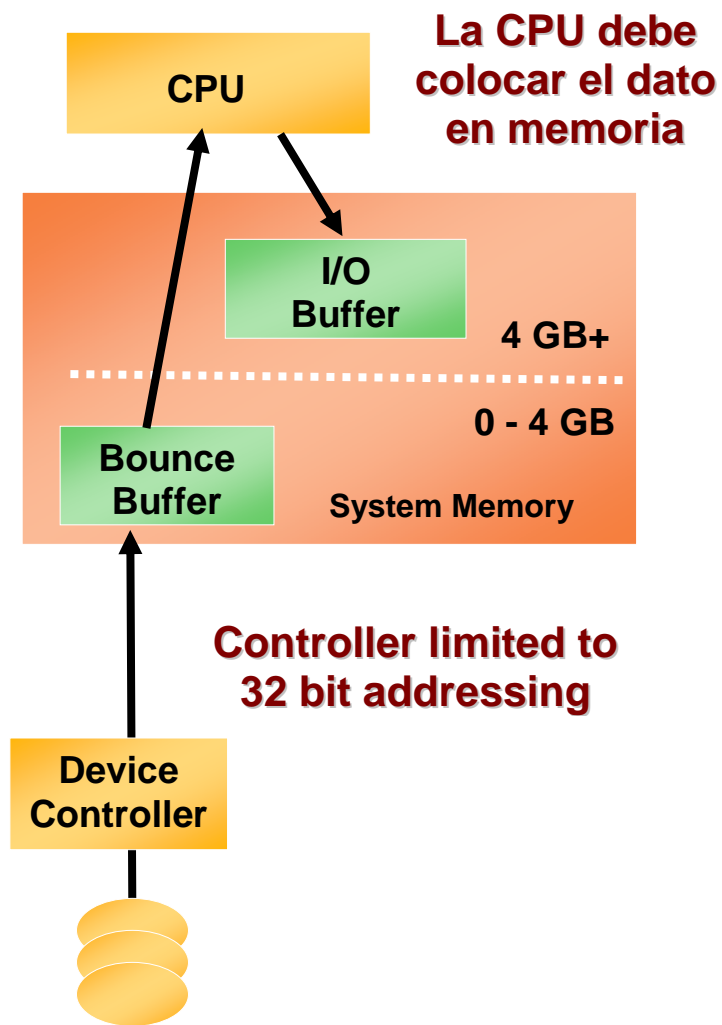
Mejora de las capacidades de virtualización con asignación directa de dispositivos desde el S.O. Permite el acceso directo a las aplicaciones del dispositivo, en modo de usuario.



## Comparación del funcionamiento de IOMMU









## **¿Cómo funciona una IOMMU?**

- Asigna un rango variable de direccionamiento virtual por cada dispositivo
- Genera una página de tamaño variable por cada dispositivo físico
- Crea un espacio de direccionamiento virtual flexible, con capacidad de compartirse
  - Los dispositivos pueden tener su propio espacio de direccionamiento virtual
  - Los dispositivos pueden compartir su espacio de direccionamiento virtual

## **Definiciones y etiquetas**

- Requester ID (**RID**) etiqueta que identifica el dispositivo fuente/destino de la transferencia
- Address Translation Cache (**ATC**). Copia coherente, local o remota, de las direcciones a través de las cuales se va a realizar la transferencia de E/S
- I/O Translation Look aside Buffer (**IOTLB**) un ATC remoto que está localizado en el dispositivo asociado con el IOMMU
- Address Translation Services (**ATS**) Extensiones (etiquetas) que soportan el caché de direcciones de las transferencias

- Page Directory Entry (**PDE**). Dirección de la tabla de traducción que apunta a una tabla asociada a un dispositivo (diccionario)
- Page Table Entry (**PTE**). Dirección de la tabla de traducción que contienen las traducciones entre direcciones físicas y virtuales
- Root translation table (**RTT**) Tabla de traducción que se sitúa en la parte superior de la jerarquía de E/S
- Device Table (**DT**) Dirección del mapa de ID que señala la tabla de traducción.

## **Petición de los dispositivos**

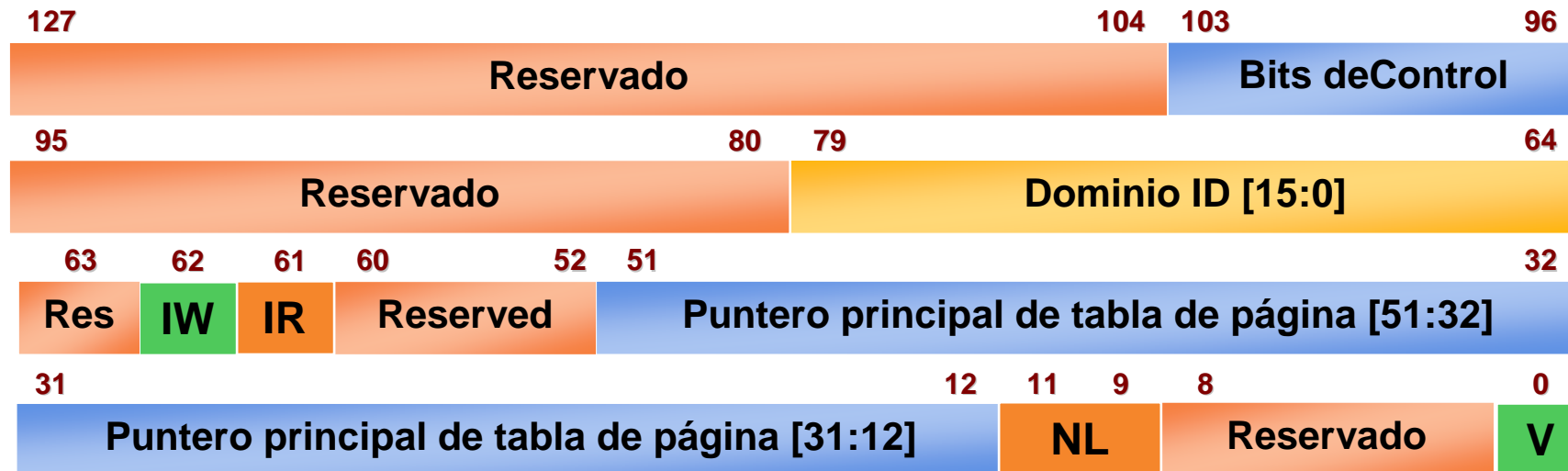
- Contenido de una petición de ID :
  - Bus/Dispositivo/Función (**BDF**) usado para PCI/IOHUB
  - Unidad de ID o BDF usado por PCI
- Extensiones para utilizar una ATC remota
  - Dirección virtual del dispositivo (Lectura o escritura)
    - Por defecto
    - La IOMMU trasladará la dirección de la petición
  - Dirección real (Dirección física) lectura o Escritura. Se puede utilizar la dirección sin necesidad de traducir.
  - Si fuese necesario, se requiere la traducción

## Tabla de dispositivos

- Es una zona de memoria del sistema
- Mapea (direcciona) los dispositivos que requieren acceso de E/S por medio de una tabla de traducción
- Soporta un espacio de direcciones virtuales por cada dispositivo
- A cada dispositivo se le asigna un identificador de dominio (**DID**) Domain ID
- Los dispositivos comparten su DID, cuando es necesario
- La IOMMU gestiona los acceso no válidos a los dominios de los dispositivos



## Ejemplo Tabla de dispositivos (AMD)

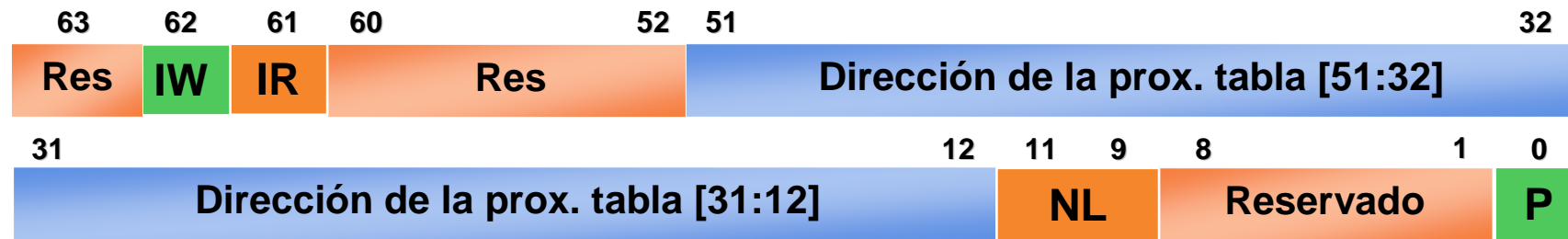


V – bit de paridad  
IW – I/O protección de escritura  
IR – I/O protección de lectura  
NL – índice de nivel adicional  
Res - reservado

## Tabla de páginas

- Las tablas de traducción de página son siempre espacios de tamaño definido (p. ej. 4K) dentro de la memoria del sistema
- La dirección base de la tabla de traducción proviene de la tabla de dispositivos. Puede apuntar a una tabla de PDE o PTE
- Intermedia entre las tablas de traducción. apuntar a una tabla de PDE (directorio) o PTE (entrada).

## Ejemplo Tabla de páginas (AMD)



### Formato PDE



### Formato PTE

IW – protección de escritura  
IR – Protección de lectura  
NL – índice de nivel adicional  
P – paridad

S – tamaño  
U – Bit de atributo ATS  
NS – bit de atributo ATS  
Res - reservado

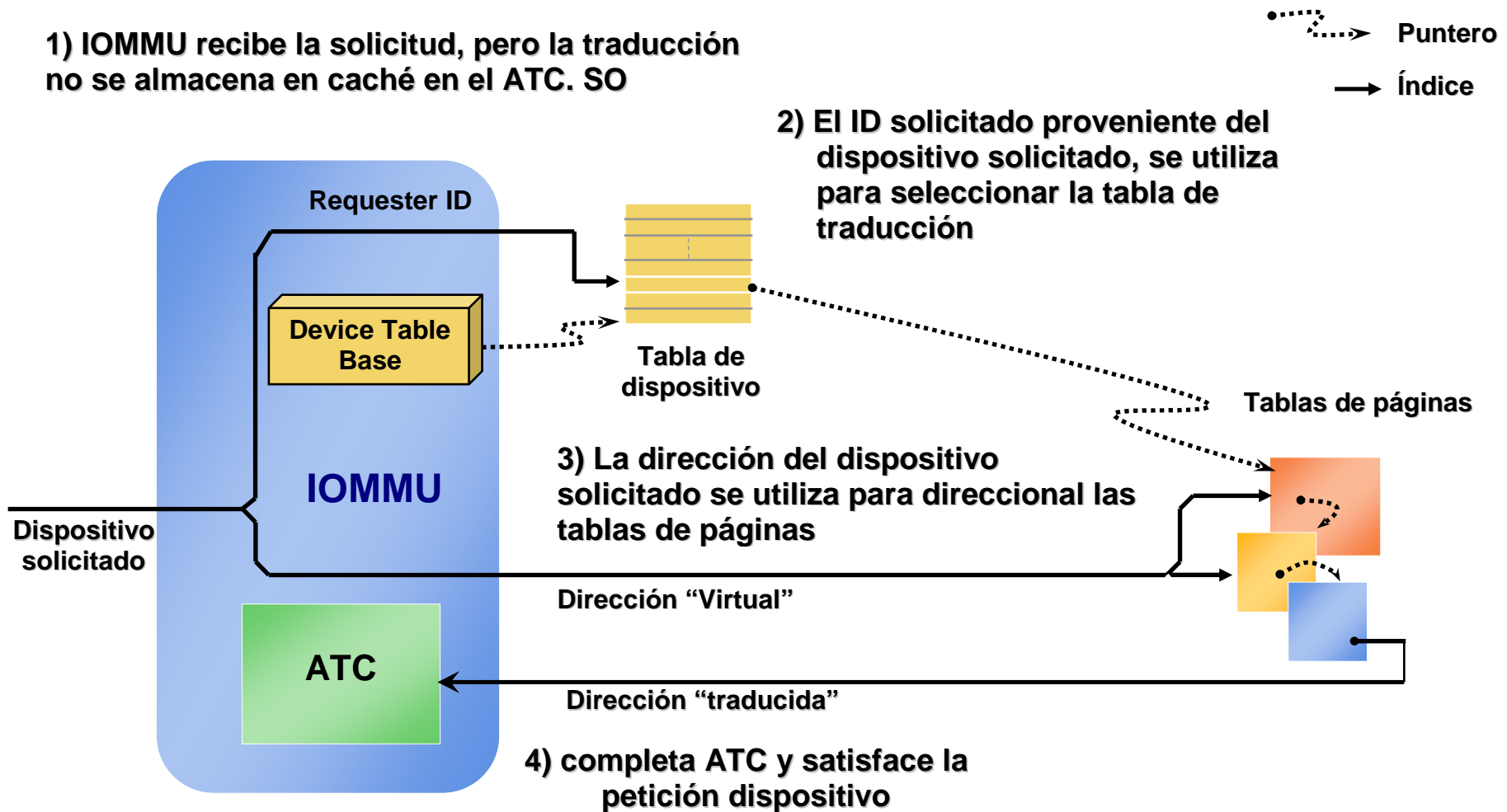
# Funcionamiento

1) IOMMU recibe la solicitud, pero la traducción no se almacena en caché en el ATC. SO

2) El ID solicitado proveniente del dispositivo solicitado, se utiliza para seleccionar la tabla de traducción

3) La dirección del dispositivo solicitado se utiliza para direccionar las tablas de páginas

4) completa ATC y satisface la petición dispositivo



## Tareas a realizar:

- Comparar la estructura genérica de IOMMU con la estructura de páginas de paginación/segmentación que se vieron hace dos clases. ¿Qué hay similar y qué hay de diferente?
- ¿Qué mejoras introducirías para que la estructura de tablas de seg/pag se pudieran transformar en una IOMMU?
- ¿Qué elementos digitales nos harán falta para esta transformación? Dibujar un esquema de bloques.