

# QoS / TCs / VCs y arbitraje en la capa de transacción PCIe

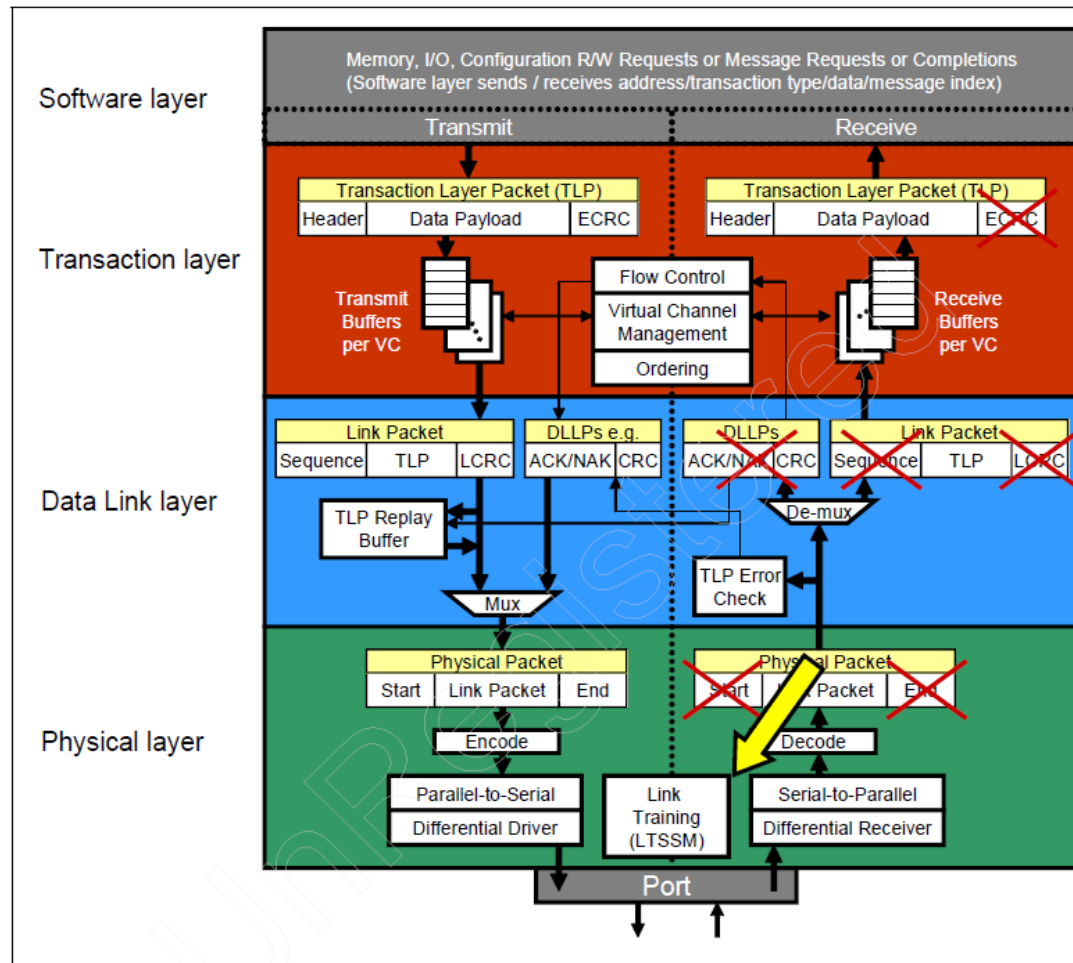
Adaptación de la arquitectura para el proyecto de diseño #2

Prof. Jorge Soto

IE-0523 Circuitos Digitales II

# Diagrama de capas PCIE

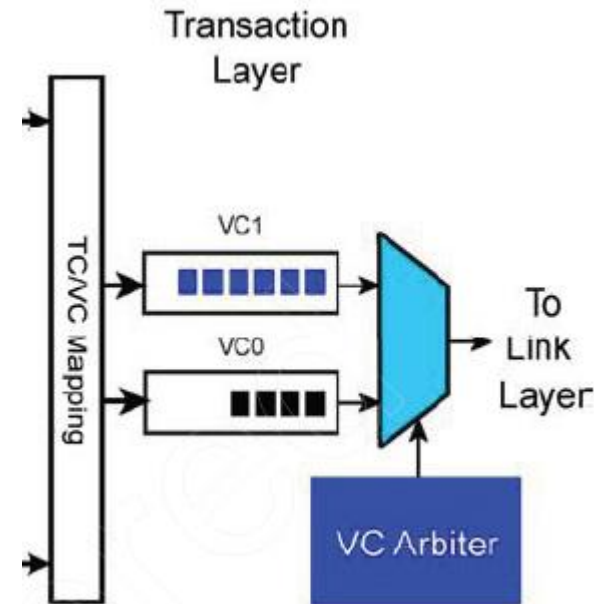
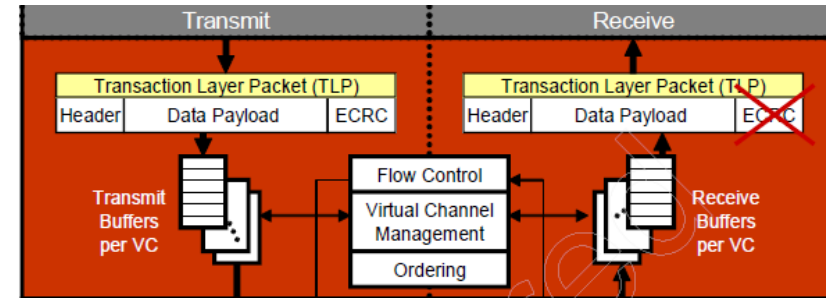
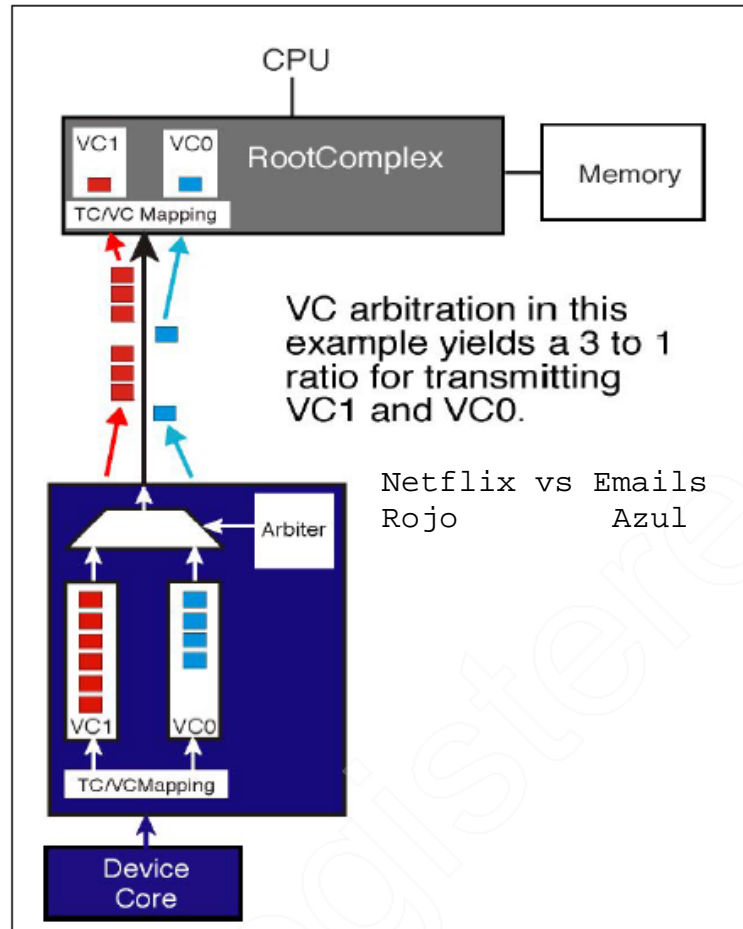
Figure 14-1: Link Training and Status State Machine Location



- Calidad de Servicio (QoS) opera en la capa de transacción
- Por lo general, cada Canal Virtual (VC) tiene una Clase de Tráfico (TC) asociada

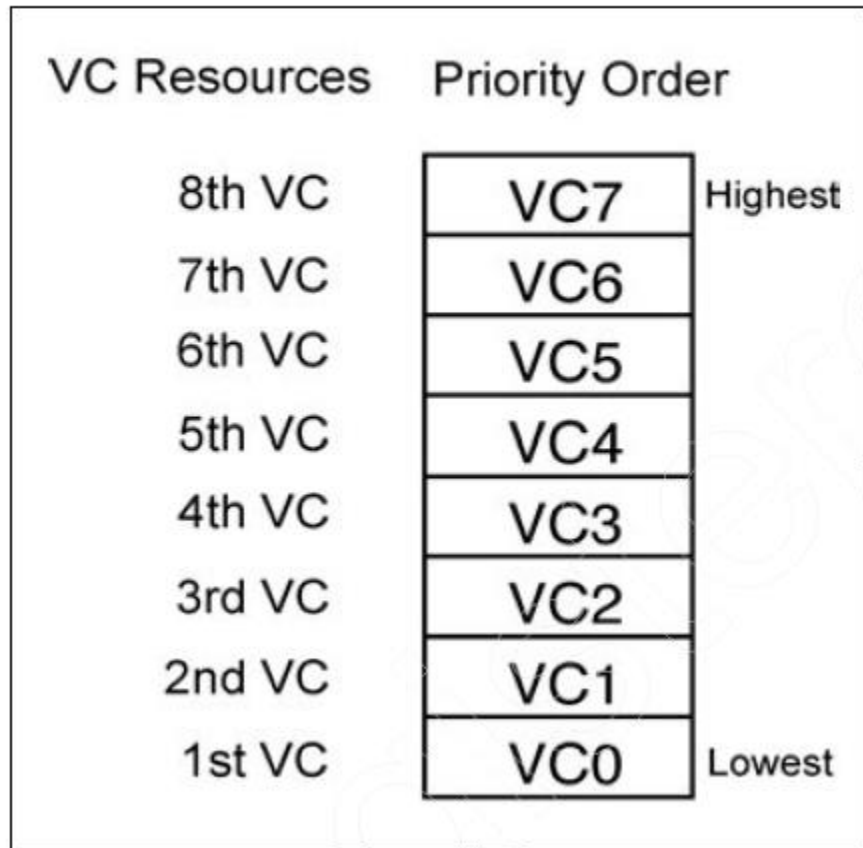
# Arbitraje en los canales virtuales (VC)

Figure 6-7: Conceptual VC Arbitration Example

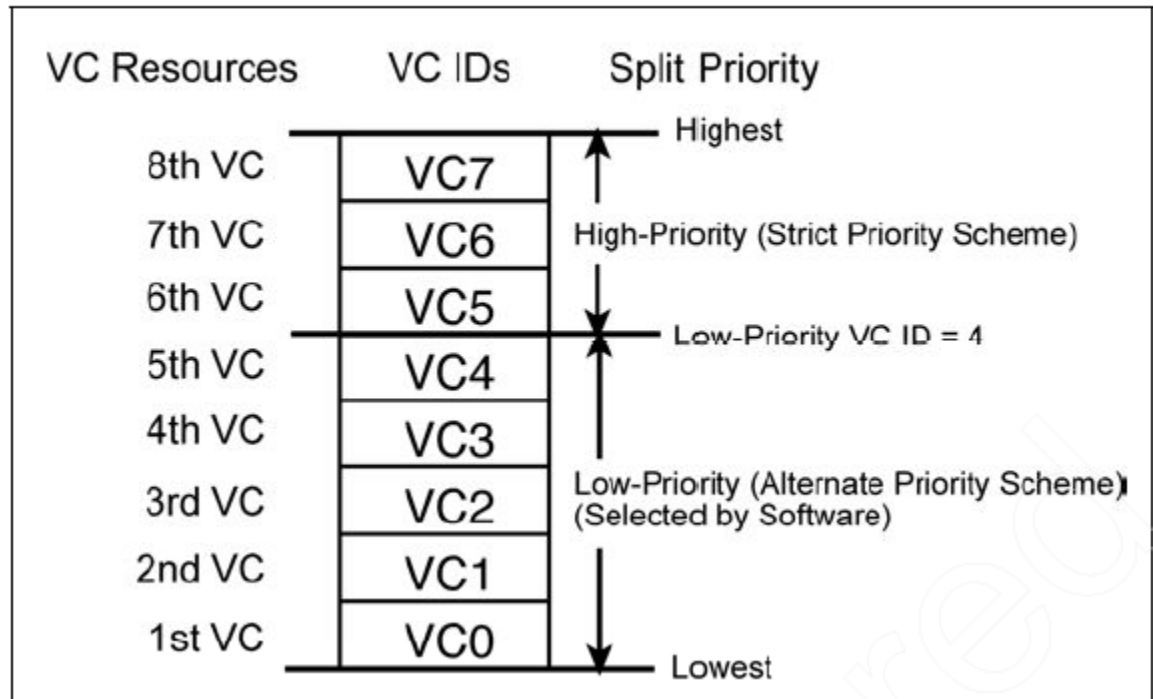


# Arbitraje en los canales virtuales

*Figure 6-8: Strict Arbitration Priority*



*Figure 6-11: VC Arbitration with Low- and High-Priority Implementations*

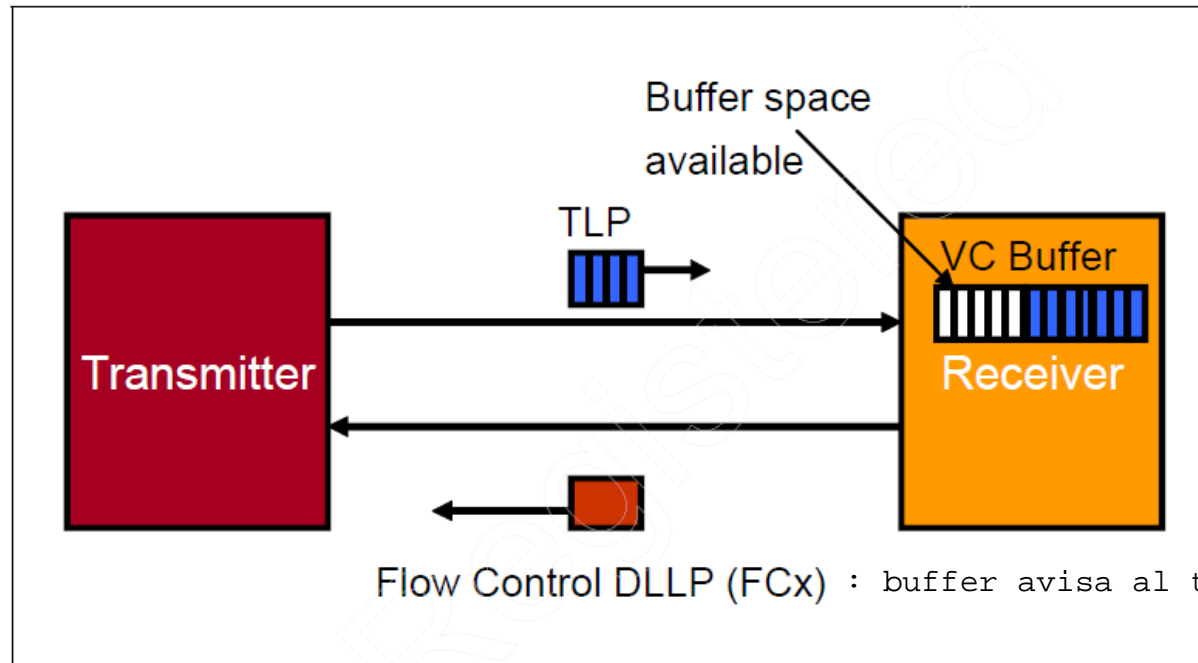


# “Flow control” en los canales virtuales

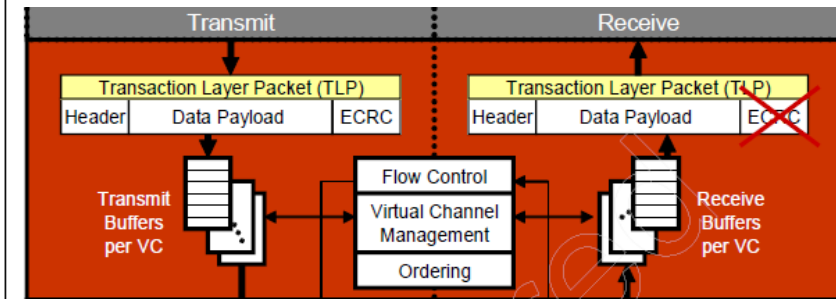
Vamos a hacer 4 canales de baja prioridad

Priority Flow control: se manda pausa pero solo a 1 de los Virtual Channels.

Figure 2-21: Flow Control Process



Ejemplo de Ramon: que mande pause al 75% de lleno y que mande continue cuando se vacia al 25%



Flow Control DLLP (FCx) : buffer avisa al transmitter que se detenga por que ya se esta llenando

Cada uno de los 4 buffers va a tener su propio Flow Control.

Adaptación de la capa de transacción para el proyecto de diseño #2, IE-0523.

# Módulos misceláneos para el proyecto de diseño #2

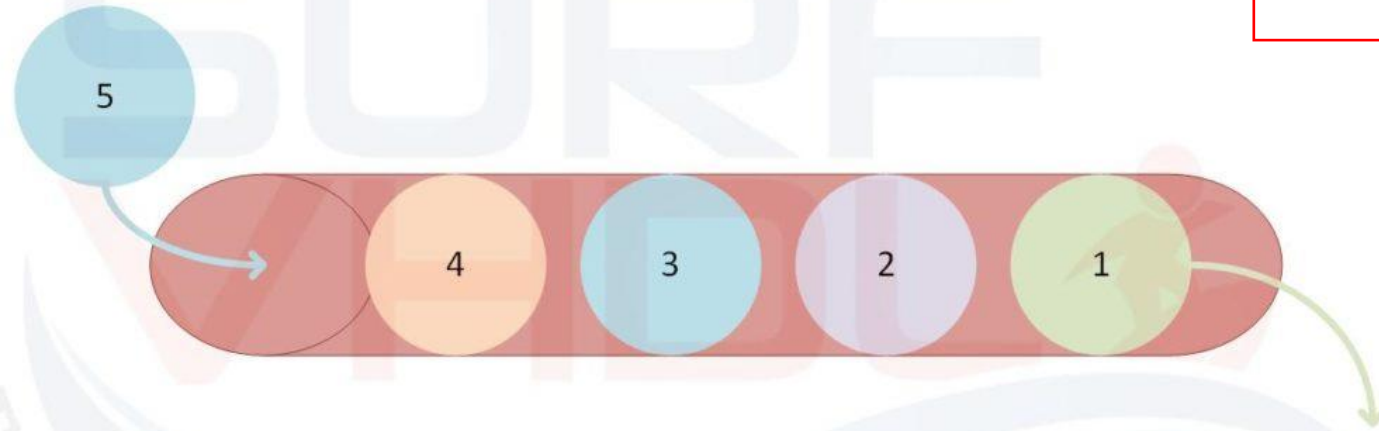
Audio: 8:00

# Buffer/FIFO

Cada FIFO debe sacar las señales:  
Full, Empty, Error

Error: estamos Full, y hacemos escritura sin haber hecho lectura (hay corrupcion de datos)  
AlmostFull: datos exceden X% o n elementos  
AlmostEmpty: datos bajan Y% o n elementos

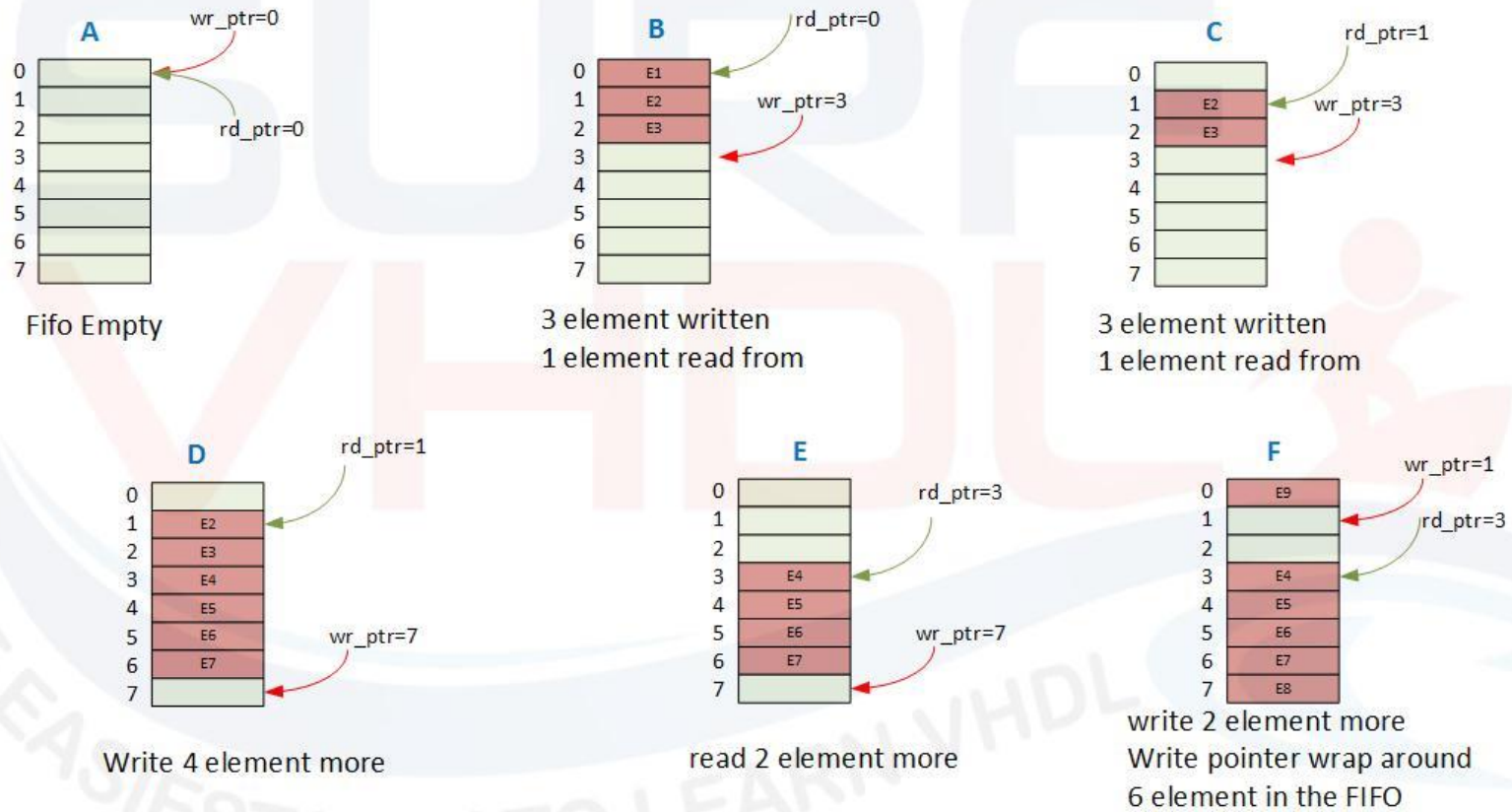
**Push**



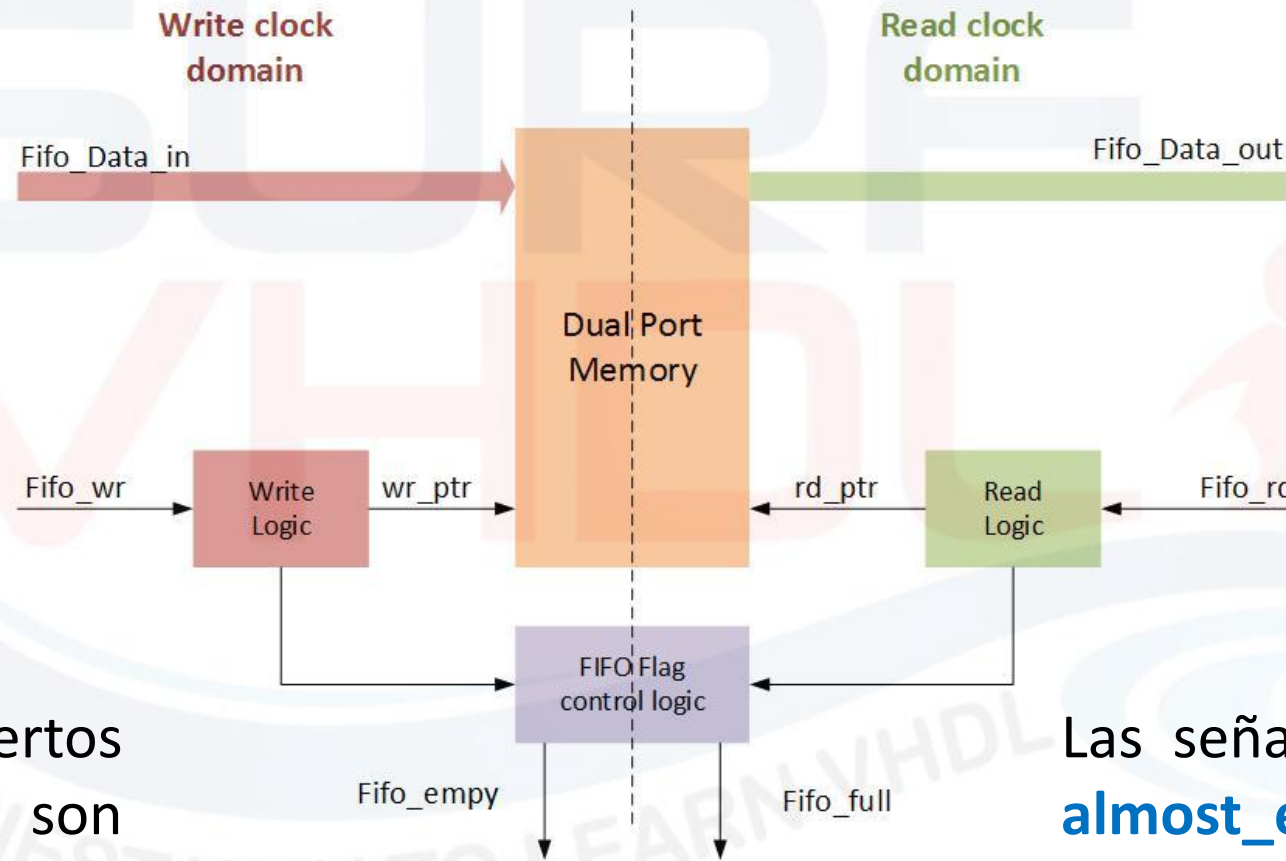
**Pop**



# Buffer/FIFO



# Buffer/FIFO



Una escritura y no lectura en full es un error (señal de salida)

¡Cuidado! Los puertos “inout” no son sintetizables.

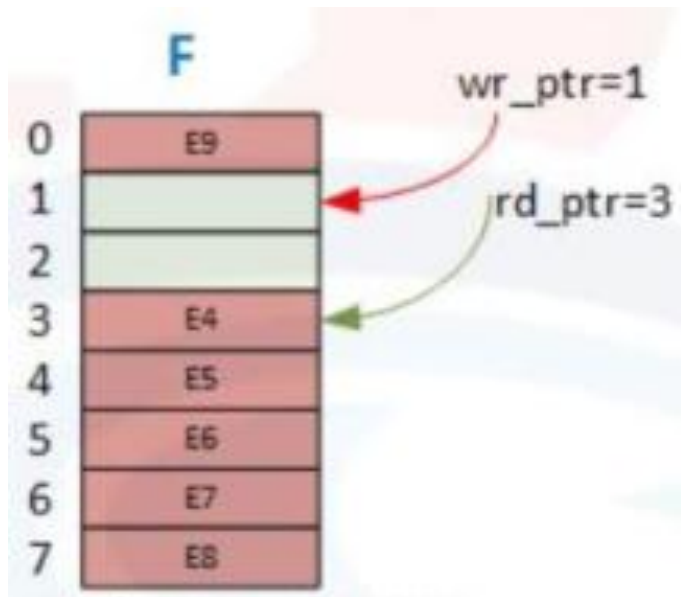
Las señales **almost\_full** y **almost\_empty** del FIFO se utilizan en Flow Control

# Flow Control

## almost\_full

Umbral lleno = 6

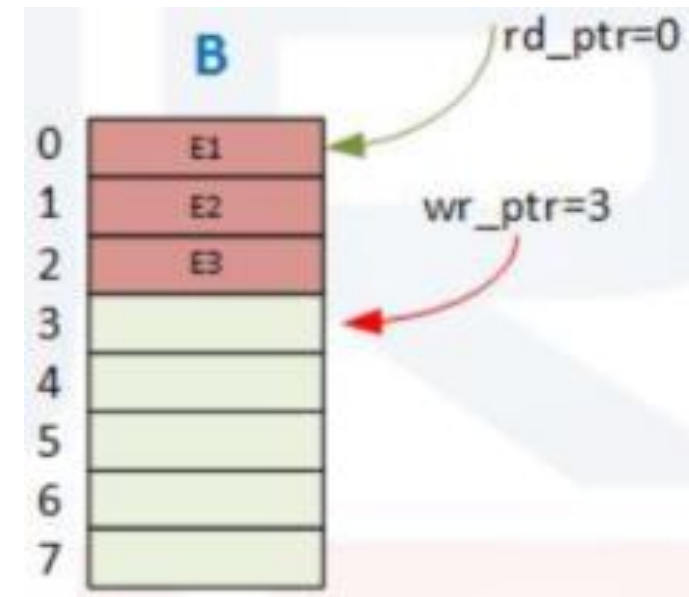
Envía una pausa



## almost\_empty

Umbral vacío = 3

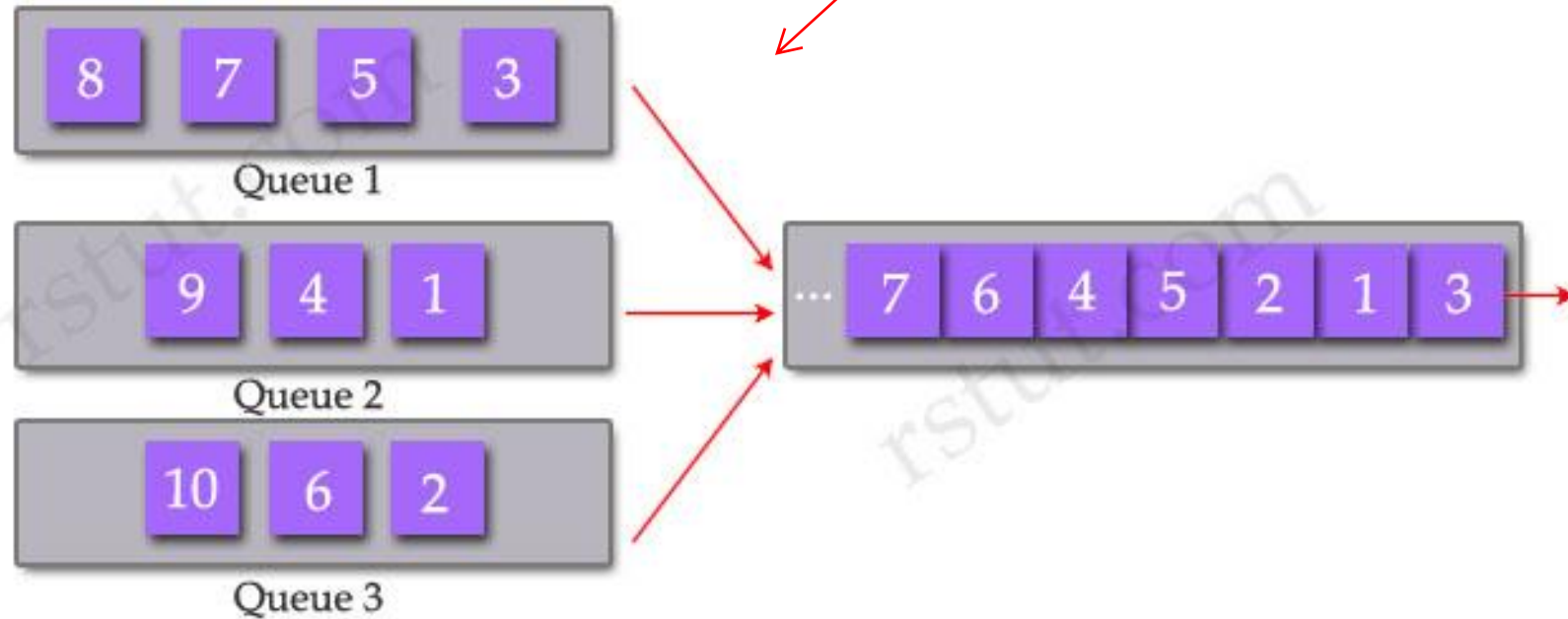
Envía un continúe



Audio: 23:50

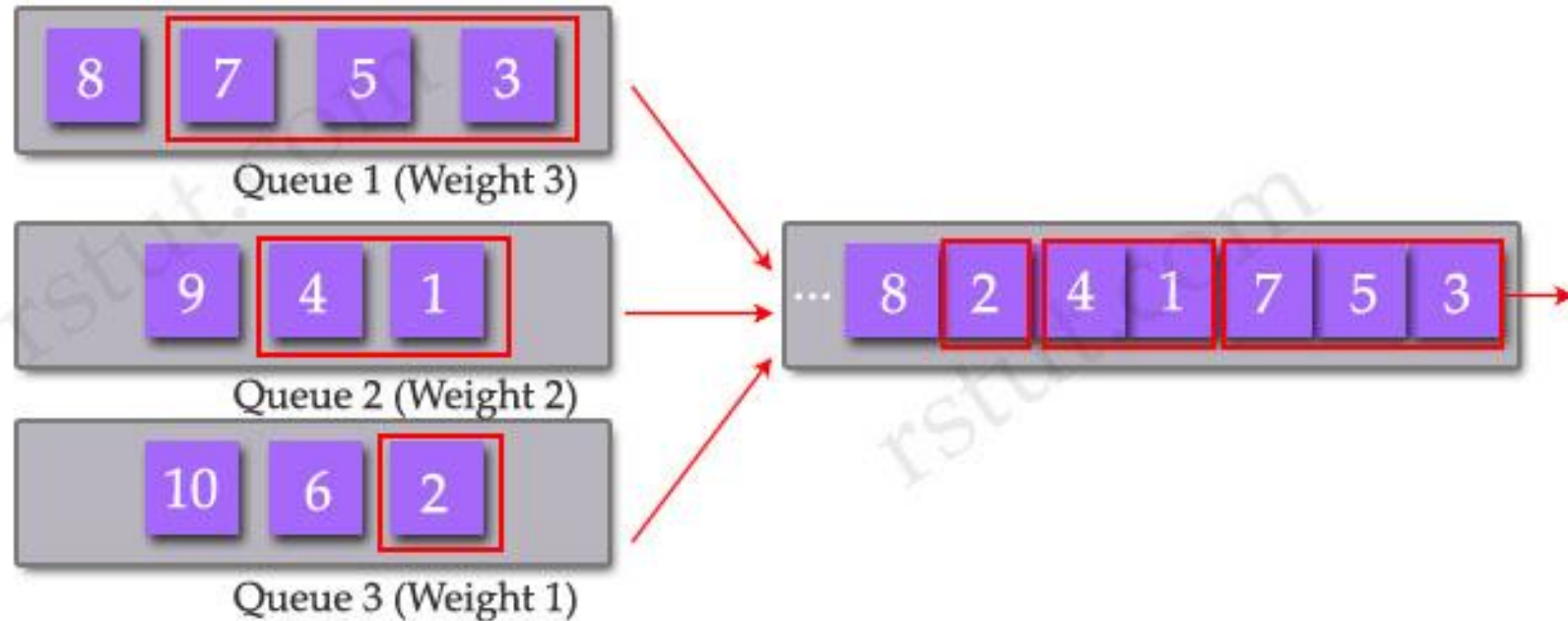
# Round Robin

Round Robin Plano: no se va a hacer, es para ejemplo



Todos los queues tienen la misma prioridad

# Weighted Round Robin



Los pesos de cada clase de tráfico son entradas al “round robin”. En el caso del Queue 1, se le asigna un valor de peso o prioridad de 3. Lo mismo para los pesos 2 y 1.

# Weighted Round Robin para arbitraje de VC en PCIe (Low-priority)

Figure 6-12: Weighted Round Robin Low-Priority VC Arbitration Table Example

Phase	VC ID
0	VC 4
1	VC 3
2	VC 2
3	VC 1
4	VC 4
5	VC 3
6	VC 0
7	VC 4
8	VC 3
9	VC 2
10	VC 1
11	VC 4
...	
62	VC 3
63	VC 0

Arbitration Logic Scans Table Entries

El árbitro recorre la tabla de arbitraje, dando permiso al canal indicado en la tabla si éste tiene una petición.





"Permiso de actualizar las cajas moradas.  
/modificar cajas SOLO cuando estamos en INIT

Despues de IDLE, o  
con todos los FIFOS  
vacios

Resolver que pasa si  
llegan 1 pause y 1  
continue. Tip: estado  
PauseContinue

Audio: 30:00

# FSM de control

- RESET: Estado de reset, cambia a INIT.
- INIT: Forzado mediante señal “init”, permite la modificación de registros “Tabla de Arbitraje” y “Umbrales”. Precedencia sobre IDLE, cambia a IDLE.
- IDLE: Todos los FIFOs están vacíos. Salida “idle” en 1 sólo en este estado. Cambia a ACTIVE al tener un FIFO no vacío.
- ACTIVE: Modo de transmisión de datos por defecto.
- PAUSE: Uno o más FIFOs envían una pausa. Indicar el ID en “pausa”. Tarda un ciclo de reloj y regresa a ACTIVE.
- CONTINUE: Uno o más FIFOs envían un continue. Indicar el ID en “continue”. Tarda un ciclo de reloj y regresa a ACTIVE.
- ERROR: Escritura y no lectura en uno o más FIFOs cuando están llenos (señal de error). Indicar el ID en “error\_full”. Sale hacia RESET únicamente al aplicar reset.



## Consideraciones del proyecto de diseño #2

- Las señales de continue y pausa son “strokes”, sólo están arriba durante un ciclo de reloj. Modelan el envío de una pausa o de un continúe.
- El probador no debe enviar datos a los puertos pausados, a menos que sea un test de error forzado. El probador debe tener lógica para saber si el puerto está pausado a partir de los “strokes” pausa y continue (PFC).
- La Tabla de Arbitraje y los Umbrales son **registros** de configuración, es decir, se debe diseñar una pequeña interfaz para modificarlos desde el probador. El Init también es un registro que habilita la escritura de los anteriores.
- El probador debe hacer “pop” a los datos del FIFO de salida con las frecuencias necesarias para cada prueba.