# Picasso-225 VO

## Arbitration/TLP re-ordering/SR-IOV

# Picasso-225 VO list

**CONTENTS**

- Arbitration
- TLP re-order
- SR-IOV

# Arbitration

# Picasso-225 VO – Arbitration

- **PCIe Arbitration**
  - Strict Priority
    - 基于固定的优先级，例如 VC0=最低，VC7=最高
  - Round Robin
    - 所有VC具有相同优先级
  - Weighted RR
    - 加权的RR

# Picasso-225 VO – Arbitration

- PCIe Switch Arbitration
  - 对于 Switch 而言，端口仲裁就是指映射到同一 VC 的 Ingress Ports 的流量在 Egress Port 进行仲裁排序。

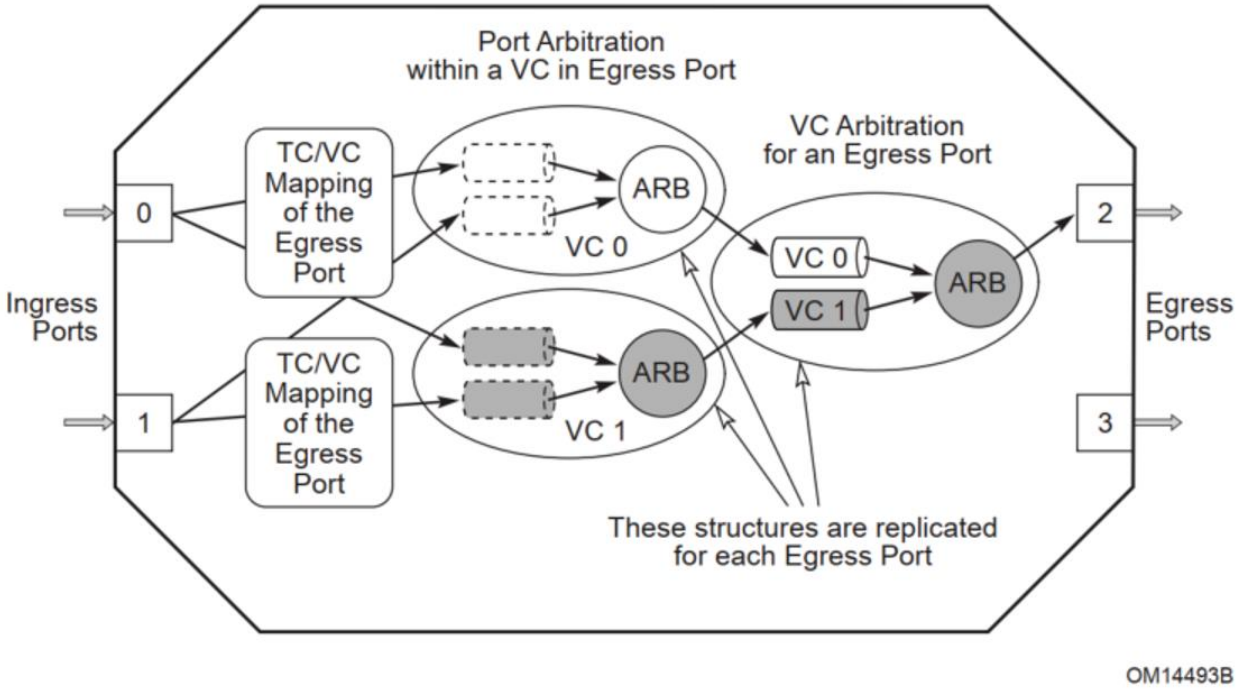
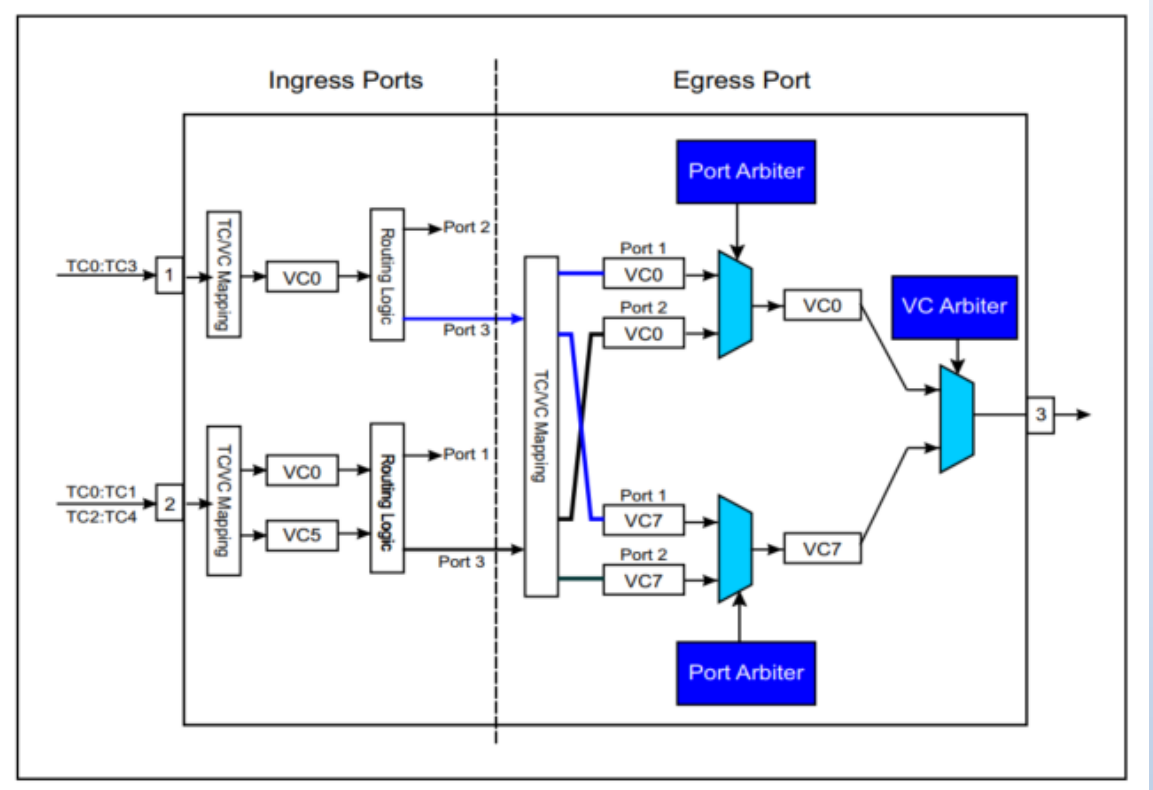



*Figure 6-8 Switch Arbitration Structure*

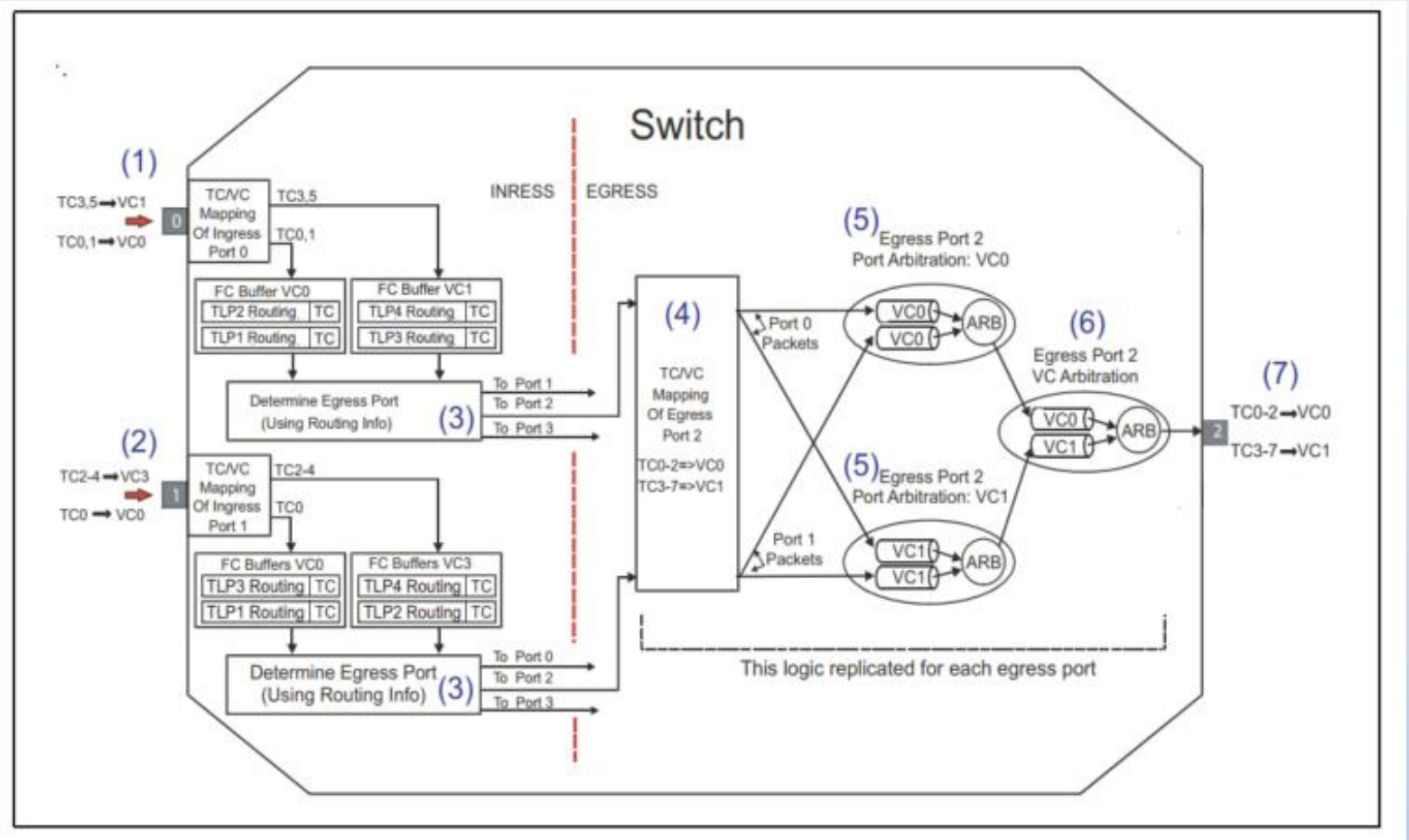
# Picasso-225 VO – Arbitration

- 依据协议：第一步根据 TLP header 选择出端口；第二步根据出端口 TC/VC 映射决定目标 VC。
- 端口仲裁：不同入端口的激励路由到出端口的相同 VC 中，在转发前必须仲裁。
- VC仲裁：出端口的不同 VCs 间的仲裁。



# Picasso-225 VO – Arbitration

- 入端口 TC/VC 映射，根据信息将激励路由到对应的出端口，如果映射的 VC 不存在则视为错误；
- 到了出端口上，进行 TC/VC 映射；
- 相同 VC 进行端口仲裁；
- 对仲裁后的不同 VCs 进行仲裁，最终发出。



# Picasso-225 VO – Arbitration

PLDA Switch VIP 使用 ***Round Robin*** 轮询仲裁，所有Ports具有相同的优先级。

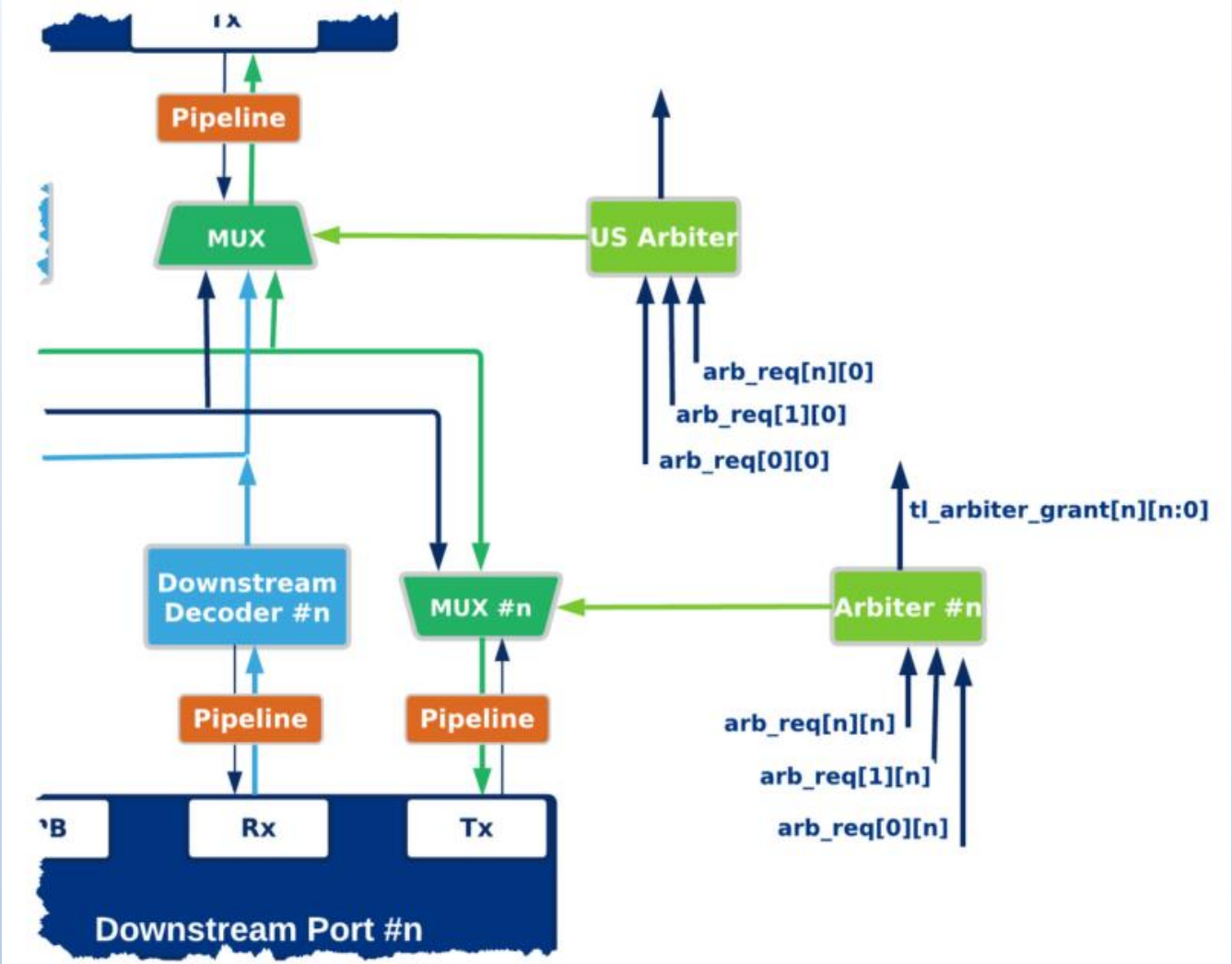



Figure 17: Mux and Arbiter Implementation in XpressSWITCH

# Picasso-225 VO – Arbitration

## VO – DPs to UP

1. wait enumeration done
2. find memory address space
3. EP1 & EP2 send multiple Transactions to UP
4. All Transactions should passed
5. Check requests' order

## VO – UP to DP

1. wait enumeration done
2. find memory address space
3. EP1 & UP send multiple Transactions to EP2
4. All Transactions should passed
5. Check requests' order

# TLP re-ordering

# Picasso-225 VO – TLP re-ordering

## PCIe TLP ordering

在严格实施**强序规则**会发生事务阻塞，影响效率。

表 8-2　基于强顺序和 RO 属性的基本顺序规则

| 行能越过列 (列 1) | | 报告的请求 | 非报告的请求 | | 完成 | |
|---|---|---|---|---|---|---|
| | | 存储器写或消息请求 (列 2) | 读请求 (列 3) | I/O 或配置写请求 (列 4) | 读完成 (列 5) | I/O 或配置写完成 (列 6) |
| 报告请求 | 存储器写或消息请求 (行 A) | a)No<br>b)Y/N | No | No | No | No |
| 非报告请求 | 读请求 (行 B) | No | No | No | No | No |
| | I/O 或配置写请求 (行 C) | No | No | No | No | No |
| 完成 | 读完成 (行 D) | a)No<br>b)Y/N | No | No | No | No |
| | I/O 或配置写完成 (行 E) | No | No | No | No | No |

# Picasso-225 VO – TLP re-ordering

## PCIe TLP ordering

在同一时刻相同 Traffic Class (TC) 需要遵循保序规则。

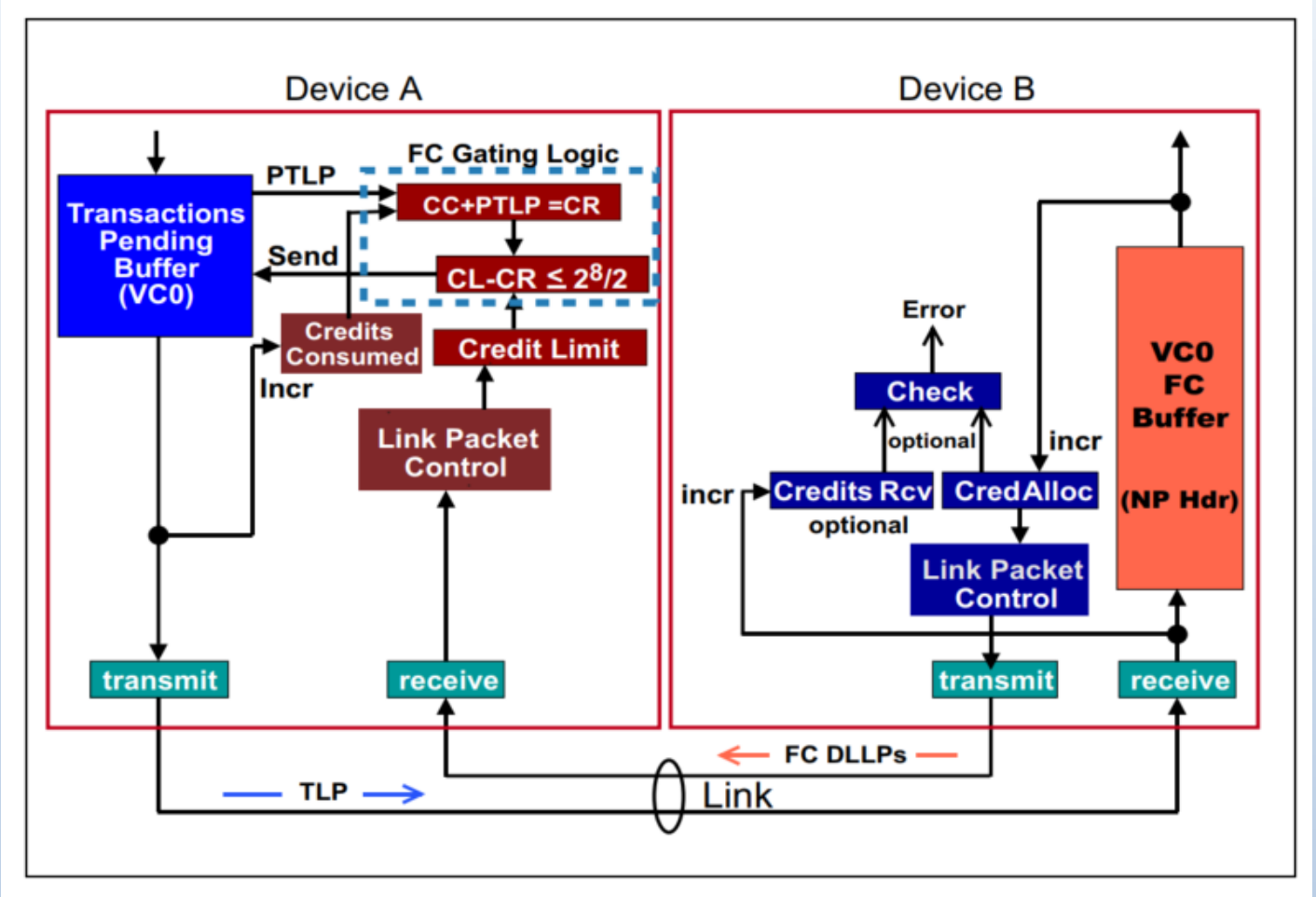
为了防止deadlock，需要reordering机制, 当 Non-Posted TLP 被阻塞住时，bypass Posted TLP 或 Completion TLP。

*Table 8-1: Simplified Ordering Rules Table*

| Row pass Column? (Col 1) | | Posted Request (Col 2) | Non-Posted Request | | Completion (Col 5) |
|---|---|---|---|---|---|
| | | | Read Request (Col 3) | NPR with Data (Col 4) | |
| Posted Request (Row A) | | a) No<br>b) Y/N | Yes | Yes | a) Y/N<br>b) Yes |
| Non-Posted Request | Read Request (Row B) | a) No<br>b) Y/N | Y/N | Y/N | Y/N |
| | NPR with Data (Row C) | a) No<br>b) Y/N | Y/N | Y/N | Y/N |
| Completion (Row D) | | a) No<br>b) Y/N | Yes | Yes | a) Y/N<br>b) No |

# Picasso-225 VO – TLP re-ordering

PCIe TLP ordering



# Picasso-225 VO – TLP re-ordering

## Picasso-225 structure

Transaction Layer 管理 TLPs，在 TLPs 传输到 DLL 层之前，检查流量控制信元。

在接收方向，TL 层收到 TLPs 后，在 Receive Buffer 中计算信元。

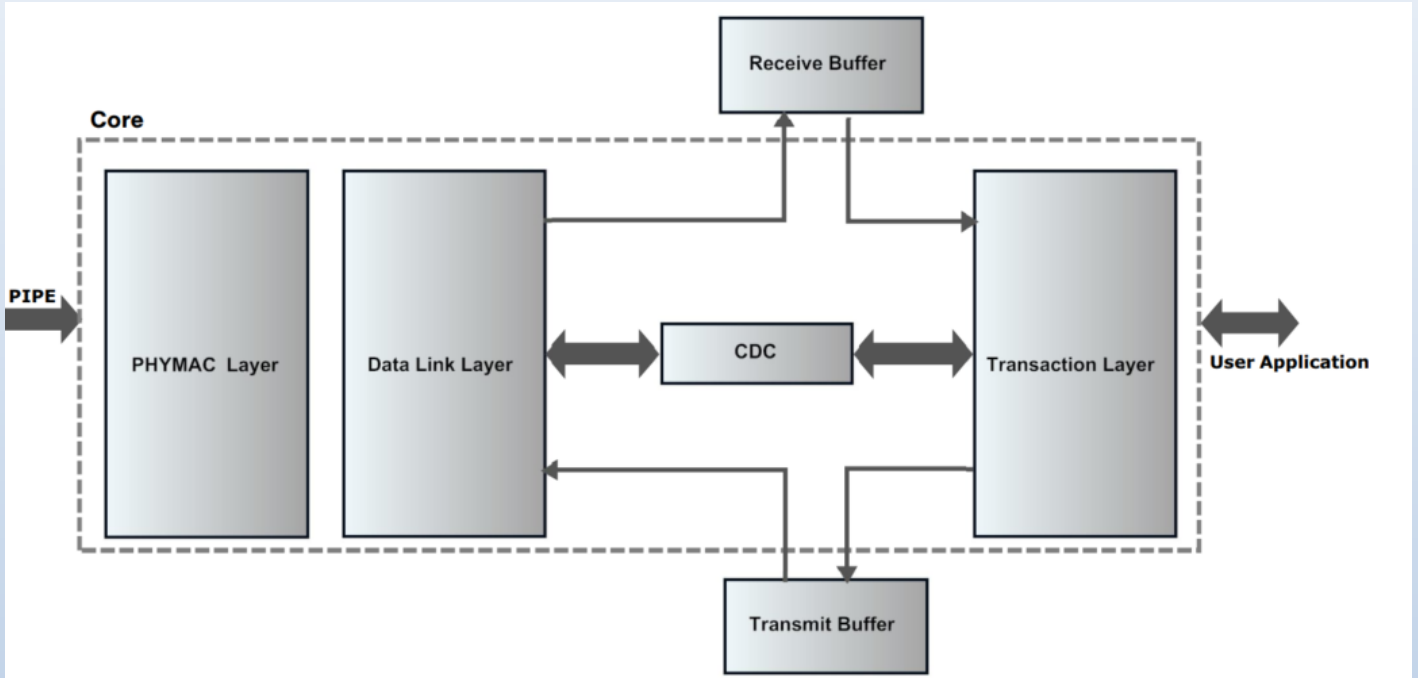
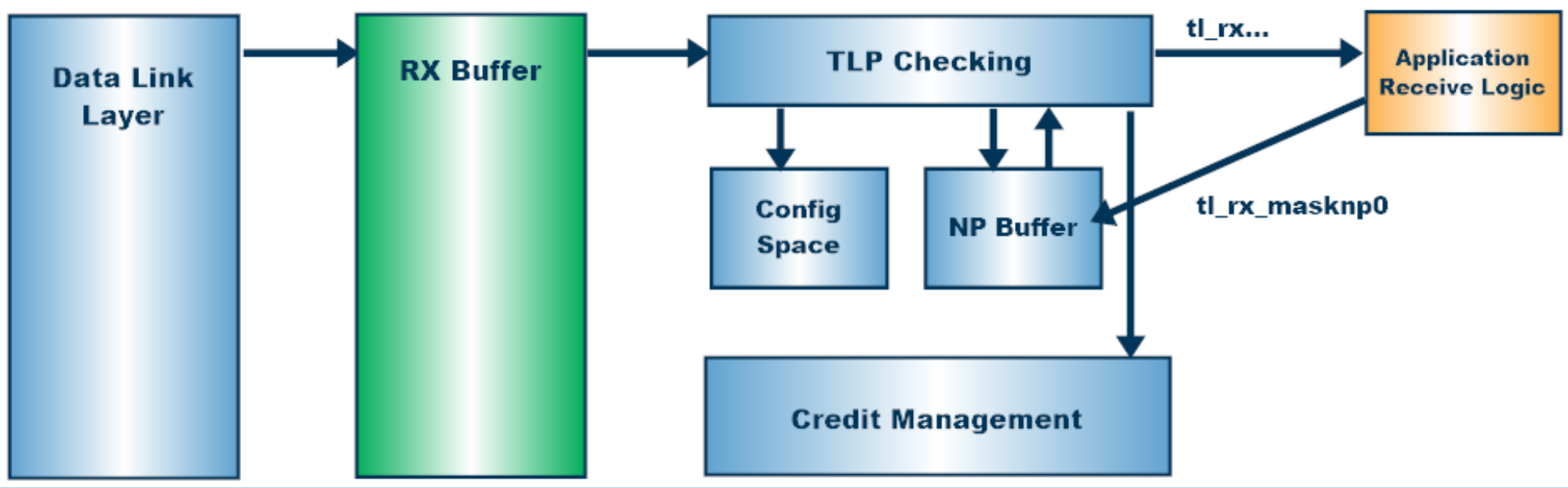



Figure 4: XpressSWITCH Port Architecture

# Picasso-225 VO – TLP re-ordering

## Picasso-225 structure -- **Port Receive Interface**

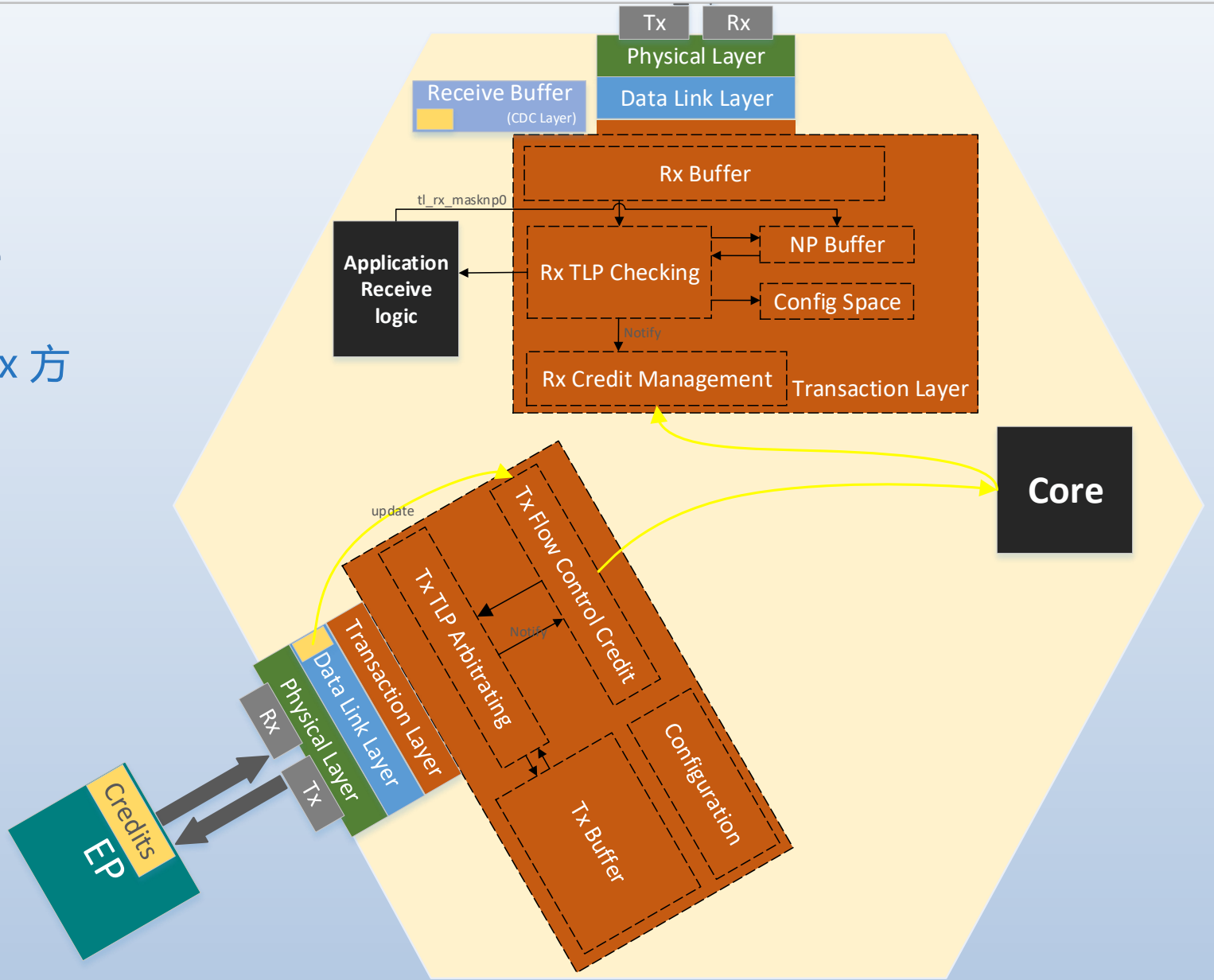



NP Buffer 可以暂时存储 Non-Posted TLPs，
从而继续接收 Posted 和 Completions TLPs，不被 Non-Posted TLPs 阻塞，避免锁死。

# Picasso-225 VO – TLP re-ordering

## Picasso-225 structure

re-ordering 功能只存在 Rx 方向，Tx 方向无此功能。



# Picasso-225 VO – TLP re-ordering

## Switch TLP re-ordering

应用收到一笔 Non-Posted TLP，然后等待一个时钟周期，拉高 tl_rx_masknp0 信号，来标明目前不再接收 NP，然后 Core 发出 Completion TLP。

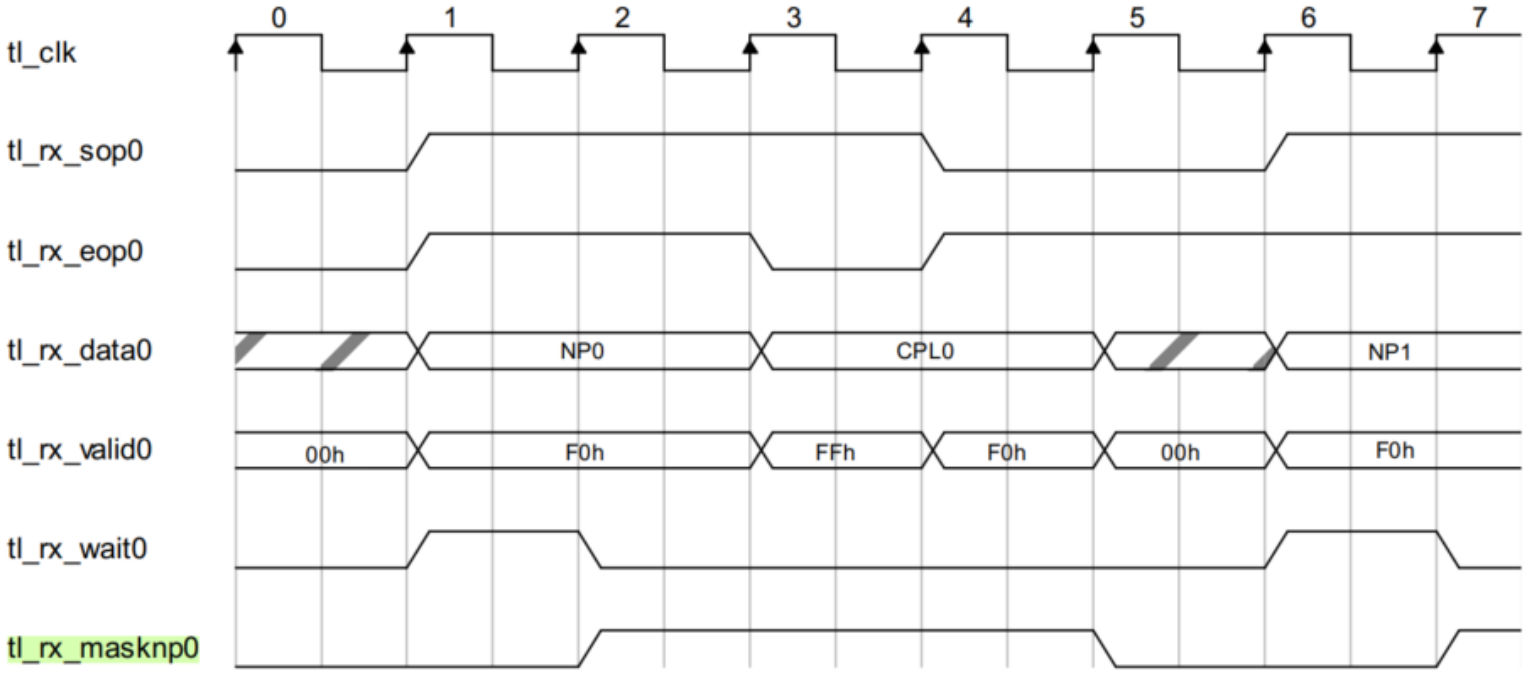
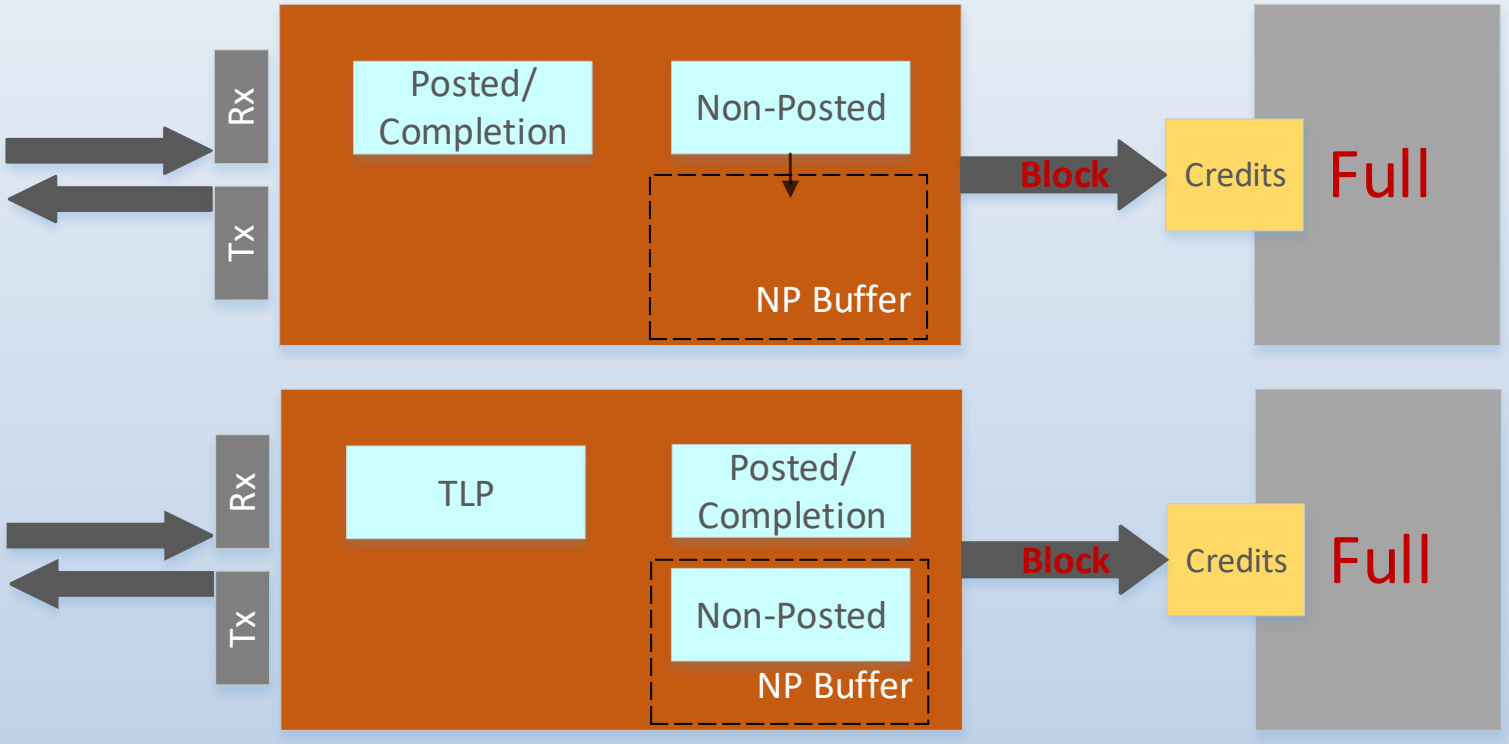



**Figure 11: Non-posted TLP received by application**

# Picasso-225 VO – TLP re-ordering

## Switch TLP re-ordering

May all Blocked??



# Picasso-225 VO – TLP re-ordering

## VO – Posted bypass

**1.** wait enumeration done
**2.** limit Posted/Non-Posted Credit and let DUT consume accredit
**3.** Block Non-Posted TLP
**4.** send Posted TLP
**5.** check *4.* should passed
**6.** Update flow credit
**7.** Check blocked TLP normally completed

## VO – Completion bypass

**1.** wait enumeration done
**2.** limit Posted/Non-Posted Credit and let DUT consume accredit
**3.** Block Non-Posted TLP
**4.** send Completion TLP
**5.** check *4.* should passed
**6.** Update flow credit
**7.** Check blocked TLP normally completed

# SR-IOV

# Picasso-225 VO – SR-IOV

- SR-IOV

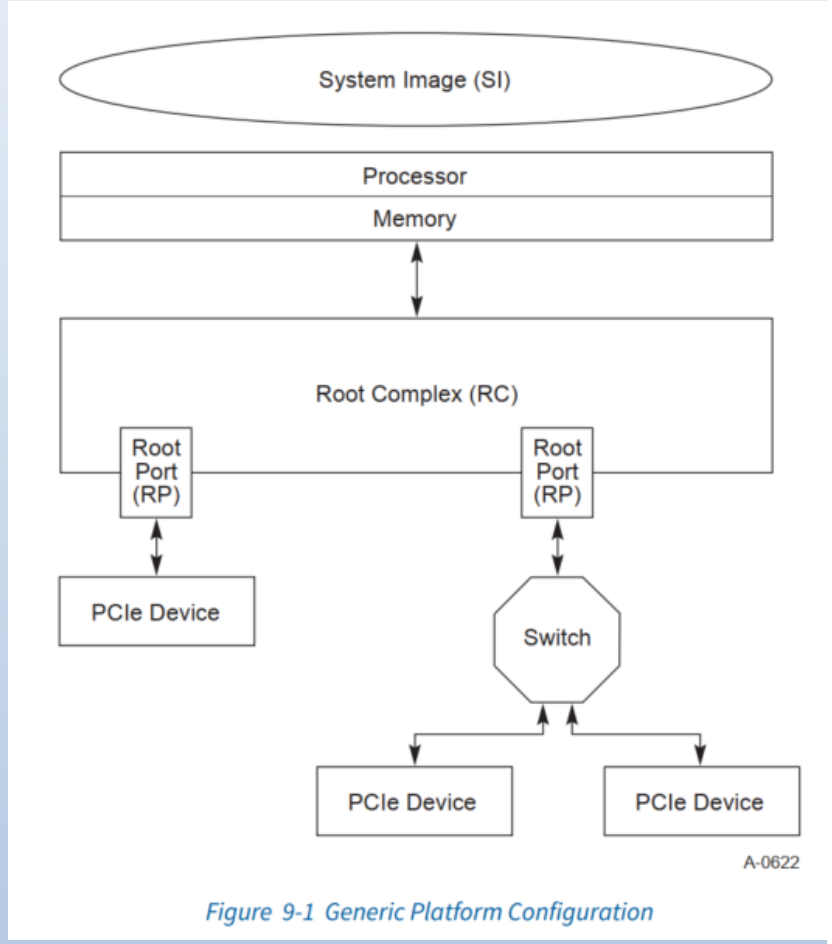



*Figure 9-1 Generic Platform Configuration*

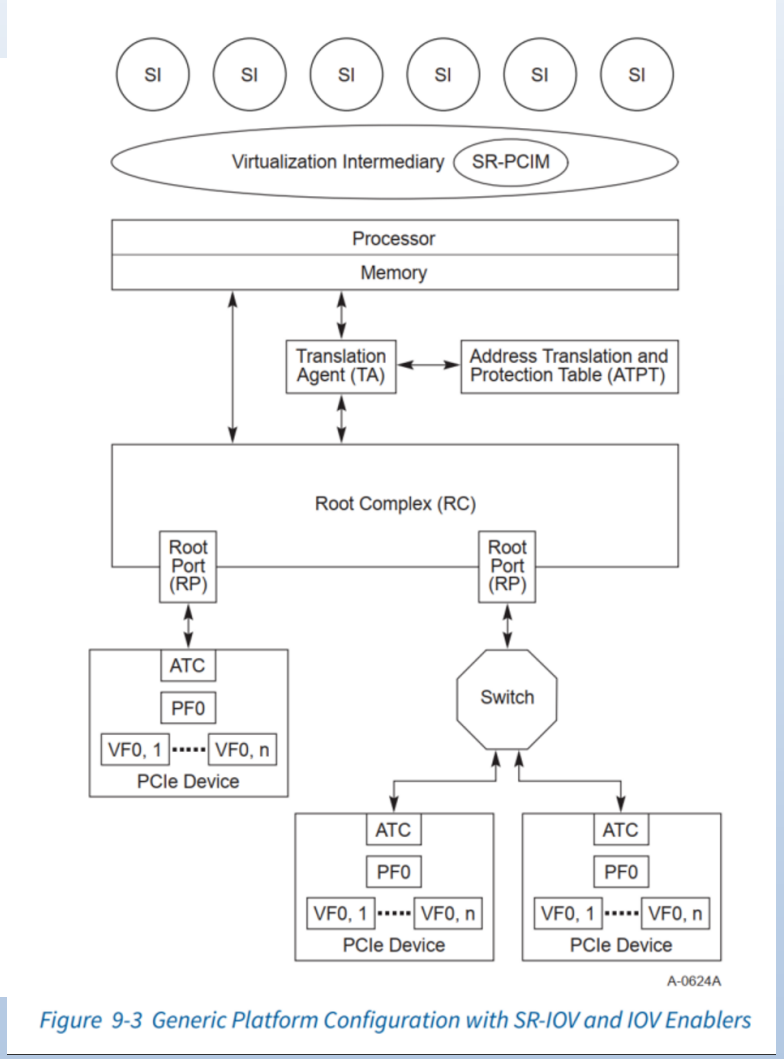



*Figure 9-3 Generic Platform Configuration with SR-IOV and IOV Enablers*

为了在不增加硬件开销的情况下，执行多个系统（SI），可以使用虚拟中介（VI），VI为每个SI提供虚拟系统。

然而当I/O数量增多时，每个I/O出站入站都需要VI处理，增大软件平台资源消耗。

SR-IOV技术可以减少这个资源消耗。

# Picasso-225 VO – SR-IOV

- SR-IOV

SR-PCIM：软件控制管理。

**Physical Function (PF)**：PF具有完整PCIe功能，PCIe Function支持SR-IOV能力扩展，可以被SR-PCIM、VI、SI访问。

**Virtual Function (VF)**：具有轻量PCIe功能，可以直接被SI访问；一个VF可以被不同SI串行共享；VF可以从PF转移到另一个PF上；与PF关联的所有VF必须与PF具有相同设备类型。

Translation Agent：PCIe地址转换为相关平台物理地址。

ATPT：地址转换保护表。

ATC：Address Translation Cache, 地址转换缓存。

**ACS**：Access Control Services, 控制TLP是否应该正确路由、阻塞、重定向。在支持SR-IOV的系统中，ACS可以阻止设备分配给VI或SI。

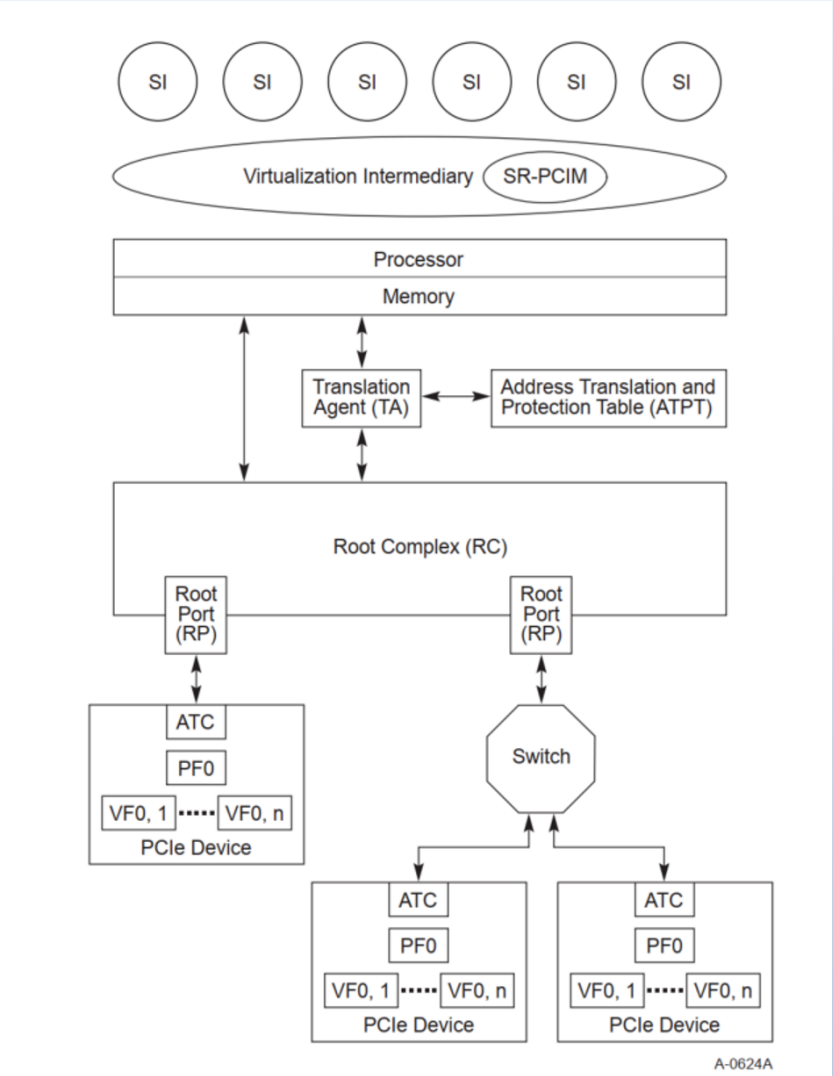



*Figure 9-3 Generic Platform Configuration with SR-IOV and IOV Enablers*

# Picasso-225 VO – SR-IOV

- SR-IOV 配置
  - NumVFs控制可见VFs数量，VFs可被PCIe发现。
  - NumVFs只有VF enable清零后才能改变。
  - 设置完VF enable，PF关联的VFs可以在PCIe组织内被访问；当VF enable清零，VFs禁用并且在PCIe组织中不可见，访问这些VFs将返回UR。
  - 配置VF enable后，设备接收 Type 0 类型的Configuration Request目标为捕获总线数内的使能VF，必须正常处理请求；
  - 当设备接收 Type 1 类型的Configuration Request目标在捕获总线号之外的使能VF，必须正常处理请求；
  - 当设备接收 Type 1 类型的Configuration Request目标在捕获总线号内，必须按照 UR 处理。
  - SR-IOV设备从 Type 0 配置类型的写配置请求中捕获总线号，不从任何 Type 1 类型的写配置请求中捕获总线号。

# Picasso-225 VO – SR-IOV

- Switch 处理总线号与基本 PCIe 流程一致。Switch 向下一级总线号到最后一级总线号范围内所有设备发送配置请求。Type 1 类型的请求访问下一级总线将被转换成 Type 0 类型，而 Type 1 类型请求访问在下一级总线号与最后一级总线号之间的总线时，将作为类型1请求转发到设备。
- ACS 功能中，具有 SR-IOV 功能的单功能设备需要处理为多功能设备；
- 必须支持 ACS P2P Request 重定向；
- 必须支持 ACS P2P Completion 重定向；
- Picasso-225 支持 ACS P2P 出口控制，阻塞Function请求。

# Picasso-225 VO – SR-IOV

- Switch SR-IOV ARI

Function Number只有3位，最多标识8个function。

拥有 ARI (Alternative Routing-ID Interpretation) 功能的设备可以扩展 function 到256个。

225支持 ARI ID 的转发。

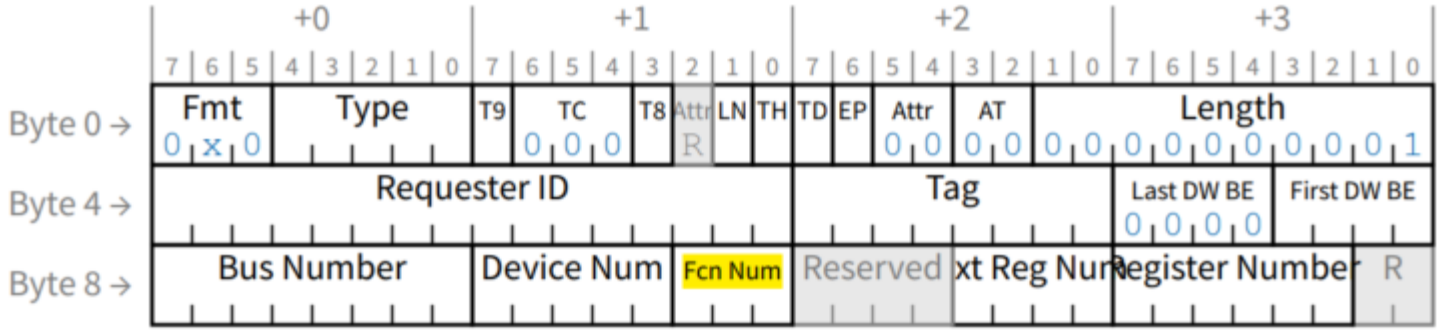



*Figure 2-20 Request Header Format for Configuration Transactions*

ARI capability is implemented in all upstream port functions; its fields are set to fixed values according to the Core configuration and can not be configured by users.

| 31:24 | 23:16 | 15:8 | 7:0 | Byte Offset |
|---|---|---|---|---|
| ARI Extended Capability Header | | | | 128h |
| ARI Control Register | | ARI Capability Register | | 12Ch |

**Table 92: ARI Extended Capability Structure**

# Picasso-225 VO – SR-IOV

- Switch SR-IOV ARI

| Bits | Field | Additional Description | Attr. |
|---|---|---|---|
| 15:0 | PCI Express Extended Capability ID | Always 000Eh. | RO |
| 19:16 | Capability Version | Hardwired to 1h. | RO |
| 31:20 | Next Capability Offset | Depends on which capabilities are implemented. | RO |

**Table 93: ARI Extended Capability Header**

| Bits | Field | Additional Description | Attr. |
|---|---|---|---|
| 0 | MFVC Function Groups Capability (M) | Hardwired to 0. | RO |
| 1 | ACS Function Groups Capability (A) | Hardwired to 0. | RO |
| 15:8 | Next Function Number | Indicates ID of next physical function; otherwise 0. | RO |

**Table 94: ARI Capability/Control Register**

# Picasso-225 VO – SR-IOV

- Switch SR-IOV ARI

## Common Decoder Signals

| | | | |
|---|---|---|---|
| **ari_forwarding_en** | in | G_NB_PORT_OUT+1 | **ARI Forwarding Enable**: Value of ARI Forwarding Enable bit in the Device Control 2 register of each Switch port. |

## Device Control 2 Register

| Bits | Field | Additional Description | Attr. |
|---|---|---|---|
| 5 | ARI Forwarding Enable | This bit is RO when k_pexconf[37]=0, RW when k_pexconf[37]=1. | RO/RW |

# Picasso-225 VO – SR-IOV

## VO – SR-IOV enable Switch access

1.wait enumeration done, VIP enables all virtual functions
2.check VF configuration
3.send CFG and MEM TLP for VF. VF must response them
4.check TLP response
5.clear VF enable
6.send CFG and MEM TLP for VF, must response UR
7.set VF enable
8.send CFG and MEM TLP for VF, must response them

## VO – SR-IOV VFNum Switch response

1.wait enumeration done
2.send TLP when VF enable is 0
3.check TLP response must as UR
4.config NumVFs
5.Sending TLPs to VF while the VF Enable bit is still 0, the VF must respond any requests as UR
6.Enable VF Enable and VF MSE
7.send TLPs to active VF
8.check should response them
9.send TLP to inactive VF
10.check should response UR

# 谢谢大家！