

ICT NCIC 系统互连组

# PCIe Transaction Converter 说明文档

V1.1

---

## Revision History

Date	Author	Issue	Comment
10/10/2021	Ning Kang	\	First Version
11/10/2021	Fan Yang	\	Add details
01/13/2022	Ning Kang	\	添加最大包长配置接口、中断接口说明

---

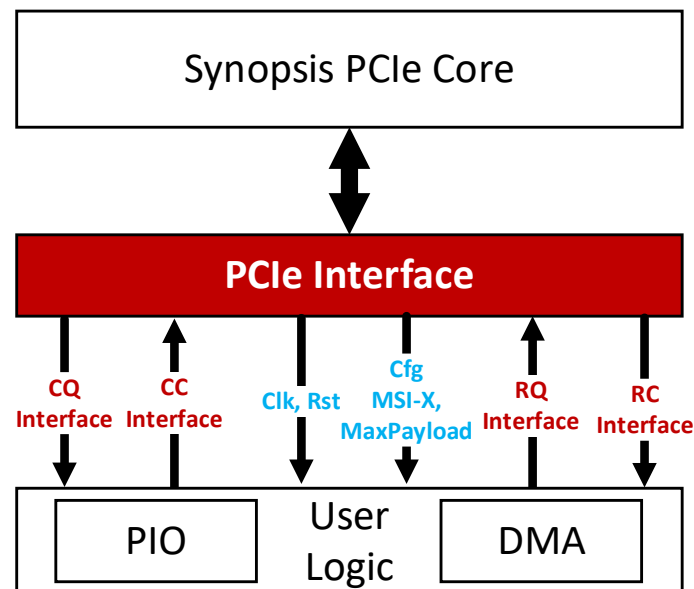
## 目录

PCIe Interface 需求说明文档 .....	1
1 功能描述.....	4
2 CQ Interface.....	4
2.1 功能描述.....	4
2.2 信号定义.....	5
2.3 包格式定义.....	5
2.3.1 tdata 信号定义 .....	5
2.3.2 tuser 信号定义 .....	6
3 CC Interface .....	6
3.1 功能描述.....	6
3.2 信号定义.....	6
3.3 包格式定义.....	6
3.3.1 tdata 信号定义 .....	6
3.3.2 tuser 信号定义 .....	7
4 RQ Interface.....	7
4.1 功能描述.....	7
4.2 信号定义.....	7
4.3 包格式定义.....	8
4.3.1 tdata 信号定义 .....	8
4.3.2 tuser 信号定义 .....	8
5 RC Interface .....	9
5.1 功能描述.....	9
5.2 信号定义.....	9
5.3 包格式定义.....	9
5.3.1 tdata 信号定义 .....	9
5.3.2 tuser 信号定义 .....	10
6 配置接口.....	10
7 时钟与复位接口.....	10
8 中断接口.....	10
9 补充说明.....	11

## 1 功能描述

由于 Synopsys PCIe Core 与目前用户逻辑（PIO+DMA）使用的接口和数据包格式并不兼容，因此，需要增加一个中间模块在两者之间进行信号和格式转换。本文档给出 PCIe Interface 的功能需求说明。

图 1.1 PCIe Interface 功能示意图



如图所示，PCIe Interface 位于 Synopsys PCIe Core 和用户逻辑之间，向用户逻辑提供六组接口。PCIe Interface 实现以下两部分功能：

- 1) **数据包分发与仲裁。**对于下行数据包，该模块根据包类型将其分发到不同的接口。其中，下行 Mem Rd 包和 Mem Wr 包被分发到 CQ Interface；下行 Cpl 包被分发到 RC Interface；对于上行数据包，该模块对 CC Interface 和 RQ Interface 进行仲裁，发往 PCIe Core。需要说明的是，Interface 保证每组接口内部的数据包按序传递即可，不同接口之间的数据包传递没有顺序关系。
- 2) **数据包格式转换。**PCIe Interface 与 Synopsys PCIe Core 之间以传递的是标准 PCIe 包，而与用户逻辑之间传递的则是自定义数据包，因此需要在两种包格式之间进行转换。需要说明的是，这种转换不涉及到数据包拆分与合并、乱序重排等复杂操作，仅涉及包头字段的增删。

后文将针对数据包格式转换给出四组接口的详细信号定义及其发送和接收的数据包格式定义。更详细的说明可以参见 Xilinx PCIe Core 的 IP 参考手册 [PG023](#) 中第 2 章，“Product Specification, Port Description”小节，以及第 3 章，“Design with the Core, Interface Operation”小节。

## 2 CQ Interface

### 2.1 功能描述

下行接口，用于转发从 CPU 下发的 PCIe 读写请求。需要说明的是，PCIe Interface 在向 CQ Interface 进行转发时，需要将 从 PCIe Core 收到的包修改为用户逻辑定义的格式（2.3.1 小节）。

2.2 信号定义

表 2.1 CQ 信号定义

信号	in/out	位宽
m_axis_cq_tvalid	in	1
m_axis_cq_tlast	in	1
m_axis_cq_tuser	in	85
m_axis_cq_tkeep	in	8
m_axis_cq_tdata	in	256
m_axis_cq_tready	out	1

2.3 包格式定义

2.3.1 tdata 信号定义

图 2.1 CQ 数据包格式（m\_axis\_cq\_tdata 第一拍的前四个 DW）

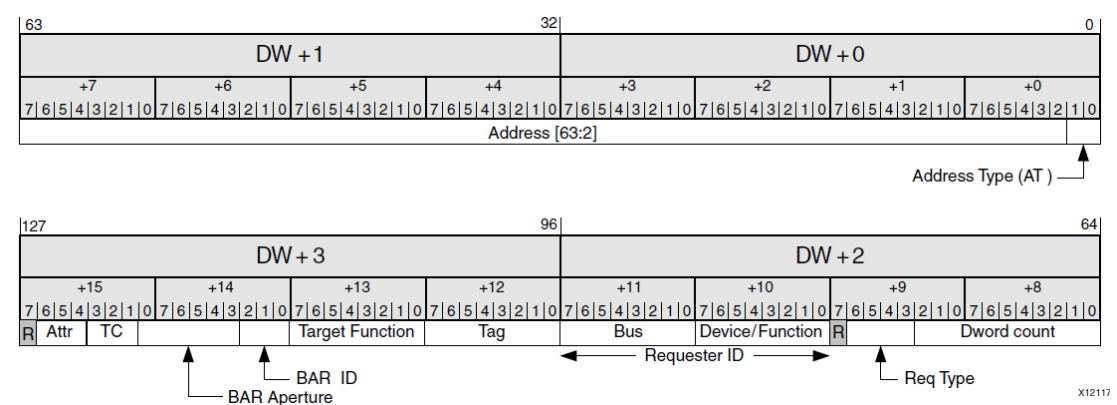


表 2.2 CQ 数据包各字段含义

Name	Description
AT	Address type（对于读请求，用户逻辑会直接转发给 CC 接口）
Address	访问的目的地址
Dword count	请求数据的大小，以双字（4Byte）为单位
Req Type	请求类型
Requester ID	请求端 ID（对于读请求，用户逻辑会直接转发给 CC 接口）
Tag	Tag 字段（对于读请求，用户逻辑会直接转发给 CC 接口）
Target Function	未使用
BAR ID	地址所在的 BAR 空间的 ID 号
BAR Aperture	未使用
TC	Transaction Class 对于读请求，用户逻辑会直接转发给 CC 接口

	(详细说明见 PG023 P100)
Attr	Attribute (对于读请求, 用户逻辑会直接转发给 CC 接口) (详细说明见 PG023 P101)

2.3.2 tuser 信号定义

表 2.3 CQ tuser 各字段含义

Name	Offset	Description
first_be	[3:0]	请求中, 第一个 DW 的字节使能 (与 PCIe 定义一致)
last_be	[7:4]	请求中, 最后一个 DW 的字节使能 (与 PCIe 定义一致)
byte_en	[39:8]	字节有效指示字段 (详细说明见 PG023 P16)
sop	[40]	start of packet, 当发送数据包的第一拍时拉高
Others	[84:41]	未使用

3 CC Interface

3.1 功能描述

CC 接口用于向 PCIe 链路发送读写请求的响应。需要说明的是, PCIe Interface 需要将用户逻辑定义的包格式转换为 PCIe 包 (见 3.3.1 小节)。

3.2 信号定义

表 3.1 CC 信号定义

信号	in/out	位宽
s_axis_cc_tvalid	out	1
s_axis_cc_tlast	out	1
s_axis_cc_tuser	out	33
s_axis_cc_tkeep	out	8
s_axis_cc_tdata	out	256
s_axis_cc_tready	in	1

3.3 包格式定义

3.3.1 tdata 信号定义

图 3.1 CC 数据包格式 (s\_axis\_cc\_tdata 第一拍的前三个 DW)

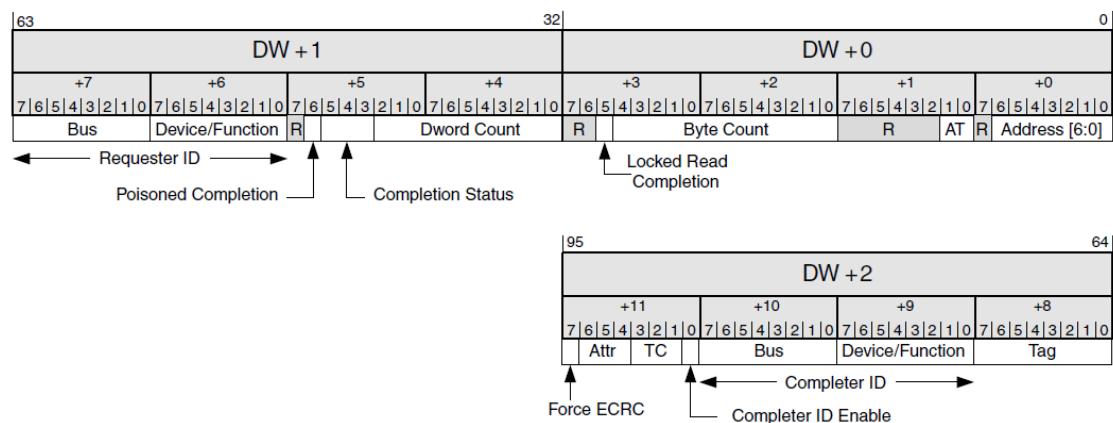


表 3.2 CC 数据包各字段含义

Name	Description
Address	读请求地址的低 7 位
AT	Address Type（从 CQ 得到）
Byte Count	请求数据的大小，以字节为单位 （详细说明见 PG023 P119）
Locked Read Completion	未使用
Dword count	响应的数据大小，以双字（4Byte）为单位
Completion Status	未使用
Poisoned Completion	未使用
Requester ID	PCI requester ID（从 CQ 得到）
Tag	tag 标签（从 CQ 得到）
Completer ID	未使用
Completer ID Enable	未使用
TC	Transaction Class（详细说明见 PG023 P129）
Attr	Attributes，（详细说明见 PG023 P129）
Force ECRC	未使用

### 3.3.2 tuser 信号定义

CC 接口的 tuser 信号未使用。

## 4 RQ Interface

### 4.1 功能描述

用于发送读写请求到 PCIe 链路。

### 4.2 信号定义

表 4.1 RQ 信号定义

信号	in/out	位宽
s_axis_rq_tvalid	out	1
s_axis_rq_tlast	out	1

s_axis_rq_tuser	out	60
s_axis_rq_tkeep	out	8
s_axis_rq_tdata	out	256
s_axis_rq_tready	in	1

4.3 包格式定义

4.3.1 tdata 信号定义

图 4.1 RQ 数据包格式（s\_axis\_rq\_tdata 第一拍的前四个 DW）

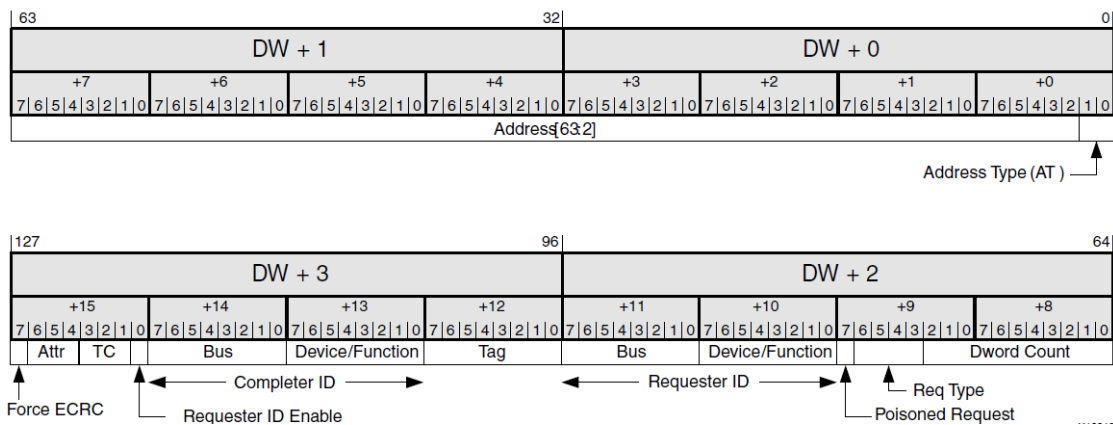


表 4.2 RQ 数据包各字段含义

Name	Description
AT	固定为 0
Address	访问的目的地址（双字对齐）
Dword count	请求数据的大小，以双字（4Byte）为单位
Req Type	请求类型
Poisioned Request	固定为 0
Requester ID	未指定（需要由 PCIe Interface 指定）
Tag	Tag 字段
Completer ID	未指定（需要由 PCIe Interface 指定）
Requester ID Enable	固定为 0
TC	固定为 0
Attr	固定为 0
Force ECRC	固定为 0

4.3.2 tuser 信号定义

表 4.3 CQ tuser 各字段含义

Name	Offset	Description
first_be	[3:0]	请求中，第一个 DW 的字节使能（与 PCIe 定义一致）
last_be	[7:4]	请求中，最后一个 DW 的字节使能（与 PCIe 定义一致）
Others	[59:8]	未使用



5 RC Interface

5.1 功能描述

用于从 PCIe 链路接收读写响应。

5.2 信号定义

表 5.1 RC 信号定义

信号	in/out	位宽
m_axis_rc_tvalid	in	1
m_axis_rc_tlast	in	1
m_axis_rc_tuser	in	75
m_axis_rc_tkeep	in	8
m_axis_rc_tdata	in	256
m_axis_rc_tready	out	1

5.3 包格式定义

5.3.1 tdata 信号定义

图 5.1 RC 数据包格式（m\_axis\_rc\_tdata 第一拍的前三个 DW）

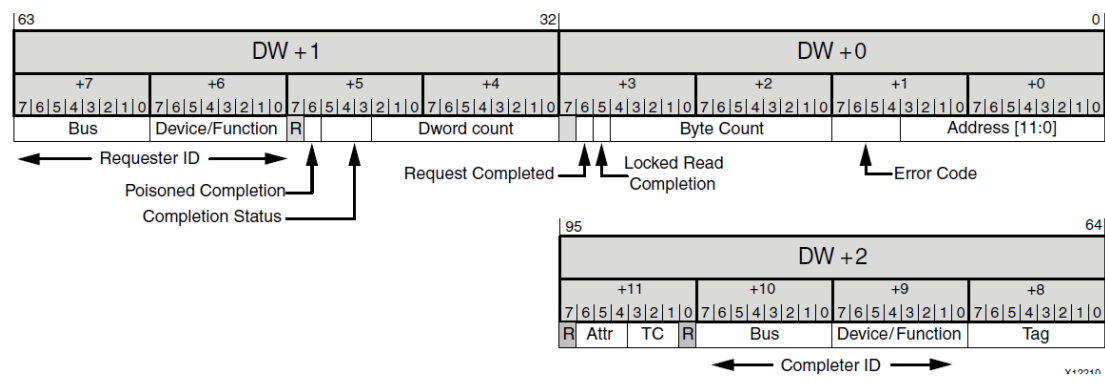


表 5.2 CC 数据包各字段含义

Name	Description
Address	读请求地址的低 12 位（只有低 7 位有效也可以）
Error Code	未实现
Byte Count	请求数据的大小，以字节为单位 （详细说明见 PG023 P120）
Locked Read Completion	未使用
Request Completed	指示一个请求的结束，（详细说明见 PG023 P154）
Dword count	响应的数据大小，以双字（4Byte）为单位
Completion Status	未使用
Poisoned Completion	未使用

<b>Requester ID</b>	未使用
<b>Tag</b>	tag 标签
<b>Completer ID</b>	未使用
<b>TC</b>	Transaction Class（详细说明见 PG023 P154）
<b>Attr</b>	Attributes（详细说明见 PG023 P154）

### 5.3.2 tuser 信号定义

表 5.3 RC tuser 各字段含义

Name	Offset	Description
byte_en	[31:0]	字节使能字段（详细说明见 PG023 P26）
Others	[74:32]	未使用

## 6 配置接口

表 6.1 PCIe 最长包大小参数说明

Name	Direction	Width	Description
max_pyld_sz	PCIe -> DMA	[2:0]	PCIe 最大数据包大小。最大数据包大小与该字段数值对应关系如下： * 3'b000 -- 128 B * 3'b001 -- 256 B * 3'b010 -- 512 B * 3'b011 -- 1024B * 3'b100 -- 2048B * 3'b101 -- 4096B
max_rd_req_sz	PCIe -> DMA	[2:0]	最大读请求大小。最大读请求大小与该字段数值对应关系如下： * 3'b000 -- 128 B * 3'b001 -- 256 B * 3'b010 -- 512 B * 3'b011 -- 1024B * 3'b100 -- 2048B * 3'b101 -- 4096B

## 7 时钟与复位接口

表 7.1 时钟与复位信号说明

Name	Direction	Width	Description
user_clk	PCIe -> DMA	[0:0]	PCIe Core 与下层应用接口时钟
user_rst	PCIe -> DMA	[0:0]	PCIe Core 输出的复位信号

## 8 中断接口

表 8.1 PCIe MSI-X 中断接口说明

Name	Direction	Width	Discription
------	-----------	-------	-------------

cfg_interrupt_msix_enable	PCIe -> DMA	[1:0]	MSI-X 中断接口信号，功能和时序完全遵照 Xilinx PCIe Core 实现 (详细信息参见 PG023 P55, P169)
cfg_interrupt_msix_mask	PCIe -> DMA	[1:0]	
cfg_interrupt_msix_data	DMA -> PCIe	[31:0]	
cfg_interrupt_msix_address	DMA -> PCIe	[63:0]	
cfg_interrupt_msix_int	DMA -> PCIe	[0:0]	
cfg_interrupt_msix_sent	PCIe -> DMA	[0:0]	
cfg_interrupt_msix_fail	PCIe -> DMA	[0:0]	
cfg_interrupt_msi_function_number	DMA -> PCIe	[2:0]	

9 补充说明

1. 目前用 Xilinx PCIe Core 进行 FPGA 验证时，发现当用户逻辑处理带宽小于 PCIe Core 的下行带宽时，PCIe Core 会出现丢包。
- 问题 1： Synopsys 的 PCIe Core 是否有同样的行为？（目前在其文档中只声明了包出现错误时会被丢弃，未提及带宽不匹配的问题）；
- 问题 2： 如果 PCIe Core 确实会有丢包行为，按照目前的设计，使用 PCIe Gen 4 x8，有效带宽约为 126Gbps。用户逻辑的位宽是 256 比特，工作频率为 780MHz 以上，对应带宽为 200Gbps，大于 PCIe 带宽，上述丢包问题也可以解决。