

文件编号：RD

发布单位：IC 研发部 SPC 项目组

发布对象：IC 研发部

发布日期：2024.10.09

生效日期：2024.10.09

# Cross Ring Spec

## 注意

本文件所含信息均具有保密性质并仅限于内部使用。不得对本文件、本文件的任何部分或本文件所含的任何信息进行未经授权地使用、披露或复制。

## NOTICE

*The information contained in this document is confidential and is intended only for internal use. Unauthorized use, disclosure or copying of this document, any part hereof or any information contained herein is strictly prohibited.*

修订记录

版本号	修订日期	作者	修订内容
1.0	2024.10.09	刘虢	初版完成
1.0.1	2024.10.11	刘虢	5.5.3 中增加 Ring Slot 中的位域关系
1.1.0	2024.10.22	刘虢	修改 ETag 架构图 等
1.2.0	2024.11.4	刘虢	增加 IQ 至 EQ 通路，修改错误
2.0.0	2025.1.27	刘虢	修改 IQ,EQ,XP 结构，更换 ETag 方案。

# 目 录

<b>1</b>	<b>Overview .....</b>	<b>5</b>
1.1	Terms and Abbreviations .....	5
1.2	Function Block .....	5
1.3	Feature .....	6
<b>2</b>	<b>Architecture .....</b>	<b>9</b>
2.1	Architecture Diagram .....	9
2.2	Architecture Description.....	9
<b>3</b>	<b>Interface .....</b>	<b>11</b>
3.1	Interface Signal Description .....	11
3.2	Interface Timing .....	13
3.3	Integration Requirement .....	13
<b>4</b>	<b>Clock and Reset.....</b>	<b>14</b>
4.1	Clock .....	14
4.2	Reset .....	14
4.3	CDC.....	14
<b>5</b>	<b>Internal Blocks .....</b>	<b>15</b>
5.1	Partition Overview.....	15
5.2	Inject Queue .....	15
5.2.1	Inject Queue Module Function.....	16
5.2.2	Inject Queue Module Interface.....	17
5.2.3	Inject Queue Module Implementation.....	17
5.3	Eject Queue .....	19
5.3.1	Eject Queue Module Function.....	19
5.3.2	Eject Queue Module Interface.....	20
5.3.3	Eject Queue Module Implementation.....	20
5.4	Ring Bridge .....	21
5.4.1	Ring Bridge Module Function.....	21
5.4.2	Ring Bridge Module Interface.....	22
5.4.3	Ring Bridge Module Implementation.....	22

5.5	Ring Slice .....	24
5.5.1	Ring Slice Module Function .....	24
5.5.2	Ring Slice Module Interface .....	24
5.5.3	Ring Slice Module Implementation .....	24
5.6	Cross Point .....	25
5.6.1	Cross Point Module Function.....	25
5.6.2	Cross Point Module Interface.....	26
5.6.3	Cross Point Module Implementation.....	28
<b>6</b>	<b>Implementation Guide.....</b>	<b>31</b>
6.1	Synthesis Guide .....	31
6.2	PD Guide .....	32
<b>7</b>	<b>Software Program Guide.....</b>	<b>33</b>
7.1	Register description .....	33
7.1.1	ETag_Config, ETag Related Configurable Register (0x04) .....	33
7.1.2	ITag_Th_H0, ITag (Horizontal Ring) Trigger Threshold Configurable Register (0x08) .....	34
7.1.3	ITag_Th_H1, ITag (Horizontal Ring) Trigger Threshold Configurable Register (0x0C) .....	34
7.1.4	ITag_Th_V0, ITag (Vertical Ring) Trigger Threshold Configurable Register (0x10) .....	34
7.1.5	ITag_Th_V1, ITag (Vertical Ring) Trigger Threshold Configurable Register (0x14) .....	35
7.1.6	ITag_Max_Num_H0, ITag (Horizontal Ring) Maximum Number Configurable Register (0x18).....	35
7.1.7	ITag_Max_Num_H1, ITag (Horizontal Ring) Maximum Number Configurable Register (0x1C).....	35
7.1.8	ITag_Max_Num_V0, ITag (Vertical Ring) Maximum Number Configurable Register (0x20).....	36
7.1.9	ITag_Max_Num_V1, ITag (Vertical Ring) Maximum Number Configurable Register (0x24).....	36
<b>8</b>	<b>Power.....</b>	<b>37</b>
<b>9</b>	<b>Future Optimisation Plan.....</b>	<b>错误!未定义书签。</b>
9.1	FT-Tag .....	错误!未定义书签。
9.1.1	EFT-Tag Module Implementation .....	错误!未定义书签。

1 Overview

1.1 Terms and Abbreviations

Table 1-1：术语和定义列表

序号	术语	定义说明
1.	STI	STI Transmission Interface
2.	DDR	Double Data-Rate
3.	EQ	Eject Queue
4.	E-Tag	Ejection Tag
5.	FIFO	First In First Out
6.	GDMA	Global Direct Memory Access
7.	IQ	Inject Queue
8.	I-Tag	Injection Tag
9.	L2M	Level-2 Memory
10.	NoC	Network on Chip
11.	QoS	Quality of Service
12.	RB	Ring Bridge
13.	SDMA	System Direct Memory Access

1.2 Function Block

Cross Ring是一种符合STI协议的，高带宽低延时的bufferless NoC设计。Cross Ring可以用于连接DDR, SDMA,GDMA,L2M等模块到同一网络。图 1-1给出了 8x8 Cross Ring 的结构图，即整个NoC的拓扑网络图。除蓝色方框为NoC挂载的设备外，其他部分构成了整个Cross Ring网络。

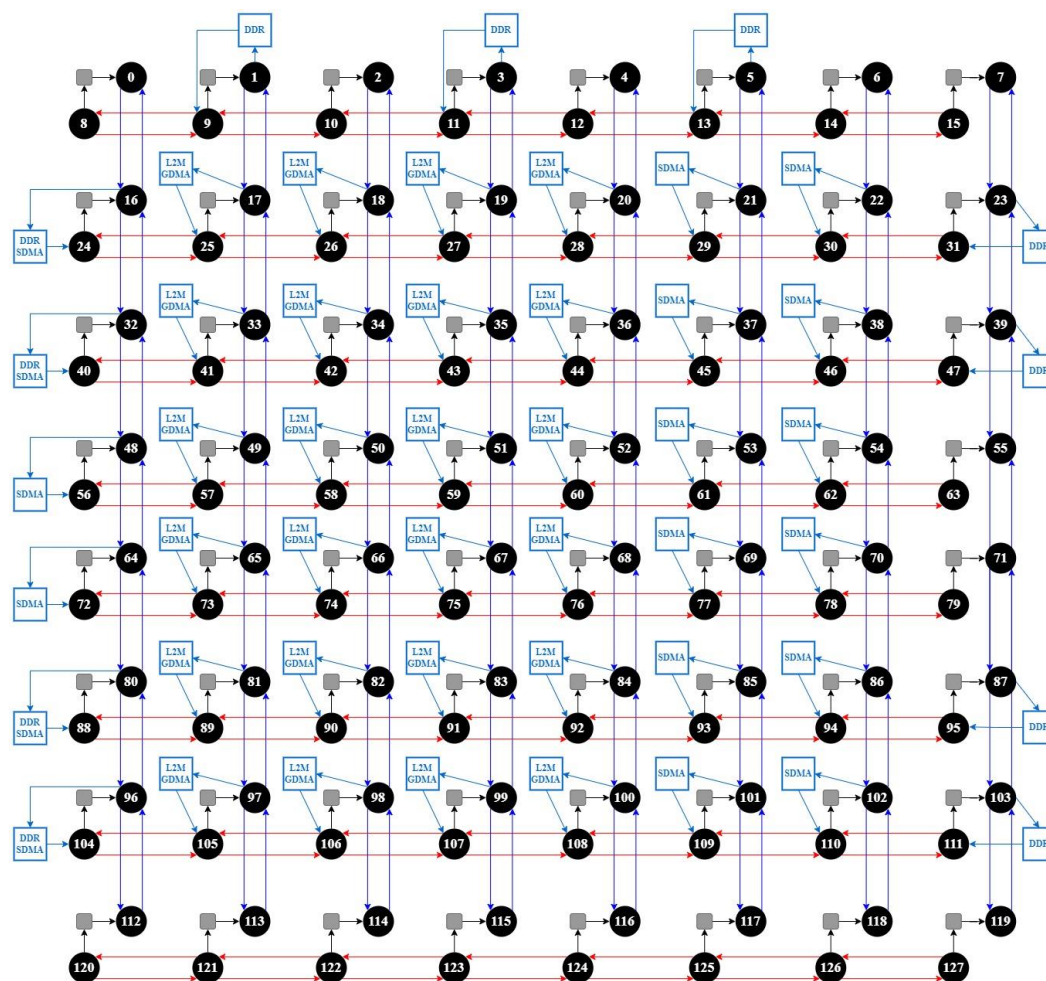


图 1-1 Cross Ring Overview

## 1.3 Feature

Cross Ring 支持以下 features:

- 数据接口： STI接口
- 配置接口：
  - 硬件可配置： Flit位宽， IP发送或接收的通道数， FIFO深度， 横、纵向环节点数。
  - 软件通过CFI接口可配置： 横、纵向环下环FIFO中各E-Tag优先等级的最大Entry保留数， 横、纵向环 I-Tag最大预留数， 横、纵向环I-Tag触发临界值。
- 连接设备的Flit首先发往横向环， 随后可通过Ring Bridge转向至纵向环， 最终到达目的地。
- 不支持QoS。
- 防饿死机制
  - 上环防饿死：
    - 支持I-Tag。
  - 下环防饿死：
    - 支持E-Tag。

该模块硬件可配置参数详见表1-2，软件可配置参数详见表 1-3。

表 1-2 硬件可配置参数

Parameter Name	Valid Value	Description
Flit Width	0:2048	Flit 位宽
NUM_IP	待定	IP 发出/接收的通道数
NODE_NUM_X	1:16	网络横向节点数
NODE_NUM_Y	1:16	网络纵向节点数
SLICE_NUM_BETWEEN_NODE_X	待定	网络节点横向间 Slice 数
SLICE_NUM_BETWEEN_NODE_Y	待定	网络节点纵向间 Slice 数
TGTID_DEVICEID_OFFSET	0:Flit Width	目标 Device ID 在 Flit 中的偏移量
TGTID_XID_OFFSET	0:Flit Width	目标 XID 在 Flit 中的偏移量
TGTID_YID_OFFSET	0:Flit Width	目标 YID 在 Flit 中的偏移量
IPSIDE_CREDIT	1:16	IP 端 Credit 值
IQ_FREQINC_FIFO_DEPTH	2:8	IQ FREQINC 模块 FIFO 深度
IQ_CH_FIFO_DEPTH	2:8	IQ CH_FIFO 模块 FIFO 深度
IQ_OUT_FIFO_DEPTH	2:8	IQ OUT_FIFO 模块 FIFO 深度
RB_IN_FIFO_DEPTH	2:8	RB 输入 FIFO 深度
RB_OUT_FIFO_DEPTH	2:8	RB 输出 FIFO 深度
EQ_IN_FIFO_DEPTH	2:8	EQ 输入 FIFO 深度
EQ_CH_FIFO_DEPTH	2:8	EQ CH_FIFO 模块 FIFO 深度
EQ_FREQDEC_IN_FIFO_DEPTH	2:8	EQ FREQDEC 模块输入 FIFO 深度
EQ_FREQDEC_OUT_FIFO_DEPTH	2:8	EQ FREQDEC 模块输出 FIFO 深度
ITAG_TH_COUNTER_WIDTH (ITAG_TH_C_W)	1:8	I-Tag 触发临界值计数器位宽

表 1-3 软件可配置参数

Parameter Name	Valid Value	Description
ITag_Trigger_Th_H	$0 : (2^{\wedge} \text{ITAG\_TH\_C\_W}) - 1$	横向环 I-Tag 触发临界值
ITag_Trigger_Th_V	$0 : (2^{\wedge} \text{ITAG\_TH\_C\_W}) - 1$	纵向环 I-Tag 触发临界值
ITag_Max_Num_H	1:4 待定	横向环 XY 节点 I-Tag 最大预留数
ITag_Max_Num_V	1:4 待定	纵向环 XY 节点 I-Tag 最大预留数
TL_Etag_T1_UE_MAX	$< \text{RB\_IN\_FIFO\_DEPTH},$ $> \text{TL\_Etag\_T2\_UE\_MAX}$	横向环向左 T1 级 ETag 可用 FIFO Entry 数
TL_Etag_T2_UE_MAX	$< \text{RB\_IN\_FIFO\_DEPTH}-2$	横向环向左 T2 级 ETag（无 ETag）可用 FIFO Entry 数
TR_Etag_T2_UE_MAX	$< \text{RB\_IN\_FIFO\_DEPTH}-1$	横向环向右 T2 级 ETag（无 ETag）可用 FIFO Entry 数
TU_Etag_T1_UE_MAX	$< \text{EQ\_IN\_FIFO\_DEPTH},$ $> \text{TL\_Etag\_T2\_UE\_MAX}$	纵向环向上 T1 级 ETag 可用 FIFO Entry 数
TU_Etag_T2_UE_MAX	$< \text{EQ\_IN\_FIFO\_DEPTH}-2$	纵向环向上 T2 级 ETag（无 ETag）可用 FIFO Entry 数
TD_Etag_T2_UE_MAX	$< \text{EQ\_IN\_FIFO\_DEPTH}-1$	纵向环向下 T2 级 ETag（无 ETag）可用 FIFO Entry 数



## 2 Architecture

### 2.1 Architecture Diagram

图 2-1 展示了 Cross Ring Piece 的架构，Cross Ring 由多个 Cross Ring Piece 互联构成。

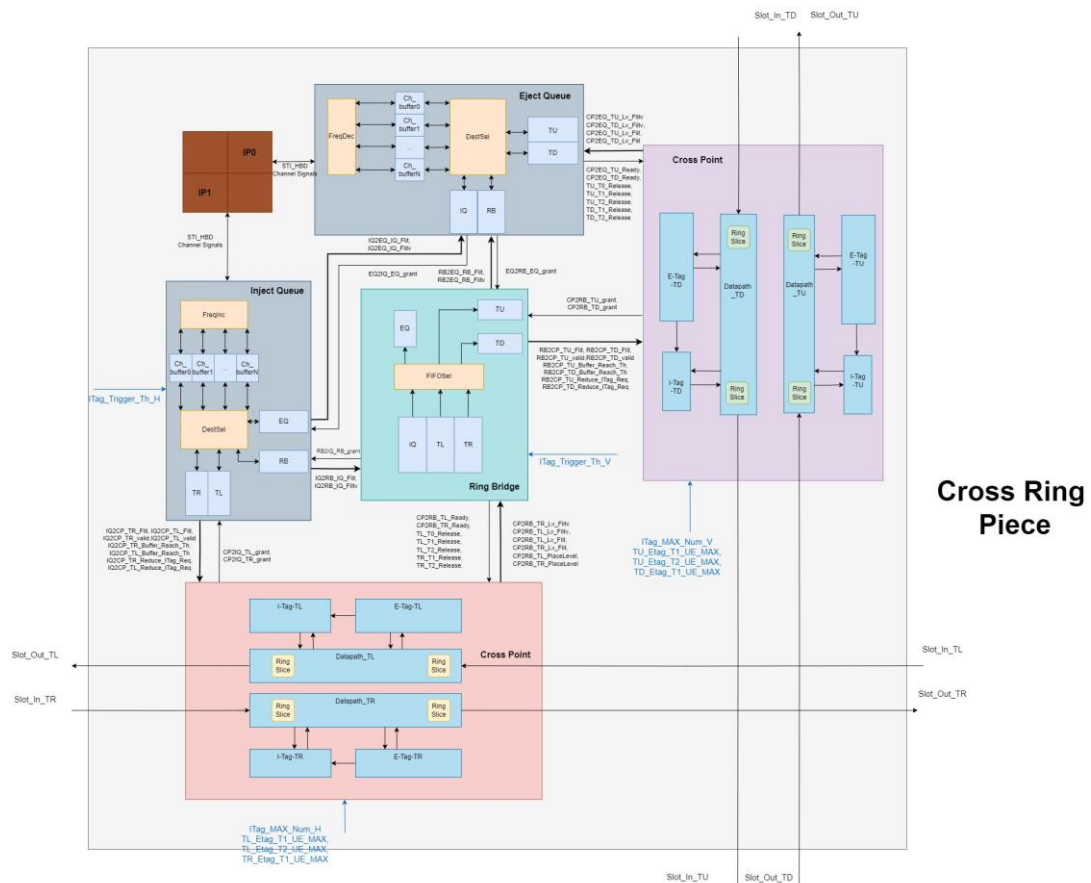


图 2-1 Cross Ring Piece Overview

### 2.2 Architecture Description

Cross Ring Piece 主要是由以下五个部分组成：

- **Inject Queues:** 注入队列模块。从IP发送Flit到Cross Point, Ring Bridge和Eject Queue
- **Eject Queues:** 发射队列模块。从Cross Point, Ring Bridge和Inject Queue接收Flit的模块。
- **Ring Bridge :** 环桥模块。帮助Flit从横向环/Inject Queue转向至Flit纵向环/Eject Queue。
- **Ring Slice:** 构成环路的最基本单元。
- **Cross Point:** 交换模块。控制上下环，进出队列或bridge的重要模块。利用I-Tag和

E-Tag机制提供防饿死保护。

Algorithm 1 展示了Cross Ring内部的工作流程。

### 2.2.1.1 Algorithm 1 IQ\_RB, IQ\_TL, IQ\_TR Module Implementation

---

#### Algorithm 1 Cross Ring Working Flow Pseudo-Code

---

```

Input:   Flit_In
Output: Flit_Out

1: IP 发送 Flit_In 给 IQ
2: Flit 进入 IQ 的输入 FIFO。
3: if (目的节点为本节点) then
4:   Flit 进入 EQ 的输入 FIFO
5:   Goto Step 32
6: else if (目的节点与本节点横坐标相同) then
6:   Flit 进入 RB 的输入 FIFO
7:   Goto Step 29
8: end if
9: else then
10:  for Flit 未上环 do
11:      if (!Slot Valid & 未被其他 FIFO 预留) then
12:          Flit 上环
13:          if (已被标记 I-Tag) then
14:              撤回 I-Tag
15:          end if
16:          else if (满足标记 I-Tag 条件) then
17:              标记 I-Tag
18:          else if (I-Tag 过量标记) then
19:              撤回 I-Tag
20:          else
21:              不标记 I-Tag
22:          end for
23:  for Flit 未下环 do
24:      if (Flit 所持 ETag 相应等级在 FIFO 中有可使用空位) then
25:          下环进入 RB (横向环)/ EQ (纵向环) FIFO
26:      else
27:          升级 ETag 等级直至本方向最高等级
28:      end for
29:  横向环 Round-Robin 竞争直至成功
30:  纵向环重复 Step9 至 Step28
31: end if
32: Flit 在 EQ 中 Round-Robin 竞争直至成功
33: Flit_Out 被发送至 IP 终点

```

---

3 Interface

3.1 Interface Signal Description

图 3-1 给出了模块的顶层接口连接。

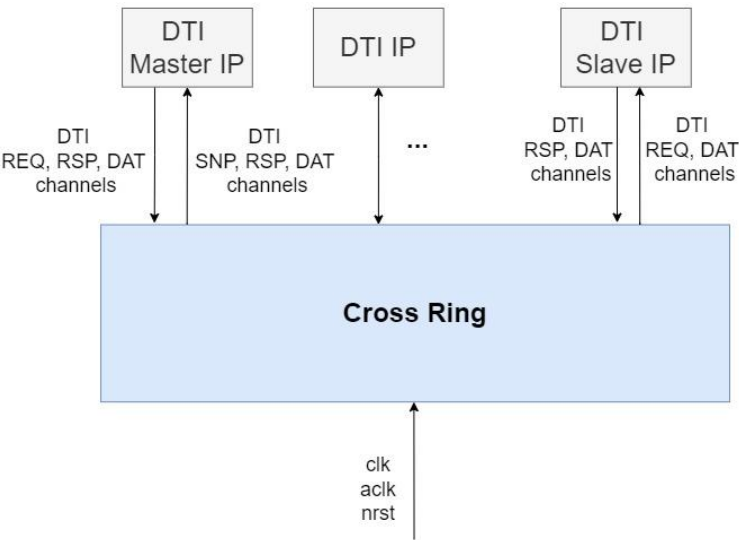


图 3-1 Connection Overview

表 3-1 给出了模块的顶层接口，数据位宽，方向及相应描述。

表 3-1: Top Interface

Function	Name	Width	Direction	description
PMU	clk	1	I	Clock
	aclk	1	I	Slower Clock
	rst_n	1	I	Reset (active low)
RN-F	RXREQFLIT	512/1024	I	接收到的 REQ channel Flit
	RXREQFLITV	1	I	接收到的 REQ channel Flit Valid
	RXREQLCRDV	1	I	接收到的 REQ channel Link Credit Valid
	RXRSPFLIT	512/1024	I	接收到的 RSP channel Flit
	RXRSPFLITV	1	I	接收到的 RSP channel Flit Valid
	RXRSPLCRDV	1	I	接收到的 RSP channel Link Credit Valid
	RXDATFLIT	512/1024	I	接收到的 DAT channel Flit
	RXDATFLIT V	1	I	接收到的 DAT channel Flit Valid
	RXDATLCRDV	1	I	接收到的 DAT channel Link Credit Valid
	TXRSPFLIT	512/1024	O	发送的 RSP channel Flit
	TXRSPFLITV	1	O	发送的 RSP channel Flit Valid
	TXRSPLCRDV	1	O	发送的 RSP channel Link Credit Valid
RN-F	TXSNPFLIT	512/1024	O	发送的 SNP channel Flit

Function	Name	Width	Direction	description
	TXSNPFLITV	1	O	发送的 SNP channel Flit Valid
	TXSNPLCRDV	1	O	发送的 SNP channel Link Credit Valid
	TXDATFLIT	512/1024	O	发送的 DAT channel Flit
	TXDATFLITV	1	O	发送的 DAT channel Flit Valid
	TXDATLCRDV	1	O	发送的 DAT channel Link Credit Valid
SN-F	RXRSPFLIT	512/1024	I	接收到的 RSP channel Flit
	RXRSPFLITV	1	I	接收到的 RSP channel Flit Valid
	RXRSPLCRDV	1	I	接收到的 RSP channel Link Credit Valid
	RXDATFLIT	512/1024	I	接收到的 DAT channel Flit
	RXDATFLITV	1	I	接收到的 DAT channel Flit Valid
	RXDATLCRDV	1	I	接收到的 DAT channel Link Credit Valid
	TXREQFLIT	512/1024	O	发送的 RSP channel Flit
	TXREQFLITV	1	O	发送的 RSP channel Flit Valid
	TXREQLCRDV	1	O	发送的 RSP channel Link Credit Valid
	TXDATFLIT	512/1024	O	发送的 DAT channel Flit
	TXDATFLITV	1	O	发送的 DAT channel Flit Valid
	TXDATLCRDV	1	O	发送的 DAT channel Link Credit Valid

### **3.2 Interface Timing**

- 时钟: 2GHz (待定)
- 时钟: 1GHz (待定)

### **3.3 Integration Requirement**

## 4 Clock and Reset

- 时钟: 2GHz (待定), 1GHz (待定)
- 复位: 低有效复位

### 4.1 Clock

- 时钟: 2GHz (待定), 1GHz (待定)

### 4.2 Reset

- 复位: 低有效复位

### 4.3 CDC

IP在连接Inject Queue时使用的频率为1GHz。而由于整个Cross Ring 网络的频率为2GHz, 在IP发送的Flit进入网络前或网络中的Flit离开网络进入IP前, 需要使用异步桥将频率进行转换。Cross Ring的跨时钟域设计仅存在于Inject Queue和Eject Queue中。其实现是依靠已有的IP kcin\_freqdiv\_l2h\_flit以及kcin\_freqdiv\_h2l\_flit实现的。

## 5 Internal Blocks

## 5.1 Partition Overview

图 5-1 展示了 Cross Ring Piece 的 internal partition。

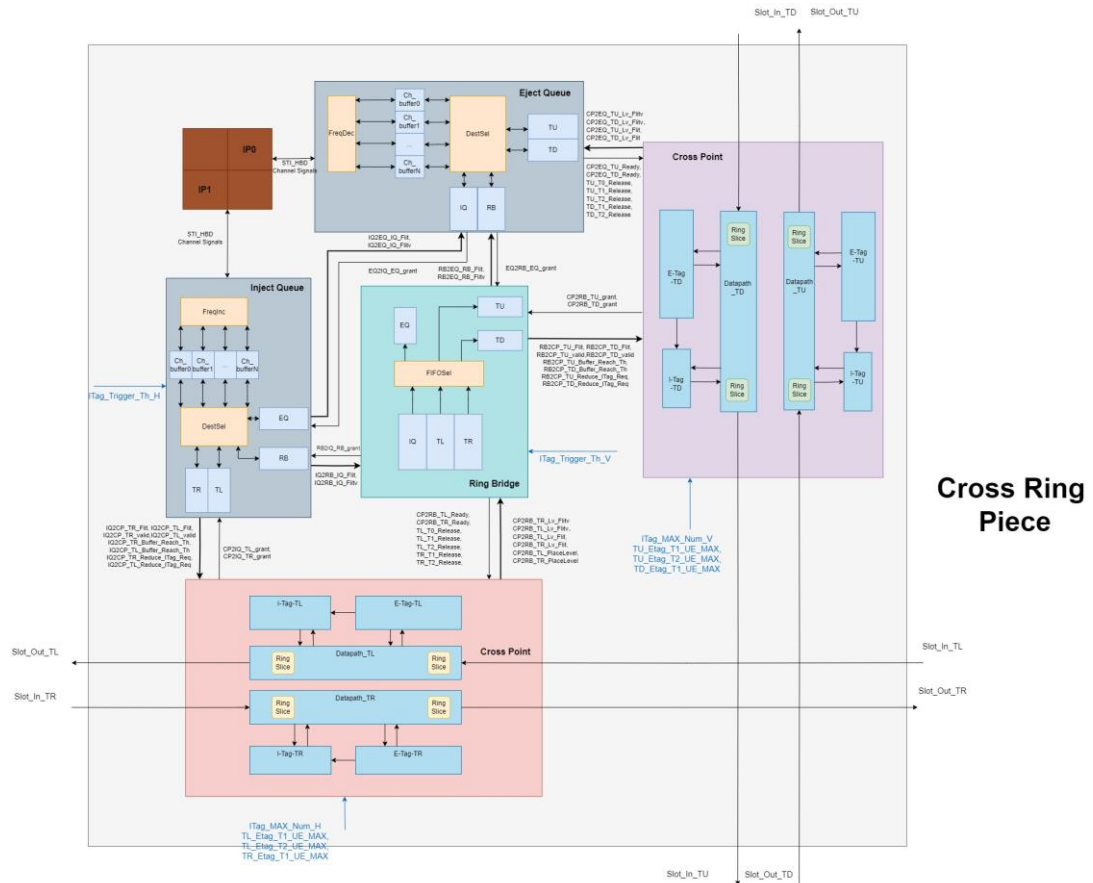


图 5-1 Internal Partition

## 5.2 Inject Queue

图 5-1 展示了 Inject Queue 的结构图。

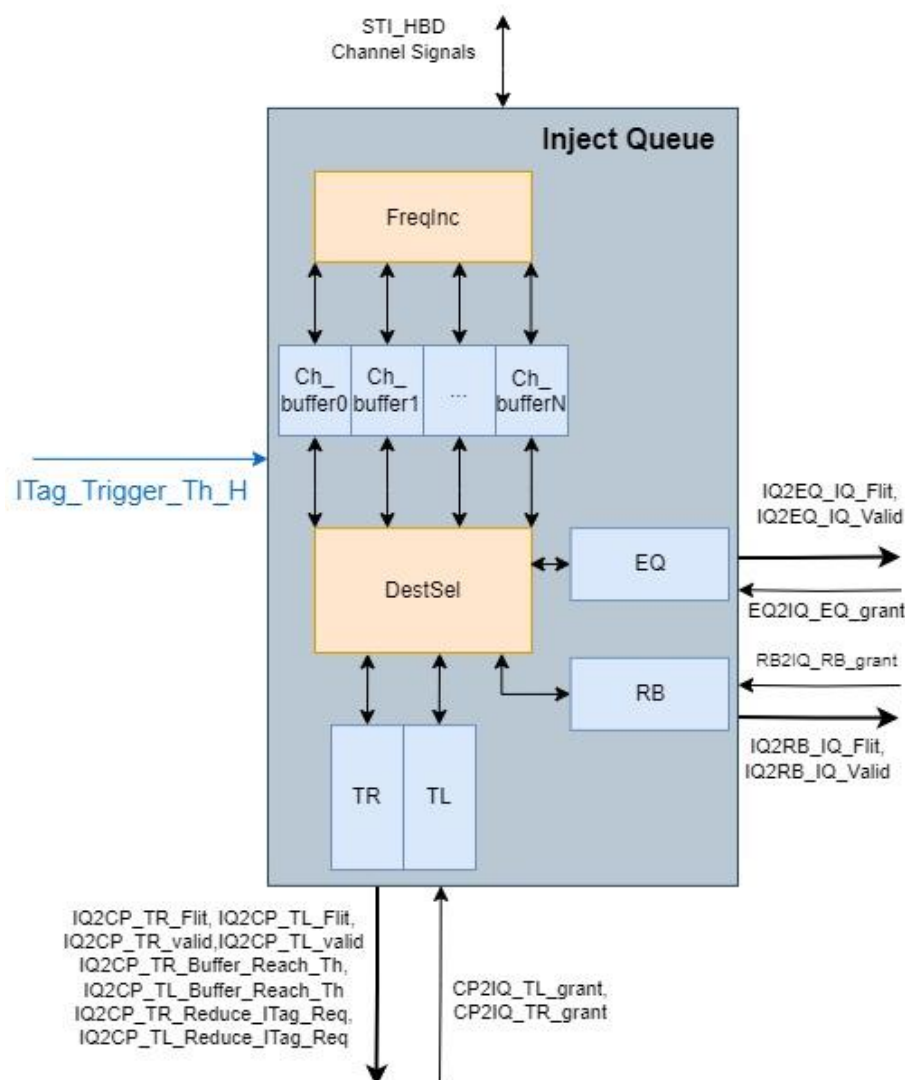


图 5-2 Inject Queue 结构图

### 5.2.1 Inject Queue Module Function

Inject Queue是从IP连接到Cross Point, Ring Bridge 以及 Eject Queue的模块, 主要功能是接收IP通过使用STI\_HBD接口的通道发出的Flit并运送至网络或本节点其他IP, 同时发送相应的控制信号配合Cross Point中的ITag模块标记或取消ITag。主要是由一个升频模块FreqInc模块, N个Channel FIFOs, 一个方向选择模块DestSel以及四个负责向不同方向发送Flit且暂存Flit的寄存模块构成。IP通过N个通道向Inject Queue的FreqInc模块发送Flit后, Flit将被以高频发送至N个Channel FIFOs中。随后Channel FIFOs内的Flit将由DestSel仲裁并判断应该进入四个方向中哪一个方向的寄存模块暂存。当Flit被发送至OUT FIFO时, 同时触发内部计数器并产生控制信号以供Cross Point判断当前是否需要标记ITag。最终根据Ring Bridge, Cross Point以及Eject Queue的控制, FIFO中的Flit将被发往相应方向。



## 5.2.2 Inject Queue Module Interface

表 5-1: IQ Module Interface

Name	Width	Direction	description
Clk	1	I	Clock
Aclk	1	I	Slower Clock
nrst	1	I	Reset (active low)
RXFLIT	512/1024	I	接收到的 Flit
RXFLITV	1	I	接收到的 Flit Valid
RXLCDRV	1	O	接收端发送的 Link Credit Valid
IQ2CP_TL_Flit	512/1024	O	IQ 发送至横向环向左的 Flit
IQ2CP_TL_Valid	1	O	IQ 发送至横向环向左的 Valid
IQ2CP_TL_Buffer_Reach_Th	1	I	IQ 触发横向环向左方向的 I-Tag 预约
IQ2CP_TL_Reduce_ITag_Req	1	I	IQ 取消横向环向左方向的 I-Tag 预约
CP2IQ_TL_Ready	1	I	IQ 接收来自横向环向左的 ready
IQ2CP_TR_Flit	512/1024	O	IQ 发送至横向环向右的 Flit
IQ2CP_TR_Valid	1	O	IQ 发送至横向环向右的 Valid
IQ2CP_TR_Buffer_Reach_Th	1	I	IQ 触发横向环向右方向的 I-Tag 预约
IQ2CP_TR_Reduce_ITag_Req	1	I	IQ 取消横向环向右方向的 I-Tag 预约
CP2IQ_TR_Ready	1	I	IQ 接收来自横向环向右的 ready
IQ2RB_IQ_Flit	512/1024	O	IQ 发送至 RB 的 Flit
IQ2RB_IQ_Flitv	1	O	IQ 发送至 RB 的 Valid
Rb2IQ_RB_Ready	1	I	IQ 接收来自 RB 的 ready
IQ2EQ_IQ_Flit	512/1024	O	IQ 发送至 EQ 的 Flit
IQ2EQ_IQ_Flitv	1	O	IQ 发送至 EQ 的 Valid
EQ2IQ_EQ_Ready	1	I	IQ 接收来自 EQ 的 ready

## 5.2.3 Inject Queue Module Implementation

Inject Queue是由一个升频模块FreqInc模块，N个Channel FIFOs，一个方向选择模块DestSel，四个负责向不同方向发送Flit且暂存Flit的FIFO构成。Flit通过credit机制从IP发送至FreqInc模块进行频率转换，随后发送至相应的Channel FIFOs。DestSel模块会对非空状态的Channel FIFOs进行方向判定以及Round-Robin仲裁，以实现N-4的转换。方向判断后，Flit将被寄存于寄存模块内。四个方向分别是 Eject Queue，Ring Bridge，横向环左行，横向环右行。其中，横向环左行，横向环右行的寄存模块除了存储Flit用的FIFO外，还包括了一组计数器用来生成该Flit的ITag相关信号，并配合Cross Point中的I-Tag模块决定何时输出Flit以及何时标记ITag。其余两者的寄存模块只含有FIFO。所有Flit离开Inject Queue是根据握手信号判断是否可以传输即可。Flit注入横向环时所生成或移除ITag的规则主要由Cross Point中的ITag控制，详见5.6。具体实现参见以下各小节。

### 5.2.3.1 FreqInc Module Implementation

由于接入Inject Queue的IP使用的频率为1GHz，而Ring Bridge与Cross Point使用的是2GHz

频率，所以Inject Queue中需进行频率转换。该模块是利用kcin\_freqdiv\_l2h\_flit实现的。具体可参见K2K Spec中的common\_freqdiv\_hxl\_v1.0.docx文档。

### 5.2.3.2 Channel FIFO Module Implementation

Channel FIFO的功能是将FreqInc发送来的Flit进行暂存，并将Flit传输至DestSel模块进行方向判断与仲裁。该模块接收FreqInc的接口使用的是credit机制，向下游DestSel使用的接口使用的是握手机制。

由于IP至IQ间的通道数为硬件可配，Channel FIFO的数量也随通道数改变。每个通道都需要一个与之相对应的FIFO。每个Channel FIFO产生的非空信号将被DestSel模块用来判断是否参与方向判断与仲裁。当某个Channel FIFO在仲裁中胜出而被选中后，FIFO内的Flit将伴随valid信号被同时发送至下游寄存器。FIFO的具体实现是使用Common IP中的common\_sync\_fifo，并将rd\_en信号作为发往上游FreqInc的LCRDV信号。

### 5.2.3.3 DestSel Module Implementation

DestSel模块实现的功能是将Channel FIFO发来的Flits通过他们的目的地ID判断应该发往IQ中三个FIFO中的其中一个。分配的方案是按照距离目的地的最短路径发送至对应方向的FIFO内。具体的方法是使用Flit的终点ID与当前节点ID进行大小比较。当终点XID (Dest\_XID) 大于，小于当前节点XID (Node\_XID) 时，分别应该进入向右寄存模块 (IQ\_OUT\_FIFOTR)，向左寄存模块 IQ\_OUT\_FIFOTL，即向右行或向左行。当终点XID (Dest\_XID) 等于当前节点XID(Node\_XID) 时，需要比较终点YID (Dest\_YID) 是否等于当前节点YID (Node\_YID)。不等于时代表Flit应传输至该节点纵向环，即发往向RB寄存模块 (IQ\_OUT\_FIFORB)。而当相等时，表示Flit应传输至该节点本身，即Flit发送至向RB寄存模块(IQ\_OUT\_FIFOEQ)。注意，只有Channel FIFO非空，即Flit有效时，方向判断结果有效。

在方向判断后，每个通道的Flit将被发送至DestSel模块中4个方向相应的Muxs。每一个Mux将根据Round-Robin竞争以及各输出的valid情况判断来自哪一个通道的输入将被传入相应的寄存模块。

### 5.2.3.4 IQ\_RB, IQ\_TL, IQ\_TR Module Implementation

IQ内的寄存模块分别是IQ\_OUT\_FIFO\_RB, IQ\_OUT\_FIFO\_EQ, IQ\_OUT\_FIFO\_TL, IQ\_OUT\_FIFO\_TR。分别对应 向RB寄存模块，向EQ寄存模块，向左寄存模块，向右寄存模块。每个寄存模块都包括了一个FIFO，其中IQ\_OUT\_FIFO\_TL, IQ\_OUT\_FIFO\_TR寄存模块还包括了一组用来提供ITag相关信号的counter。

四个模块中的FIFO可复用标准common module中的FIFO。但由于IQ\_Counters\_TL和IQ\_Counters\_TR 两组 counters 都需要与其相对应的 FIFO 模块中的 WtPtr 和 RdPtr，IQ\_OUT\_FIFO\_TR, IQ\_OUT\_FIFO\_TL两者的FIFO内信号WtPtr和RdPtr需要从FIFO中被引出。

每组IQ\_Counters的数量应该与相对应的FIFO深度一致。当FIFO接收到Push信号并接收新的Flit时，对应Counter的Enable寄存器也将被置于1，Counter被重置为0。每一组counter都会根据相应的FIFO所输出的WtPtr（写指针）去寻找其对应的Counter。而当ITag\_Ctrl生成Ready信号，即FIFO中的Flit需要被Pop出去时，该信号也会被IQ\_Counters接收，并且根据相应FIFO输出的RdPtr（读指针）将对应的Enable寄存器置于0，Counter被重置为0。计数器始终会将它的输出和相应Enable寄存器内的值进行相加作为下一个周期它的输出。所有计数器的输出结果都将被进行比较。如果其相应的Enable为1且计数器输出等于ITag\_Trigger\_Th，将触发Buffer\_Reach\_Th。如果此时接收到Ready信号，并且被读指针指向的计数器满足以上条件或者Enable为0时，则会额外触发Reduce\_ITag\_Req信号。这两个信号都会发送给Cross Point中的ITag模块，供其进行判断。

## 5.3 Eject Queue

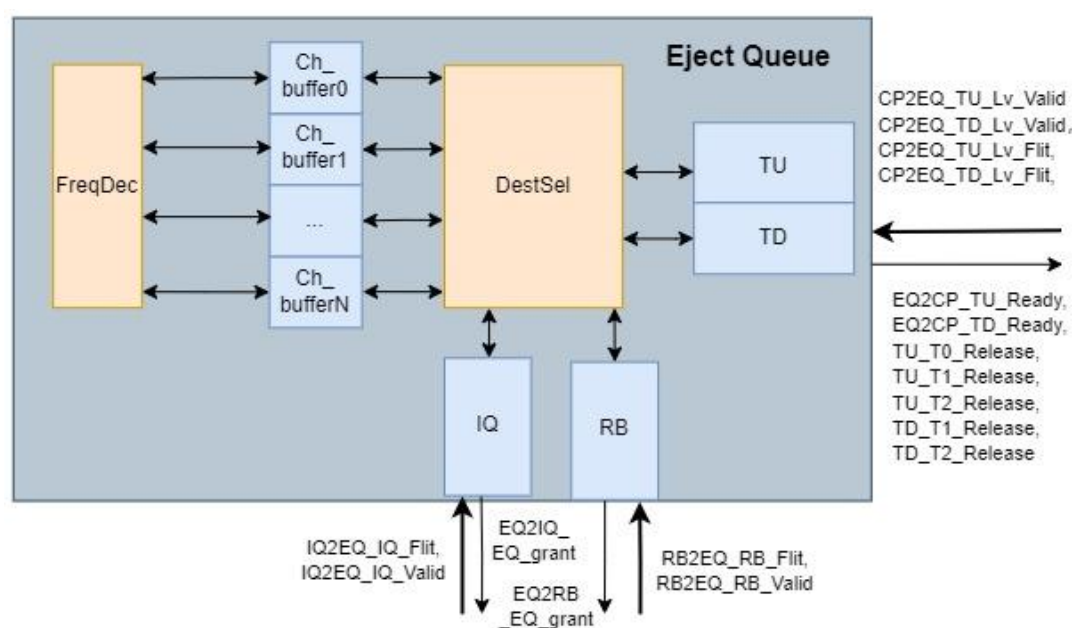


图 5-3 Eject Queue 结构图

### 5.3.1 Eject Queue Module Function

Eject Queue是从Cross Point和Ring Bridge连接到IP的模块，由四组FIFO，一个DestSel模块，N个Channel FIFO以及一个FreqDec模块组成。其功能与Inject Queue相反，主要是将网络中的Flit通过通道发送至Flit目的地IP。

### 5.3.2 Eject Queue Module Interface

表 5-2: EQ Module Interface

Name	Width	Direction	description
Clk	1	I	Clock
Aclk	1	I	Slower Clock
nrst	1	I	Reset (active low)
TXFLIT	512/1024	O	发送的 Flit
TXFLITV	1	O	发送的 Flit Valid
TXLCRDV	1	I	发送端接收的 Link Credit Valid
CP2EQ_TU_Lv_Flit	512/1024	I	纵向环向上发送至 EQ 的 Entry 使用等级及 Flit
CP2EQ_TU_Lv_Valid	1	I	纵向环向上发送至 EQ 的 Valid
EQ2CP_TU_Ready	1	O	纵向环向上接收到来自 EQ 的 Ready
CP2EQ_TD_Lv_Flit	512/1024	I	纵向环向下发送至 EQ 的 Entry 使用等级及 Flit
CP2EQ_TD_Lv_Valid	1	I	纵向环向下发送至 EQ 的 Valid
EQ2CP_TD_Ready	1	O	纵向环向下接收到来自 EQ 的 Ready
TU_T0_Release	1	O	纵向环向上 T0 等级 Entry 被释放
TU_T1_Release	1	O	纵向环向上 T1 等级 Entry 被释放
TU_T2_Release	1	O	纵向环向上 T2 等级 Entry 被释放
TD_T1_Release	1	O	纵向环向下 T1 等级 Entry 被释放
TD_T2_Release	1	O	纵向环向下 T2 等级 Entry 被释放
IQ2EQ_IQ_Flit	512/1024	I	IQ 发送至 EQ 的 Flit
IQ2EQ_IQ_Valid	1	I	IQ 发送至 EQ 的 Valid
EQ2IQ_EQ_Ready	1	O	IQ 接收到来自 EQ 的 Ready
RB2EQ_RB_Flit	512/1024	I	RB 发送至 EQ 的 Flit
RB2EQ_RB_Valid	1	I	RB 发送至 EQ 的 Valid
EQ2RB_EQ_Ready	1	O	RB 接收到来自 EQ 的 Ready

### 5.3.3 Eject Queue Module Implementation

虽然 Eject Queue 结构与 Inject Queue 相似，但由于 EQ 中并不需要支持 ITag 相关信号，所以 EQ 中 TU,TD,RB 并非 IQ 中的寄存模块。由于 EQ 需要支持所连纵向环 Cross Point 中的 E-Tag，此处的 TU,TD 除了 ready 和 flitv 组成的握手信号外，还需要输出两个方向各个等级 Entry 的释放信号以供上游 Cross Point 中的 E-Tag 使用（即 TX\_Tx\_Release）。该信号的判断是根据相应的 Lv\_Flit 信号的前两位进行确认。当这两位为 2'b00, 2'b10, 2'b11 时，分别对应相应方向的 T2\_Release, T1\_Release, T0\_Release 拉高。值得注意的是，TD 方向没有 T0 Entry，故没有 TD\_T2\_Release 信号。

四组FIFO分别对应不同的数据源头：Ring Bridge，纵向环上行，纵向环下行。四组FIFO分别命名为EQ\_FIFO\_IN\_RB, EQ\_FIFO\_IN\_IQ, EQ\_FIFO\_IN\_TU(Towards Up), EQ\_FIFO\_IN\_TD(Towards Down)。Flit的发射规则主要由Cross Point中的E-Tag模块控制，详见5.6。Flit在FIFO中停留后，会等待下游FIFO有空位时根据DestSel模块的分配进入相应的Channel FIFO。

Eject Queue的DestSel模块需要判断哪一组寄存器的输出应该被传送至IP方向的哪一个Channel FIFO内。其中的功能可被分为两部分：方向判断以及Channel FIFO的分配。方向判断是由4组1-N Muxs实现,其控制信号是根据Flit中TgtID内的DeviceID进行区分。而Channel FIFO的分配是由N组4-1 Muxs实现，每一个Mux将根据Round-Robin竞争以及各输出的valid情况判断当下时钟周期Channel FIFO的使用权应该分配给哪一组寄存器。

最后，寄存在Channel FIFO内的Flit会被FreqDec模块降频并发送至相应IP。与Inject Queue相似，该模块是利用kcin\_freqdiv\_h2l\_flit实现的。

## 5.4 Ring Bridge

图 5-1 展示了 Ring Bridge 的结构图。

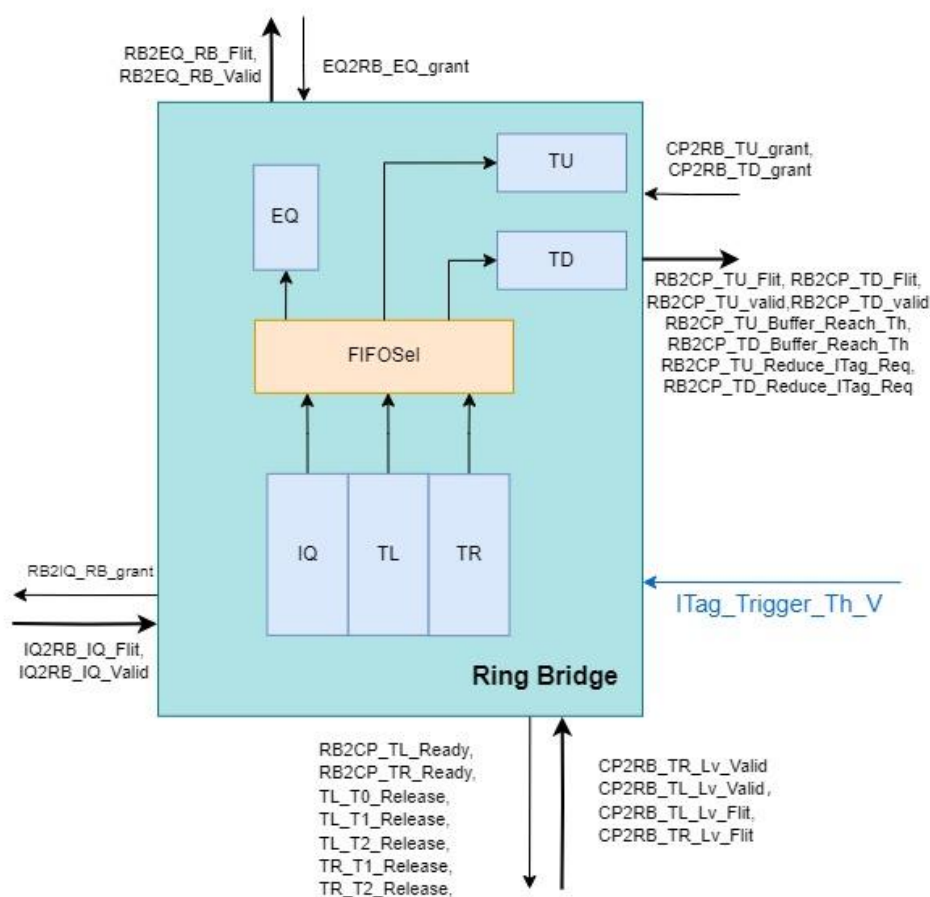


图 5-4 Ring Bridge 结构图

### 5.4.1 Ring Bridge Module Function

Ring Bridge是帮助Flit从横向环或者Inject Queue转向至纵向环或Eject Queue的模块，其中不包含Inject Queue向Eject Queue转向的通路。Ring Bridge主要由四组FIFO，三个不同方向的寄存模块, Muxs以及FIFOSel组成。三个方向的寄存模块用来分别暂存不同来源或满足优先条件

下进入Ring Bridge的Flits。RB中的寄存模块与IQ中的寄存模块相同。FIFOsel模块根据四组FIFO的竞争结果决定优先选择哪一组FIFO输出至哪一个寄存模块中。相应判断信号将被传入Muxs模块供其选择输入输出。最后，寄存模块将输出Flit至纵向环或Eject Queue，并向Cross Point提供ITag相关的信号。

## 5.4.2 Ring Bridge Module Interface

表 5-3: RB Module Interface

Name	Width	Direction	description
Clk	1	I	Clock
Aclk	1	I	Slower Clock
nrst	1	I	Reset (active low)
IQ2RB_IQ_Flit	512/1024	I	IQ 发送至 RB 的 Flit
IQ2RB_IQ_Valid	1	I	IQ 发送至 RB 的 Valid
RB2IQ_RB_Ready	1	O	IQ 接收来自 RB 的 Ready
RB2EQ_RB_Flit	512/1024	O	RB 发送至 EQ 的 Flit
RB2EQ_RB_Valid	1	O	RB 发送至 EQ 的 Valid
EQ2RB_EQ_Ready	1	I	RB 接收来自 EQ 的 Ready
CP2EQ_TULv_Flit	514/1026	I	横向环向左发送至 RB 的 Entry 使用等级及 Flit
CP2EQ_TULv_Valid	1	I	横向环向左发送至 RB 的 Valid
RB2CP_TL_Ready	1	O	横向环向左接收来自 RB 的 Ready
CP2RB_TR_Lv_Flit	514/1026	I	横向环向右发送至 RB 的 Entry 使用等级及 Flit
CP2RB_TR_Lv_Valid	1	I	横向环向右发送至 RB 的 Valid
RB2CP_TR_Ready	1	O	横向环向右接收来自 RB 的 Ready
TL_T0_Release	1	O	横向环向左 T0 等级 Entry 被释放
TL_T1_Release	1	O	横向环向左 T1 等级 Entry 被释放
TL_T2_Release	1	O	横向环向左 T2 等级 Entry 被释放
TR_T1_Release	1	O	横向环向右 T1 等级 Entry 被释放
TR_T2_Release	1	O	横向环向右 T2 等级 Entry 被释放
RB2CP_TU_Flit	512/1024	O	RB 发送至纵向环向上的 Flit
RB2CP_TU_Valid	1	O	RB 发送至纵向环向上的 Valid
CP2RB_TU_Ready	1	I	RB 发送至纵向环向上的 Ready
RB2CP_TD_Flit	512/1024	O	RB 发送至纵向环向下的 Flit
RB2CP_TD_Valid	1	O	RB 发送至纵向环向下的 Valid
CP2RB_TD_Ready	1	I	RB 发送至纵向环向下的 Ready
RB2CP_TU_Buffer_Reach_Th	1	O	RB 触发纵向环向上方向的 I-Tag 预约
RB2CP_TD_Buffer_Reach_Th	1	O	RB 触发纵向环向下方向的 I-Tag 预约
RB2CP_TU_Reduce_ITag_Req	1	O	RB 取消纵向环向上方向的 I-Tag 预约
RB2CP_TD_Reduce_ITag_Req	1	O	RB 取消纵向环向下方向的 I-Tag 预约

## 5.4.3 Ring Bridge Module Implementation

Ring Bridge的输入主要是由横向环的E-Tag模块和Inject Queue的IQ\_FIFO\_RB控制，输出

则主要是由纵向环的I-Tag模块和Eject Queue的IQ\_FIFO\_RB控制。Ring Bridge IN FIFO, Ring Bridge OUT FIFO 和 Ring Bridge FIFOSel 将分别被讨论。

#### 5.4.3.1 Ring Bridge IN FIFO Module Implementation

Ring Bridge In FIFO中的三组FIFO分别命名为RB\_IN\_FIFO\_IQ, RB\_IN\_FIFO\_TL(Towards Left), RB\_IN\_FIFO\_TR(Towards Right)。三组FIFO分别对应三种不同的数据源头: Inject Queue, 横向环左行, 横向环右行。其中RB\_FIFO\_TL(Towards Left) FIFO的Flit来自与横向环左行链路。所有FIFO都可以将寄存的Flit发送至纵向环上行, 纵向环下行和Eject Queue三个方向的OUT FIFO中。而RB\_FIFO\_IQ只能向纵向环上下行两个方向的OUT FIFO发送Flit。三组FIFO在输出寄存的Flit时, 会根据Ring Bridge FIFOSel模块发出的控制信号使Flit进入相应的寄存模块中。

另外, TL/ TR IN FIFO还需要输出两个方向各个等级Entry的释放信号以供上游Cross Point中的E-Tag使用 (即TX\_Tx\_Release)。该信号的判断是根据相应的Lv\_Flit信号的前两位进行确认。当这两位为2'b00, 2'b10, 2'b11时, 分别对应相应方向的T2\_Release, T1\_Release, T0\_Release拉高。值得注意的是, TR方向没有T0 Entry, 故没有TR\_T2\_Release信号。

#### 5.4.3.2 Ring Bridge OUT FIFO Modules Implementation

Ring Bridge 中的三个 OUT FIFO 分别是 RB\_OUT\_FIFO\_EQ, RB\_OUT\_FIFO\_TU (Towards Up), RB\_OUT\_FIFO\_TD (Towards down)。和 IQ 中的 IQ 的寄存模块相同, RB\_OUT\_FIFO\_TU, RB\_OUT\_FIFO\_TD 除了内部的 FIFO 以外还各有一组 counters。而 RB\_OUT\_FIFO\_EQ 由单个 FIFO 组成。具体 Ring Bridge OUT FIFO 的行为可参照 5.2.3.4。

各个 OUT FIFO 在接收到从 IN FIFO 传来的 Flit 后暂时寄存, 等待纵向环 Cross Point 的 I-Tag 模块或者 Eject Queue 发来的 grand 信号 (EQ2RB\_EQ\_Ready) 后将 Flit 传出至相应的方向。

#### 5.4.3.3 Ring Bridge FIFOSel Modules Implementation

FIFOSel 模块的功能是将 Ring Bridge 中的 3 个 IN FIFO 根据其当前输出的 Flit 决定应该具体发向哪个方向的模块。FIFOSel 模块根据 Ring Bridge 中的 3 组 IN FIFO 的空满情况进行判断。随后, FIFOSel 模块中的 Mux 会根据 FIFOSel 模块输出的控制信号选择哪个 Ring Bridge IN FIFO 中的 Flit 应该发往哪个 Ring Bridge OUT FIFO。

Ring Bridge 共有 3 个 Flit 输出的方向, 分别是纵向环向上, 纵向环向下, Eject Queue 中的 EQ\_IN\_FIFO\_RB。FIFOSel 模块中有判断优先级的模块。根据优先级, FIFOSel 依次判断三个输出寄存模块应该被哪个 IN FIFO 占用。如果某一个 IN FIFO 中没有有效 Flit 需要从该方向输出, 那么该方向的 valid 信号为 0, 则不参与该方向的竞争。反之, valid 为 1 时将参与竞争。如果某一个输出方向中只有一组对应的 IN FIFO, 那么该 IN FIFO 中的一个 Flit 将从此方向传出。如果当下该方向中有两个或者多个 IN FIFO 竞争, 几组 IN FIFO 将通过 Round-Robin 的竞争方式决定谁拥有输出的优先权。即如果某个 IN FIFO 刚刚输出 Flit, 则该 IN FIFO 将被置为优先级队列的最后位。三组 IN FIFO 在 Flit 从 IN FIFO 传至 Ring Bridge OUT FIFO 后更新 FIFOSel 模块中的三组 IN FIFO 的优先顺序。



当OUT FIFO可以接收IN FIFO输出的Flit时，该OUT FIFO会向FIFOSel模块发送相应的Ready信号。反之，FIFOSel模块会等待OUT FIFO直到其有剩余空位。FIFOSel将通过方向竞争判断告知每个IN FIFO当前是否输出Flit以及相应的输出方向。

5.5 Ring Slice

5.5.1 Ring Slice Module Function

Ring Slice是组成环路的最基本单元。每一个Ring Slice是由若干个寄存器构成，具体数量由其位宽决定。其功能为暂存环上的Flit并在下一周期发向下游。

5.5.2 Ring Slice Module Interface

表 5-4: Ring Slice Module Interface

Name	Width	Direction	description
Clk	1	I	Clock
nrst	1	I	Reset (active low)
Slot_In_TL(TU)	见表 5-5	I	Slot 向左（上）方向输入
Slot_In_TR(TD)	见表 5-5	I	Slot 向右（下）方向输入
Slot_Out_TL(TU)	见表 5-5	O	Slot 向左（上）方向输出
Slot_Out_TR(TD)	见表 5-5	O	Slot 向右（下）方向输出

5.5.3 Ring Slice Module Implementation

由于Ring Slice的功能简单，内部只由若干位的寄存器构成。具体位数由ETag信号位宽（2位），ITag信号位宽（与网络节点数相关），Status位宽（1位），Flit位宽决定，具体可参见表 5.5。该表以环上最多拥有16个节点为例。每个Slice之间传递的数据被称为Slot。除了载有Flit中的信号外，Ring Slot中还保存着其他Status和Tag信号。Status信号指Valid信号，表示该Slot中的Flit是否有效。当Slot为空时valid信号为0，反之则为1。Tag信号包括I-Tag信号以及E-Tag信号。I-Tag信号代表了该Slot是否被某个I-Tag模块预留。有预留时I-Tag最高位置1。0XXXX为无预留。次高位表示表示预约方向，0为向左（或向上），1为向右（或向下）。剩余位表示的是预约节点在该环上相应的节点XID或YID。E-Tag信号代表了该Slot是否已经通过某个E-Tag模块在相应FIFO中预留，最高位表示是否标记E-Tag，1为标记，0为未标记，即T2优先级。当标记时，余位为0表示T1优先级，余位为1表示T0优先级。具体关于优先级的内容可参见 5.6 小节关于E-Tag的部分。

表 5-5 Ring Slot的位域关系（FW此处表示FLIT\_WIDTH）

ETag 2 bit [FW+7:FW+6]	ITag 2+log2(相应方向 网络节点数) bit [FW+5 : FW+1]	Valid 1bit [FW]	Flit [FLIT_WIDTH-1:0]
---------------------------	--	--------------------	--------------------------



除Cross Point中的Ring Slice, 每个时钟周期后上游的Ring Slice中的Slot将会传递给相邻的下游Ring Slice中。沿途的Cross Point将通过传递到其内部的Ring Slice中的Slot Status和Tag信号判断该Slot中的Flit应该被如何传输或操作。

## 5.6 Cross Point

图 5-5 展示了横向环上 Cross Point 的结构图。

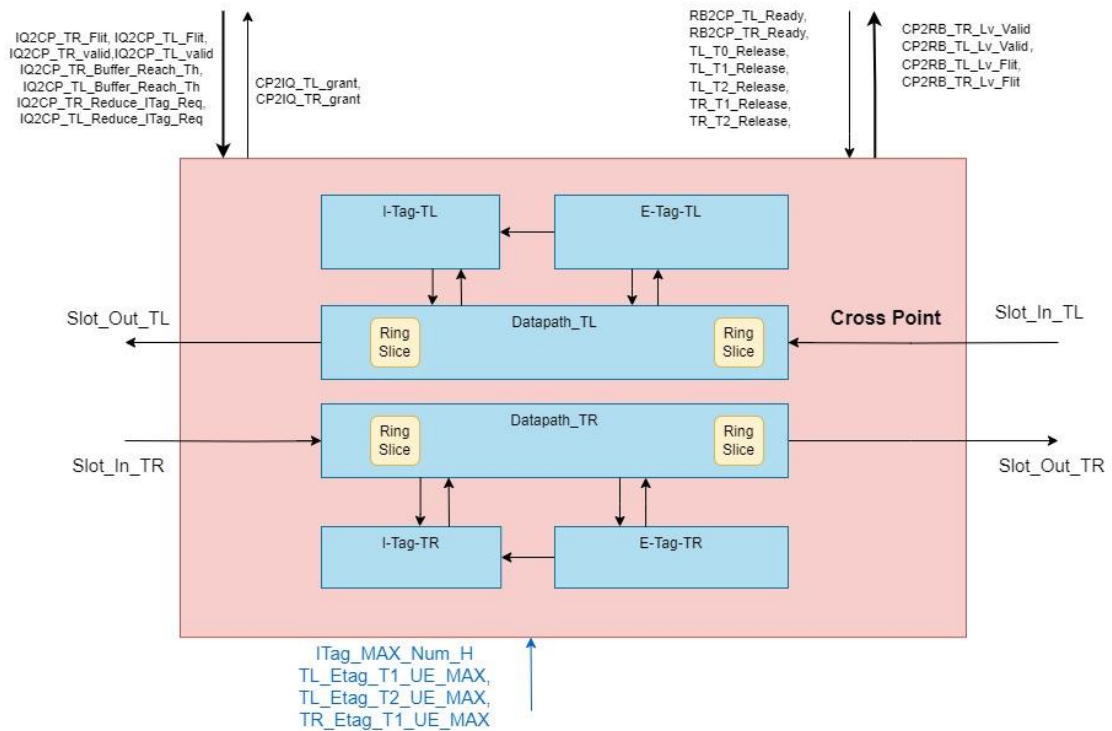


图 5-5 横向环 Cross Point 结构图

### 5.6.1 Cross Point Module Function

Cross Point是Cross Ring中的重要组成部分，连接了Inject Queue，Ring Bridge，Eject Queue和环路。主要功能是根据Slot Status，Slot Tag以及IQ、EQ、RB的状态决定各Flit何时以及如何上进行上环或下环。其中，I-Tag模块和E-Tag模块分别控制了上环和下环的逻辑判断与控制，同时也用来实现Cross Ring中的防饿死机制。Datapath内部连接了两个Ring Slice，并根据控制信号传输IQ、EQ、RB内或环上Flit。各模块具体功能参见各小节。

#### 5.6.1.1 I-Tag Module Function

I-Tag模块用于判定何时需要生成I-Tag在对应的Ring Slot上进行标记，以及何时上环。它由ITag\_Ctrl, Tagged\_Counter, ITag\_Req\_Counter三个模块组成。ITag\_Ctrl生成上环或标记ITag的相关信号，Tagged\_Counter用来记录环上被该I-Tag预约的Slot数，ITag\_Req\_Counter用来记录当前上游FIFO所需的I-Tag预约数。

该模块的功能行为如下：如果Datapath上游中有空置的Ring Slot或是该Slot刚好需要下环，则将RB或者IQ内FIFO的Flit在下一个时钟周期中注入该Slot。当RB或者IQ在一定时间内无法向下游纵向环或者横向环发送Flit时，I-Tag模块会在当前可预约的Ring Slot上标记I-Tag，提前预定该Ring Slot。被标记的Ring Slot在卸下原本载有的Flit后将不会运载其他的Flit，直到该Ring Slot重新回到被标记时的Cross Point，相应的Ring Bridge FIFO会将Flit输出到Ring Slot上。随后Ring Slot上的I-Tag将被取消。Flit即使提前预定过I-Tag，也仍可以提前通过其他Slot上环。这种情况则需要后续取消Slot上多余的I-Tag。

### 5.6.1.2 E-Tag Module Function

E-Tag模块用于判定Slot中的Flit当下是否可以下环，以及不可以下环时对该Slot未来优先下环的准备工作。该模块主要用来支持E-Tag下环机制。

E-Tag下环机制是指Flit在通过当前优先级无法下环时对该Flit提升下环优先级。优先级的提升可以增加Flit可使用的下环FIFO Entry范围，从而可以有更大的下环概率。其中T0级（最高优先级）会额外需要根据Round-Robin结果选出下一个需要下环的T0级Flit。

### 5.6.1.3 Datapath Module Function

Datapath模块用于传输Flit从上游FIFO以及上游Ring Slice到下游FIFO以及下游Ring Slice中，同时根据该方向I-Tag模块以及双向E-Tag模块产生的控制信号决定下游Ring Slice接收的Status和Tag信号。

## 5.6.2 Cross Point Module Interface

以下为横向环 Cross Point 对应输入输出信号名：

表 5-6: Cross Point Module Interface

Name	Width	Direction	description
Clk	1	I	Clock
Aclk	1	I	Slower Clock
nrst	1	I	Reset (active low)
Slot_In_TL	见表 5-5	I	Slot 向左方向输入
Slot_In_TR	见表 5-5	I	Slot 向右方向输入
Slot_Out_TL	见表 5-5	O	Slot 向左方向输出
Slot_Out_TR	见表 5-5	O	Slot 向右方向输出
IQ2CP_TR_Flit	512/1024	I	IQ 发送至横向环向右的 Flit
IQ2CP_TR_Valid	1	I	IQ 发送至横向环向右的 Valid
CP2IQ_TR_Ready	1	O	IQ 接收来自横向环向右的 Valid
IQ2CP_TR_Buffer_Reach_Th	1	I	IQ 发送至横向环向右的 I-Tag 预约触发信号
IQ2CP_TR_Reduce_ITag_Req	1	I	IQ 发送至横向环向右的 I-Tag 预约取消信号
IQ2CP_TL_Flit	512/1024	I	IQ 发送至横向环向左的 Flit
IQ2CP_TL_Valid	1	I	IQ 发送至横向环向左的 Valid
CP2IQ_TL_Ready	1	O	IQ 接收来自横向环向左的 Valid

Name	Width	Direction	description
IQ2CP_TL_Buffer_Reach_Th	1	I	IQ 发送至横向环向左的 I-Tag 预约触发信号
IQ2CP_TL_Reduce_ITag_Req	1	I	IQ 发送至横向环向左的 I-Tag 预约取消信号
CP2RB_TL_Lv_Flit	514/1026	O	横向环向左发送至 RB 的 Entry 使用等级及 Flit
CP2RB_TL_Lv_Valid	1	O	横向环向左发送至 RB 的 Valid 信号
RB2CP_TL_Ready	1	I	横向环向左接收来自 RB 的 Ready 信号
TL_T0_Release	1	I	横向环向左 T0 等级 Entry 被释放
TL_T1_Release	1	I	横向环向左 T1 等级 Entry 被释放
TL_T2_Release	1	I	横向环向左 T2 等级 Entry 被释放
CP2RB_TR_Lv_Flit	514/1026	O	横向环向右发送至 RB 的 Entry 使用等级及 Flit
CP2RB_TR_Lv_Valid	1	O	横向环向右发送至 RB 的 Valid 信号
RB2CP_TR_Ready	1	I	横向环向右接收来自 RB 的 Ready 信号
TR_T1_Release	1	I	横向环向右 T1 等级 Entry 被释放
TR_T2_Release	1	I	横向环向右 T2 等级 Entry 被释放
ITag_Max_Num_H	4	I	I-Tag 横向环最大预留数
TL_ETag_T1_UE_MAX	见表 1-3	I	横向环向左 T1 级 ETag 可用 FIFO Entry 数
TL_ETag_T2_UE_MAX	见表 1-3	I	横向环向左 T2 级 ETag 可用 FIFO Entry 数
TR_ETag_T1_UE_MAX	见表 1-3	I	横向环向右 T1 级 ETag 可用 FIFO Entry 数

以下为纵向环 Cross Point 对应输入输出信号名：

Name	Width	Direction	description
Clk	1	I	Clock
Aclk	1	I	Slower Clock
nrst	1	I	Reset (active low)
Slot_In_TU	见表 5-5	I	Slot 向上方向输入
Slot_In_TD	见表 5-5	I	Slot 向下方向输入
Slot_Out_TU	见表 5-5	O	Slot 向上方向输出
Slot_Out_TD	见表 5-5	O	Slot 向下方向输出
RB2CP_TD_Flit	512/1024	I	RB 发送至纵向环向下的 Flit
RB2CP_TD_Valid	1	I	RB 发送至纵向环向下的 Valid
CP2RB_TD_Ready	1	O	RB 接收来自纵向环向下的 Valid
RB2CP_TD_Buffer_Reach_Th	1	I	RB 发送至纵向环向下的 I-Tag 预约触发信号
RB2CP_TD_Reduce_ITag_Req	1	I	RB 发送至纵向环向下的 I-Tag 预约取消信号
RB2CP_TU_Flit	512/1024	I	RB 发送至纵向环向上的 Flit
RB2CP_TU_Valid	1	I	RB 发送至纵向环向上的 Valid
CP2RB_TU_Ready	1	O	RB 接收来自纵向环向上的 Valid
RB2CP_TU_Buffer_Reach_Th	1	I	RB 发送至纵向环向上的 I-Tag 预约触发信号
RB2CP_TU_Reduce_ITag_Req	1	I	RB 发送至纵向环向上的 I-Tag 预约取消信号
CP2EQ_TU_Lv_Flit	514/1026	O	纵向环向上发送至 EQ 的 Entry 使用等级及 Flit
CP2EQ_TU_Lv_Valid	1	O	纵向环向上发送至 EQ 的 Valid 信号
RB2CP_TU_Ready	1	I	纵向环向上接收来自 EQ 的 Ready 信号
TU_T0_Release	1	I	纵向环向上 T0 等级 Entry 被释放
TU_T1_Release	1	I	纵向环向上 T1 等级 Entry 被释放

Name	Width	Direction	description
TU_T2_Release	1	I	纵向环向上 T2 等级 Entry 被释放
CP2EQ_TD_Lv_Flit	514/1026	O	纵向环向下载送至 EQ 的 Entry 使用等级及 Flit
CP2EQ_TD_Lv_Valid	1	O	纵向环向下载送至 EQ 的 Valid 信号
EQ2CP_TD_Ready	1	I	纵向环向下接收来自 EQ 的 Ready 信号
TD_T1_Release	1	I	纵向环向下 T1 等级 Entry 被释放
TD_T2_Release	1	I	纵向环向下 T2 等级 Entry 被释放
ITag_Max_Num_V	4	I	I-Tag 纵向环最大预留数
TU_ETag_T1_UE_MAX	见表 1-3	I	纵向环向左 T1 级 ETag 可用 FIFO Entry 数
TU_ETag_T2_UE_MAX	见表 1-3	I	纵向环向左 T2 级 ETag 可用 FIFO Entry 数
TD_ETag_T1_UE_MAX	见表 1-3	I	纵向环向右 T1 级 ETag 可用 FIFO Entry 数

### 5.6.3 Cross Point Module Implementation

Cross Point 主要由 Datapath 模块，I-Tag 模块和 E-Tag 模块组成。Datapath 内连接了两个 Ring Slice。Ring Slice 的实现请参见 5.5.3。其他各模块详解下文各小节。

#### 5.6.3.1 I-Tag Module Implementation

I-Tag 是通过提前预约当前 Slot 使该 Slot 稍后只能暂时被该 Cross Point 的当前方向使用的机制。I-Tag 模块不但用来生成与移除 I-Tag，同时也负责判断 IQ 或 RB 中的 Flit 何时上环。I-Tag 模块主要由 ITag\_Ctrl, Tagged\_Counter, ITag\_Req\_Counter 三个模块组成。其中 ITag\_Ctrl 是 I-Tag 模块中的核心，负责根据外部输入和内部两个计数器判断输出的模块，Tagged\_Counter 是用来记录环上当前有多少被该 I-Tag 模块标记的 Slot，ITag\_Req\_Counter 是用来记录上游 FIFO 中有多少 Flits 需要被标记 ITag。下文将详细说明 I-Tag 模块的设计。

##### 5.6.3.1.1 ITag\_Ctrl Module Implementation

当 IQ\_FIFO\_TX\_valid (X 为方向) 或 RB\_FIFO\_TX\_valid 信号为 1 时，代表有数据存储在 IQ 或 RB 中相应的 FIFO 内。I-Tag 开始进行判断。

当 I-Tag 模块读到 Datapath 上游的 Ring Slice 内 Flit 的 Valid 信号为 0 或者 E-Tag 模块的 Eject 信号为 1 时，表示该 Slot 当前或即将空载。此时，当 ITag-bits=0 时，即该 Slot 未被预约，FIFO 中的 Flit 可以准备进入该 Slot 上环，即生成 Ready 信号。反之，该 Slot 已被预约。I-Tag 模块此时则需要根据 ITag-bits 来判断预约者是否是该 I-Tag 模块自己。如果 ITag-bits 和该 I-Tag 模块的 I-Tag ID 相同，则为自己提前预留的 slot。这种情况下，上游 FIFO 内的 Flit 可准备进入该 Slot，清除 Slot 中的 ITag-bits 并减少 Tagged Counter 中的值，即生成 ITag\_SlotRemove, Ready, Tagged\_Counter\_Dec 信号。如果 ITag-bits 和该 I-Tag 模块的 I-Tag ID 不同，则为其他 I-Tag 模块预留的 slot。这种情况下 Flit 不被允许进入该 Slot。

当 I-Tag 模块读到上游的 Ring Slice 内 Flit 的 Valid 信号和 E-Tag 模块的 Eject 信号分别不为 0 和 1 时，表示此时 Slot 中存在有效 Flit 并且该 Flit 将继续绕环。在此情况下，如果 Slot 中的 ITag-bits 为 0，

表示该Slot可以被预留。ITag\_Ctrl将检查Tagged\_Counter是否小于ITag\_Max\_Num以及ITag\_Req\_Counter, 该检查将输出ITag\_Request信号。如果满足以上条件, 则预留该Slot, 即生成ITag\_Insert信号并增加Tagged Counter中的值。如果不满足, 则说明当前标记ITag数量达到上限或者上游FIFO中当前所有的ITag需求都已经被满足。如果Slot中的ITag-bits不为0, 则表示该Slot已被预留。如果ITag-bits与该I-Tag模块中的ID不匹配时, 该预留一定来自其他ITag模块, 那么该Slot不应被操作。反之, 如果匹配, 则该预留一定来自于该I-Tag模块。此时ITag\_Ctrl将检查Tagged\_Counter值是否大于ITag\_Req\_Counter, 该检查将输出Overtagged信号。大于则说明之前存在着预留ITag但最终通过无预约的普通方式上环的Flit。这种情况下, 需要撤回多余的ITag以保证下游其他I-Tag模块在该Slot中的Flit下环后仍能够使用该Slot, 而不是完整的绕环回到该I-Tag。

注意, ITag\_Ctrl的判断信号是根据Datapath模块的上游Ring Slice生成的。这些信号将被寄存于ITag\_Ctrl并直接在下一时钟周期发送给Datapath模块的下游Ring Slice。

#### 5.6.3.1.2 Tagged Counter Module Implementation

Tagged Counter记录着已经在环上标记过的ITag的数量。其增减完全由ITag\_Ctrl生成的Tagged\_Counter\_Inc和Tagged\_Counter\_Dec信号控制。其输出由ITag\_Ctrl模块接收并生成其他判断信号。

#### 5.6.3.1.3 ITag\_Req\_Counter Module Implementation

ITag\_Req\_Counter记录着上游FIFO中申请ITag的数量。当上游FIFO相关联的计数器模块发送Buffer\_Reach\_Th信号, 即表示FIFO中某个Flit已经到达了请求生成ITag的临界值, ITag\_Req\_Counter准备增加。当上游FIFO相关联的计数器模块发送Reduce\_ITag\_Req信号, 即表示FIFO中最先进入FIFO的Flit满足了请求生成ITag的临界值。如果此时ITag\_Ctrl发出了Ready的信号, 那么说明该Flit准备上环, 并且ITag\_Req\_Counter曾经或正在为该Flit增加ITag\_Req\_Counter (记录该Flit的ITag申请)。这种情况下, ITag\_Req\_Counter准备减少。结合以上判断结果, ITag\_Req\_Counter将决定内部数值+1, -1或不变。

Buffer\_Reach\_Th信号和Reduce\_ITag\_Req信号的生成已经在5.2.3.4中进行了详细介绍。

#### 5.6.3.2 E-Tag Module Implementation

E-Tag模块是由E-Tag下环机制构成。E-Tag机制是通过提升下环优先等级使得环上陈旧的Flit可以比新Flit更容易下环。E-Tag模块负责了所有下环相关的控制。该模块共有4个子模块: 用来生成控制信号的ETag\_Ctrl模块, 记录并仲裁T0级Slot的T0 Entry\_Allocator, 输出当前Slot ID的Order Counter, 以及记录所有优先等级当前可使用FIFO Entry的UE\_Counters模块。

各个等级可使用的FIFO深度范围如图5-6所示, 向左或向上方向下环的下游FIFO存在三种可使用范围, 即T2可使用范围, T1可使用范围, T0可使用范围。T2级Slot的Flit仅可使用公用的Entry进行下环。T1级Slot的Flit可使用T0和T1共用的Entry以及公用的Entry下环。而T0级Slot的

Flit可使用所有Entry下环。向右或向下方向下环的下游FIFO只存在两种可使用范围，即T2可使用范围，T1可使用范围。由于这一侧不支持T0级，所以T1可使用范围即是整个FIFO的范围。

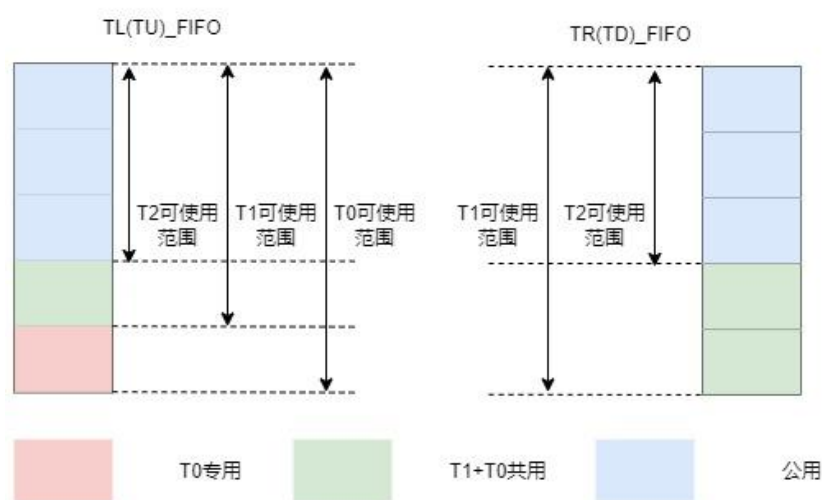


图 5-6 E-Tag 各等级可使用 Entry 范围

当环路上的Slot第一次抵达目标节点的Cross Point后，由于处于下游的Ring Bridge或者Eject Queue中留给T2级Flit下环的输入FIFO已满，该Slot中的Flit无法离开环路。此时，Cross Point会使用E-Tag机制为该Slot提高优先级至T1级，即E-Tag位置为10。该操作可以允许该Slot再次绕回至该Cross Point时，可以使用留给T1级下环的FIFO Entry。当该Slot以T1级回到该Cross Point的向左（或向上）方向依然无法下环时，将被标记为T0级。与此同时，该Slot的ID将被记录在T0\_Etag\_Order\_FIFO中。随后，每当该Slot到达该节点向左（或向上）方向时，Flit将以T0级尝试下环。而由于只有向左（或向上）方向支持T0优先级。当该Slot到达该节点向右（或向下）方向时，将仍然以T1级尝试下环。当Slot中的Flit尝试以T0优先级下环时，需要根据E-Tag模块中所有T0级Slot的Round-Robin仲裁结果为是否下环的判断依据。

当仲裁结果等于当前Slot的ID时（即Order Counter的输出时），表示该Slot中的Flit可以下环时，则该Flit下环，同时T0\_Etag\_Order\_FIFO将会把该Slot的ID移除。实际的Flit信号将有datapath模块发送给下游，E-Tag模块仅需要将Valid信号在此时发出即可。另外，由于不同等级Slot中的Flit在自己相应的优先级相对应的FIFO Entry满时可以使用其下级优先级的FIFO Entry，例如T0级Slot中的Flit可以在T0级专用的FIFO Entry满时使用T1级可用的FIFO Entry，故E-Tag除了Valid信号还需要向下游FIFO发送实际使用的Entry等级（PlaceLevel）。PlaceLevel等于00,10,11时分别表示使用的FIFO Entry是T0+T1+T2公用的，T1+T0共用的以及T0专用的。

E-Tag中的UE\_Counters用来记录了每一等级的FIFO Usable Entry的数量。其加减由ETag\_Ctrl模块发送的UE\_DEC以及UE\_INC控制。UE\_DEC的信号根据所实际使用的Entry等级决定。而UE\_INC的来源是ETag\_Ctrl模块分析来自下游FIFO发来的Release信号。当T0\_Release为1时，只有T0\_UE\_INC置1。当T1\_Release为1时，T0\_UE\_INC与T1\_UE\_INC置1。当T2\_Release为1时，T0\_UE\_INC，T1\_UE\_INC以及T2\_UE\_INC全部置1。

5.6.3.3 Datapath Module Implementation

Datapath是根据I-Tag和E-Tag模块生成的控制信号决定上游FIFO中或者内部上游Slice中的Flit是否传入内部下游Slice，内部上游Slice中的Flit是否下环进入下游FIFO，以及内部Slice中的Tag和Status位是否需要变更。

Datapath接收到来自E-Tag模块的信号为Eject, Reset\_Etag。当Datapath接收到Eject信号时，下游Slice输入的valid位将被更改为0，且上游Slice的输出将被发送给下游FIFO而非下游Slice。

Datapath接收到来自I-Tag模块的信号为ITagSlotRemove, GoToRing, ITagInsert。当Datapath接收到ITagSlotRemove信号时，datapath的输出Slot\_Out中对应的ITag位将不采用下游Slice的输出而是直接将ITag位重置为0。而当接收到ITagInsert信号时，datapath的输出Slot\_Out中对应的ITag位将输出该方向I-Tag模块的ID。当GoToRing为1时，datapath的输出Slot\_Out中对应的Flit位和valid位将输出上游FIFO中的Flit和为1的valid信号而非下游Slice的输出。反之，当ITagInsert或GoToRing为0时，Slot\_Out取上游Slice相应位作为输出。并且，E-Tag位将被置0。

6 Implementation Guide

6.1 Synthesis Guide

< List the guide and special requirements when synthesizing the module, say, used macros, false path, multi-cycle path. It is better you can provide the area/instance information of this module under certain configuration.>

XXX

表 6-1: Synthesis Result

Process	Area/Instance	Key Configuration
Tsmc5		

## 6.2 PD Guide

给 *PD floorplan* 的 *Guide*，包括主要数据流走向，*Harden Block* 的接口连接等。



## 7 Software Program Guide

软件根据需要向 ETag\_Config, ITag\_Th\_H0, ITag\_Th\_H1, ITag\_Th\_V0, ITag\_Th\_V1, ITag\_Max\_Num\_H0, ITag\_Max\_Num\_H1, ITag\_Max\_Num\_V0, ITag\_Max\_Num\_V1 写入配置。设置I-Tag, E-Tag所需的可配参数。详情参考7.1。

### 7.1 Register description

表 7-1 展示了软件可配置的寄存器，表 7-2，7-3，7-4，7-5 分别展示了 I-Tag 相关的可配参数。

表 7-1 软件可配寄存器总汇

Offset	Name	Description	Default	Attr	Width
0x04	ETag_Config	配置 E-Tag 相关可变参数	0x0F	RW	64bit
0x08	ITag_Th_H0	配置横向环 I-Tag 触发条件的可变参数	0x00	RW	64bit
0x0C	ITag_Th_H1	配置横向环 I-Tag 触发条件的可变参数	0x00	RW	64bit
0x10	ITag_Th_V0	配置纵向环 I-Tag 触发条件的可变参数	0x00	RW	64bit
0x14	ITag_Th_V1	配置纵向环 I-Tag 触发条件的可变参数	0x00	RW	64bit
0x18	ITag_Max_Num_H0	配置横向环 I-Tag 最大预留可变参数	0x00	RW	64bit
0x1C	ITag_Max_Num_H1	配置横向环 I-Tag 最大预留可变参数	0x00	RW	64bit
0x20	ITag_Max_Num_V0	配置纵向环 I-Tag 最大预留可变参数	0x00	RW	64bit
0x24	ITag_Max_Num_V1	配置纵向环 I-Tag 最大预留可变参数	0x00	RW	64bit

#### 7.1.1 ETag\_Config, ETag Related Configurable Register (0x04)

表 7-2 ETag 可配寄存器

51:48	47: 44	43:40	39:32
RW	RW	RW	RW
TL_Etag_T1_UE_MAX	TL_Etag_T2_UE_MAX	TR_Etag_T1_UE_MAX,	TL_Etag_Opposite_Slot_Offset

27:24	23: 20	19:16	15:0
RW	RW	RW	RW
TU_Etag_T1_UE_MAX	TU_Etag_T2_UE_MAX	TD_Etag_T1_UE_MAX,	TU_Etag_Opposite_Slot_Offset

Default value: 0x0131\_0100\_0131\_0100

TL\_Etag\_T1\_UE\_MAX 代表横向环向左 T1 可使用下环 FIFO Entry 数的最大值，  
TL\_Etag\_T2\_UE\_MAX 代表横向环向左 T2 可使用下环 FIFO Entry 数的最大值，

TR\_Etag\_T1\_UE\_MAX 代表横向环向右 T2 可使用下环 FIFO Entry 数的最大值，  
TL\_Etag\_Opposite\_Slot\_Offset 代表横向环左与横向环向右 id 的 offset。  
TU\_Etag\_T1\_UE\_MAX 代表纵向环向上 T1 可使用下环 FIFO Entry 数的最大值，  
TU\_Etag\_T2\_UE\_MAX 代表纵向环向上 T2 可使用下环 FIFO Entry 数的最大值，  
TD\_Etag\_T1\_UE\_MAX 代表纵向环向下 T2 可使用下环 FIFO Entry 数的最大值，  
TU\_Etag\_Opposite\_Slot\_Offset 代表纵向环向上与纵向环向下 id 的 offset。

7.1.2 ITag\_Th\_H0, ITag (Horizontal Ring) Trigger Threshold Configurable

Register (0x08)

表 7-3 横向环 ITag 触发临界值可配寄存器（组 0）

63:62	61:60	...	7:6	5:4	3:2	1:0
RW	RW	...	RW	RW	RW	RW
ITag_Trigger_Th_H37	ITag_Trigger_Th_H36	...	ITag_Trigger_Th_H03	ITag_Trigger_Th_H02	ITag_Trigger_Th_H01	ITag_Trigger_Th_H00
Default value: 0x00						

横向环X=0,1,2,3的节点的ITag触发值配置，00，01，10，11分别代表绕0.25圈，0.5圈 1圈，2圈所需要的时钟周期。该配置仅会在相应节点启用时有效。

7.1.3 ITag\_Th\_H1, ITag (Horizontal Ring) Trigger Threshold Configurable

Register (0x0C)

表 7-4 横向环 ITag 触发临界值可配寄存器（组 1）

63:62	61:60	...	7:6	5:4	3:2	1:0
RW	RW	...	RW	RW	RW	RW
ITag_Trigger_Th_H77	ITag_Trigger_Th_H76	...	ITag_Trigger_Th_H43	ITag_Trigger_Th_H42	ITag_Trigger_Th_H41	ITag_Trigger_Th_H40
Default value: 0x00						

横向环X=4, 5, 6, 7的节点的ITag触发值配置，00，01，10，11分别代表绕0.25圈，0.5圈 1圈，2圈所需要的时钟周期。该配置仅会在相应节点启用时有效。

7.1.4 ITag\_Th\_V0, ITag (Vertical Ring) Trigger Threshold Configurable

Register (0x10)

表 7-5 纵向环 ITag 触发临界值可配寄存器（组 0）

63:62	61:60	...	7:6	5:4	3:2	1:0
RW	RW	...	RW	RW	RW	RW
ITag_Trig	ITag_Trig	...	ITag_Trigge	ITag_Trigge	ITag_Trigge	ITag_Trigge

ger_Th_V 37	ger_Th_V 36		r_Th_V33	r_Th_V32	r_Th_V31	r_Th_V30
Default value: 0x00						

纵向环X=0,1,2,3的节点的ITag触发值配置，00，01，10，11分别代表绕0.25圈，0.5圈 1圈，2圈所需要的时钟周期。该配置仅会在相应节点启用时有效。

### 7.1.5 ITag\_Th\_V1, ITag (Vertical Ring) Trigger Threshold Configurable Register (0x14)

表 7-6 纵向环 ITag 触发临界值可配寄存器（组 1）

63:62	61:60	...	7:6	5:4	3:2	1:0
RW	RW	...	RW	RW	RW	RW
ITag_Trigger_Th_V 77	ITag_Trigger_Th_V 76	...	ITag_Trigger_Th_V43	ITag_Trigger_Th_V42	ITag_Trigger_Th_V41	ITag_Trigger_Th_V40
Default value: 0x00						

纵向环X=4, 5, 6, 7的节点的ITag触发值配置，00，01，10，11分别代表绕0.25圈，0.5圈 1圈，2圈所需要的时钟周期。该配置仅会在相应节点启用时有效。

### 7.1.6 ITag\_Max\_Num\_H0, ITag (Horizontal Ring) Maximum Number Configurable Register (0x18)

表 7-7 横向环 ITag 最大可预留数可配寄存器（组 0）

63:62	61:60	...	7:6	5:4	3:2	1:0
RW	RW	...	RW	RW	RW	RW
ITag_Max_Num_H3 7	ITag_Max_Num_H3 6	...	ITag_Max_Num_H03	ITag_Max_Num_H02	ITag_Max_Num_H01	ITag_Max_Num_H00
Default value: 0x00						

横向环X=0,1,2,3的节点的ITag最大可预留数配置，00，01，10，11分别代表1至4个。该配置仅会在相应节点启用时有效。

### 7.1.7 ITag\_Max\_Num\_H1, ITag (Horizontal Ring) Maximum Number Configurable Register (0x1C)

表 7-8 横向环 ITag 最大可预留数可配寄存器（组 1）

63:62	61:60	...	7:6	5:4	3:2	1:0
RW	RW	...	RW	RW	RW	RW

ITag_Max_Num_H7	ITag_Max_Num_H7	...	ITag_Max_Num_H43	ITag_Max_Num_H42	ITag_Max_Num_H41	ITag_Max_Num_H40
7	6					
Default value: 0x00						

横向环X=4,5,6,7的节点的ITag最大可预留数配置，00，01，10，11分别代表1至4个。该配置仅会在相应节点启用时有效。

**7.1.8 ITag\_Max\_Num\_V0, ITag (Vertical Ring) Maximum Number Configurable Register (0x20)**

表 7-9 纵向环 ITag 最大可预留数可配寄存器（组 0）

63:62	61:60	...	7:6	5:4	3:2	1:0
RW	RW	...	RW	RW	RW	RW
ITag_Max_Num_V3	ITag_Max_Num_V3	...	ITag_Max_Num_V33	ITag_Max_Num_V32	ITag_Max_Num_V31	ITag_Max_Num_V30
7	6					
Default value: 0x00						

纵向环X=0,1,2,3的节点的ITag最大可预留数配置，00，01，10，11分别代表1至4个。该配置仅会在相应节点启用时有效。

**7.1.9 ITag\_Max\_Num\_V1, ITag (Vertical Ring) Maximum Number Configurable Register (0x24)**

表 7-10 纵向环 ITag 最大可预留数可配寄存器（组 1）

63:62	61:60	...	7:6	5:4	3:2	1:0
RW	RW	...	RW	RW	RW	RW
ITag_Max_Num_V7	ITag_Max_Num_V7	...	ITag_Max_Num_V43	ITag_Max_Num_V42	ITag_Max_Num_V41	ITag_Max_Num_V40
7	6					
Default value: 0x00						

纵向环X=4，5，6，7的节点的ITag最大可预留数配置，00，01，10，11分别代表1至4个。该配置仅会在相应节点启用时有效。

## 8 Power

介绍 IP 的功耗信息