# MSI中断请求（已实现）

当MSI (Message signaling Interrupts)功能使能时，每个Switch Downstream端口可以检测并产生以下中断(按优先级高的顺序排列):

* PME中断
* 热插拔事件中断
* AER中断
* 链路均衡请求事件
* 链路带宽管理事件
* 链路自治带宽事件
* DPC中断

如果同时发生多个事件，则从优先级高的开始依次产生事件中断。更复杂的优先级方案(如轮询)没有实现，因为这样的中断很少发生，并且没有中断被更高优先级的中断阻塞较长时间的实际情况。

XpressSWITCH提供了三种可能的MSI向量:一种用于DPC(由K\_PEXCONF[292:288]定义)，一种用于AER(由K\_PEXCONF[199:195]定义)，最后一种用于剩余的下游端口事件，包括热插拔，链路均衡，链路带宽和链路自治带宽事件(由K\_PEXCONF[204:200]定义)。如果DPC中断使用另一个向量，它将服从上述中断生成的优先级规则。

注意:switch上行端口不需要生成任何MSI请求。

XpressSWITCH支持的MSI capability结构：

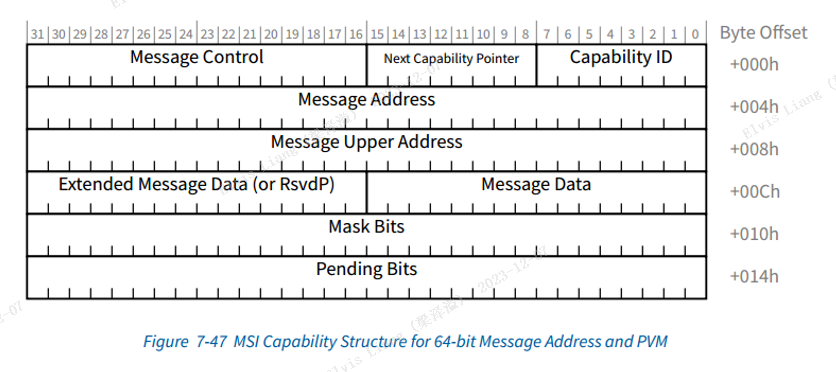


图 ‑1

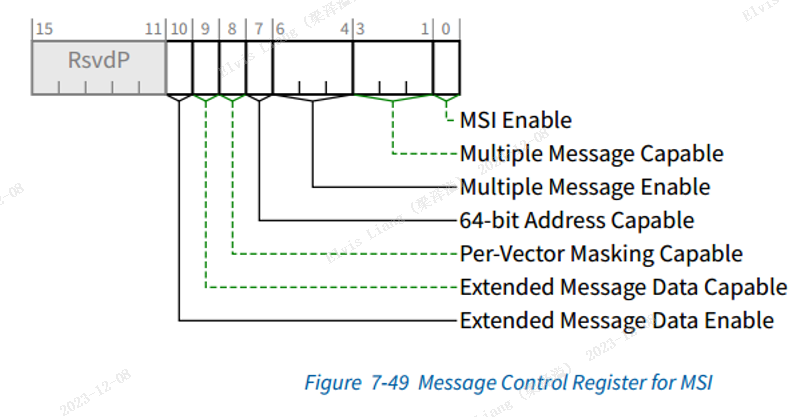


图 ‑2

# INTX中断请求（已实现）

在不能使用消息信号中断(MSI)或MSI- x机制的情况下，INTx虚拟线中断信令机制用于支持legacy EP和PCIE桥。交换逻辑根据其下游端口的设备编号执行中断重映射。结果通过相应的TLP message报文向upstream转发。使用隐式路由，路由规则是terminate at Receiver。

# MSIX中断请求(未在UG中找到描述)

未在UG中找到支持MSIX功能的说明。

pcie5\_intreg模块实现时，将k\_misx\_impl(k\_pciconf[`KPCICONF\_MSIX\_IMPL])配置为MSIX使能时，内部的MSIX capability寄存器导出到pcie5\_top\_pip顶层进行输出。

# pcie5\_intreg模块实现

## 接口信号

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 端口分组 | 名称 | 位宽 | 方向 | 功能 |
| clock reset | clk | 1 | I | 主时钟 |
| rstn | 1 | I | 异步复位，低有效 |
| crstn | 1 | I | 异步复位，低有效 |
| crst | 1 | I | 同步复位，高有效 |
| srst | 1 | I | 同步复位，高有效 |
| flrst | 1 | I | function level reset |
| internal hardware configure | test\_in | PCIE\_TESTIN\_SIZE | I | 测试信号 |
| k\_gen | PCIE\_KGEN\_SIZE | I | 外部中断 |
| k\_pciconf | PCIE\_KPCICONF\_SIZE | I | PCIE配置 |
| k\_pexconf | PCIE\_KPEXCONF\_SIZE | I | PCIE downstream 中断向量编号 |
| k\_devtype | 4 | I | 设备类型 |
| system software configure(from R2C port) | cfg\_addr | 12 | I | 配置capability地址 |
| cfg\_wdata | 32 | I | 配置capability数据 |
| cfg\_wbv | 4 | I | 配置capability数据strb |
| capability registers output | reg\_3c | 32 | O |  |
| msi\_cap | 6\*32 | O | MSI capability |
| msix\_00 | 32 | O | MSIX capability offset 0 |
| msix\_04 | 32 | O | MSIX capability offset 4 |
| msix\_08 | 32 | O | MSIX capability offset 8 |
| 非downstream的自定义中断接口 | tl\_pex\_enable |  | I |  |
| tl\_report\_state |  | I |  |
| tl\_int\_status |  | I |  |
| tl\_int\_msireq |  | I |  |
| tl\_int\_msinum |  | I |  |
| tl\_int\_msiack |  | O |  |
| int status | int\_pinstate | 4 | O | pin state |
| int\_status | 1 | O | status |
| C2T MSI transmission | msi\_req | 1 | O | 与tl接口对应的MSI请求 |
| msi\_reqaddr | 64 | O | 与tl接口对应的MSI addr |
| msi\_reqdata | 16 | O | 与tl接口对应的MSI data |
| msi\_ack | 1 | I | 与tl接口对应的MSI请求应答 |
| int source | pme\_int\_status | 1 | I | PME中断 |
| aer\_int\_status | 1 | I | AER中断 |
| serr\_event\_status | 1 | I | 未触发内部进程，直接通过tl\_report\_event输出 |
| hotplug\_int\_status | 1 | I | 热插拔事件中断 |
| equaliz\_int\_status | 1 | I | 链路均衡请求事件 |
| link\_bw\_int\_status | 1 | I | 链路带宽管理事件中断 |
| auto\_bw\_int\_status | 1 | I | 链路自治带宽事件中断 |
| dpc\_int\_status | 1 | I | DPC中断 |
| tl\_report\_event | 8 | O | 事务层向外报告的事件 |
| downstream port interrupt | dnint\_pin | 4 | O | int pin state |
| dnmsi\_req | 1 | O | downstream port对应的MSI请求 |
| dnmsi\_reqaddr | 64 | O | downstream port对应的MSI addr |
| dnmsi\_reqdata | 16 | O | downstream port对应的MSI data |
| dnmsi\_ack | 1 | I | downstream port对应的MSI请求应答 |

## 状态机



图 ‑1 MSI req状态机

|  |  |  |  |
| --- | --- | --- | --- |
| 状态说明 | 状态名 | 下一状态 | 转换条件 |
| 空状态 | MSI IDLE | MSI PREP | 当tl接口tl\_int\_msireq为1 |
| MSI准备 | MSI PREP | MSI REQ | 立即 |
| 请求发送MSI报文 | MSI REQ | MSI IDLE | 收到请求发送MSI ack |



图 ‑2 downstream MSI req状态机

|  |  |  |  |
| --- | --- | --- | --- |
| 状态说明 | 状态名 | 下一状态 | 转换条件 |
| 空状态 | DNMSI IDLE | DNMSI PME | msi enable为1，同时PME中断为1 |
| DNMSI HPG | msi enable为1，同时热插拔事件中断为1 |
| DNMSI EQU | msi enable为1，同时链路均衡请求事件中断为1 |
| DNMSI AER | msi enable为1，同时AER中断为1 |
| DNMSI LBW | msi enable为1，同时链路带宽管理事件中断为1 |
| DNMSI ABW | msi enable为1，同时链路自治带宽事件中断为1 |
| DNMSI DPC | msi enable为1，同时DPC中断为1 |
| 请求发送有关PME中断的MSI报文 | DNMSI PME | DNMSI IDLE | 收到请求发送MSI ack |
| 请求发送有关热插拔中断的MSI报文 | DNMSI HPG | DNMSI IDLE | 收到请求发送MSI ack |
| 请求发送有关链路均衡请求事件的MSI报文 | DNMSI EQU | DNMSI IDLE | 收到请求发送MSI ack |
| 请求发送有关AER中断的MSI报文 | DNMSI AER | DNMSI IDLE | 收到请求发送MSI ack |
| 请求发送有关链路带宽管理事件中断的MSI报文 | DNMSI LBW | DNMSI IDLE | 收到请求发送MSI ack |
| 请求发送有关链路自治带宽事件中断的MSI报文 | DNMSI ABW | DNMSI IDLE | 收到请求发送MSI ack |
| 请求发送有关DPC中断的MSI报文 | DNMSI DPC | DNMSI IDLE | 收到请求发送MSI ack |

## 模块实现



图 ‑3 MSI系统框架图

1. downstream port相关的MSI中断发送的message data的与中断向量相关的域段只有三个取值，为AER的中断向量号，DPC的中断向量号，其余都使用PEX的中断向量号。
2. k\_gen, k\_pciconf, k\_pexconf, k\_devtype为与硬件设计相关的配置端口，例如msi是否实现，multiple message capable这种对系统软件为RO的字段。
3. cfg\_addr, cfg\_wdata, cfg\_wbv为系统软件配置的接口，如msi enable, multiple message enable, message addr, message data等，这些配置由系统软件下发，通知此节点switch的执行方式。
4. msi\_cap, msix\_00, msix\_04, msix\_08，为内部的capability寄存器输出。
5. tl\_int\_x信号，一直拉到了pcie5\_top\_pipe顶层，由HSIO顶层例化时将req信号tie0，目前推测这组信号是给user logic预留的发送内部自定义的中断的接口，如图4-1 MSI req状态机定义了这组接口是如何向upstream口的tx发送request信号的，request发送通过接口msi\_grp组信号完成。
6. int\_pinstate, int\_status，中断pin state和中断状态输出，此组接口与INTx相关。当MSI使能和MSIX使能都关的时候，内部如果有中断产生int\_status为1，根据k\_intpin（INTX的配置使能）的配置，将相对应的pinstate拉高。在mfconfctrl收集所有function的pinstate后，由function0代表其他function向外发送INTx。
7. x\_int\_status定义的这组信号为外部向此模块输入的中断状态，用于产生downstream MSI中断请求，如图4-2 downstream MSI req状态机定义了通过dnmsi\_grp这组接口向downstream 的C2R发送request的流程。C2R随后向downstream 的decoder发送C2R组包好的数据。

# cfgrw模块TLP组包实现

1. C2T端口组包后的MSI报文，也就是memory write报文的组成：

包头定义有如下两种

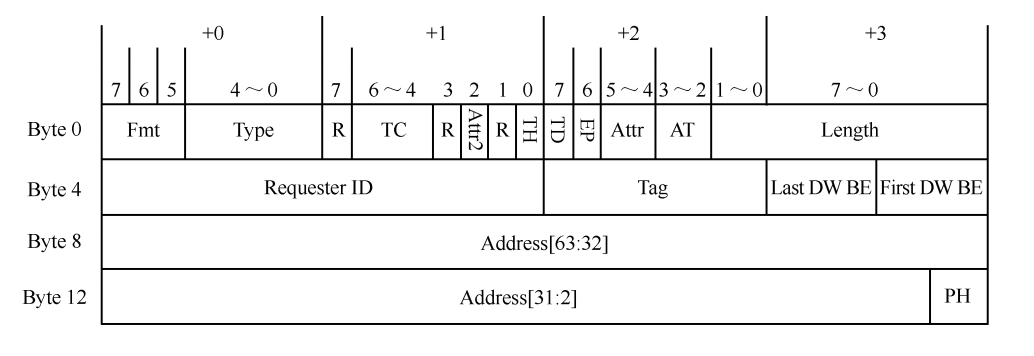


图 ‑1

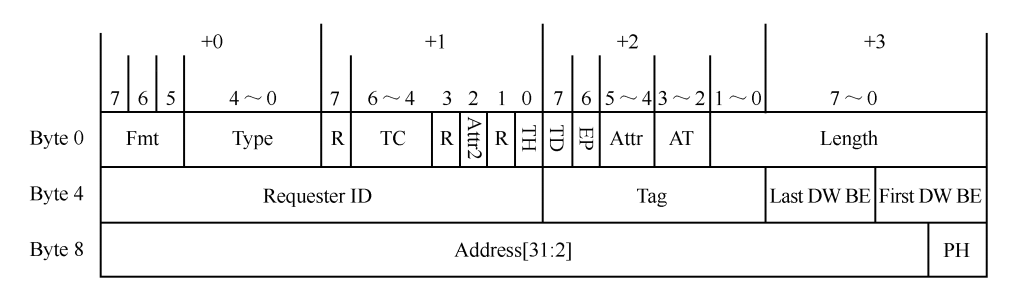


图 ‑2

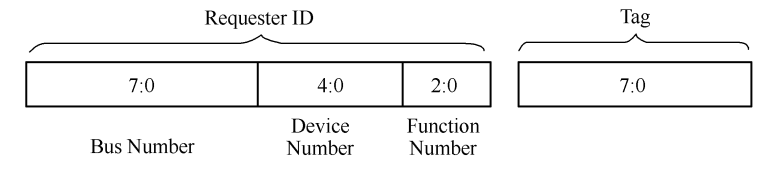


图 ‑3

fmt: [7:6]: 2’b01, [5]: 是否为64位msi addr(0->包头3个双字，即32bit地址，图4-5的包格式，1->包头4个双字，即64bit地址，图4-4)

TYPE: 0

TC: 0

Attr2: 0

TH: 0

TD: 0

EP: 0

Attr: 0

AT: 0

Length: 10’h01

requester ID:

* bus number:

rootport, switch down, BFM, bridge forward -> captured by application

endpoint, switch up&others -> captured from CfgWr0



图 ‑7 CfgWr0 fmt type定义

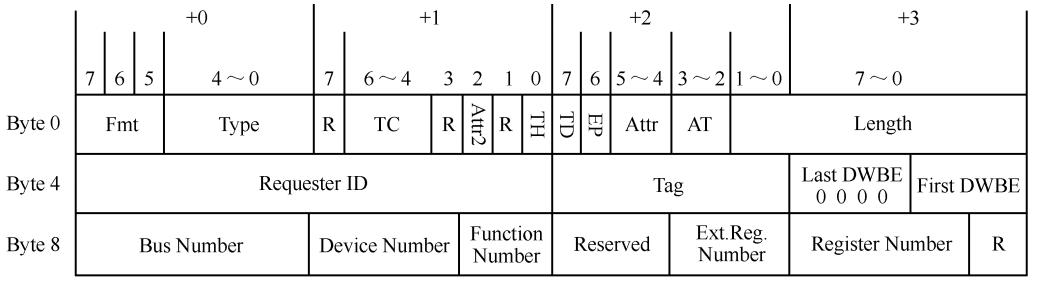


图 ‑8 CfgWr0报文头格式

* device number：

rootport, switch down, BFM -> k\_pexconf

bridge forward, endpoint, switch up&others -> 0

* function number：由模块自己定义。

TAG: 0

Last DW BE: 0

First DW BE: 4’hf

地址根据目前为64bit还是32bit，进行相应包头位置的填充。

data payload:

数据填充为大小端调整后的数据。

1. C2R端口组包后的MSI报文，也就是memory write报文的组成：

包头定义有如下两种

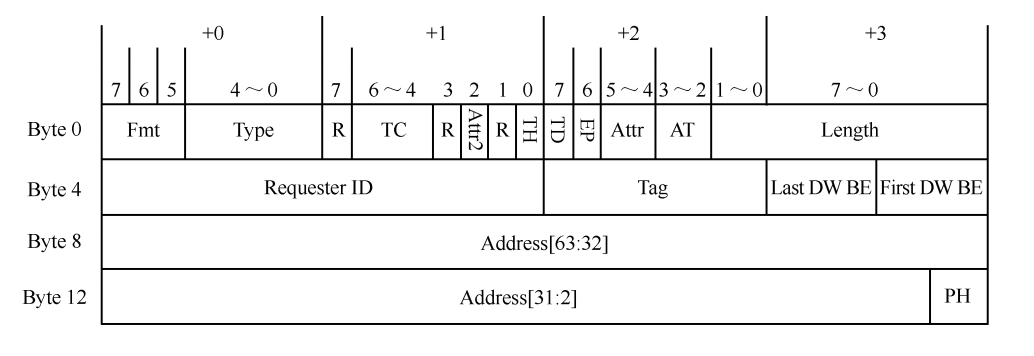


图 ‑4

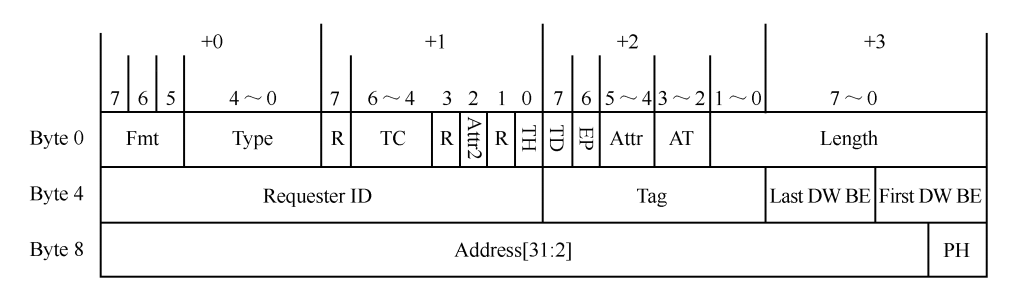


图 ‑5

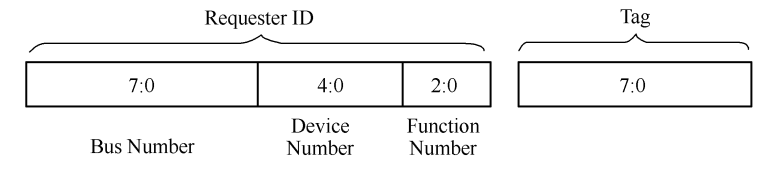


图 ‑6

fmt: 64位地址 -> 011，32位地址 -> 010

TYPE: 0

TC: 0

Attr2: 0

TH: 0

TD; tl\_pex\_enable[`PEXEN\_ECRCGEN\_EN]

EP: 0

Attr: 0

AT: 0

Length: 10’h01

requester ID:

* bus number: 与1中bus number相同
* device number：与1中device number相同
* function number：0

TAG: 0

Last DW BE: 0

First DW BE: 4’hf

地址根据目前为64bit还是32bit，进行相应包头位置的填充。

data payload:

数据填充为大小端调整后的数据。

1. C2T端口组包后的INTX报文，也就是Message报文的组成：

包头定义如下：

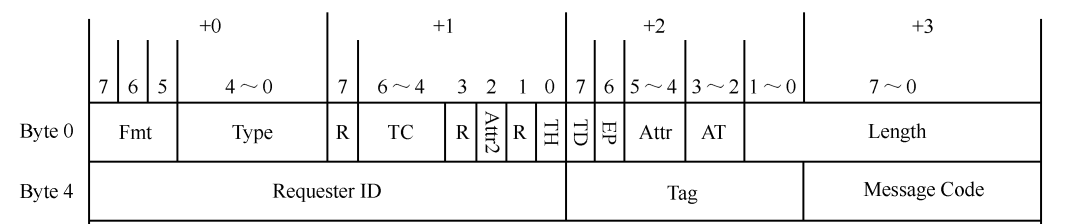


图 ‑9

Fmt: 3’b001

Type: 4’b10100

TC: 0

Attr2: 0

TH: 0

TD: 0

EP: 0

Attr: 0

AT: 0

Length: 0

requester ID: 与2中requester ID相同

TAG: 0

Message code: [7:3]: INTX宏定义；

[2]: Assert/Deassert；

[1:0]: 中断管脚编号，tx\_int\_typ