OpenTSN 交换节点硬件 UM 设计文档

（V1.0）

一、设计目标

1.1 OpenTSN 开源项目简介

OpenTSN 是由 FAST 开源项目发起，将用于对 TSN 中的关键技术和算法进行原型系统验证。由于 TSN 在时间同步、输入输出调度、状态与资源管理等方面与传统以太网完全不同，目前还没有一种通用的模型用于 TSN 交换原型系统的快速实现。针对这一问题，FAST 团队提出了 OpenTSN 开源项目，实现了 TSN 对网络的相关要求。

1.2 OpenTSN 交换的主要功能

OpenTSN 支持的功能包括：

1. 基于 CQF 的输出调度机制；
2. 基于令牌桶的 P3-P5 优先级的流量预留调度；
3. 支持本地 TSN 节点对本地状态通过 Beacon 消息进行周期性上报；
4. 对进入流水线中的所有报文在 TAP 口进行镜像用于分析。

二、概要设计

2.1 总体架构

OpenTSN 原型系统的硬件总体设计采用 FAST 架构，但简化了控制通路的设计。OpenTSN 架构如图 1 所示，白色模块为 OpenTSN 需增加的关键功能模块，需要重新开发；橙色模块为对 OpenTSN 功能进行支持需要部分修改的模块。

OpenTSN 架构如图 1 所示，白色模块为 OpenTSN 需增加的关键功能模块，需要重新开发；橙色模块为对 OpenTSN 功能进行支持需要部分修改的模块。

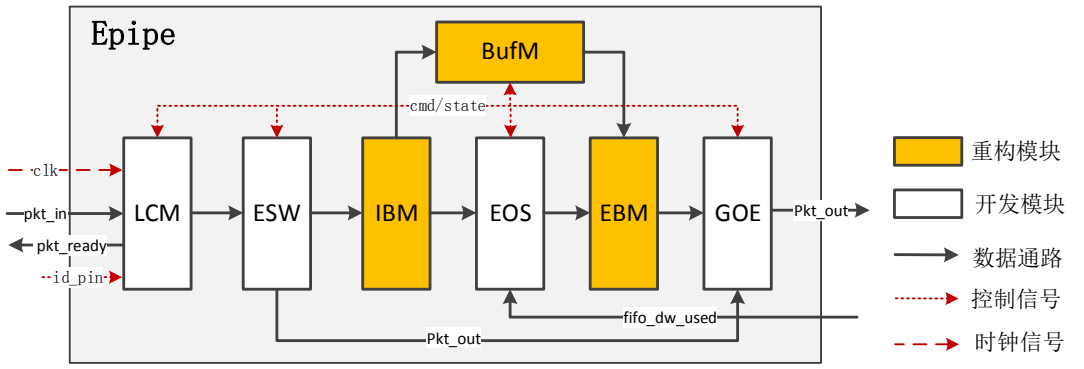


图 1 OpenTSN 原型系统硬件总体设计图

其中关键模块包含以下四个部分：

LCM 模块：是 TSN 交换节点本地的管理与控制模块。主要负责维护各模块控制相关寄存器的维护与请求、周期性设备状态信息上报、处理来自 CNC 节点的寄存器修改请求。

ESW 模块：是 TSN 交换节点的交换模块。主要负责分组类型解析、流量监管、转发动作生成以及 TSN-metadata（详见附录二）生成的操作。该模块将进入流水线的分组区分为四类：控制相关分组、TSN 分组、预约带宽分组以及 best effort 分组。在流水线中 BufM 资源紧张可以引起丢包时，根据上述优先级丢弃低优先级的报文保证 TSN

分组的可靠传输。另外，ESW 还需要根据 CNC 的配置确定所有分组的目的端口。

EOS 模块：是 TSN 节点的出队列调度模块，用于实现简化的 cqf 转发。主要负责支持核心的 TSN 元数据的循环队列转发功能，确保 TSN 分组调度输出的确定性延时，以及带宽预约分组的基于令牌桶的流量整形功能的实现。

GOE 模块:主要负责对接收到的来自 EBM 的分组按照 metadata 中 action 字段定义与来自 ESW 的分组进行发送，并对发往所有发送报文进行统计计数。

2.1.1 UM 模块信号定义

FPGA OS 与 UM 的连接信号图如图 1-1 所示。

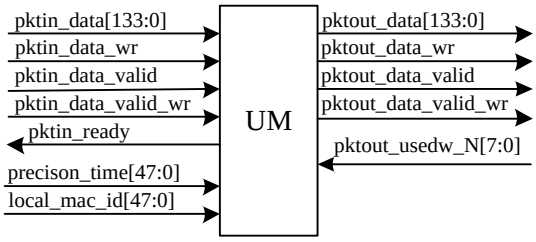


图 1-1 UM 外围接口信号定义图

根据 OpenTSN 整体架构，UM 的接口定义如下：

信号名	方向	位宽	描述
FPGA OS Ingress to UM 信号定义			
pktin_data_wr	Input	1	报文数据写信号
pktin_data	Input	134	报文数据
pktin_data_valid	Input	1	报文数据标志位
pktin_data_valid_wr	Input	1	报文数据标志位写信号
pktin_ready	output	1	数据 ready 信号
precision_time	Input	48	精确同步时间值，[47:17]为毫秒（ms）计数器，[16:0]为周期数器即每周期 8ns。
Local_mac_id	Input	48	生成的本地 MAC 地址
UM to FPGA OS Egress 模块			
pktout_data_wr_N	output	1	输出报文写信号，N 为 0-3
pktout_data_N	output	134	输出报文数据，N 为 0-3
pktout_data_valid_N	output	1	输出报文标志位，N 为 0-3
pktout_data_valid_wr_N	output	1	输出报文标志位写信号，N 为 0-3
pktout_usedw_N[7:0]	input	8	输出报文 fifo usedw 信号，N 为 0-1

2.1.2 数据分组结构定义

输入及输出数据分组包括 Metadata 头部及有效数据分组两部分，格式如图 1-2 所示，Metadata 在 FAST 报文的前 32 字节携带，每个分组进出 UM 的第 1 拍 16 字节为 Metadata0，第二拍数据为 Metadata1。

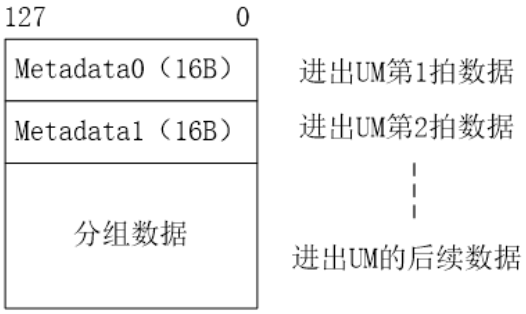


图 1-2 分组数据传输格式

接口分组(packet)是应用在 FPGA OS 与 UM 接口上的 134bit 的数据格式，其中高 6 位为控制信息，低 128 位为报文数据。分组的前两拍为 FPGA OS 添加的 32 字节的 metadata，两拍后的数据为有效分组数据。134 位的数据由 2 位的头尾标识，4 位无效字节数，128 位的有效数据组成。

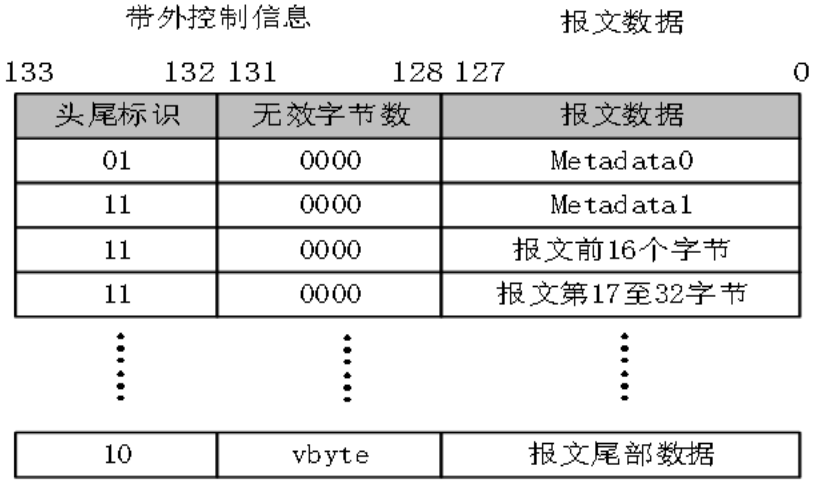


图 1-3 报文分组传输格式

其中，[133:132]位为报文数据的头尾标识，01 代表报文头部，11 代表报文中间数据，10 代表报文尾部；[131:128]位为 4 位的无效字节数，其中 0000 表示 16 个字节全部有效，0001 表示最低一个字节无效，最高 15 个字节有效，依次类推，1111 表示最低 15 个字节无效，最高一个字节有效。格式如图 1-3 所示。

2.2 FPGA_OS

FAST 平台的 FPGA 中，除了平台无关的 UM 外，还有平台相关的逻辑，称为 FPGA_OS。在 OpenTSN 开源项目中保持了 FPGA_OS 的屏蔽平台的易购性以及为 UM 的实现提供共性服务这两个功能，同时基于原 FPGA_OS 的基础上对其进行了改动。

2.2.1 增加了时间同步功能

TSN 网络中要求时间同步，实现 CNC 对整个 TSN 网络的配置以及管理。在 OpenTSN 开源项目中增加了时间同步功能，可实现设备内的时间同步以及设备间的时间同步。

2.2.2 删除了 DMA 通道

在 TSN 网络中整个 UM 中的相关寄存器的配置都是由 CNC 通过 ptp 报文进行统一

配置管理，因此去除了 DMA 通道。

2.3 LCM 模块

2.3.1 LCM 模块功能

根据上节叙述，LCM 模块负责 1) 周期性读取本地所有控制相关寄存器的值，并构造包含 beacon report 消息的以太网帧进行发送；2) 解析收到的来自 CNC 节点的 beacon update 消息的以太网帧，并根据对应字段值修改对应的寄存器。3）丢弃本地 LCM 发送的且未找到目的地址时返回到本地 LCM 的 beacon 报文。

2.3.2 LCM 模块概要设计

LCM 模块的整体架构如图 1-4 所示：

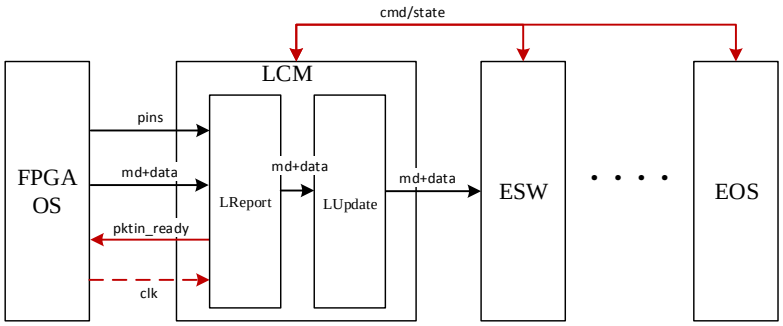
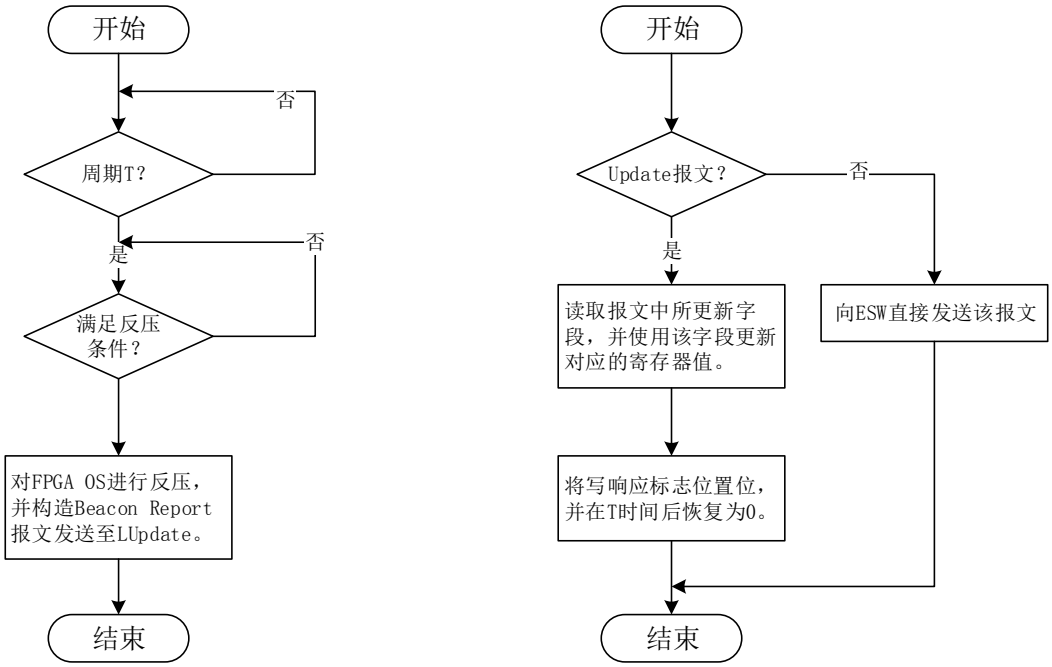


图 1-4 LCM 模块整体架构图

LCM 从 FPGA OS 接收分组，并在下一拍将分组转发到 ESW 模块。当到达发送周期时，将触发 LCM 对 FPGA OS 发送一个反压信号，并构造 Beacon Report 消息，将相关的寄存器值填写到报文体中，发送至 ESW 模块。另外，若 LCM 收到来自 FPGA OS 的消息帧（为 Beacon Update 消息），则取帧内需更新的字段的值，对对应的寄存器值进行更新（详见表 6-1）。综上，LCM 可进一步被划分为 2 个子模块分别用于周期性上报消息帧构造和 CNC 可配置寄存器值更新。两个子模块的流程图如图 1-5 所示。



(a) LReport 模块处理流程图 (b)LUpdate 模块处理流程图

图 1-5 LCM 子模块处理流程图

LReport 模块：负责接收并将来自 FPGA OS 的帧发送到 LUpdate 模块（由 LUpdate 模块发送至 ESW 模块），当时钟到达预订周期 T 时，LReport 对 FPGA OS 进行反压，并构造一个包含当前全部状态信息的 Beacon Report 消息，将其发送到 LUpdate 模块中。

LUpdate 模块：负责在收到来自 LReport 子模块的 Beacon Update 消息帧后，判断该消息帧是否为本地 LCM 发送的，若是则需要丢弃，若不是则利用帧数据字段包含的需更新寄存器的值更新对应的寄存器。若收到的帧非 Beacon Update 消息帧，则直接向 ESW 进行转发。

2.4 ESW 模块

2.4.1 ESW 模块功能

根据在总体架构中队 ESW 模块的叙述，ESW 模块的主要功能有如下：

1）分组类型解析；
2）MAC 地址比较；
3）分组转发；
4）元数据的修改以及 TSN_MD 数据的构造；
5）流量监管。

具体的功能实现如下所述：

2.4.1.1 分组类型解析

需要对分组进行解析，支持带 VLAN 头以及标准以太网的解析，两种分组使用不同的解析方法。带 VLAN 头的分组根据 VLAN 字段的优先级（PCP）值进行分组类型区分，可区分出 TSN 分组、预约带宽分组、best effort 分组。标准以太网的分组则根据协议字段区分出 PTP 分组或其他分组。

（1）判断是否带有 VLAN 头：

表 3-2-1-1：VLAN 分组的头部格式

[127:80]DMAC	[79:32]SMAC	[31:16]TPID(0x8100)	[15:13]PCP	[12]CFI	[11:0]VLAN ID

注：TPID（Tag ProtocolIdentifier）:标签协议标识。是 IEEE 定义的新的类型，表明这是一个加了 802.1 标签的帧。TPID 中包含了一个固定的值 0X8100。

PCP（user priority）：这三位指明帧的优先级，一共 8 种优先级，从 0 到 7。

CFI（canonical fomat indicator）：CFI 值为 0 说明是规范格式，1 为非规范格式。它被用在令牌环/源路由 FDDI 介质访问方法中来指示封装帧中所带地址的比特次序信息。

VLAN ID（VLAN Identified）：这是一个 12 位的域，指明 VLAN 的 ID，取值范围为 0～4095，一共 4096 个，由于 0 和 4095 为协议保留取值，所以 VLAN ID 的取值范围为 1～4094 每个支持 802.1Q 协议的交换机发送出来的数据包都会包含这个域，以指明

自己属于哪一个 VLAN。

表 3-2-1-2：标准以太网分组的头部格式

[127:80]DMAC	[79:32]SMAC	[31:16]eth type

表 3-2-1-3：PTP 协议分组的头部格式

[127:80]DMAC	[79:32]SMAC	[31:16]eth type（0x88F7）

可根据首个 16 字节的低 32 位（带 VLAN 头的值是 0x8100）判断是否是带有 VLAN 头的分组。不是带有 VLAN 头的分组再判断是否是 PTP 协议分组（PTP 协议的值是 0x88f7）或是其他的标准以太网分组。

（2）判断分组类型

可根据 VLAN 头中的 PCP 优先级进行判断，PCP 的值的对应关系如下表：

表 3-2-1-4：PCP 优先级级别对应的分组类型

PCP 的值	6、7	3、4、5	0、1、2
分组类型	TSN 分组	预约带宽分组	best effort 分组

根据分组的协议类型字段信息以及 VLAN 中的 PCP 值将分组分为以下几种分组类型：TSN 分组、预约带宽分组、best effort 分组，并在 ACTION 中的分组协议类型字段记录。

2.4.1.2 MAC 地址比较

根据分组的源 MAC、目的 MAC 分别进行比较。源 MAC 与本地直连设备 MAC 进行比较，若源 MAC 等于本地直连设备 MAC 再看输入端口是否为 2（即本地直连设备），若输入端口不是 2 则将分组丢弃，具体原因参考后面的注；若源 MAC 不等于本地直连设备 MAC 或源 MAC 等于本地直连设备且输入端口是 2 则进行目的 MAC 比较。目的 MAC 本地直连设备的 MAC 地址比较。根据比较结果进行接下来的处理。

[注：考虑到分组的目的 MAC 可能不存在于整个网络，因此在网络中传输了一个轮回之后还未找到目的，这样就可能导致分组一直在网络中循环，为了解决这一问题增加一个检测机制。若分组的源 MAC 是本地直连设备，而输入端口不为 2 则表示该分组是循环一次的，需要进行丢弃。]。

（1）源 MAC 地址比较

将分组的源 MAC 与本地直连设备的 MAC 地址进行比较，并根据不同的情况对报文进行不同的处理，得出以下结果：

比较		结果
源 MAC 等于本地直连设备地址	输入端口为 2	进行目的 MAC 比较
	输入端口不为 2	丢弃
源 MAC 不等于本地直连设备地址	输入端口为 2	丢弃

	输入端口不为 2	进行目的 MAC 比较
	输入端口为 3	丢弃

（2）目的 MAC 地址比较

将分组的目的 MAC 与本地直连设备 MAC 地址进行比较。得出以下结果：

表 3-2-1-5：目的 MAC 比较结果

比较		结果
目的 MAC 等于本地直连设备地址		outtype = 2’b00 outport = 6’h2
目的 MAC 不等于本地直连设备地址 SMID > 4（该报文由本地 LCM 或本地 PTP 发送的报文）		outtype = 2’b00 outport = direction
目的 MAC 不等于本地直连设备地址 SMID < 4		outtype = 2’b00 outport = ~inport
目的 MAC 等于 FF:FF:FF:FF:FF:FF	输入端口为 2	outtype = 2’b10 outport = direction
	SMID > 4	outtype = 2’b10 outport = direction
	其他	outtype = 2’b10 outport = ~inport

（3）ACTION 的构造

根据比较结果对 ACTION 进行构造，ACTION 的格式以及含义如下：

表 3-2-1-6：ACTION 格式和含义

名称	分组传输类型 （pkt_outtype）	分组协议类型 （pkt_type）	分组输出端口/输出 ID （outport）
位宽	2	3	6
位置	[10:9]	[8:6]	[5:0]
含义	00：单播 10：广播	0：best effort 1：预约带宽 2：PTP 3：TSN	

2.4.1.3　分组转发

根据比较结果进行分组的转发。单播时：若目的 MAC 是本地直连设备的 MAC 地址则往 GOE 模块传输。若目的 MAC 未比较对上则执行本地 LCM 和本地直连设备进的分组由 direction 信号决定由哪个端口出、0 号网口进的分组由 1 号网口出、1 号网口进的分组由 0 号网口出的规则进行处理。广播时：执行本地 LCM 和本地直连设备进的分组由 direction 信号决定由哪个端口出、0 号网口进的分组由 1 号网口出、1 号网口进的分组由 0 号网口出的规则进行处理；同时还需要将该广播分组往 2 号口复制一份。同时将所有需要进行转发的分组都复制一份转发给 TAP 口。

2.4.1.4　元数据的修改以及 TSN_MD 数据的构造

分组在传输时都需要进行元数据的修改；同时若分组需要往下级模块传输还要构造 TSN_MD。

（1）元数据的修改

ACTION 表的表项中已经给出可判断分组去向的三个重要信息：分组输出类型、分组输出目的、分组输出端口。元数据可在查表结束之后根据 ACTION 表对自身的这三个重要信息进行修改，以保证在之后的模块中可正确地对分组进行传输。

（2）TSN_MD 数据的构造

在 EOS 中需要对分组的类型进行判断，以便存入对应的队列。因此 ESW 根据需求构造仅包含这些信息的 TSN_MD 数据传输给 EOS 模块，同时 IBM 模块需要 8 位的数据进行储存分组存入 bufm 的 ID 号。因此在 ESW 定义的 MD 数据的内容如下：

表 3-2-1-7：TSN_MD 数据内容

分组协议类型[23:21] （pkt_type）	分组长度[20:9] （pkt_ len）	输出端口[8] （outport）	分组存储 ID[7:0] （bufm_ID）
0：best effort 1：预约带宽 2：PTP 3：TSN	*注：按字节数计算*	0：0 号网口输出 1：1 号网口输出	

2.4.1.5　流量监管

若分组是往下级模传输的，则需要根据 bufm 的空闲 ID 数量进行判断分组是否应该丢弃。为了避免在非 TSN 分组过多的情况下造成 bufm 资源不够用而导致 TSN 分组无法正常地在流水线中进行传输，有必要在 ESW 模块对分组进行流量监管。可根据 bufm 中的空闲 ID 块对分组进行管理，当 bufm 中的空闲 ID 小于某值 1 时，可控制 ESW 模块对 best effort 分组进行丢弃；当 bufm 中的空闲 ID 大于某值 1 但小与某值 2 时，可控制 ESW 模块对预约带宽分组进行丢弃。

3.2.2 ESW 模块概要设计

3.2.2.1　模块划分

设计中将 ESW 模块设计为 3 个模块：ESW 顶层模块、PKE 模块、PAC 模块。具

体框架图以及子模块实现如下：

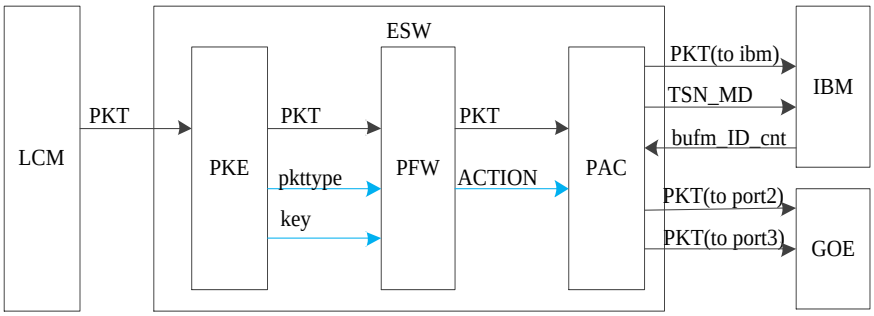


图 3-2-2-1：模块划分图

图中一些关键数据的格式定义：

表 3-2-2-1：key 格式定义

Key		
DMAC[101:54]	SMAC[53:6]	inport[5:0]

表 3-2-2-2：ACTION 格式定义

名称	分组输出类型（pkt_outtype）	分组协议类型（pkt_type）	分组输出端口/输出 ID（outport）
位宽	2	3	6
位置	[10:9]	[8:6]	[5:0]

表 3-2-2-3：TSN_MD 格式定义

分组协议类型[23:21]（pkt_type）	分组长度[20:9]（pkt_len）	输出端口[8]（outport）	分组存储 ID[7:0]（bufm_ID）

根据对 ESW 模块的功能分析，给出 ESW 模块处理流程图如下：

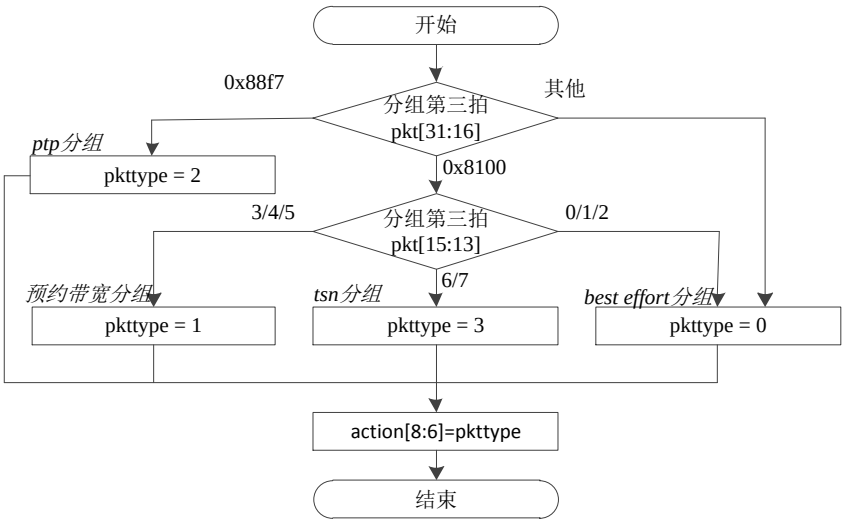


图 3-2-2-2:PKE 模块流程图

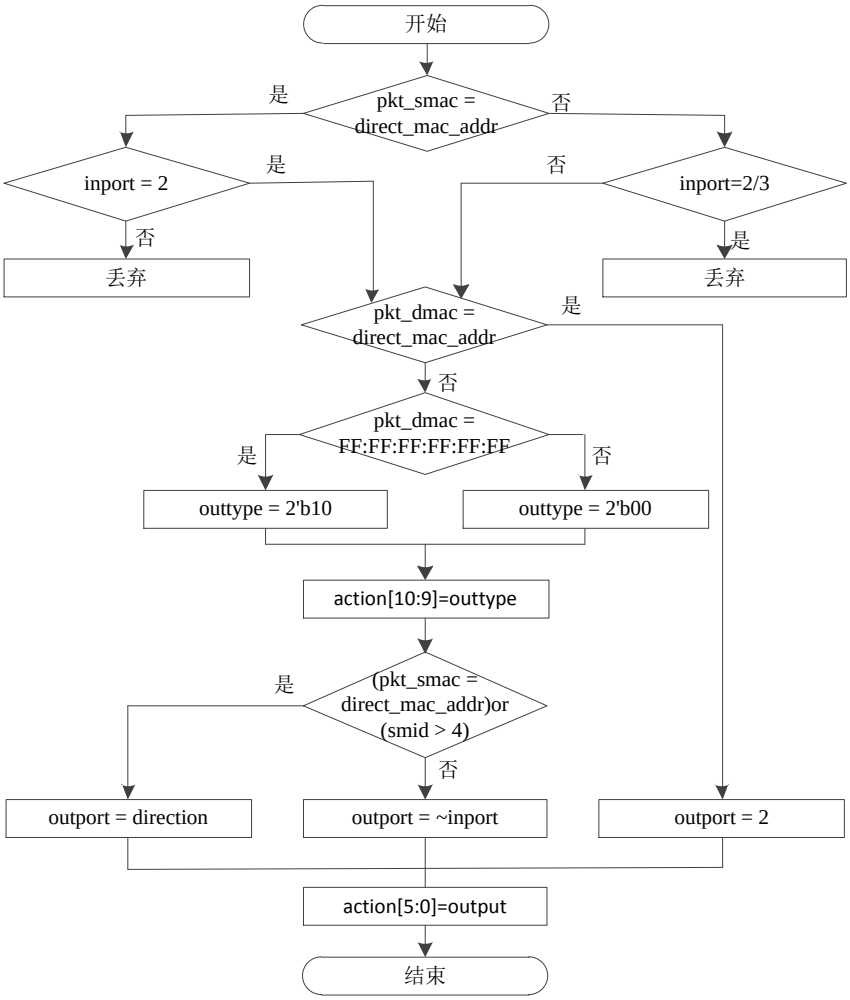


图 3-2-2-3:PFW 模块流程图

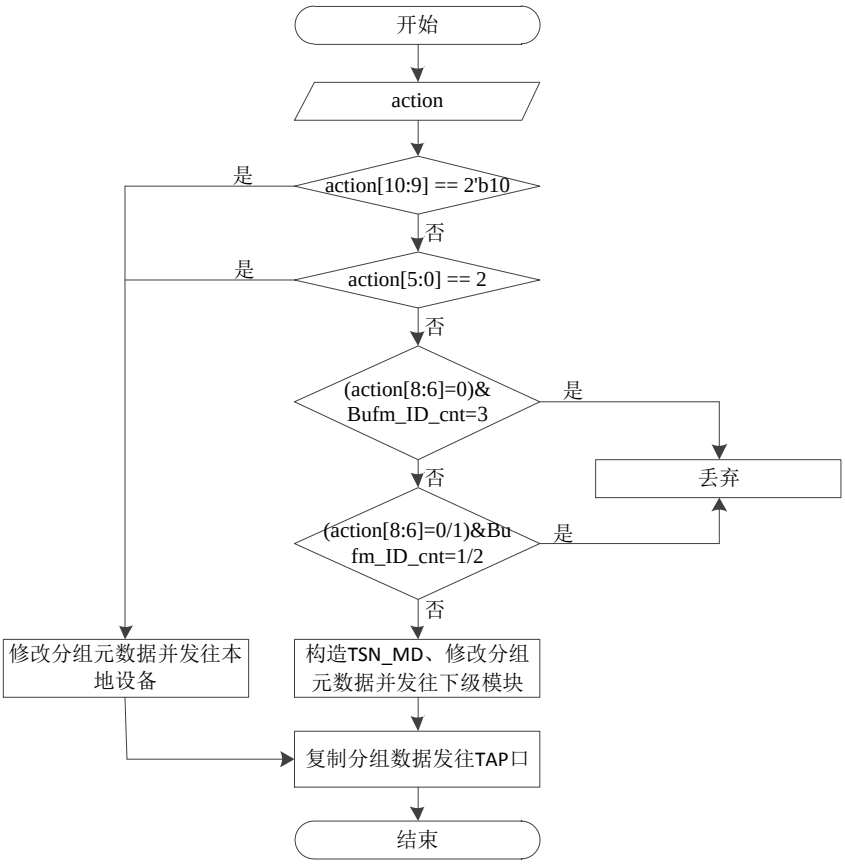


图 3-2-2-3:PAC 模块流程图

PKE 模块：判断是带 VLAN 头的分组还是标准以太网分组。若是带 VLAN 头的分组则根据 PCP 值进行分组类型的区分：TSN 分组、预约带宽分组、best effort 分组；若是标准以太网分组则根据协议字段进行分组类型区分：PTP 分组、best effort 分组。并将解析的结果写到 pkttype 中；同时提取出分组的三元组{目的 MAC、源 MAC、输入端口}与分组同时传输给 PFW 模块。

PFW 模块：根据三元组的信息与本地直连设备 MAC 进行比较：源 MAC 比较：根据 3.2.1.2 节中的表项进行不同的处理方式。目的 MAC 比较：目的 MAC 为全 F、目的 MAC 等于本地直连设备 MAC、目的 MAC 不等于本地直连设备 MAC 三种结果。根据 3.2.1.2 节中的表项进行 action 字段的修改。

PAC 模块： 根据 ACTION 中的内容进行分组的转发。单播时：若 ACTION 的输出端口为 2 则将分组元数据根据 ACTION 进行修改之后将分组往本地直连设备传输；若 ACTION 的分组输出端口不为 2 则根据 bufm 的空闲 ID 进行分组的丢弃或转发。当 bufm 大于 2，分组都可直接进行元数据修改之后传输分组；当 bufm 大于 1 小于 2 时，分组类型为 best effort 的需要进行丢弃，其他分组进行正常传输；当 bufm 小于 1 时，分组类型为预约带宽或 best effort 的都需要进行丢弃，其他分组进行正常传输。不丢弃时，在分组传输时需要构造 TSN_MD。广播时：处理与单播一样的处理方式之外还需要将分组复制一份往 2 号口进行转发。在所有需要进行转发的分组都复制一份转发给 TAP 口。

2.5 EOS 模块

2.5.1 EOS 模块功能

为了支持基于 FAST 的 TSN 交换，需要在 EOS 模块实现:

1. 分队列缓存不同类型的报文的元数据，即使用 Q0、Q1 缓存 TSN 报文的元数据，使用 Q2 缓存 PTP 和预约带宽报文的元数据，以及使用 Q3 缓存 best effort 的以太网报文的元数据；
2. 根据报文优先级调度队列，Q0 = Q1 > Q2 > Q3；
3. 实现 CQF 调度功能，即使用 Q0、Q1 实现乒乓队列；
4. 使用令牌桶对带宽预约流量（即队列 Q2 的元数据）进行整形；
5. 配置令牌桶参数，目前包含两个，即 B 表示令牌桶的桶深，R 表示往令牌桶中添加令牌的速率；
6. 供其他模块（LCM）读取 EOS 模块相关寄存器信息。

2.5.2 EOS 模块概要设计

本节主要介绍 EOS 模块的整体架构设计，EOS 模块的整体架构如图 3-2-1 所示：

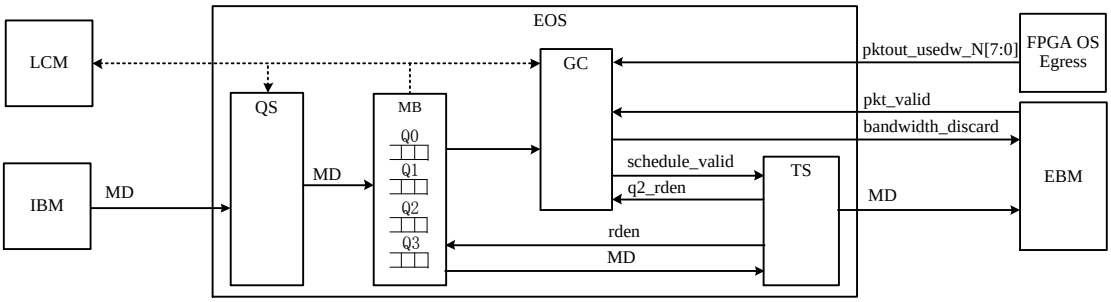
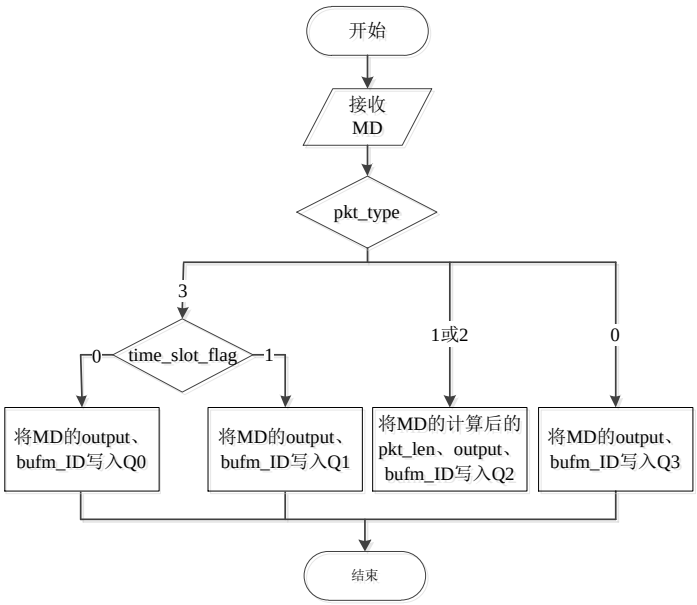
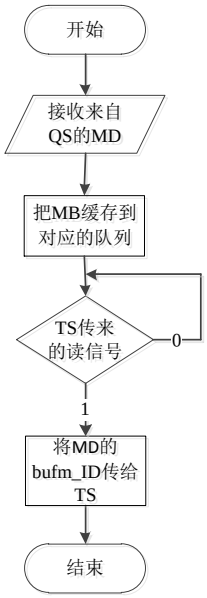


图 3-2-1 EOS 模块的整体架构

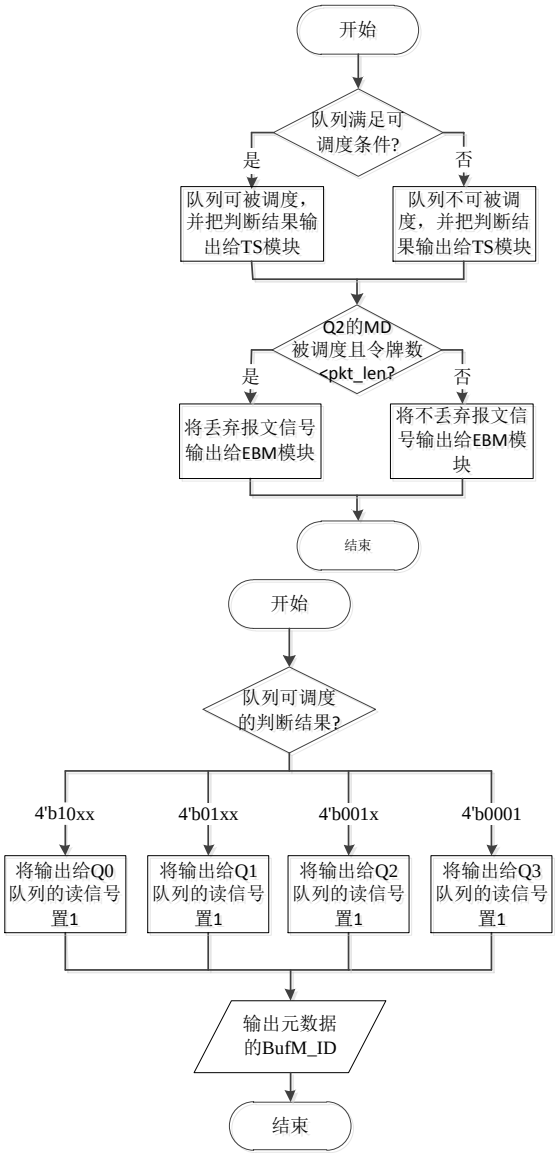
根据对 EOS 模块的功能分析，给出 EOS 模块处理流程图如下：



(a) QS 模块处理流程图



(b) MB 模块处理流程图



(c) GC 模块处理流程图　　(d) TS 模块处理流程图

图 3-2-2　EOS 子模块处理流程图

如上图 3-2-2 所示，EOS 包含 4 个处理流程，即 QS(Queue Selecting)模块负责元数据的队列选择，MB(Metadata Buffer)模块负责元数据的缓存。GC(Gate Control)模块负责判断 4 个队列是否可被调度，最后 TS(Transmitting and Scheduling)模块负责根据优先级进行调度。每个模块的具体功能如下所示：

QS 模块：队列选择模块，将元数据写入相应的队列中。根据当前时间槽的奇偶和元数据[23:21]类型决定将元数据传输到哪个队列：若元数据[23:21]为 3，当前为奇数时间槽，则将元数据[8:0]传输到 Q1；若元数据[23:21]为 3，当前为偶数时间槽，则将元数据[8:0]传输到 Q0；若元数据[23:21]为 2，则将元数据[19:9]pkt_len 置 0（PTP 分组不需要进行流量整形，不需要消耗令牌），若元数据[23:21]为 1，则将元数据[19:9]pkt_len 减去两拍 metadata 长度（32 字节），并连同元数据[8:0]传输给 Q2；若元数据 [23:21]为 0，则将元数据[8:0]传输给 Q3。

表 3-2-2 TSN_MD 数据内容

分组协议类型[23:21] （pkt_type）	分组长度[20:9] （pkt_ len）	输出端口[8] （outport）	分组存储 ID[7:0] （bufm_ID）
0：best effort 1：预约带宽 2：PTP 3：TSN	包括两拍 metadata 长度；按字节数计算。	0：0 号网口输出 1：1 号网口输出	BufM 中缓存的分组对应的 ID。

MB 模块：元数据缓存模块， Q0 为缓存 TSN 分组的元数据的偶数时钟队列，Q1 为缓存 TSN 分组的元数据的奇数时钟队列，Q2 为缓存带宽预约分组和 PTP 分组的元数据队列，Q3 为缓存 best effort 元数据的队列。等待 TS 的元数据读信号，将相应队列的元数据读出。

GC 模块：门控模块，负责令牌桶中的令牌计数；判断 MB 的 4 个队列是否可被调度，把判断结果传给 TS 模块；判断带宽预约流量是否应被丢弃，并把判断结果传给 EBM。

TS 模块：调度发送模块，根据 GC 模块传来的判断结果，对可进行调度的队列采用绝对优先级调度策略来调度 MB 的元数据。

2.6 GOE 模块

根据 GOE 模块分析，对 GOE 模块的设计如下图：

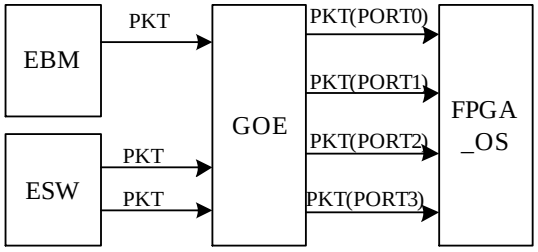


图 5-2-1：GOE 模块结构图

该模块的功能比较简单，根据进入的分组判断往哪个网口进行传输，之后便将分组传输给对应的网口。同时对所有传输的分组进行计数。

处理流程如下图：

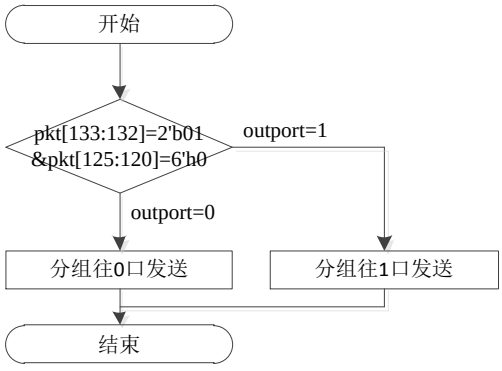


图 5-2-2：GOE 模块流程图

2.7 交换机管理控制

在 2.2 节中提到了 TSN 网络中是由 CNC 进行统一的配置管理，CNC 对各节点的管理是通过 ptp 报文，根据 ptp 报文的目的 MAC 找到对应的节点进行配置。因此各节点都需要拥有自己的一个 MAC 地址，在设计中规定了节点 MAC 的前 40 位为 00：06：06：00：00，而后 8 位可进行其他的方式进行配置，这后 8 位配置是在 FPGA_OS 中完成的。在 FPGA_OS 中可通过 CPU 的总线进行配置，也可以直接由外部芯片通过 IO 口直接给 8 位的脉冲信息进行配置，在本设计中是通过 CPU 的总线进行配置的。

CNC 知道每个节点的 MAC 地址之后，可发送 ptp 报文进行配置，节点在收到目的 MAC 为本地 MAC 的 ptp 报文之后会提取出报文中的一些特定字段信息对本地寄存器的值进行更新。

每个节点也会按周期向 CNC 上报一个 ptp 报文，该报文携带着节点的一些信息，包括 BufM 使用情况、各模块的计数器等。用户可在 CNC 通过对收到的 ptp 报文信息对各节点的运行状态进行了解，若一个节点或多个节点没有上报的报文可直接在 CNC 便可定位问题是出现在哪个交换机上。

三、详细设计

3.1 LCM 模块

3.1.1 LCM 需求与功能分析

为了简化设计，在 OpenTSN 中取消了 FAST 中的控制通路，使用 LCM 模块（本地管理模块)进行 TSN 节点全局的状态信息获取和更新。LCM 负责以下两个功能的实现：

1. LCM 模块周期性构造包含 beacon 消息的帧 (report 消息)，并根据附录一中定义将状态字段填写到 payload 字段。
2. 而当收到来自 CNC 节点的寄存器配置消息(update 消息)后，将可读写字段的值读出，并替换对应寄存器中的原始值。
3. 对进入 LCM 模块的 beacon 报文进行检测，若该 beacon 报文是本地 LCM 发出的（可根据 SMAC 地址进行判断），则将该报文进行丢弃。

3.1.2 LCM 模块详细设计

3.1.2.1 模块顶层设计

LCM 模块的外围信号定义图如图 1-6 所示。

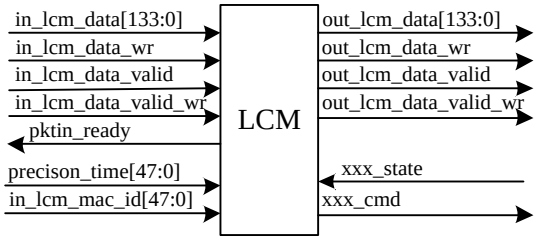


图 1-6 LCM 模块外围信号定义图

根据 UM 的外围接口定义和 OpenTSN 的模块连接关系，LCM 模块的接口定义如下，其中：

信号名	方向	位宽	描述
UM-LCM 信号定义			
in_lcm_data_wr	input	1	报文数据写信号
in_lcm _data	input	134	报文数据
in_lcm _data_valid	input	1	报文数据标志位
in_lcm _data_valid_wr	input	1	报文数据标志位写信号
pktin_ready	output	1	数据 ready 信号
precision_time	input	48	精确同步时间值，[47:17]为毫秒（ms）计数器，[16:0]为周期数器即每周期 8ns。
in_lcm_mac_id	input	48	根据引脚电平生成的 8 位 ID，用于生成 local_mac_addr。
LCM-ESW 信号定义			
out_lcm_data_wr	output	1	输出到 ESW 的报文写信号
out_lcm _data	output	134	输出到 ESW 的报文数据
out_lcm _data_valid	output	1	输出到 ESW 的数据标志位
out_lcm _data_valid_wr	output	1	输出到 ESW 数据标志写信号
LCM to other Modules 信号定义			
xxx_cmd	output	-	输出到其它模块的命令字段，详见第六章与附录一
xxx_state	input	-	输入其它模块的计数器值，详见第六章与附录一
out_direction	output	1	输出当前网络中的 direction
out_token_bucket_para	output	32	输出时间片大小/时间片翻转周期
time_slot_flag	output	1	输出时间片信息，1 为奇数时间片；0 为偶数时间片

在顶层模块中根据全局时钟 precision_time 以及配置的时间槽大小 out_token_bucket_para 来进行时间片的翻转。由于 precision_time 是由高 31 位单位为 ms 的计数器以及低 17 位单位为拍（8ns）的计数器组成的时钟信号，因此可直接通过 precision_time 的低 17 数据控制时间片的翻转。具体的设计如下：

① precision_time[16:0]==0 时，将时间片 time_slot_flag 置为 0，这是为了让全网络节点进行统一，时间片不会出现一个节点为奇数而另一个节点为偶数的情况；
② precision_time[16:0]==out_token_bucket_para 时，将时间片 time_slot_flag 翻转一次；
③ precision_time[16:0]==(out_token_bucket_para>>1)时，将时间片 time_slot_flag 翻转一次；
④ precision_time[16:0]=={(out_token_bucket_para>>1)+(out_token_bucket_para)} 时，将时间片 time_slot_flag 翻转一次；

在这个设计当中对时间槽大小 out_token_bucket_para 是有要求的，让 precision_time

的低 17 位的最高值 125000 正好是 out_token_bucket_para 的倍数，这样可以避免在 precision_time 计满 125000 之后归零的情况下造成上一个时间片提前翻转。

3.1.2.2 LReport 子模块设计

LReport 模块接收来自 FPGA OS 的时钟信号，并在到达 beacon 消息上报周期时，向 LUpdate 发送一个包含本地状态寄存器值的 beacon report 消息。

3.1.2.2.1 接口设计

LReport 接收来自 FPGA OS 的 48 位时间信号，并对 FPGA OS 的报文输出进行反压后向 LUpdate 发送所构造的 beacon report 消息。其接口定义如图 1-7 所示。

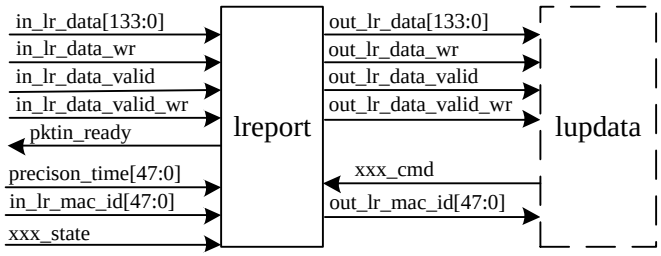


图 1-7 LReport 子模块接口信号图

LReport 子模块的具体接口信号定义如表 1-2 所示：

信号名	方向	位宽	描述
UM-LReport 信号定义			
in_lcm_data_wr	input	1	报文数据写信号
in_lcm _data	input	134	报文数据
in_lcm _data_valid	input	1	报文数据标志位
in_lcm _data_valid_wr	input	1	报文数据标志位写信号
pktin_ready	output	1	数据 ready 信号
precision_time	input	48	精确同步时间值，[47:17]为毫秒（ms）计数器，[16:0]为周期数器即每周期 8ns。
in_lr_mac_id	input	48	根据引脚电平生成的 8 位 ID，用于生成 local_mac_addr。
LReport-LUpdate 信号定义			
out_lcm_data_wr	output	1	输出到 ESW 的报文写信号
out_lcm _data	output	134	输出到 ESW 的报文数据
out_lcm _data_valid	output	1	输出到 ESW 的数据标志位
out_lcm _data_valid_wr	output	1	输出到 ESW 数据标志写信号
out_lr_mac_id	output	8	根据引脚电平生成的 8 位 ID，用于生成 local_mac_addr。
xxx_cmd	input	-	来自 lupdata 模块的命令字段，详见第六章与附录一
LCM - other Modules 信号定义			
xxx_state	input	-	来自其它功能模块的状态信息，

			详见第六章与附录一

3.1.2.2.2 模块逻辑设计

根据 LReport 子模块功能，可使用一个状态机进行控制。其状态机如图 1-8 所示。

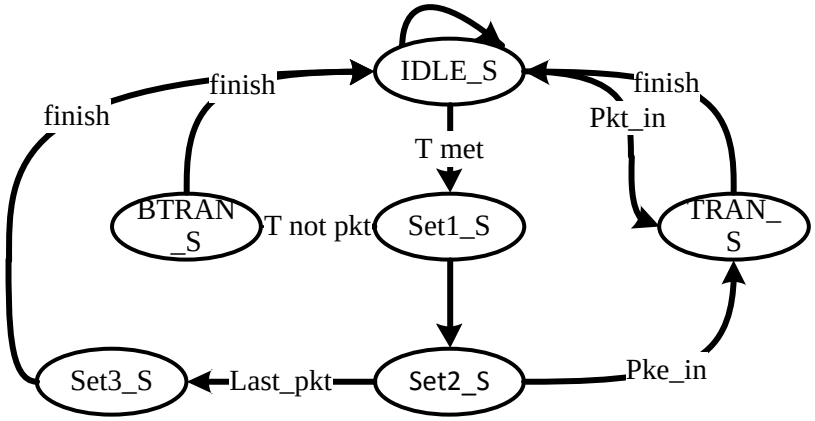


图 1-8 LReport 模块状态机

IDLE_S: 初始状态，当周期 T 未满足且有报文到达时，跳转到 TRAN_S 进行报文发送。当周期 T 满足时：若此时没有报文发送，则跳转到 Set1_S 状态，准备进行 Beacon Report 消息的发送。

TRAN_S：分组发送状态，在该状态下 UReport 进行分组的发送，直到分组全部发送完成，跳转回 IDLE_S 状态。

Set1_S: 等待状态，判断此时是否还有分组传输，若有则继续报文的传输并跳转到 Set2_S 状态；若此时没有分组传输，则表明反压成功，跳转到 BTRAN_S 状态。

Set2_S: 传输状态，将反压未成功时传输进来的分组发送给下级模块，若此时分组发送结束，则跳转到 Set3_S 状态；若此时分组还未发送结束，则跳转到 TRAN_S 状态。

Set3_S:传输最后一拍状态，将分组最后一拍传输完成之后跳转到 IDLE_S 状态。

BTRAN_S: Beacon Report 分组发送状态，调度 beacon report 报文进行发送，关于具体每一拍携带的寄存器信息详见附录一。当调度发送完成后，将跳转到 IDLE_S 状态。

3.1.2.3 LUpdate 子模块设计

LUpdate 子模块负责接收并处理来自 LReport 的 beacon update 消息，并根据 beacon update 消息更新对应的控制寄存器值。另外，还负责将收到的来自 LReport 的其它分组向 ESW 进行发送。

3.1.2.3.1 接口设计

其接口定义如图 1-9 所示。

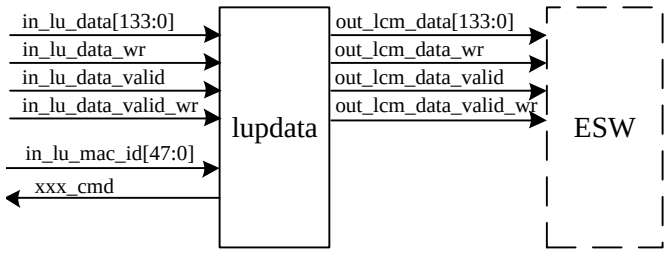


图 1-9 LUpdate 子模块接口信号图

接口信号的详细定义如表 1-3 所示：

信号名	方向	位宽	描述
LReport-LUpdate 信号定义			
in_lu_data_wr	input	1	输入到 LUpdate 的报文写信号
in_lu_data	input	134	输入到 LUpdate 的报文数据
in_lu_data_valid	input	1	输入到 LUpdate 的数据标志位
in_lu_data_valid_wr	input	1	输入到 LUpdate 数据标志写信号
in_lu_mac_id	input	1	根据引脚电平生成的 8 位 ID，用于生成 local_mac_addr。
LReport-ESW 信号定义			
out_lcm_data_wr	output	1	输出到 ESW 的报文写信号
out_lcm _data	output	134	输出到 ESW 的报文数据
out_lcm _data_valid	output	1	输出到 ESW 的数据标志位
out_lcm _data_valid_wr	output	1	输出到 ESW 数据标志写信号
LCM - other Modules 信号定义			
xxx_cmd	output	–	输出到其它模块的控制寄存器值，详见第六章或附录一

3.1.2.3.2 模块逻辑设计

根据 LUpdate 子模块功能，需要在接收到来自 ESW 模块的配置寄存器时，对本地 OpenTSN 相关的寄存器值进行配置，其逻辑使用状态机表示如图 1-10。

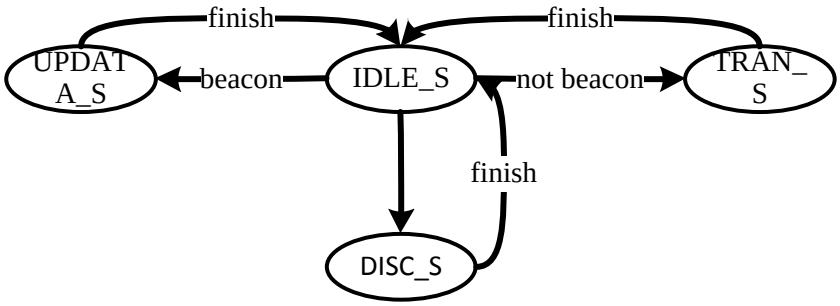


图 1-10 LUpdate 模块状态机设计

IDLE_S: 初始状态，当收到配置分组则跳转到 UPDATA_S 状态；当收到其他分组时则跳转到 TRAN_S 状态；当收到自己发送的 beacon 分组（表示该分组经过网络回环一次但未找到目的）则跳转到 DISC_S 状态。

UPDATA_S: 提取信息状态，根据分组的拍数取出第 5 拍的配置信息，跳转到 IDLE_S 状态。

DISC_S: 丢弃状态。将该分组全部丢弃完成，跳转为 IDLE_S 状态。

Trans_S: 分组发送状态。在完成报文发送后跳转回 IDLE_S 状态。

3.2 ESW 模块

3.2.1 ESW 需求与功能分析

ESW 模块主要的功能需求为：分组类型的解析、比较 MAC 地址、元数据的修改以及 TSN_MD 数据的构造、分组的转发、流量监管。

分组在进入 ESW 模块之后需要对分组进行解析，并将分组类型传输给 EOS 模块以供 EOS 模块根据分组的不同类型进入不同的队列；若分组的目的是本地直连设备，则直接由 ESW 模块发往 GOE 模块做计数；其他的分组则都往下级模块进行传输。往下级模块传输的分组还需根据流水线中的 bufm 空闲 ID 的多少进行分组的丢弃（仅丢弃非 TSN 分组）。

针对 TAP 的需求，将所有需要进行转发的分组都复制一份往 TAP 口进行转发。

3.2.2 ESW 模块详细设计

3.2.2.1 ESW 顶层模块

3.2.2.1.1 功能分析

顶层模块只是做一些接口转接。

3.2.2.1.2 接口定义

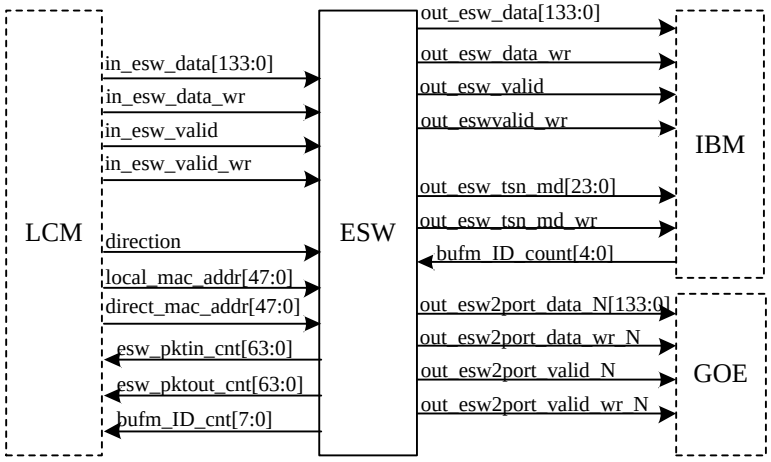


图 3-3-1-1：ESW 接口信号

表 3-3-1-1：ESW 接口定义

From LCM			
in_esw_data	134	input	分组数据输入

in_esw_data_wr	1	input	分组数据输入有效，高有效
in_esw_valid	1	input	分组数据输入有效位
in_esw_valid_wr	1	input	分组数据输入有效位有效，高有效
direction	1	input	交换机的数据流向
local_mac_addr	48	input	TSN 本地 LCM 的 MAC 地址
direct_mac_addr	48	input	TSN 本地外连设备的 MAC 地址
esw_pktin_cnt	64	output	进入 ESW 模块的分组计数器
esw_pktout_cnt	64	output	ESW 模块输出的分组计数器
bufm_ID_cnt	8	output	Bufm 中空闲的 ID 计数器
TO IBM			
out_esw_data	134	output	分组数据往下级模块输出
out_esw_data_wr	1	output	分组数据往下级模块输出有效，高有效
out_esw_valid	1	output	分组数据往下级模块输出有效位
out_esw_valid_wr	1	output	分组数据往下级模块输出有效位有效，高有效
out_esw_tsn_md	24	output	TSN_MD 数据输出
out_esw_tsn_md_wr	1	output	TSN_MD 数据输出有效信号，高有效
bufm_ID_count	5	input	Bufm 空闲的 ID 数量
TO GOE			
out_esw2port_data_N	134	output	分组数据往网口输出（N=2/3）
out_esw2port_data_wr_N	1	output	分组数据往网口输出有效，高有效（N=2/3）
out_esw2port _valid_N	1	output	分组数据往网口输出有效位（N=2/3）
out_esw2port_valid_wr_N	1	output	分组数据往网口输出有效位有效，高有效（N=2/3）

3.2.2.2 PKE 模块

3.2.2.2.1 功能分析

判断是带 VLAN 头的分组还是标准以太网分组。若是带 VLAN 头的分组则根据 PCP 值进行分组类型的区分：TSN 分组、预约带宽分组、best effort 分组；若是标准以太网分组则根据协议字段进行分组类型区分：PTP 分组、best effort 分组。并将解析的结果写到 pkttype 中；同时提取出分组的三元组{目的 MAC、源 MAC、输入端口}与分组同时传输给 PFW 模块。

3.2.2.2.2 接口定义

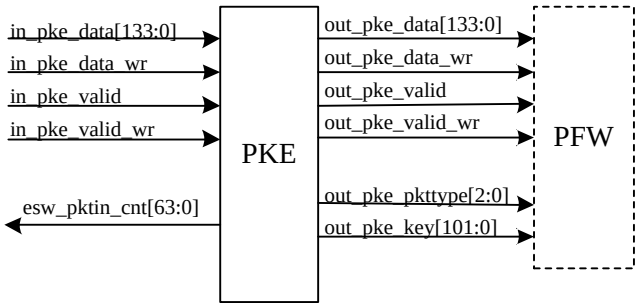


图 3-3-2-1：PKE 接口信号

表 3-3-2-1：PKE 接口定义

FROM LCM			
in_pke_data	134	input	分组数据输入
in_pke_data_wr	1	input	分组数据输入有效，高有效
in_pke_valid	1	input	分组数据输入有效位
in_pke_valid_wr	1	input	分组数据输入有效位有效，高有效
esw_pktin_cnt	64	output	进入 ESW 模块的分组计数器
TO PFW			
out_pke_data	134	output	分组数据输出
out_pke_data_wr	1	output	分组数据输出有效，高有效
out_pke_valid	1	output	分组数据输出有效位
out_pke_valid_wr	1	output	分组数据输出效位有效，高有效
out_pke_pkttype	3	output	分组协议类型信息输出
out_pke_key	102	output	分组三元组信息输出

3.2.2.2.3 模块实现

根据对 PKE 模块的功能分析，可直接用一个状态机实现：

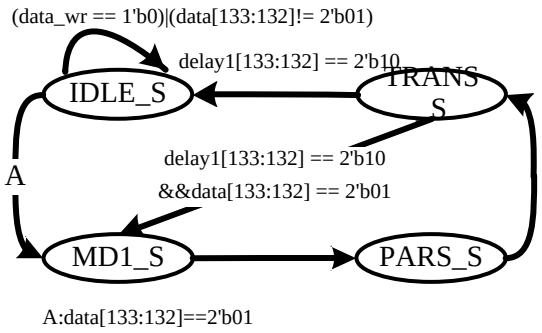


图 3-3-2-2：pke_state 状态机

IDLE_S：空闲状态机。等待分组进入子模块；确保分组是第一拍，将第一拍分组存入 delay0 寄存器当中。跳转到 MD1_S。

MD1_S：元数据传输状态机。将分组的第二拍数据继续用寄存器 delay0 进行缓存，将 delay0 寄存器的值赋给 delay1；将寄存器 delay0 中的输入端口字段提取出来填到 key[5:0]并将分组数据写入寄存器中进行缓存，跳到 **PARS_S**。

PARS_S：分组解析状态机。根据分组的该拍（以太网头）数据进行区分协议类型：若该拍的 pkt[32:16]字段值为 0x88f7，则将 pkttype 置为 2；若值为 0x8100，再根据 pkt[15:13]的值将 pkttype 分别置为 0、1、3（具体可参照 3.2.1.1）；若值为其他则将 pkttype 置为 0。将分组的 DMAC、SMAC 提取出来分别填到 key[101:54]、key[53:6]。将第一拍分组数据从寄存器中读出并传输给 PFW 模块，再将第三拍数据进行缓存，跳转到 TRANS_S。

TRANS_S：传输状态机。将分组数据传输给 PAC 模块，确保分组是正常顺序传输的，直到分组全部传输完成，可根据寄存器 delay1 的头尾标志进行判断，同时，若分组输入数据信号的头尾标志为 01（意味着下一个分组已经来到）的话则跳转到 MD1_S 并将分组输入数据信号进行缓存；若分组输入数据信号没有数据的话则跳转到 IDLE_S。

3.2.2.3 PFW 模块

3.2.2.3.1 功能分析

根据三元组的信息与本地直连设备 MAC 进行比较：源 MAC 比较：根据 3.2.1.2 节中的表项进行不同的处理方式。目的 MAC 比较：目的 MAC 为全 F、目的 MAC 等于本地直连设备 MAC、目的 MAC 不等于本地直连设备 MAC 三种结果。根据 3.2.1.2 节中的表项进行 action 字段的修改。

3.2.2.3.2 接口定义

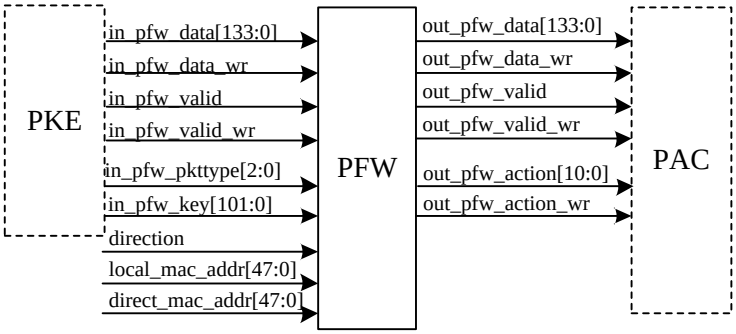


图 3-3-2-1：PKE 接口信号

表 3-3-2-1：PKE 接口定义

FROM PKE			
in_pfw_data	134	output	分组数据输入
in_pfw_data_wr	1	output	分组数据输入效，高有效
in_pfw_valid	1	output	分组数据输入效位
in_pfw_valid_wr	1	output	分组数据输入位有效，高有效
in_pfw_pkttype	3	output	分组协议类型信息输入
in_pfw_key	102	output	分组三元组信息输入
TO PAC			
out_pfw_data	134	output	分组数据输出
out_pfw_data_wr	1	output	分组数据输出有效，高有效
out_pfw_valid	1	output	分组数据输出有效位
out_pfw_valid_wr	1	output	分组数据输出效位有效，高有效
out_pfw_action	11	output	action 信息输出
out_pfw_action_wr	1	output	action 信息输出有效，高有效
FROM LCM			
direction	1	input	交换机的数据流向
local_mac_addr	48	input	TSN 本地 LCM 的 MAC 地址
direct_mac_addr	48	input	TSN 本地外连设备的 MAC 地址

3.2.2.3.3 模块实现

根据对 PKE 模块的功能分析，可直接用一个状态机实现：

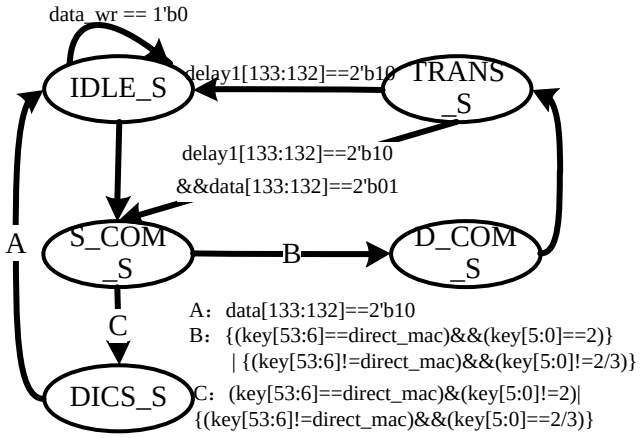


图 3-3-2-2：pke_state 状态机

IDLE_S：空闲状态机。等待分组进入子模块；再根据分组第一拍元数据的 smid 字段信息[95:88]，若值大于 4（表示该分组由本地 LCM 或本地 PTP 构造的报文）则将用一个寄存器 flag 置高进行标识；然后跳转到 S_COM_S。同时将分组第一拍数据用寄存器 delay0 进行缓存。

S_COM_S：分组源MAC比较状态机。将PKE传输过来的key[53:6]与direct_mac进行比较，若{(key[53:6]==direct_mac)&&(key[5:0]==2)})则跳转到D_COM_S；若{(key[53:6]==direct_mac)&(key[5:0]==2)}则跳转到DICS_S。若{(key[53:6]！=direct_mac)&&（(key[5:0]==2)‖ (key[5:0]==3)）}则跳转到DICS_S。若{(key[53:6]！=direct_mac)&&（(key[5:0]！=2)‖ (key[5:0]！=3)）}则跳转到D_COM_S。

若源MAC与本地直连设备地址相等，将寄存器flag置高。同时将分组第二拍数据用寄存器delay0进行缓存，寄存器delay0的值赋值给delay1。

D_COM_S：分组目的 MAC 比较状态机。将 PKE 传输过来的 key[101:54]与 direct_mac 进行比较，若 key[101:54]==direct_mac 则将 action[5:0]填为 2；若 key[101:54]！=direct_mac 则再根据 flag 位：若 flag 为低，则将 action[5:0]填为~key[0]；若 flag 为高（代表分组是来自本地直连设备或本地 LCM），则将 action[5:0]填为 direction。若 key[101:54]==FF:FF:FF:FF:FF:FF 则根据 flag 位：若 flag 为低，则将 action[5:0]填为~key[0]；若 flag 为高，则将 action[5:0]填为 direction。跳转到 TRANS_S。将分组第一拍数据进行传输并将第三拍数据进行缓存。

TRANS_S：传输状态机。将分组数据传输给 PAC 模块，直到分组全部传输完成，可根据寄存器 delay1 的头尾标志进行判断，同时，若分组输入数据信号的头尾标志为 01（意味着下一个分组已经来到）的话则跳转到 S_COM_S 并将分组输入数据信号进行缓存；若分组输入数据信号没有数据的话则跳转到 IDLE_S。。

DICS_S：丢弃状态机。将分组进行丢弃，直到分组全部丢弃完成（根据输入分组数据信号的头尾标准判断），同时将 delay0 以及 delay1 初始化，跳转到 IDLE_S。

3.2.2.4 PAC 模块

3.2.2.4.1 功能分析

根据 ACTION 中的内容进行分组的转发。单播时：若 ACTION 的输出端口为 2 则

将分组元数据根据 ACTION 进行修改之后将分组往本地直连设备传输；若 ACTION 的分组输出端口不为 2 则根据 bufm 的空闲 ID 进行分组的丢弃或转发。当 bufm 大于 2，分组都可直接进行元数据修改之后传输分组；当 bufm 大于 1 小于 2 时，分组类型为 best effort 的需要进行丢弃，其他分组进行正常传输；当 bufm 小于 1 时，分组类型为预约带宽或 best effort 的都需要进行丢弃，其他分组进行正常传输。不丢弃时，在分组传输时需要构造 TSN_MD。广播时：处理与单播一样的处理方式之外还需要将分组复制一份往 2 号口进行转发。在所有需要进行转发的分组都复制一份转发给 TAP 口。

3.2.2.4.2 接口定义

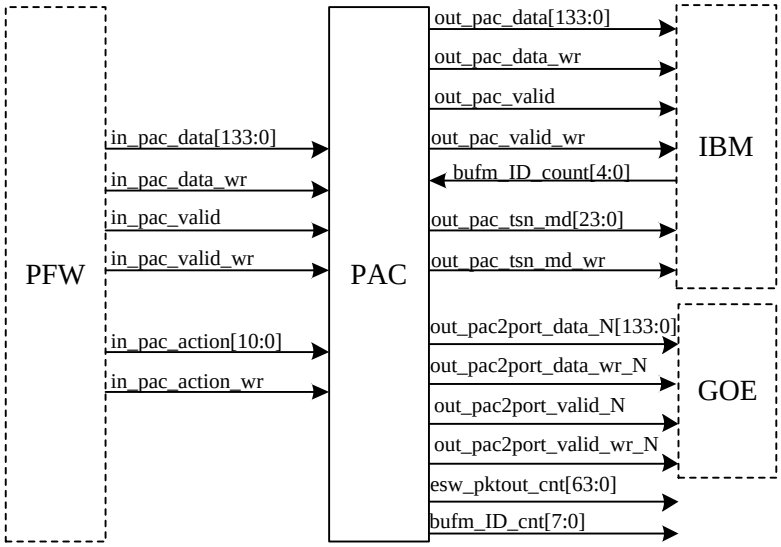


图 3-3-3-1：PAC 接口信号

表 3-3-3-1：PAC 接口定义

From PFW			
in_pac_data	134	input	分组数据输入
in_pac_data_wr	1	input	分组数据输入有效，高有效
in_pac_valid	1	input	分组数据输入有效位
in_pac_valid_wr	1	input	分组数据输入效位有效，高有效
in_pac_action	11	input	action 数据输入
in_pac_action_wr	1	input	action 数据输入有效，高有效
TO IBM			
out_pac_data	134	output	分组数据往下级模块输出
out_pac_data_wr	1	output	分组数据往下级模块输出有效，高有效
out_pac_valid	1	output	分组数据往下级模块输出有效位

out_pac_valid_wr	1	output	分组数据往下级模块输出效位有效，高有效
out_pac_tsn_md	24	output	TSN_MD 数据输出
out_pac_tsn_md_wr	1	output	TSN_MD 数据输出有效，高有效
bufm_ID_count	5	input	Bufm 中空闲的 ID 计数器
TO LCM			
bufm_ID_cnt	8	output	ESW 模块输出的 bufm 空闲 ID 计数器
esw_pktout_cnt	64	output	ESW 模块输出分组计数器
TO GOE			
out_pac2port_data_N	134	output	分组数据往网口输出（N=2/3）
out_pac2 port_data_wr_N	1	output	分组数据往网口输出有效，高有效（N=2/3）
out_pac2 port_valid_N	1	output	分组数据往网口输出有效位（N=2/3）
out_pac2 port_valid_wr_N	1	output	分组数据往网口输出效位有效，高有效（N=2/3）

3.2.2.4.3 模块实现

根据对 PAC 模块的功能分析，可直接用一个状态机实现：

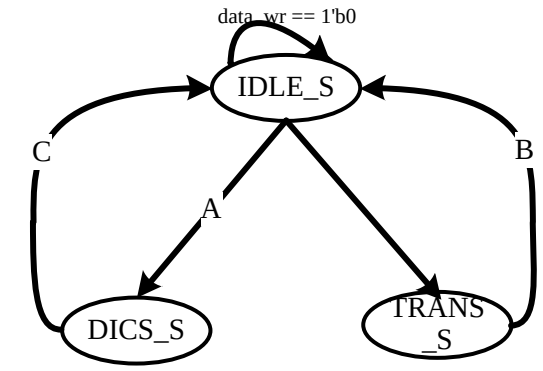


图 3-3-3-2：pac_state 状态机

IDLE_S：空闲状态机。等待分组进行，若满足{(action[8:6]==0)&&Bufm_ID_cnt==3}|{(action[8:6]==0/1)&&Bufm_ID_cnt=1/2}|{ Bufm_ID_cnt=0}则跳转到DICS_S；否则跳转到TRANS_S。将分组第一拍使用delay0进行缓存，同时将输出端口写入到delay0中，并构造TSN_MD。

TRANS_S：网口传输状态机。根据 action[10:9]判断分组是单播还是广播。单播时，

判断输出端口是否为 2，若是 2 则将分组往 2 号口传输；若不是 2 则将分组往下级模块传输。广播时，将分组往 2 口和下级模块都传输。根据 delay0 的头尾标志进行判断数据是否传输完成，完成之后跳转到 IDLE_S。

DICS_S：丢弃状态机。将分组进行丢弃，直到分组全部丢弃完成（根据分组输入数据信号的头尾标准进行判断），跳转到 IDLE_S。

由于还需将所有的分组复制一份发往 TAP 口，因此添加一个 always 块来实现：根据两组输出信号的“数据有效”信号来判断当前传输的分组是什么流向的，则相应地将这组输出信号复制一份给 TAP 口；若两组“数据有效”信号都为高，则将其中一组的数据复制给 TAP 口即可。

3.3 EOS 模块

3.3.1 EOS 需求与功能分析

EOS 模块主要的功能是实现 CQF 调度机制，同时对于预约带宽的分组需要利用令牌桶对分组的速率进行限制，若分组速率超过了要求则将分组进行丢弃。

在模块中有时间片的翻转、令牌桶的桶深和令牌桶的速率这三个参数需要 LCM 进行配置。可通过 LCM 直接连三根线进行配置即可。

3.3.2 EOS 模块详细设计

3.3.2.1 EOS 顶层模块设计

3.3.2.1.1 模块接口设计

EOS 顶层模块接口定义如图 3-3-1 所示，具体接口信号的含义如下表 3-3-1 所示：

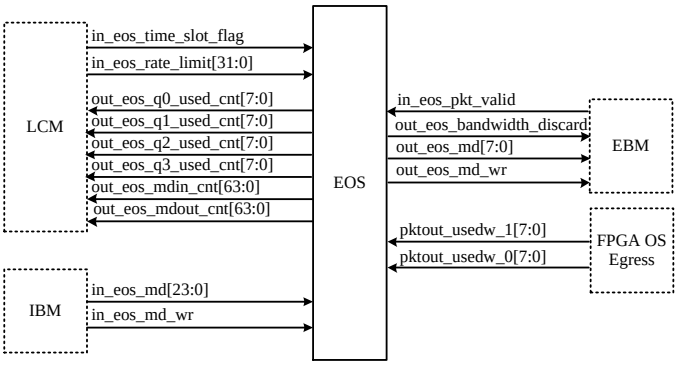


图 3-3-1　EOS 模块外部接口信号定义

表 3-3-1　EOS 模块外部接口信号定义

接口信号	位宽	方向	含义
clk	1	input	时钟
rst_n	1	input	复位，低有效
EOS-LCM 信号定义			
in_eos_time_slot_flag	1	input	时间槽切换信号，当前位于奇数时间

			槽为 1，偶数时间槽为 0
in_eos_rate_limit	32	input	每 800ns 往令牌桶中添加令牌的数量；支持的最小带宽为 10Mbps
out_eos_q0_used_cnt	8	output	MB 模块 Q0 队列的已使用长度计数器
out_eos_q1_used_cnt	8	output	MB 模块 Q1 队列的已使用长度计数器
out_eos_q2_used_cnt	8	output	MB 模块 Q2 队列的已使用长度计数器
out_eos_q3_used_cnt	8	output	MB 模块 Q3 队列的已使用长度计数器
out_eos_mdin_cnt	64	output	进入 EOS 模块的元数据计数器
out_eos_mdout_cnt	64	output	EOS 模块输出的元数据计数器
EOS-IBM 信号定义			
in_eos_md	24	input	元数据输入
in_eos_md_wr	1	input	元数据输入有效信号
EOS-FPGA OS Egress 信号定义			
pktout_usedw_0	8	input	0 号端口 fifo 已存储的报文深度
pktout_usedw_1	8	input	1 号端口 fifo 已存储的报文深度
EOS-EBM 信号定义			
out_eos_md	8	output	元数据输出
out_eos_md_wr	1	output	元数据输出有效信号
in_eos_pkt_valid	1	input	EBM 中分组最后一拍输出信号
out_eos_bandwidth_discard	1	output	带宽预约分组丢弃信号

3.3.2.1.2 模块实现

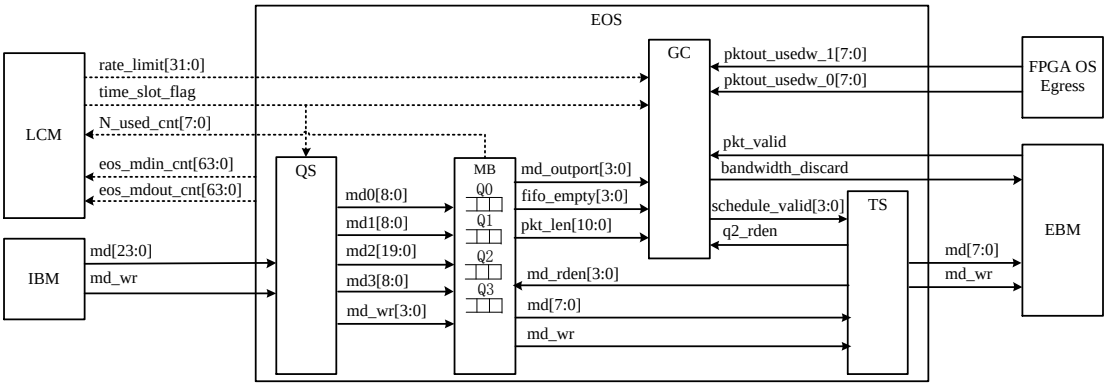


图 3-3-2 EOS 模块的整体架构的所有接口信号

注：图中 MB 传给 LCM 的 N_used_cnt[7:0]中的 N 为 q0、q1、q2、q3。

需要一个 always 块来实现对进入 EOS 模块的元数据和 EOS 模块输出的元数据进行计数。

3.3.2.2 QS 模块设计

QS 模块功能是选择将元数据写入的队列。根据当前时间槽的奇偶和元数据 [23:21] 类型决定将元数据传输给 MB 模块 4 个队列中的哪一个。

3.3.2.2.1 模块接口设计

QS 模块接口定义如图 3-3-3 所示，具体接口信号的含义如下表 3-3-2 所示：

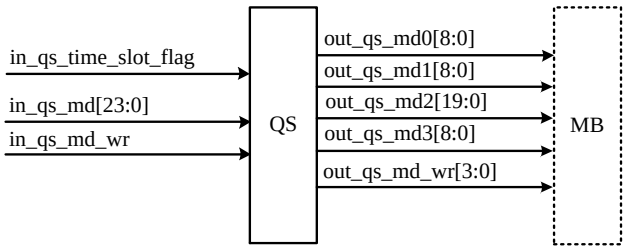


图 3-3-3 QS 模块接口信号定义

表 3-3-2 QS 模块接口信号定义

接口信号	位宽	方向	含义
clk	1	input	时钟
rst_n	1	input	复位，低有效
QS-EOS 信号定义			
in_qs_time_slot_flag	1	input	时间槽切换信号，当前位于奇数时间槽为 1，偶数时间槽为 0
in_qs_md	24	input	元数据输入
in_qs_md_wr	1	input	元数据输入有效信号

QS-MB 信号定义			
out_qs_md0	9	output	Q0 队列元数据输出
out_qs_md1	9	output	Q1 队列元数据输出
out_qs_md2	20	output	Q2 队列元数据输出
out_qs_md3	9	output	Q3 队列元数据输出
out_qs_md_wr	4	output	Q0、Q1、Q2、Q3 队列元数据输出有效信号

3.3.2.2.2 模块实现

根据上述的 QS 子模块元数据的队列选择功能，可使用一个 always 块来实现。具体功能实现如下。

等待元数据到来，判断该元数据[23:21]类型：

若为 3 且当前处于奇数时间槽，则将输出到 Q1 队列的写信号置 1，元数据[8:0] 输出到 Q1 队列；

若为 3 且当前处于偶数时间槽，则将输出到 Q0 队列的写信号置 1，元数据[8:0] 输出到 Q0 队列；

若为 2，则将输出到 Q2 队列的写信号置 1，同时将元数据[19:9]pkt_len 置 0（PTP 分组不需要进行流量整形，不需要消耗令牌），并连同元数据[8:0]输出到 Q2 队列；

若为 1，则将输出到 Q2 队列的写信号置 1，同时将元数据[19:9]pkt_len 减去两拍 metadata 长度（32 字节），并连同元数据[8:0]输出到 Q2 队列；

若为 0，则将输出到 Q3 队列的写信号置 1，元数据[8:0]输出到 Q3 队列。

3.3.2.3 MB 模块设计

MB 模块进行元数据缓存，Q0 为缓存 TSN 元数据的偶数时钟队列，Q1 为缓存 TSN 元数据的奇数时钟队列，Q2 为缓存带宽预约元数据和 PTP 元数据的队列，Q3 为缓存 best effort 元数据的队列。等待 TS 的元数据读信号，将相应队列的元数据读出。

3.3.2.3.1 模块接口设计

MB 模块接口定义如图 3-3-4 所示，具体接口信号的含义如表 3-3-3 所示：

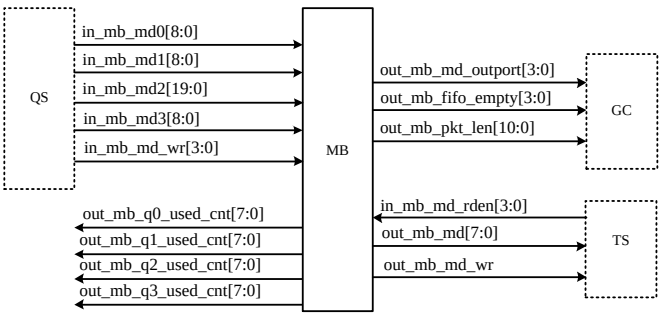


图 3-3-4 MB 模块接口信号定义

表 3-3-3　MB 模块接口信号定义

接口信号	位宽	方向	含义
clk	1	input	时钟
rst_n	1	input	复位，低有效
MB-EOS 信号定义			
out_mb_q0_used_cnt	8	output	MB 模块 Q0 队列的已使用长度计数器
out_mb_q1_used_cnt	8	output	MB 模块 Q1 队列的已使用长度计数器
out_mb_q2_used_cnt	8	output	MB 模块 Q2 队列的已使用长度计数器
out_mb_q3_used_cnt	8	output	MB 模块 Q3 队列的已使用长度计数器
MB-QS 信号定义			
in_mb_md0	9	input	Q0 队列元数据输入
in_mb_md1	9	input	Q1 队列元数据输入
in_mb_md2	20	input	Q2 队列元数据输入
in_mb_md3	9	input	Q3 队列元数据输入
in_mb_md_wr	4	input	Q0、Q1、Q2、Q3 队列元数据输入有效信号
MB-TS 信号定义			
out_mb_md	8	output	队列元数据输出
out_mb_md_wr	1	output	队列元数据输出有效信号
in_mb_md_rden	4	input	Q0、Q1、Q2、Q3 队列元数据读使能信号
MB-GC 信号定义			
out_mb_md_outport	4	output	Q0、Q1、Q2、Q3 队列元数据输出端口
out_mb_fifo_empty	4	output	Q0、Q1、Q2、Q3 队列空信号
out_mb_pkt_len	11	output	Q2 队列元数据对应的报文需消耗的令牌数（1 字节消耗 1 个令牌）

3.3.2.3.2 模块实现

MB 模块只是进行元数据的缓存，实现较简单，不需要使用状态机；关键是给 4 个

队列的 fifo 设置合适的位宽和深度，分析计算过程如下。

MB 模块任意时刻的元数据最大缓存量的分析计算：为了给 Q0 和 Q1 队列设置合适的队列长度和给 BufM 缓存区分配合适的 ID 数量， 需要计算 MB 模块任意时刻的元数据最大缓存量，因为 BufM 的一个 ID 对应 MB 的一个元数据，并且一个 ID 对应 2Kb 的存储空间，足够缓存一个最长分组。根据乒乓队列的读写特点，只要保证在一个时间槽内 BufM 能缓存全部的 TSN 分组便可，所以只需计算 MB 模块在一个时间槽内 TSN 元数据的最大缓存量；考虑极端情况，在一个时间槽内，当从端口进入的 TSN 报文全为最短报文（64 字节），以太网最小帧间距是 12 字节，以太帧前导码为 7 字节、帧开始符为 1 字节，且 TSN 流量达到链路负载的最大比例 5%（50Mbps）时，MB 模块的 TSN 元数据的缓存量最大。假设时间敏感流量乒乓队列切换的时间槽为 125μs（802.1Qch 中给出的典型切换时间），因此 Q0 和 Q1 每个队列缓存深度（ID 数量）最大需要

$$\frac{125\mu s * 5\% * 1Gbps}{(64B + 12B + 8B) * 8bit/B} = 10$$

由上述分析知，Q0 和 Q1 队列长度最大需要 10，fifio 深度可设置为 16；Q2、Q3 队列的 fifo 深度不应低于 BufM 中 ID 数量，可设置为 16。

综上分析：4 个队列的 fifo 位宽、深度设置如下：

表 3-3-4 4 个队列的 fifo 参数

队列 fifo	位宽	深度
Q0	9	16
Q1	9	16
Q2	20	16
Q3	9	16

3.3.2.4 GC 模块设计

GC 模块负责令牌桶中的令牌计数；判断 MB 的 4 个队列是否可被调度，把判断结果输给 TS 模块；判断被调度的带宽预约流量是否要被丢弃，把判断结果输给 EBM 模块。

3.3.2.4.1 模块接口设计

GC 模块接口定义如图 3-3-5 所示，具体接口信号的含义如表 3-3-5 所示：

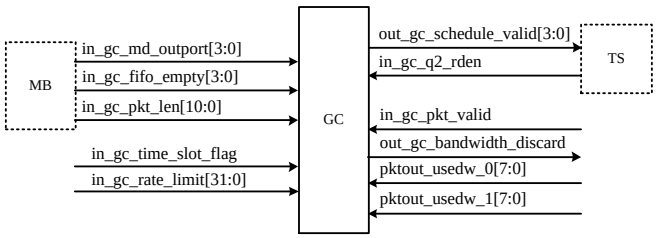


图 3-3-5 GC 模块接口信号定义

表 3-3-5　GC 模块接口信号定义

接口信号	位宽	方向	含义
clk	1	input	时钟
rst_n	1	input	复位，低有效
GC-EOS 信号定义			
in_gc_time_slot_flag	1	input	时间槽切换信号，当前位于奇数时间槽为 1，偶数时间槽为 0
in_gc_rate_limit	32	input	每 800ns 往令牌桶中添加令牌的数量；支持的最小带宽为 10Mbps
in_gc_pkt_valid	1	input	EBM 中分组最后一拍输出信号
out_gc_bandwidth_discard	1	output	带宽预约分组丢弃信号
pktout_usedw_0	8	input	0 号端口 fifo 已存储的报文深度
pktout_usedw_1	8	input	1 号端口 fifo 已存储的报文深度
GC-MB 信号定义			
in_gc_md_outport	4	input	Q0、Q1、Q2、Q3 队列元数据输出端口
in_gc_fifo_empty	4	input	Q0、Q1、Q2、Q3 队列空信号
in_gc_pkt_len	11	input	Q2 队列元数据对应的报文需消耗的令牌数（1 字节消耗 1 个令牌）
GC-TS 信号定义			
in_gc_q2_rden	1	input	Q2 元数据的读信号，用于判断 TS 是否调度了 Q2 的元数据，令牌桶中是否应减去相应的令牌
out_gc_schedule_valid	4	output	Q0、Q1、Q2、Q3 队列元数据可调度信号，高有效

3.3.2.4.2 模块实现

门控模块负责令牌桶中的令牌计数；并根据 pkt_valid、time_slot_flag、fifo_empty、md_outport 和 pktout_usedw_N(N=0、1)来判断 MB 的 4 个队列是否可被调度，把判断结果输给 TS 模块；当带宽预约流量被调度时，根据令牌桶中的剩余令牌数与报文字节数的大小关系来决定该报文是否应在 EBM 中被丢弃。

为了实现上述功能，需要 3 个 always 块。每个 always 块的具体功能实现如下。

1. 一个 always 块用来判断 Q0、Q1、Q2、Q3 是否可被调度，可使用一个状态机进行控制。其状态机如图 3-3-6 所示。

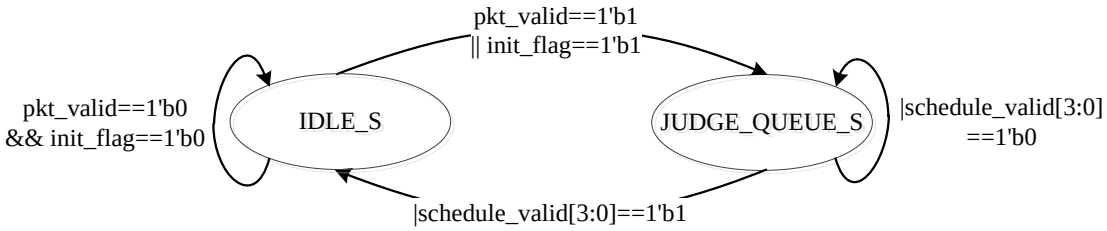


图 3-3-6 判断 4 个队列可被调度状态机

IDLE_S: 起始状态。当 pkt_valid==1'b1 || init_flag==1'b1 时，跳转到 JUDGE_Q0_S。（init_flag 作用：为了使 FPGA 开始工作时，队列能被调度，设置寄存器 init_flag，复位时置为 1。）

JUDGE_QUEUE_S: 判断状态。若|schedule_valid[3:0]==1'b1，将 schedule_valid[3:0]置 0，init_flag 置 0，跳转到 IDLE_S；否则留在 JUDGE_QUEUE_S，并判断 Q0、Q1、Q2、Q3 是否可被调度，具体判断情况如下。

判断 Q0 是否可被调度，若满足以下 3 个条件：

1）time_slot_flag = =1'b1；

2）Q0 的 empty == 1'b0；

3）输出端口为 N 且 pktout_usedw_N<a；

则将 schedule_valid[0]置 1；否则将 schedule_valid[0]置 0。

判断 Q1 是否可被调度，若满足以下 3 个条件：

1）time_slot_flag = =1'b0；

2）Q1 的 empty = = 1'b0；

3）输出端口为 N 且 pktout_usedw_N<a；

则将 schedule_valid[1]置 1；否则将 schedule_valid[1]置 0。

判断 Q2 是否可被调度，若满足以下 2 个条件：

1）Q2 的 empty = = 1'b0；

2）输出端口为 N 且 pktout_usedw_N<a；

则将 schedule_valid[2]置 1，；否则将 schedule_valid[2]置 0。

判断 Q3 是否可被调度，若满足以下 2 个条件：

1）Q3 的 empty = = 1'b0；

2）输出端口为 N 且 pktout_usedw_N<a；

则将 schedule_valid[3]置 1；否则将 schedule_valid[3]置 0。

2. 一个 always 块用来对令牌桶中令牌进行计数。设置 1Byte 消耗令牌桶中一个令牌；令牌桶的容量为 BC(可容纳 2047 个令牌)，令牌桶中剩余令牌为 RT，往令牌桶中加令牌的速率为 rate(bps)，在调度 Q2 队列元数据时，需消耗的令牌为 CT，令牌桶的计时器为 TB_cnt。每经 100*8ns(TB_cnt 计 100 个时钟下降沿)，往令牌桶中添加

$$\mathrm{AT} = 100 * 8ns * \frac{\mathrm{rate}}{8bit/B * 10^9 ns/s} = \frac{\mathrm{rate}}{10\mathrm{Mbps}}$$

个令牌(支持的最小带宽为 10Mpbs)；若 RT+AT-CT<BC，添加令牌后令牌桶中的剩余令牌为 RT+AT-CT；若 RT+AT-CT≥BC(令牌数量超过桶的容量)，添加令牌后令牌桶中的剩余令牌为 BC。

3. 一个 always 块用来判断令牌是否要被消耗，带宽预约流量是否要被丢弃。若 TS 输出给 Q2 的读信号 q2_rden 为 1 且 RT≥报文字节数，则将该报文字节数赋给 CT，out_gc_bandwidth_discard 置 0(在 EBM 模块不丢弃该带宽预约流量)；若 q2_rden 为 1 且 RT<报文字节数，则将 CT 置 0，out_gc_bandwidth_discard 置 1(在 EBM 模块丢弃该带宽预约流量)。

3.3.2.5 TS 模块设计

TS 模块根据 GC 模块传来的判断结果，对可进行调度的队列采用绝对优先级调度策略选择发送元数据，优先级为 TSN 元数据>带宽预约元数据=PTP 元数据>best effort 元数据；若 Q0 或 Q1 队列是可被调度的，则将该队列的元数据读出；若 Q0 或 Q1 队列是不可被调度的，Q2 队列是可被调度的，则将 Q2 的元数据读出；若 Q0 或 Q1 队列、Q2 队列是不可被调度的，Q3 队列是可被调度的，则将 Q3 的元数据读出。

3.3.2.5.1 TS 模块接口设计

TS 模块接口定义如图 3-3-7 所示，具体接口信号的含义如表 3-3-6 所示：

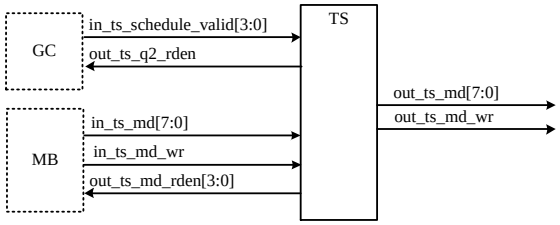


图 3-3-7 TS 模块接口信号定义

表 3-3-6 TS 模块接口信号定义

接口信号	位宽	方向	含义
clk	1	input	时钟
rst_n	1	input	复位，低有效
TS-MB 信号定义			
in_ts_md	8	input	元数据输入
in_ts_md_wr	1	input	元数据输入有效信号
out_ts_md_rden	4	input	Q0、Q1、Q2、Q3 队列元数据读信号
TS-GC 信号定义			
in_ts_schedule_valid	4	input	Q0、Q1、Q2、Q3 队列元数据可被调度信号
out_ts_q2_rden	1	output	TS 输出给 Q2 的读信号，用于判断 TS 是否调度了 Q2 的元数据，在 GC 模块中令牌桶中是否应减去相应的令牌

TS-EOS 信号定义			
out_ts_md	8	output	元数据输出
out_ts_md_wr	1	output	元数据输出有效信号

3.3.2.5.2 模块实现

根据上述的 TS 子模块队列元数据调度发送的功能，可使用 2 个 always 块实现。每个 always 块的具体功能实现如下。

1. 一个 always 块用于实现队列调度功能。若 Q0 的 schedule_valid 为 1，则将输出给 Q0 队列的读信号置 1；若 Q1 的 schedule_valid 为 1，则将输出给 Q1 队列的读信号置 1；若 Q0、Q1 的 schedule_valid 都为 0，Q2 的 schedule_valid 为 1，则将输出给 Q2 队列的读信号置 1；若 Q0、Q1、Q2 的 schedule_valid 都为 0，Q3 的 schedule_valid 为 1，则将输出给 Q3 队列的读信号置 1。
2. 一个 always 块用于输出元数据。等待输入的元数据有效信号为 1，将元数据的[7:0]输出。

3.3.2.5.3 UDO 端口 a 的计算

为了使报文到达 UDO 相应端口时，该端口有足够的空闲空间来存储报文，需计算 UDO 中的 fifo 剩余报文拍数 a 最大为多少时，TS 才能调度元数据。在 UDO 中端口每个时钟周期输出 8bit，传 128bit 需要 16 个时钟周期，从 TS 发出读信号给元数据缓存队列到报文写入相应端口的 fifo 需要 16 个时钟周期左右，这段时间内端口发完一个 128bit；GOE 到 UDO 至多 2 拍；而 fifo 深度 128-最长报文拍数 95=33，所以 a 为 33+1-2=32。在实际工程中，将 a 设为 20。

3.4 GOE 模块

3.4.1 GOE 需求与功能分析

对分组的输出端口进行区分，并将分组根据输出端口进行转发。

分组的输出端口只可能为 0 或 1；根据分组第一拍元数据的“输出端口”字段进行判断，将分组输出到 FPGA_OS 对应的网口。

对传输的分组进行计数，包括全部发往 4 个网口的分组计数。

3.4.2 GOE 模块详细设计

3.4.2.1 功能分析

该模块只是将分组按照输出端口进行转发，直接根据元数据的“outport”字段信息便和进行区分。

3.4.2.2 接口定义

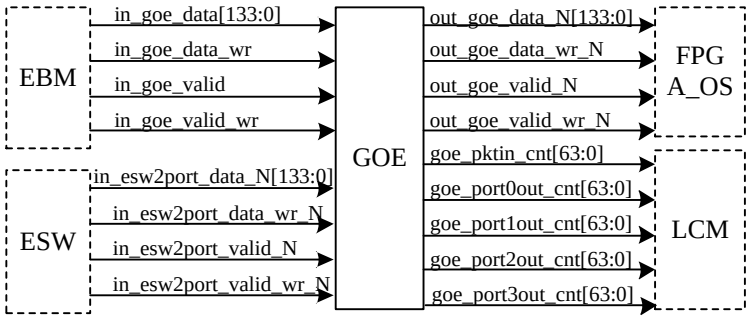


图 5-3-1：GOE 模块接口信号

表：5-3-1：GOE 模块接口定义

对接模块	名称	位宽	走向	含义
EBM	in_goe_data	134	input	分组数据输入
	in_goe_data_wr	1	input	分组数据输入有效，高有效
	in_goe_valid	1	input	分组数据输入有效位
	in_goe_valid_wr	1	input	分组数据输入有效位有效，高有效
ESW	in_esw2port_data_N	134	input	分组数据输入（N=2/ 3）
	in_esw2port_data_wr_N	1	input	分组数据输入有效，高有效（N=2/ 3）
	in_esw2port_valid_N	1	input	分组数据输入有效位（N=2/ 3）
	in_esw2port_valid_wr_N	1	input	分组数据输入有效位有效，高有效（N=2/ 3）
LCM	goe_pktin_cnt	64	output	进入 GOE 模块的分组计数器
	goe_port0out_cnt	64	output	GOE 模块往 0 端口输出分组计数器
	goe_port1out _cnt	64	output	GOE 模块往 1 端口输出分组计数器
	goe_port2out_cnt	64	output	GOE 模块往 2 端口输出分组计数器
	goe_port3out _cnt	64	output	GOE 模块往 3 端口输出分组计数器
FPGA_OS	out_goe_data_N	134	output	分组数据往网口输出（N=0 到 3）
	out_goe_data_wr_N	1	output	分组数据往网口输出有效，高有效（N=0 到 3）
	out_goe_valid_N	1	output	分组数据往网口输出有效位（N=0 到 3）
	out_goe_valid_wr_N	1	output	分组数据往网口输出有效位有效，高有效（N=0 到 3）

3.4.2.3 模块实现

根据对模块的分析，传输功能的实现用一个状态机便可实现功能。

状态机的设计如下：

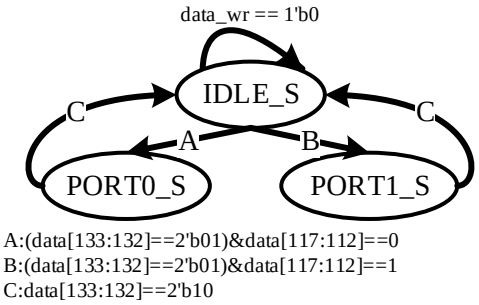


图 5-3-2：goe_state 状态机

IDLE_S：空闲状态机。等待分组进入子模块；根据分组第一拍元数据的[117:112]位信息判断分组的输出端口是 0 还是 1：是 0 则跳转到 PORT0_S；是 1 则跳转到 PORT1_S。同时需要将分组的第一拍传输给 FPGA_OS。

PORT0_S：分组往 0 号口传输状态机。将分组数据进行传输，直到分组全部传输完成，跳转到 IDLE_S。

PORT1_S：分组往 1 号口传输状态机。将分组数据进行传输，直到分组全部传输完成，跳转到 IDLE_S。

由 ESW 模块传输过来的两组分组数据信号则直接往 FPGA_OS 进行传输。

5 个计数器的设计可直接根据数据的各组信号的“valid_wr”信号计数，每当该信号为高一次便对计数器进行加一。

四、可读写寄存器设计

CNC 可配置/读写寄存器均迁移到 LCM 中实现。LCM 向其它模块提供对应的寄存器输出信号。如果维持所有寄存器作为周期性上报信息，则需要扩展 PTP 报文的数据字段，详见附录一。所需要交互的寄存器信息如下表所示：

读写	信号名	含义	位宽	来源/目的模块
读写	Direction	设备口接收报文的转发方向，0 为 0 口转发，1 为 1 口转发。	1	ESW, EOS
只读	Flag	置为 1 为写响应报文，为 0 为正常 report 报文。	1	LCM
读写	Token_bucket_para	[31:0]为令牌桶 r。	32	EOS
读写	Direct_mac_addr	TSN 节点直连设备 MAC 地址。	48	ESW

只读	Esw_pktin_cnt	进入 ESW 模块的分组计数器。	64	ESW
只读	Esw_pktout_cnt	ESW 模块输出的分组计数器。	64	ESW
只读	Local_mac_addr	TSN 本地 MAC 地址。	8	ESW
只读	Time_slot_flag	奇偶时间槽，当前位于奇数时间槽为 1；偶数时间槽为 0。	1	EOS
只读	Eos_mdin_cnt	进入 EOS 模块的元数据计数器	64	EOS
只读	Eos_mdout_cnt	EOS 模块输出的元数据计数器	64	EOS
只读	Eos_q0_used_cnt	EOS 模块 Q0 队列已使用长度计数器	8	EOS
只读	Eos_q1_used_cnt	EOS 模块 Q1 队列已使用长度计数器	8	EOS
只读	Eos_q2_used_cnt	EOS 模块 Q2 队列已使用长度计数器	8	EOS
只读	Eos_q3_used_cnt	EOS 模块 Q3 队列已使用长度计数器	8	EOS
只读	Goe_pktin_cnt	进入 GOE 模块的分组计数器	64	GOE
只读	Goe_port0out_cnt	GOE 模块往 0 口输出的分组计数器	64	GOE
只读	Goe_port1out_cnt	GOE 模块往 1 口输出的分组计数器	64	GOE
只读	Goe_discard_cnt	GOE 模块丢弃的分组计数器	64	GOE
只读	Bufm_ID_cnt	Bufm 中所使用的 ID 计数器	8	ESW

附录一：Beacon 协议与报文设计

1、Beacon 消息通信模型

在 OpenTSN 中，我们使用 beacon 协议支持两种关键功能：

1）TSN 节点当前状态信息周期性上报；

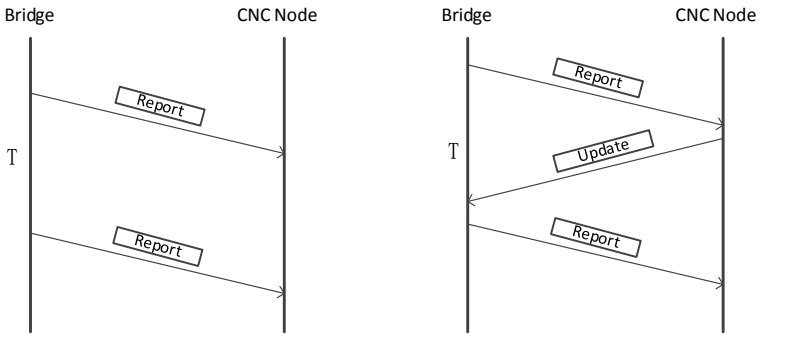
TSN 节点周期性将 CNC 节点关注的本地状态信息填写到 beacon 协议报文（扩展的 PTP 报文）中，并上传 CNC 节点，其中包括各队列的利用率，各端口接收与发送的分组数等等 。另外，通过周期性上报机制可对 TSN 节点是否正常运行进行检测。

2）CNC 节点对 TSN 节点进行配置；

CNC 节点可将 TSN 节点的需配置信息（直连设备 MAC 地址、分组转发方向、时间片大小等）写入 beacon 报文中并发送到目的端，TSN 节点接收到该报文后对本地相关寄存器进行修改。

综上，功能一与功能二可以使用相同的发送时间周期，因此可集成在同一协议中实现（借助 PTP 协议报文进行扩展）。而为了简化设计，我们考虑将功能三同样集成在扩

展的 PTP 的周期上报报文中。目前的 PTP 协议设计中保留了很多 reserved 域，可被我们用于寄存器读写的地址和数据。



（a）周期性上报场景　　（b）写寄存器场景

图 6-1 CNC-Bridge 通过 NMAC 协议的通信场景

beacon 在 OpenTSN 中支持两种 CNC-Bridge 的通信场景：周期性上报场景与写寄存器场景。周期性上报场景如图 6-1（a）所示，beacon 协议支持 TSN 节点对本地参数（如规则表项、计数器等）向 CNC 集中控制节点以固定时间周期 T 进行上报。而写寄存器场景与图 6-1（b）所示，支持 CNC 对 Bridge 设备的流表规则、本地 MAC 地址进行写入或配置。需要注意的是，beacon 的写报文采用“完全覆盖”的方式对寄存器进行写操作，即当 TSN 节点收到 CNC 写报文时，将报文中对应的寄存器值全部替换寄存器的当前值。

2、Beacon 协议报文格式设计

本节介绍 BEACON 协议报文格式的详细设计。BEACON 协议定义了远端 CNC 节点与 Bridge 节点的通信方式，目前包含两种消息类型：Report Message 与 Update Message。两种消息的 MsgType 分别为 0x1F 与 0x2F。

两种消息均基于扩展的 PTP 协议报文实现，其定义如图 6-1 所示。其中白色字段为 PTP 协议使用字段，灰色部分为修改字段，绿色部分为数据扩展字段。数据扩展字段从第 65 字节起始（FAST 传输报文数据的第 5 拍）。根据第六章读写寄存器详细设计，需要上报的状态信号与 ESW, EOS 与 GOE 三个模块相关。我们为每个模块分配 256 位的 payload 大小用于存储状态信息。另外，再分配 128 位 payload 用于可读写寄存器的配置，从 PTP 报文的第 65 字节起算，具体定义如表 6-1 所示，并在表 6-2 给出 PTP 同步报文的格式定义。

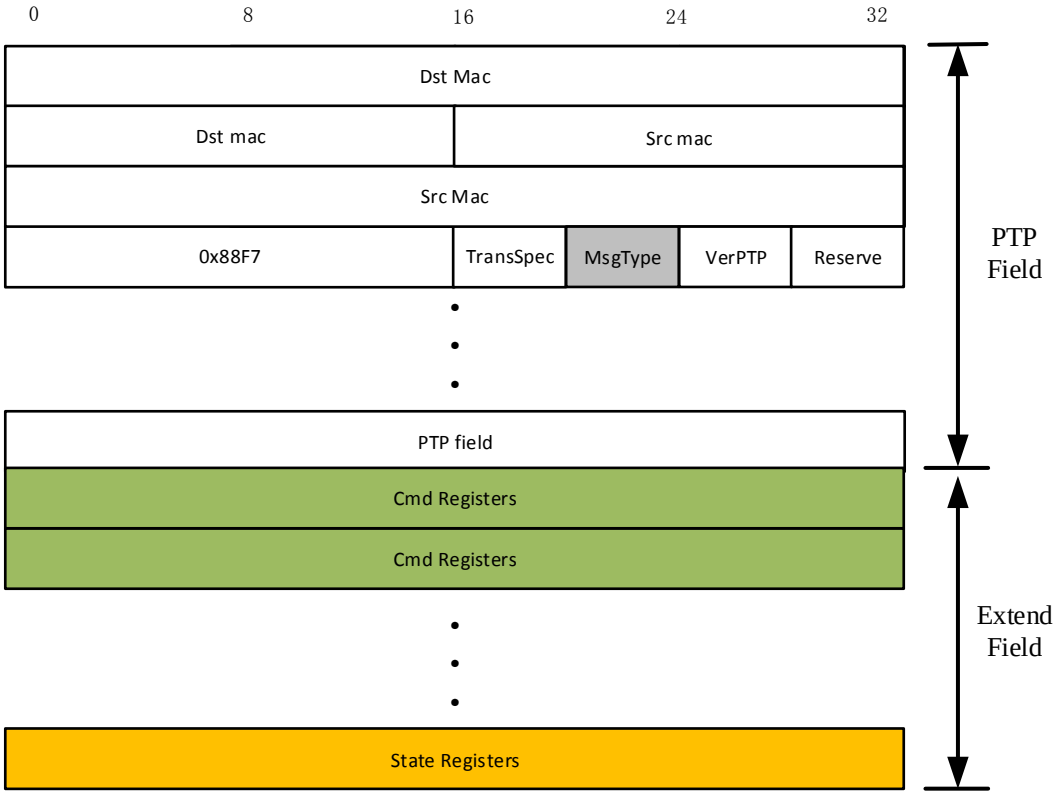


图 6-2 Beacon 报文格式定义

表 6-1 Beacon Report/Update 消息域划分表

模块	拍数	位置	字段	含义
LCM	5 (读写)	[127:80]	direct_mac_addr	TSN 节点直连设备 MAC 地址。
		[79]	direction	设备口接收报文的转发方向，0 为 0 口转发，1 为 1 口转发.
		[63:32]	token_bucket_rate	令牌桶速率，控制 P3-P5 优先级流量整形
		[31:0]	time_slot_period	时间槽大小、时间片翻转周期。
ESW	6 (只读)	[127:64]	esw_pktin_cnt	进入 ESW 模块的分组计数器。
		[63:0]	esw_pktout_cnt	ESW 模块输出的分组计数器。
	7 (只读)	[127:120]	local_mac_addr	TSN 本地 MAC 地址。
		[119:112]	bufm_id_cnt	Bufm 中所使用的 ID 计数器
	8 (只读)	[127:64]	eos_mdin_cnt	进入 EOS 模块的元数据计数器
		[63:0]	eos_mdout_cnt	EOS 模块输出的元数据计数器
	9	[127:120]	eos_q0_used_cnt	EOS 模块 Q0 队列已使用长度计

EOS	(只读)			数器
		[119:112]	eos_q1_used_cnt	EOS 模块 Q1 队列已使用长度计数器
		[111:104]	eos_q2_used_cnt	EOS 模块 Q2 队列已使用长度计数器
		[103:96]	eos_q3_used_cnt	EOS 模块 Q3 队列已使用长度计数器
GOE	10 (只读)	[127:64]	goe_pktin_cnt	进入 GOE 模块的分组计数器
		[63:0]	goe_port0out_cnt	GOE 模块往 0 口输出的分组计数器
	11 (只读)	[127:64]	goe_port1out_cnt	GOE 模块往 1 口输出的分组计数器
		[63:0]	goe_discard_cnt	GOE 模块丢弃的分组计数器

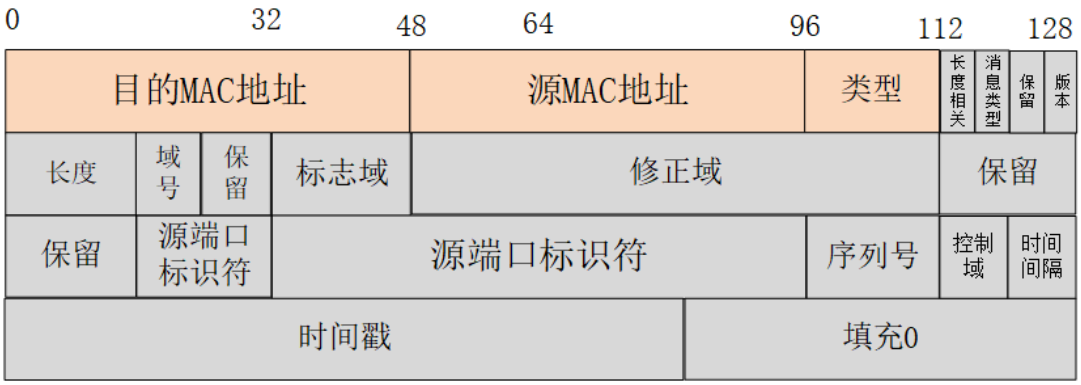


图 6-2 PTP 同步报文格式

[注：类型：16'h88F7；

消息类型：sync 为4'd1，delay_req 为4'd3，delay_resq 为4'd4，delay_test 为4'd5；

长度为：16'd64 字节；

修正域：透明时钟，起始时，该域为0；

时间戳：为时间戳（其他无需关系的PTP 字段填充0）]

附录二：OpenTSN metadata 定义

为了基于 FAST 编程框架支持 TSN 相关功能，OpenTSN 在 FAST 所定义的 metadata 基础上增加了 TSN metadata, 用于在 OpenTSN 中以较小的存储开销实现 TSN 对不同优先级报文的转发调度（如循环队列转发等）。我们定义该 metadata 为 TSN metadata (TSN-MD)。

1、FAST metadata 定义

详见《FAST 2.0 开源项目白皮书》

表 7-1 FAST metadata 定义表

字段	名称	备注
FAST_MD[127]	分组输入类型, pkttype	0: 数据报文，1: 控制报文
FAST_MD[126]	分组目的, pktdst	0: 网络接口输出，1: 送 CPU
FAST_MD[125:120]	分组输入端口号, inport	
FAST_MD[119:118]	输出类型，outtype	00:单播 01:组播 10:泛洪 11:从输入接口输出
FAST_MD[117:112]	输出，output	单播：分组输出端口 ID 组播/泛洪：表地址索引
FAST_MD[111:109]	优先级，priority	分组优先级
FAST_MD[108]	丢弃位，discard	
FAST_MD[107:96]	分组长度，len	包含 metadata 字段的分组长度（按字节计算）
FAST_MD[95:88]	上次处理模块号, smid	最近一次处理分组的模块 ID
FAST_MD[87:80]	目的模块号, dmid	下一次处理分组的模块 ID
FAST_MD[79:72]	pst	标准协议类型
FAST_MD[71:64]	序列号, seq	分组接收序列号
FAST_MD[63:50]	flowid	流 ID
FAST_MD[49]	分组的来源, pktsrc	0: 网络接口输入，1: CPU 输入
FAST_MD[48:40]	reserve	保留
FAST_MD[39:32]	分组在 bufm 缓存的 ID	
FAST_MD[31:0]	32 位时间戳, ts	时间戳

2、TSN metadata 定义

TSN metadata 定义如表 7-2 所示。

表 7-2 TSN metadata 定义表

字段	名称	备注
TSN_MD[23:21]	分组协议类型, pkt_type	0: best effort, 1: 预约带宽, 2: PTP,

		3: TSN
TSN_MD[20:9]	分组长度, pkt_len	按字节数计算
TSN_MD[8]	输出端口, outport	0: 0 号端口输出， 1: 1 号端口输出
TSN_MD[7:0]	分组存储 ID, bufm_ID	对应 bufM 模块地址编号

附录三：版本管理

文档版本号	修改人	修改时间	备注
1.0	杨翔瑞、陈波、彭锦涛	2019.05.12	根据 0509 概要设计的第一版详细设计
1.1	杨翔瑞、陈波	2019.05.14	根据 0514 讨论后文档进行修订，增加附录，对 LCM 进行重构，使其符合 FAST 流水线架构，并对应修改子模块划分和状态机设计。 将 ESW 模块进一步划分为三个子模块；将发往网口的分组都改为发往 GOE 以供 GOE 对分组进行计数。
1.2	彭锦涛	2019.05.15	将 EOS 调度解耦为 GC 模块和 TS 模块；增加 GC 模块详细设计； QS 和 TS 的功能不用状态机实现；画出整体框架图的所有接口信号并说明；画 MB、GC、TS 模块的流程图。
1.3	陈波、彭锦涛	2019.06.19	调试完成后，根据调试时作出的改动相应地调整文档。
1.4	陈波	2019.07.12	修改文档格式