

TSN 交换机（TSNSwitch3.1）设计方案

（版本 1.0）

OpenTSN

OpenTSN 开源项目组

2021 年 10 月

版本历史

版本	修订时间	修订内容	文件标识
1.0	2021.10.17	初版编制	OpenTSN3.1

目录

1. 项目概述.....	4
1.1. 设计目标	4
1.2. 设计指标	4
2. 总体设计.....	5
2.1. TSS 总体设计	6
2.1.1. TSS 内部功能划分	6
2.1.2. TSS 的处理流程	10
2.2. HCP 总体设计	12
2.2.1. HCP 内部功能划分	12
2.2.2. HCP 的处理流程.....	15
附录 A.数据格式定义.....	19
A.1. TSN 标签格式.....	19
A.2. 报文描述符格式	20
附录 B.资源统计.....	21
B.1. TSNSwitch3.1 资源统计	21
B.2. 时间敏感交换 TSS 资源统计	21
B.3. 硬件控制点 HCP 资源统计	22
附录 C.RAM 使用情况统计	23
附录 D.内部表项格式定义	24
附录 E.内部寄存器定义.....	25
附录 F.NMAC 报文格式	30
附录 G.command/command_ack 命令格式	33
附录 H.PTP 协议格式.....	33
附录 I.TSMP 消息协议格式.....	34

1. 项目概述

本文档是介绍 OpenTSN（版本 3.1）开源项目中的 TSN 交换机设计，主要分为项目概述、总体设计、TSS 详细设计以及 HCP 详细设计四大部分。

1.1. 设计目标

TSN 对传统以太网在时间同步、延迟确定性、可靠性传输和管理控制等方面进行增强，其应用场景已经由最初的工业互联网扩充至运营商网络、车载网络和航空航天器网络等。随着不同领域网络应用的丰富和扩展，应用场景已呈现多样化和差异化的特点。为了满足上述场景多样化与差异化的应用需求，OpenTSN（版本 3.1）开源项目设计了 TSN 交换机。TSN 交换机通过提取 TSN 标准中合适的子集进行设计，旨在设计一套能够满足不同领域对 TSN 网络的多样化及差异化需求的架构。

1.2. 设计指标

- 支持 IEEE 802.1AS、802.1Qch、802.1Qbv、802.1Qcc 标准
- 同时支持时间敏感、带宽预约和尽力转发三种流量的转发交换
- 支持 9 个千兆以太网接口（8 个网络接口，1 个控制接口）
- 硬件调度时间槽设置范围为[4us, 512us]
- 时间敏感流量最大支持 1024 个硬件调度时间槽的延迟
- 交换容量 16Gbps

- 交换延迟小于 30us
- 支持 16K 条流的转发配置
- 集中式报文交换缓存管理

2. 总体设计

TSNSwitch3.1 的总体设计如图 2-1 所示，TSNSwitch3.1 包括时间敏感交换 TSS 和硬件控制点 HCP 两部分逻辑，将数据处理逻辑和控制逻辑进行解耦。

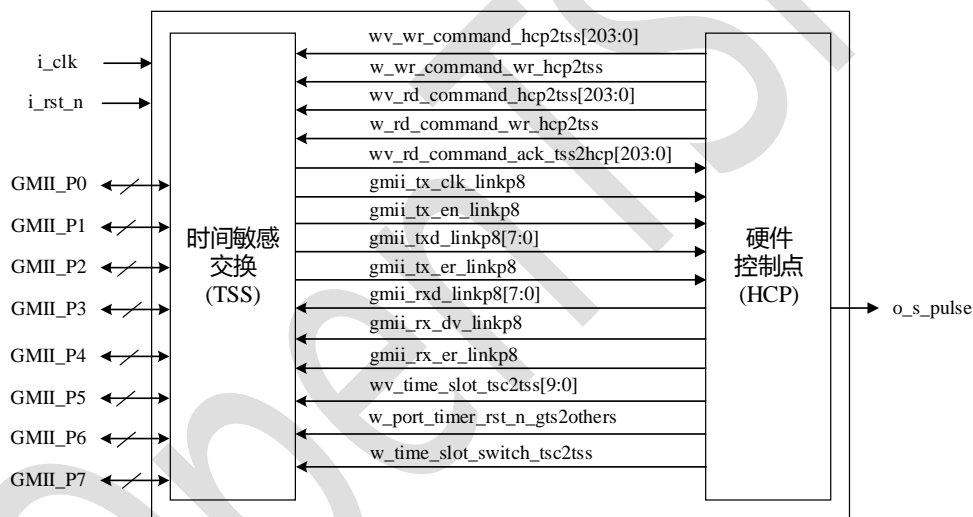


图2-1. TSNSwitch3.1 总体架构

TSS(Timing Sensitive Switch, 时间敏感交换)模块： 主要功能是对以太网报文进行交换；基于 IEEE 802.1Qch/IEEE 802.1 Qbv 调度模型对流量进行整形；以及计算 PTP 报文在 TSS 中传输的时间，即透明时钟，并将透明时钟累加到 PTP 报文透明时钟域中。

HCP(Hardware Control Point, 硬件控制点)模块： 主要功能是对 PTP 报文/状态报文进行封装、对配置封装报文/PTP 封装报文进行

解封装；对配置报文进行解析，并生成写命令，收集 TSN 交换机状态并周期性进行上报；在 PTP 报文中记录时间戳，计算 PTP 报文在 HCP 中传输的时间，即透明时钟，并将透明时钟累加到 PTP 报文透明时钟域中。

2.1. TSS 总体设计

2.1.1. TSS 内部功能划分

时间敏感交换 TSS 的内部功能划分如图 2-2 所示，下面对模块的功能进行描述。

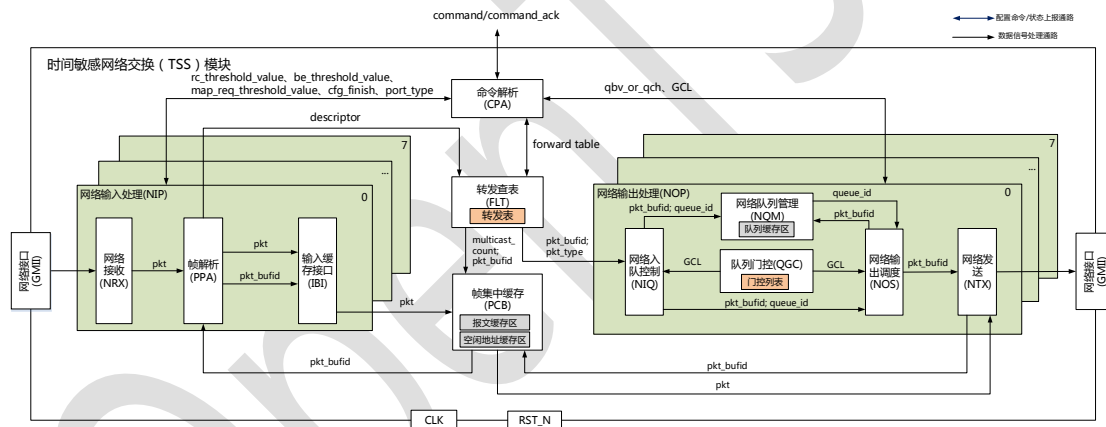


图2-2. 交换逻辑 TSS 内部功能划分

NRX(Network RX, 网络接收)模块：主要功能是接收网络接口传来的报文，将报文传输时钟域从 GMII 接口时钟域切换到架构内部时钟域，以及记录架构接收时间同步报文的时间，并存放在 TSNTag 中。根据交换处理逻辑所处的阶段来决定是否传输报文：若交换处理逻辑处于初始化阶段，丢弃所有报文；若交换处理逻辑处于配置阶段，只传输配置报文；若交换处理逻辑处于时钟同步阶段，传输非时间敏感报文；若交换处理逻辑处于正常工作阶段，接收所有报文。

FPA(Frame Parse, 帧解析)模块: 主要功能是将报文每拍数据位宽由 9bit 转换为 134bit, 构造报文描述符, 对报文进行监管。在构造报文描述符时, 需要根据接口类型来判断该报文是否经过映射, 若接口为非合作类型, 则该报文未经映射, 构造报文描述符使其从主机接口输出, 若接口为合作类型, 则该报文已经映射, 从 TSNtag 中提取相关信息构造报文描述符。报文监管是根据 bufid 剩余数量与报文阈值的大小关系来实现的: 若 bufid 剩余数量少于 RC 报文阈值时, 将 RC 报文和 BE 报文均丢弃, 若 bufid 剩余数量少于 BE 报文阈值时, 将 BE 报文丢弃, 若 bufid 剩余数量少于未经映射报文阈值时, 将未经映射报文丢弃。

IBI(Input Buffer Interface, 输入缓存接口)模块: 主要功能是将报文数据发送给报文集中缓存模块进行缓存。本模块接收帧解析模块传来的 134bit 数据, 并使用两个寄存器进行缓存, 其中任何一个寄存器有数据则往报文集中缓存模块发出写请求, 在接收到报文集中缓存模块传来的响应后, 才完成一拍数据的写入; 并将帧解析模块传来的数据写入另一个寄存器中。

FLT(Forward Lookup Table, 转发查表)模块: 主要功能是根据接收到的报文描述符提取出 flow_id, 根据 flow_id 进行查找转发表, 根据查表得到的输出端口进行 pkt_bufid 和 pkt_type 的转发。当描述符中查表使能信号 (lookup_en) 为低 (代表本描述符不需要查表) 时, 直接将描述符中的输出端口 (outport) 字段作为输出端口号进行转发。当得到的输出端口多位为高时, 意味处理的描述符对应报文是

个多播报文，需要将 pkt_bufid 和 pkt_type 往多个端口进行转发，同时将 pkt_bufid 与输出端口的数量发送给报文集中缓存模块进行缓存以便进行组播计数。

PCB(Pkt Centralize Buffer, 报文集中缓存)模块：主要功能是对架构需要转发的所有报文进行集中缓存,对空闲地址缓存区 pkt_bufid 进行缓存管理。每个 pkt_bufid 都需要设计一个计数器来对输出端口数量进行计数，pkt_bufid 每释放一次，将计数器减一；当 pkt_bufid 使用完进行释放的时候需要检测该 pkt_bufid 对应计数器的值，只有当计数器值为 0 才能进行将 pkt_bufid 写入空闲地址缓存区中，当不为 0 意味着该 pkt_bufid 存着一个组播报文并且该报文还未从所有需要输出的端口输出。报文缓存区将 1024KB 的空间划分成 512 个报文缓存块，每个报文缓存块能缓存一个 2KB 的报文。

CPA(Command Parse, 命令解析)模块：负责将接收到 HCP 的 command 命令进行解析，来实现对本地寄存器、转发表、门控列表的配置；根据接收到的读命令，将读数据封装在响应中输出给 HCP 模块。

NIQ(Network Input Queue, 网络入队控制)模块：主要功能是将 pkt_bufid 写入到网络队列管理模块中进行缓存。本模块需要根据接收到的报文类型信息、队列门控模块传来的门控信息进行 queue_id 的映射,并将 pkt_bufid 与 queue_id 发送给网络队列管理模块进行缓存。同时将 pkt_bufid 与 queue_id 发送给网络输出调度模块，以便对队列首地址进行管理。本模块还需要根据写入队列的信息与网络输出调度

模块传输的调度队列的信号来对队列的状态进行管理，主要是对所有队列中目前所写入的 `bufid` 数量进行管理。

NQM (Network Queue Manage, 网络队列管理) 模块: 主要功能是对网络输出端口的队列集中缓存管理

QGC(Queue Gate Control, 队列门控)模块: 主要功能是根据架构的全局时钟进行门控列表的读取, 并将门控列表中 8 个队列的门控开关信息发送给网络入队控制模块和网络输出调度模块。

NOS(Network Output Schedule, 网络输出调度)模块: 主要功能是根据调度后的队列信息从网络队列管理模块提取出 `pkt_bufid`。本模块需要根据当前队列信息与队列门控模块传来的门控信息进行计算, 得出一个最优先调度的队列, 并从网络队列管理模块的对应队列中得到 `pkt_bufid`。

NTX(Network TX, 网络发送)模块: 主要功能是从报文缓存区中读取报文并释放 `pkt_bufid`、计算时间同步报文的透明时钟并累加到透明时钟域、将报文传输时钟域从架构内部处理时钟域切换到 GMII 发送时钟域、构造帧前导符和帧开始符并将报文从网络接口输出。读取报文时, 需要先将 `pkt_bufid` 映射成报文在集中缓存区中缓存的基地址, 并根据此地址往报文集中缓存模块读取报文, 将读出报文的位宽由 134bit 转换为 8bit, 在接收到最后一拍数据时, 将此 `pkt_bufid` 传输给报文集中缓存模块进行释放, 以便后续进入架构的报文使用。

2.1.2. TSS 的处理流程

2.1.2.1. 网络口进网络口出的处理流程

每个网口配置的端口类型皆为合作类型，所有从网口进入的报文皆为映射后的报文，且都是通过查表去确认输出端口，报文的详细处理流程如下：

- 1) 报文由网口（GMII_RX）进入网络接收模块，完成 GMII 时钟域到架构内部处理时钟域的转换，并记录报文接收时间戳；然后根据交换处理逻辑所处的阶段来决定是否传输报文：若交换处理逻辑处于初始化阶段，丢弃所有报文；若交换处理逻辑处于配置阶段，只传输配置报文；若交换处理逻辑处于时钟同步阶段，传输非时间敏感报文；若交换处理逻辑处于正常工作阶段，传输所有报文。将报文往帧解析模块发送；
- 2) 帧解析模块收到报文后，对报文进行解析，并提取报文的特征信息与报文集中缓存模块分配的 pkt_bufid 构造成描述符，然后根据 bufid 剩余数量与报文阈值的大小关系来对报文进行监管：若 bufid 剩余数量少于 RC 报文阈值时，将 RC 报文和 BE 报文均丢弃，若 bufid 剩余数量少于 BE 报文阈值时，将 BE 报文丢弃。将报文描述符发送到转发查表模块；同时帧解析模块将 8bit 的数据写入到 134bit 的寄存器中，当一个寄存器写满了 134bit 便将寄存器中的数据传输到输入缓存接口模块，完成报文的位宽转换；

- 3) 输入缓存接口模块接收 134bit 的分组数据，因报文集中缓存区是采用分时复用的方式进行数据的写入，因此输入缓存接口模块需要等到报文集中缓存模块返回一个确认信号之后才能进行下一次的数据写入。当其中一个寄存器写满，且还未收到报文集中缓存模块返回的确认信号时，后续传来的数据就写入到另一个寄存器，每个寄存器写满再发送给报文集中缓存模块进行报文的缓存，以此轮询，直到报文全部写入完成；
- 4) 转发查表模块会根据报文描述符的 `lookup_en` 信号判断该报文是否需要查表转发，若不需要查表，则直接将报文描述符中的 `outport` 信息作为转发的输出端口，将 `pkt_bufid` 与 `pkt_type` 信息发送给对应输出端口逻辑；若需要查表，则从报文描述符中提取出 `flow_id` 进行查找转发表，得到输出端口号，并将 `pkt_bufid` 与 `pkt_type` 信息发送给对应输出端口逻辑；
- 5) 输出端口逻辑中的网络入队控制模块接收到 `pkt_bufid` 与 `pkt_type` 后，根据队列门控模块的门控信息与 `pkt_type` 进行缓存队列的选择，并将选择好的 `queue_id` 与 `pkt_bufid` 一起发送给网络队列管理模块进行缓存。
- 6) 网络输出调度模块根据队列门控模块的门控信息以及每个队列的状态信息计算出最优先调度的 `queue_id`，并将此 `queue_id` 中的第一个 `pkt_bufid` 缓存的地址发送给网络队列管理模块，等待网络队列管理模块将对应的队列中的数据读取出来，将第一个 `pkt_bufid` 的地址作为 `pkt_bufid` 发送给网络发送模块，并

根据读取出来的数据更新此 queue_id 中第一个 pkt_bufid 所缓存的地址。

- 7) 网络输出调度模块接收到 pkt_bufid 之后发送给网络发送模块，网络发送模块根据 pkt_bufid 往报文集中缓存模块进行报文的提取。网络发送模块内部维持两个寄存器，依次将 134bit 的数据转换成 8bit 的数据往后发送，跨时钟域处理后往接口（GMII_TX）输出；当一个寄存器读空之后再往报文集中缓存模块进行下一个 134bit 数据的读取，直到报文数据全部读取完成。

2.2. HCP 总体设计

2.2.1. HCP 内部功能划分

硬件控制点 HCP 的内部功能划分如图 2-3 所示，下面对模块的功能进行描述。

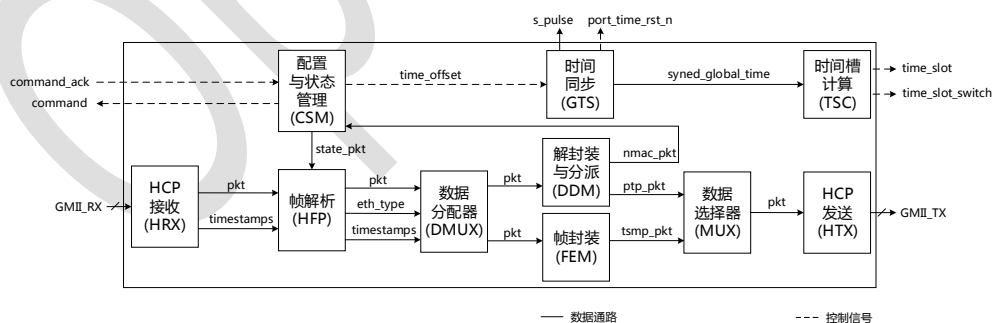


图2-3. 硬件控制点 HCP 内部功能划分

HRX (HCP RX, hcp 接收) 模块：主要功能是接收控制接口传来的报文，将报文传输时钟域从 GMII 接收时钟域切换到架构内部时钟域，以及记录控制接口接收时间同步报文的时间，并存放在 TSNTag 中。根据硬件控制点逻辑所处的阶段来决定是否传输报文：若硬件控

制点逻辑处于初始化阶段，丢弃所有报文；若硬件控制点逻辑处于配置阶段，只传输配置报文；若硬件控制点逻辑处于时钟同步阶段，传输非时间敏感报文；若硬件控制点逻辑处于正常工作阶段，接收所有报文。

HFP(Hcp Frame Parse, 帧解析)模块：主要功能是从报文中提取以太网类型，将以太网类型和接收时间戳组成报文描述符；对控制接口传来的报文和状态上报报文进行选择，将控制接口传来的报文及其描述符先缓存在队列中，在硬件控制点无上报请求时，将队列中的报文及其描述符读出并输出给下一级模块，若硬件控制点有上报请求且控制接口传来的报文未进行传输时，将状态上报报文输出给下一级模块。

DMUX (DeMultipleXer, 数据分配器) 模块：主要功能是根据以太网类型对报文进行分派，并根据硬件控制点所处的阶段决定是否丢弃报文。本模块在初始化阶段将所有报文丢弃，在配置阶段/时钟同步阶段/正常工作阶段将接收到的 TSMP 报文（芯片配置帧、PTP 封装帧、ARP 封装帧）及其描述符分派给解封装与分派模块，在时钟同步阶段/正常工作阶段将接收到的非 TSMP 报文（标准以太网非 IP 报文、PTP 报文、NMAC 状态上报报文）分派给帧封装模块进行封装处理。

DDM (Decapsulation_Dispatch_Module, 解封装与分派) 模块：将接收到的芯片配置帧、PTP 封装帧、ARP 封装帧分别解封装成 NMAC 配置帧、PTP 帧、ARP 响应帧，并将 NMAC 配置帧分派至配置与状态管理模块，记录 PTP 帧发送的时间戳，将其和 ARP 响应帧

分派至数据选择器模块。

FEM (Frame_Encapsulation_Module, 帧封装) 模块: 主要功能是将接收到的 ARP 请求帧、PTP 帧、NMAC 状态上报帧封装到 TSMP 报文中, 针对 PTP 帧, 会记录其接收时间戳、计算透明时钟并累加到透明时钟域。

MUX (MUltipleXer, 数据选择器) 模块: 主要功能是对封装后的帧和解封装后的帧进行选择, 并输出到控制发送模块。

CSM (Configuration and State Manage, 配置与状态管理) 模块: 主要功是对接收到的 NMAC 报文进行解析, 生成写命令; 根据接收到的报文接收/发送/丢弃脉冲进行计数; 以及周期性上报状态。

GTS (Global_Time_Sync, 全局时钟同步) 模块: 主功能是维护一个 48bit 的全局时钟并根据接收到的本地时钟与主时钟的偏移值 offset 对全局时钟进行修正; 维护一个局部时钟, 将局部时钟传给其它模块, 用于记录时间同步报文在硬件控制点的接收/发送时钟, 进而计算时间同步报文在硬件控制点中传输的透明时钟; 维护一个上报周期计数器, 当每经过一个上报周期输出一个上报脉冲信号。

TSC (Time_Slot_Calculation, 时间槽计算) 模块: 根据全局时间、时间槽长度和周期, 计算当前时刻处于一个周期内的哪个时间槽。

HTX (HCP TX, hcp 发送) 模块: 主要功能是计算时间同步报文的透明时钟并更新透明时钟域、将报文传输时钟域从内部处理时钟域切换到 GMII 发送的时钟域、增加帧前导符和帧开始符后将报文从控制接口输出。

2.2.2. HCP 的处理流程

硬件控制点逻辑的处理流程包括报文的封装与解封装、配置报文的解析、状态报文的上报。

2.2.2.1. 配置报文解析

HCP 会对芯片配置报文进行解析，实现对 HCP 和 TSE 或 HCP 和 TSS 中寄存器、表的配置。

控制接收模块接收到芯片配置报文后，将报文每拍数据的位宽由 8bit 转换为 9bit 后，对报文进行跨时钟域（从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域）转换，并根据 HCP 所处的阶段来对报文进行丢弃或传输：若 HCP 处于初始化阶段/配置阶段，丢弃芯片配置报文；若 HCP 处于时钟同步阶段/正常工作阶段，传输 TSMP 报文。

帧解析模块从芯片配置报文中提取以太网类型，将以太网类型和接收时间组成报文描述符，现将报文及其描述符写到 fifo 中，在 HCP 无上报请求时，将 fifo 中的报文和描述符读出并输出给数据分配器。

数据分配器根据报文的以太网类型为 0xff01 判定该报文为 TSMP 报文（芯片配置报文属于 TSMP 报文中的一种类型），在 HCP 处于初始化阶段时，将芯片配置报文丢弃，在 HCP 处于配置阶段/时钟同步阶段/正常工作阶段，将芯片配置报文输出给解封装与分派模块。

在解封装与分派模块对芯片配置报文进行解封装与分派。将芯片配置帧解封装为 NMAC 配置帧，并分派给配置与状态管理模块进行解析。

配置与状态管理模块解析 NMAC 报文，从 NMAC 报文提取该报文中携带的配置数据个数，并根据配置的基地址，将每个配置数据封装都封装到写命令中并输出给端处理逻辑 TSE 或交换处理逻辑 TSS。

2.2.2.2. 状态报文上报

HCP 会对端处理逻辑 TSE 和硬件控制点逻辑 HCP 的状态，或交换处理逻辑 TSS 和硬件控制点逻辑 HCP 的状态进行周期性上报。

本地时钟每经过一个上报周期，时钟同步模块会输出一个上报脉冲给配置与状态管理模块，配置与状态管理模块检测到上报脉冲后，向帧解析模块发送上报请求，在接收到帧解析模块传来的响应信号后，会根据上报类型，向端处理逻辑 TSE 或交换处理逻辑 TSS 发送读命令，读命令中包含需读取数据的地址；当接收到端处理逻辑 TSE 或交换处理逻辑 TSS 发来的读响应后，从读响应中解析出读数据并将其作为状态报文中的数据输出给帧解析模块。

帧解析模块在监测到配置与状态管理模块传来的上报请求后，当本模块无报文输出时，发送响应给配置与状态管理模块，然后开始将配置与状态管理模块传来的状态报文输出给数据分配器。

数据分配器根据报文的以太网类型为 0x1662 判定该报文为状态上报报文，在 HCP 处于时钟同步阶段/正常工作阶段，将状态上报报文输出给帧封装模块。

帧封装模块将状态上报报文封装到 TSMP 报文中并输出给数据选择器。

数据选择器对解封装后的报文和封装后的报文进行选择，在封装后的报文有传输请求时，将该路报文输出给控制发送模块。

控制发送模块将报文传输时钟域从内部处理时钟域切换到 GMII 发送的时钟域、增加帧前导符和帧开始符后将报文从控制接口输出。

2.2.2.3. PTP 封装报文和 ARP 封装报文解封装

HCP 会对 PTP 封装报文和 ARP 封装报文进行解封装。

控制接收模块接收到 PTP 封装报文、ARP 封装报文后，将报文每拍数据的位宽由 8bit 转换为 9bit 后，对报文进行跨时钟域（从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域）转换，并记录控制接口接收报文的时间，根据 HCP 所处的阶段来对帧进行丢弃或传输：若 HCP 处于初始化阶段/配置阶段，丢弃 PTP 封装报文、ARP 封装报文；若 HCP 处于时钟同步阶段/正常工作阶段，传输 PTP 封装报文、ARP 封装报文。

帧解析模块从 PTP 封装报文、ARP 封装报文中提取以太网类型，将以太网类型和接收时间组成报文描述符，先将报文及其描述符写到 fifo 中，在 HCP 无上报请求时，将 fifo 中的报文和描述符读出并输出给数据分配器。

数据分配器根据报文的以太网类型为 0xff01 判定该报文为 TSMP 报文（PTP 封装报文、ARP 封装报文属于 TSMP 报文），在 HCP 处于初始化阶段时，将 PTP 封装报文、ARP 封装报文丢弃，在 HCP 处于配置阶段/时钟同步阶段/正常工作阶段，将 PTP 封装报文、ARP 封

装报文输出给解封装与分派模块。

在解封装与分派模块对 PTP 封装报文、ARP 封装报文进行解封装与分派。将 PTP 封装报文、ARP 封装报文分别解封装为 PTP 报文、ARP 响应报文，并分派给数据选择器；针对 PTP 封装报文，会记录 PTP 报文发送时间戳。

数据选择器对解封装后的报文和封装后的报文进行选择，在解封装后的报文有传输请求时，将该路报文输出给控制发送模块。

控制发送模块将报文传输时钟域从内部处理时钟域切换到 GMII 发送的时钟域、增加帧前导符和帧开始符后将报文从控制接口输出，其中针对 PTP 报文，会计算 PTP 报文的透明时钟并更新透明时钟域。

2.2.2.4. ARP 请求报文和 PTP 报文报文封装

HCP 会对 ARP 请求报文和 PTP 报文进行封装，下面对控制接口接收的 ARP 请求报文、PTP 报文的封装处理流程进行介绍。

控制接收模块接收到 ARP 请求报文和 PTP 报文后，将报文每拍数据的位宽由 8bit 转换为 9bit 后，对报文进行跨时钟域（从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域）转换，并记录控制接口接收报文的时间，根据 HCP 所处的阶段来对帧进行丢弃或传输：若 HCP 处于初始化阶段/配置阶段，丢弃 ARP 请求报文和 PTP 报文；若 HCP 处于时钟同步阶段/正常工作阶段，传输 ARP 请求报文和 PTP 报文。

帧解析模块从 ARP 请求报文和 PTP 报文中提取以太网类型，将以太网类型和接收时间组成报文描述符，先将报文及其描述符写到

fifo 中，在 HCP 无上报请求时，将 fifo 中的报文和描述符读出并输出给数据分配器。

数据分配器根据报文的以太网类型为 0x0806、0x98f7 判定该报文为 ARP 报文、PTP 报文，在 HCP 处于初始化阶段/配置阶段时，将 ARP 报文、PTP 报文丢弃，在 HCP 处于时钟同步阶段/正常工作阶段，将 ARP 报文、PTP 报文输出给帧封装模块。

帧封装模块将状态上报报文封装到 TSMP 报文中并输出给数据选择器，其中针对 PTP 报文，会计算 PTP 报文在 HCP 中传输的透明时钟并更新透明时钟域。

数据选择器对解封装后的报文和封装后的报文进行选择，在封装后的报文有传输请求时，将该路报文输出给控制发送模块。

控制发送模块将报文传输时钟域从内部处理时钟域切换到 GMII 发送的时钟域、增加帧前导符和帧开始符后将报文从控制接口输出。

附录 A.数据格式定义

A.1. TSN 标签格式

表 A-1 时间同步报文的 TSNtag

字段	位宽 bit	位置	描述
pkt type	3	[47:45]	报文类型，其中 3'b100 表示时间同步报文（其他报文类型见表 A-2）
Flow id/IMAC	14	[44:31]	流标识或者内部 MAC 地址，其中静态流量使用 flowID，每条静态流有唯一一个 flowID，动态流使用 imac
Reserve	12	[30:19]	保留
Rx_timestamps	19	[18:0]	接口接收到时间同步报文的本地时间，用于计算 PTP 报文在设备中的透

字段	位宽 bit	位置	描述
			明时钟

表 A-2 非时间同步报文的 TSNTag

字段	位宽 bit	位置	描述
pkt type	3	[47:45]	报文类型，其中 3'b000 表示 ST 报文，3'b001 表示 ST 报文，3'b010 表示 ST 报文 3'b011 表示 RC 报文，3'b101 表示 NMAC 分组，3'b110 表示 BE 分组，3'b111：需重组的分组
Flow id/IMAC	14	[44:31]	流标识或者内部 MAC 地址，其中静态流量使用 flowID，每条静态流有唯一的一个 flowID，动态流使用 imac
Seq id	16	[30:15]	用于标识每条流中报文的序列号
Frag flag	1	[14]	分片标识位，该字段在报文分片情况下会被使用到，用于标识报文最后一个分片，其中 1'b0 表示报文非最后一个分片，1'b1 表示报文最后一个分片
Frag ID	4	[13:10]	分片编号，该字段在报文分片情况下会被使用到，用于为报文的每个分片进行编号
inject addr	5	[9:5]	ST 流在 TSN 网卡发送端的缓存地址
submit addr	5	[4:0]	ST 流在 TSN 网卡接收端的缓存地址

A.2. 报文描述符格式

TSNSwitch3.1 中报文描述符格式如下表 A-3 所示。

表 A-3 报文描述符格式

内容	位宽	位置	含义
inject_addr/ submit_addr	5	[45:41]	ST 流注入/提交时缓存的地址。
Reserve	1	[40]	保留。
inport	4	[39:36]	报文的输入端口，用于最终构造 metadata。
pkt_type	3	[35:33]	报文类型，用于入队控制时区分报文类型，选择队列。
flow ID/IMAC	14	[32:19]	流 ID，用于 FLT 模块查表时的地址索引。
lookup_en	1	[18]	查表使能，用于 FLT 判断该报文是否需要 进行查表操作。

outport	9	[17:9]	输出端口号,用于 FLT 模块区分输出端口。
pkt_bufid	9	[8:0]	报文在缓存区中缓存的 ID 号,用于标识每个报文。

附录 B.资源统计

B.1. TSNSwitch3.1 资源统计

在 quartus 中编译 8 接口的 TSNSwitch3.1 工程所得到的 TSNSwitch3.1 使用的资源如下表 B-1 所示。

表 B-1 TSNSwitch3.1 资源统计

模块		自适应逻辑模块 ALMS	寄存器 Register	块存储资源 Block memory bits	M20Ks
TSN 交换机	交换逻辑 TSS	11169.2	19463	9057360	596
	硬件控制点 HCP	2303	2673	2716	4
	总和	13472.2	22136	9060076	600

B.2. 时间敏感交换 TSS 资源统计

TSNSwitch3.1 工程中时间敏感交换 TSS 使用的资源如下表 B-2 所示。

表 B-2 时间敏感交换 TSS 资源统计

模块		自适应逻辑模块 ALMS	寄存器 Register	块存储资源 Block memory bits	M20Ks
命令解析模块 CPA	总和	273.8	396	0	0
转发查表模块 FLT	总和	426.8	1040	151232	14
网络输入	帧解析模块	144	318	0	0

处理 NIP	输入缓存接口模块	107.8	301	0	0
	网络接收模块	97.2	217	144	1
	总和	349	836	144	1
9*NIP	总和	3141	7524	1296	9
网络输出模块 NOP	网络入队模块	109.1	189	0	0
	网络输出调度模块	98.4	111	0	0
	网络队列管理	11.3	15	4608	1
	网络发送模块	425.3	635	128	1
	队列门控模块	9.1	26	8192	1
	总和	653.1	976	12928	3
9*NOP	总和	5877.9	8784	116352	27
集中缓存模块 PCB	总和	1449.7	1719	8788480	538
交换逻辑 TSS	总和	11169.2	19463	9057360	596

B.3. 硬件控制点 HCP 资源统计

TSNSwitch3.1 工程中硬件控制点 HCP 使用的资源如下表 B-3 所示。

表 B-3 HCP 资源统计

模块		自适应逻辑模块 ALMS	寄存器 Register	块存储资源 Block memory bits	M20Ks
硬件控制点 HCP	配置与状态管理模块	832	1338	0	0
	帧解封装模块	116	317	0	0
	帧封装模块	348	220	0	0
	数据选择模块	43	79	0	0
	HCP 帧	83	148	2444	2

模块		自适应逻辑 模块 ALMS	寄存器 Register	块存储资源 Block memory bits	M20Ks
	解析模块				
	HCP 发 送模块	310	214	128	1
	HCP 接 收模块	53	127	144	1
	全局时钟 同步模块	395	187	0	0
	数据分配 模块	5	12	0	0
	时间槽计 算模块	118	31	0	0
	总和	2303	2673	2716	4

附录 C.RAM 使用情况统计

表 C-1 TSNSwitch3.1 工程 RAM 使用情况

模块名	ram 类型	ram 位宽 (bit)	ram 深度	资源 (bit)
command_parser	伪双端口	44	4	176
forward_lookup_table	伪双端口 (用 以生成 fifo)	61	32	1952
	真双端口	57	32	1824
	真双端口	9	16384	147456
network_rx (*9)	伪双端口 (用 以生成 fifo)	9	16	144×9=1296
network_queue_manage (*9)	伪双端口	9	512	4608×8=41472
network_tx (*9)	伪双端口 (用 以生成 fifo)	8	16	128×9=1152
queue_gate_control (*9)	真双端口	8	1024	8192×9=73728
pkt_centralized_buffer	伪双端口 (用 以生成 fifo)	9	512	4608
	真双端口	4	512	2048

	真双端口	134	65536	8781824
hcp_frame_parse	伪双端口（用以生成 fifo）	9	256	2304
	伪双端口（用以生成 fifo）	35	4	140
network_rx_hcp	伪双端口（用以生成 fifo）	9	16	144
network_tx_hcp	伪双端口（用以生成 fifo）	8	16	128
合计	/	/	/	9060076

附录 D.内部表项格式定义

● 转发表

转发表的格式如表 D-1 所示。

表 D-1 转发表数据格式

名称	含义	备注
outport[8:0]	输出端口号(bitmap)，总共 8 个网络接口与 1 个主机接口。	使用 RAM 进行实现，深度为 16K

转发表设计在转发查表(FLT)模块，该模块会根据接收到的描述符提取出 flow_id，再将 flow_id 作为查表地址进行查找转发表。查表结果为输出端口号，其中某位为高则代表往某个对应的输出端口进行转发，本模块根据查表结果将描述符内容转发给对应的输出端口。转发表的深度为 16K，代表由 TSNSwitch 组成的整个网络最多能够支持 16K 条流的转发。

● 门控表

门控表的格式如表 D-2 所示：

表 D-2 门控表数据格式

名称	含义	备注
gate_ctrl_vector[7:0]	门控向量，对应 8 个门控信息。	使用 RAM 进行实现，深度为 1024

门控表设计在交换发送处理逻辑中的队列门控(QGC)模块，该模块根据接收到的时间槽信息以及时间槽切换信号来进行查表。本模块每接收到一个时间槽切换信号，便将时间槽的 ID 作为查表地址进行查找门控表，查表结果为门控向量，对应 8 个门控信号，当某位为高则代表打开对应的某个队列的门控，网络输出调度(NOS)模块根据该门控进行对队列进行调度。门控表的深度为 1024，代表 TSNSwitch 最多能够支持 1024 个时间槽做为一个周期。

附录 E.内部寄存器定义

架构内部可配置地址空间主要有两部分，包括：MDID 模块号和真实地址空间，其中 MDID 模块号主要用来区分不同模块，而后 20 位为各个模块使用的地址空间。地址的第 19bit 位用于区别地址类型，控制/表项寄存器可读可写，调试和版本寄存器只读，每个模块的地址空间为 1024k,其中可读可写和只读寄存器各有 512k。具体地址含义如下。

表 E-1 地址格式

ADDR[26:0]		
MDID[26:20]	ADDR[19]	ADDR[18:0]
MDID : 0-127	0	该模块的控制寄存器,表项等,可读可写
	1	只读

每个处理模块的 MDID 号分配如下：

表 E-2 模块中的 MDID 和地址

处理模块	CSM	TIS	TSS	QGC	GTS	FLT
MDID	0x0	0x1	0x2	0x3-0xa	0xb	0xc
地址	0x0-0xffff	0x100000-0x1ffff	0x200000-0x2ffff	0x300000-0xafffff	0xb00000-0xbffff	0xc00000-0xcffff

● CSM 模块

地址范围为 Addr 0x0-0xffff。

表 E-3 CSM 模块寄存器

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
0x0	offset_l			
0x1	offset_h			
0x2	time_slot			
0x3	cfg_finish			
0x4	port_type			
0x5	qbv_or_ach			
0x6	report_type			
0x7	report_en			
0x8	inject_slot_period			
0x9	submit_slot_period			
0xa	report_period			
0xb	offset_period			
0xc	rc_regulation_value			
0xd	be_regulation_value			
0xe	unmap_regulation_value			
0xf ~ 0xffff	reserve			

表 E-4 寄存器的具体含义

name	bit	R/W	description	default
offset_l	31:17	R/W	代表时间偏移的高位值的低15位，表示毫秒	0
	16:0	R/W	时间偏移的低位，表示拍数	0
offset_h	31:17	R/W	保留位	0

name	bit	R/W	description	default
	16	R/W	代表时间偏移的正负值，1 代表正值，如果为 0，则代表负值	0
	15:0	R/W	代表时间偏移的高位值的高 16 位，表示毫秒	0
time_slot	31:11	R/W	保留	
	10:0	R/W	时间槽大小	0
cfg_finish	31:2	R/W	保留	0
	1:0	R/W	配置完成寄存器， 0 代表架构正在初始化，不接收任何报文， 1 代表初始化完成，可以接收 NMAC 配置报文 2 代表配置完成，可以接收除 ST 报文的任何报文 3 代表可以接收任何报文	0
port_type	31:8	R/W	保留	0
	7:0	R/W	网络端口类型寄存器，架构共有 8 个网络端口，寄存器的 0-7 位分别代表 0-7 端口的类型，1 代表非合作类型，处理标准以太网类型的报文，0 代表合作类型，处理 TSN 报文	0
qbv_or_ach	31:1	R/W	保留	0
	0	R/W	调度模式选择信号，网络输出逻辑中的调度机制是 QBV 模式还是 QCH 模式 0 代表 QBV 模式；1 代表 QCH 模式	0
report_type	31:16	R/W	保留	0
	15:0	R/W	上报类型，具体参考附录 D	0
report_en	31:1	R/W	保留	0
	0	R/W	上报使能信号，配置与状态管理模块是否进行周期性上报 0 代表不上报；1 代表上报	0
inject_slot_period	31:12	R/W	保留	0
	10:0	R/W	注入时间槽周期，架构内部时间槽切换的周期值 配置的值范围：1-1024 个	0
submit_slot_period	31:12	R/W	保留	0

name	bit	R/W	description	default
	10:0	R/W	提交时间槽周期，架构内部时间槽切换的周期值 配置的值范围：1-1024 个	0
report_period	31:12	R/W	保留	0
	11:0	R/W	上报周期，配置与状态管理模块上报的周期值 配置的值范围：1（ms）或 1000（ms）	0
offset_period	31:24	R/W	保留	0
	23:0	R/W	offset 补偿的配置周期	
rc_regulation_value	31:9	R/W	保留	0
	8:0	R/W	RC 流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃 RC 报文	
be_regulation_value	31:9	R/W	保留	0
	8:0	R/W	BE 流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃 BE 报文和 RC 报文	
unmap_regulation_value	31:9	R/W	保留	0
	8:0	R/W	非映射流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃非映射报文	
reserve	31:9	R/W	保留	0

● QGC 模块

地址范围为 Addr 0x300000-0xafffff，其中 0x300000-0x3fffff 表示第一个端口的门控表，以此类推，共有 8 个端口门控。

表 E-5 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
port0_gate_table_N 0x300000-0x3003ff	0 号端口的门控表，N=0、1、...、1023，输出门控 port0_gate_table_0 表示 0 号端口的第一个时刻的门控状态			
port1_gate_table_N 0x400000-0x4003ff	1 号端口的门控表，N=0、1、...、1023			
port2_gate_table_N 0x500000-0x5003ff	2 号端口的门控表，N=0、1、...、1023			

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
port3_gate_table_N 0x600000-0x6003ff	3 号端口的门控表, N=0、1、...、1023			
port4_gate_table_N 0x700000-0x7003ff	4 号端口的门控表, N=0、1、...、1023			
port5_gate_table_N 0x800000-0x8003ff	5 号端口的门控表, N=0、1、...、1023			
port6_gate_table_N 0x900000-0x9003ff	6 号端口的门控表, N=0、1、...、1023			
port7_gate_table_N 0xa00000-0xa003ff	7 号端口的门控表, N=0、1、...、1023			

表 E-6 寄存器的具体含义

name	bit	R/W	description	default
port0_gate_table_0	31:8	R/W	保留	0
	7:0	R/W	0-7 位分别代表 0-7 共 8 个队列的门控状态, 0 代表该队列的门控关闭, 1 代表开启	0
.....				
port7_gate_table_1023	31:8	R/W	保留	0
	7:0	R/W	0-7 位分别代表 0-7 共 8 个队列的门控状态, 0 代表该队列的门控关闭, 1 代表开启	0

● FLT 模块

地址范围为 Addr 0xc00000-0xcfffff。

表 E-7 地址格式

Addr	Data			
	[31:24]	[23:16]	[15:8]	[7:0]
0xc00000-0xc03fff	forward_table_N, 表示转发表, N=0,1,2, ...16383, forward_table_0 表示第 0 个转发表			
0xc04000-0xcfffff	保留			

表 E-8 寄存器的具体含义

name	bit	R/W	description	default
forward_table_0	31:16	R/W	保留	0
	8:0	R/W	转发表的内容, 使用 bitmap 的形式, 0-8 位分别代表向	0

name	bit	R/W	description	default
			0-8 号端口，每位值 0 代表不向该端口转发，1 代表向该端口转发	
.....				
forward_table_16383	31:16	R/W	保留	0
	8:0	R/W	转发表的内容，使用 bitmap 的形式，0-8 位分别代表向 0-8 号端口，每位值为 0 代表不向该端口转发，1 代表向该端口转发	0
0xc04000-0xcfffff			保留	

附录 F.NMAC 报文格式

● 配置报文格式

配置报文格式：在报文中用 count 字段（8bit）表示报文中包含的配置条目数，报文最小为 64 字节，最后不够 64 字节的报文需要补零。NMAC 命令在以太网报文中的封装如图 F-1 所示。

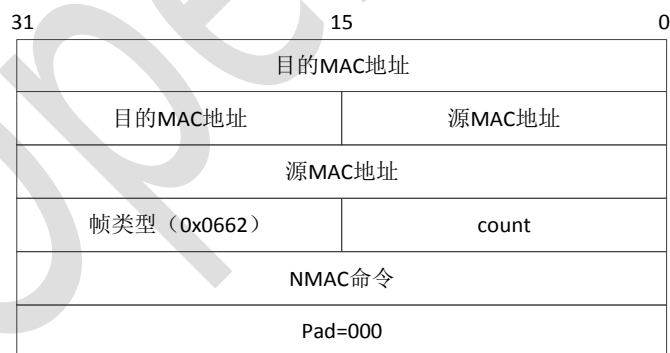


图 F-1 NMAC 配置报文格式

NMAC 命令的格式如图 F-2 所示。

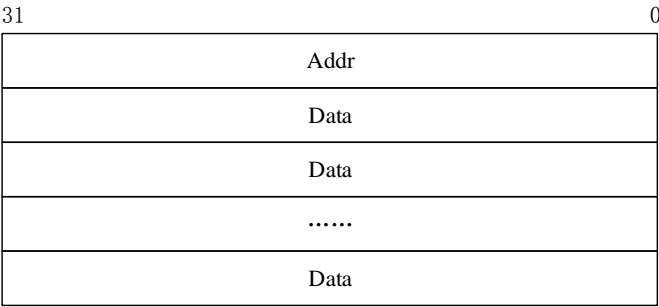


图 F-2 NMAC 命令格式

当配置的寄存器数量为 1，NMAC 命令就包括 32bit 的 ADDR 和 32bit 的 DATA；当配置的寄存器数量为 N（N>1），NMAC 命令就包括 32bitADDR 和 N*32bit 的 DATA, 第一个 DATA 以 ADDR 作为 RAM 写地址，第二个以及后续 DATA 以 ADDR 循环加 1 作为 RAM 写地址。

● 上报报文格式

上报报文格式：其中报文类型为 NMAC 报文（0x1662）。

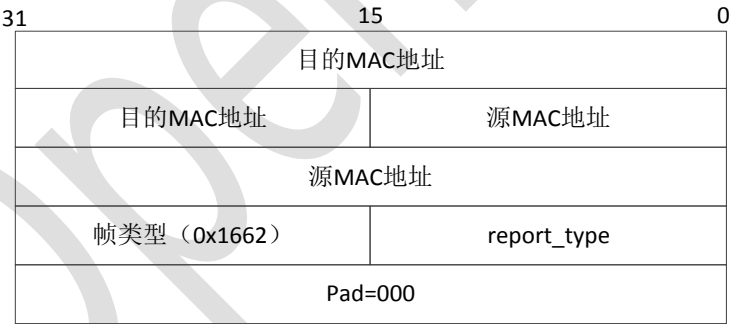


图 F-3 NMAC 上报报文格式

表 F-1 上报类型格式

上报类型（16bit）		含义
高 6bit	低 10bit	
000000 单个寄存器	0	配置的单个寄存器，包含配置完成寄存器、端口状态寄存器、时间槽大小寄存器、时间偏移寄存器、上报周期寄存器、上报类型寄存器、应用周期寄存器
000001	0	第 0-63 条转发表

上报类型（16bit）		含义
高 6bit	低 10bit	
转发表	1	第 64-127 条转发表
	2-255	第 128-16383 条转发表
000010 注入时刻表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
000011 提交时刻表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
000100-001 011 P0-P7 输出 门控表	0	第 0-63 条注入时刻表
	1	第 64-127 条注入时刻表
	2-15	第 128-1023 条注入时刻表
001100 xx_state	0	暂未开发

单个寄存器，report_type 高 6bit 为 0000，低 10bit 为 0。

转发表上报报文格式，report_type 高 6bit 为 000001，低 10bit 为上报的第几块，转发表一共有 16K 条，每条转发表占用 2 字节（其中低 9bit 为有效数据），因此每个报文可以携带 64 条，总共需要 256 个报文。

注入时刻表上报报文格式，report_type 高 6bit 为 000010，低 10bit 为上报的第几块，注入时刻表一共有 1024 条，每条转发表占用 2 字节（其中低 9bit 为有效数据），因此每个报文可以携带 64 条，总共需要 16 个报文。

提交时刻表上报报文格式，report_type 高 6bit 为 000011，低 10bit 为上报的第几块，注入时刻表一共有 1024 条，每条转发表占用 2 字节（其中低 9bit 为有效数据），因此每个报文可以携带 64 条，总共需要 16 个报文。

门控表按照端口划分，每个门控占用一块 RAM，共有 8 个端口。
每个端口两块 RAM，总共需要 16 块 RAM。

附录 G.command/command_ack 命令格式

表 G-1 command/command_ack 命令格式

位置	位宽	名称	说明
[203:180]	8	node_id	该字段用来标识对哪个节点进行读写。每个 TSE 或 TSS 都有一个唯一的节点 ID。该字段在 TSN 网卡+TSN 交换机模式下使用到。
[179:172]	8	dest_module_id	该字段用来标识对一个节点内的哪个模块进行控制。TSE 或 TSS 内部每个子模块都有一个唯一的模块 ID
[171:168]	4	type	4'b0001:寄存器或表项的写命令; 4'b0010:寄存器或表项的读命令; 4'b0110:寄存器或表项的读响应。
[167:152]	16	addr	寄存器或表项的读/写地址
[151:0]	152	data	寄存器或表项的读/写数据；其中五元组映射表的表项位宽最大，为 152bit

附录 H.PTP 协议格式

● Sync, Delay_req, Delay_resq 与 test 报文格式

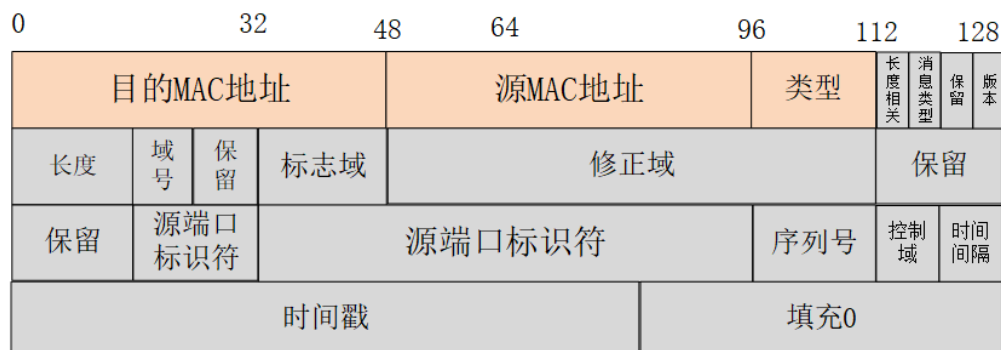


图 H-1 PTP 报文格式

1) 类型为: 16'h88F7;

- 2) 消息类型: sync 为 4'd1, delay_req 为 4'd3, delay_resq 为 4'd4, delay_test 为 4'd5;
- 3) 长度为: 16'd64 字节;
- 4) 修正域:透明时钟, 起始时, 该域为 0;
- 5) 时间戳: 为时间戳 (其他无需关系的 PTP 字段填充 0)

附录 I.TSMP 消息协议格式

TSMP (时间敏感消息协议) 是 TSN 控制器进行网络拓扑探测、对 TSN 芯片和 HCP 进行配置以及对帧进行封装的协议

● TSMP 帧设计原则

- 1) PTP 帧是 TSMP 帧的一种子类型;
- 2) TSNtag 是帧映射后的结果, 在 TSN 网络中根据 TSNtag 对帧进行逻辑处理 (包括查表转发, 入队, 调度优先级, ST 流的按时注入、按时提交、输出门控等);
- 3) 在 TSMP 帧头中设计相关字段用来标识不同类型的 TSMP 帧。

● TSMP 帧格式

TSMP 帧的格式设计如下图所示。

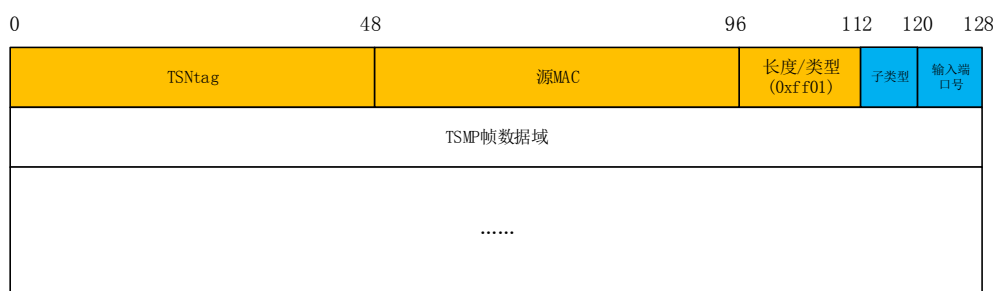


图 I-1 TSMP 帧的格式

图中黄色字段为以太网帧头，蓝色字段为 TSMP 帧头，白色字段为 TSMP 帧数据域。TSMP 帧以太网头和 TSMP 帧头中各字段的含义详见下表。

表 I-1 TSMP 帧头各字段的含义

字段	位宽	说明
TSNtag	48	TSMP 帧经映 2 射所得的结果。
源 mac	48	暂未使用
长度/类型	16	TSMP 帧类型为 0xff01（自定义）。
子类型	8	用来标识不同类型的 TSMP 帧，目前包含 6 种类型：ARP 封装帧、Beacon 帧、芯片配置帧、HCP 配置帧、ICMP 封装帧、Probe 帧。
输入端口号	8	主机发给 TSN 芯片的帧进入 TSN 芯片的端口号

表 I-2 TSMP 帧类型

帧类型	子类型的值	含义
ARP 封装帧	8'h0	ARP 帧封装到 TSMP 帧中在网络中进行传输，将 ARP 帧完整地存放在 TSMP 数据域
Beacon 帧	8'h1	交换机、网卡上报到控制器的状态帧，将交换机、网卡的状态上报帧完整地存放在 TSMP 数据域
芯片配置帧	8'h2	控制器对交换机、网卡进行配置的帧，控制器将 NMAC 配置帧封装到 TSMP 帧中，其中 NMAC 配置帧完整地存放在 TSMP 数据域
HCP 配置帧	8'h3	控制器对 HCP 进行配置的帧；配置信息存放在 TSMP 数据域。
HCP 状态上报帧	8'h4	HCP 上报的状态信息存放在 TSMP 数据域
PTP 封装帧	8'h5	将 PTP 帧（sync 帧、delay_req 帧、delay_resp 帧）封装到 TSMP 帧中，其中 PTP 帧完整地存放在 TSMP 数据域