

硬件控制点 HCP 设计方案

（版本 1.0）

OpenTSN 开源项目组

2021 年 5 月

版本历史

版本	修订时间	修订内容	修订人	文件标识
1.0	2021.5	初版编制		OpenTSN2.0

目录

1、概述	4
2、总体设计	4
2.1. 总体架构	4
2.2 帧的处理流程	6
2.2.1 配置帧处理流程	7
2.2.2 状态上报帧的处理流程	8
2.2.3 帧解封装处理流程	9
2.2.4 帧封装处理流程	10
2.2.5 帧映射的处理流程	12
2.2.6 帧逆映射的处理流程	14
附录 A 数据传输格式	16
附录 B 各表/缓存区/队列的参数	18
附录 C TSMP 消息协议	22
附录 D TSntag 的格式	23
附录 E HCP 的寄存器/表项的说明	24
附录 F HCP 上报的状态信息	26

1、概述

硬件控制点（Hardware Control Point，简称 HCP）是 OpenTSN 开源项目 TSN 网络中针对硬件控制开发的逻辑，负责对集中式控制器和 TSN 芯片（TSN 交换机、TSN 网卡）通信的帧进行封装/解封装、对标准以太网帧查表映射为 TSN 帧以及对 TSN 帧查表逆映射为标准以太网帧。HCP 支持 32 条五元组映射表，支持 256 条逆映射表，支持 1 个千兆以太网接口。

2、总体设计

2.1. 总体架构

HCP 的总体架构框图如图 2-1。

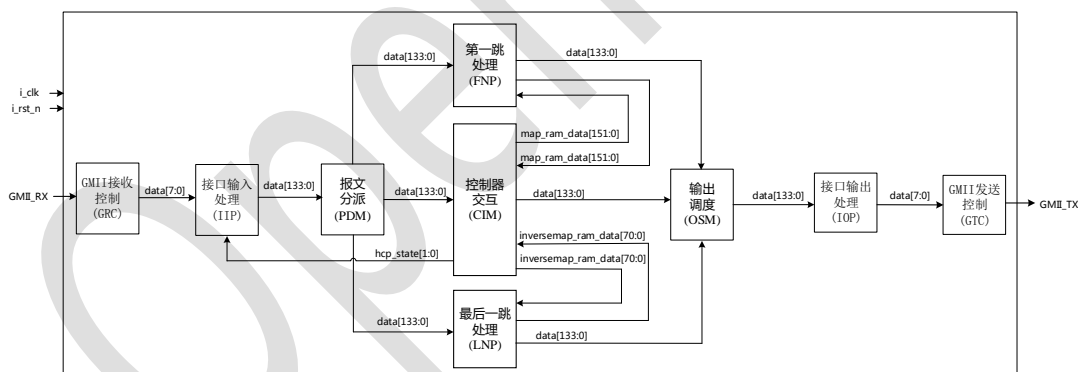


图 2-1 总体架构框图

下面对总体架构图中的逻辑模块进行一一介绍。

GRC（Gmii RxCtrl, GMII 接收控制）模块：主要功能是丢弃帧的 7B 前导码、1B 帧开始符和 4B 的 CRC，并控制帧长度（不包括 CRC）在 60B~1514B。

IIP（Interface Input Process，接口输入处理）模块：主要功能

是对接收到的帧进行跨时钟域（从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域）转换，从帧的 metadata 中提取出 48bit 时间戳和 4bit 输入端口号并去掉该 metadata，将帧的每拍数据位宽由 8bit 转换为 134bit 并根据 HCP 当前所处的状态决定对帧进行丢弃或传输，将无需丢弃的帧和 4bit 输入端口号一起输出给帧分派模块，其中针对 PTP 帧，需要将 48bit 的时间戳存放在 PTP 帧末尾 6B。

PDM（Packet Dispatch Module，报文分派）模块：主要功能是根据帧的以太网类型、在 TSN 芯片中的输入网口号以及网口类型来对帧进行分派：将以太网类型为 0x1662（NMAC 状态上报帧）或者 0x98f7（PTP 帧）或者 0x0806（ARP 帧）或 0xff01（TSMP 帧）的帧，以及从非合作类型的网口输入的非 IP 帧分派至控制器交互模块；将从非合作类型的网口输入的 IP 帧或分片分派至第一跳处理模块；将从合作类型的网口输入的 IP 帧或分片分派至最后一跳处理模块。

CIM（Controller Interactive Module，控制器交互）模块：主要功能是负责将接收到的 NMAC 状态上报帧、PTP 帧、ARP 请求帧、从非合作类型的网口输入的非 IP 帧封装到 TSMP 帧中，并上送给集中式控制器；以及对 HCP 配置封装帧进行解析，完成对 HCP 的配置，将芯片配置封装帧、PTP 封装帧、ARP 封装帧分别解封装成 NMAC 配置帧、PTP 帧、ARP 响应帧后传输给 TSN 芯片。

FNP（First Node Process，第一跳处理）模块：主要功能是根据 IP 首部中的帧总长度字段确定每个分片在原始帧中的偏移，以及将标准以太网帧或分片查表映射为 TSN 帧，即将查表结果 TSntag 替换

DMAC 并将第一个分片的以太网类型由 0x0800 改为 0x1800。

LNP（Last Node Process，最后一跳处理）模块：主要功能是根据 flowID 将不同帧的分片筛选出来，将同一个帧的所有分片按照分片输入 HCP 的顺序连续输出；并且用帧或分片的 flowID 去查逆映射表，在每个帧或分片的 metadata 中指定分片从 TSN 芯片输出的端口号，用查表得到的 DMAC 去替换帧或第一个分片中的 TSNtag，将帧或第一个分片的以太网类型由 0x1800 改为 0x0800，去掉非第一个分片的第一个 16B。

OSM（Output Schedule Module，输出调度）模块：主要功能是对 4 条数据通路的帧进行轮询调度。

IOP（Interface Output Process，接口输出处理）模块：主要功能是实现帧每拍数据位宽的转换（由 134bit 转换为 8bit）并去掉 16B 的 metadata 中低 8B（metadata[63:0]）无效数据，对帧进行跨时钟域处理，在帧从 GMII 接口输出时，添加 7B 的前导码、1B 的帧开始符，并控制每个帧与帧之间的帧间隔最少 16 拍。

GTC（Gmii TxCtrl,GMII 发送控制）模块：主要功能是计算帧的 CRC 并添加到帧的末尾。

2.2 帧的处理流程

HCP 对帧的处理主要有 HCP 配置帧的解析和 HCP 状态帧的上报，ARP 请求帧、PTP 帧、NMAC 状态上报帧、标准以太网非 IP 帧的封装和 TSMP 帧的解封装、标准以太网 IP 帧的映射和 TSN 帧的逆映射。

2.2.1 配置帧处理流程

HCP 配置帧由 GMII 接口输入，在 GMII 接口接收控制模块将帧末尾 4B 的 CRC 去掉，并控制帧的长度（不包括 4B 的 CRC）在 60B~1514B 之间：若接收到的帧长度小于 60B，则在帧的末尾填充 0 至帧的长度为 60B；若接收到的帧长度大于 1514B，则去掉超出 1514B 的数据；若接收到的帧长度在 60B~1514B，则将帧直接输出给接口输入处理模块。

接口输入处理模块先对帧或分片进行跨时钟域（从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域）转换，然后从帧或分片的 metadata（详细格式见附录 A 中 metadata 数据格式）中提取出 48bit 时间戳和 4bit 输入端口号并去掉该 metadata，接下来将帧或分片每拍数据的位宽由 8bit 转换为 134bit 后，根据 HCP 所处的状态 hcp_state 来对帧或分片进行丢弃或传输：在 hcp_state 为 0 或 1 时，HCP 只接收处理 TSMP 帧并丢弃非 TSMP 帧或分片；在 hcp_state 为 2 时，HCP 接收处理所有帧或分片。将需要传输的帧或分片及其在 TSN 芯片的输入端口号一起输出给帧分派模块。

帧分派模块根据帧以太网类型为 0xff01 判定该帧为 TSMP 帧，将 TSMP 帧分派给控制器交互模块。

控制器交互模块根据帧以太网类型为 0xff01 且子类型为 0x3 判定该帧为 HCP 配置帧，然后根据该帧 payload 中的配置有效位和配置地址来将配置数据赋给对应的寄存器或写到相应的表项中。

2.2.2 状态上报帧的处理流程

控制器交互模块每次接收到 TSN 芯片传来的 NMAC 状态上报帧后，都会触发 HCP 上报一次本地状态；HCP 根据控制器配置的上报类型来决定上报的内容：若上报类型高 4bit 为 0 且低 12bit 为 0x000，则 HCP 会上报当前时刻本地寄存器的值，同时会将所有帧的计数器清零；若上报类型高 4bit 为 1，则 HCP 会上报当前时刻本地五元组映射表内容，每次上报 2 个表项内容，具体上报哪一块表项内容由上报类型的低 12bit 指定；若上报类型高 4bit 为 2，则 HCP 会上报当前时刻本地逆映射表内容，每次上报 4 个表项内容，具体上报哪一块表项内容由上报类型的低 12bit 指定。控制器交互模块构造 HCP 状态帧，并在帧前面增加 16B 的 metadata（详细格式见附录 A 中 metadata 数据格式），然后将帧写到 fifo 中等待调度。

控制器交互模块 fifo 缓存的帧经输出调度模块调度后输出给接口输出处理模块。

接口输出处理模块将 HCP 状态帧的位宽由 134bit 转换为 8bit 并丢弃 16B 的 metadata 中的低 8B（即 metadata[63:0]），以及实现帧的跨时钟域（从 HCP 内部逻辑工作时钟域到 GMII 发送时钟域）转换，然后将每拍 8bit 的帧输出给 GMII 接口发送控制模块。

GMII 接口发送控制模块将接收到的数据从 GMII 接口输出，同时根据接收到的数据计算得到 4B 的 CRC 并填充在帧的末尾。

2.2.3 帧解封装处理流程

HCP 对接收的 TSMP 帧进行解封装，该帧的解封装处理流程如下所述。

数据帧由 GMII 接口输入，在 GMII 接口接收控制模块将帧末尾 4B 的 CRC 去掉，并控制帧的长度(不包括 4B 的 CRC)在 60B~1514B 之间，若接收到的帧长度小于 60B，则在帧的末尾填充 0 至帧的长度为 60B；若接收到的帧长度大于 1514B，则去掉超出 1514B 的数据；若接收到的帧长度在 60B~1514B，则将帧直接输出给接口输入处理模块。

接口输入处理模块先对帧进行跨时钟域(从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域)转换，然后从帧的 metadata(详细格式见附录 A 中 metadata 数据格式)中提取出 48bit 时间戳和 4bit 输入端口号并去掉该 metadata，接下来将帧的位宽由 8bit 转换为 134bit 后，根据 HCP 所处的状态来 hcp_state 对帧进行丢弃或传输：在 hcp_state 为 0 或 1 时，HCP 只接收处理 TSMP 帧并丢弃非 TSMP 帧或分片；在 hcp_state 为 2 时，HCP 接收处理所有帧或分片。将需要传输的帧或分片及其在 TSN 芯片的输入端口号一起输出给帧分派模块。

帧分派模块根据帧以太网类型为 0xff01 判定该帧为 TSMP 帧，将 TSMP 帧分派给控制器交互模块。

控制器交互模块根据帧以太网类型为 0xff01 且子类型字段为 0x0 或 0x2 或 0x5 判定该帧为 ARP 封装帧或 PTP 封装帧或芯片配置帧，

则将其解封装（去掉 16B 的 TSMP 帧头）为 ARP 应答帧、PTP 帧、NMAC 配置帧，并在解封装后的帧前面增加 16B 的 metadata，其中针对 ARP 应答帧，需要将 ARP 封装帧的 TSMP 帧头中的输入端口号作为 ARP 应答帧在最后一跳芯片的输出端口号（在 metadata 中指定该 ARP 应答帧的输出端口号，在 TSN 芯片中不查找转发表），然后将帧写到 fifo 中等待调度。

控制器交互模块 fifo 中缓存的帧经输出调度模块调度后输出给接口输出处理模块。

接口输出处理模块将帧的位宽由 134bit 转换为 8bit 并丢弃 16B 的 metadata 中的低 8B（即 metadata[63:0]），以及实现帧的跨时钟域（从 HCP 内部逻辑工作时钟域到 GMII 发送时钟域）转换，然后将每拍 8bit 的帧输出给 GMII 接口发送控制模块。

GMII 接口发送控制模块将接收到的数据从 GMII 接口输出，同时根据接收到的数据计算得到 4B 的 CRC 并填充在帧的末尾。

2.2.4 帧封装处理流程

HCP 会对接收到的 ARP 请求帧、PTP 帧、NMAC 状态上报帧、标准以太网非 IP 帧进行封装，这些帧的封装处理流程如下所述。

数据帧由 GMII 接口输入，在 GMII 接口接收控制模块将帧末尾 4B 的 CRC 去掉，并控制帧的长度（不包括 4B 的 CRC）在 60B~1514B 之间：若接收到的帧长度小于 60B，则在帧的末尾填充 0 至帧的长度为 60B；若接收到的帧长度大于 1514B，则去掉超出 1514B 的数据；

若接收到的帧长度在 60B~1514B，则将帧直接输出给接口输入处理模块。

接口输入处理模块先对帧进行跨时钟域（从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域）转换，然后从帧的 metadata（详细格式见附录 A 中 metadata 数据格式）中提取出 48bit 时间戳和输入端口号并去掉该 metadata，接下来将帧的位宽由 8bit 转换为 134bit 后，根据 HCP 所处的状态 hcp_state 来对帧进行丢弃或传输：在 hcp_state 为 0 或 1 时，HCP 只接收处理 TSMP 帧并丢弃非 TSMP 帧或分片；在 hcp_state 为 2 时，HCP 接收处理所有帧或分片。将需要传输的帧或分片及其在 TSN 芯片的输入端口号一起输出给帧分派模块，其中针对 PTP 帧，需要将 48bit 的时间戳放在 PTP 帧末尾 6B。

帧分派模块根据帧以太网类型为 0x0806、0x98f7、0x1662 判定该帧为 ARP 请求帧、PTP 帧、NMAC 状态上报帧，根据输入网口的类型为非合作类型且帧以太网不是 0x0800 判定该帧为标准以太网非 IP 帧并将帧分派给控制器交互模块。

控制器交互模块将 ARP 请求帧、PTP 帧、NMAC 状态上报帧封装到 TSMP 帧中（即在 ARP 请求帧、PTP 帧、NMAC 状态上报帧前面增加 16B 的 TSMP 帧头），针对 NMAC 状态上报帧，在封装时会去掉 NMAC 状态上报帧的 DMAC、SMAC、以太网类型，并将以太网类型后面的 2B 上报类型字段放在帧的末尾；在封装后的 TSMP 帧前面增加 16B 的 metadata，然后将帧写到 fifo 中等待调度。

控制器交互模块 fifo 中缓存的帧经输出调度模块调度后输出给

接口输出处理模块。

接口输出处理模块将帧的位宽由 134bit 转换为 8bit 并丢弃 16B 的 metadata 中的低 8B（即 metadata[63:0]），以及实现帧的跨时钟域（从 HCP 内部逻辑工作时钟域到 GMII 发送时钟域）转换，然后将每拍 8bit 的帧输出给 GMII 接口发送控制模块。

GMII 接口发送控制模块将接收到的数据从 GMII 接口输出，同时根据接收到的数据计算得到 4B 的 CRC 并填充在帧的末尾。

2.2.5 帧映射的处理流程

HCP 会对接收的标准以太网 IP 帧或分片进行映射，该帧或分片的映射处理流程如下所述。

数据帧或分片由 GMII 接口输入，在 GMII 接口接收控制模块将帧或分片末尾 4B 的 CRC 去掉，并控制帧或分片的长度（不包括 4B 的 CRC）在 60B~1514B 之间，若接收到的帧或分片长度小于 60B，则在帧或分片的末尾填充 0 至帧的长度为 60B，若接收到的帧或分片长度大于 1514B，则去掉超出 1514B 的数据；若接收到的帧或分片长度在 60B~1514B，则将帧或分片直接输出给接口输入处理模块。

接口输入处理模块先对帧或分片进行跨时钟域（从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域）转换，然后从帧或分片的 metadata（详细格式见附录 A 中 metadata 数据格式）中提取出 48bit 时间戳和 4bit 输入端口号并去掉该 metadata，接下来将帧或分片的位宽由 8bit 转换为 134bit 后，根据 HCP 所处的状态 hcp_state 来对帧或分片进行

丢弃或传输：在 hcp_state 为 0 或 1 时，HCP 只接收处理 TSMP 帧并丢弃非 TSMP 帧或分片；在 hcp_state 为 2 时，HCP 接收处理所有帧或分片。将需要传输的帧或分片及其在 TSN 芯片的输入端口号一起输出给帧分派模块。

帧分派模块根据帧或分片的输入网口（输入 TSN 芯片的网口）的类型为非合作类型判定该帧为标准以太网帧或分片；再根据以太网类型字段是否为 0x0800 判断该标准以太网帧或分片是否为 IP 帧或分片，将标准以太网非 IP 帧分派给控制器交互模块进行封装，将标准以太网 IP 帧或分片分派给第一跳处理模块进行映射。

第一跳处理模块先根据 IP 首部中的帧总长度字段确定分片在原始帧中的偏移量，并判断该帧的分片（如果该帧有分片的话）是否有丢失，若帧的第一个分片丢失，则将该帧的其余所有分片丢弃；若帧的中间分片或最后一个分片丢失，则将该帧的其余所有分片去查表；若帧的分片无丢失，则将该帧的所有分片去查表。在本模块维护 2 个表，一个是五元组到 TSNTag 的映射表，即五元组映射表（32 条表项）；另一个是输入端口号到 TSNTag 的映射表，即 TSNTag 动态表（8 条表项）；从第一个分片（如果该帧未被分片，则第一个分片即为完整的帧）中提取五元组，用五元组去查找五元组映射表，若查表匹配某一条表项，则将查表结果 TSNTag 写到 TSNTag 动态表中，写地址为该分片的输入端口号，同时将 TSNTag（frag_flag 和 frag_id 需要根据分片在原始帧中的偏移量来填充具体的值）替换第一个分片的 DMAC 并将以太网类型由 0x0800 改为 0x1800，然后将帧写到 fifo 中等待调

度；非第一个分片则根据该分片的输入端口号去查找 TSNtag 动态表，将 TSNtag（frag_flag 和 frag_id 需要根据分片在原始帧中的偏移量来填充具体的值）替换该分片的 DMAC，然后将该分片写到 fifo 中等待调度。

第一跳处理模块 fifo 中缓存的帧或分片经输出调度模块调度后输出给接口输出处理模块。

接口输出处理模块将帧或分片的位宽由 134bit 转换为 8bit 并丢弃 16B 的 metadata 中的低 8B（即 metadata[63:0]），以及实现帧或分片的跨时钟域（从 HCP 内部逻辑工作时钟域到 GMII 发送时钟域）转换，然后将每拍 8bit 的帧或分片输出给 GMII 接口发送控制模块。

GMII 接口发送控制模块将接收到的数据从 GMII 接口输出，同时根据接收到的数据计算得到 4B 的 CRC 并填充在帧的末尾。

2.2.6 帧逆映射的处理流程

HCP 会对接收到的 TSN 帧或分片进行逆映射，该帧或分片的逆映射处理流程如下所述。

数据帧或分片由 GMII 接口输入，在 GMII 接口接收控制模块将帧或分片末尾 4B 的 CRC 去掉，并控制帧或分片的长度（不包括 4B 的 CRC）在 60B~1514B 之间：若接收到的帧或分片长度小于 60B，则在帧或分片的末尾填充 0 至帧的长度为 60B；若接收到的帧或分片长度大于 1514B，则去掉超出 1514B 的数据；若接收到的帧或分片长度在 60B~1514B，则将帧或分片直接输出给接口输入处理模块。

接口输入处理模块先对帧或分片进行跨时钟域（从 GMII 接收时钟域到 HCP 内部逻辑工作时钟域）转换，然后从帧或分片的 metadata（详细格式见附录 A 中 metadata 数据格式）中提取出帧 48bit 时间戳和 4bit 输入端口号并去掉该 metadata，接下来将帧或分片的位宽由 8bit 转换为 134bit 后，根据 HCP 所处的状态 hcp_state 来对帧或分片进行丢弃或传输：在 hcp_state 为 0 或 1 时，HCP 只接收处理 TSMP 帧并丢弃非 TSMP 帧或分片；在 hcp_state 为 2 时，HCP 接收处理所有帧或分片。将需要传输的帧或分片及其在 TSN 芯片的输入端口号一起输出给帧分派模块。

帧分派模块根据帧或分片的输入网口（输入 TSN 芯片的网口）的类型为合作类型且以太网类型非 0x0806（ARP 帧）、非 0x98f7（PTP 帧）、非 0x1662（NMAC 状态上报帧）、非 0xff01（TSMP 帧）判定该帧为 TSN 帧或分片，并将 TSN 帧或分片分派给最后一跳处理模块。

最后一跳处理模块维护 32 个队列，每个队列的深度为 16（1514B 的最长帧的分片数不超过 16）；1 个位宽为 134、深度为 4096 的帧缓存区，为每个帧预留一定的存储空间；1 个位宽为 9 深度为 512 的 fifo，用来存储帧缓存逻辑地址 bufid；1 个深度为 32 的流表，每条表项用来记录帧的 flowID 以及队列已使用的深度。本模块根据分配的空闲 bufid 将帧写到对应的缓存块中，并用 flowID 按照顺序查流表，记录第一个无效表项，若匹配命中某一条表项，则更新该表项（将队列深度加 1）；若 32 条表项均未匹配命中，则将该 flowID 和队列深度（值填充 1）写到第一个无效表项中；根据表项的地址（即队列 ID）和该

表项中队列已使用的深度字段将 `bufid` 和最后一个分片标识写到队列中。当最后一个分片写到队列中，则开始读取该队列，并根据从队列中读出的 `bufid` 从帧缓存区中读取帧，在最后一个分片从帧缓存区读出后，将对应的队列释放，即该队列可以存储新的帧控制信息。然后用每个分片 `TSNtag` 中的 `flowID` 按照顺序去查逆映射表获得 `DMAC` 和 `outport`，在每个分片增加 16B 的 `metadata` 并在 `metadata` 中指定帧在 TSN 芯片的输出端口号 `outport`，分片在 TSN 芯片中不查找转发表，若分片为第一个分片，则将得到的 `DMAC` 去替换帧中的 `TSNtag`，并将以太网类型由 `0x1800` 改为 `0x0800`；若该分片为中间分片或最后一个分片，则将该分片第一个 16B 去掉。将分片写到 `fifo` 中等待调度。

最后一跳处理模块 `fifo` 中缓存的帧经输出调度模块调度后输出给接口输出处理模块。

接口输出处理模块将帧的位宽由 134bit 转换为 8bit 并丢弃 16B 的 `metadata` 中的低 8B（即 `metadata[63:0]`），以及实现帧的跨时钟域（从 HCP 内部逻辑工作时钟域到 GMII 发送时钟域）转换，然后将每拍 8bit 的帧输出给 GMII 接口发送控制模块。

GMII 接口发送控制模块将接收到的数据从 GMII 接口输出，同时根据接收到的数据计算得到 4B 的 CRC 并填充在帧的末尾。

附录 A 数据传输格式

● `pkt_data` 数据格式

`pkt_data` 位宽为 134 位，包含 2bit 头尾标志、4bit 无效字节数、

128bit 帧数据。头尾标志：01 表示帧头；11 表示帧体中间数据；10 表示帧尾。无效字节数用于标识帧尾中无效的字节数。具体如图 A-1 所示。

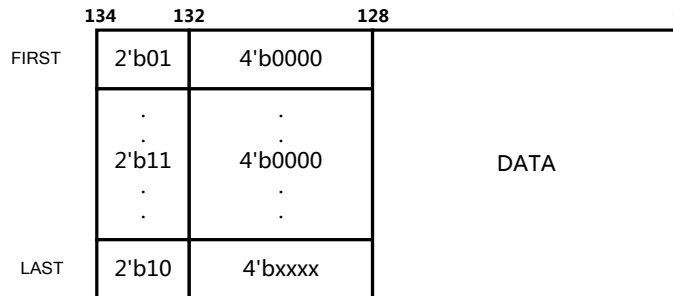


图 A-1 Pkt_data 格式定义

● metadata 数据格式

在 HCP 与 TSN 芯片之间传输的 pkt_data 前面有 8B 的 Metadata。

表 A-1 HCP 发送给 TSN 芯片的帧的 metadata 格式

位置	位宽	名称	含义	备注
[63:61]	3	pkttype	帧的类型。	000: TS 分组 001: TS 分组 010: TS 分组 011: RC 分组 100: PTP 分组 101: NMAC 分组 110: BE 分组 111: 需重组分组
[60:56]	5	inject_addr	TS 注入缓存地址	/
[55:47]	9	outport	帧的输出端口号	/
[46]	1	lookup_en	查表使能信号	/
[45]	1	frag_last	重组帧的最后一个片	与 pkttype 配合使用，帧类型为“需重组分组”时，frag_last 才是有效的。默认为 1
[44:0]	45	reserve	保留	/

表 A-2 HCP 接收到 TSN 芯片传来的帧的 metadata 格式

位置	位宽	名称	含义	备注
[63:16]	48	ts	同步帧全局时间信息	用于记录时间同步帧在主时钟设备接收到 req 帧的全局时间信息，以及从时钟设备接收到 sync 帧的全局时间信息
[15:12]	4	inport	帧的输入端口号	/
[11:0]	12	reserve	保留	/

附录 B 各表/缓存区/队列的参数

● 五元组映射表

表 B-1 五元组映射表格式

索引地址	表项内容	位置	位宽 (bit)	字段	含义
地址从0到31依次查找，直至读出的表项内容中五元组与帧五元组命中。	5tuple	[223:131]	92	reserve	保留
		[130:123]	8	ip_protocol	五元组中 IP 协议
		[122:91]	32	sip	五元组中源 IP
		[90:59]	32	dip	五元组中目的 IP
		[58:43]	16	sport	五元组中源端口
		[42:27]	16	dport	五元组中目的端口
	TSNtag 部分字段	[26:24]	3	flow_type	000: TS 分组 001: TS 分组 010: TS 分组 011: RC 分组 100: PTP 分组 101: NMAC 分组 110: BE 分组 111: 需重组分组
		[23:10]	14	flowID/imac	流标识
		[9:5]	5	injection_addr	TS 流在源端等待发送调度时缓存地址
		[4:0]	5	submit_addr	TS 流在终端等待接收调度时缓存地址

缓存五元组映射表的 ram 位宽为 131，深度为 32（TSN 芯片最多支持 32 条 TS 流）。

● TSNtag 动态表

表 B-2 TSNtag 动态表格式

地址	表项内容	位置	位宽 (bit)	字段	含义
inport (3bit)	TSNtag	[47:45]	3	Flow type	000: TS 分组 001: TS 分组 010: TS 分组 011: RC 分组 100: PTP 分组 101: NMAC 分组 110: BE 分组 111: 需重组分组
		[44:31]	14	Flowid	流标识
		[30:15]	16	Seq id	用于标识每条流中帧的序列号
		[14]	1	Frag flag	用于标识分片后的尾。0: 分片后的中间帧 1: 尾拍
		[13:10]	4	Frag ID	用于表示当前分片帧在原帧中的分片序列号
		[9:5]	5	inject addr	TS流在源端等待发送调度时缓存地址
		[4:0]	5	submit addr	TS流在终端等待接收调度时缓存地址

缓存 TSNtag 动态表的 ram 位宽为 48，深度为 8。

● 流映射表

该表一共有 32 条表项，对应队列缓存区的 32 个队列。

表 B-3 flowID 映射表格式

索引地址	字段	位置	位宽 (bit)	含义
依次从地址0到31读出表项内容	Valid	[18]	1	表项有效位
	flowID	[17:4]	14	流标识
	usedw	[3:0]	4	该队列中已使用的深度

缓存 flowID 映射表的 ram 位宽为 19，深度为 32。

● 逆映射表

该表一共有 256 项，表项内容为 14bit 的 flowID、48bit 的 DMAC 和 9bit 的 outport。

表 B-4 逆映射表格式

索引地址	字段	位置	位宽 (bit)	含义
依次从地址 8'd0 到 8'd 255 读出表项内容	flowID	[70:57]	14	流ID
	DMAC	[56:9]	48	目的mac
	outport	[8:0]	9	输出端口号

缓存逆映射表的 ram 位宽为 71，深度为 256。

● 队列缓存区

目前设计同时最多支持缓存 32 个完整帧，一个完整帧最多有 16 个分片，bufid 应有 $32 \times 16 = 512$ ，即 bufid 位宽为 9。

表 B-5 队列缓存区格式

地址	字段	位置	位宽 (bit)	含义
queueid	last_frag_flag	[9]	1	最后一个分片的标识位
	bufid	[8:0]	9	帧在缓存区的逻辑地址

队列缓存区的 ram 位宽为 10，深度为 $32 \times 16 = 512$ 。

● 分片帧缓存区

目前设计最多支持缓存 512 个帧，每个帧长度不超过 128B（8 拍），帧缓存区深度应为 $32 \times 16 \times 8 = 4096$ ，位宽为 134bit。

● 空闲地址缓存队列

目前设计最多支持缓存 512 个帧，相应的 bufid 个数设计为 512 个，即队列深度为 512，队列位宽即 bufid 位宽为 9。

● 帧映射缓存队列

由于五元组映射表有 32 条表项，从帧中提取五元组需要 4 拍，查表最多需要 32 拍才能得到查表结果，所以需要使用一个队列来缓存帧，该队列深度只需要设计为 64 即可，队列的位宽设计为 134bit。

● 帧逆映射缓存队列

由于逆映射表有 256 条表项，查表最多需要 256 拍才能得到查表结果，所以需要使用一个队列来缓存帧，该队列深度设计为 128，队列的位宽设计为 134bit。

● 帧缓存队列

由于输出调度模块需要对 4 条通路的帧进行轮询调度，每条通路都有一个队列来缓存帧：

缓存封装/解封装后的帧的队列需要能缓存 4 个帧，为每个帧预留 128 的缓存深度，该队列位宽设置为 134bit，深度设置为 512；

缓存 HCP 上报 Beacon 帧的队列只需要能缓存 1 个帧即可，因为 HCP 状态上报周期最小为 1ms，在 1ms 时间内，输出调度模块能将 4 条通路轮询调度一遍，所以该队列位宽设置为 134bit，深度设置为 128；

缓存映射后帧的队列需要能缓存 4 个帧，为每个帧预留 128 的缓存深度，该队列位宽设置为 134bit，深度设置为 512；

缓存逆映射后帧的队列只需要能缓存 1 个帧即可，因为可等到该队列有足够的空闲空间时，才从报文集中缓存模块读出帧，所以该队列位宽设置为 134bit，深度设置为 128。

附录 C TSMP 消息协议

TSMP（时间敏感管理协议）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) TSMP 帧长度不超过 128B；
- 2) PTP 帧是 TSMP 帧的一种子类型；
- 3) 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）；
- 4) 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 帧的格式设计如图 C-1 所示。

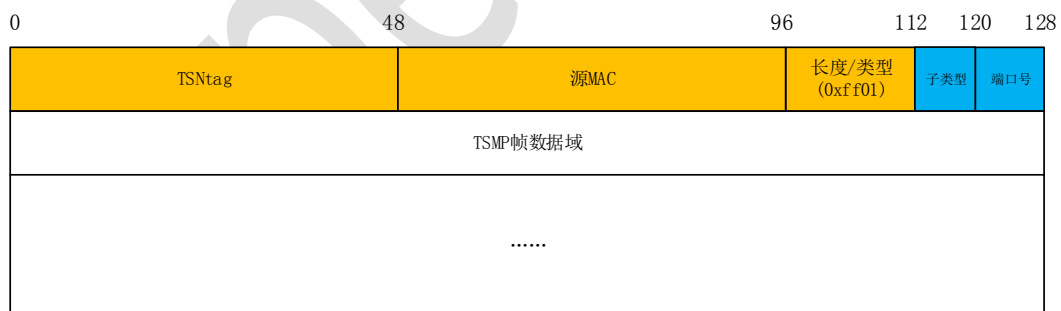


图 C-1 TSMP 帧格式

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表 C-1。

表 C-1 TSMP 帧以太网头和 TSMP 帧头各字段的含义

字段	位宽	说明
TSNtag	48	TSMP 帧经映射所得的结果。
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、芯片上报 Beacon 帧、芯片配置帧、HCP 配置帧、HCP 上报 Beacon 帧、PTP 封装帧。
端口号	8	帧在第一跳的输入端口号（由 HCP 填充该字段）或者帧在最后一跳的输出端口号（由集中式控制器填充该字段）

表 C-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
芯片上报 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 Beacon 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

附录 D TSNtag 的格式

表 D-1 非时间同步帧的 TSNtag

位(bit)	名称	位置	描述
3	Flow type	[47:45]	000: TS 分组 001: TS 分组 010: TS 分组 011: RC 分组 100: PTP 分组 101: NMAC 分组 110: BE 分组 111: 需重组分组

14	Flow id/ Imac	[44:31]	静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换节点命中同一条表项。
16	Seq id	[30:15]	用于标识每条流中帧的序列号
1	Frag flag	[14]	用于标识分片后的尾。0：分片后的中间帧 1：尾拍
4	Frag ID	[13:10]	用于表示当前分片帧在原帧中的分片序列号
5	inject addr	[9:5]	TS 流在源端等待发送调度时缓存地址
5	submit addr	[4:0]	TS 流在终端等待接收调度时缓存地址

附录 E HCP 的寄存器/表项的说明

HCP 的寄存器/表项等由控制器发送 HCP 配置帧（格式详见图 E-1）来进行配置，配置寄存器/表项需要包含 3 部分信息：配置有效位，配置地址，配置寄存器/表项。

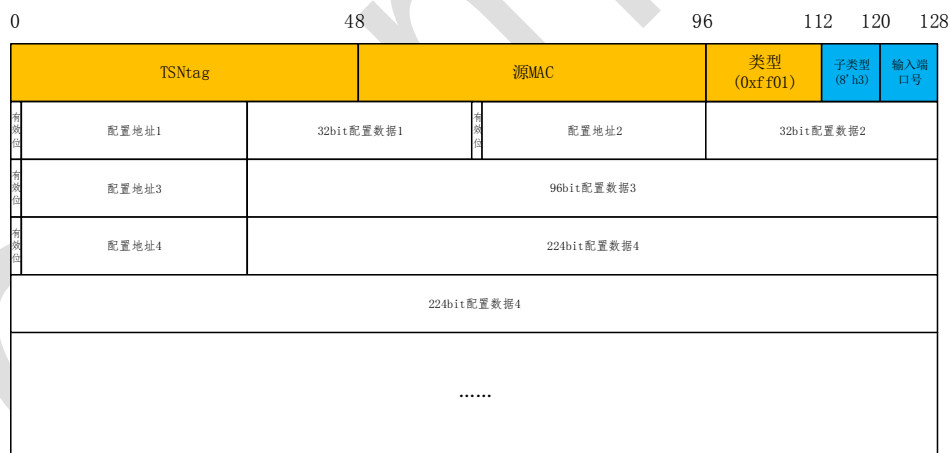


图 E-1 HCP 配置帧格式

表 E-1 每个配置寄存器/表项包含的信息

字段	位宽 (bit)	说明
配置的有效位 Valid	1	1 表示寄存器/表项配置有效；0 表示寄存器/表项配置无效；
配置的地址 addr	31	高 7bit(addr[30:24])表示该寄存器/表项所在的硬件模块号 MID，最大支持 128 个硬件模块； 低 24bit(addr[23:0])表示该寄存器/表项在硬件模块中被

		分配的地址。
配置的寄存器/ 表项	/	该字段根据实际的寄存器/表项的位宽进行设计。 针对芯片端口类型寄存器，该字段为 32bit； 针对五元组映射表（每项为 152bit），该字段为 224bit； 针对逆映射表（每项为 57bit），该字段为 96bit。

表 E-2 HCP 中模块的地址划分

模块 地址	帧分派模块 PDM	TSMP 帧处理模 块 TPM	第一跳处理模块 FNP	最后一跳处理 模块 LNP
MID	7'h0	7'h1	7'h2	7'h3
Addr	31'h00000000 - 31'h00ffffff	31'h01000000 - 31'h01ffffff	31'h02000000 - 31'h02ffffff	31'h03000000 - 31'h03ffffff

表 E-3 HCP 中所有需配置信息

所属 模块	地址	寄存器/ 表项	位宽 (bit)	说明	读 写	默认 值
报文 分派 模块 PDM	31'h00 00000 0	chip_port_t ype	[31:8]	保留	R/W	8'hff
			[7:0]	HCP 直连的 TSN 芯片 的 8 个网口的类型，每个 网口对应 1bit; 1 表示 非协作类型，0 表示协 作类型。		
控制 器交 互模 块 CIM	31'h01 00000 0	hcp_state	[31:2]	保留	R/W	1'b0
			[1:0]	HCP 的状态；0 表示 HCP 初始化 512 个 bufid 的状态，1 表示只接收 处理 TSMP 帧的状态,2 表示接收处理所有帧的 状态。		
	31'h01 00000 1	report_type	[31:16]	保留	R/W	1'b0
			[15:0]	上报类型。report_type [15:12]表示上报种类； 0: 上报寄存器；1: 上 报五元组映射表；2: 上 报逆映射表。 report_type[11:0]表示具 体上报的某一块内容。 详见下表。		
第一 跳处 理模 块	31'h02 00000 0 -	map_table _N	[223:132]	保留	R/W	132'h0
			[131]	读取五元组映射表时， 读写冲突检查信号；0: 未出现同时读写；1: 出		

所属模块	地址	寄存器/表项	位宽 (bit)	说明	读写	默认值
FNP	31'h0200001F			现同时读写		
			[151:144]	五元组中 IP 协议		
			[143:112]	五元组中源 IP		
			[111:80]	五元组中目的 IP		
			[79:64]	五元组中源端口		
			[63:48]	五元组中目的端口		
			[47:45]	TSNtag 中的 flow_type		
			[44:31]	TSNtag 中的 flowID/imac		
			[30:15]	TSNtag 中的 Seq id;在硬件 ram 中不存储该字段。		
			[14]	TSNtag 中的 Frag flag;在硬件 ram 中不存储该字段。		
			[13:10]	TSNtag 中的 Frag ID;在硬件 ram 中不存储该字段。		
			[9:5]	TSNtag 中的 injection_addr		
			[4:0]	TSNtag 中的 submit_addr		
最后一跳处理模块 LNP	31'h03000000 - 31'h03003FFF	regroup_table_N	[77:64]	flowID	R/W	78'h0
			[63:16]	每条重组表表项的高 48bit 为 DMAC		
			[15:10]	reserve		
			[9]	读取逆映射表时，读写冲突检查信号；0：未出现同时读写；1：出现同时读写		
			[8:0]	每条重组表表项的低 9bit 为输出端口号		

附录 F HCP 上报的状态信息

HCP 上报的状态主要包括端口/各模块的收发帧计数器和各个模块的状态信息，具体如下。

表 F-1 上报的信息

上报类型（16bit）		说明
高 4bit	低 12bit	
4'h0 单个寄存器	12'h0	详见下表
4'h1 五元组映射表	12'h0	第 0-1 条五元组映射表
	12'h1	第 2-3 条五元组映射表

	12'hf	第 30-31 条五元组映射表
4'h2 逆映射表	12'h0	第 0-3 条逆映射表
	12'h1	第 4-7 条逆映射表

	12'h3f	第 60-63 条逆映射表

表 F-2 各模块上报的计数器统计

名称	位宽	含义
port_inpkt_cnt	16	接口接收帧个数计数器
port_outpkt_cnt	16	接口发送帧个数计数器
port_rx_asynfifo_overflow_cnt	16	端口接收端异步 fifo 上溢计数器
port_rx_asynfifo_underflow_cnt	16	端口接收端异步 fifo 下溢计数器
port_tx_asynfifo_overflow_cnt	16	端口发送端到异步 fifo 上溢计数器
port_tx_asynfifo_underflow_cnt	16	端口发送端异步 fifo 下溢计数器
frc_discard_pkt_cnt	16	全网处于配置状态时，FRC 模块丢弃的非 TSMP 帧的个数计数器
pdm_first_node_notip_discard_pkt_cnt	16	第一个节点 HCP 接收到主机发来的非 IP 帧（被丢弃）个数计数器
cim_inpkt_cnt	16	控制器交互模块接收帧个数计数器
cim_outpkt_cnt	16	控制器交互模块发送帧个数计数器
lcm_inpkt_cnt	16	本地配置管理模块接收帧个数计数器
srm_outpkt_cnt	16	状态上报模块发送帧个数计数器
cim_extfifo_overflow_cnt	16	存储封装/解封装后的帧的 fifo 上溢计数器
cim_intfifo_overflow_cnt	16	存储 HCP 状态上报帧的 fifo 上溢计数器
fnp_inpkt_cnt	16	第一跳处理模块接收帧个数计数器

名称	位宽	含义
fnp_outpkt_cnt	16	第一跳处理模块发送帧个数计数器
fnp_fifo_overflow_cnt	16	第一跳处理模块中 fifo 上溢计数器
fnp_no_1st_frag_cnt	16	第一跳处理模块接收到的被分片帧的第一个分片被丢弃的计数器
fnp_no_not1st_frag_cnt	16	第一跳处理模块接收到的被分片帧的非第一个分片（中间分片或最后一个分片）被丢弃的计数器
lnp_inpkt_cnt	16	最后一跳处理模块接收帧个数计数器
lnp_outpkt_cnt	16	最后一跳处理模块发送帧个数计数器
lnp_no_notlast_frag_flag_cnt	16	最后一跳处理模块接收到的被分片帧的非最后一个分片（第一个分片或中间分片）被丢弃的计数器
lnp_no_last_frag_flag_cnt	16	最后一跳处理模块接收到的被分片帧的最后一个分片被丢弃的计数器
lnp_fifo_overflow_cnt	16	最后一跳处理模块中 fifo 上溢计数器
lnp_flow_table_overflow_cnt	16	最后一跳处理模块中流表溢出的计数器

48				96				112		120		128	
TSNtag					源MAC					类型 (0xf01)		子类型 (8'h4)	输入端 口号
report_type	chip_port_type	hep_state	reserve										
port_inpkt_cnt		port_rx_asyncfifo_overflow_cnt		port_rx_asyncfifo_underflow_cnt		fre_discard_pkt_cnt		reserve					
pdm_first_node_notip_discard_pkt_cnt		reserve											
cim_inpkt_cnt		cim_outpkt_cnt		lcm_inpkt_cnt		srm_outpkt_cnt		cim_extfifo_overflow_cnt		cim_intfifo_overflow_cnt		reserve	
fnp_inpkt_cnt		fnp_outpkt_cnt		fnp_fifo_overflow_cnt		fnp_no_1st_frag_cnt		fnp_no_not1st_frag_cnt		reserve			
lnp_inpkt_cnt		lnp_outpkt_cnt		lnp_no_notlast_frag_flag_cnt		lnp_no_last_frag_flag_cnt		lnp_fifo_overflow_cnt		lnp_flow_table_overflow_cnt		reserve	
port_outpkt_cnt		port_tx_asyncfifo_overflow_cnt		port_tx_asyncfifo_underflow_cnt		reserve							

图 F-1 HCP 寄存器上报帧格式



图 F-2 HCP 五元组映射表上报帧格式

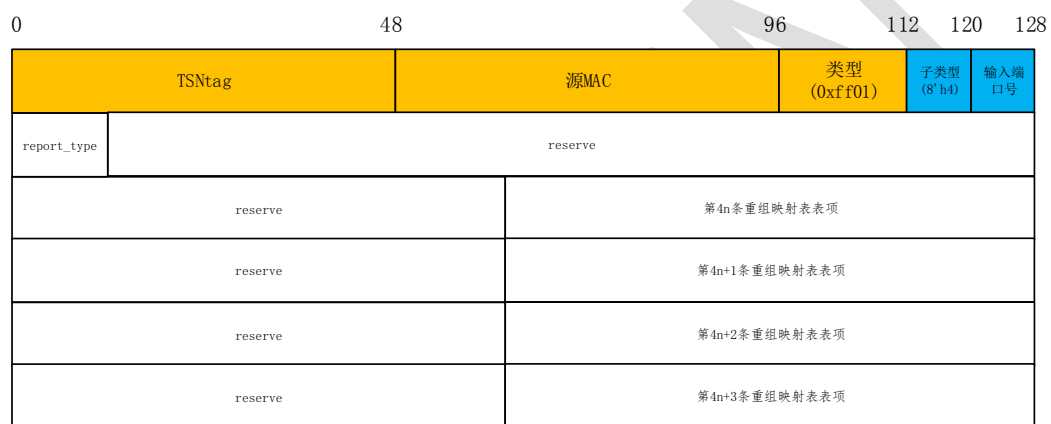


图 F-3 HCP 逆映射表上报帧格式