MASTER THESIS XIDIAN UNIVERSITY

西安电子科技大学

硕士学位论文

兼容AFDX的专用以太网交换机分组处理的设计与实现

作者姓名 何 振

指导教师姓名、职称 鲍民权 副教授

申请学位类别 工学硕士

学校代码 10701 学 号 1301120007
分 类 号 TN915 密 级 公开

西安电子科技大学

硕士学位论文

兼容AFDX的专用以太网交换机分组处理的设计与实现

作者姓名：何振

一级学科：信息与通信工程

二级学科：通信与信息系统

学位类别：工学硕士

指导教师姓名、职称：鲍民权 副教授

学　　院：通信工程学院

提交日期：2015 年 11 月

Design and Implementation of Packet Processing in Private Ethernet Switch Compatible with AFDX

A thesis submitted to

XIDIAN UNIVERSITY

in partial fulfillment of the requirements

for the degree of Master

in Communications and Information Systems

By He Zhen

Supervisor: Bao Minquan Associate Professor

November 2015

摘要

随着航天航空电子技术的快速发展以及复杂的电子设备的增加，信息的传输变得越来越频繁，交换在其中扮演了重要的角色。本文结合实验室承接的“航天专用分组交换机的设计与实现”，根据该项目的要求，本人主要设计并实现了能够同时支持以太网及机载全双工以太网（AFDX）的分组处理模块及转发表模块。

本文首先介绍了研究的背景，以太网交换以及 AFDX 技术。其次，根据对项目的功能及性能需求，给出可行的实现方案，并对总体的设计结构以及各个模块的功能进行介绍。第三，详细介绍分组处理模块及转发表模块的功能、接口及实现。第四，在 Modelsim 中对分组处理及转发表模块等主要模块进行功能仿真。最后，搭建测试平台，进行板级测试，并对仿真及测试中出现的问题进行说明和总结。目前该设计方案已经通过板级验证，表明所设计的方案满足项目需求，达到了预期的目标。

关 键 词：分组处理，VLAN，二层转发，AFDX

ABSTRACT

With the rapid development of aerospace technology and the increasing complexity of electronic devices, the transmission of information has become more and more frequent exchange has played an important role in it. This thesis is based on a research program "Design and Implementation of Private Switch in Aerospace". According to the requirements of the project, the key technologies of packet processing and transformation talble which support both Ethernet and AFDX are focused on in this thesis .

This thesis introduces the research background , Ethernet switching and AFDX technology firstly. Secondly, according to functional and performance requirements of the project, practical implementation is given and design of the structure and function of each module are described overall. Third, detailed design and implementation of packet processing and transformation talble which bases on VLAN technology are discussed. Fourth, the main modules are simulated by the Modelsim software. Finally, The test platform for board-level testing are built and the testing problems are analysed. At present, the design has been verified by board level, indicating that the design is reasonable and meets design requirments.

Keywords: Packet Processing, VLAN, Layer-2 Transformation, AFDX

插图索引

表格索引

符号对照表

符号	符号名称
bit	位
ms	毫秒
ns	纳秒
us	微秒
byte	字节

缩略语对照表

缩略语	英文全称	中文对照
SEU	Single Event Upset	单粒子翻转
CMOS	Complementary Metal Oxide Semiconductor	互补金属氧化物半导体
SRAM	Static Random Access Memory	静态随机存取存储器
FPGA	Field Programmable Gate Array	现场可编程门阵列
MAC	Media Access Control	媒体访问控制
CRC	Cyclic Redundancy Check	循环冗余校验
IEEE	Institute of Electrical and Electronics Engineers	电机电子工程师学会
ATM	Asynchronous Transfer Mode	异步传输模式
AFDX	Avionics Full-duplex Ethernet	航空全双工以太网
ES	End System	端系统
VL	Virtual Link	虚链路
SN	sequence number	序列号
BAG	Bandwidth Allocation Gap	带宽分配间隔
VLAN	Virtual Local Area Network	虚拟局域网
RFC	Request For Comments	征求修正意见书
CPU	Central Processing Unit	中央处理器
MII	Medium Independent Interface	介质无关接口
FIFO	First Input First Output	先进先出队列
LAN	Local Area Network	局域网
TPID	Tag Protocol Identifier	标签协议标志
CFI	Canonical Format Indicator	规范格式指示器

目录

第一章　绪论

1.1 研究背景

近年来，我国的航空航天技术的发展迅速，达到了世界先进的水平。随着航天器中的电子系统复杂度的增加，也就代表着内部的控制越来越复杂，内部的通信也变的越来越复杂。具有良好的通用性及开方性的以太网技术在航天器中使用越来越广泛。

随着以太网的快速发展，其在越来越多的领域中得到了应用，除了普通的商用领域外，在工业控制，国防军工，航空航天等领域的应用也越来越多。但是应用于航天器中的以太网交换机与普通的商用交换机是有些不同的[1] 。由于空间站中存在的高能粒子辐射可能对采用 SRAM 工艺的 FPGA 产生较大的影响，例如，其内部配置存储器的逻辑状态可能受到粒子的撞击而造成状态的翻转，即单粒子翻转（SEU）[2] 。所以需要在普通的交换机的基础上增加可靠性的设计。针对 SEU 目前主要有两类方法：第一类为针对 SEU 的屏蔽，主要是各种抗辐射设计，高性能抗辐射 CMOS 工艺等。第二类为 SEU 恢复，主要包括 Xilinx 公司刷新，回读检测等。本方案中采用 xilinx Virtex 系列的宇航级 FPGA 来实现专用交换机来满足空间站中交换机的可靠性。

1.2 以太网交换简介

随着科学技术的进步及发展，作为通信网中的核心技术交换有了长足的发展。目前主要的交换有电路交换，分组交换，快速分组交换等。不同的交换技术各有其优缺点，为满足不同的需求需要选择不同的交换技术。例如电路交换的传输时延恒定，这一特点使它能够很好的支持恒定比特率的电话和视频业务。结合本项目本节主要介绍以太网交换。

20 世纪 90 年代以后，在局域网的市场上以太网脱颖而出，几乎成为了局域网的代名词。以以太网交换机为核心的交换式以太网可以明显的提高局域网的性能。交换机中的不同的端口可以平行，双向同时的传输数据，能够增加网络的带宽，减少冲突。交换式以太网的基本结构如图 1.1 所示：

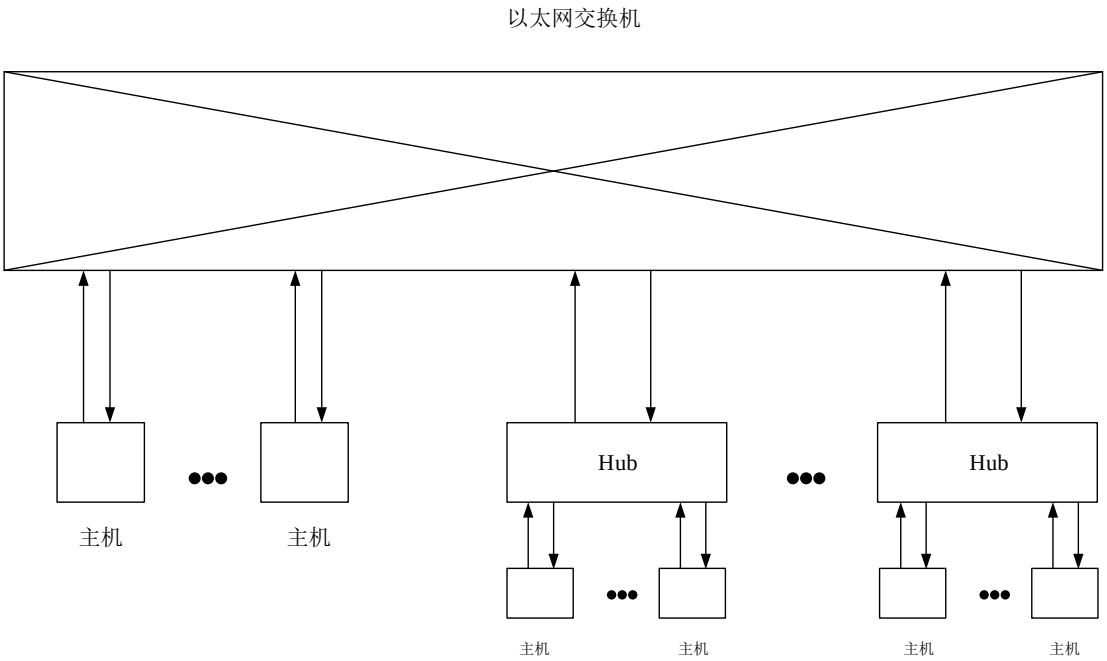

图1.1交换式以太网的基本结构

以太网交换的基本原理：

以太网交换是一种基于 MAC（Media Access Control ，介质访问控制）地址的识别，来完成数据帧的转发的，所以又称第二层交换。以太网交换机主要端口、内部交换网络和交换控制器等构成。端口及与之相连的内部交换网络构成了数据转发，交换控制器的作用是实现转发控制[3] 。结构如图 1.2 所示：

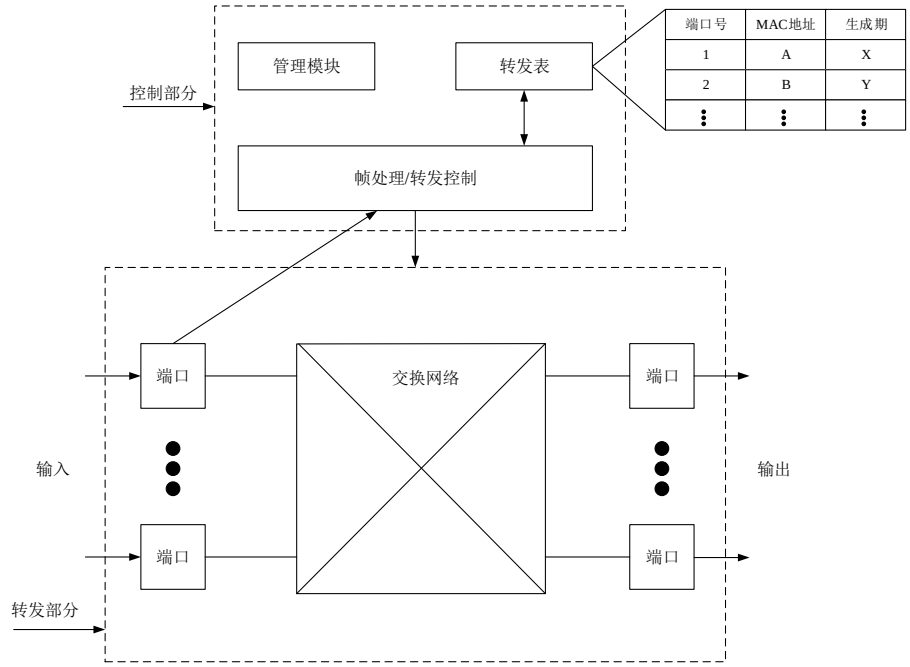

图1.2以太网交换机的基本结构

在以太网交换机中，为了实现数据帧的转发控制需要构建和维护一张转发表。转发表的建立是通过自学习来实现的，该表中表项的内容为端口号，MAC 地址，生存期计数器。交换机初始化时转发表中没有内容，此时有数据帧到来时，对数据帧进行广播，并根据源端口及源 MAC 进行表项的学习，为了允许网络拓扑结构的变化，每一个表项都配有生存期计时器。当某一表项中的计数器超时时，就将该表项删除。需要注意的是:自学习只能建立单播表，组播表需要交换机中的管理设备来配置表项。

以太网交换机中，帧的转发方式主要有以下三种：

（1） 直通转发：

交换机在接收到数据帧后，提取目的 MAC 地址，进行查找转发表，找到输出的端口号。然后转发控制模块开始建立输入输出端口之间的连接，将数据帧从输出端口转发出去，当数据帧转发完成后对此连接进行释放。直通转发的特点是：当数据帧的目的地址接收后，即可获知输出的端口，因此转发速度快，减小了转发时延。但是由于其不需要帧的接收完成就开始转发，所以不能对数据帧进行检错，而且不能实现速率的转换。当有多个端口的数据帧需要同时或间隔很小的向同一个端口转发，就会产生冲突，导致数据帧的丢弃。

（2） 存储转发：

当数据帧到来时，把整个数据帧缓存起来，然后对该数据帧进行格式检查并进行 CRC 校验，若有错误则把数据帧进行丢弃。若数据帧没有错误，将数据帧进行缓存，并提取源 MAC 地址、目的 MAC 地址及源端口号，然后根据源 MAC 地址及源端口进行学习，根据目的 MAC 地址进行地址转发表的查找，找到目的端口号后，将该数据帧送入到相应的缓存中。然后按一定的规则将各数据帧进行调度输出。存储转发的缺点是可能对数据帧的处理延时较大。优点是有很好的可靠性，而且支持不同端口之间的速度的转换，还可以解决交换网络内部和输出端口之间可能存在的冲突问题。本设计中采用的为存储转发模式。

（3） 无碎片转发：

无碎片转发方式基本和直通转发一样，只是比直通转发下接受了更多的帧信息后再进行转发，无碎片转发是在接收 64bytes 后才开始转发的，减少了转发出错的概率。

1.3 AFDX 技术简介

AFDX 是在 IEEE802.3 以太网协议的基础上发展确立的确定性网络，借用了交换式以太网技术及 ATM 技术，保障航天航空通信中的所需的可靠性及实时性。AFDX 网络的结构如图 1.3 所示：

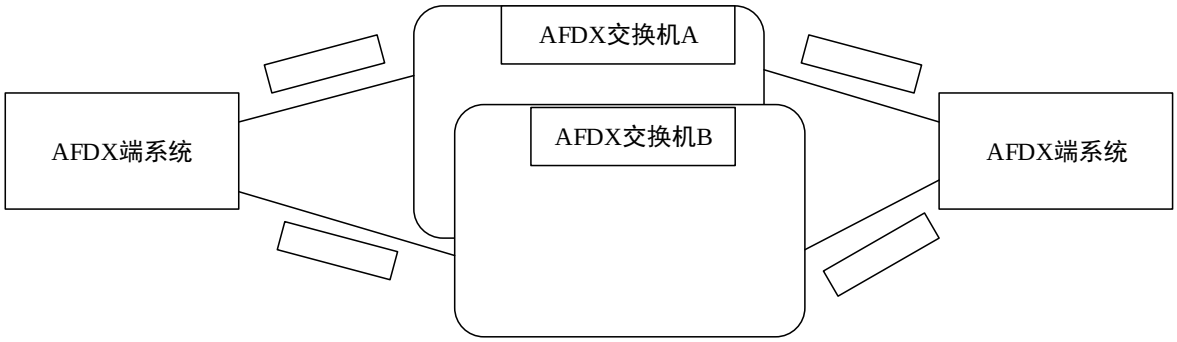

图1.3 AFDX 网络结构图

AFDX 网络主要包括端系统（End System, ES）及交换机两个部分[5] [6]，其中端系统主要负责将数据进行封装，然后将数据帧按照 VL 的要求将数据帧发送出去。AFDX 交换机主要是根据虚链路（Virtual Link ,VL）来进行数据警管及过滤，然后将数据帧从正确的端口转发出去。如图 1.3 所示 AFDX 网络中采用双交换机网络，端系统将同一数据帧发送两份分别给两个交换机，正常情况下一个端系统会收到两份数据帧，然后根据序列号对数据帧进行接收或丢弃，通过冗余备份保证了数据传输的可靠性。

AFDX 的帧格式：

AFDX 的帧格式与 IEEE802.3 以太网的帧格式基本相同，区别在于 AFDX 帧中有一个字节是用来表示序列号的以及一些字段赋予了不同的意义。AFDX 帧格式如表 1.1 所示：

表1.1AFDX 帧格式

Pre	SFD	DA	SA	Type	Data	SN	FCS
7 字节	1 字节	6 字节	6 字节	2 字节	45~1499 字节	1 字节	4 字节

目的地址（DA）：AFDX 帧中的目的地址包含 Constant 域和 VL_ID 号两个部分。Constant 类似于网络号，同一个端系统发送的数据帧的 Constant 域是相同的。Constant 域第一个字节中最后两个 bit 值为 11，分别表示该地址支持单播和多播。VL_ID 为表明该数据帧所属的 VL。目的地址的结构如图 1.4 所示：

目的MAC地址 48bits	
Constant域 (32 bits)	VL_ID (16 bits)
xxxx xx11 xxxx xxxx xxxx xxxx xxxx xxxx	xxxx xxxx xxxx xxxx

图1.4 AFDX 目的地址结构

源地址（SA）：AFDX 源地址的结构如图 1.5 所示：

源MAC地址 48bits						
Constant field (24bits)	Network_ID (8 bits)		Equipment_ID (8 bits)		Interface_ID (3 bits)	Constant 域 (5 bits)
	Constant域 (4-bits)	4-bits	3-bits	5-bits		
0000 0010 0000 0000 0000 0000 0000 0000	0000	xxxx	xxx	xxxxx	xxx	00000

图1.5 AFDX 源地址结构

其中，由于源地址为单播地址，因此其 Constant 域第一个字节中最后两个 bit 值为 10。Interface_ID 值为 3’b001 时，表明该数据帧发往 A 网络。若值为 3’b010 时，表明该数据帧发往 B 网络。

序列号（sequence number, SN）：序列号的值从 0~255，端系统复位后的每个 VL 发送的第 1 帧的序列号为 0，之后依次加 1，当加到 255 时，下一次为序列号的值为 1[9] 。

1.4 AFDX 的 QoS 保障

(1) 虚链路（VL）：VL 是在源与目的端系统之间建立连接，每条 VL 的带宽是可以配置的。数据帧所属的 VL 是在该帧中的目的 MAC 中的低 16 位标识的。本设计中采用低 12 位来标识数据帧所属的 VL。AFDX 交换机最多可以支持 4096 条 VL，每个端口最多支持 256 条 VL。

(2) 带宽分配间隔（Bandwidth Allocation Gap, BAG）：BAG 是用来对与之相对应的 VL 进行带宽限制的。同一条 VL 的两个数据帧之间的最小间隔为 BAG。BAG 的取值为 1ms、2ms、4ms、8ms、16ms、32ms、64ms 及 128ms。

(3) 抖动（Jitter）：Jitter 的定义为每个 BAG 的开始处到数据帧开始传输之间的长度。对于一条 VL 来说，当其 BAG 及 Jitter 确定后，就确定了一个最小带宽。若数据帧开始传输时的时间大于 Jitter 则将该数据帧丢弃[7] [8] 。

1.5 本文的主要工作

本文结合实验室承担的科研项目——“航天专用分组交换机的设计与实现”，首先，对专用交换机的整体设计需求及性能需求进行介绍，然后给出整体的设计结构并对各个模块进行简单的介绍，之后重点介绍分组处理，转发表模块的设计及实现。最后，根据实验室现有的设备搭建测试平台，进行板级测试并对测试中出现的问题进行分析和总结。文章的内容安排如下：

第一章：绪论。主要介绍了本文的选题背景，对以太网交换的原理及结构进行介绍，最后对 AFDX 技术及相关的概念进行介绍。

第二章：专用分组交换机的总体设计。首先，对项目的需求进行简单的介绍，然后给出整体的设计结构框图，最后对各个子模块进行介绍并对数据的处理流程进行介绍。

第三章：专用分组交换机的分组处理模块的设计及实现。首先，对分组处理模块的功能进行分析，然后给出分组处理模块的整体设计以及该模块的具体实现。最后对分组处理模块的主要功能进行仿真。

第四章：专用分组交换机转发表模块的设计及实现。首先，对 vlan 的概念进行简单介绍，给出 vlan 表转发模块的总体设计以及具体模块的实现。然后对 vl 转发表及警管模块的具体实现进行介绍。最后给出转发表模块的仿真结果。

第五章：专用分组交换机板级测试。首先，对测试环境进行介绍，然后，给出测试的结果，最后对测试中出现的问题进行分析和总结。

第二章 专用交换机的总体设计

本章结合实验室承担的项目“航天专用分组交换机的设计与实现”，介绍分组交换机的设计需求，并根据需求给出整体的设计方案，对主要模块的功能进行介绍并给出数据的处理流程。

2.1 设计需求

- 以太网数据格式采用 IEEE Std 802.3TM-2005 中 MAC 帧格式，不使用 LLC 子层；
- VLAN 标记帧格式符合 IEEE 802.1Q 标准；
- 上层数据符合 RFC 791 标准。

物理接口满足的标准：

- 物理接口采用 10/100Mbps 自适应电接口；
- 10/100Mbps 自适应电接口符合 IEEE 802.3、802.3u 中 10BASE-T 和 100BASE-TX 的规定；
- 物理和电气特性满足 IEEE 802.3、ANSI X.263 中 10BASE-T 和 100BASE-TX 的规定；
- 10/100Mbps 自协商机制应满足 IEEE 802.3 中的规定。

功能及性能要求：

- 接口数量：需提供不少于 10 个 10/100Mbps 自适应电接口；
- 接口工作模式：采用全双工模式进行数据交换；
- 支持 VLAN；
- 吞吐量：按 10 个接口计算，满负荷时流量为 1Gbps，交换机需保证满负荷时能达到线速转发；
- 丢包率：10 个接口中有 8 个接口同时工作，即数据流量为 800Mbps 时，应无丢包；
- MAC 地址表：深度不少于 1k。本方案只支持二层查找，不提供三层查找功能；
- 可动态配置各数据队列的最大和最小存储空间；
- 队列优先级：每个接口支持 8 个优先级队列。

根据对设计需求的分析，采用 MAC IP 来实现全双工的 10/100M 以太网接口，MAC 地址表是通过 RAM 和 hash 来实现，用两级查找来实现 VLAN 下的数据帧的转

发具体实现见第四章。

在考虑到本交换机应用的环境为空间站，空间站中的网络结构比较稳定以及AFDX与以太网的相似性。本设计方案在满足项目的设计需求的基础上，同时实现了对AFDX的支持。

2.2 总体设计结构

该分组交换机配置12个标准以太网接口，其中每个接口速率为100M，并且配备有CPU接口，可以实现CPU对接口数据帧队列进行插入和捕获，同时具备vlan功能以及对AFDX的支持。

本方案采用航天级芯片实现设计，具备防辐射等特点，具有较高的抗干扰性，适合应用于空间环境下的以太网交换机的可靠性研究。同时，考虑到空间环境下的需求，该方案采用刷新技术定期对FPGA实行刷新，保证数据传输的高可靠性。考虑可获取性和可替代性，本方案选用XILINX 公司的XQ4VXS55器件用来完成实现。考虑到XQ4VXS55中的逻辑资源为24576个slice存储资源为320*18KBRAM，我们采用4块FPGA来完成设计。总体设计结构如图2.1所示：

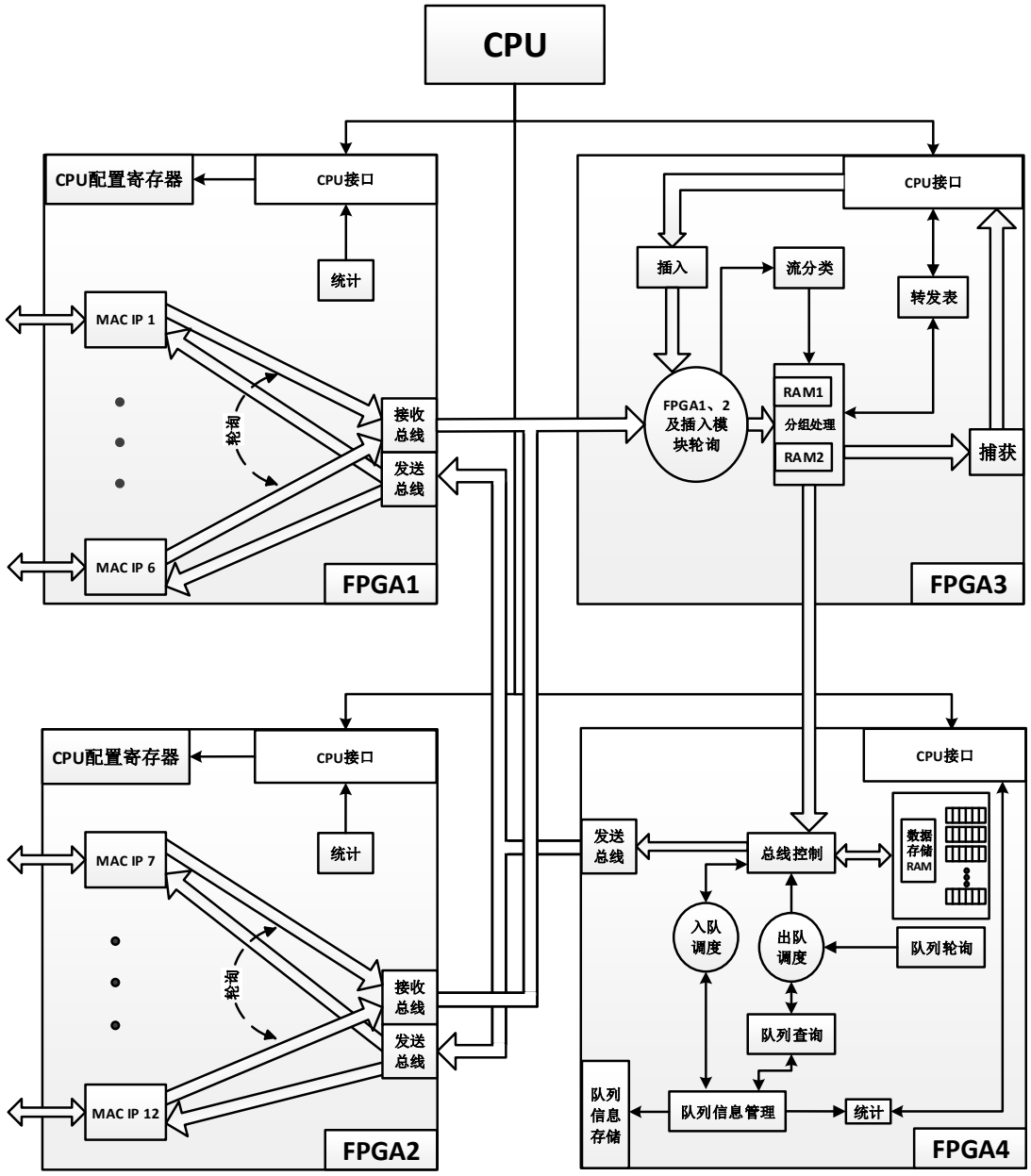

图2.1 分组交换机的整体设计结构

4 块 FPGA 中存储资源的使用情况如表 2.1 所示：

表2.1 4 块 FPGA 存储资源使用情况

	FPGA 描述	所需 BRAM 个数（18kb）	XQ4VXS55 中的 BRAM 个数 (18kb)	占用率（共 320 个）
FPGA1	6 个 MAC IP 、接收调度模块	12	320	3.8%

FPGA2	6 个 MAC IP 、接收调度模块	12	320	3.8%
FPGA3	接收调度模块 、分组处理模块、流分类模块、转发表模块（包括以太网帧的转发模块及 AFDX 帧的转发模块），捕获模块及插入模块	144	320	45%
FPGA4	总调度、队列管理、缓存分配、发送调度、总线控制、CPU 捕获模块、以及共享缓存	242	320	76%

由于片间接口发送与接收的相似性以及 FPGA1,3 与 FPGA2,3 之间接口的相似性，本文仅介绍 FPAG1,3、FPGA3,4 之间的接口。FPGA1 与 FPGA3 之间的接口如图 2.2 所示：

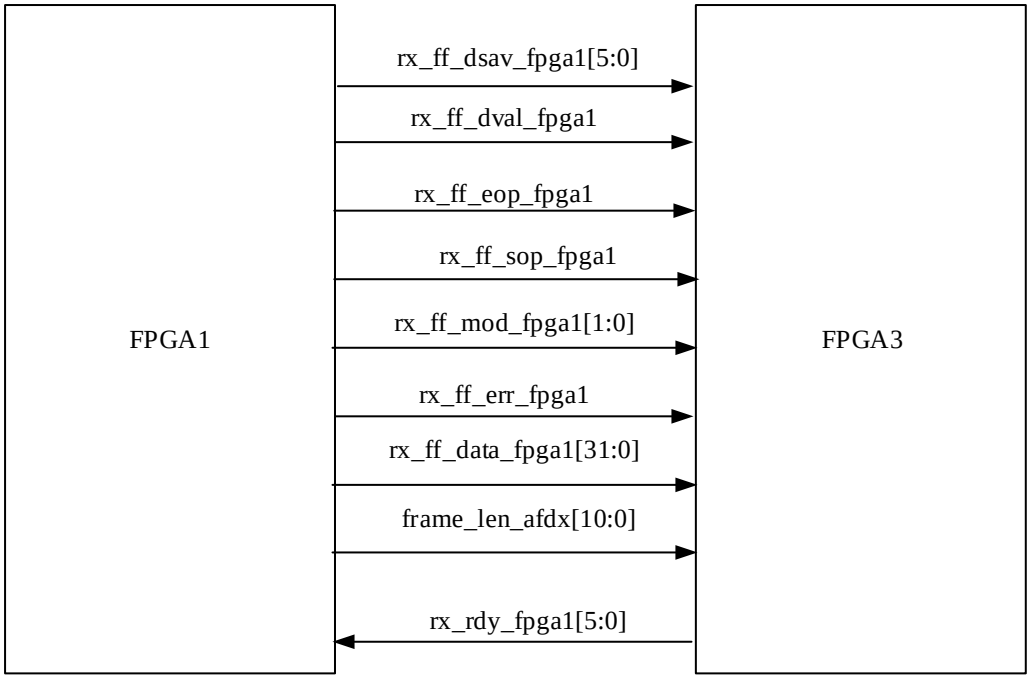

图2.2FPGA1,3 之间的接口

管脚的定义如表 2.2 所示：

表2.2 FPGA1,3 之间的管脚定义

命名	位宽（bit）	来源	说明
rx_ff_dsav_fpga1	6	FPGA 1	位宽为六对应于六个 MAC IP，若相应的位为 1 则表明该端口有数据帧
rx_ff_dval_fpga1	1	FPGA 1	数据有效信号，在整个数据帧传输的过程置高
rx_ff_eop_fpga1	1	FPGA 1	数据结束标志位，持续一个周期，高有效
rx_ff_sop_fpga1	1	FPGA 1	数据帧开始标志位，持续一个周期，高有效
rx_ff_mod_fpga1	2	FPGA 1	指示数据帧的最后一个传输周期中的有效的数据。00 表示[31:0]都是有效位，01 表示[31:24]是有效位，10 表示[31：16]是有效位，11 表示[31:8]是有效位。
rx_ff_err_fpga1	1	FPGA 1	错误帧标志
rx_ff_data_fpga1	32	FPGA 1	传输数据
frame_len_afdx	11	FPGA 1	数据帧长度
rx_rdy_fpga1	6	FPGA 3	表明分组处理模块能接受相应 MAC IP 中的数据

FPGA 3 与 FPGA 4 之间的接口如图 2.3 所示：

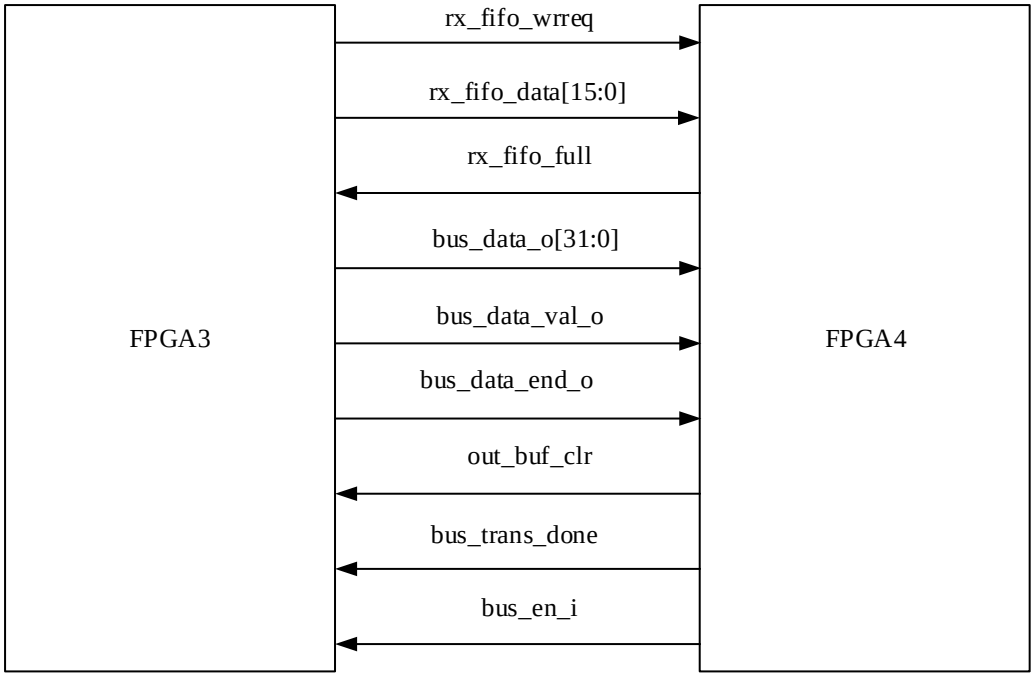

图2.3 FPGA 3 与 FPGA 4 之间的接口

FPGA 3 与 FPGA 4 之间的管脚定义如表 2.3 所示：

表2.3 FPGA 3 与 FPGA 4 之间的管脚定义

命名	位宽	来源	说明
rx_fifo_wrreq	1	FPGA 3	使能信号，高有效时将帧信息写入 rx_fifo 中
rx_fifo_data	16	FPGA 3	传输数据
rx_fifo_full	1	FPGA 4	rx_fifo 的满信号，高有效
bus_data_o	32	FPGA 3	传输数据
bus_data_val_o	1	FPGA 3	数据有效信号，高有效，持续整个数据的传输过程
bus_data_end_o	1	FPGA 3	数据传输完成信号，高有效
out_buf_clr	1	FPGA 4	清空信号，高有效

bus_trans_done	1	FPGA 4	数据接收完成信号
bus_en_i	1	FPGA 4	总线搬移信号

2.3 模块功能介绍

在整个数据的处理流程中，大致将系统分为如下模块：MAC IP 模块、接收调度模块、分组处理模块、流分类模块、转发表模块、总线控制模块、总调度模块、队列管理模块、发送轮询模块、发送调度模块、此外，还包含有插入模块、捕获模块、统计模块、CPU 配置模块等。

- MAC IP 模块

MAC IP 负责数据帧的接收和发送，MAC IP 能够实现 10/100/1000 兆以太网接入控制功能，本设计中采用 100 兆全双工模式。在本设计中使用的 MAC IP 功能主要是以下几点：

- 在接收时对数据帧进行 CRC 校验（也可选择不进行校验）。
- 通过一个 4bit 的 MII 接口（工作频率为 25MHZ），实现与 100 Mbps 的以太网 PHY 的无缝连接。
- 在发送时为数据帧加上前导码 Preamble 和帧起始标志 SFD，前导码为七个字节的 0x55，帧起始标志一个字节值为 0xd5。
- 在发送时为数据帧添加 CRC。
- 对于长度小于 64 字节的数据帧进行填充，保证从数据帧的长度在 64 字节以上。
- 在接收或发送时实现 32 位的数据位宽与 4 位的数据位宽的转换。

MAC IP 的引脚如图 2.4 所示：

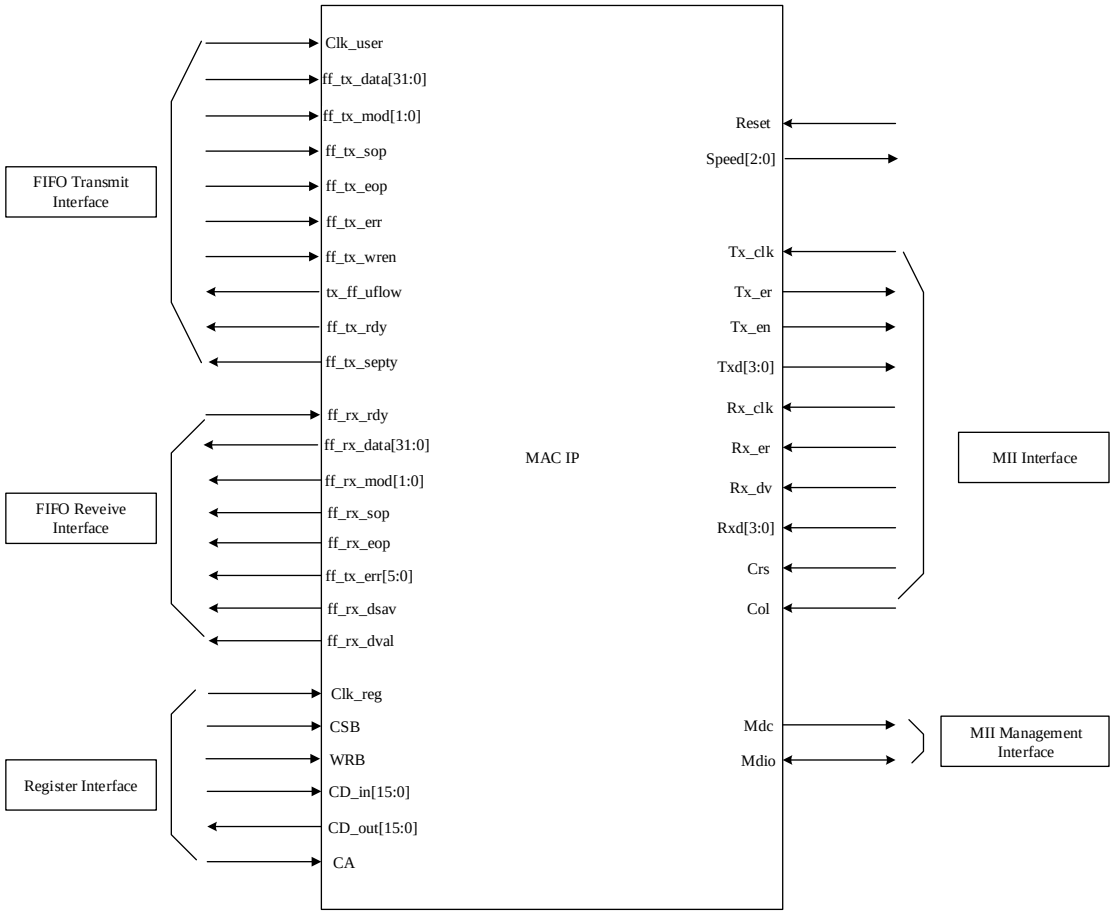

图2.4 MAC IP 接口

➢ 接收调度模块

接收调度模块的主要功能是公平的轮询 12 个 MAC IP 及插入模块，若所轮询的端口有数据帧（ff_rx_dsav 为 1 表明有数据帧要传送），则接收数据并将数据传给分组处理模块及流分类模块。本设计中采用的是两级轮询。第一级轮询为片内 MAC IP 轮询，第二级为片间轮询。由于两级轮询的相似性，下面仅介绍 FPGA 2 中的轮询模块。图 2.5 为轮询模块中的状态转移图（由于该状态机状态较多，若状态转移的条件在图中标注后不易观看，因此未在图中标出状态转移条件。）：

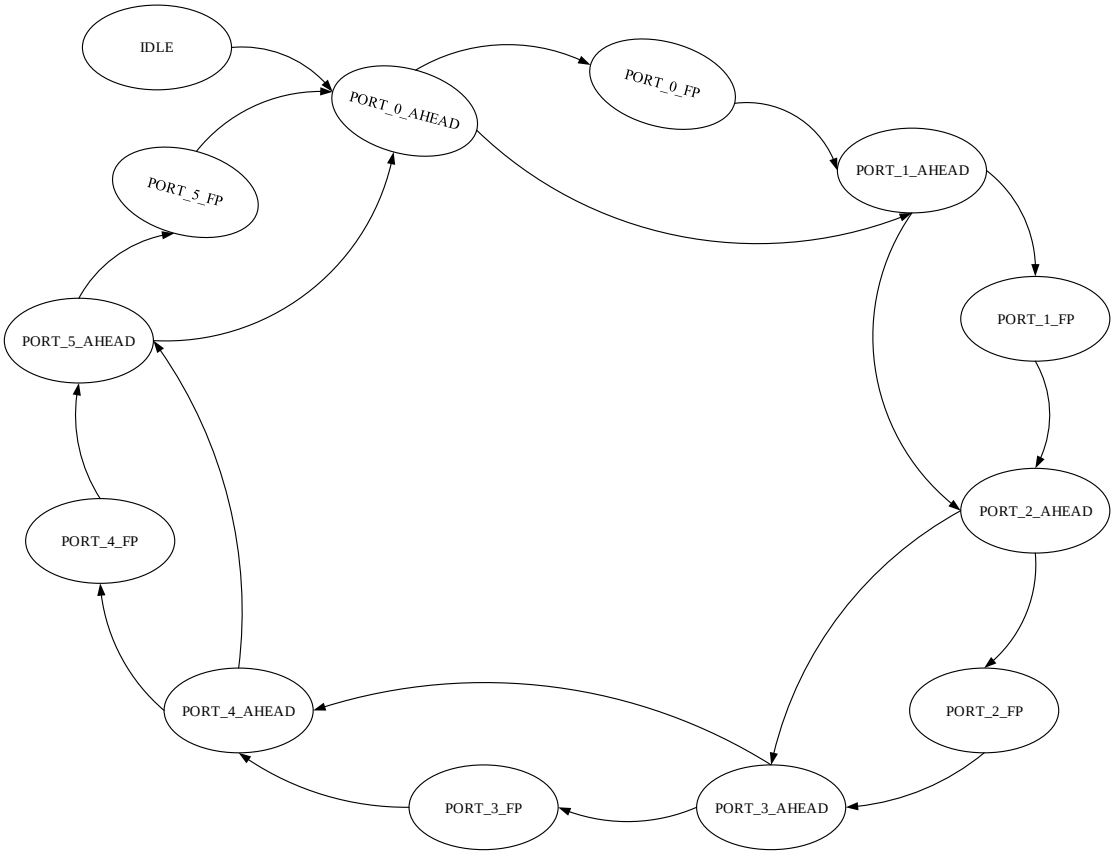

图2.5 轮询模块状态转移图

当复位后状态机进入 IDLE 状态，之后开始对第 1 个 MAC IP 进行查询是否有数据传输，若没有数据要传输则查询第 2 个 MAC IP。若有则将数据传递给分组处理模块，当数据传递完后，查询第 2 个 MAC IP 是否有数据要传输。之后重复之前的操作。当第 6 个 MAC IP 查询后则查询第 1 个 MAC IP。然后依次对 MAC IP 进行查询。

- 分组处理模块

分组处理模块主要是对接收调度模块传输来的数据帧进行缓存，对流分类模块产生的指令码进行分析并执行。将提取的帧头信息给转发表模块进行查找并在获得查找结果后将调度信息传给队列管理模块。在收到总线控制模块的搬移指令后将数据帧从乒乓缓存中搬移出来。分组处理模块的具体实现在本文的第 3 章给出了介绍。

- 流分类模块

流分类模块主要是通过对数据帧中的信息进行提取及识别，然后与规则表中的规则进行匹配，若匹配成功则产生相应的指令给分组处理模块。涉及的指令主要有数据帧丢弃、队列优先级的划分、捕获帧等[11] 。

流分类模块的接口如图 2.6 所示：

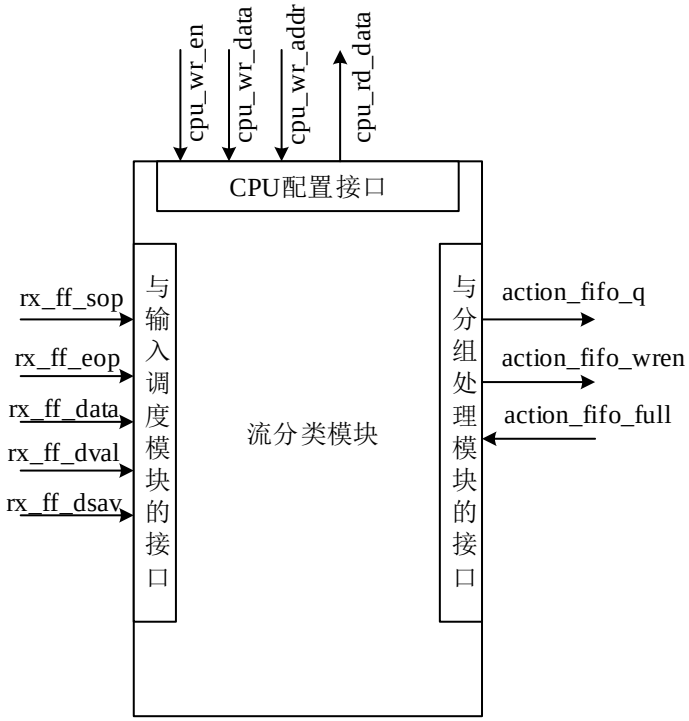

图2.6 流分类模块的接口

流分类模块的对外管脚信号如表 2.4：

表2.4 流分类模块管脚定义

序号	名称	输入或输出	来源或去向	位宽	说明
1	rx_ff_sop	input	输入调度模块	1	数据帧开始标志，持续一个周期，高有效
2	rx_ff_eop	input	输入调度模块	1	数据帧结束标志，持续一个周期，高有效
3	rx_ff_data	input	输入调度模块	32	数据帧数据
4	rx_ff_dval	input	输入调度模块	1	数据帧有效信号 持续整个数据帧的传递过程，高有效
5	rx_ff_dsav	input	输入调度模块	1	数据帧有效信号，高有效，比 rx_ff_dval 信号提取两个周期
6	action_fifo_q	output	分组处理模块	80	指令码
7	action_fifo_wren	output	分组处理模块	1	写使能，高有效
8	action_fifo_full	input	分组处理模块	1	action_fifo 满信号，高为满

					信号,若不满则可以输出指令码
9	cpu_rd_data	output	CPU 接口模块	32	CPU 读数据
10	cpu_wr_addr	input	CPU 接口模块	16	CPU 读写数据的地址
11	cpu_wr_data	input	CPU 接口模块	32	CPU 写数据
12	cpu_wr_en	input	CPU 接口模块	1	CPU 写数据使能，高有效

流分类模块的对外接口主要是与输入调度的接口、与分组处理模块的接口以及与 CPU 配置模块的接口。其中，CPU 配置模块的接口主要是用来配置规则表以及字段列表的。与输入调度的接口主要是来获取用来与规则表中的条款相匹配的数据帧的帧信息。与分组处理模块的接口是来传递指令码给分组处理模块。

- 转发表模块

转发表模块主要包括用来对以太网数据帧目的端口查找的单播转发表、组播转发表及 vlan 转发表，用于对 AFDX 帧进行转发的 VL 转发表及警管过滤模块。其中单播表是自学习来维护，组播表及 vlan 表是 CPU 来维护，VL 转发表及相应的警管也是同过 CPU 来维护的。对于以太网帧来说转发表的功能是根据源端口号，源 MAC 地址，VLAN ID 来获得该数据帧转发出去时的端口号。对于 AFDX 帧来说是根据目的 MAC 中的 VL 来对数据帧进行过滤及警管，并查找该数据帧的转发出去时的端口。该模块的具体的实现在第 4 章来介绍。

- 总线控制模块

该模块的功能是接受总线控制实现从分组处理到共享缓之间存的数据搬移，发送总线控制实现从共享缓存到发送接口 MAC IP 之间的数据搬移，总线的搬移操作是根据调度模块生成的调度信息执行的。

- 入队总调度模块

该模块的功能是协调总线控制、队列管理以及缓存管理之间的各项工作，根据接收到的数据帧信息向向缓存管理、队列管理等模块发出请求，实现缓存地址分配、逻辑入队等操作，同时生成接收调度信息，指示总线控制模块的搬移操作。

- 队列管理

队列管理模块采用共享缓存的方法存储数据帧，并在共享缓存中，根据数据帧目的接口和优先级构造相应的虚拟输出队列 VOQ，帧信息管理模块会对数据帧进行逻辑出入队管理。此外，由于本系统使用定长 buffer 缓存使用策略，缓存管理模块会根据入队数据帧长度执行缓存空间分配操作，根据出队数据帧长度执行缓存空间释放操作，维护表示存储 buffer 使用情况的 bitmap 表。出入队操作是由队列发起的，所以实现时应当保证队列之间调度算法的公平高效，同时保证数据帧优先级也得到满足。

CPU 的配置往往也是以队列为单位的，例如队列门限控制等。

- 发送轮询

该模块的功能是根据队列轮询的调度算法，生成下一数据帧所在队列号。

- 发送调度

该模块的功能是根据出队帧队列号，配合帧信息查询模块和缓存查询模块，定位出队帧所在位置，生成发送调度信息，指示发送总线搬移操作。

- 插入模块

当 CPU 有数据帧下发到交换机时，交换机的处理是将插入模块当作一个普通的以太网端口来进行轮询。插入模块在将数据传递到分组处理的同时也将队列优先级以及目的端口号，帧长等信息传递给分组处理。因此插入帧与普通帧的区别在于插入帧不需要进行流分类也不需要通过转发表来查找目的端口。该模块的结构图如图 2.7 所示：

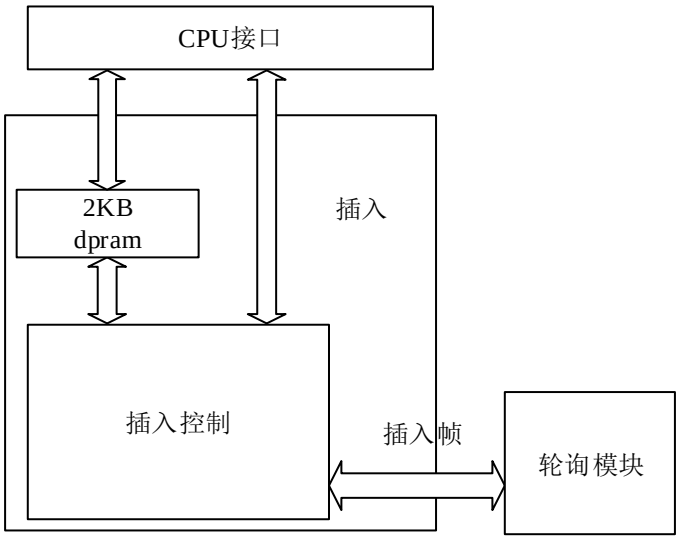

图2.7 插入模块的结构图

- 捕获模块

捕获模块的功能是用来将需要上交给 CPU 的数据帧交给 CPU。数据帧是否上交到 CPU 由流分类模块来判断。有两种数据帧需要上传给 CPU。（1） 重定向到 CPU 的数据帧，即只需要上传到 CPU，不需入队转发。（2） 复制到 CPU 的数据帧，即该数据帧既需要上传到 CPU，又需要入队转发。捕获模块的结构图如图 2.8 所示：

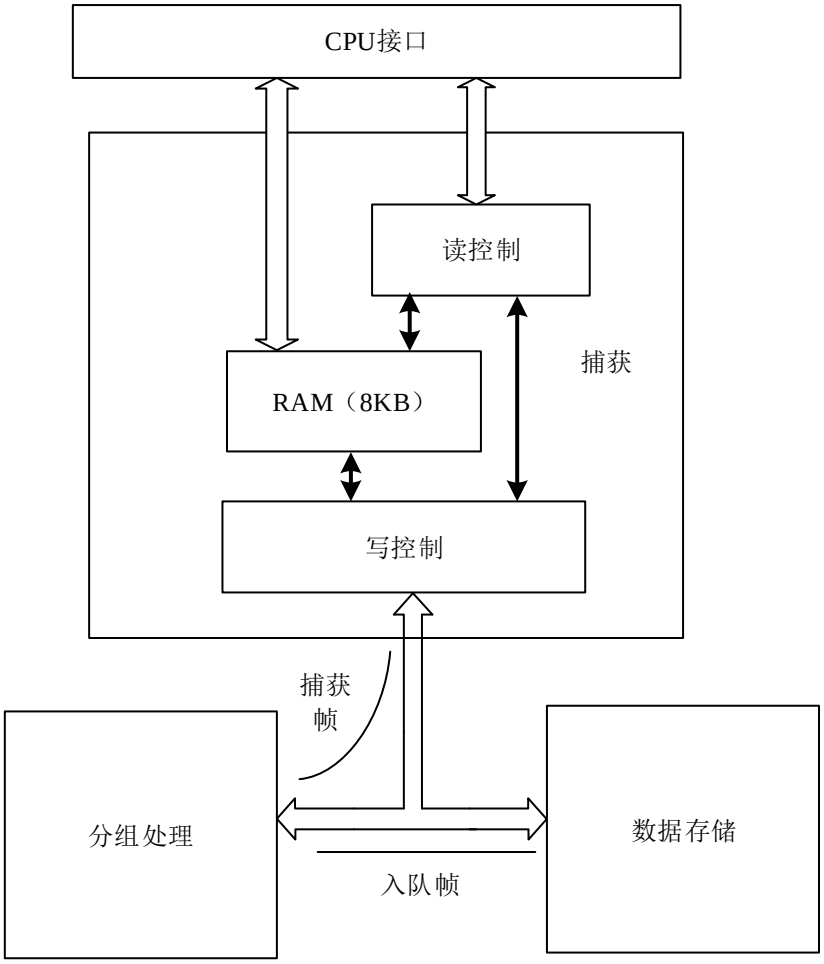

图2.8 捕获模块的结构

- 统计模块

统计的主要作用是为了使 CPU 了解交换机中对数据帧的处理情况，需要对数据帧的处理情况进行统计。例如，在 MAC IP 处，单位时间内接收的数据帧中 CRC 校验出错的数据帧的个数。

- CPU 配置模块

配置模块的功能主要是对一些寄存器或缓存进行配置使专用交换机具有动态灵活可配性。例如：CPU 对流分类中的规则表的动态配置，则可以根据用户的需求来对数据帧进行区分，执行不同的操作。

2.4 数据的处理流程

下面对数据帧从进入专用交换机到出来的整个流程进行简单的介绍。

2.4.1 对以太网数据帧的处理流程

- 接收处理过程

- MAC IP 通过 MII 接口与 PHY 相连接进行以太网帧的接收。如果 MAC 在接收帧时出现了错误例如帧长度不符合要求（带 vlan 标记的帧长的有效范围为 68 字节到 1522 字节，非 vlan 标记的帧长的有效范围为 64 字节到 1518 字节）、CRC 校验出错、帧间隔不满足要求等等，MAC 会自动丢弃该数据帧，通过 rx_err[0] 向内部指示在接收帧时出现了错误并把帧丢弃了。当 MAC 已经把一个完整的合格的以太网数据帧接收到 MAC 内部的 FIFO 后，将 ff_rx_dsav 置高。

- 调度模块轮询各接口及插入模块，若当前轮询到的接口的 ff_rx_dsav 为 1 或当前轮询到的插入模块非空，则当分组处理的乒乓 FIFO 有空间时将该接口或插入模块的数据帧送入分组处理处理模块及流分类。

- 流分类模块按照规则表将以太网帧的相应的字段提取出来并进行规则的匹配得到需要进行的操作例如丢弃、捕获到 CPU、队列优先级等等。流分类模块以指令码的形式将需要的操作发给分组处理模块，分组处理模块来执行这些操作。

- 分组处理模块执行流分类模块产生的指令码根据指令码判断该帧是否要丢弃，若不是则判断是否要重定向，若是则通知 CPU 捕获模块，否则的话将数据帧中的源 MAC 地址、源端口号，目的 MAC 地址、帧长、是否是带 vlan 标记、VLAN ID 等信息发送给转发表模块。查找转发表后分组处理模块会将输出端口号、优先级、帧长等信息写入 rx_fifo 中，然后等待总调度模块搬移数据并在搬移的过程中进行数据帧的插入、删除、修改操作。另外，若正在处理的数据帧为监控帧，则生成以输入监控端口号为目的端口号的信息写入 rx_fifo 中，并在数据帧的搬移过程中不对数据帧做改变。另若数据帧为插入帧，目的端口由插入模块提供，因此不需要查找目的端口。

- 转发表模块根据分组处理送来的信息查找 vlan 表判断该帧是否允许进入，若不被允许进入（不被允许进入的以太网帧是其 vlan ID 与端口所属的 vlan 不同）则丢弃，否则根据源端口号及源 MAC 地址对地址表进行学习及更新，根据目的 MAC 地址来查找目的端口号，然后再进行 vlan 表查询判断目的端口所在的 vlan 是否与源端口所在的 vlan 相同，若相同则判断目的端口在该 vlan 下是 untag 还是 tag，若是 tag 则打上该 vlan 标记否则不打 vlan 标记。若在地址表查询获得目的端口号失败则在 vlan 内广播。

- 入队调度查询 rx_fifo 是否为空，若不为空，则读出队首信息，进行缓存分配/释放和入队处理；

- 缓存分配/释放模块接收到入队调度发来的缓存申请请求，查询该队列是否拥有足够的存储空间，若足够，则分配指定数目的 BD 块，返回 BD 块起始地址，更新 BD 块使用信息；若空间不足，返回申请失败信息；

- 入队调度获得反馈信息，若失败，则向接收信息缓存发出清空信号，若申请成功，则向队列管理模块发出请求，同时生成 sr_rx_fifo 信息入队；
- 队列管理模块接收到入队请求，更新各队列信息、接口信息，进行入队处理；
- 总线控制模块访问 sr_rx_fifo，根据分配的地址按照 BD 为单位，将分组处理缓存中的数据搬移到指定的存储地址（此时的数据帧是将已修改的帧头和原始剩余数据进行合并之后的数据帧）；

➢ 发送处理过程

- 发送轮询模块查看对应队列是否有数据传送，若没有，则轮询下一接口；否则，获取当前接口非空的最高优先级的队列，生成 tx_fifo 信息；
- 发送调度访问 tx_fifo，查询指定队列信息，获取第一个队列帧的存储地址，向缓存查询模块发出查询请求，并将每一个查询到的缓存地址写入 sr_tx_fifo，查询结束后，将需要出队的以太网帧写入 releaseEmacNodeFifo，准备用于出队处理；
- 出队调度模块获取出队帧信息，向队列管理模块发出出队请求，队列管理模块进行出队处理，并将出队帧信息写入 releaseAddrFifo，用于缓存释放；
- 缓存分配/释放模块读取 releaseAddrFifo 信息，获得该数据帧起始存储地址，并查询 linkedRAM，逐一释放 BD 存储地址，更新 BD 使用情况 bitmap 表；
- 同时总线控制模块读取 sr_tx_fifo 数据，若制定接口的发送缓存为空，则将指定地址上的数据帧搬移到接口发送缓存等待发送；
- MAC IP 对缓存中的数据进行发送。在发送时对数据帧进行 CRC 校验并在发送时为帧加上前导码和帧起始标志 SFD，前导码为七个字节的 0x55，SFD 为 0xd5。对于长度不足的帧会产生填充部分，使帧长在 64 字节以上。

2.4.2 对 AFDX 数据帧的处理流程

由于对 AFDX 数据帧的发送处理基本相同，因此，本文只介绍接收处理部分。

- MAC IP 通过 MII 接口与 PHY 相连接进行数据帧的接收。如果 MAC 在接收帧时出现了错误例如帧长度不符合要求（帧长的有效范围为 64 字节到 1518 字节）、CRC 校验出错、帧间隔不满足要求等等，MAC 会自动丢弃该数据帧，通过 rx_err[0]向内部指示在接收帧时出现了错误并把帧丢弃了。当 MAC 已经把一个完整的合格的以太网数据帧接收到 MAC 内部的 FIFO 后，将 ff_rx_dsav 置高。
- 调度模块轮询各接口，若当前轮询到的接口的 ff_rx_dsav 为 1，则当分组处理的乒乓 FIFO 有空间时将该接口的数据帧送入分组处理处理模块。
- 分组处理模块负责对接收的数据帧进行缓存并在缓存的同时对数据帧

的目的 MAC 地址进行识别，若目的 MAC 地址中的第一个字节中最后两个 bit 值为 11，则认为收到的数据帧为 AFDX 帧。将数据帧的目的地址中的 VL 值和 Constant 域以及数据帧长度信息传递给 VL 转发模块。

- VL 转发模块根据 VL 来查找获得对该数据帧进行的过滤及警管操作。并将最终的结果发送给分组处理模块。
- 分组处理模块分析 VL 转发模块传来的结果，若为丢弃，则将该数据帧丢弃，若查找成功，则将帧长，目的端口号，优先级等信息写入 rx_fifo 中。
- 入队调度查询 rx_fifo 是否为空，若不为空，则读出队首信息，进行缓存分配/释放和入队处理。
- 缓存分配/释放模块接收到入队调度发来的缓存申请请求，查询该队列是否拥有足够的存储空间，若足够，则分配指定数目的 BD 块，返回 BD 块起始地址，更新 BD 块使用信息；若空间不足，返回申请失败信息。
- 入队调度获得反馈信息，若失败，则向接收信息缓存发出清空信号，若申请成功，则向队列管理模块发出请求，同时生成 sr_rx_fifo 信息入队。
- 队列管理模块接收到入队请求，更新各队列信息、接口信息，进行入队处理。
- 总线控制模块访问 sr_rx_fifo，根据分配的地址按照 BD 为单位，将分组处理缓存中的数据搬移到指定的存储地址。

第三章　专用交换机分组处理模块的设计及实现

3.1 分组处理模块的功能分析

分组处理模块是本设计方案中的重要的功能模块，连接着接收调度模块、流分类模块、转发表模块、捕获模块、队列管理模块、总线控制模块。通过与不同模块之间的数据交互，使数据帧得到有序正确的处理。本章对分组处理模块的实现进行详细的介绍。

分组处理模块的主要的功能如下：

- 当乒乓缓存中能够接收数据时，从接收调度模块中正确快速的接受数据帧，并将数据帧存入乒乓缓存中。
- 分析并执行流分类模块给出的指令码，根据指令码中的信息对数据帧进行相应的操作。例如，丢弃、捕获到 CPU，队列优先级等。
- 在接收数据帧的同时对数据帧的帧头信息进行提取，根据目的 MAC 地址来判断该数据帧是以太网数据帧还是 AFDX 数据帧。若为 AFDX 帧，则将该数据帧的 VL 值及 Constant 域传递给 VL 转发模块。并根据 VL 转发模块的查找结果，生成帧信息写入 rx_fifo 中。若为以太网帧则主要提取包括源 MAC、目的 MAC、源端口号、VLAN ID 、插入帧等信息。并将提取的信息传递该转发表模块进行查找，给出该数据帧的目的端口号以及转发出去时是否为 TAG 帧等信息。若为插入帧时，不需要进行查找。分组处理模块将数据帧的信息写入 rx_fifo 中。
- 若分组处理模块判断出数据帧为输入监控帧时，会生成目的端口为输入监控端口的帧信息，并将该信息写入 rx_fifo 中。另监控帧在从乒乓缓存中搬移出去时不做任何的改变。若为捕获帧时，在捕获模块非满的情况下将数据帧从乒乓缓存中搬移到捕获模块中若捕获模块满时则丢弃该数据帧。若为非捕获帧时总线控制模块将乒乓缓存中的数据帧搬移出来并在搬移的过程中实现数据帧的插入修改删除操作。

3.2 分组处理模块的总体设计

分组处理模块的对外的主要接口有和输入调度模块的接口、和转发表模块的接口（转发表模块的接口包括图 3.1 中的与 VLAN 转发表的接口、与单组播的接口以及与 VL 转发模块的接口）、和总线控制模块的接口、和 CPU 模块的接口、和队列管理模块的接口以及和流分类模块的接口等。分组处理对外的主要接口（注：与 CPU 模块

的接口未在图中标注）如图 3.1 所示：

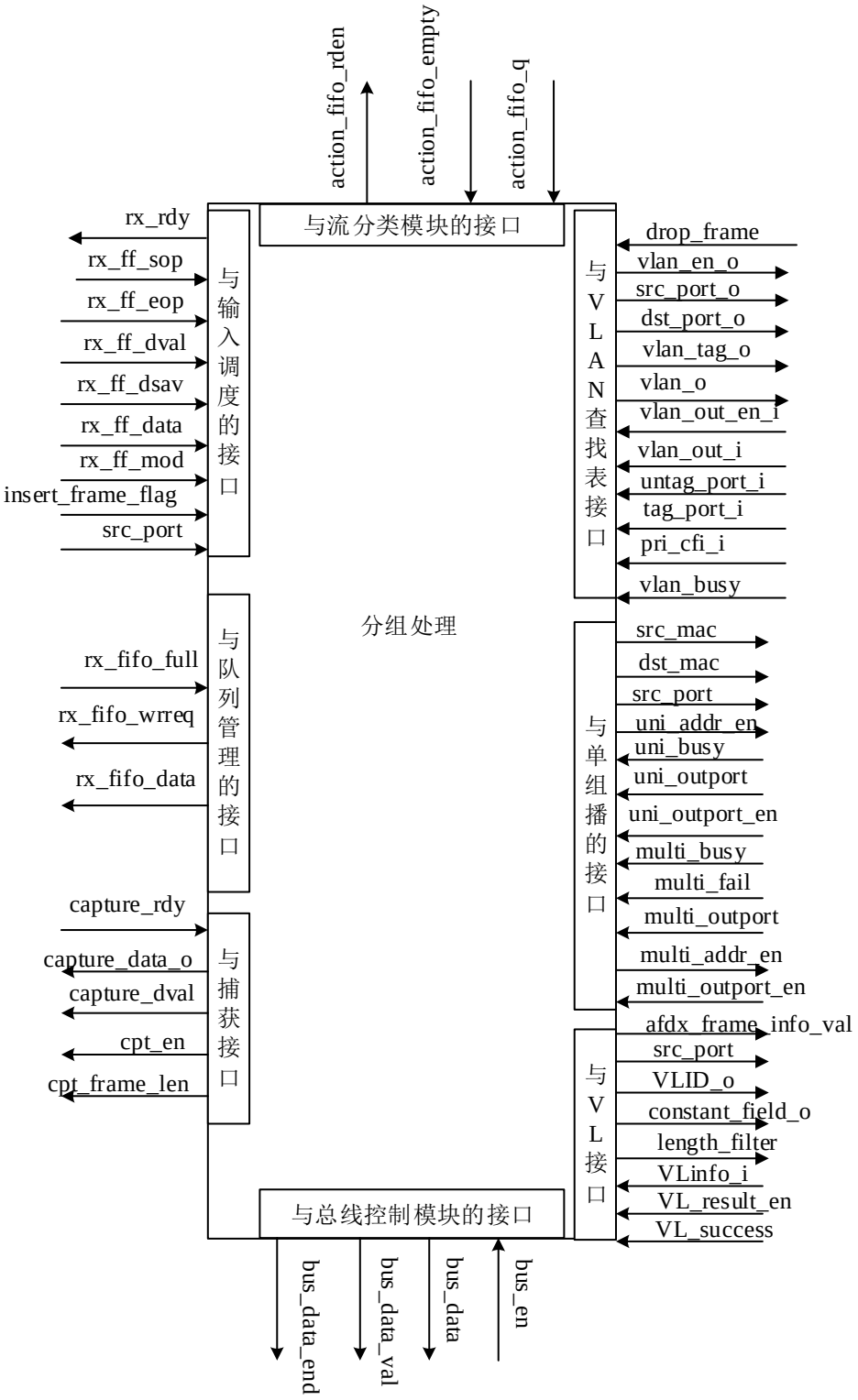

图3.1 分组处理对外的主要接口

分组处理模块的输入接口信号的定义如表 3.1 所示：

表3.1 分组处理模块输入信号定义

序号	名称	来源	位宽	定义
1	clk		1	100M 时钟信号
2	rst_n		1	复位信号
3	rx_ff_sop	接收调度模块	1	数据帧开始信号，高有效
4	rx_ff_eop	接收调度模块	1	数据帧结束信号，高有效
5	rx_ff_dval	接收调度模块	1	数据帧有效信号，高有效
6	rx_ff_dsav	接收调度模块	1	数据有效信号，高有效，比 rx_ff_dval 提前两个时钟信号
7	rx_ff_data	接收调度模块	32	数据帧的数据
8	rx_ff_mod	接收调度模块	2	用于标识最后一个 32 位 bit 中的有效字节 0：[31:0]有效，1：[7:0]有效， 2：[15:0]有效，3：[23:0]有效
9	insert_frame_flag	接收调度模块	1	插入帧标志位，高有效
10	src_port	接收调度模块	4	源端口号
11	rx_fifo_full	队列管理模块	1	rx_fifo 的空满信号，高表示满
12	capture_rdy	捕获模块	1	捕获模块准备好
13	bus_en	总线控制模块	1	数据搬移使能，高有效
14	multi_outport	多播转发表模块	12	多播查找结果
15	multi_outport_en	多播转发表模块	1	多播查找完成

16	multi_fail	多播转发表模块	1	多播查找失败
17	multi_busy	多播转发表模块	1	多播转发表忙
18	uni_outport_en	单播转发表模块	1	单播查找完成
19	uni_outport	单播转发表模块	12	单播查找结果
20	uni_busy	单播转发表模块	1	单播查找忙
21	vlan_busy	vlan 转发表模块	1	vlan 查找忙
22	pri_cfi_i	vlan 转发表模块	4	转发出去为 TAG 帧的相应的优先级
23	tag_port_i	vlan 转发表模块	12	转发出去为 TAG 帧的端口号
24	untag_port_i	vlan 转发表模块	12	转发出去为非 TAG 帧的端口号
25	vlan_out_i	vlan 转发表模块	12	转发为 TAG 帧时的 VLAN ID
26	vlan_out_en_i	vlan 转发表模块	1	查找完成信号
27	drop_frame	vlan 转发表模块	1	查找失败，丢弃信号
28	action_fifo_q	流分类模块	80	指令码
29	action_fifo_empty	流分类模块	1	指令码空满标志
30	VL_result_en	VL 转发模块	1	VL 查找结果有效信号
31	VL_success	VL 转发模块	1	查找结果成功信号
32	VLinfo_i	VL 转发模块	23	VL 转发模块给出的查找结果

分组处理模块的输出接口信号的定义如表 3.2 所示：

表3.2 分组处理模块输出信号定义

序号	名称	去向	位宽	定义
1	rx_rdy	接收调度模块	1	分组处理模块准备好能接受数据， 高有效
2	rx_fifo_wrreq	队列管理模块	1	rx_fifo 写使能信号，高有效
3	rx_fifo_data	队列管理模块	16	帧信息信号
4	capture_data_o	捕获模块	32	帧数据
5	capture_dval	捕获模块	1	捕获有效信号
6	cpt_en	捕获模块	1	捕获使能信号
7	cpt_frame_len	捕获模块	11	捕获帧的帧长
8	bus_data_end	总线控制模块	1	数据帧搬移完成
9	bus_data_val	总线控制模块	1	数据帧搬移有效，高有效
10	bus_data	总线控制模块	32	搬移的数据
11	src_mac	单组播转发表模块	48	源 MAC 地址
12	dst_mac	单组播转发表模块	48	目的 MAC 地址
13	src_port	单播转发表模块	12	源端口号
14	uni_addr_en	单播转发表模块	1	单播查找使能，高有效
15	multi_addr_en	多播转发表模块	1	多播查找使能，高有效
16	vlan_en_o	vlan 转发表模块	1	vlan 查找使能，高有效
17	src_port_o	vlan 转发表模块	12	源端口号
18	dst_port_o	vlan 转发表模块	12	目的端口号
19	vlan_tag_o	vlan 转发表模块	1	VLAN ID 有效信号，高有效

20	vlan_o	vlan 转发表模块	12	VLAN ID
21	action_fifo_rden	流分类模块	1	读取指令码信号，高有效
22	afdx_frame_info_val	VL 转发模块	1	信息有效信号
23	VLID_o	VL 转发模块	12	VL 值
24	constant_field_o	VL 转发模块	32	Constant 域
25	length_filter	VL 转发模块	11	数据帧长
26	src_port	VL 转发模块	12	源端口号，作用是判断将提取的信息传递给哪个 VL 转发模块

3.3 分组处理模块的具体实现

3.3.1 分组处理模块的内部结构

分组处理模块所包含的模块对以太网数据帧及 AFDX 数据帧的处理流程有些是相同的，有些是不同的。本章在对各个模块介绍时，未加说明是对以太网数据帧进行介绍。分组处理模块主要包括 fp_rx_ctrl 模块、fp_action_exe 模块、fp_bus_ctrl 模块、fp_action_analysis 模块。分组处理模块的内部结构如图 3.2 所示：

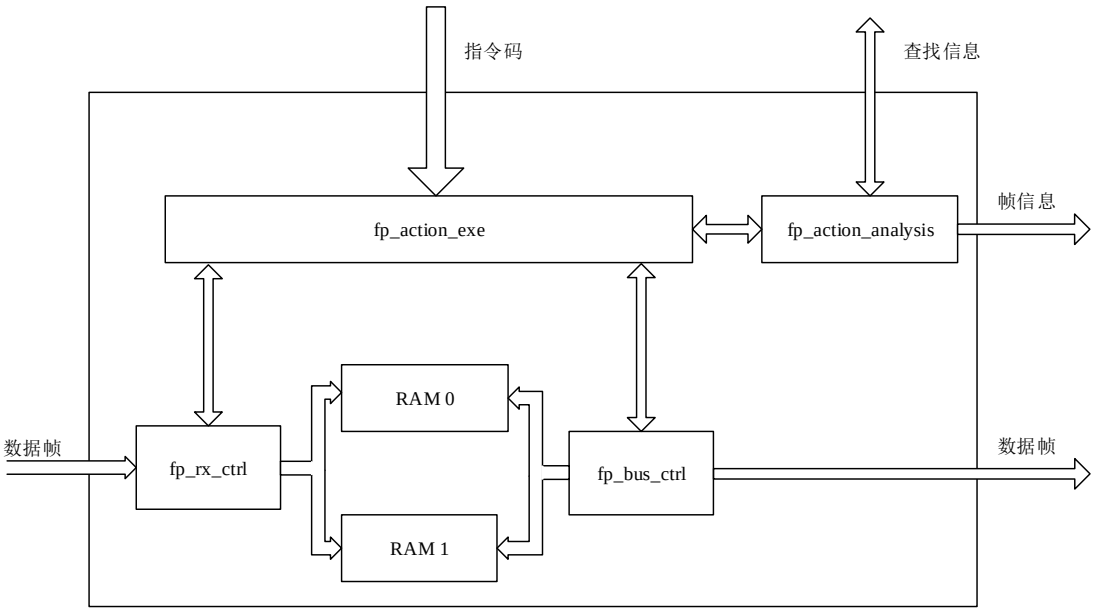

图3.2 分组处理模块的内部结构

当 fp_rx_ctrl 模块中的乒乓缓存非满时，若此时端口有数据帧输入则通过输入调度模块将数据接收进乒乓缓存中，在接收数据帧的同时进行简单的帧信息的提取，将提取后的信息传递给 fp_action_analysis 模块，fp_action_analysis 模块将帧信息给转发表模块，根据查找的结果，生成帧调度信息，并将该信息写入 rx_fifo 中共队列管理模块使用。fp_action_exe 模块接收流分类模块产生的指令码，对指令码中的信息进行分析并执行相应的操作，例如丢弃指令，则将数据帧进行丢弃，并将清空信号写入 clr_fifo 中来控制乒乓缓存的反转。fp_action_exe 模块中还要对数据帧进行输入监控操作。fp_bus_ctrl 模块负责数据帧的搬移并在搬移的过程完成数据帧的插入修改删除的操作。

3.3.2 fp_action_analysis 模块

fp_action_analysis 模块主要负责将提取的信息给相应的转发表模块，并根据转发表模块转发的结果产生相应的信息写入 rx_fifo 中。本模块在对以太网数据帧及 AFDX 数据帧的处理过程中有些区别，下面将对两种数据帧的处理进行分别介绍。

- 当该模块处理的为以太网数据帧时：

本模块主要是根据由 fp_frame_info_collect 模块提供的帧头信息，主要包括源端口号（src_port）、源 MAC 地址（src_mac）、目的 MAC 地址（dst_mac）、插入帧标志（insert_frame_flag）、TAG 帧标志及相应的 VLAN ID 等信息用来进行单播转发表、组播转发表、vlan 转发表的查找。根据查找的结果产生调度信息并将该信息写入接收调度的结果缓存（rx_fifo）中。当数据以广播或组播的形式转发出去时，在交换机支持 vlan 时，当从属于不同 VLAN ID 的端口转发出去时可能会存在同一个帧在转发时有可能为在某些端口转发出去时为非 TAG 帧而在其他端口转发出去时为 TAG 帧。因此，本模块会根据 vlan 转发表的结果生成关于同一个数据帧在向多个端口转发时哪些端口为 TAG 帧哪些端口为非 TAG 帧的信息，然后将该信息存到 frame_modify_fifo 中。

fp_action_analysis 模块的状态转移图如图 3.3 所示（由于本状态机的状态较多，倘若将状态转移的条件也在图上标出来的话，状态机的转移图将会比较复杂，不利于观看。本状态机的跳转过程将结合本模块的工作流程一个详细的介绍。）：

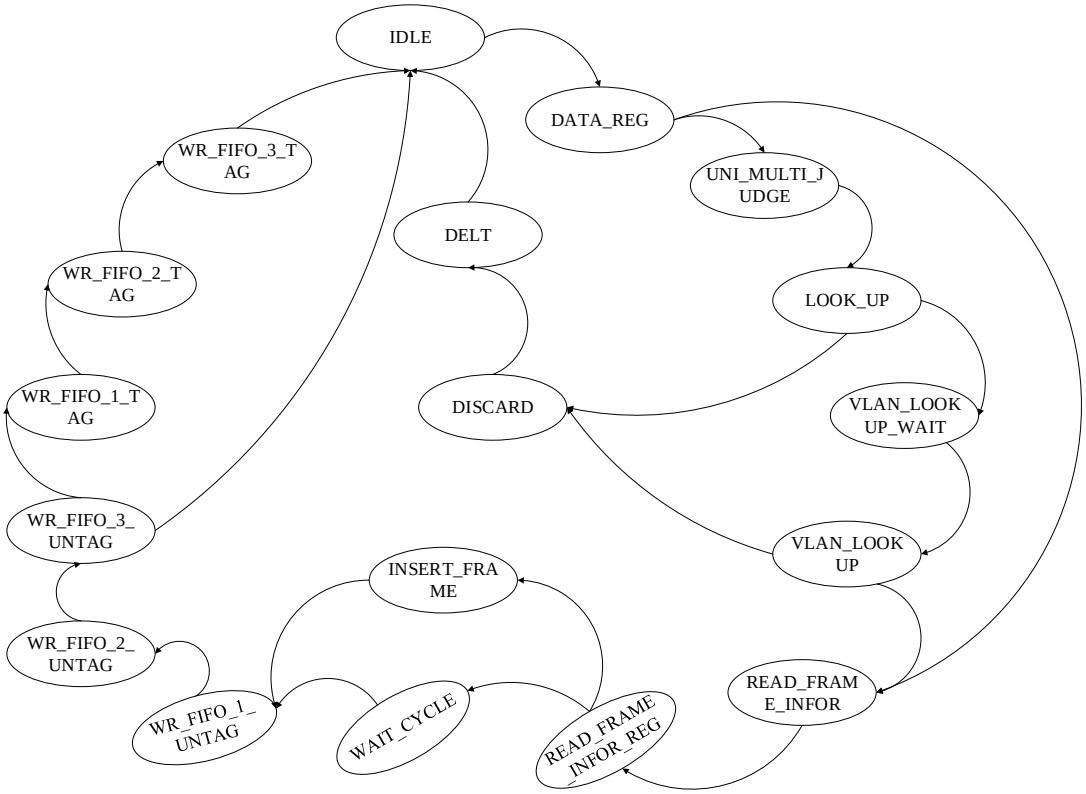

图3.3 fp_action_analysis 模块的状态转移图

本模块的工作流程如下：

（1） 当 fp_frame_info_collect 模块提取完帧头信息后将会产生一个帧头信息有效的信号给 fp_action_analysis 模块，若此时 rx_fifo 不满且 frame_modify_fifo 不满，则状态机将开始启动，进入 DATA_REG 状态开始接受帧头信息。

（2） 根据帧头信息中的插入帧标志位的值来判断正在处理的帧是否为插入帧，若为插入帧，则不需要进行转发表的操作状态机跳转到 READ_FRAME_INFOR 状态即跳转到（5）。若为非插入帧，则根据目的 MAC 地址来判断该帧是单播帧或是组播帧。然后进行相应的查找。

（3） 根据 MAC 地址进行相应的转发表查找后，得到查找的结果。当对组播帧进行查找且查找结果失败，则状态机将会跳转到 DISCARD 状态，对数据帧执行丢弃操作。单播时若查表失败则会对该帧在 vlan 内进行广播。当组播或单播查找成功后则跳转到 VLAN_LOOKUP_WAIT 状态等待 vlan 查找模块不忙时进行 vlan 查找。

（4） 当 vlan 查找模块不忙时，fp_action_analysis 模块将源端口号、目的端口号、TAG 帧标志及相应的 VLAN ID 传给 vlan 查找模块来进行查找。若查找失败则状态机跳转到 DISCARD 状态，对数据帧进行丢弃操作。若查找成功则跳转到 READ_FRAME_INFOR 状态。

（5） 若帧信息 FIFO（frame_info_fifo）非空，则从中提取数据帧的帧长，优先级及该帧是否为重定向的帧。根据 vlan 转发表的结果将数据帧转发出去时是否为 TAG 帧及相应的 VLAN ID 以及优先级等信息传个 fp_action_exe 模块。

（6） 根据 vlan 查找的结果以及帧信息 FIFO 中的信息生成调度信息并将该信息写入 rx_fifo 中。值得注意的是，由于当数据帧从多个端口转发出去时，则可能会出现要写两次调度信息的情况。例如：数据帧转发的端口中既包含 Access 口又包含 Trunk 口，从 Access 口转发的数据帧为非 TAG 帧，若从 Trunk 口转发时 VLAN ID 与 PVID 不同且该 VLAN ID 为该 Trunk 口所支持的 vlan，则从 Trunk 口转发的数据帧将为 TAG 帧。因此在这种情况发生时将会往 rx_fifo 中写两次信息。

➢ 当该模块处理的为 AFDX 帧时：

fp_action_analysis 模块中对 AFDX 帧的处理流程很相似。主要的区别如下：根据 fp_frame_info_collect 模块提供的 VL 值，Constant 域，以及数据帧长。然后 fp_action_analysis 模块将这些信息传递给 VL 转发模块，VL 转发模块根据该信息进行相应的过滤及警管，然后将结果传递给 fp_action_analysis 模块。当 fp_action_analysis 模块收到 VL 转发模块给出的结果后，若警管失败或过滤失败，则将该数据帧丢弃，若警管及过滤都成功，则将 Vlinfo 中的内容写入 rx_fifo 中。

本模块中使用了一些 FIFO 作为模块之间的通信，接下来对 FIFO 中的有效位的定义进行介绍：

rx_fifo：在本设计方案中 rx_fifo 中的数据是通过 FPGA2 与 FPAG3 的总线来传递的，为了节省管脚资源，在不影响交换机性能的情况下，考虑到调度信息为 48bit，我们将该总线的数据位宽设为 16bit，调度信息需要通过三次来传完。调度信息的结构如下图，信息中的各 bit 位的定义如图 3.4 所示：

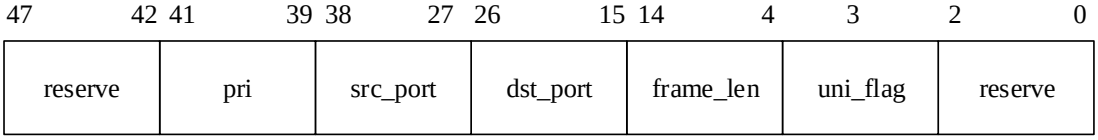

图3.4 调度信息的结构

0~2：保留位。

3：单组播标志位 表明该数据帧为单播还是组播。

4~14：数据帧长。

15~26：目的端口号，若为单播时其数值即为端口号，若为组播时该值为 bit_map 映射。

27~38：源端口号，该数据的值即为端口号。

39~41：该数据帧的优先级，供入队使用。

42~47：保留。

帧修改 FIFO（frame_modify_fifo）：

该 FIFO 中的信息主要是为了提供在数据帧搬移时对数据帧进行修改的信息，主要就是 VLAN ID、优先级以及转发时是否为 TAG 帧等信息。由于在多播或广播时同一个帧可能以 TAG 帧和非 TAG 帧来转发，这样在数据搬移时对同一个数据帧就有可能需要搬移两次，因此，本 FIFO 中的信息主要包括两个部分，一部分为从端口转发出去的数据帧为非 TAG 帧，另一部分为从端口转发出去的数据帧为 TAG 帧。帧修改信息的结构如图 3.5 所示：

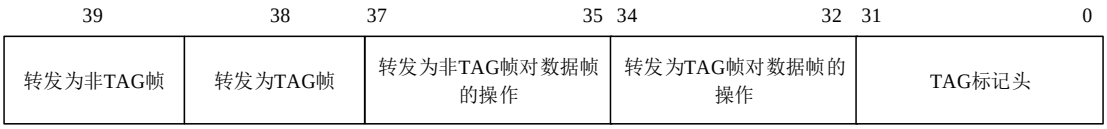

图3.5 帧修改信息的结构

帧修改信息的具体 bit 位的信息定义如下：

0~31：从端口转发出去的数据帧为 TAG 帧时相应的 4 个字节的 TAG 标记头。

32~34：当端口转发出去的数据帧为 TAG 帧时，在数据搬移过程中对原来的数据帧进行操作插入修改或删除的标志。

35~37：当端口转发出去的数据帧为非 TAG 帧时，在数据搬移过程中对原来的数据帧进行操作插入修改或删除的标志。

38：有效位，表明转发出去的数据帧为 TAG 帧。

39：有效位，表明转发出去的数据帧为非 TAG 帧（38bit 与 39bit 有可能同时有效）。

3.3.3 fp_action_exe 模块

本模块的主要功能是以下的四个方面：

- 当有数据要处理时，根据源端口号来判断该数据帧是否为输入监控帧（监控分为输入监控及输出监控。输入监控为对从某个特定的端口进入的数据帧进行复制并从监控端口转发出去。输出监控为对从某个特定的端口出去的数据帧进行复制并从监控端口转发出去。本模块只处理输入监控，输出监控在队列管理模块处理）。若为输入监控帧，则产生调度信息，等待数据帧的搬移。若不是监控帧则等待指令码的到来。
- 接收流分类产生的指令码，然后根据对指令码中的信息进行提取，将帧长、

优先级、重定向到 CPU 等信息传递给 fp_action_analysis 模块，在 fp_action_analysis 模块中用来产生部分的调度信息。

- 将数据帧的插入修改删除等信息传递给总线搬移模块（fp_bus_ctrl），fp_bus_ctrl 模块在对数据帧搬移的过程中根据该信息对数据帧进行插入修改删除的操作。
- 当数据帧搬移完成后或者对数据帧执行丢弃操作时，将清空信号（clr_fifo_data）写进 clr_fifo 中，乒乓状态控制模块（fp_pingpang_state_ctrl）根据该信息可知道数据帧已搬移完成，然后控制乒乓缓存的反转。

fp_action_exe 模块的工作流程如图 3.6 所示：

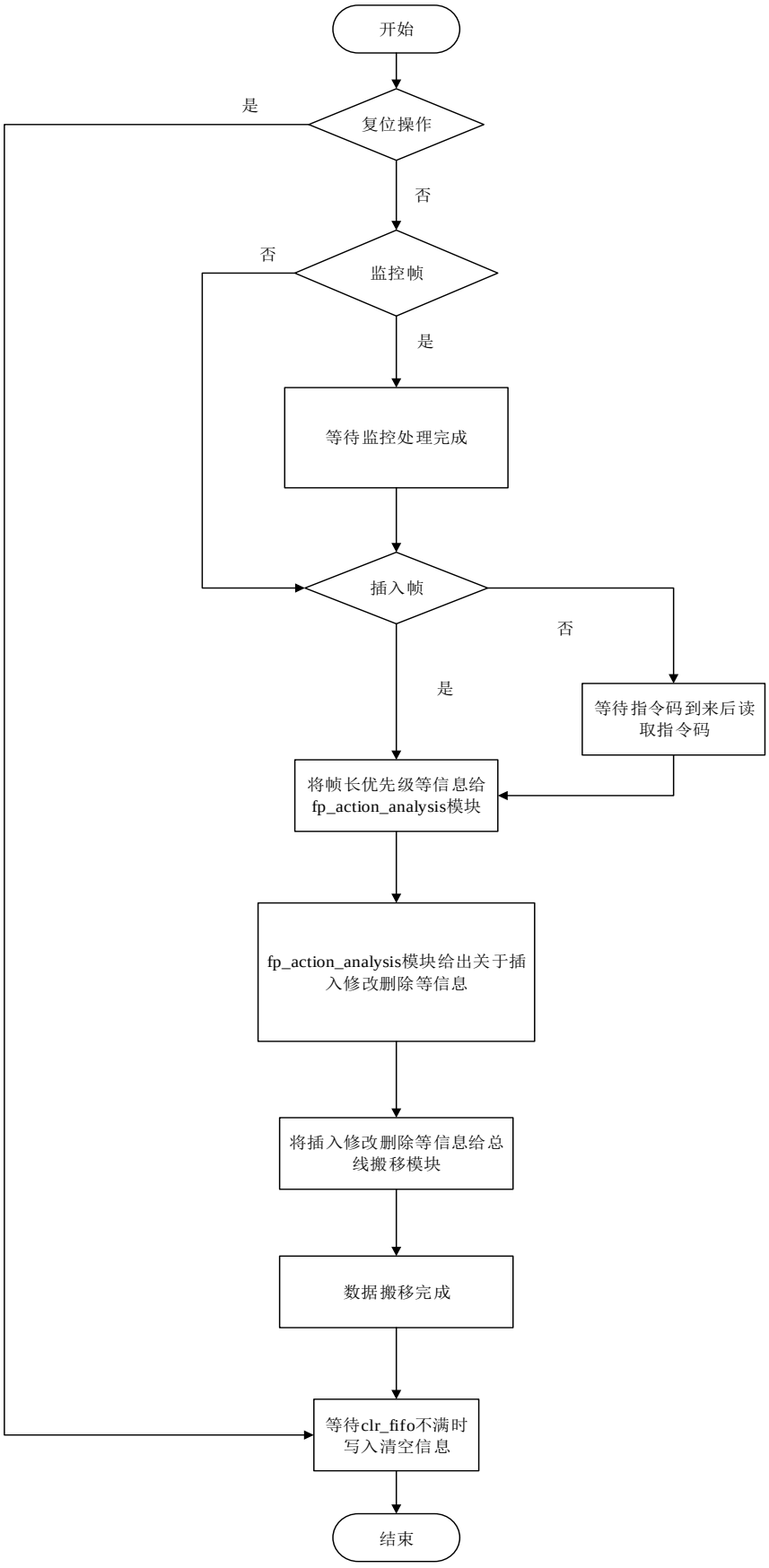

图3.6 fp_action_exe 模块的工作流程

（1） 为了使乒乓缓存模块能正常的工作，fp_action_exe 模块要在上电初始化后往 clr_fifo 中写入清空数据，详细介绍请见乒乓状态控制模块（fp_pingpang_state_ctrl）。

（2） 当数据帧到来时，判断该数据帧是否为输入监控帧，若不是监控帧，则判断是否为插入帧。若为监控帧，则生成调度信息，由于本交换机中端口 11 为输入监控口，因此生成的调度信息中的目的端口的值为 11，将生产的调度信息在 rx_fifo 非满时写入。等待数据帧的搬移完成后判断该数据帧是否为插入帧。

（3） 若正在处理的数据帧为插入帧，则该帧不需要经过流分类模块产生指令码也不需要通过转发表进行查找，直接生成调度信息并等待数据帧的搬移完成。若为非插入帧，则需要等待流分类模块的指令码，并对指令码进行分析。

（4） 分析指令码，若丢弃为有效，则该帧为丢弃帧，等待 clr_fifo 非空时将清空信号写入 clr_fifo。若为捕获到 CPU 的数据帧，若捕获模块满时则丢弃该帧，若捕获模块非满时将数据帧搬移到捕获模块。若数据帧不是捕获帧，则将数据的优先级，帧长等信息传给 fp_action_analysis 模块用来产生调度信息。

（5） 从 fp_action_analysis 模块获得在数据搬移过程中对数据帧执行的插入修改删除的信息。本模块将根据该信息产生数据帧需要搬移的次数，将搬移的次数以及插入修改删除的信息发送给 fp_bus_ctrl 模块，当数据搬移完成后在 clr_fifo 非空时将清空信号写入 clr_fifo 中。

3.3.4 接收控制模块（fp_rx_ctrl 模块）

接收控制模块主要完成的功能是对接收调度发送来的数据帧进行正确的接收和缓存，并对数据帧进行识别，判断数据帧为以太网数据帧还是 AFDX 帧，然后进行简单的帧信息提取，以太网数据帧提取的主要为源 MAC 地址、目的 MAC 地址、源端口号、VLAN ID 等，AFDX 帧提取的主要为 VL 值及 Constant 域。

分组处理模块中的缓存只是对数据帧进行暂存，等得到了数据帧的输出端口后，便通过总线控制模块将数据帧搬移出去，因此不需要太大的缓存，为了提高处理速率，使缓存中的数据在被搬移的同时，也能有新的数据帧写入。我们在本设计中采用乒乓缓存[4] 。其具体实现如图 3.7 所示：

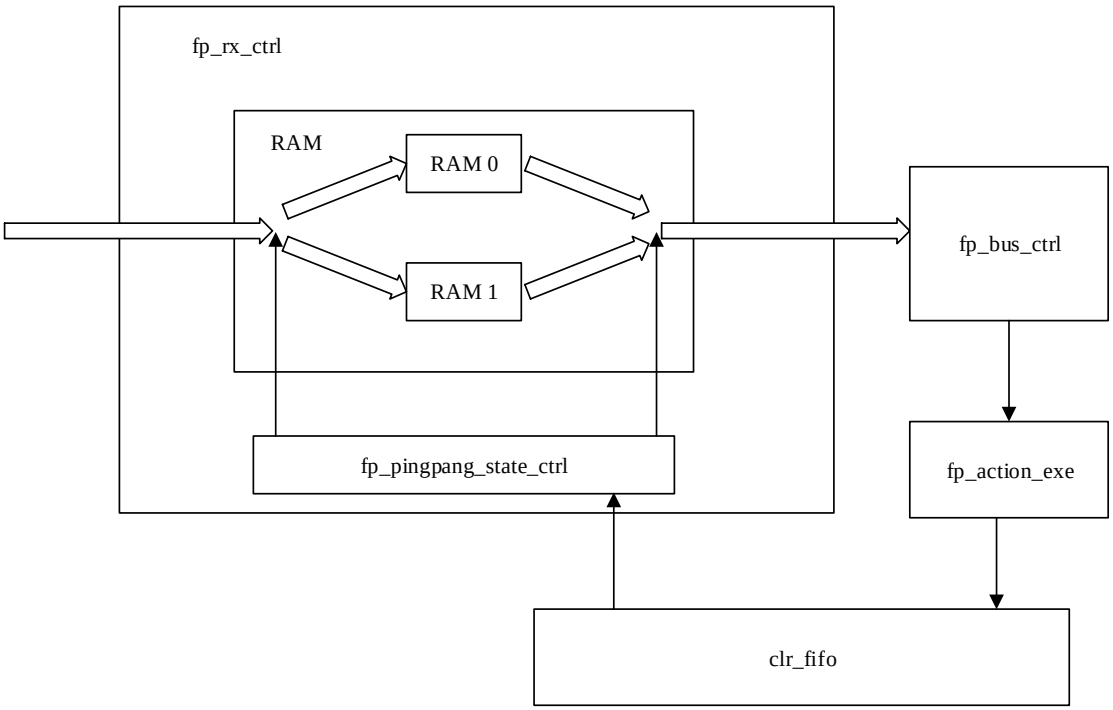

图3.7 分组处理模块中乒乓操作的实现

乒乓状态控制模块将两块缓存 RAM 0 和 RAM 1 对外封装成一块 RAM，两块 RAM 之间的切换是由乒乓状态控制模块（fp_pingpang_state_ctrl），指令执行模块（fp_action_exe），总线搬移模块（fp_bus_ctrl）以及清空信号（clr_fifo）来控制的。其中乒乓状态控制模块为控制两块 RAM 切换的核心模块，该模块的状态转移图如图 3.8 所示：

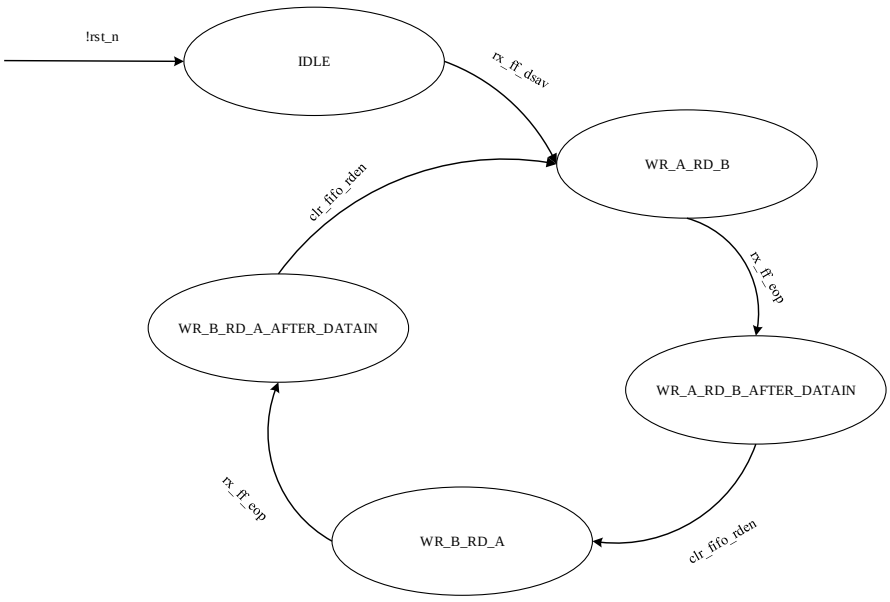

图3.8 乒乓状态控制模块状态转移图

在第一个缓冲周期时，fp_pingpang_state_ctrl 模块将数据帧存入 RAM 0 中，同时 fp_action_exe 模块中在上电复位后往 clr_fifo 写入一个清空信号。在第二个缓冲周期时，fp_pingpang_state_ctrl 模块在获知 clr_fifo 中存在清空信号时则将数据帧存入 RAM 1 中，此时总线搬移模块（fp_bus_ctrl）将 RAM 1 中的数据帧搬移出去，搬移完成后给 fp_action_exe 模块一个搬移完成信号，fp_action_exe 模块在 clr_fifo 非空时往 clr_fifo 中写入一个清空信号。在第三个周期时乒乓状态切换 往 RAM 0 中写数据，从 RAM 1 中读数据。之后循环往复。

3.3.5 总线搬移模块（fp_bus_ctrl）

总线搬移模块的主要功能为将数据帧正确的从乒乓缓存中搬移出来，并在数据帧搬移的同时执行添加、修改或删除的操作[10] 。AFDX 帧在搬移的过程中不需要执行添加、修改或删除操作。该模块的流程图如图 3.9 所示：

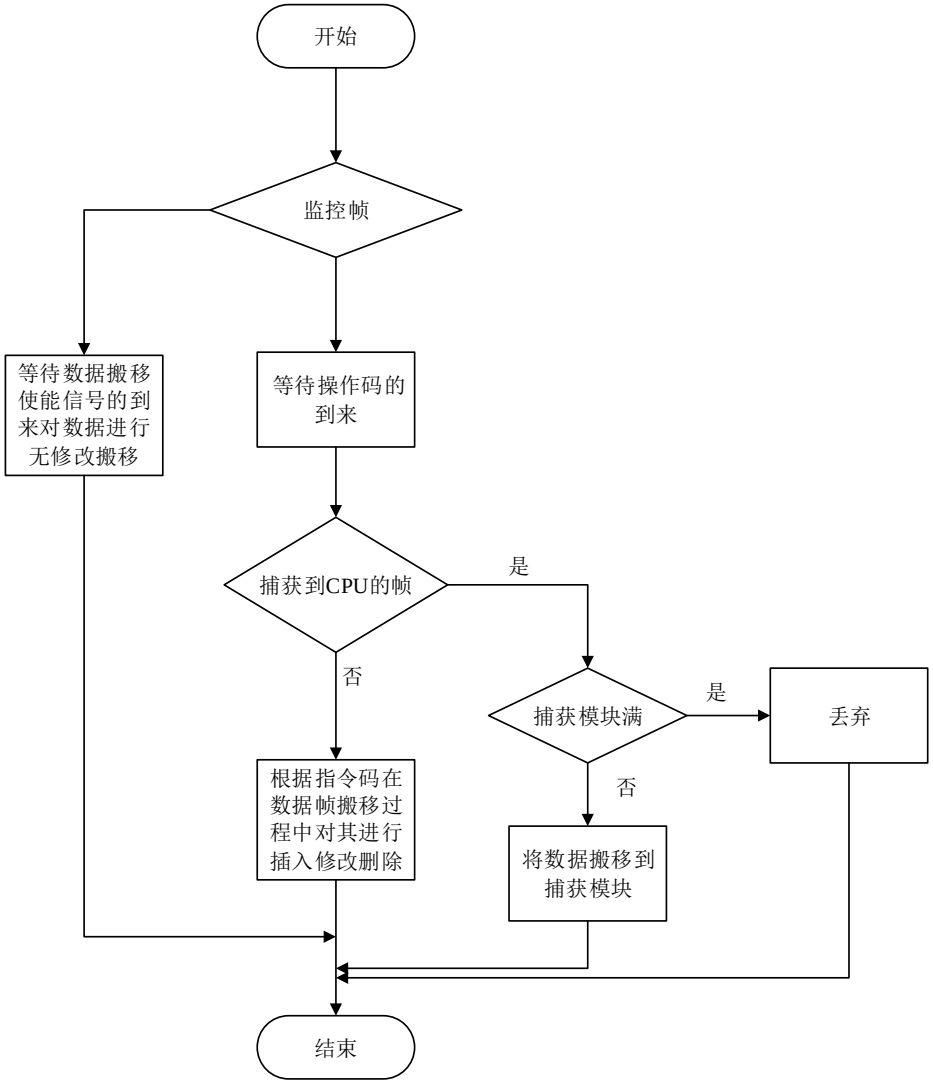

图3.9 总线搬移模块的处理流程

当处理的数据帧为输入监控时且该帧转发出去时为 TAG 帧与非 TAG 帧共存，则该数据帧需要被搬移三次，具体过程如下：

（1） 第一次为作为监控帧时不需要对数据帧进行修改，不需要等待 fp_action_exe 模块传递的操作码，在 fp_action_exe 模块中不需要流分类模块产生指令码，而是直接生成调度信息给队列管理模块，队列管理模块根据调度信息生成分配存储空间，并产生数据搬移使能信号给 fp_bus_ctrl 模块，fp_bus_ctrl 模块在收到使能信号后对乒乓缓存中的数据进行无修改的搬移。

（2） 当数据帧第一次搬移完成后，总线控制模块（bus_master_rx）将搬移完成信号给 fp_action_exe 模块及 fp_bus_ctrl 模块，fp_action_exe 模块根据流分类模块产生的指令码及 fp_action_analysis 模块中的关于数据帧插入修改删除的消息产生操作码给总线搬移模块，与此同时 fp_action_analysis 模块产生调度信息给队列管理模块，队列管理模块分配完地址后，总线控制模块给出数据搬移使能信号给总线搬移模块，总线搬移模块根据操作码在数据帧搬移的过程中对数据进行插入修改删除操作。数据搬移完成后，总线控制模块给出搬移完成信号给 fp_action_exe 模块及 fp_bus_ctrl 模块。

（3） 数据帧第二次搬移完成后，第三次的搬移的搬移过程基本上同第二次的搬移过程，主要区别是在总线控制模块给出搬移完成信号给 fp_action_exe 模块后，fp_action_exe 模块在 clr_fifo 非满时往 clr_fifo 中写入清空信息，来控制乒乓 RAM 的切换。

3.4 分组处理模块的功能仿真

3.4.1 分组处理模块对以太网数据帧处理的功能仿真

功能仿真是对所设计的 RTL 级的代码进行验证[13] ，通过仿真的结果可以知道设计是否与自己的构想相一致。通过仿真能快速发现设计中的问题及缺陷，能够缩短开发的周期。本文中的功能仿真是利用 Modelsim 10.1a 及 Xilinx ISE 14.5 的联合仿真来实现的。

接收控制模块（fp_rx_ctrl）中的乒乓缓存的正确切换是数据帧能够正确接收的重要保证。我们以随机帧长且随机帧间隔来发送数据帧，来查看乒乓缓存的翻转是否正常。当写 RAM 0，读 RAM 1 时 path_ctrl 的值为 0，当写 RAM 1，读 RAM 0 时 path_ctrl 的值为 1。由图 3.10 可知，乒乓翻转正常。

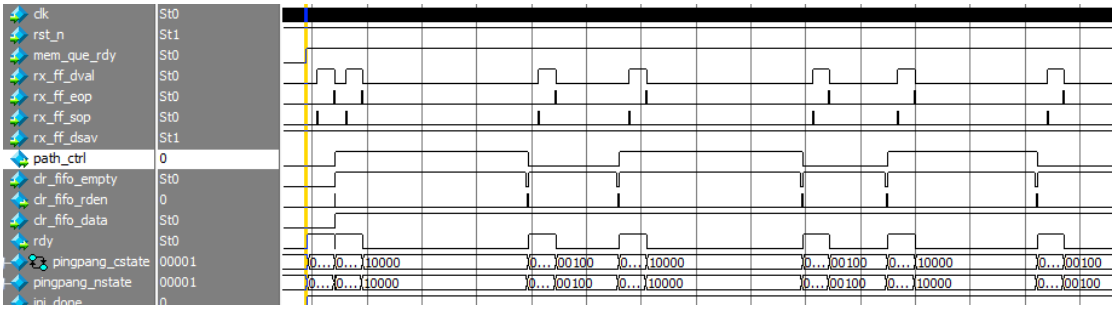

图3.10 乒乓操作仿真结果

接收控制模块（fp_rx_ctrl）中的帧信息收集模块（fp_frame_info_collect）负责对数据帧识别并对帧头信息的提取。本小节针对以太网数据帧的信息提取，例如源端口号，源 MAC 地址，目的 MAC 地址，以及是否为 TAG 帧等信息。在数据帧信息提取完毕后，生成有效信号，当 fp_action_analysis 模块收到有效信号且该模块不忙，则接收帧头信息，并生成帧信息接收完成信号。

对非 TAG 帧进行帧信息的提取的仿真如图 3.11 所示：可知数据帧为非 TAG 帧，目的 MAC 地址为 0X010203040506，源 MAC 地址为 0X644604030201。端口号为 1，对比的提取的信息可知，信息提取正确。

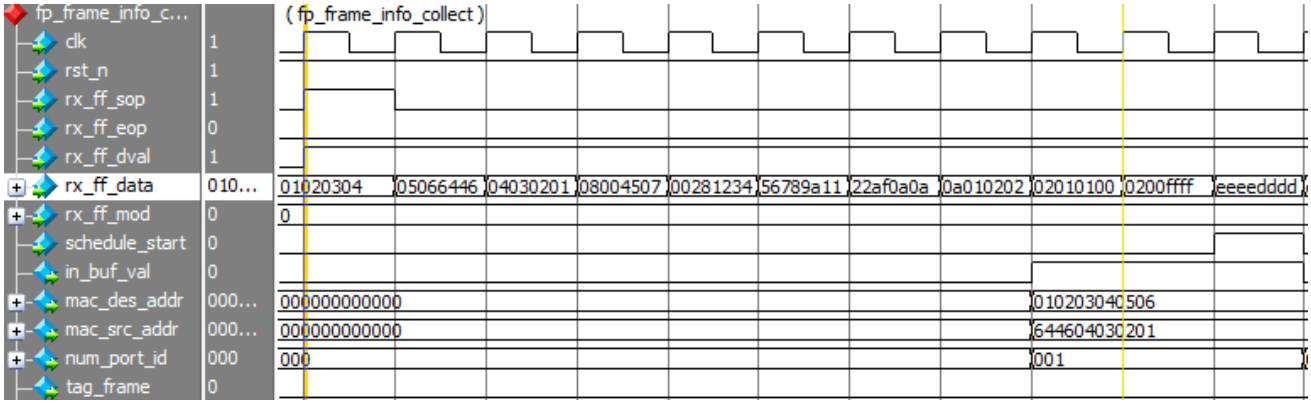

图3.11 非 TAG 帧的信息提取仿真

当数据帧为 TAG 帧时的信息提取仿真图如图 3.12 所示：根据数据帧中的帧类型的值为 0X8100 可知该帧为 vlan 帧，该数据帧的目的 MAC 地址为 0X644604030201，源 MAC 地址为 0X010203040506，VLAN ID 为 0X001，对比提取的帧信息可知信息提取正确。

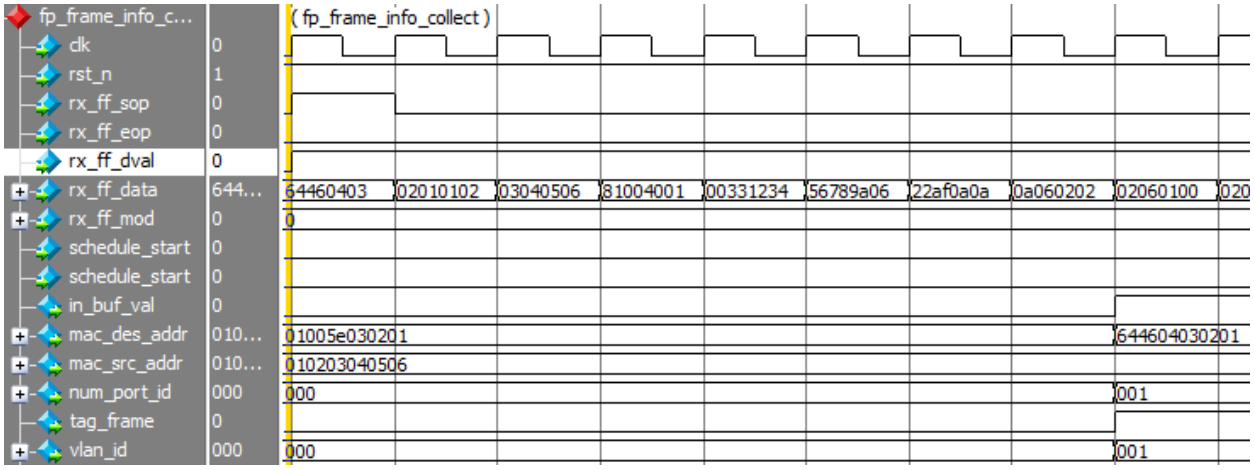

图3.12　　TAG 帧的信息提取仿真

fp_action_analysis 模块传递给队列管理模块的调度信息对于数据帧的缓存分配以及入队等有很重要的作用，下面是关于调度信息的仿真。该信息的传递是通过 rx_fifo 来实现的。由于本设计中专用交换机支持 access 口、trunk 口或 hybird 口，因此数据帧转发出去时有可能为 TAG 帧或非 TAG 帧抑或二者同时存在。当二者同时存在时，需要对乒乓缓存中的同一个数据帧搬移两次，因此也需要进行两次入队的申请，即需要往 rx_fifo 中写入两次调度信息。另外，当数据帧为输入监控帧时，需要对数据帧进行直接转发到输入监控端口，因为该帧不需要目的端口的查找，在获知该帧为监控帧时就将该帧的帧信息写入 rx_fifo 中。之后再将该监控帧当做普通的数据帧进行处理。下面的测试包括以下几种情况：

（1）　写一次调度信息。由于 3 个时钟周期写一次调度信息进 rx_fifo，由仿真图 3.13 中的状态机的跳转可知仅往 rx_fifo 中写了一次调度信息。

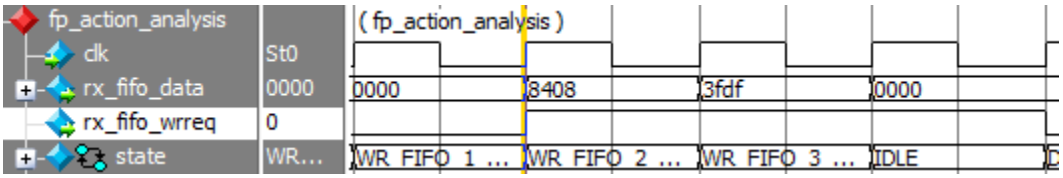

图3.13　　写一次调度信息仿真

（2）　写两次调度信息。由图 3.14 中的状态机可知往 rx_fifo 中写了两次调度信息。

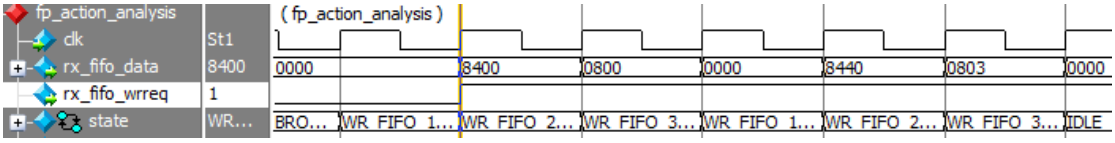

图3.14　　写两次调度信息仿真

（3）　监控帧的调度信息。本次仿真中对到达端口 1 的数据帧进行监控，监控端口

为端口 11。由仿真图 3.15 可知，图中的源端口号（sou_id）为 0X001，监控端口号（monitor_port）为 0X001。可知该数据帧为监控帧，在 rx_fifo_wrreq 为高时将信息写入 rx_fifo 中。

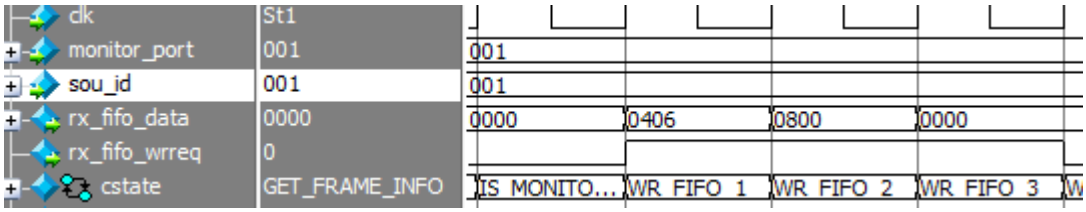

图3.15　　监控帧的调度信息的仿真

总线搬移模块（fp_bus_ctrl）负责将数据帧从乒乓缓存中正确搬移出来并在搬移的过程中根据指令对数据帧进行插入、修改、删除的操作。由于专用交换机为 2 层交换机，所以对数据帧的插入、修改、删除操作仅限于 TAG 标记头的插入，修改，删除。例如，当进入交换机的数据帧为非 TAG 帧转发出去时为 TAG 帧，此时在数据帧搬移的时候就要执行插入操作。

对数据帧在搬移过程中的操作取决于对该帧的进行的操作指令，即 action_out 的值。action_out 的位宽为 3，若 action_out[2]为 1 时，表示对数据帧进行插入操作，action_out[1]为 1 时，表示对数据帧进行删除操作，action_out[0]为 1 时，表示对数据帧进行修改操作。插入的值为 value 的值，由于只是对数据帧中的 TAG 标记头进行插入、修改、删除操作，所以对数据帧改变的位置是定的即在数据帧源 MAC 地址后进行操作。

如图 3.16 所示，action_out 的值为 3’b100，value 的值为 0X81001001，即在数据帧搬移过程中插入 TAG 标记头，且插入的值为 0X81001001。实现的方法是在从乒乓缓存中读取源 MAC 地址后停止从乒乓缓存中读出新的数据，在数据帧搬移时将 TAG 标记头添加进去，之后再从乒乓缓存中读新的数据。对比图 3.16 和图 3.17，可知插入成功。

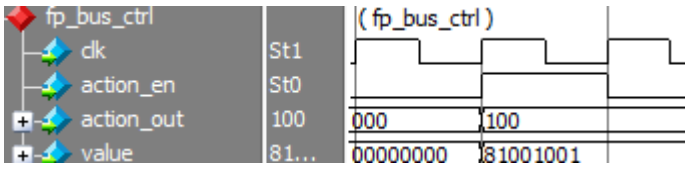

图3.16　　插入操作指令

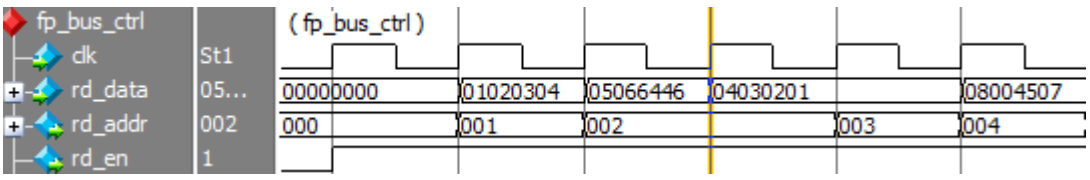

图3.17　　从乒乓缓存中搬移的数据

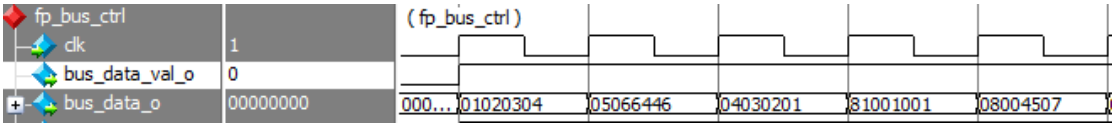

图3.18　在搬移过程中执行插入后的数据

如图 3.19 所示：action_out 的值为 3'b001，可知对数据帧进行删除操作，即将数据帧的 TAG 标记头去掉。去掉 TAG 标记头的方式为从乒乓缓存中读数据时跳过 TAG 标记头，如图中从缓存中读地址在 0X002 之后读 0X004 地址中的数据。比对图 3.20 和图 3.21，可知将 TAG 标记头去掉。

图3.19　删除操作指令

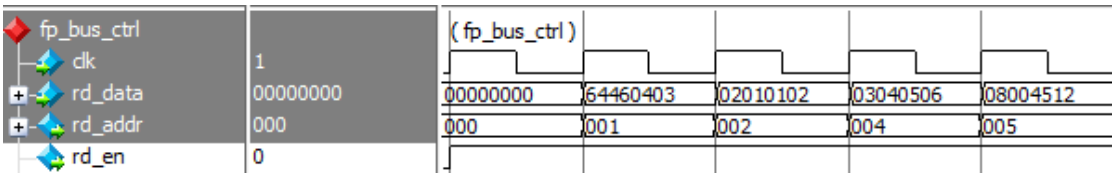

图3.20　从乒乓缓存中搬移的数据

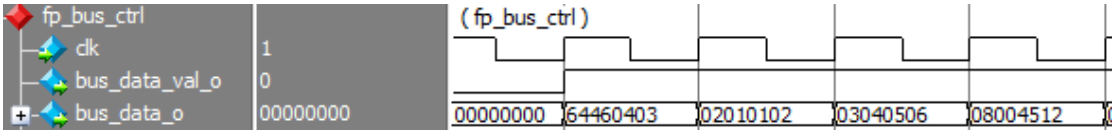

图3.21　在搬移过程中执行删除后的数据

3.4.2　分组处理模块对 AFDX 帧的功能仿真

分组处理模块中有些模块对 AFDX 帧及以太网数据帧的处理是相同的。在本小节中只描述了与以太网数据帧处理不同的部分，相同的部分就不再赘述了。

接收控制模块（fp_rx_ctrl）中的帧信息收集模块（fp_frame_info_collect）负责对数据帧的帧头信息的提取以及对数据帧的识别。对 AFDX 帧识别及信息提取的仿真如图 3.22 所示：当发送的数据帧的目的 MAC 地址为 0X034604030001 时，根据目的 MAC 地址的第一个字节的最后 2 个 bit 的值为 11，所以将该数据帧识别为 AFDX 帧，并将 AFDX 帧信息有效位 afdx_frame 置高，vlid 的值为 0X001，constant_field 的值为 0X03460403，当 fp_action_analysis 模块收到有效信号且该模块不忙，则接收帧头信息，并生成帧信息接收完成信号。

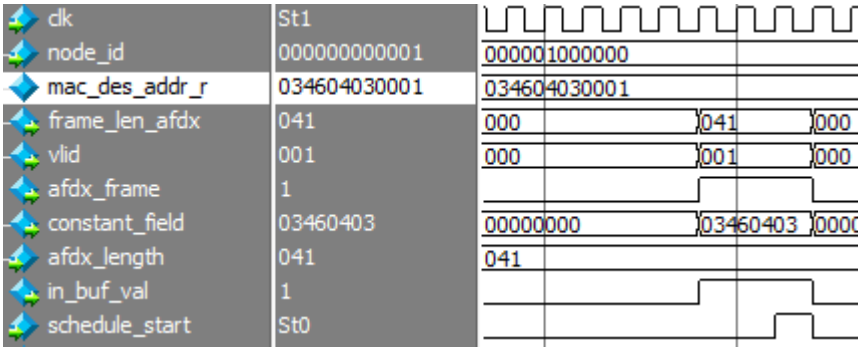

图3.22 AFDX 帧信息提取

如图 3.23 所示：fp_action_analysis 模块在收到帧信息收集模块（fp_frame_info_collect）提取的数据帧头信息，根据 AFDX 帧有效判断帧头信息是关于 AFDX 数据帧的，之后根据源端口号（src_port）将接收到的 VL 值（VLID_O）0X001，Constant 域（constant_field_o）0X03460403 以及数据帧长（length_filter）0X041 在 AFDX 帧信息有效（afdx_frame_info_val）时发送给相应的 VL 转发模块，进行目的端口的查找，以及过滤和警管。在 8 个时钟周期后 VL 转发表模块给出了 VL 查找结果有效(VL_result_en)信号，VL 查找成功信号（VL_success）以及查找的结果（VLinfo）。

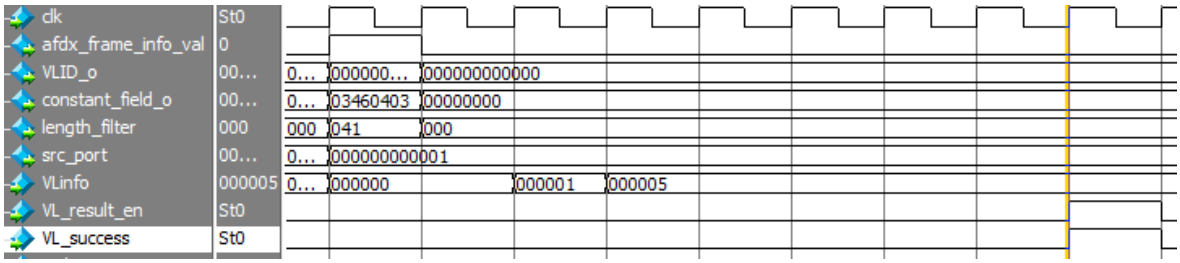

图3.23 fp_action_analysis 模块对 AFDX 帧的处理 1

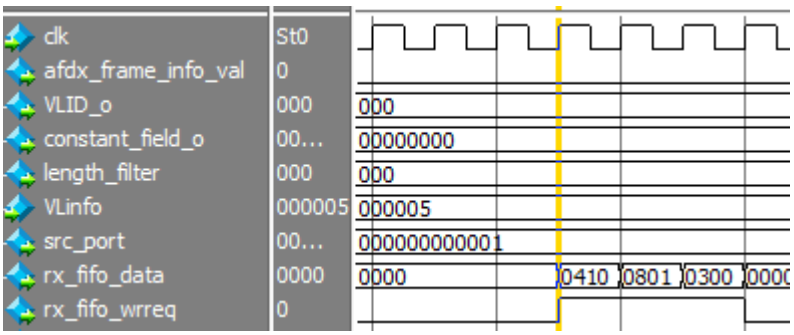

图3.24 fp_action_analysis 模块对 AFDX 帧的处理 2

如图 3.24 所示：fp_action_analysis 在收到 VL 转发模块的结果后，判断是否警管

及过滤成功，若失败则将数据帧丢弃，若成功则跟据查找的结果来生成调度信息并写入 rx_fifo 中，在 VLinfo 中，Vlinfo[12:1]为转发的目的端口，VLinfo[17]为优先级指示位，VLinfo[18]为多播指示位。我们将 AFDX 数据帧的优先级设置比较高有两种，若 VLinfo[17]为 1 时，在生成 rx_fifo 时的队列优先级为 3’b111。若 VLinfo[17]为 0，在生成 rx_fifo 时的队列优先级为 3’b110。对比 rx_fifo 中的信息可知结果符合期望。

第四章　专用交换机转发表模块的设计及实现

4.1 以太网转发表模块

4.1.1 以太网转发表模块的整体设计

以太网转发表模块的主要的功能为根据分组处理模块给出的帧信息（主要包括源MAC 地址、目的 MAC 地址、源端口号、VLAN ID 等）进行查找并给出查找的结果。分组处理模块根据查找的结果来决定数据帧从哪个或哪些端口转发出去以及从该端口转发出去的数据帧是否为 TAG 帧。以太网转发表模块主要包括单播转发表模块，组播转发表模块以及 vlan 转发表模块。本节重点介绍 vlan 转发表模块的设计及实现。转发表模块的处理流程如图 4.1 所示：

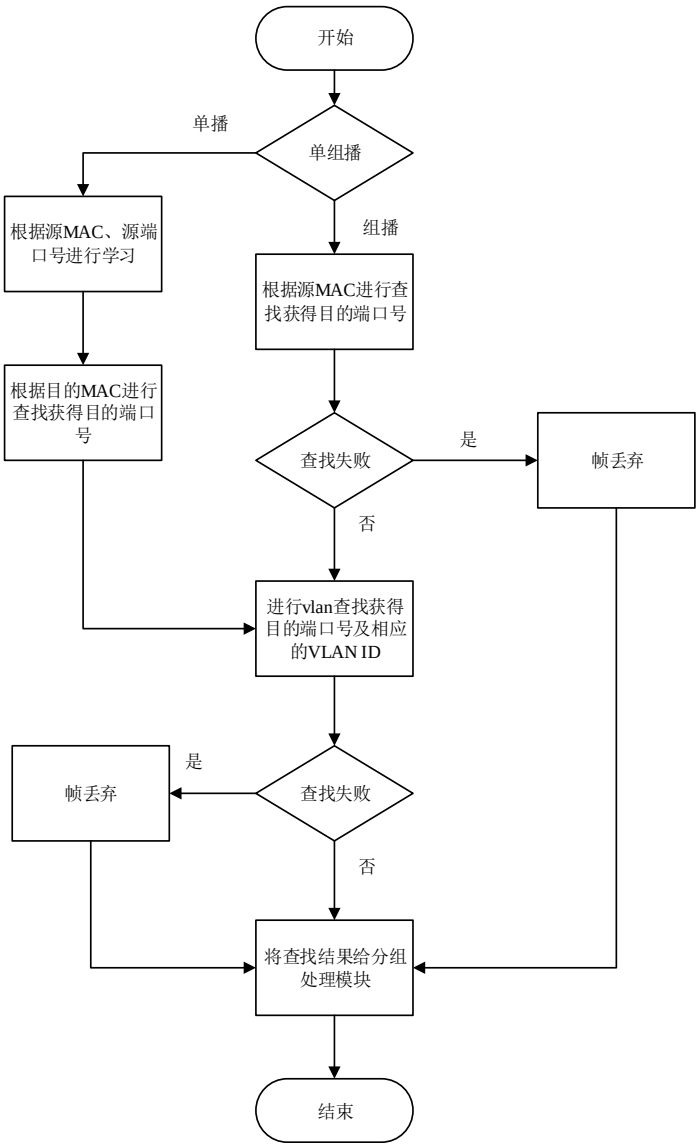

图4.1 转发表模块的工作流程

分组处理模块根据目的 MAC 地址来判断数据帧是单播帧还是组播帧，然后根据判断结果将帧信息发送到单播转发表模块或是组播转发表模块。若为单播，则单播转发表首先根据源 MAC 地址，源端口号进行地址表的学习，然后根据目的 MAC 地址进行查找，查找的结果为目的端口号，若查找失败的话，则将该数据帧进行广播。若为组播，因为组播地址不可能出现在源地址上，所以组播表不能进行自学习。组播表是由 CPU 来进行更新，分组处理模块只需将目的 MAC 地址传递给组播转发表模块，然后组播转发表模块根据目的 MAC 地址进行查找，查找的结果为一组端口号，若查找失败则将数据帧丢弃。单组播查找完成后进行 vlan 查找，vlan 转发表模块根据目的端口号以及源端口号进行查找，查找的结果为目的端口号，以及从该端口转发出去时该数据帧是否为 TAG 帧，若查找失败给出丢弃信号。

4.1.2 单播转发表模块的具体实现

1) 单播转发表模块

单播转发表模块主要包括 hash 映射模块（hash）、学习查找模块（study_lookup）、老化删除模块（delete）、单播地址表模块（addr_table）构成[12] 。根据设计需求，地址表的表项为不少于 1k，本设计中将地址表项的深度设计为 1024。将 48 位的 MAC 地址通过 hash 映射为 8 位的 hash 值。为防止 hash 冲突，将一个 hash 值对应于四个地址表中的表项，相应的表项地址依次为{hash，2’b00}，{hash，2’b01}，{hash，2’b10}，{hash，2’b11}。单播转发表的结构如图 4.2：

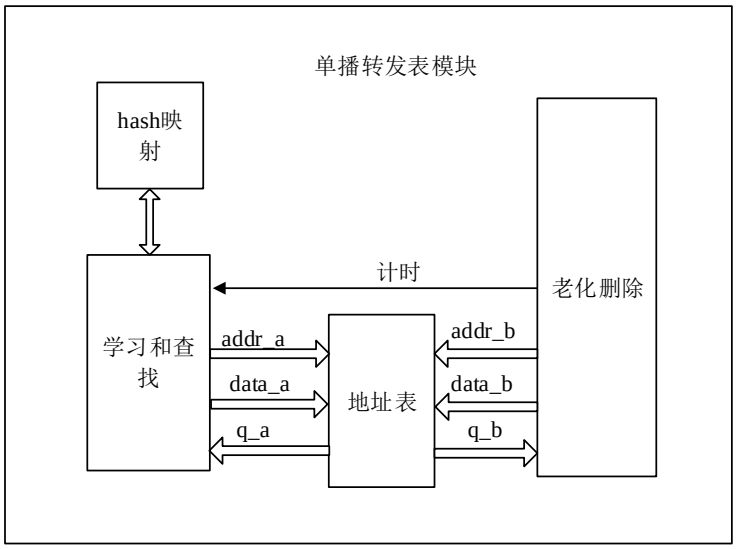

图4.2 单播转发表的内部结构

2) 学习查找模块

该模块是单播转发表模块的核心模块，主要负责地址表的学习以及目的端口的查找。该模块状态转移图如图 4.3：

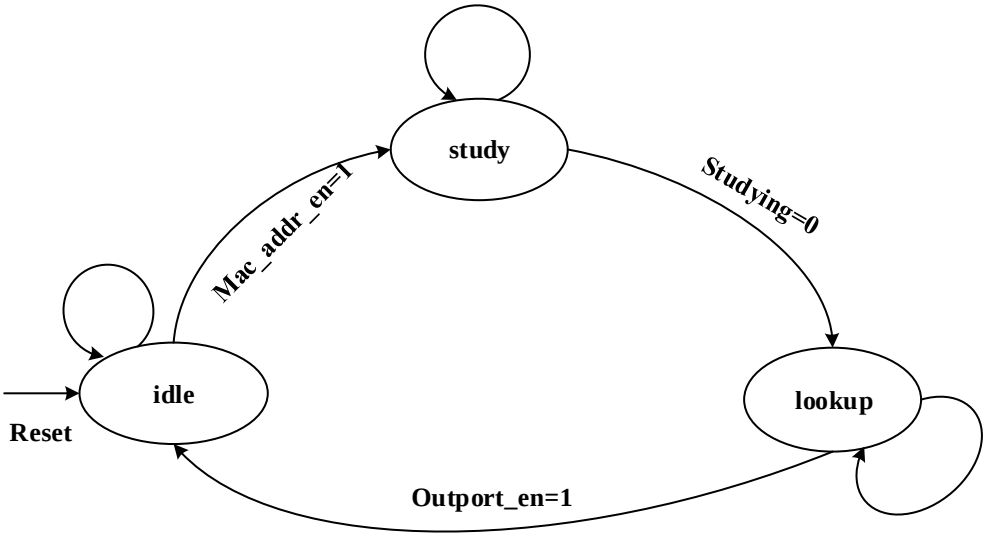

图4.3 学习查找模块的状态转移图

- 学习查找模块的学习过程：

当该模块收到数据帧的源 MAC 地址经过 hash 映射后的值时，根据{ hash，2’b00}对地址表进行第一次学习。若该表项中的 MAC 地址与源 MAC 地址相等且该表项有效，表明此 MAC 地址已经学习过了，对该表项中的老化时间进行更新即可。若源 MAC 地址与表项中的 MAC 地址不等，则与{hash，2’b01}中的 MAC 地址进行比较，若相等，学习结束，否则比对{hash，2’b02}，过程同上。若查到{hash，2’b03}时，仍然不相等，则表明该 MAC 地址之前没被学习过，需要将其添加到地址表中。依次对 hash 映射所对应的 4 个地址顺序的进行有效位的查看，若 4 个表项中存在有效位的值为 0，则表明该表项为空，将 MAC 地址与端口号加入该表项即可。若都非空，则不对该 MAC 地址进行学习。

- 学习查找模块的查找过程：

- 目的 MAC 地址经过 hash 映射后产生相应的 4 个单播表的地址。对 4 个地址表项中的内容依次进行匹配查找。
- 若表项中的 MAC 地址与目的 MAC 地址相匹配，则查找成功，根据表项中的端口转发出去。若 4 个地址中的 MAC 地址均未能与目的 MAC 地址相匹配则查找失败，将该数据帧广播出去。

3) 老化删除模块

该模块的主要功能是将地址表中的长时间未使用到的表项进行删除，释放单播表的存储空间。老化删除模块中有一个计数器，每过一秒计数器加 1，计数器的最大值为 300。当地址表中的一个表项建立或被更新时，表项中记录的时间为当前计数器中的值减 1(将表项中的计数记为比当前计数少 1，是为了避免将刚更新的表项误删。)。老化删除模块当学习查找不忙时对地址表中的表项进行轮询，来查看表项中的时间计数。若发现表项中的计数与计数器中的计数相等时，表明该表项长时间未更新，然后将其清空。该模块的状态转移图如图 4.4 所示：

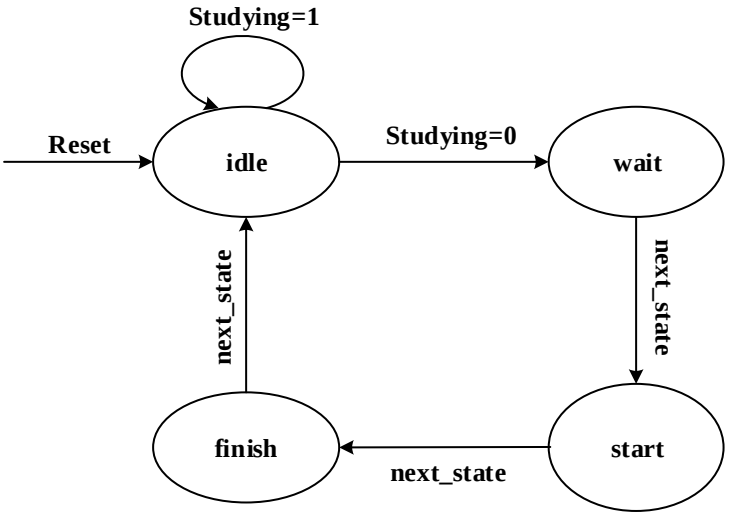

图4.4老化删除模块状态转移图

4) 单播地址表模块

单播地址表模块是用来存储自学习的表项，也是数据帧转发端口查找的依据。单播地址表的表项深度设为 1024，位宽设为 76bit。如图 4.5 所示：

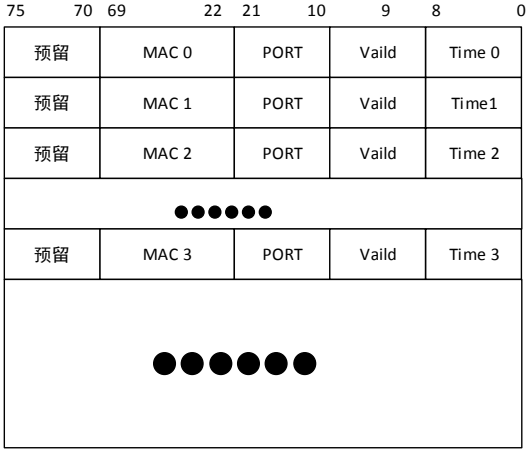

图4.5 单播地址表的结构

表项中各 bit 位的定义如下：

- 0~8：该表项最近一次被学习到的时间。
- 9：表项有效的指示位。
- 10~21：数据帧的源端口号。
- 22~69：数据帧中的源 MAC 地址。
- 70~75：保留位。

4.1.3 组播转发表模块的具体实现

1) 组播转发表的结构

组播转发表主要包括更新模块（renew）、查找模块（lookup）、hash 映射模块（multi_hash_renew 及 multi_hash_lookup）、组播地址表（multi_addr_table）。组播转发表模块中更新是由 CPU 来完成的，地址的表项为 256，由于组播地址的特殊性，将 24 位的 MAC 地址映射为 8 位的 hash 值，为了预防 hash 冲突，同一个 hash 值对应 4 个转发表的地址，分别为{hash,hash+1,hash+2,hash+3}。组播转发表的结构图 4.6。

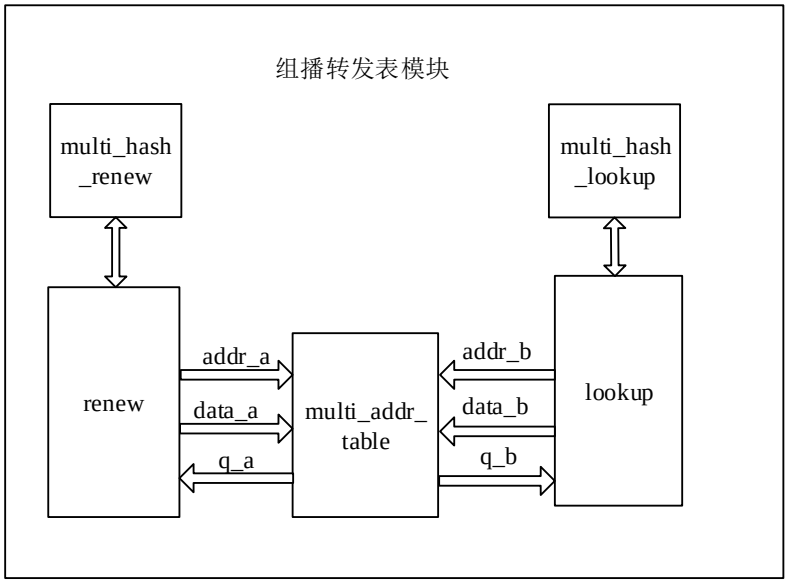

图4.6 组播转发表模块的内部结构

2) 更新模块

由于组播目的地址的特殊性,组播表不能自学习，地址表的更新需要 CPU 来维护，查找过程是通过 MAC 地址来查找。可见，查找与更新是两个独立的过程，因此将更新模块与查找模块分开。更新模块的状态跳转如图 4.7。

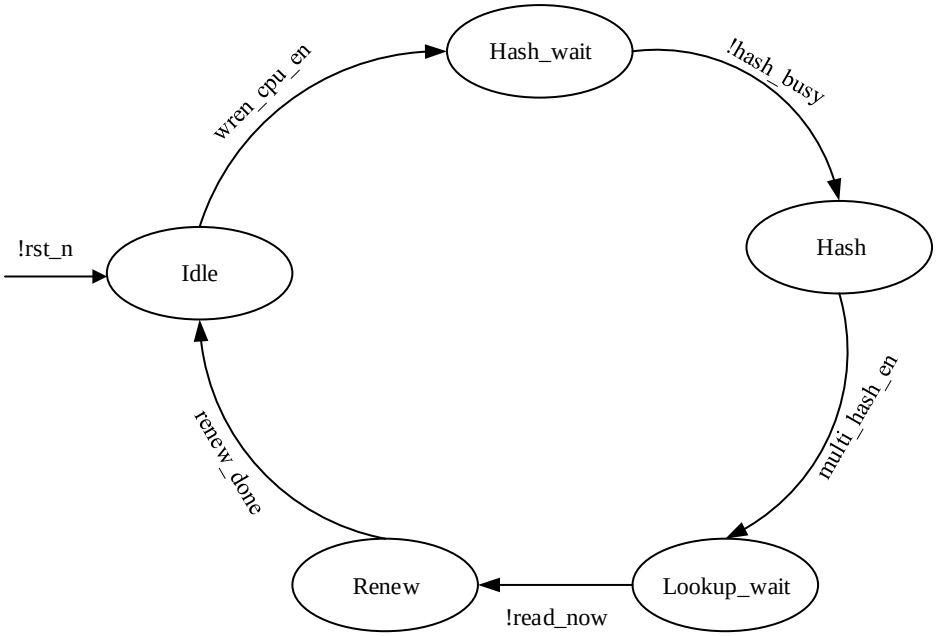

图4.7 更新模块的状态跳转

（1）对 CPU 接口模块的 wren_cpu_en 信号进行检测，若有效时，则将寄存器中的内容读出来。寄存器中的有效信息主要包括：组播 MAC 地址、组播成员、添加、删除、或更新表项。

（2）然后根据 MAC 地址进行 hash 映射，判断 hash 映射后的地址与查找模块的地址是否相等，若相等则是对同一地址进行操作此时需要等待，不相等则跳转到（3）。

（3）判断是对表项进行添加、删除、还是更新操作，若为添加，则根据 hash 映射的 4 个地址，依次判断相应的表项是否为空，若为空则添加进去。若 4 个地址中都有有效的数据，则丢弃。若为删除操作，则依次比对四个地址中的 MAC 地址是否与要删除的 MAC 地址相同，若相同则删除该表项，若都不相同，则不操作。 若为更新操作，依次比对四个地址中的 MAC 地址是否与要更新的 MAC 地址相同，若相同且 valid 为 1 则对该表项进行更新。若没有相同的，则不操作。

3) 组播查找模块

由于该查找模块的实现与单播转发表模块中的学习查找模块中的查找过程基本相同，此处就不在对该模块进行介绍了。

4) 组播地址表

组播地址表采用双口 RAM 来实现，深度为 256，位宽为 76bit。结构见图 4.8 所示：

75 69	68 45	44 33	32	31 0
保留	multi_mac_00	multi_group_00	valid	member_id_00
保留	multi_mac_00	multi_group_00	valid	member_id_00
保留	multi_mac_00	multi_group_00	valid	member_id_00
●●●●●●				

图4.8 组播地址表

各 bit 的定义如下：

- 0~31：组播成员 ID 号码。
- 32： 有效位。
- 33~44：组播组号。
- 45~68：组播 MAC 地址。
- 69~75：预留。

4.1.4 vlan 转发表模块的具体实现

1) vlan 相关的概念

VLAN(Virtual Local Area Network 虚拟局域网)是把一个LAN（Local Area Network 局域网）分为若干个逻辑子网。VLAN 技术有很多优点[14] ，具体如下：

（1）控制广播风暴。由于二层广播只在 VLAN 内进行，不会扩散到其他的 VLAN 上，合理的对 VLAN 进行划分将会缩小广播的范围，减小带宽的占用。如图 4.9 所示：

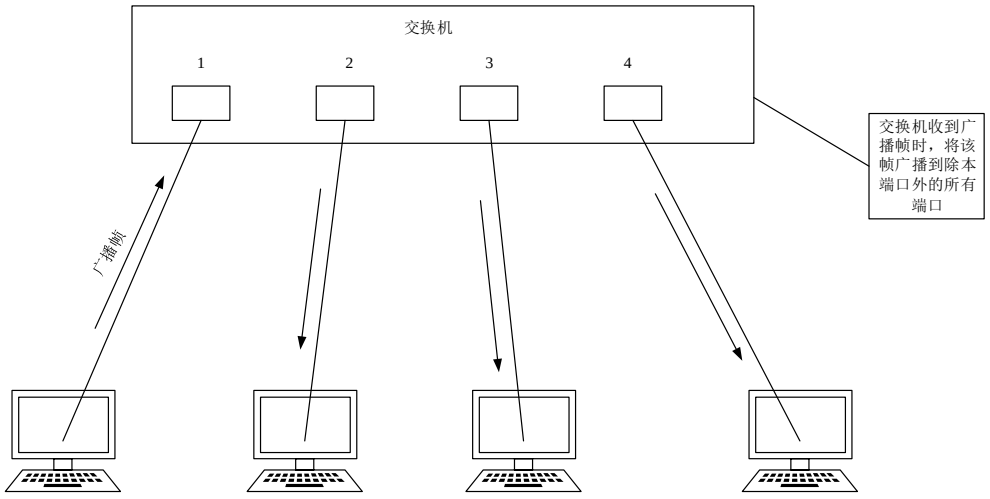

图4.9 广播风暴抑制

（2）提高网络的安全性。可以使同一个 VLAN 内的用户与其他 VLAN 内的用户隔离。这样就可以避免某些非法用户的入侵。

（3）提高管理效率。 对于非 VLAN 网络，若网络中的站点发生变动，则需要重新改变布局及配置，增大网络管理的工作量 。当采用 VLAN 网络时，当 VLAN 中用户的物理位置改变的的话，只要其还在同一个 VLAN 中，就不需要进行调整。

（4）降低了网络的维护费用。提高网络的监控性能。通过对某个 VLAN 监控信息的采集及分析，能很好的分析网络中流量的分布，缩小检测的范围，能够快速的找到故障。

IEEE802 委员会定义 802.1Q 协议定义了同一 vlan 跨交换机通信桥接的规则以及 vlan 帧的格式。在如表 4.1 及 4.2 所示 802.1Q 数据帧的格式是在以太网数据帧的标准格式中添加一个四字节的 TAG 标记头。添加的 TAG 标记头位于数据帧中“发送源 MAC 地址(Source Addresses , SA)”与“长度或类型(Length/Type , L/T)”之间的位置上[16] 。

表4.1以太网数据帧的标准格式

Pre	SFD	DA	SA	L/T	Data	FCS
7 字节	1 字节	6 字节	6 字节	2 字节	46~1500 字节	4 字节

表4.2 802.1Q 数据帧的格式

Pre	SFD	DA	SA	TAG		L/T	Data	FCS
				TPID	TCI			
7 字节	1 字节	6 字节	6 字节	4 字节		2 字节	46~1500 字节	4 字节

TAG 共有 4 个字节，由 2 个字节的标签协议标志（Tag Protocol Identifier , TPID）和 2 个字节的标签控制信息（TCI）构成。TCI 又由用户优先级（User Priority , P）、规范格式指示器（Canonical Format Indicator , CFI） 以及 VLAN ID 构成。TAG 中各字段的定义如表 4.3:

表4.3 TAG 中各字段的定义

字段	长度 （bit）	取值	定义
TPID	16	0X8100	表明该帧是 802.1Q

				数据帧即 vlan 帧
TCI	P	3	0 到 7	表明该数据帧的优先级别
	CFI	1	0 或 1	0 表示为以太网地址。
	VLAN ID	12	0 到 4095	指示数据帧属于哪个 vlan（1~4094）。当值为 0 时仅用于传输数据帧的优先级，值 4095（0XFFF）保留。

下面介绍一下 VLAN 中一些常见的概念。例如，PVID、交换机的端口类型。

PVID：Port VLAN ID,指端口的缺省 VLAN ID。Hybrid 端口和 Trunk 端口能够支持多个 VLAN，所以需要 PVID 的存在。缺省情况下，Hybrid 端口和 Trunk 端口的缺省 VLAN 为 VLAN 1。PVID 主要作用是对收到的非 TAG 帧添加 PVID 后进行查找转发。

交换机的端口可以分为 Access 端口、Trunk 端口、Hybrid 端口三种。三种端口类型的定义如下。

- Access 端口：access 端口只绑定 1 个 VLAN，一般是与计算机相连的端口。
- Trunk 端口： trunk 端口可以支持多个 VLAN，一般用于交换机之间的连接。
- Hybrid 端口：Hybird 口可以支持多个 VLAN，可以用于交换机的连接，也可用于连接计算机。

当交换机从端口收到数据帧时，不同的端口类型对数据帧的处理如下：

- Access 端口

（1）收到一个二层帧。

（2）判断是否有 VLAN 标签：若没有跳向（3），有则丢弃。

（3）打上端口的 PVID，并进行交换转发。

- Trunk 端口

（1）收到一个二层帧。

（2）判断是否有 VLAN 标签：若没有跳到（3），有则跳到（4）。

（3）打上端口的 PVID，并进行交换转发。

（4）判断该 trunk 端口是否支持该 VLAN 帧：支持则转发，否则丢弃。

- Hybrid 端口

（1）收到一个二层帧。

（2）判断是否有 VLAN 标签：若没有跳到（3），有则跳到（4）。

（3）打上端口的 PVID，并进行交换转发。

（4）判断该 hybrid 端口是否支持 VLAN 帧：允许则转发，否则丢弃。

当数据帧从交换机转发出去时，不同的端口对数据帧的处理如下：

- Access 端口

（1）将数据帧的 VLAN 标签去掉，发送出去。

- Trunk 端口

（1）比较 PVID 和数据帧的 VLAN 标签。

（2）如果相等跳到（3），否则跳到（4）。

（3）去掉 VLAN 标签发送。

（4）直接发送。

- Hybrid 端口

（1）判断 VLAN 在本端口的属性即该端口对哪些 VLAN 是 untag，哪些 VLAN 是 tag。

（2）若是 untag 跳到（3），否则跳到（4）。

（3）去掉 VLAN 标签发送。

（4）直接发送。

2) vlan 转发表模块的结构

在对 VLAN 帧在交换机中的转发规则了解之后进行了关于 VLAN 帧在交换机中转发的设计。

VLAN 转发表的主要模块包括查找模块(vlan_lookup)、vlan 更新模块（vlan_renew）、端口 PVID 更新模块（port_pvid_renew）、vlan 转发表（vlan_table）、port_pvid 转发表（port_pvid_table）。

其内部的结构如图 4.10 所示：

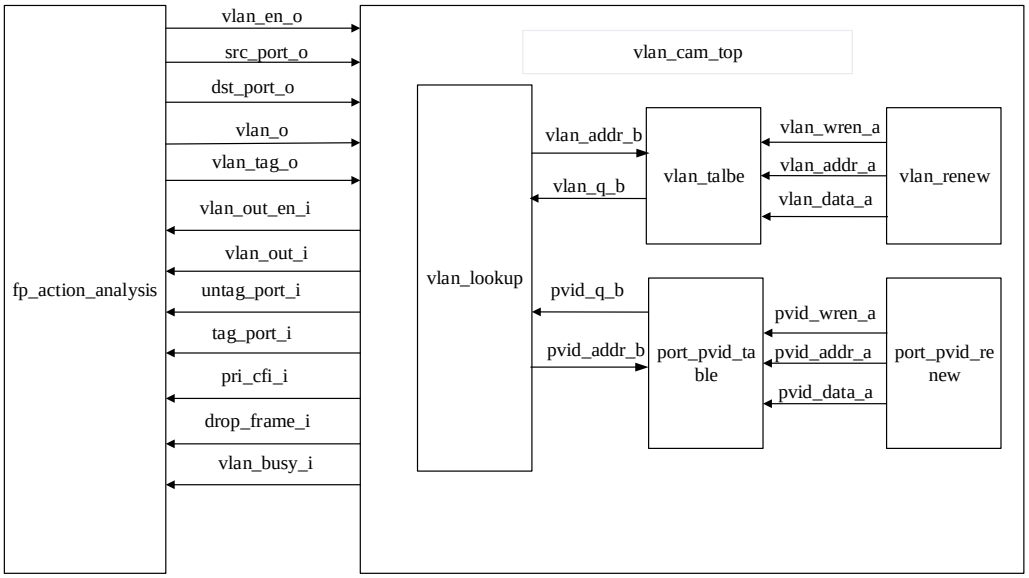

图4.10 vlan 转发表的内部结构

vlan 转发表的主要功能为：fp_action_analysis 模块将源端口号 src_port_o、目的端口号（dst_port_o）、以及正在处理的以太网帧是否为 TAG 帧（vlan_tag_o），若为 TAG 帧则该帧的 VLANID（vlan_o）信号在 vlan_en_o 的时候传送给 vlan_lookup 模块，然后 vlan_lookup 模块根据该信息进行查找并将查找的结果传送给 fp_action_analysis 模块。

3) vlan_lookup 模块

vlan_lookup 是 vlan 转发表模块中的核心模块。该模块的状态如图 4.11：

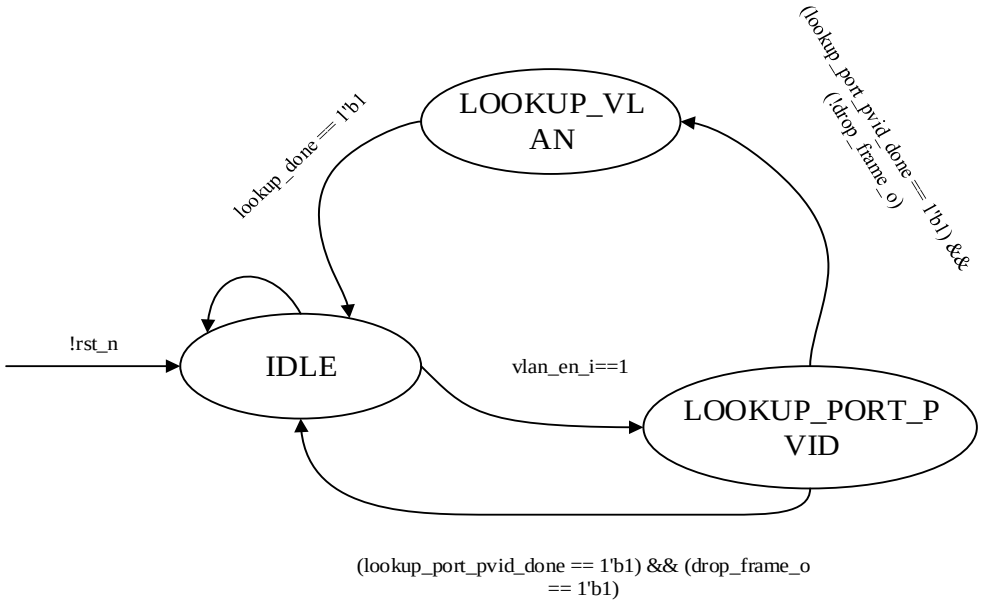

图4.11 vlan_lookup 模块状态转移图

该模块的处理流程如图 4.12：

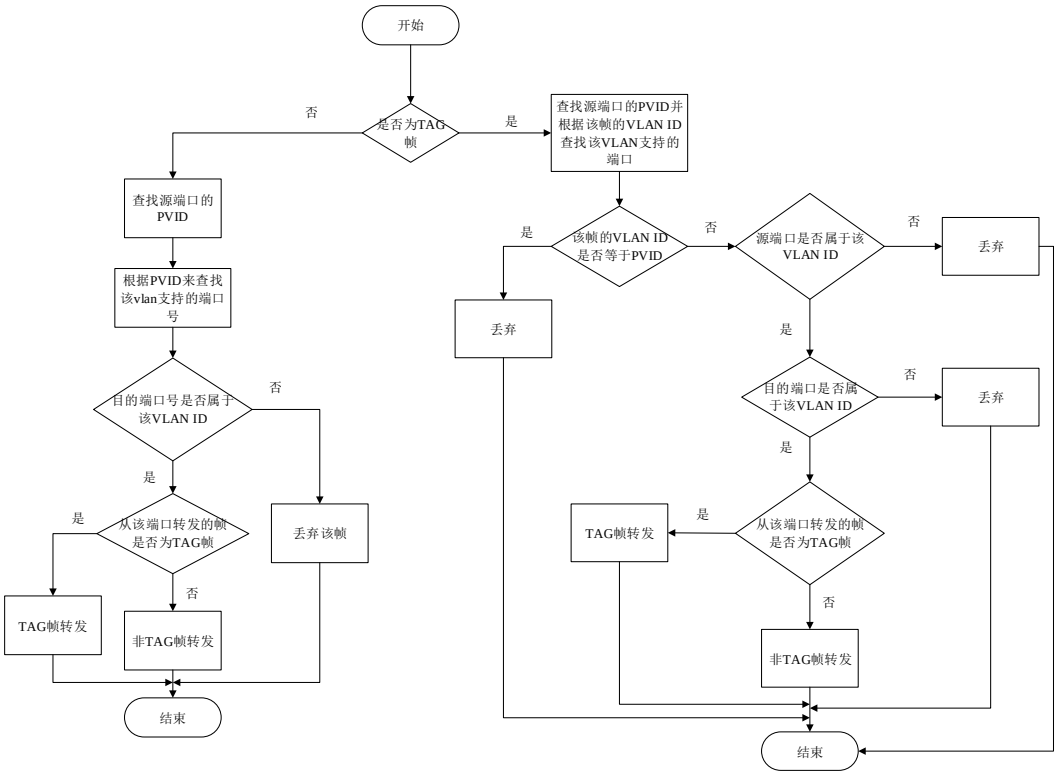

图4.12　　vlan_lookup 模块处理流程

该模块的工作流程

在 vlan_en_i 为 1 时 vlan_lookup 的状态机开始启动，下面本文将根据处理的帧是否为 TAG 帧，将处理流程分为两种来介绍。

- 当处理的帧为非 TAG 帧

（1）　在 vlan_en_i 为 1 时，源端口号（src_port_i）及目的端口号（dst_port_o）传给 vlan_lookup 模块，由于该帧为非 TAG 帧，因此需要获知该端口 PVID。

（2）　PVID 是在状态机的跳转到 LOOKUP_PORT_PVID 时将源端口号作为 port_pvid 转发表的地址来进行查找获得的。

（3）　状态机在 LOOKUP_VLAN 状态时将源端口的 PVID 作为 vlan 转发表的读地址来获取该 VLAN ID 下所支持的端口，以及该 VLAN ID 支持的端口中有哪些端口转发出去的帧是 TAG 帧以及哪些端口转发出去的帧为非 TAG 帧。

（4）　然后将目的端口号与该 VLAN ID 下所支持的端口号进行比较，若目的端口号不是该 VLAN ID 所支持的端口，则查找结束并给出丢弃信号（drop_frame_o）。若是该 VLAN ID 所支持的端口，则判断该端口转发出去的帧是否为 TAG 帧，然后给出查找结束信号。

- 当处理的帧为 TAG 帧

（1） 在 vlan_en_i 为 1 时，源端口号（src_port_i）、目的端口号（dst_port_o）、及该帧的 VLAN ID（vlan_o） 传给 vlan_lookup 模块，由于该帧为 TAG 帧，需要判断一下该 VLAN ID 是否是该端口的 PVID 以及是否被该端口所支持获知该端口的缺省 VLAN ID （PVID）。

（2） PVID 是在状态机的跳转到 LOOKUP_PORT_PVID 时将源端口号作为 port_pvid 转发表的地址来进行查找获得的。以该帧的 VLAN ID 作为 vlan 转发表的读地址，来获取该 VLAN ID 所支持的端口号。

（3） 判断该帧的 VLAN ID 是否等于源端口的 PVID，若二者相等，则查找结束并给出丢弃信号。若二者不相等，则判断源端口是否为该 VLAN ID 所支持的端口，若不是则查找结束，并对该帧执行丢弃操作。若是则判断该 VLAN ID 是否支持目的端口号，如果该目的端口号不是该 VLAN ID 所支持的则查找结束并丢弃该帧，如果该目的端口号为该 VLAN ID 所支持，则判断从该目的端口转发出去时是否为 TAG 帧（由于一个帧可能从不只一个端口转发出去，因此同一个帧从不同端口转发出去有可能为 TAG 帧也有可能为非 TAG 帧），查找结束，给出查找完成信号。

4) port_pvid 转发表模块

port_pvid 转发表模块的主要功能为根据端口号来获得该端口的 PVID，然后以此 PVID 作为从该端口进入的非 TAG 帧的 VLAN ID ，或者是当进入的为 TAG 帧时则起到过滤的作用。根据设计要求本交换机共有 12 个端口，因此本方案中将 port_pvid 转发表的深度设为 16。

port_pvid 转发表是一块连续的存储单元，一端跟更新模块相连，另一端与 vlan 查找模块相连。port_pvid 转发表的结构如图 4.13 所示：

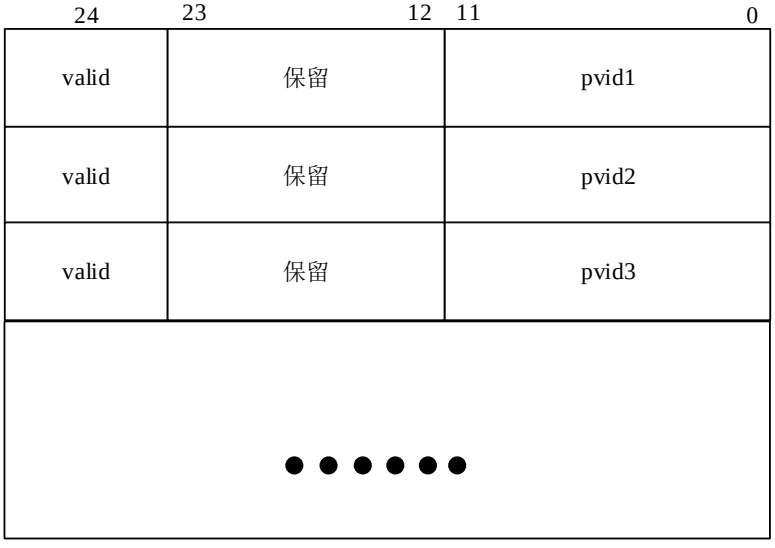

图4.13 port_pvid 转发表的结构

双端口 RAM 的深度为 16，位宽为 25bit，各 bit 的定义如下：

0~11：由于帧进入交换机时的端口号即为该转发表的地址，故其对应的表项中的 PVID 即为该端口的 PVID。

12~23：保留。

24： 指示该表项是否有效。

5) vlan 转发表模块

由于本交换机只有 12 个端口，故而初步设定该交换机所支持的 VLAN ID 的个数为 32 个。vlan 转发表模块的功能是根据 VLAN ID 来通过查表获得该 VLAN ID 所支持的端口号以及从该端口转发出去的是否为 TAG 帧。

vlan 转发表是一块连续的存储单元，一端跟更新模块相连，另一端与 vlan 查找模块相连转发表的结构如图 4.14 所示：

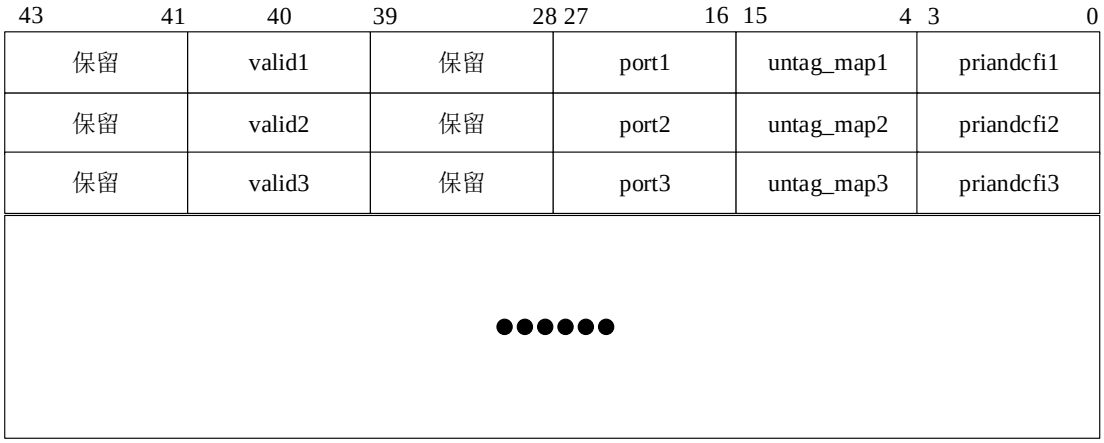

图4.14 vlan 转发表

双端口 RAM 的深度为 32，位宽为 44bit，各 bit 的定义如下：

0~3：该 VLAN ID 下默认的优先级及规范格式指示器的值。

4~15：该 VLAN ID 下支持哪些端口转发为非 TAG 帧。

16~27：属于该 VLAN ID 的端口。

28~39：保留。

40：有效位。

41~43：保留。

6) vlan 更新模块

本方案中支持 vlan 表的动态配置，更新模块主要功能为：当 CPU 接口中 vlan 信息寄存器使能信号使能后，从 CPU 接口模块中依次读取两个 vlan 信息寄存器中的内

容，然后根据更新的地址对 vlan 表项的内容进行更新或删除。该状态机跳转如图 4.15：

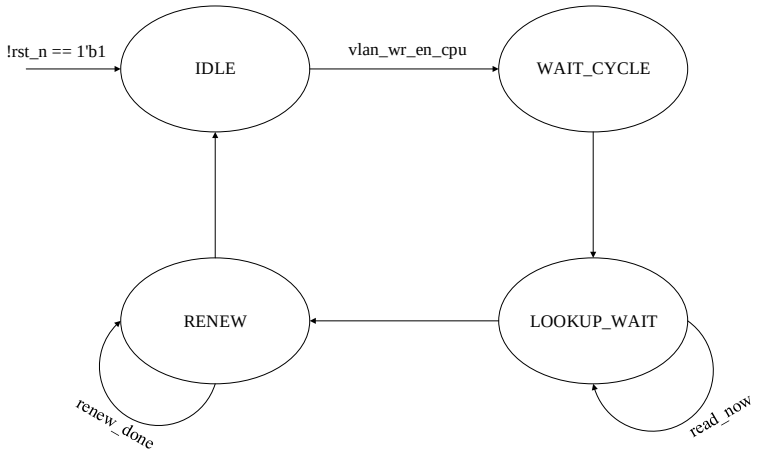

图4.15　　vlan 更新模块状态转移图

状态机的功能如下所示：

Idle：检测 CPU 接口模块的 vlan_wr_en_cpu 信号，若该信号为高时则跳到 WAIT_CYCLE 信号并读取 2 个寄存器中的内容。

WAIT_CYCLE：延迟一个周期，来判断将要更新的地址与当前正在查找的地址是否冲突。

LOOKUP_WAIT：若查找地址是即将要更新的地址，则更新模块等待，直到查找结束。

RENEW：在当前状态下，根据 vlan_table_value_1_cpu 寄存器中的 28~29bit 的值来判断本次的操作是对当前表项进行更新还是对当前表项的内容进行删除。当更新完成后将 renew_done 信号置高并将状态跳转到 IDLE 状态。

7)　端口 PVID 更新模块

本模块的主要功能是读取 PVID 信息的寄存器中的内容，然后对 port_pvid 转发表中的相应的表项进行更新或删除。该模块中的状态机的跳转与 vlan 更新模块中的状态机跳转及功能基本相同，具体请参照 vlan 更新模块中的状态跳转的过程，在此就不再赘述。

4.2 VL 转发模块的具体实现

4.2.1　VL 转发模块的总体结构

VL 转发模块主要包括 VL 查找表模块，过滤模块，警管模块。该模块的主要功能是根据分组处理模块传来的 VL 值、Constant 域及数据帧长来对数据帧进行过滤及警管。由于每个端口对应一个 VL 转发模块，因此分组处理模块首先根据数据帧的源

端口号来将该数据帧的帧信息传递给相应的 VL 转发模块。VL 转发模块的结构如图 4.16 所示：

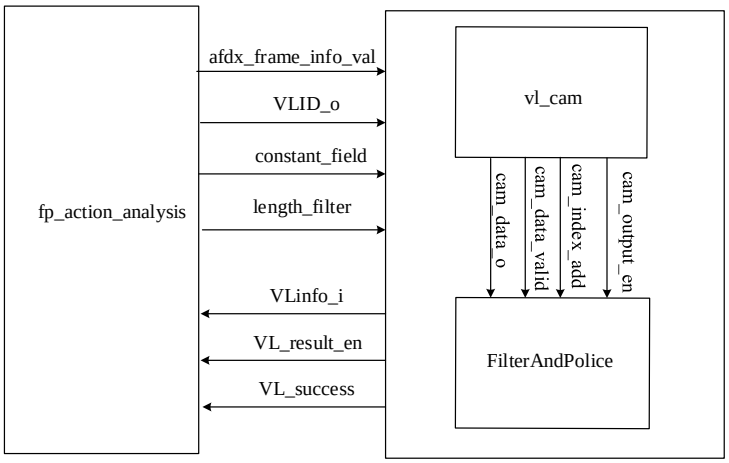

图4.16　　VL 转发模块的结构

VL 模块与 fp_action_analysis 模块的管脚定义如表 4.4：

表4.4 VL 模块与 fp_action_analysis 模块的管脚定义

序号	名称	位宽（bit）	来源	定义
1	afdx_frame_info_val	1	fp_action_analysis 模块	信息有效信号，高有效
2	VLID_o	12	fp_action_analysis 模块	数据帧的 VL 值
3	constant_field	32	fp_action_analysis 模块	数据帧中目的 MAC 地址中的 constant 域
4	length_filter	11	fp_action_analysis 模块	数据帧的长度
5	VLinfo_i	23	VL 转发模块	VL 转发模块产生的信息
6	VL_result_en	1	VL 转发模块	VL 查找结果有效信号
7	VL_success	1	VL 转发模块	VL 查找成功信号

4.2.2 VL 查找表模块

ARINC664 规范中每个端口可以支持 256 条 VL，一个 AFDX 交换机可以支持 4096 条 VL，但是端口支持的 VL 的值是不确定的。因此，本设计中采用两级的查找表：第一级为 RAM_index，RAM 深度为 4096 对应于 4096 条 VL，表项中的内容为下一级查找表的地址及该地址有效指示位。第二级中为 RAM_content，表项中的内容为该 VL 所相关的参数信息。RAM_content 的结构如图 4.17 所示：

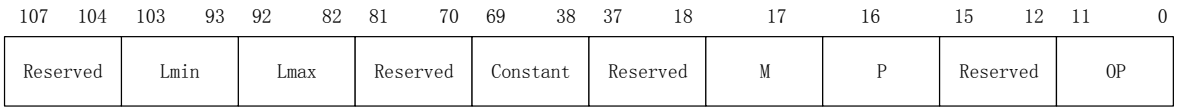

图4.17 RAM_content 的结构

各 bit 的定义如下：

0~11：输出端口号。

16：优先级。

17：多播指示位。

38~69：Constant 域。

82~92：数据帧的最大长度。

93~103：数据帧的最小长度。

4.2.3 过滤及警管模块

1) 过滤模块

过滤模块主要的作用是根据该 RAM_content 中的表项的内容即 VL 的参数信息，来对收到的数据帧进行过滤。过滤的依据主要是数据帧长及 Constant 域。例如，根据数据帧的长度与 RAM_content 中的表项的 Lmax 及 Lmin 进行比较，若数据帧的长度大于 Lmax 或小于 Lmin 则该数据帧丢弃[15] 。

2) 警管模块

常用的警管包括基于字节的警管及基于帧的警管。其中基于字节的警管的 ACi（Account for VLi）每次减少的值为将要转发的数据帧的字节数 S，而基于帧的警管为 ACi 每次减少的值为一个固定的最大值。本设计中采用的是基于帧的警管。警管模块中的存储的深度为 256，位宽为 76bit，其结构如图 4.18 所示：

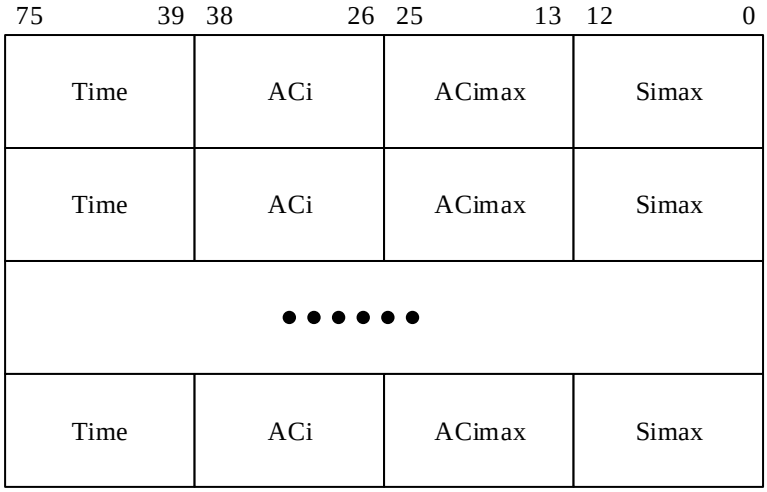

图4.18　警管模块的存储结构

各个 bit 的定义如下：

0~12：Simax，即该 VL 下收到一个数据帧时减少的值。

13~25：ACimax，即该 VL 的计数器的最大值。

26~38：ACi，指上一个帧到达后剩余的值。

39~75：Time，指上一个帧到达时的时间。

根据 ARINC 664 协议给出的公式 ACimax=Simax*(1+Ji /BAGi)，可以推知计数器的上限值 ACimax。由于 BAGi 及 Ji 的值可以自己配置，因此确定了 Simax 则 ACimax 的值也就确定了。由于数据帧到来时不管数据帧的实际帧长而是减去 Simax 的值。因此 Simax 的取值就很随意了，当 Simax=BAGi 时公式可化简为 ACimax=（BAGi+Ji）。由于 BAGi 和 Ji 都是时间，ACi 的值就能用时间来表示。

该模块的工作流程为：警管模块中有一个计数器 ACi，上电的复位后，ACi 的值为 ACimax，当数据帧到达时，根据 VL 查找表模块给出的地址获得警管模块中的表项，将警管模块中的计数器的值减去表项中记录的上一个数据帧到达时的时间，将该值记为 para。将 para 与表项中的 ACi 相加赋给 para，将该值与 ACimax 相比较，若大于等于 ACimax，将 ACimax 的值赋给表项中的 ACi（由于 ACi 的最大值为 ACimax），并给出警管成功信号。若 para 小于 ACimax 且大于 Simax，则将 para 与 Simax 的差值赋给表项中的 ACi，并给出警管成功信号。若 para 小于 Simax，则将 para 的值赋给表项中的 ACi 并给出警管失败信号。

4.3　以太网转发表模块的功能仿真及验证

转发表模块主要是根据帧信息给出目的端口使数据帧能够正常的转发。单播表的

表项为自学习获得，多播表以及 vlan 表为的表项通过 CPU 端口配置获得。本文所做的仿真主要是关于单播转发表的学习查找老化的过程，以及 vlan 表的查找。

对单播转发表的功能仿真及验证：

如图 4.19 所示：该仿真为转发表上电复位后，表项为空时的学习查找情况，在地址使能时（mac_addr_en）为高时将源 MAC 地址（mac_sour）0X654321030201，源端口号（port_sour）0X001 以及目的地址（mac_dest）0X020103040506 传递给 hash 映射模块，经过四个周期后源 MAC 地址的映射值（hash_sour）0XA4，目的 MAC 地址的映射值（hash_dest）为 0X31，且在映射使能（hash_en）时将映射值传递给学习查找模块。在学习查找模块中当 hash_en 为高时进行地址表的学习。以源 MAC 地址为例，四个表项地址分别为{hash_sour，2'b00}、{hash_sour，2'b01}、{hash_sour，2'b10}、{hash_sour，2'b11}。因为源 MAC 地址的映射值为 0XA4，因此表项学习的地址（addr_a）的高 8 位的值为 0XA4，为了方便查看，在图 4.20 中，将 addr_a 的高 8 位拿出来构成一个总线（hash_add），hash_add 的值为 0XA4，可知学习的地址正确。四个学习的地址分别为 0X290、0X291、0X292、0X293。依次从 4 个地址中读出数据发现与源 MAC 地址 0X654321030201 皆不匹配，即关于该 MAC 地址的表项之前不存在，需要新建一个表项。比对地址表中的有效位，可知地址 0X290 中为空表项，因此将新建的表项在写使能（wren_a）为高时将表项 0X01950c840c0804007ff 写入地址表中。

查找过程是根据目的 MAC 地址（mac_dest）0X020103040506 经过 hash 映射后的值（hash_dest）0X31 来作为单播表地址的高八位来查找，相应的 4 个地址为 0X0c4、0X0c5、0X0c6、0X0c7，依次从 4 个地址中读出表项的内容，根据目的 MAC 地址进行匹配，结果未能匹配成功。因此需要将该数据进行广播，在输出使能（outport_en）拉高时将输出端口号（outport）0Xfff（注：广播应该是将数据帧发往除源端口之外的所有端口，本设计中在学习查找模块中给出的广播是包括源端口的，在 fp_action_analysis 模块生成调度信息的时候将广播中包含的源端口去掉。），输出到 fp_action_analysis 模块。

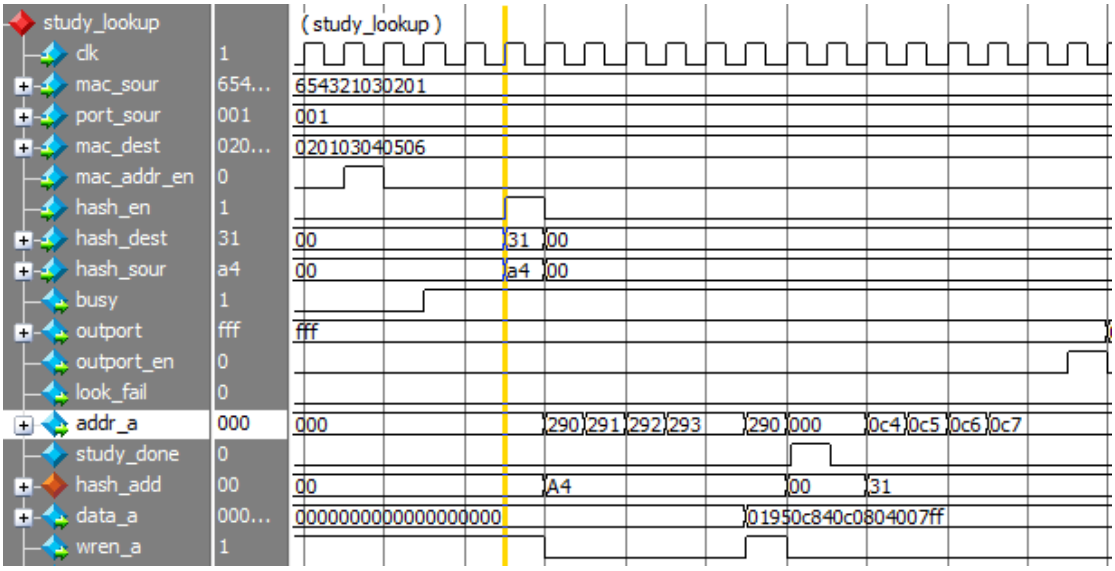

图4.19　　单播转发表第一次学习及查找 1

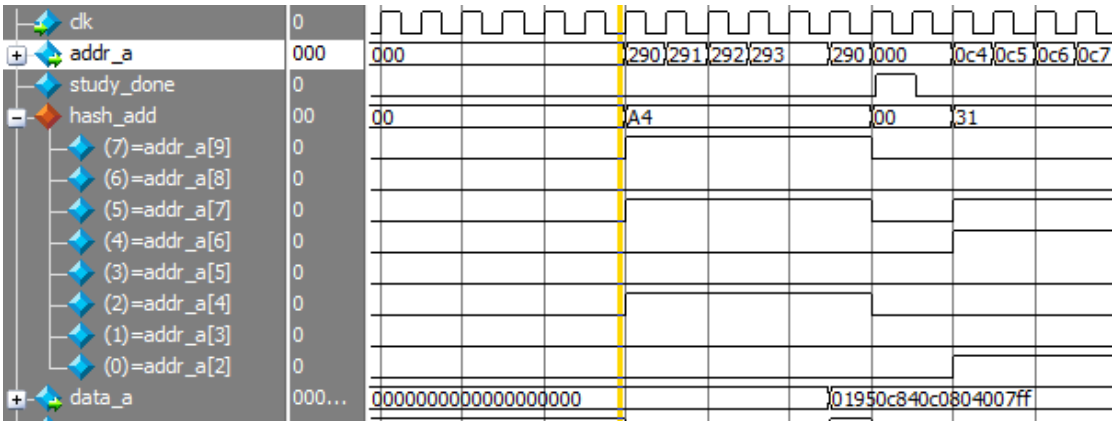

图4.20　　单播转发表第一次学习及查找 2

在经过上面的学习后地址表中有一个 MAC 地址 0X654321030201，源端口号（port_sour）0X001 的表项，下面为了验证地址表的查找功能，激励产生一个目的 MAC 地址为 0X654321030201 的数据帧，此时输出的端口号应为 0X001。如图 4.21 所示：源 MAC 地址为 0X020103040506，源端口号为 0X040，目的 MAC 地址为 0X654321030201，源 MAC 地址与目的 MAC 地址经过 hash 映射后的值分别为 0X31，0Xa4。源 MAC 映射的 4 个表项地址分别为 0X0c4、0X0c5、0X0c6、0X0c7，根据表项中的内容与源 MAC 地址进行匹配，未能匹配成功，在地址为 0X0c4 写入刚学习的表项。目的 MAC 地址所对应的 4 个地址分别为 0X290、0X291、0X292、0X293，从这 4 个地址中读出表项与目的 MAC 地址相匹配，在地址为 0X290 的表项中匹配成功，输出的端口号为 0X001。与所希望的结果相同。

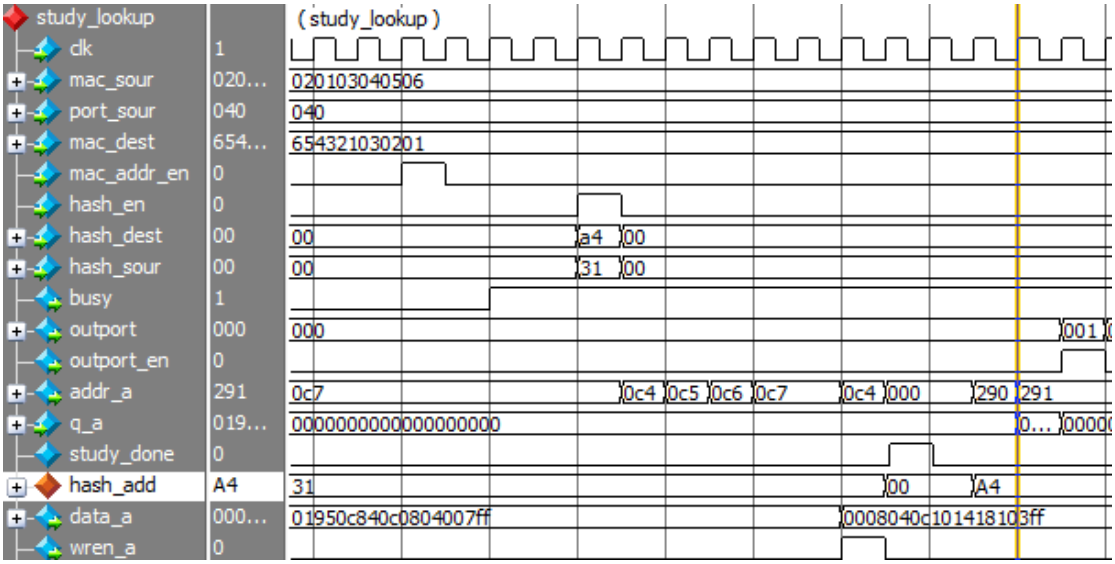

图4.21 单播转发表的第二次学习及查找

在地址表项建立的时候，表项中记录着该表项的建立的时间，该建立时间是当前时间减一。老化删除模块负责将长时间没用的表项移除。如图 4.22：根据地址 0X290 中的低 9 位，为了方便观察，将表项中的低 9 位拿出来建立一个总线（time_table），可知该表项中的时间为 10 与当前时间（timer）相等，因此在写使能（wren_b）为 1 时将该地址中的表项清空。

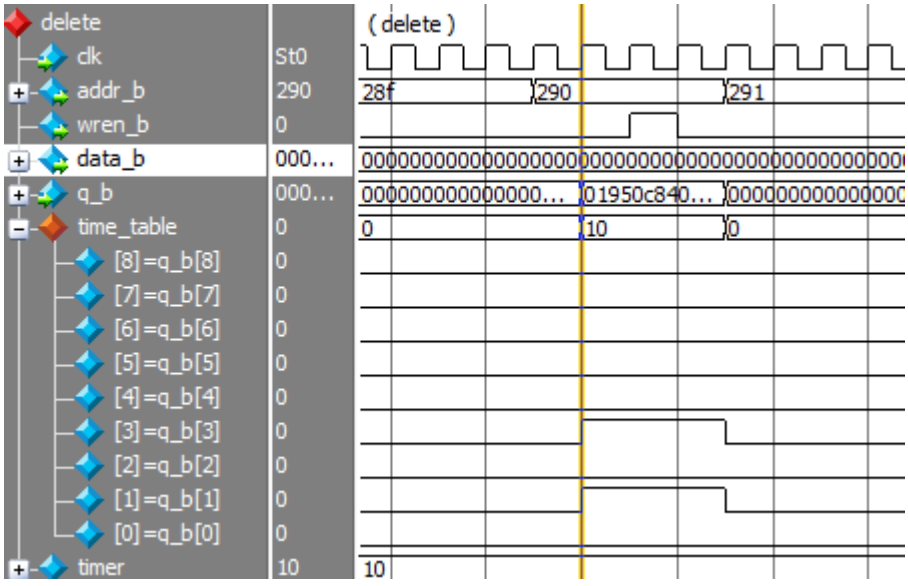

图4.22 老化删除仿真

vlan 转发表模块的功能仿真及验证：

vlan 转发表中的 vlan 转发表与 port_pvid 转发表均由 CPU 端口配置，为验证专用交换机的 vlan 功能，为了简单方便，仅给出我们使用到的 5 个端口（端口 1、端口 7）的配置。如表 4.5：

表4.5 端口及相应的 PVID

端口号	端口缺省 VLAN ID （PVID）
1	1
2	1
3	2
7	2
8	4

表4.6 VLAN ID 支持的端口

VLAN ID	支持的端口	转发为非 TAG 帧的端口	优先级
1	1、2、7	1、2	1
2	3、7	3、7	2
4	7、8	8	4

由表 4.6 中可知，端口 1 为 access 口，PVID 为 1。端口 2 为 access 口，PVID 为 1，端口 3 为 access 口，PVID 为 2。端口 7 为 trunk 口，支持值为 1、2、3 和 4 的 VLAN ID，PVID 为 2。端口 8 为 access 口，PVID 为 4。本文主要做了以下的测试：

（1） 从 access 口进入的数据帧。

（2） 对数据帧的丢弃。

（3） 从 trunk 口进入的数据帧。

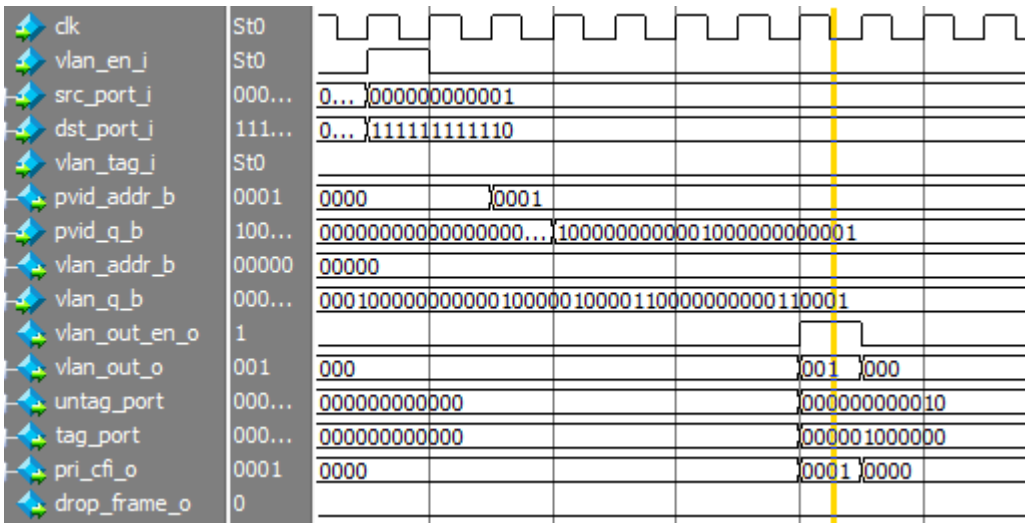

图4.23 access 口进入的数据帧

在 vlan 查找使能（vlan_en_i）时将数据帧的源端口号，经过单播转发表或多播转发表查找后的目的端口号，数据帧是否为 TAG 帧以及相应的 VLAN ID 等传递给 vlan 转发表。由图 4.23 可知，该数据帧的源端口号（src_port_i）为 1，目的端口号（dst_port_i）为 12'b111111111110， TAG 帧标记（vlan_tag_i）为低可知该数据帧为非 TAG 帧。根据源端口号查找该端口的 PVID，即将源端口号的值 4'b0001 作为 port_pvid_table 的地址（pvid_addr_b），由该地址中的值（pvid_q_b）的低 12 位可知该端口的 PVID 为 1，然后根据 PVID 来查找该 VLAN ID 所支持的端口号。由于 vlan_table 中的内容是从地址 0 开始存储的，因此以比 PVID 小 1 的值作为 vlan_table 的地址（vlan_addr_b），然后由该地址中存储的内容可获得以下信息：本 vlan 下所支持的端口号 12'b000001000011 可知 VLAN ID 为 1 时所支持的端口为 1，2，7。然后与目的端口号 12'b111111111110 相比对可知，数据帧转发出去的端口为 2 和 7。在与从该 vlan 下转发出去时为 UNTAG 的端口号 12'b000000000011，即在该 vlan 下从端口 1 和 2 中转发出去时为非 TAG 帧。该数据帧转发端口为 7 时转发出去的数据帧为 TAG 帧，该数据转发端口为 2 时转发出去的数据帧为 UNTAG 帧。即图 4.23 中的 tag_port 值为 12'b000001000000，且 VLAN ID 的值（vlan_out_o）为 0X001，untag_port 的值为 12'b000000000010。在 vlan 转发表输出使能为 1 时（vlan_out_en_o）将 vlan 转发表查找的结果传递给 fp_action_analysis 模块。

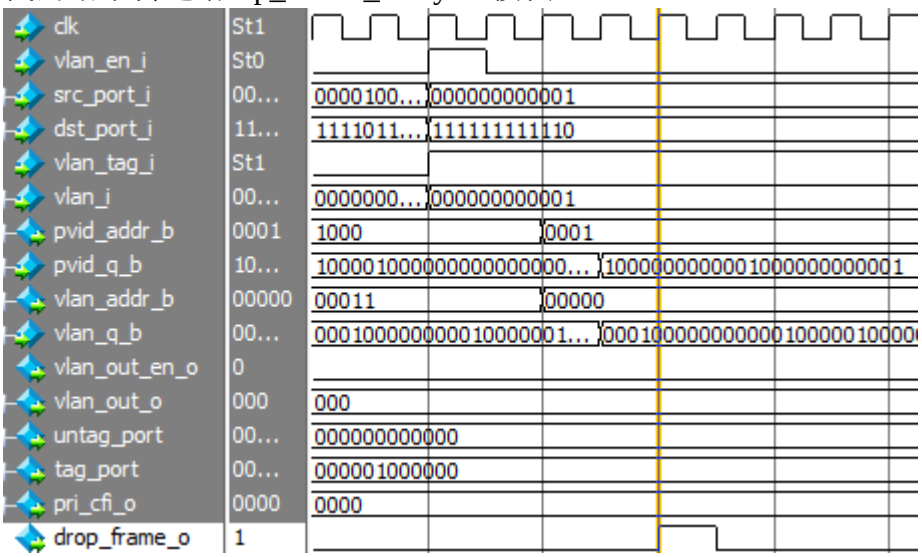

图4.24　　数据帧的丢弃 1

如图 4.24 所示：源端口号为 1，数据帧为 TAG 帧且 VLAN ID 的值为 0X001，以该端口号为地址查找 pvid_table，获得该端口的 PVID 为 0X001，因为 PVID 的值与数据帧的 VLAN ID 相等，因此将该帧丢弃，图中的丢弃信号（drop_frame_o）为 1，满足所需的功能。

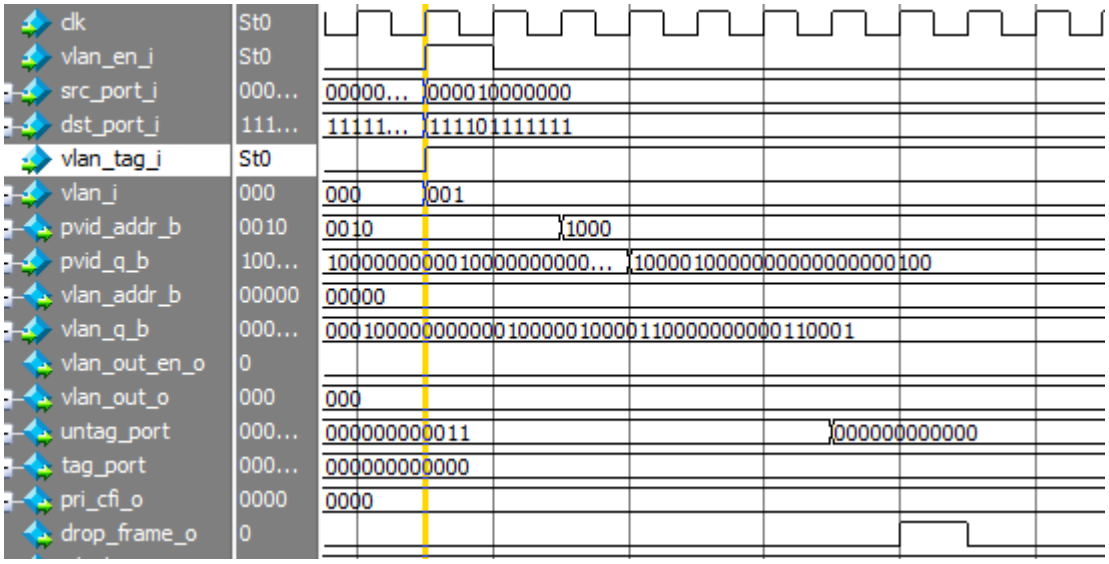

图4.25　　数据帧丢弃 2

如图 4.25 所示：源端口号为 8，数据帧为 TAG 帧且该数据帧的 VLAN ID 为 0X001，以源端口号为 pvid_table 表的地址查找，获得端口 8 的 PVID 为 4，PVID 与数据帧的 VLAN ID 不相同。然后根据数据帧的 VLAN ID 进行 vlan_table 表查找获得 VLAN ID 为 1 所支持的端口号为 1，2，7。可知端口 8 并不支持该 VLAN ID。因此将该数据帧丢弃，与预期相符。

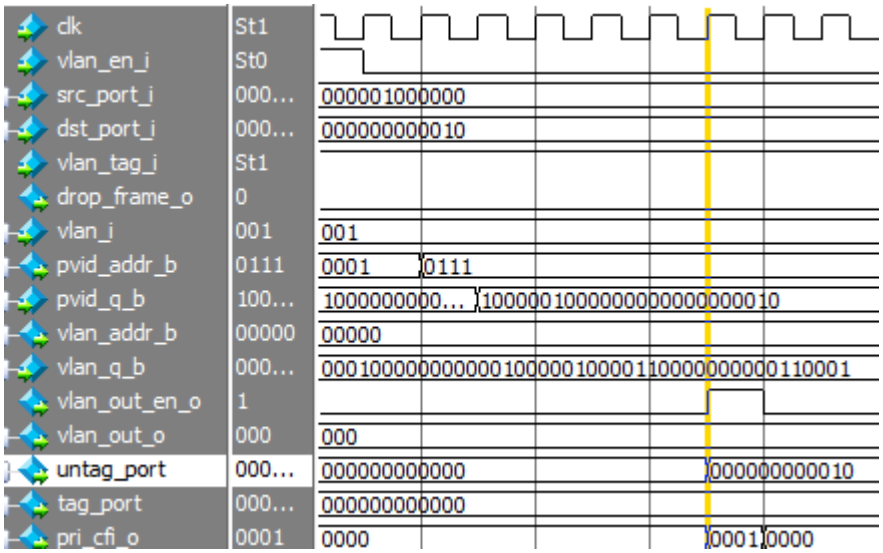

图4.26　　从 trunk 口进入的数据帧

如图 4.26 可知，从端口 7 输入的数据帧为 TAG 帧且 VLAN ID 为 1。根据源端口号查找获得该端口的 PVID 为 2，与 VLAN ID 不同，通过 VLAN ID 查找获知该 VLAN ID 支持的端口号为 1，2，7，且从端口 1，2 转发的数据帧为 UNTAG 帧。可

知该数据帧能被允许从 7 口进入，又因为该数据帧的目的端口为 2，所以该数据帧转发出去时为 UNTAG 帧。

4.4 VL 转发模块的功能仿真及验证

VL 转发模块主要是根据 fp_action_analysis 模块传来的 VL 值，Constant 域以及数据帧长，对 AFDX 帧进行过滤，查找目的端口，以及对该 VL 下的数据帧进行警管操作。

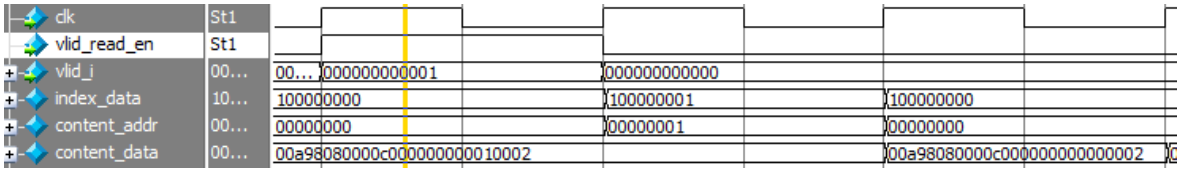

图4.27 VL 查找表模块仿真

由图 4.27 所示，在 VL 值使能（vlid_read_en）为高时将 VL 值（vlid_i）0X001 读入 VL 查找表模块，将 VL 值作为 index_ram 的读地址，在下一个周期读出表项内容（index_data）9’b100000001，由于最高为为 1 表明该值有效，然后用低 8 为 8’b00000001 为 content_ram 的读地址（content_addr）在下一个周期获得表项中的内容（content_data）。与预期相符。

过滤模块根据 content_ram 中的内容，与 fp_action_analysis 传来的 Constant 域，数据帧长进行比较，若不满足条件给出过滤失败信号。

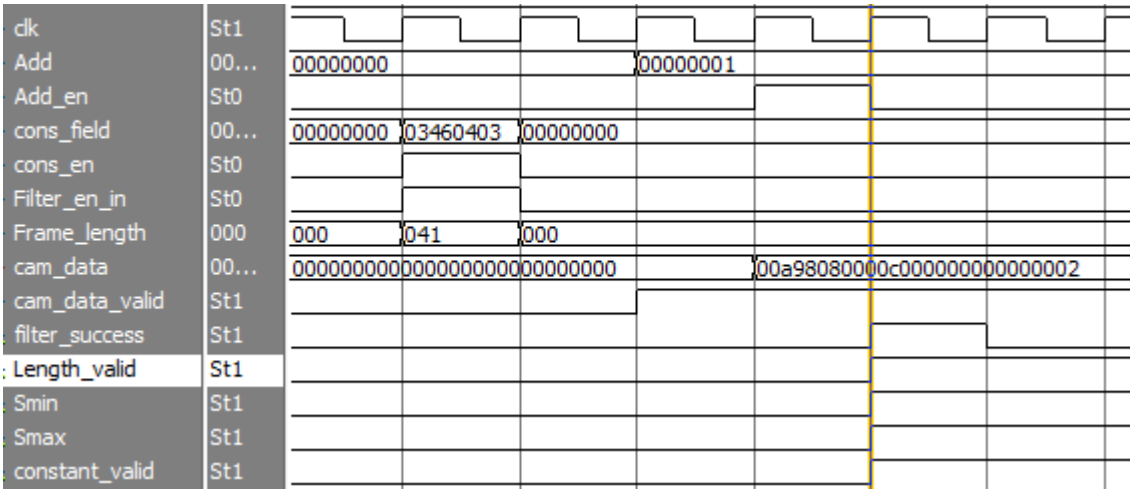

图4.28 VL 过滤模块仿真

如图 4.28 所示：由 VL 查找模块传来的数据 cam_data，与在过滤使能（Filter_en_in）

为高时接收的constant域（cons_field）值0X03460403以及帧长（Frame_length）0X041，比较的内容为长度有效（Length_valid），最小值比较（Smin），最大值比较（Smax）。比较的结果为都满足，输出过滤成功信号（filter_success）给警管模块。

警管模块确定每条VL的带宽，本设计中采用的是基于帧的警管，即当数据帧到来时比较当前的ACi值是否大于Simax，若大于等于则警管成功，允许该数据帧通过，否则警管失败，丢弃该帧。当前ACi的值为由上一个帧到来后剩余的ACi与从上一个帧到当前帧到来时新增的ACi值的和。如图4.29所示：本设计中采用的主时钟（clk）周期为10ns，counter为警管中的计数器，每经过20us则加1。当前counter的值与表中记录的上一个帧到达时的计数值的差值即为新增的ACi值。

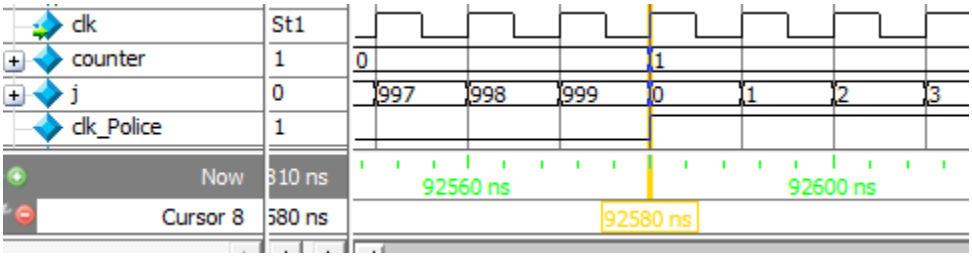

图4.29　新增ACi值

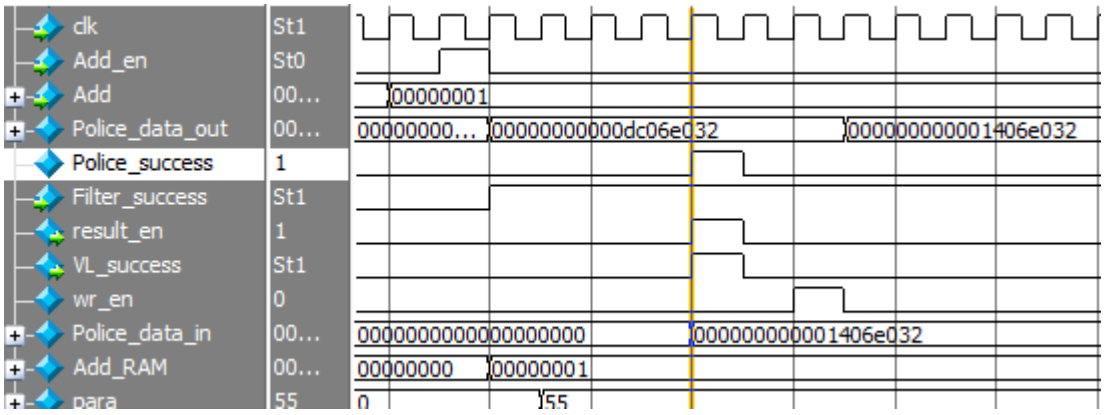

图4.30　警管成功仿真

在对警管模块仿真时，将VL1的BAG设为1ms，jitter的值为100us。在本设计中计数器的精度为20us，在一个BAG内计数器增加的值为50，这个值即为Simax，由公示ACimax=Simax*(1+Ji /BAGi)，可知ACimax的值为55。如图4.30所示：在地址使能时（Add_en）读取（Add）地址中的相应VL警管信息，获取当前的ACi的值（para）为55，大于Simax的值50，可知该数据帧警管成功（Police_success为高），将警管成功信号（Police_success）与过滤成功信号（Filter_success）相与得到VL查找成功信号（VL_success）并在查找结果有效（result_en）时将该信号发送给fp_action_analysis模块。并对该地址中的计数值进行更新。

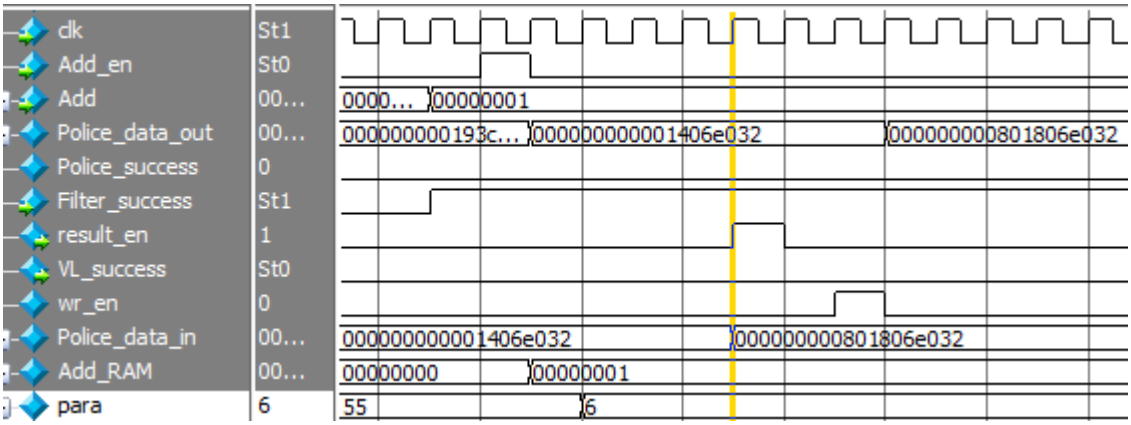

图4.31 警管失败仿真

如图 4.31 所示：当前的 ACi 的值为 6 小于 Simax 的值 50，因此警管失败。在输出有效（result_en）时给出警管失败信号即 Police_success 的值为 0。

第五章 专用交换机板级测试

5.1 以太网数据帧板级测试

5.1.1 以太网数据帧测试环境的搭建

在对专用分组交换机的以太网数据帧板级测试中，根据实验室所拥有的设备，搭建测试环境，如图 5.1 所示：

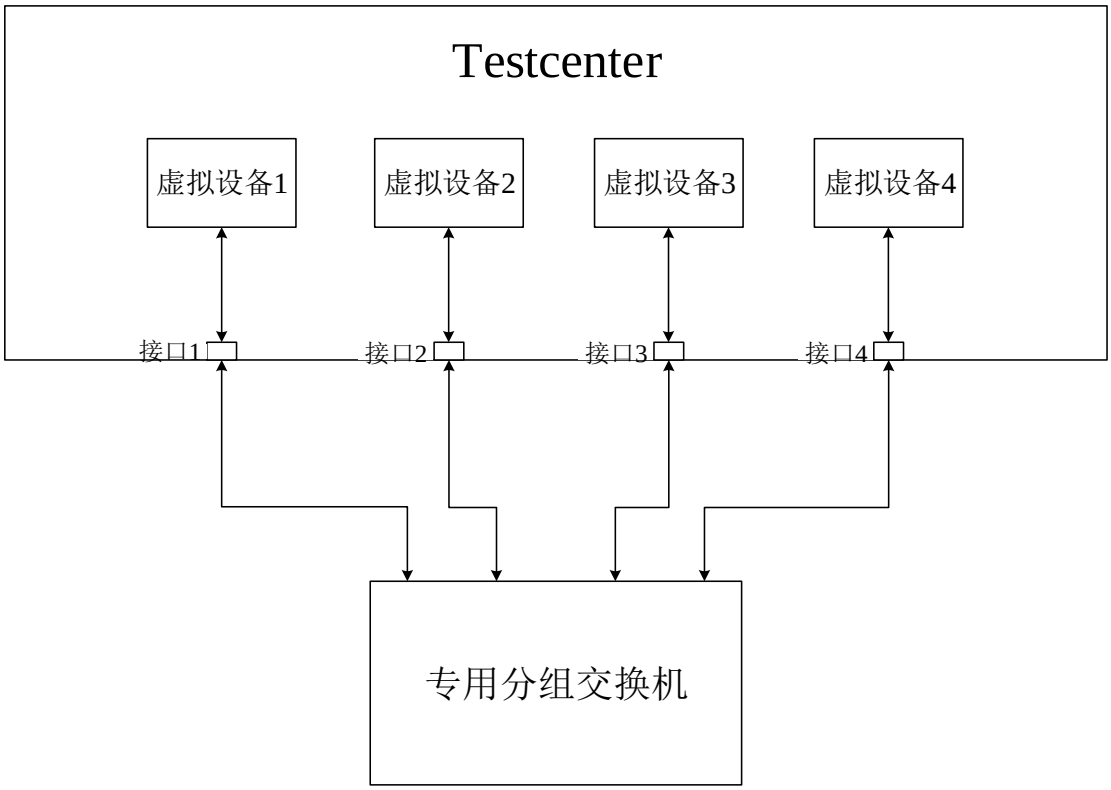

图5.1 专用交换机以太网数据帧测试环境

测试环境组要是由两部分组成：网络测试仪和专用交换机。网络测试仪采用的思博伦公司的 Testcenter，其拥有 24 个以太网接口，支持 10M/100M/1000M 的接口，能够灵活的配置各种数据流。主要用于数据源的产生以及对数据帧接收并进行统计分析。专用分组交换机的实现采用的是 Xilinx 公司的 Zynq 系列的 XC7Z020 芯片。该芯片扩展板上有 4 个以太网接口，将这 4 个以太网口与 Testcenter 的 4 个接口相连。Testcenter 的 4 个接口虚拟 4 个设备产生数据流来对专用交换机进行测试。

5.1.2 测试结果

地址表学习功能测试：

测试步骤如下：

（1） 按照图 5.1 的连接，启动 Testcenter，启动 FPGA 并烧写程序，单播表为自学习，对 vlan 转发模块中的两个表的配置，配置的结果是：端口 1 为 access 口，PVID 为 1，端口 2 位 access 口，PVID 为 1，端口 3 为 trunk 口，支持值为 1，2，3 的 VLAN ID 该端口的 PVID 为 3。端口 4 为 access 口，PVID 为 3。

（2） 将 Testcenter 的 4 个接口启动，都配置在 100M 全双工模式，然后对四个端口分别添加虚拟设备，四个虚拟设备的地址依次为：00:00:00:00:00:01、00:00:00:00:00:02、00:00:00:00:00:03、00:00:00:00:00:04。

（3） 构建数据流 1，数据帧的源 MAC 地址为虚拟设备 1 的地址，目的 MAC 地址为虚拟设备 2 的地址，带宽占用率为 50%，且发送的数据帧为非 TAG 帧，帧长为 64 字节。

（4） 构建数据流 2，数据帧的源 MAC 地址为虚拟设备 2 的地址，目的 MAC 地址为虚拟设备 1 的地址，带宽占用率为 50%，发送的数据帧为非 TAG 帧，帧长为 64 字节。

（5） 开始发送数据流 1，观察各个端口接收数据的结果。如表 5.1 所示：

表5.1 地址表功能测试 1

接口号	Testcenter 数据帧（入）个数	Testcenter 数据帧（出）个数	比特数（入）	比特数（出）
1	0	64346788	0	32945555456
2	64346788	0	32945555456	0
3	64346788	0	35004652672	0
4	0	0	0	0

当专用交换机上电复位后，单播表为空，此时端口 1 发送的非 TAG 帧，经过单播学习后单播表中存有一条关于 MAC 地址为 00:00:00:00:00:01 端口号为 1 的表项，单播查找后得到的目的端口为除源端口外的所有端口，然后进行 vlan 转发表的查找，根据源端口号获知该端口的 PVID，根据 PVID 获得该 VLAN ID 所支持的端口为端口 1，2，3，这些端口中端口 1，2 转发出去的数据帧为非 TAG 帧。因此端口 1 发送的数据能够到达端口 2 及端口 3，且到达端口 2 的数据帧为非 TAG 帧，到达端口 3 的数据帧为 TAG 帧。由数据帧个数及 bit 数的对应关系可知与预期相同。

（6） 停止发送数据流 1，清空统计结果，发送数据流 2，观察各个端口所接收的数据结果。如表 5.2：

表5.2 地址表功能测试 2

接口号	Testcenter 数据帧（入）个数	Testcenter 数据帧（出）个数	比特数（入）	比特数（出）
1	45646674	0	23371097088	0
2	0	45646674	0	23371097088
3	0	0	0	0
4	0	0	0	0

由于之前发送的数据流 1，在单播表中学习到了关于端口 1 与 MAC 地址 00:00:00:00:00:01 的表项。所以在数据流 2 中端口 2 发送的数据帧是目的 MAC 为 00:00:00:00:00:01。在单播转发表中，可以获得目的端口为 1，然后查找 vlan 转发表模块，该端口的 PVID 值为 1，VLAN ID 为 1 时支持的端口号为 1，2，3，转发时为非 TAG 帧时的端口号为 1，2。由于之前单播表查找的目的端口为 1，因此该数据帧输出端口为端口 1 且为非 TAG 帧，根据表中的内容可知与所预期的相同。

输入监控测试：

测试步骤如下：

（1） 按照图 5.1 的连接，启动 Testcenter，启动 FPGA 并烧写程序，配置端口 1 为被监视的输入端口，端口 4 为对被监视的数据帧的输出端口，配置 vlan 转发表，使 4 个端口都同处一个 vlan。

（2） 将 Testcenter 的 4 个接口启动，都配置在 100M 全双工模式，然后对四个端口分别添加虚拟设备，四个虚拟设备的地址依次为：00:00:00:00:00:01、00:00:00:00:00:02、00:00:00:00:00:03、00:00:00:00:00:04。

（3） 构建数据流 1，源 MAC 地址为虚拟设备 1 的地址，目的地址为虚拟设备 2 的地址，带宽占用率为 50%。

（4） 构建数据流 2，源 MAC 地址为虚拟设备 1 的地址，目的地址为虚拟设备 3 的地址，带宽占用率为 50%。

（5） 构建数据流 3，源 MAC 地址为虚拟设备 2 的地址，目的地址为虚拟设备 1 的地址，带宽占用率为 50%。

（6） 构建数据流 4，源 MAC 地址为虚拟设备 3 的地址，目的地址为虚拟设备 1 的地址，带宽占用率为 50%。

（7） 先启动数据流 3 和 4，进行地址表项的学习。

（8） 关闭数据流 3 和 4，清空数据的统计，启动数据流 1 和 2，并观察统计的结果，如表 5.3 所示。

表5.3 监控功能测试

接口号	Testcenter 数据帧个数（入）	Testcenter 数据帧个数（出）
1	0	656443213
2	328221606	0
3	328221607	0
4	656443213	0

由表 5.3 可知监控端口 4 的输出的帧个数等于被监控端口 1 所发送的数据帧的个数，与预期的结果相同。

专用交换机的性能测试：

对交换机性能的测试主要包括吞吐量，时延以及丢包率等。

测试步骤如下：

（1） 按照图 5.1 的连接，启动 Testcenter，启动 FPGA 并烧写程序，配置 vlan 转发表，使 4 个端口都同处一个 vlan。

（2） 将 Testcenter 的 4 个接口启动，都配置在 100M 全双工模式，然后对四个端口分别添加虚拟设备，四个虚拟设备的地址依次为：00:00:00:00:00:01、00:00:00:00:00:02、00:00:00:00:00:03、00:00:00:00:00:04。

（3） 构建数据流 1，源 MAC 地址为虚拟设备 1 的地址，目的地址为虚拟设备 2 的地址，带宽占用率为 100%，发送的数据帧长为 64 字节。

（4） 构建数据流 2，源 MAC 地址为虚拟设备 2 的地址，目的地址为虚拟设备 3 的地址，带宽占用率为 100%，发送的数据帧长为 64 字节。

（5） 构建数据流 3，源 MAC 地址为虚拟设备 3 的地址，目的地址为虚拟设备 4 的地址，带宽占用率为 100%，发送的数据帧长为 64 字节。

（6） 构建数据流 4，源 MAC 地址为虚拟设备 4 的地址，目的地址为虚拟设备 1 的地址，带宽占用率为 100%，发送的数据帧长为 64 字节。

（7） 同时启动数据流 1,2,3,4 进行地址表的学习。

（8） 单播表学习完成后，关闭数据流，清空统计数据，同时启动数据流观察结果。如表 5.4 所示。

表5.4 专用交换机的性能测试 1

数据流	发送数据帧个数	平均时延（us）	最小时延（us）	最大时延（us）	丢包率（%）	发送速率（Mbps）	接收速率（Mbps）
1	594614246	9.62	9.45	9.81	0	100	100

2	594614246	9.62	9.45	9.83	0	100	100
3	594614245	9.62	9.46	9.81	0	100	100
4	594614246	9.62	9.45	9.81	0	100	100

（9） 将 4 个数据流都关闭，清空统计数据，启用数据流 1，带宽占用率为 100%，发送的数据帧长为变长。观察结果，如表 5.5 所示。

表5.5专用交换机的性能测试 2

数据帧长	发送数据帧数	平均时延（us）	最小时延（us）	最大时延（us）	丢包率（%）	发送速率（Mbps）	接收速率（Mbps）
128	335242451	15.23	15.05	15.41	0	100	100
512	644787424	48.79	48.55	49.02	0	100	100
1518	577635473	128.34	127.97	128.79	0	100	100

5.2 AFDX 数据帧板级测试

5.2.1 AFDX 数据帧测试环境的搭建

在对专用分组交换机的 AFDX 数据帧板级测试中，根据实验室所拥有的设备，搭建测试环境，如图 5.2 所示：

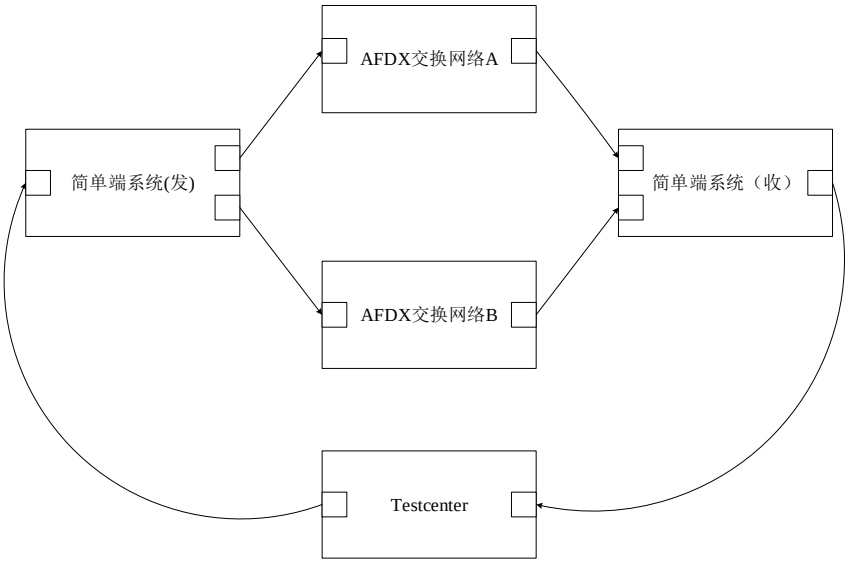

图5.2 专用交换机 AFDX 数据帧测试

测试环境组要是由三部分组成：网络测试仪，简单的端系统及 AFDX 交换网络。网络测试仪采用的思博伦公司的 Testcenter，用于产生数据帧，本测试中的端系统的

功能比较简单只是在发送时根据数据帧中的目的 MAC 地址，判断数据帧所属的 VL 给其添加相应的 SN 并将该 AFDX 帧发往 AFDX 交换网络 A 和 AFDX 交换网络 B。在接收端系统处根据 SN 去冗余，将 SN 去掉后将数据发往 Testcenter。一个根据 AFDX 帧中的 SN 来去冗余。简单端系统及 AFDX 交换网络的实现采用的是 Xilinx 公司的 Zynq 系列的 XC7Z020 芯片。该芯片扩展板上有 4 个以太网接口。

5.2.2 测试结果

VL 查找测试主要做的是关于警管的测试。测试步骤如下：

（1） 按照图 5.2 搭建测试环境，启动 Testcenter，启动 FPGA 并烧写程序。

（2） 将 Testcenter 的 2 个接口启动，都配置在 100M 全双工模式，然后每个端口添加两个虚拟设备，四个虚拟设备的地址依次为：03:46:04:03:00:01、03:46:04:03:00:02、03:46:04:03:00:03、03:46:04:03:00:04。

（3） 构建数据流 1，数据帧的源 MAC 地址为虚拟设备 1 的地址，目的 MAC 地址为虚拟设备 3 的地址，带宽占用率为 10%，帧长为 64 字节。

（4） 构建数据流 2，数据帧的源 MAC 地址为虚拟设备 2 的地址，目的 MAC 地址为虚拟设备 4 的地址，带宽占用率为 10%，帧长为 64 字节。

（5） 同时启动数据流 1 及数据流 2。一段时间后暂停并观察结果，如表 5.6 所示。

（6） 构建数据流 1，数据帧的源 MAC 地址为虚拟设备 1 的地址，目的 MAC 地址为虚拟设备 3 的地址，带宽占用率为 0.1%，帧长为 64 字节。

（7） 构建数据流 2，数据帧的源 MAC 地址为虚拟设备 2 的地址，目的 MAC 地址为虚拟设备 4 的地址，带宽占用率为 0.1%，帧长为 64 字节。

（8） 同时启动数据流 1 及数据流 2。一段时间后暂停并观察结果，如表 5.6 所示。

表5.6 警管测试的结果

VL ID	设置的 BAG	Testcenter 发送的数据帧数 (Packet/sec)	该 VL 能通过的最大数据帧数 (Packet/sec)	实际通过的数据帧数 (Packet/sec)
VL3	32ms	14880.95	31.25	31.25
VL4	1ms	14880.95	1000	1000
VL3	32ms	148.80	31.25	31.25
VL4	1ms	148.80	1000	148.80

当带宽占用率为 10%时，发送的帧长为 64 字节时，每秒发送的数据帧数为14880.95pps，大于 VL3 允许通过的 31.25 pps 也大于 VL4 允许通过的 1000pps。因此每秒实际通过的数据帧数即为相应 VL 允许通过的最大数据帧数。当带宽占用率为 0.1%时，发送的帧长为 64 字节时，每秒发送的数据帧数为 148.80pps，大于 VL3 允许通过的 31.25pps 小于 VL4 允许通过的 1000pps。因此实际通过的数据帧 VL3 为 31.25pps，VL4 通过的数据帧数为 148.80pps。测试的结果与预期的相同。

冗余测试：

（1） 按照图 5.2 搭建测试环境，启动 Testcenter，启动 FPGA 并烧写程序。

（2） 将 Testcenter 的 2 个接口启动，都配置在 100M 全双工模式，然后每个端口对应一个虚拟设备，两个虚拟设备的地址依次为：03:46:04:03:00:01、03:46:04:03:00:03。

（3） 构建数据流 1，数据帧的源 MAC 地址为虚拟设备 1 的地址，目的 MAC 地址为虚拟设备 2 的地址，带宽占用率为 10%，帧长为 64 字节。

（4） 启动数据流 1，观察结果。然后一段时间后将端系统与 AFDX 交换网 B 连接的网线拔掉，观察结果。最后，将端系统与 AFDX 交换网络 A 断开，与 AFDX 交换网络 B 连接，观察结果。如表 5.7 所示：

表5.7 冗余测试

与交换网络的连线状态	VL ID	设置的 BAG	Testcenter 发送的数据帧数 (Packet/sec)	该 VL 能通过的最大数据帧数 (Packet/sec)	实际通过的数据帧数 (Packet/sec)
连接 A 网络，连接 B 网络	VL3	32ms	14880.95	31.25	31.25
连接 A 网络，断开 B 网络	VL3	32ms	14880.95	31.25	31.25
断开 A 网络，连接 B 网络	VL3	32ms	14880.95	31.25	31.25

AFDX 交换网络 A 与 AFDX 交换网络 B 互为备份，一个断开后不影响 AFDX 数据帧的正常传输。观察的结果与预期相同。

5.3 测试问题总结

在对专用交换机进行测试时遇到许多的问题，通过对出现的问题进行分析并通过 chipscope 对相关的信号进行观察，定位问题并解决问题。下面对测试时所遇到的问题进行介绍。

（1） FPGA 内部不存在高阻态：

在设计中我们采用 4 块 XQ4VXS55 来实现，片间的总线如图 5.3 所示：

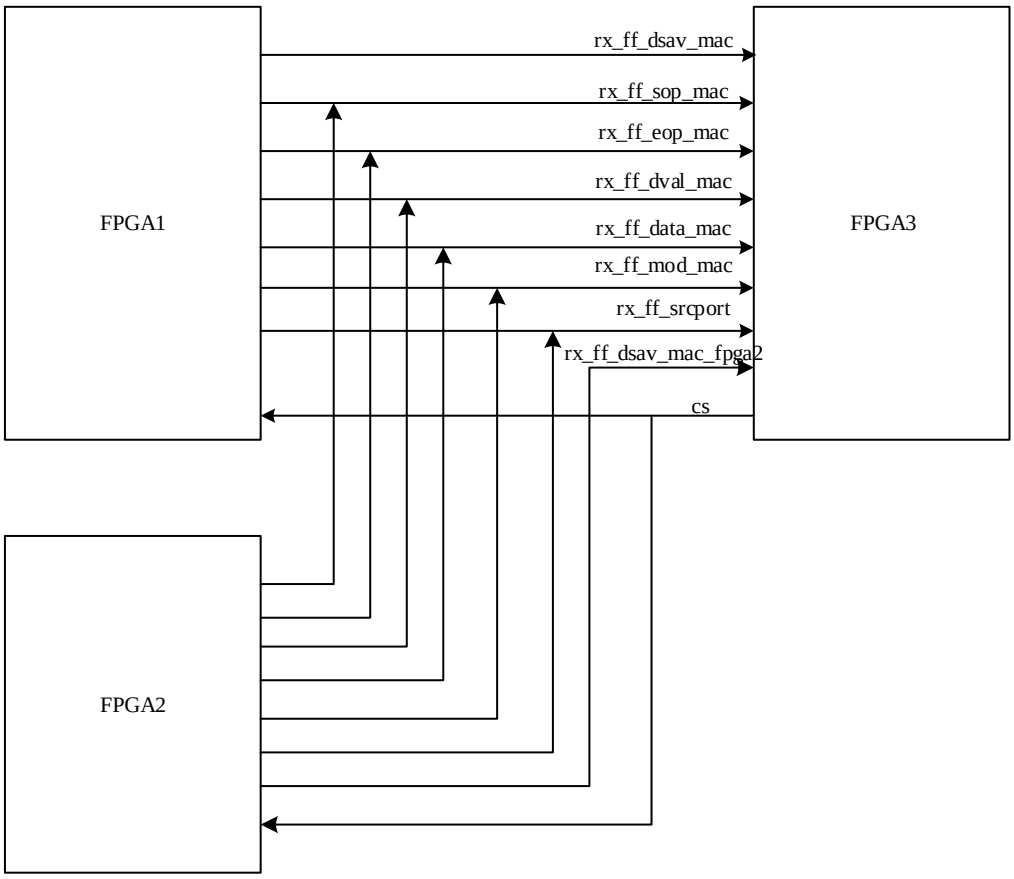

图5.3 FPGA1,2 与 FPGA3 的片间总线

为了节省管脚资源，FPGA1, 3 之间的数据总线与 FPGA2, 3 之间的数据总线是共用的一套。实现的方式为，当 FPGA3 能接受 FPGA1 中的数据时，则向 FPGA1 中发送一个片选选中信号，选中的 FPGA1 发送数据，同时 FPGA2 的数据置为高阻态。然后我们在一片 Zynq 系列的 XC7Z020 芯片上做的板级验证。之前的代码中的片间总线就变成了片内总线了。在板级测试时，通过 Testcenter 发送数据帧，发现接收的数据帧为错误帧，用 chipscope 沿着数据帧在专用分组内的传递方向进行观察，在 MAC IP 处数据帧是正确的，在轮询后发现数据全为 1，定位问题为轮询处。然而用 modelsim 仿真却没有问题，通过分析轮询处的代码，轮询中独立的控制信息没有问题，而共用

的数据总线出现错误，基本上可以确定是与高阻态相关。在网上查找资料，获知在FPGA 内部是不存在高阻态及不定态的。我们针对该问题对设计进行修改，由于此时的轮询为内部的模块不存在在片间时的管脚个数的限制，因此我们将共享总线变为两套独立的总线解决此问题。

（2） 数据搬移过程中对数据帧的修改失败：

在测试的时候发现当发送的数据帧从多个端口转发出去时，若从不同的端口转发时同时存在 TAG 帧及非 TAG 帧。对数据帧的修改将会失败。例如端口 1,2 为 access 口且都属于值为 1 的 VLAN ID ，端口 3 为 trunk 口，支持值为 1 的 VLAN ID ，且 PVID 为 2。此时若从端口 1 发送一个为非 TAG 帧的广播帧，则该数据帧将会被转发到端口 2 和 3，且从端口 2 转发时为非 TAG 帧从端口 3 转发时为 TAG 帧。我们测试的结果却是端口 2 和 3 收到的都为非 TAG 帧，然后查看端口 2 和 3 收到的数据帧的个数，发现端口 2 和 3 收到的数据帧与端口 1 发送的数据帧个数相同，表明入队申请正确。

对问题进行分析后，做了以下一些测试：首先让单播表进行学习使端口 1 发送的数据帧只转发到端口 3，此时端口 3 收到的数据帧为 TAG 帧，接着使端口 1 发送的数据帧只转发到端口 2，此时端口 2 收到的数据帧为 UNTAG 帧。然后将 4 个端口都分配到一个 VLAN ID 下，然后端口 1 发送广播帧，四个端口都能正常收到数据帧。以上的测试都正常，表明 vlan 转发表的结果没有问题，表明对数据帧修改的指令的产生没有问题，问题应该是出在指令的执行上。复位专用交换机，回到最开始时出错时的情况，我们通过 chipscope 对 fp_bus_ctrl 模块中的信号及状态机的跳转进行观察，发现在向端口 3 发送数据时根本没有收到对数据帧的修改指令，因此会默认对数据帧不进行操作的搬移。然后查看 fp_bus_ctrl 模块状态机的跳转，发现 fp_action_exe 所给出的指令时 fp_bus_ctrl 模块是不能接受的，分析发现是由于当需要对数据帧重复两次搬移时，在第二次搬移时，fp_action_exe 搬移修改指令的给出是根据第一次数据帧搬移结束的信号给出的，然而 fp_bus_ctrl 模块中能够第二次接受指令是在总线控制模块给出搬移使能信号拉低后来接受的。而问题就出在总线控制模块总线搬移使能信号是在数据搬移完成几个周期后才拉低的。因此同一数据帧的重复搬移时，第二次指令的产生是提前于 fp_bus_ctrl 模块对指令的接收。而当一个数据帧搬移一次时，也只产生一次修改指令，下一次的修改指令的产生是针对下一数据帧的，因此不会出现接收不到指令的情况。解决的方法是 fp_action_exe 模块中第二次指令是在总线控制模块给出的搬移使能拉低后产生的，重新测试后解决了出现的问题。

结束语

本论文是结合实验室承担的项的“航天专用分组交换机”来完成的。以航天专用的交换技术为背景，根据设计的需求研究了适用于航空航天环境下的专用交换机，本人重点研究了分组处理及转发表的实现。本人完成的主要工作如下：第一章是对专用交换机工作的环境进行介绍，并对分组交换中的以太网交换及 AFDX 相关的技术进行介绍，第二章是在对专用交换机的设计需求的分析基础上给出了整体的设计方案，并给出了对数据帧的处理流程。第三章及第四章主要是对分组处理及转发表的实现、接口以及功能的仿真及验证等进行了介绍。第五章是搭建测试平台，对专用交换机进行板级测试，按照设计的需求对专用交换机进行功能及性能测试。对测试中所遇到的问题，通过采用 chipscope 对信号进行抓取并观察，然后根据信号进行分析，来解决测试中遇到的问题。

限于时间及个人能力，本设计中还有需要改进及完善的地方。

（1） 本项目所需要的交换容量较低，我们采用共享缓存的交换方式。为了提高交换容量及扩充性，可以考虑采用带缓存的 Crossbar 结构来实现交换。

（2） 在单播转发表中，本设计中学习和查找都是对 hash 映射后的 4 个地址依次进行学习或查找，可以将 hash 映射的 4 个地址的表项同时读出来，然后进行学习或查找。这样就可以减少地址表项的学习和查找所需的时间。

（3） 本设计中的专用交换机，为了适用空间站的环境，需要交换机的具有较高的抗辐射及可靠性，这些方面需要进一步的学习。

参考文献

[1] 周乐文. 高可靠千兆以太网交换机研究[D]. 长沙: 国防科技大学, 2011.

[2] 马寅.航天用 SRAM 型 FPGA 抗单粒子翻转设计[J]. 航天器环境工程,2011,28(6):551~555.

[3] 刘增基,鲍民权,邱智亮. 交换原理与技术[M]. 西安: 西安电子科技大学出版社,2007.9.

[4] 吴厚航.《深入浅出玩转 FPGA》[M]. 北京: 北京航空航天大学,2013.

[5] CES White Paper on AFDX (Avionics Full Duplex Switched Ethernet), V1.0, November 25, 2003.

[6] AFDX/ARINC 664 Tutorial (1500-049),Condor Engineering, Inc, May 2005, V3.0.

[7] ARINC Draft 3 of Project Paper 664, Aircraft Data Network, Part 7, Avionics Full Duplex Switched Ethernet (AFDX) Network. 13 September 2004.

[8] 沈磊. 航空全双工以太网（AFDX）交换机关键技术研究与实现[D]. 西安: 西安电子科技大学, 2011.

[9] 陈昕,杨杰,周拥军,航空全双工交换以太网冗余管理机制研究[J]. 计算机工程与应用,2009,45（2）.

[10] 邱蔚. 专用交换单元流分类和分组处理的设计与实现[D]. 西安: 西安电子科技大学, 2013:16.

[11] 贺亮. HINOC2.0 流分类的实现及 MAC 协处理器功能验证技术[D]. 西安: 西安电子科技大学, 2012.

[12] 张涛.sHINOC 网络 MAC 协处理器地址转发表功能的设计与实现[D]. 西安: 西安电子科技大学,2011,1.

[13] 夏宇闻. Verilog 数字系统设计教程[M]. 北京: 北京航空航天大学出版社, 2013.

[14] 周昌权. 局域网端口 VLAN 技术的实现[J]. 电脑编程技巧与维护, 2010, 02 期: 96-97.

[15] 姜丽云. AFDX 网络关键技术研究[D]. 西安: 西安电子科技大学, 2013.

[16] 李德水. 基于 IEEE802.1Q 帧标记的 VLAN 实现原理[J]. 信息技术, 2006, 10 期(10):68-70.

致谢

衷心感谢我的指导老师邱智亮教授在研究生期间对我的谆谆教诲和悉心指导。邱老师严谨的治学态度，渊博的专业知识，孜孜不倦的工作作风，严以律己、诲人不倦的高尚品格，带给我很大的影响。读研期间，在邱老师的耐心栽培下，我的科研能力得到了很大的提高，学习方法也有了极大的改进。工作中，邱老师为我们提供良好的科研环境，生活中，老师关心每一位学生，教会我们劳逸结合。正是由于邱老师细心的教导，我的课题研究及论文才得以顺利进行，再次向邱老师致以由衷的感谢和深深的敬意。

感谢我的导师鲍民权副教授，鲍老师渊博的知识、敏锐的思维和严谨的作风让我获益匪浅，无论是在生活中还是学习中，鲍老师都给予我很多的帮助和关心。

感谢 ISN 实验室的老前辈刘增基教授，已经退休的刘老师对学术的态度是整个实验室的精神榜样。

感谢项目组的潘伟涛老师、姚明昨老师、刘焕峰老师在日常学习工作中对我的帮助和指导。

另外，感谢跟我一起完成项目的姜晨同学。项目设计期间，遇到棘手的问题，我们一起探讨，互相启发，逐渐解决了一连串的技术难题；生活中，作为生活在同一屋檐下的舍友，我们互相扶持，互相鼓励，相似的兴趣爱好增强了我们之间的默契程度，使我们在工作中更加高效。所以，在此要感谢姜晨同学对我的包容和理解。

实验室杜长刚师兄、高明君师兄、李泽师兄，在项目初期耐心解答我的各种问题，并对项目设计提出了多项宝贵的改进意见，这里也要对他们表示感谢。感谢 14 级的李兴旺、刘春锐、于东洋、许晶等师弟师妹，在实验室的项目中共同学习，共同探讨。

和我一起读研的实验室同学，戴上杰、段聪、高铮、梅益波、车香蕾、陈傲、张宇航、赵驰、陈登、屈显品，陪伴我度过了一段快乐且充实的研究生时光，这段记忆将成为我一辈子的财富，感谢你们的帮助和陪伴。

最后我要感谢我的父母，在我的成长的每一个瞬间，你们都一直陪伴在我身边，尽一切努力来帮助我，你们的爱让我能够一直前进，我将用尽自己的一生来报答你们。

西安电子科技大学

XIDIAN UNIVERSITY

地址：西安市太白南路2号

邮编：710071

网址：www.xidian.edu.cn