

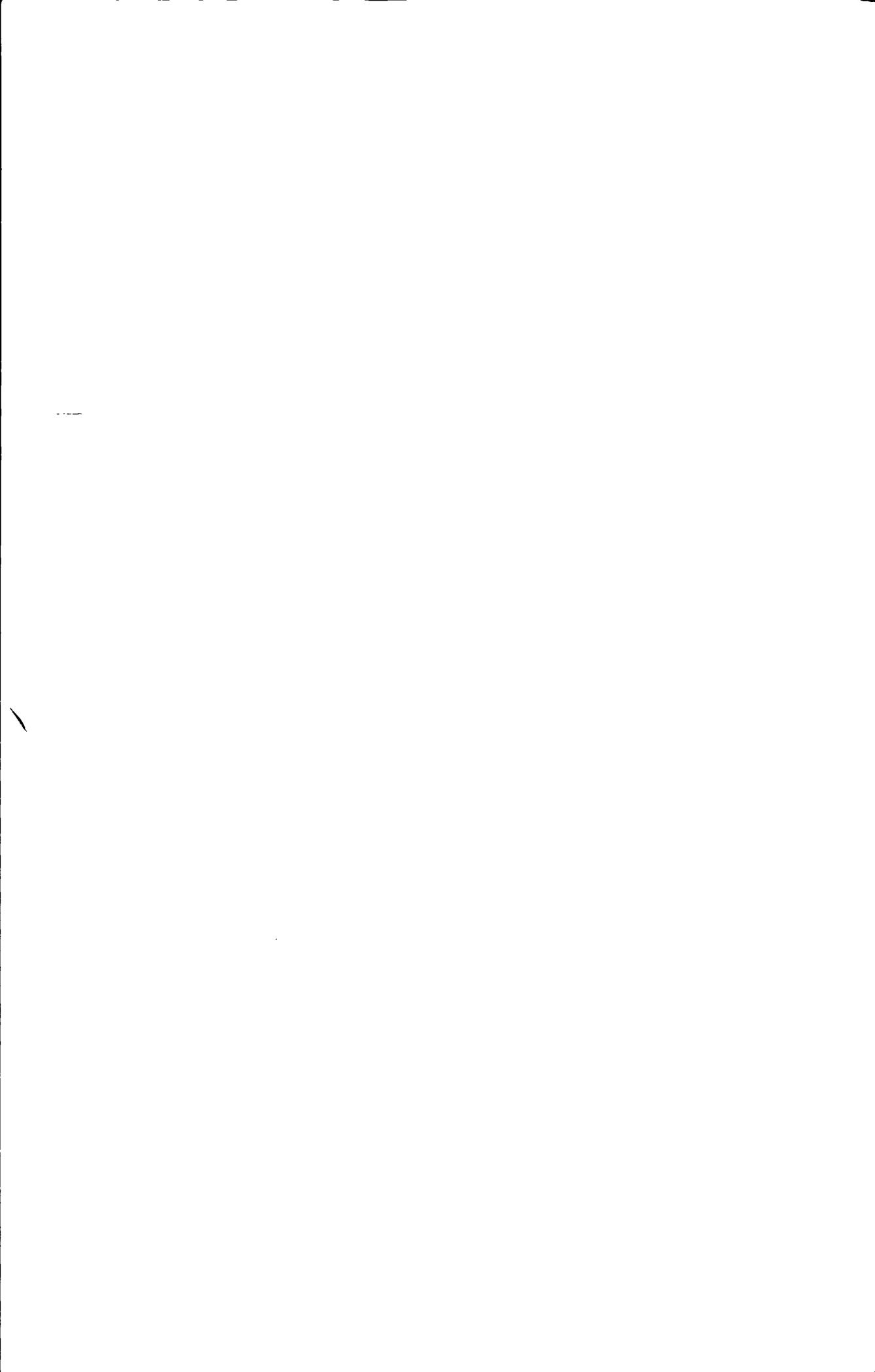
1888

## 摘要

宽带卫星通信是当今通信领域一个最重要的发展方向。为了满足用户所需要的服务质量(QoS),具有星上处理和交换功能的卫星 ATM 系统成为未来卫星通信发展的重点。本文基于星上 10Gbps ATM 交换机的设计,重点研究信元预处理部分的设计与实现。文中主要完成的工作包括:

1. 在简述 ATM 交换基本原理基础上,按照技术指标给出了星上大容量 ATM 交换机的总体设计方案。该方案对整个交换机进行结构的划分,并且阐述了各个部分的具体功能。
2. 对信元预处理部分进行了详细的设计。信元预处理包括两个方向:接收方向和发送方向,接收方向主要完成了外部信元到内部信元的转换,发送方向主要完成了内部信元到外部信元的转换。文中对本部分的具体功能进行详细论述,并对具体子模块进行了信号的设计、参数的选择。
3. 在 ISE 仿真平台上实现了对信元预处理子模块基于 FPGA 的 VHDL 实现,并进行了各个模块的级联仿真,结果证明,它们可以很好地工作,实现了信元预处理的功能。

**关键词:** ATM FPGA 信元预处理 VHDL



## abstract

Broadband satellite communication is believed to be a very important direction of communications. The satellite communication systems with on-board processing and on-board switching become more and more important in the future in order to meet the quality-of-service(QoS) requirements. This thesis gives an implementation scheme of a 10Gbps ATM switch used on satellite. The detailed design method and realization using VHDL of the cell preprocess part are presented in this thesis. The main contributions of this thesis are as follows:

1. Based on the brief discussion of ATM switching principle, this thesis describes an implementation scheme of ATM switching according to its technical requirements. The classification and function of each component of the switch are described.
2. A detailed design of cell preprocess was given. This part contains two directions' process: the receiving direction and the transmitting direction. The receiving direction process changes the outer cell to be inner cell, on the contrary, the transmit direction process changes the inner cell to be outer cell. In the thesis, we give a clear description on the principle of this part. The design of the signal and the parameter of this part are also mentioned.
3. The implementation and simulation of the cell preprocess sub-module have been done based on the platform of ISE. The simulation result proves that they can work well and the function of cell preprocess is implemented.

**Key word: ATM FPGA cell preprocess VHDL**



# 目录

<b>第一章 绪论</b> .....	<b>1</b>
1.1 星上 ATM 技术概述 .....	1
1.2 卫星 ATM 的研究现状 .....	2
1.3 本文的研究任务和章节安排 .....	3
<b>第二章 ATM 技术的基本原理</b> .....	<b>5</b>
2.1 ATM 信元的基本结构 .....	5
2.2 ATM 交换机的组成与原理 .....	7
2.3 ATM 交换机的交换原理 .....	8
2.4 ATM 交换的技术特点 .....	8
<b>第三章 星上 ATM 交换机的整体设计</b> .....	<b>11</b>
3.1 星上 ATM 交换机的性能指标要求 .....	11
3.2 星上 ATM 交换机的整体设计 .....	11
3.2.1 线路接口部分 .....	12
3.2.2 信元处理与缓存 .....	12
3.2.3 交换模块 .....	13
3.2.4 控制与管理 .....	14
<b>第四章 信元预处理部分的详细设计</b> .....	<b>15</b>
4.1 信元预处理模块功能介绍 .....	15
4.1.1 接收方向信元预处理过程 .....	15
4.1.2 发送方向信元预处理过程 .....	16
4.1.3 信令与 OAM 信元处理 .....	17
4.1.4 微处理器控制接口 .....	17
4.2 内部寄存器 .....	17
4.3 信元预处理内部存储器介绍 .....	18
4.3.1 LR_FIFO (线路侧接收 FIFO) .....	18
4.3.2 SCI_RAM (交换控制标示符 RAM) .....	18
4.3.2.1 VI_VPI_TABLE .....	20
4.3.2.2 VCI_TABLE .....	20
4.3.3 SCn_FIFO (服务等级_n FIFO) .....	21
4.3.4 CHE_RAM (信头变换 RAM) .....	22
4.3.5 MC_FIFO (多播 FIFO) .....	22
4.3.6 其他缓存器的功能介绍 .....	23

<b>第五章 信元预处理各主要模块的设计实现</b> .....	<b>25</b>
5.1 FPGA 技术相关知识 .....	25
5.1.1 FPGA 的简介 .....	25
5.1.2 ISE 的简介 .....	26
5.1.3 VHDL 的简介 .....	27
5.2 带有信元计数功能的 FIFO 模块的设计与实现 .....	29
5.3 接收端信元头处理模块的设计实现 .....	30
5.4 接收端调度模块的设计实现 .....	32
5.5 72bit 到 36bit 位宽转换 FIFO 的设计 .....	34
5.6 36bit 到 72bit 位宽转换模块的设计 .....	35
5.7 发送端信头处理模块的设计实现 .....	37
5.8 多播模块的设计实现 .....	38
<b>结束语</b> .....	<b>41</b>
<b>致谢</b> .....	<b>43</b>
<b>参考文献</b> .....	<b>45</b>
<b>作者在读期间的研究成果</b> .....	<b>47</b>

## 第一章 绪论

### 1.1 星上 ATM 技术概述

卫星通信是人类通信史上的一项伟大发明,从它开始出现以来,极大的促进了社会文明的不断进步,并且将作为未来的主要通信方式。它有着通信容量大、覆盖地域广、不受地理条件限制和通信方式机动灵活等优点,这是任何现有的其它通信手段所无法比拟的。因此,自从卫星通信问世以来的 40 多年里,卫星通信技术(包括通信卫星技术、星上有效通信载荷技术和地球站设备技术等)获得了惊人的进步,可成为人类科技发展最快的领域之一。

许多在高空运行的同步卫星,其星上转发器装置没有信号处理和交换的功能,称为“透明”的卫星,它们已经无法满足用户所需的服务质量(QoS)要求。因此具有星上处理功能的卫星 ATM(Asynchronous transfer mode 异步传递方式)网络成为未来卫星通信发展的重点。在同步卫星上安装信号处理设备,由卫星本身执行 ATM 交换功能,卫星 ATM 网就可由星上 ATM 交换机与地面网络控制中心(NCC)分担控制功能。卫星通信技术和 ATM 网络技术都具有诸多特点和优势,二者相结合形成的综合性网络——卫星 ATM 网具有以下几个突出的优点:

1. 由于卫星的广域覆盖性,将其加入到 ATM 网络以后,可在广阔的地理范围内提供 ATM 业务;
2. 卫星通信和 ATM 网络综合以后按需分配带宽的能力更强,可以进一步提高信道的利用率,而且还可以动态分配接入容量,不受复杂地面网络拓扑的影响,减少了中间多次分配的环节,满足了广播、多点传送和多媒体通信的要求;
3. 在不同地点建立 ATM 网时,卫星通信能提供灵活的网络互连、网络配置和容量分配方案;
4. 利用卫星通信的广播特性和易于(多)点对多点的通信特性,可以快速建立 ATM 网的(多)点对多点的通信;
5. 可以方便的为无法预测带宽要求和业务特征的连接分配备用信道,这样既可以保证该连接的服务质量,又能够实现资源的充分利用;
6. 网络扩展简单、方便,只要按照用户的要求,在用户处安装 ATM 地面站即可;
7. 卫星可以作为地面光纤 ATM 网的安全备份,在地面网出现故障时,根据要求,通过卫星信道确定通信路由,能够快速解决网络的拥塞问题,提高了系统的传输性能。

将 ATM 方式用于卫星通信,不仅适应通信网络向 B-ISDN 发展的必然趋势,

而且具有满足未来应用需求的必要性。但是,我们必须看到,将卫星与 ATM 技术结合也会面对相当大的挑战。由于 ATM 技术目前在 ISDN 网络中的应用基础是高速率、低误码率光纤信道的使用,因而在卫星环境下采用 ATM 技术将面临一系列特殊的问题。相对于地面 ATM 网络所使用的光纤信道,卫星的带宽是有限的,更容易在数据传输中发生拥塞,再加上卫星通信不可避免的长传输时延,便需要对星上 ATM 结构进行相应的改进和创新,以降低交换时延,减少内部阻塞,达到降低丢包率、满足用户所需 QoS 的目的。

## 1.2 卫星 ATM 的研究现状

早期的宽带卫星网络是利用传统卫星(特别是 GEO)作透明中继的卫星 ATM(S-ATM, Satellite ATM)。在美国和欧洲已经开展了利用传统卫星传输 ATM 业务的实验。1996 年, NASA 的 ACTS 卫星(Advanced Communication Technology Satellite)进行了 155.54Mbit/s 的 ATM 试验。试验情况得出的结论是卫星 ATM 能提供和地面 ATM 同样的传输性能。ACTS 系统目前进入了使用。相类似比较著名的海域 RACECATALIST WISDOM 等计划。卫星透明中继 S-ATM 系统的特点是地球站昂贵,数量少,每一地球站的业务量大。由于每一地球站的业务较大,这类系统不需要复杂的多址方式就可以达到很好的信道利用率。

星上处理(OBP, On-Board Processing)概念是由欧洲空间局(European Space Agency)提出。OBP 概念包括星上基带交换、中继。OBP 技术是提高卫星系统性能的有效手段,具有如下优点:

1. 地球站复杂性减少,成本降低,发射功率降低;
2. 灵活的路由功能,可以构成灵活的网状网;
3. 减少 BOD(bandwidth on demand)时延;
4. 上行链路与下行链路可以分开设计,可以改善信道的 BER 性能<sup>[25]</sup>。

在星上处理 S-ATM 系统中,卫星具有星上交换功能,不再是透明中继。Spaceway 是 Hughes 通信公司开发的提供宽带接入的卫星通信系统,它利用 GEO 星座,在使用 66cm 的固定超小孔径终端时,提供宽带的通信(16kbit/s-6Mbit/s)。在卫星有效载荷中包括数字信号再生和 ATM 交换。与之相类似的星上 ATM 交换系统有 Vice span(AT&T), Astrolink (Lockheed Martin), Cyberstar(LORAL)等。这类系统一般是提供宽带接入,属于接入网范畴。与透明卫星 ATM 相比,这类方案的特点是有数量众多的用户终端站和少量的关口站。用户终端站的业务量波动比较大,因此多址接入技术对系统的性能影响很大。另外用户终端站的价格及大小影响这一方案的推广使用。这一方案由于利用星上交换可获得灵活的全网状的点对点 and 点对多点结构。尽管利用星上交换具有很多优点,但还是有很多限制条件。众所

周知, 卫星是带宽有限, 质量有限的系统。此外还要经受强烈的空间辐射。因此, 必须开发能耗低、不复杂的星上处理系统, 同时星上尽可能只有少数功能。卫星通信技术和 ATM 网络技术都具有诸多特点和优势, ATM 与卫星相互取长补短的组合, 既能发挥卫星投资少、见效快、通信容量大等特点, 又能充分发挥 ATM 网的灵活性和适应性。卫星通信与 ATM 网的综合为信息时代的人们提供了一个随时随地以合理的费用获取信息的良好宽带网络平台。

### 1.3 本文的研究任务和章节安排

本论文是基于横向委托项目——星上 10Gbps 大容量交换原型样机的研制。本论文主要完成了星上使用的 ATM 交换机信元预处理部分基于 FPGA 的设计与实现。所谓信元预处理分为信元接收方向处理、信元发送方向处理两个部分。信元接收方向处理是指标准的 ATM 信元经过 UTOPIA 接口进入接收端预处理模块产生内部信元并发往交换模块的流程。信元发送方向处理是指由交换模块送出的内部信元进入发送端预处理模块产生标准的 ATM 信元并经 UTOPIA 接口送出的流程。

本文详细的介绍了信元预处理部分的系统设计、模块划分及其 VHDL 实现, 并且进行了各个模块的联合调试。

本论文的章节安排如下:

第一章介绍了星上 ATM 技术产生的背景、研究现状以及本文的研究任务和章节安排。

第二章详细的介绍了 ATM 技术的基本原理, 包括 ATM 信元的结构、ATM 交换机的工作原理和技术特点。

第三章详细的介绍了基于 FPGA 的 ATM 交换机的整体结构设计, 包括了交换机的整体划分、技术指标以及方案论证。

第四章介绍了交换机信元预处理部分的详细设计方案, 包括信元预处理的功能介绍、模块划分以及信元结构定义。

第五章介绍了信元预处理部分各个子模块的 FPGA 实现, 主要包括缓存器 FIFO 子模块、接收方向信头处理模块、发送方向信头处理模块、72bit 到 36bit 位宽转换模块、36bit 到 72bit 位宽转换模块、接收端信元调度模块的设计实现。



## 第二章 ATM 技术的基本原理

ATM 网络技术是一种基于交换的网络技术，ATM 交换机是实现网络互连的一种重要设备，它所完成的功能有交换、连接、路由等。ATM 技术是一种面向连接的快速分组交换技术。它是建立在统计复用基础上的，其基本思想是首先通过呼叫在源与目的地之间建立好虚连接，然后以 53 字节的信元为载体在该虚连接上传输用户信息。

### 2.1 ATM 信元的基本结构

ATM 信元格式如图 2-1 所示，它是由 5 字节信头和 48 字节信息载荷组成，信头中装有控制信息，载荷中装有被分解成数据块的用户信息或其它管理信息。

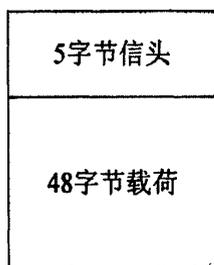


图 2.1 信元结构示意图

信元的发送顺序是从信元的第一字节开始，然后按字节顺序增加；在一个字节内，首先发送的是字节的第 8 比特，这一比特是最高有效位(MSB)<sup>[1][2][3]</sup>。信头的结构在用户与网络接口(UNI)和网间接口(NNI)上有所不同，如图 2-2 所示。

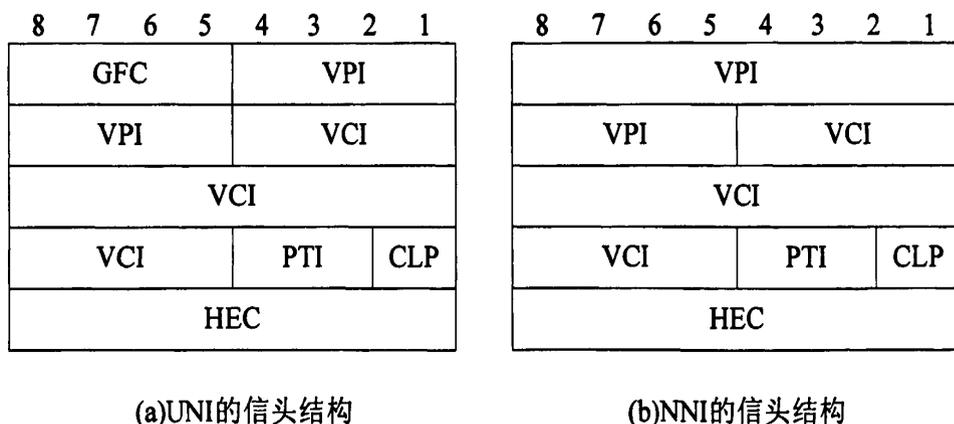


图 2.2 信头结构示意图

下面说明各标识符的功能:

GFC (一般流量控制) ——只在 UNI 中出现, 占有信元头第一个字节的前四个比特, 用于协助用户端对不同质量的信息控制业务流量。

VPI (虚通道标识) ——它对 ATM 信元提供明确的通道路径识别, 即用于识别同一物理媒质中的不同虚通道 (VP)。

VCI (虚通路标识) ——它为 ATM 信元提供明确的通路识别, 即用于识别同一虚通道中不同的虚通路 (VC)。

虚通路是用户到用户、用户到网络、网络到网络的端点之间信息转移的通信链路, 是 ATM 交换服务的最基本的传送实体。

在通道和通路前面加上“虚”字是因为它们不固定于某个用户, 而是根据用户是否要传送信息来建立和释放的, 即需要时将信道分配给用户, 用户使用完毕后就立即拆掉。

PT (载荷类型, 3 比特) ——各种不同的信元类型及其对应的载荷类型标识 (PTI) 的编码如表 2.1 所示。在系统的设计中我们用 000 表示信令信元, 其他的表示用户信元。

表 2.1 载荷类型标识编码表

PTI 编码	含义
000	用户数据信元 无阻塞 AUU=0
001	用户数据信元 无阻塞 AUU=1
010	用户数据信元 有阻塞 AUU=0
011	用户数据信元 有阻塞 AUU=1
100	段的 OAM F5 信元
101	端到端的 OAM F5 信元
110	资源管理信元
111	保留

其中 AUU 代表 ATM 层用户至 ATM 层用户标识。

CLP (信元丢失优先级, 1 比特) ——它主要用于表明信元丢失优先级。若 CLP=1, 则当网络出现拥塞时, 此信元首先被丢弃; 若 CLP=0, 则信元具有更高优先级。CLP 是用户或者业务提供者置入的, 置入的 CLP 值因业务的不同而异。例如属于一个 CBR (恒定比特率) 连接的信元常常具有高优先级; 许多 VBR (可变比特率) 业务可借助 CLP 值区分对于被丢弃敏感程度不同的信元。

HEC (信头差错控制) ——它用于控制信元头的差错, 是由物理层产生的。基于循环码原理, 在只有一个误码时它予以校正, 而在两个及以上误码时, 则对信元予以丢弃<sup>[4][5]</sup>。

## 2.2 ATM 交换机的组成与原理

ATM 交换机主要有三个功能单元，即输入、输出线路接口（I/O LIU: line interface unit）、交换单元（SU: switching unit）和控制单元（CU: control unit），如图 2.3 所示：

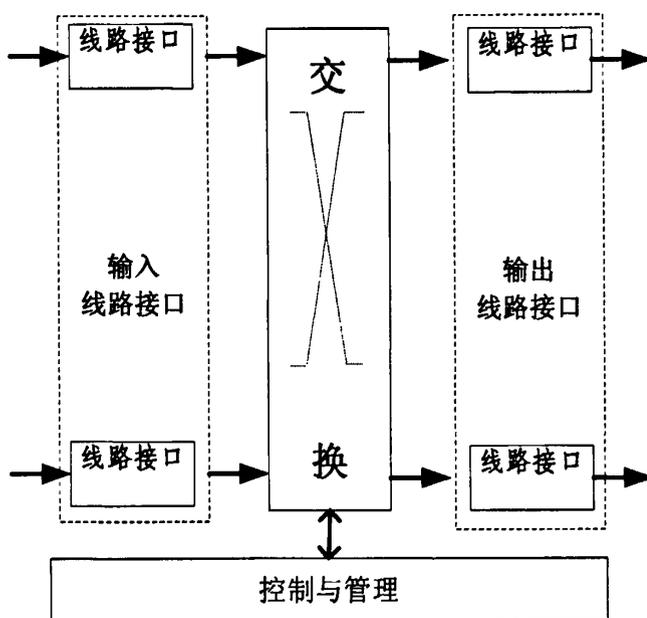


图 2.3 交换机组组成示意图

输入、输出线路接口单元主要实现网络侧物理层和 ATM 层的功能。无论是采取基于 SDH(同步数字系列)还是基于信元流的线路传输形式,经过线路接口单元送往交换单元的都将用于 ATM 层的信元,这些信元在交换单元的各个输入端口是统一定时的,并且各个信元信头中 VPI/VCI 是经过识别的。在信元送出交换单元时线路接口单元将完成适合线路传输的必要转换。

控制单元具有呼叫连接控制(CAC)、资源管理(RM)、操作维护(OAM)及优先级控制(PC)等功能。该单元根据呼叫申请的业务特性及业务质量要求,结合资源占用情况,决定是否接收该次呼叫。若该次呼叫被接收,则将有关路由信息(VPI/VCI 值、带宽、优先级等)填入路由表(Routing Table),该表主要由输入端口、输入信元 VPI/VCI 以及相应的输出端口、输出信元 VPI/VCI 组成,另外,该表中还有供交换单元使用的路由标签(Routing Tag)和优先级等。路由标签用于表示属于该次呼叫的信元在交换单元内的路由或输出端口号,优先级表示该次呼叫的时延及丢失优先级,这个优先级在这个呼叫过程中保持不变<sup>[22]</sup>。

交换单元实现信元从输入到输出的转移,把输入端口到来的信元送往指定的目的输出端口,信元的转移是根据路由表来完成的。交换单元是 ATM 交换机的核

心之一，采取何种交换结构将决定着该 ATM 交换机的性能。因此，在设计 ATM 交换机时，首先应根据设计的指标和要求选定一种合适的 ATM 交换结构。

### 2.3 ATM 交换机的交换原理

ATM 交换技术是 ATM 交换机中最重要的技术。完成 ATM 交换需要的基本功能是排队功能，交换功能和路由功能。排队功能由队列完成；交换功能由交换机构完成；路由方式有两种：一种是表路由，另一种是自路由。ATM 交换的基本原理如图 2.4 所示<sup>[21]</sup>。

图 2.4 中入口  $I_j$  上的输入信元交换到  $O_j$ ，同时信元头的输入值  $a$  转换成输出值  $b$ 。在每一输入和输出端口上，信元头的值是唯一的，但在不同的端口上可以出现相同的信元头(如在端口  $I_1$  和  $I_n$  上的  $X$ )。

从转换表中可以看到，凡是在输入端口  $I_1$  上信元头为  $X$  的所有信元都交换到输出端口  $O_1$ ，并且其信元头的值转换为  $K$ ，所有在输入端口  $I_n$  上具有信元头  $X$  的信元都交换到输出端口  $O_1$ ，信元头的值由  $X$  转换为  $N$ 。另外，从不同输入端口出发的信元可能会同时到达 ATM 交换机并竞争同一个输出端口，但它们不能在同一时刻从输出端口输出，因此在交换机内部要设置一些缓冲器存储被阻塞的信元<sup>[6][7]</sup>。

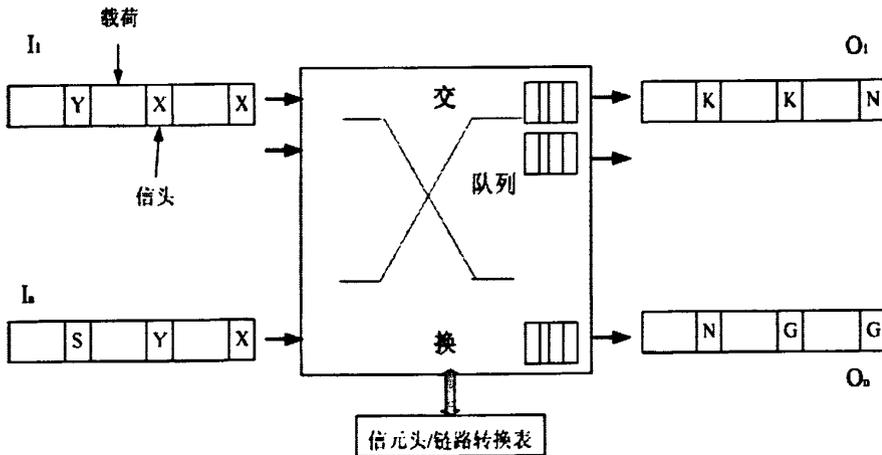


图 2.4 交换机工作原理示意图

### 2.4 ATM 交换的技术特点

ATM 可支持多种业务(语音、数据、视频)，各业务之间的性能特性差异很大，

ATM 必须满足各种不同业务对交换的要求。

1. ATM 是面向连接的技术。这种传输方式在信息传输前,必须先请求建立一个虚连接,要求网络为通信双方预留必要的资源,如果没有足够的资源可用,就会拒绝建立这个虚连接;通信结束后再通过信令来拆除连接,同时释放资源。这种面向连接的工作方式不仅大大减少了由于资源不够用(队列溢出)造成的分组丢失,而且也提高了信息传输的安全性和可靠性。
2. 多种交换速率。ATM 交换可以根据不同业务的要求,灵活地提供各种不同的交换速率,这些交换速率可以为不变比特速率(CBR)、可用比特速率(ABR)、可变比特速率(VBR)、未确定比特速率(UBR)等。
3. 多种连接方式。ATM 交换支持点到点、点到多点和广播方式,以及单向连接、对称或非对称的双向连接等多种连接方式。ATM 对各种不同业务采用不同的连接方式,例如检索型业务采用非对称的连接方式,有的业务要求呼叫建立多条不同特性的连接。
4. 多种服务质量。不同业务对传输和交换的质量要求也有所不同,有的业务对实时性要求高,要求时延小,有的业务则要求无失真地传送信息,对信元的丢失或错误敏感,因此 ATM 交换应该允许业务提出不同的服务质量要求,并在保证业务服务质量的前提下更有效地利用网络带宽。此外,ATM 交换的另一大特点是通过统计复用,各虚连接共享交换资源,从而达到较高的资源利用率。



## 第三章 星上 ATM 交换机的整体设计

### 3.1 星上 ATM 交换机的性能指标要求

文中要设计的星上 ATM 交换机主要的技术指标如下：

1. 星上 10Gbps 大容量交换原型样机端口数为 16，端口速率 622Mbps，光线路接口采用标准 SDH 光接口，可以采用专用芯片实现 SDH 光接口；
2. 信元预处理和交换部分采用 FPGA 实现；
3. ATM 交换性能指标（无差错输入信道）：
  - a) 信元差错率（CER）= $1 \times 10^{-12}$ ；
  - b) 信元丢失率（CLR）= $2 \times 10^{-10}$ ；
4. 支持每端口不少于 256 个 PVCs，并且支持点到点和点到多点、对称和不对称，广播和组播等功能；
5. 支持不少于 4 种业务优先级；
6. 原型样机具有监视、管理和控制功能，以及简单的外部管理控制接口。

### 3.2 星上 ATM 交换机的整体设计

根据交换机的性能指标要求，图 3.1 给出交换样机的硬件实现总体方案原理框图。

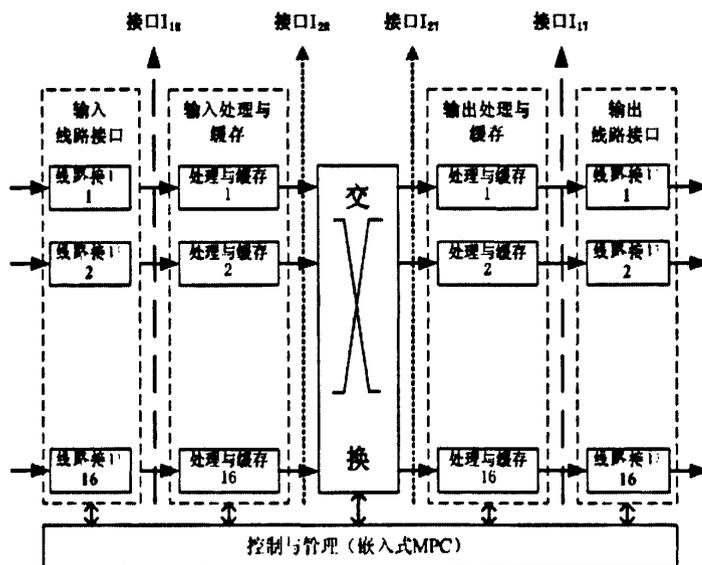


图 3.1 交换样机的硬件实现原理框图

它由如下 4 部分组成：输入/输出线路接口、输入/输出处理与缓存、交换和控制与管理。为便于表述，我们将线路接口和信元处理/缓存模块间的接口称为接口 1，信元处理/缓存模块和交换模块间的接口称为接口 2；以交换模块为参考点，将左边和右边分别称为接收和发送，接口 1 和接口 2 分为接口 I1R、I1T 和接口 I2R、I2T。

### 3.2.1 线路接口部分

线路接口分为输入和输出两部分，线路接口包括光电转换和 SDH 处理芯片两部分。

其中光电转换模块的主要功能：在接收方向上，将光纤的光信号转换为 PECL 电信号，送给 SDH 芯片；如果其接收未检测到光信号，还可向微处理器 PPC 报告。在发送方向上，将从 SDH 芯片接收的 PECL 电信号，完成电光转换，将光信号发送到光纤中。

输入方向 SDH 处理芯片的功能是：

1. 完成接收时钟恢复和数据恢复；
2. 完成 SDH 的帧定界，开销处理和载荷提取；
3. ATM 信元定界功能，删除速率调整的空信元，通过信元传输标准接口 UTOPIA 接口将信元交给输入处理与缓存模块；

输出方向 SDH 处理芯片的功能是：

1. 通过 UTOPIA 信元传输标准接口，从信元输出处理与缓存模块接收信元，并加入速率调整的空信元；
2. 将信元装入传输帧，产生 SDH 开销字段，生成 SDH 帧；
3. 经并串变换，将数据发到线路上。

在本系统的设计中选用的 SDH 芯片是 PMC 公司的 PM5358，其工作始终频率为 77.76MHz，这个时钟也选作我们整个系统的全局时钟。

### 3.2.2 信元处理与缓存

信元处理与缓存分为输入和输出两部分，其功能分别是：

#### 1. 输入部分：

通过标准的信元传送接口(UTOPIA Level 3)，将接收到的信元存入缓存队列 FIFO 中，当输入缓存区有信元时，控制电路读出其信元标头的虚通道标识符 VPI，经硬件查表取得该信元的连接控制块 CCB，判断连接是虚通道连接 VPC，还是虚通路连接 VCC。

当为 VPC 连接时, 取出信元的交换控制标识符 SCI (Switching Control Identifier); 当为 VCC 时, 通过该 VPI 对应的 VCC 变换表起始地址和输入信元的 VCI, 查表取得信元的 SCI。

将处理后的 SCI、信元标头 PTI 和 CLP 字段、信元载荷, 一起组成内部交换信元, 放入信元输出缓存队列, 等待交换模块处理。

根据 VP 或 VC 交换, CCB 可分为 CCB1 和 CCB2; CCB1 在 VP 交换时包括 SCI, 在 VC 交换时包含该 VP 的所有 VC 交换的 CCB2 起始地址; CCB2 包括 SCI 值。

SCI 包括某个连接的业务优先级、输出端口和信元交换内部信道号, 还能实现单播、多播和广播功能。

## 2. 输出部分:

当输入缓存区有信元时, 根据信元业务的优先级, 取出一个信元。根据信元的 SCI 值中的内部信道号查表判断是 VP 连接或是 VC 连接, 取出信元标头新的 VPI/VCI 值, 将其处理后得到的 ATM 信元放入输出缓存区; 根据该信元对应连接的单播、多播、广播属性, 取出物理端口的 tag 值, 将信元输出到若干物理端口队列 (1~4)。

信元输出处理部分, 根据信元的端口值, 通过标准的信元传送接口将信元传送给线路接口。

本部分是此论文研究的重点, 以后的章节将对其进行重点的介绍。

### 3.2.3 交换模块

图 3.2 和表 3.1 分别给出 ATM 交换结构和缓冲分类, 一般 ATM 交换机可分为时分和空分两大类。输入缓冲交换机的主要缺点是存在队头阻塞效应 (blocking of head line), 吞吐率最大为 58%; 输出缓冲要求总线速率提高 N 倍, 缓冲器的利用率较低, 交叉点缓冲容量需求最大<sup>[8][9][10]</sup>。

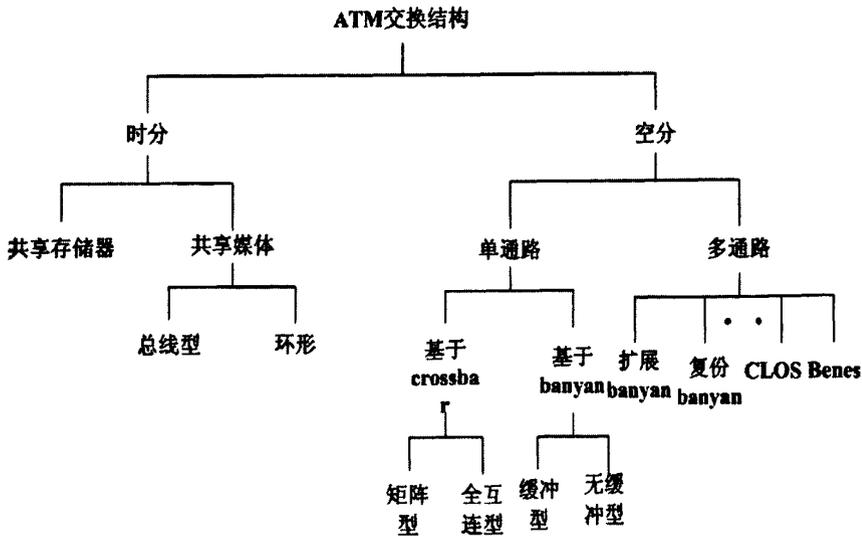


图 3.2 ATM 交换机结构分类

表 3.1 ATM 交换机缓冲策略

	共享缓冲	输入缓冲	输出缓冲	交叉点缓冲
缓冲控制	多重 FIFO	FIFO 或 FIRO	FIFO	FIFO
缓冲容量	$O(N)$	$O(N)$	$O(N)$	$O(N^2)$
	小-----大			
特性	缓冲器利用率高 复杂的控制	吞吐率低 速度要求低	吞吐率较高 控制较简单	控制简单 缓冲器容量大

共享存储器的资源利用率高，需要较复杂的控制。在 FPGA 功能日益增强的条件下，复杂度已不再是主要问题，现有交换芯片一般采用结合输入输出的排队策略 CIOQ (Combined Input and Output Queuing) 以提高性能，故我们选择方案是实现 CIOQ 型共享缓存交换机<sup>[11]</sup>。

### 3.2.4 控制与管理

控制与管理模块完成如下任务：

1. 输入线路接口、输入处理与缓存、交换模块、输出缓存和输出线路接口的初始化、工作方式设置。
2. 根据管理终端的连接设置，修改输入信元处理中的交换变换表，并将测试信元和 OAM 信元添加到合适的输出信元缓存端口。

在系统的设计中我们采用嵌入式微处理器及其应用平台作为控制与管理软件的运行平台，选用华恒公司的 Motorola PowerPC 8280 核心板 (HHPPC 8280)。

## 第四章 信元预处理部分的详细设计

### 4.1 信元预处理模块功能介绍

本系统的设计中信元预处理和交换部分是基于 FPGA 实现的，设计的方案是交换部分使用一片 FPGA 来实现。预处理部分使用四片 FPGA 实现，每一片 FPGA 完成四路接收和发送方向的处理，完成共计十六路的处理任务。其硬件结构如图 4.1 所示。

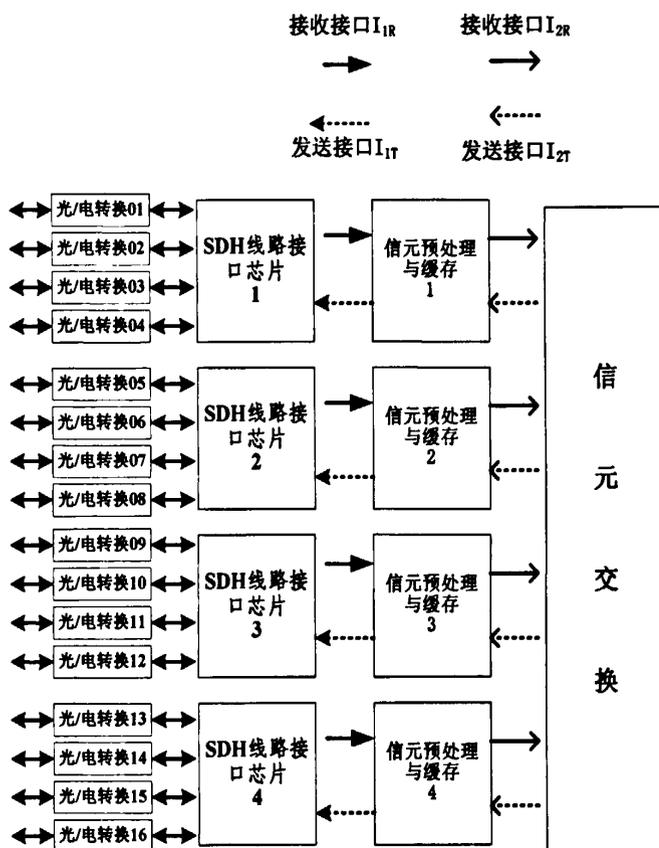


图 4.1 交换样机的硬件实现原理框图

图 4.2 给出信元预处理模块原理框图(图中表示了一路接收和发送方向的处理流程)，它分为信元接收方向处理、信元发送方向处理、信令和 OAM 信元处理和微处理器控制接口 4 部分。

#### 4.1.1 接收方向信元预处理过程

1. 线路侧接收接口通过 UTOPIA 接口(level 3)接收信元，并通过 36bit 到 72bit 的

位宽转换 FIFO 完成位宽的转换，并将信元缓存在 LR\_FIFO 中。

2. 信元接收处理单元从 LR\_FIFO 中取出信元的头部，在 SCI-RAM 中查找对应的优先级、交换标签和内部信道号，然后将该信息连同输入 FIFO 中的信元载荷一起，发送给 SCn\_FIFO。SCn\_FIFO 共五路，其中四个分别存储去往四个不同端口的单播信元，另外一个存储去往多个端口的多播信元。信元去往端口是利用交换标签的后四位确定的：当后四位为 0001 时对应第一个端口，当为 0010 时对应第二个端口，当为 0100 时对应第三个端口，当为 1000 时对应第四个端口，其他的情况对应多播情况，例如 0011 时表示去往第一和第二个端口的多播信元，将此信元存入 SC5\_FIFO 中。信元接收处理单元同时还将所有信令信元和 OAM 信元取出，传送给信令接收接口。信令接收接口与微处理器交互，将信令信元和 OAM 信元送往微处理器。
3. 接收调度单元按照一定的调度算法，从不同的优先级信元队列中取出信元，存放于 SR\_FIFO 中。在本次调度算法的设计中我们采用五个 SCn\_FIFO 中的信元轮流读取的调度策略，每次读取一个完整的信元。
4. 交换侧接收接口将信元存入 72bit 到 36bit 位宽转换 FIFO 中，并通过 UTOPIA 接口(level 1)发送给交换模块。

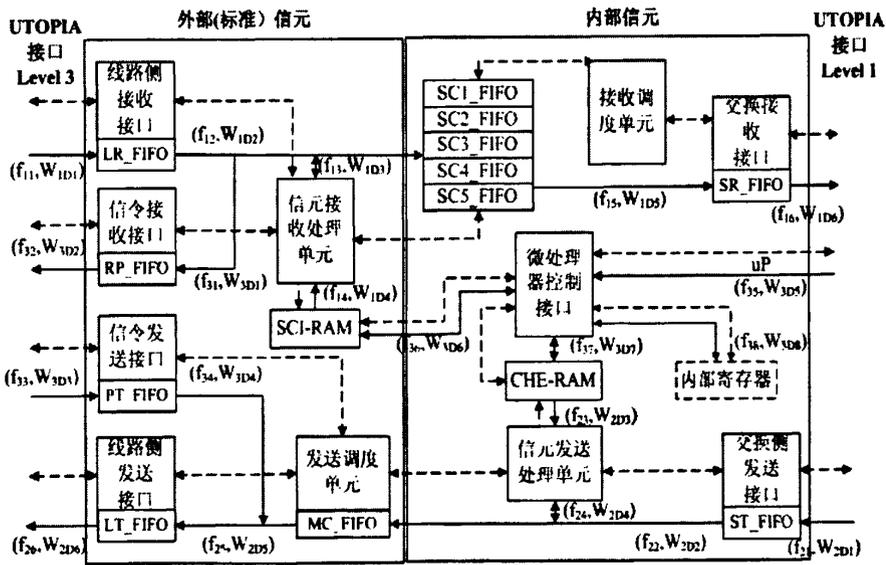


图 4.2 信元预处理模块原理框图

### 4.1.2 发送方向信元预处理过程

1. 线路侧发送部分通过 UTOPIA 接口(level 1)接收来自于交换模块的信元，并经

过 36bit 到 72bit 转换, 将信元缓存在 ST\_FIFO 中。

2. 信元发送处理单元首先将信头从 ST\_FIFO 中取出, 然后根据信头中的内部信道号, 在 CHE\_RAM 中查表, 得到该信元的出端口号和新的 VPI/VCI 的值, 生成新的信头。并将新的信头和 ST\_FIFO 中的信元载荷一起, 在 MC\_FIFO 中缓存, 进入排队等待发送。
3. 发送调度单元从 MC\_FIFO 中取出信元, 存放于 LT\_FIFO 中, 准备发送。发送调度单元同时还 将微处理器传送给信令发送接口的信令信元和 OAM 信元也送到 LT\_FIFO 中。
4. 线路侧发送接口将信元从 LT\_FIFO 中取出, 交换侧接收接口完成 72bit 到 36bit 位宽转换, 并通过 UTOPIA 接口(level 3)发送给物理线路接口模块。对于多播信元, 根据其输出的物理端口号, 由 UTOPIA 接口完成信元的多播发送。

#### 4.1.3 信令与 OAM 信元处理

如图 4.2 所示, 在信元接收处理单元处, 根据 PT 中所指明的信元类型, 将信令信元和 OAM 取出, 送至信令接收接口, 传递给微处理器进行操作。

需要发送出去的信令信元, 通过信令发送接口在 PT\_FIFO 中缓存, 并由发送调度单元取出, 送至线路侧发送接口, 通过 UTOPIA 接口发送出去。

#### 4.1.4 微处理器控制接口

通过微处理器控制接口, 样机主控 CPU 实现以下功能:

1. 各个模块进行初始化控制。
2. 各个模块的状态和异常检测的报告。
3. 样机主控 CPU 通过该接口直接对 SCI\_RAM 和 CHE\_RAM 进行读写, 写入相关的控制信息, 并核查写入信息的正确性。

## 4.2 内部寄存器

为完成内部数据存储、标志位存储等功能, 拟设计约 128~256 个内部寄存器, 每个寄存器为 8 比特。寄存器的主要类别有: 内部控制模块、存储器模块与有关控制信号的复位, 有关控制信号的置位, 内部控制模块、存储器模块与有关控制信号的状态, 每个存储器的信元个数计数值等。这些寄存器分布于预处理芯片的不同功能模块中。

### 4.3 信元预处理内部存储器介绍

存储器结构的设计在交换机设计中至关重要，对信元结构、内容的定义直接反映到存储器的设计上，下面我们详细介绍一下信元预处理过程中各个缓存结构的定义。

#### 4.3.1 LR\_FIFO (线路侧接收 FIFO)

通过 UTOPIA 接口(Level 3)接收到的信元通过位宽转换以后变成 72bit 宽度在 LR\_FIFO (Line Receive FIFO) 中缓存。由于 UTOPIA 接口一次传送的是 32bits 的数据，为了提高数据的处理速度将 UTOPIA 接口两次传送的数据组合在一起存储，组合的原则是将每次传送的 32bits 数据之前加上 4bits 的冗余信息，当是信元的头部时此 4 比特的两个比特用 10 表示信元的起始(SOC)，剩余的两个比特表示端口号(VI)；当是载荷部分时此 4 比特全部为 0。表 4.1 给出 LR\_FIFO 存储结构，它是由标准信元去掉 HEC 字节并添加起始信息与输入端口信息构成的。

表 4.1 LR\_FIFO 存储结构

偏移量	信元						
	71:68	67:60	59:52	51:44	43:36	35:32	31:0
0	10 VI[1:0]	VPI[11:4]	VPI[3:0] VCI[15:12]	VCI[11:4]	VCI[3:0], PTI[2:0], CLP[0]	0000	payload
1	0000	payload	payload	payload	payload	payload	payload
2	0000	payload	payload	payload	payload	payload	payload
3	0000	payload	payload	payload	payload	payload	payload
4	0000	payload	payload	payload	payload	payload	payload
5	0000	payload	payload	payload	payload	payload	payload
6	0000	payload	payload	payload	payload	payload	全 0

#### 4.3.2 SCI\_RAM (交换控制标识符 RAM)

本样机采用的交换控制标识符 SCI 的格式如图 4.3 所示，其主要功能是指出该信元的如下特征：

1. CHN[11:0]表示信元内部信道号，用于在信元预处理的输出部分，完成输出信元的 VPI 或 VPI/VCI 插入，以及信元在该信元预处理芯片的 4 个 SDH 物理端

口上的多播、广播。

2. P[3:0]表示信元所属连接的业务优先级,可表示 16 个优先级;为提高队列的利用率和管理效率,样机采用 4 比特位表示对应的 4 个优先级, P[3:0]代表的优先级从高到低;
3. TAG[7:0]的每个比特标识 8 个信元预处理信元芯片(目前仅用 3:0, 共 4 位), TAG[7:0]的某个位置 1 表示信元需在该信元预处理芯片对应的交换端口输出,不管某个信元是 1 个端口上单播,还是 2~4 端口上多播和广播;

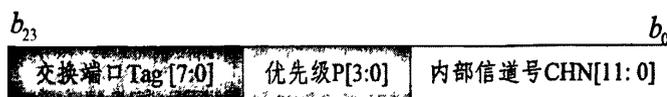


图 4.3 交换控制标识符 SCI 格式

根据接收到信元的 VI、VPI 和 VCI 的值,能够在表 SCI\_RAM (Switch Control identifier RAM) 中查到对应的连接的优先级、交换标签和内部信道号等信息。

查表的内容分为两步:首先根据 VI 和 VPI 的值,在 VI\_VPI\_TABLE 中进行查表,如果是 VP 交换,直接查出该信元的优先级、交换标签和内部信道号;如果是 VC 交换,再在表 VCI\_TABLE 中进行一次查表,得到该信元的优先级、交换标签和内部信道号。

为满足样机每端口 256 个连接的要求,并尽可能减小变换表的 RAM 占用,样机的每个物理端口当前支持 4 条 VP 连接,其中的 3 条是 VP 交换,剩下的 1 条 VP 中支持 256 条 VC 连接。

图 4.4 给出了 SCI\_RAM 存储空间分配情况。

VI_VPI_TABLE	00h	VI=0, VP=0
		VI=0, VP=1
		⋮
	0Fh	VI=3, VP=3
VCI_TABLE	10h	VI=0 VC=0~255
		VI=1 VC=0~255
		VI=2 VC=0~255
	40Fh	VI=3 VC=0~255

图 4.4 SCI\_RAM 存储空间分配图

## 4.3.2.1 VI\_VPI\_TABLE

根据接收到信元的 VI 和 VPI 的值，能够在表 VI\_VPI\_TABLE 中查到该信元对应的 VPC 的优先级、交换标签和内部信道号。该表的细节如下：

基地址：00<sub>h</sub>

条目数：16

每条目存储空间大小：4 Bytes

寻址方式：基地址 + VI\*2<sup>2</sup> + VPI mod 2<sup>2</sup>

总大小：64 Bytes

VI\_VPI\_TABLE 有两种不同的表结构，对于 VP 交换 (VPC\_ENTRY=1)，其结构如表 4.2 所示；对于 VC 交换 (VPC\_ENTRY=0)，其结构如表 4.3 所示。

表 4.2 VI\_VPI\_TABLE 存储结构 (VPC\_ENTRY = 1)

范围 (bits)	内容	描述
31	VPC_ENTRY	1: 代表 VP 交换 0: 代表 VC 交换
30	STATE	1: 代表当前连接有效 0: 代表当前连接无效
29:24	Not used	未使用
23:16	TAG[7:0]	交换标签
15:12	priority[3:0]	优先级
11:0	CHN[11:0]	信道号

表 4.3 VI\_VPI\_TABLE 存储结构 (VPC\_ENTRY = 0)

范围 (bits)	内容	描述
31	VPC_ENTRY	1: 代表 VP 交换 0: 代表 VC 交换
30	STATE	1: 代表当前连接有效 0: 代表当前连接无效
29:12	Not used	未使用
11:0	OFFSET[11:0]	该 VP 所属的 VC 在表 VCI_TABLE 中的 起始地址

## 4.3.2.2 VCI\_TABLE

如果是 VC 交换，根据 VI 和 VCI 的值，可以在 VCI\_TABLE 中查到当前 VC 连接对应的优先级、交换标签和内部信道号。表 4.4 给出 VCI\_TABLE 的表结构。

该表的构成如下：

条目数：1K

每条目存储空间大小：4 Bytes

寻址方式：OFFSET + VCI mod  $2^8$

总大小：4K Bytes

表 4.4 VCI\_TABLE 存储结构 (VPC\_ENTRY = 0)

范围 (bits)	内容	描述
31	Not used	未使用
30	STATE	1: 代表当前连接有效 0: 代表当前连接无效
29:24	Not used	未使用
23:16	TAG[7:0]	交换标签
15:12	priority [3:0]	优先级
11:0	CHN[11:0]	信道号

#### 4.3.3 SCn\_FIFO (服务等级\_n FIFO)

SCn\_FIFO (Service Class n FIFO) 共五个，其中四个分别存储去往四个不同端口的单播，另外一个存储去往两个以上端口的多播信元。表 4.5 给出了 SCn\_FIFO 存储结构，它是内部信元结构，头部的 4 字节是内部交换标头。

表 4.5 SCn\_FIFO 存储结构

偏移量	信元						
	71:68	67:60	59:52	51:44	43:36	35:32	31:0
0	10 00	0000 TAG[7:4]	TAG[3:0] Priority[3:0]	CHE[11:4]	CHE[3:0], PTI[2:0], CLP[0]	0000	payload
1	0000	payload	payload	payload	payload	payload	payload
2	0000	payload	payload	payload	payload	payload	payload
3	0000	payload	payload	payload	payload	payload	payload
4	0000	payload	payload	payload	payload	payload	payload
5	0000	payload	payload	payload	payload	payload	payload
6	0000	payload	payload	payload	payload	payload	全 0

信元接收处理单元依据在 SCI\_RAM 中获得的 TAG 值的后四位，判断其是单播还是多播信元，将其存入不同的缓存器 SCn\_FIFO 中，等待被接收调度模块取出，

经过位宽的转换以后通过交换侧接收接口，发送到交换模块。

#### 4.3.4 CHE\_RAM (信头变换 RAM)

根据信元的内部信道号，在 CHE\_RAM (Cell Head Exchange RAM) 查表得到发送输出信元新 VPI/VCI 值、输出的物理端口号等信息。表 4.6 给出 CHE\_RAM 存储结构，其细节如下：

基地址：00H

条目数：1K

每条目存储空间大小：4 Bytes

总大小：4K Bytes

寻址方式：信道号 CHN

表 4.6 CHE\_RAM 存储结构

范围 (bits)	内容	描述
31:28	OUT_NUM[3:0]	输出的物理端口号
27:0	NEW_VPI_VCI[27:0]	输出信元新的 VPI/VCI 值

#### 4.3.5 MC\_FIFO (多播 FIFO)

MC\_FIFO (Multicast FIFO) 是完成信元多播复制与发送的主要模块。送入 MC\_FIFO 中的信元是从发送端信头变换模块处理以后送出的信元，其结构如表 4.7 所示。

表 4.7 MC\_FIFO 输入的信元结构

偏移量	信元						
	71:68	67:60	59:52	51:44	43:36	35:32	31:0
0	10 00	VPI[11:4]	VPI[3:0] VCI[15:12]	VCI[11:4]	VCI[3:0], PTI[2:0], CLP[0]	OUT_NUM[3:0]	payload
1	0000	payload	payload	payload	payload	payload	payload
2	0000	payload	payload	payload	payload	payload	payload
3	0000	payload	payload	payload	payload	payload	payload
4	0000	payload	payload	payload	payload	payload	payload
5	0000	payload	payload	payload	payload	payload	payload
6	0000	payload	payload	payload	payload	payload	全 0

信元被加入新的信头后, 在 MC\_FIFO 中缓存, 并且完成多播复制, 等待发送调度单元读出, 送往 UTOPIA 接口发送。MC\_FIFO 所需完成的功能是根据输入信元头的出信号 OUT\_NUM[3:0]来判断信元所需多播到的端口号, 然后确定多播的次数与多播信元格式的转变, 得到新的信元结构(如表 4.1 所示), 等到调度模块调度经过 UTOPIA Level3 接口发送到 SDH 芯片。

#### 4.3.6 其他缓存器的功能介绍

##### ◆ SR\_FIFO (交换模块接收端缓存 Switch Receive FIFO)

SR\_FIFO 是接收端与交换模块之间的缓存, 接收调度模块监督 5 个信元缓存器的长度, 依据一定的调度算法, 从 5 个信元缓存器读出一个信元, 并写入该信元存储区。其信元结构见表 4.5。

##### ◆ ST\_FIFO (交换模块发送端缓存 Switch Transmit FIFO)

ST\_FIFO 是发送端与交换模块之间的缓存, 交换模块向其写入信元, 发送调度模块取出一个信元, 完成进一步处理。

##### ◆ LT\_FIFO (线路接口发送端缓存 Line Transmit FIFO)

LT\_FIFO 是发送端 UTOPIA 接口的信元缓存, 信元最终通过线路接口传送到 SDH 物理层芯片处。该接口采用标准的 UTOPIA Level3 规范。

##### ◆ RP\_FIFO (接收端微处理器信令信元缓存 Receive Processor FIFO)

接收处理单元从接收信元流中过滤信令信元和 OAM 信元, RP\_FIFO 存放该信元, 信令接收接口将其送往微处理器, 以进行进一步处理。该存储器的结构与 LR\_FIFO 相同, 见表 4.1。

##### ◆ PT\_FIFO (微处理器发送端信令信元缓存 Processor Transmit FIFO)

PT\_FIFO 是微处理器送到发送端的信令信元的缓存。该存储器的结构与 LR\_FIFO 相同, 见表 4.1。



## 第五章 信元预处理各主要模块的设计实现

### 5.1 FPGA 技术相关知识

#### 5.1.1 FPGA 的简介

FPGA 是英文 Field Programmable Gate Array 的缩写,即现场可编程门阵列,它是在 PAL、GAL、EPLD 等可编程器件的基础上进一步发展的产物。它是作为专用集成电路(ASIC)领域中的一种半定制电路而出现的,既解决了定制电路生产周期长、少量生产代价高的不足,又克服了原有可编程器件门电路数有限的缺点。

FPGA 采用了逻辑单元阵列 LCA (Logic Cell Array) 这样一个新概念,内部包括可配置逻辑模块 CLB (Configurable Logic Block)、输入输出模块 IOB (Input Output Block) 和内部连线 (Interconnect) 三个部分。

FPGA 的基本特点主要有:

1. 采用 FPGA 设计 ASIC 电路,用户不需要定做生产,就能得到合用的芯片。
2. FPGA 可做其它全定制或半定制 ASIC 电路的测试样片。
3. FPGA 内部有丰富的触发器和 I/O 引脚。
4. FPGA 是 ASIC 电路中设计周期最短、开发费用最低、风险最小的器件之一。
5. FPGA 采用高速 CHMOS 工艺,功耗低,可以与 CMOS、TTL 电平兼容。可以说, FPGA 芯片是小批量系统提高系统集成度、可靠性的最佳选择之一。目前 FPGA 的品种很多,有 XILINX 的 XC 系列、TI 公司的 TPC 系列、ALTERA 公司的 FIEX 系列等。

FPGA 是由存放在片内 RAM 中的程序来设置其工作状态的,因此,工作时需要对片内的 RAM 进行编程。用户可以根据不同的配置模式,采用不同的编程方式。加电时, FPGA 芯片将 EPROM 中的数据读入片内编程 RAM 中,配置完成后, FPGA 进入工作状态。掉电后, FPGA 恢复成白片,内部逻辑关系消失,因此, FPGA 能够反复使用。FPGA 的编程无须专用的 FPGA 编程器,只须用通用的 EPROM、PROM 编程器即可。当需要修改 FPGA 功能时,只需换一片 EPROM 即可。这样,同一片 FPGA,不同的编程数据,可以产生不同的电路功能。因此, FPGA 的使用非常灵活<sup>[12]</sup>。

FPGA 有多种配置模式:并行主模式为一片 FPGA 加一片 EPROM 的方式;主从模式可以支持一片 PROM 编程多片 FPGA;串行模式可以采用串行 PROM 编程 FPGA;外设模式可以将 FPGA 作为微处理器的外设,由微处理器对其编程<sup>[13]</sup>。

在本系统的设计中，我们考虑到航天环境的特殊要求，我们选用了 Xilinx 公司的 QPro Virtex-II 系列芯片。其此系列具体的逻辑资源如表 5.1 所示。

表 5.1 QPro Virtex-II 系列芯片资源

Device	System Gates	CLB			Multiplier Blocks	Select Blocks		DCM	Max I/O Pads
		Array Row×Col	Slices	Max DRAM Kbits		18K Blocks	Max RAM Kbits		
XQR2V1000	1M	40×32	5120	160	40	40	720	8	432
XQR2V3000	3M	64×56	14336	448	96	96	1728	12	720
XQR2V3000	6M	96×88	33792	1056	144	144	2592	12	1104

估计设计所需要的资源，预处理模块选用的是 XQR2V3000 芯片，交换模块选择的是 XQR2V6000 芯片。

### 5.1.2 ISE 的简介

Xilinx 是世界最大的可编程逻辑器件供应商之一。Xilinx 的器件能达到高的性能和集成度，不仅仅是因为它采用了先进的工艺和全新的逻辑结构，还在于它提供了现代化的设计工具。Xilinx 公司的 ISE 可编程逻辑开发软件，提供了一种与结构无关的全集成化设计环境，使设计者能方便的对其公司的 PLD 系列的产品进行设计输入、快速处理和器件编程。ISE 开发系统的处理能力强且灵活性高，它的优点主要表现在以下几个方面：

1. ISE 是一个集成环境，可以完成整个 FPGA/CPLD 开发过程。ISE 集成了很多著名的 FPGA/CPLD 设计工具，根据设计流程合理应用这些工具，会使工程师的设计工作如鱼得水。
2. ISE 界面风格简洁流畅，易学易用。ISE 的界面秉承了可视化编程技术，界面根据设计流程而组织，整个设计过程只需按照界面组织结构依次点击相应的按钮或选择相应的选项即可。
3. ISE 有丰富的在线帮助信息，结合 Xilinx 的技术支持网站，一般设计过程中可能遇到的问题都能得到很好的解决。
4. 强大的设计辅助功能。ISE 秉承了 Xilinx 设计软件的强大辅助功能。在编写代

码时可以使用编写向导生成文件头和模块框架,也可以使用语言模版(Language Templates)帮助编写代码。在图形输入时可以使用 ECS 的辅助项帮助设计原理图。另外,ISE 的 Core Generator 和 LogiBLOX 工具可以方便地生成 IP Core(IP 核)与高效模块为用户所用,大大减少了设计工作者的工作量,提高了设计效率与质量。

下面我们介绍一下基于 ISE 的开发流程:

1. 电路设计与输入。电路设计与输入是根据工程师的设计方法将所设计的功能描述给 EDA 软件 ISE。常用的设计输入方法有硬件描述语言(HDL)和原理图输入法。
2. 功能仿真。电路设计完成后,要用专门的仿真工具对设计进行功能仿真,验证电路功能是否符合设计要求。功能仿真有时也被称为前仿真。
3. 综合优化。综合优化是指将 HDL 语言、原理图等设计输入翻译成由与、或、非门、RAM、触发器等基本逻辑单元组成的逻辑连接(网表),并根据目标要求(约束条件)优化所生成的逻辑连接,供 FPGA/CPLD 厂家的布局布线器进行实现。
4. 综合后仿真。综合完成后需要检查综合结果是否与原设计一致,需要做综合后仿真。在综合时,把综合生成的延时文件反标到综合仿真模型中去,可以估计门延时带来的影响。
5. 实现。综合结果的本质是一些由与、或、非门、触发器、RAM 等基本逻辑单元组成的逻辑网表,它与芯片实际的配置情况还有较大差距。此应该使用 FPGA/CPLD 厂商提供的工具软件,根据所选芯片的信号,将综合输出的逻辑网表适配到具体的 FPGA/CPLD 器件上,这个工程叫做实现(Implementation)过程。
6. 时序仿真与验证。时序仿真中应该将布局布线的时延文件反标到设计中,使仿真既包含门延时,又包含线延时信息。与前面的各种仿真相比,这种后仿真包含的延时信息最为全面、准确,能够较好的反映芯片的实际工作情况。
7. 调试与加载配置。设计开发的最后步骤就是在线调试或者将生成的配置文件写入芯片进行测试。

### 5.1.3 VHDL 的简介

VHDL 是 Very High Speed Integrated Circuit Hardware Description Language 的英文缩写,意思是超高速集成电路的硬件描述语言,是一种快速的电路设计工具,功能涵盖了电路描述、电路合成、电路仿真等三大电路设计工作。今天,VHDL 语言已经成为了事实上的通用硬件描述语言。

VHDL 诞生于 1982 年。1987 年底, VHDL 被 IEEE(The Institute of Electrical and Electronics Engineers)和美国国防部确认为标准硬件描述语言。由于半导体技术的快速进步, VHDL 所能提供高级电路描述语言的方式, 让复杂的电路可以通过 VHDL 编译器的电路合成方式, 轻易而快速地达到设计的规格。

传统硬件系统电路的设计采用自下至上的设计思想(Down-Top), 存在设计周期长、设计资料不宜阅读修改等缺点。VHDL 语言采用自上至下(Top-Down)的方法来设计系统硬件(如图 5.1)。

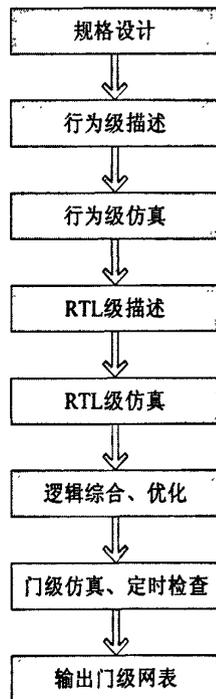


图 5.1 VHDL 语言系统设计流程

第一层次是行为描述, 即对整个系统的数学模型的描述。通过对系统行为描述的仿真来发现设计中存在的问题。第二层次是 RTL 方式描述, 即寄存器传输。经过 RTL 方式改写后, 再用仿真工具进行仿真。第三层次是逻辑综合。利用逻辑综合工具, 将 RTL 方式描述的程序转换成用基本逻辑元件表示的文件(门级网络表), 进一步画出系统的逻辑电路原理图。在门电路级再进行仿真, 并检查定时关系<sup>[14]</sup>。

在产生门级网络表后, 可以将网络表转换成相应的 ASIC 芯片制造工艺, 做出 ASIC 芯片, 或转化成 FPGA 的编程码点, 利用 FPGA 完成硬件电路的设计。这种自上至下的设计每一步都要进行仿真检查, 有利于尽早发现系统设计中存在的问题, 从而可以大大缩短系统硬件的设计周期。

系统设计中可以根据硬件电路设计的需要, 设计自用的 ASIC 芯片或可编程逻辑器件, 不必受只能使用通用元器件的限制。降低了硬件电路设计难度。主要设

计文件是用 VHDL 语言编写的源程序，易归档、保存、可继承性好。VHDL 语言可以与工艺无关编程，完成仿真检验后，可用相应的工具将设计映射成不同的工艺(如 MOS, CMOS 等)，无须修改原设计程序。

VHDL 主要用于描述数字系统的结构、行为、功能和接口。除了含有许多具有硬件特征的语句外，VHDL 的语言形式和描述风格与句法十分类似于一般的计算机高级语言。VHDL 的程序结构特点是将一项工程设计，或称设计实体（可以是一个元件、一个电路模块或一个系统）分成外部（或称可视部分，即端口）和内部（后称不可视部分），即设计实体的内部功能和算法完成部分。在对一个设计实体定义了外部界面后，一旦其内部开发完成后，其他的设计就可以直接调用这个实体。这种将设计实体分成内外部分的概念是 VHDL 系统设计的基本点<sup>[15][16]</sup>。

由于 VHDL 电路描述语言所能涵盖的范围相当广，能适用于各种不同阶层的设计工程师的需求。从 ASIC 的设计到 PCB 系统的设计，VHDL 电路描述语言都能够派上用场，所以 VHDL 电路设计毫无疑问的成为硬件设计工程师的必备工具。

## 5.2 带有信元计数功能的 FIFO 模块的设计与实现

FIFO（先进先出队列）是交换机中的重要模块，用于数据的缓存和用于容纳异步信号的频率或相位的差异。FIFO 的实现通常是利用双口 RAM 和读写地址产生模块来实现的，如图 5.2 所示。

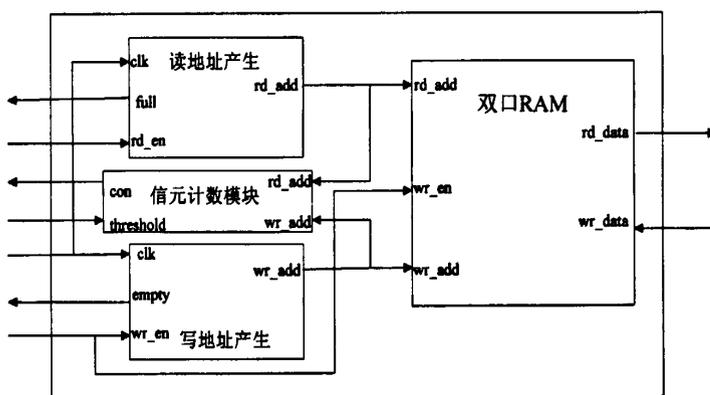


图 5.2 FIFO 逻辑功能示意图

FIFO 的接口信号包括时钟（clk）、与时钟同步的写有效（wr\_en）和写数据（wr\_data）、与时钟同步的读有效（rd\_en）和读数据（rd\_data）。为了实现正确的读写和避免 FIFO 的上溢或下溢，通常还应该给出与读时钟同步的 FIFO 的空标志（empty）和满标志（full）以禁止读写操作，并且为了交换机对于缓存空间的控制，提供了拥塞门限设定信号(threshold)，此拥塞门限是对信元进行计数的，当 FIFO 队列达到此门限值对应信元个数的时候就会置位相应的拥塞指示信号(con)。

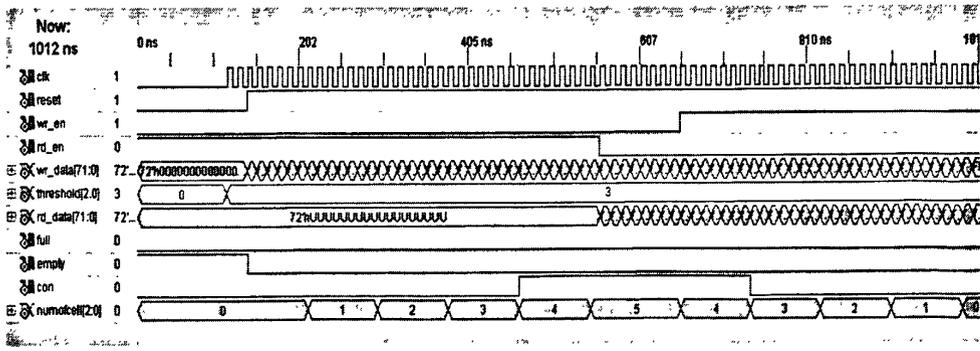


图 5.3 FIFO 模块仿真图

图 5.3 即是 FIFO 的 VHDL 仿真图，图中仿真的时钟周期是 12ns，也即是 83.3M(文中所出现的仿真都是基于此时钟周期)。图中所有的使能信号(reset, rd\_en, wr\_en)都是低电平有效,空满表示信号以及拥塞表示信号都是高电平有效(full, empty, con) (文中所出现的此类信号都是如此定义)。由图 5.3 中可以看出输入端口设定的拥塞门限(图中 5.3 中 threshold 信号)为 3 个信元,当 FIFO 中的信元个数达到 3 个的时候,拥塞信号 con 值即被置高,为拥塞控制提供反压指示<sup>[17][18]</sup>。

### 5.3 接收端信元头处理模块的设计实现

接收端信元头处理模块是整个接收流程的关键模块,其工作示意图如图 5.4 所示,阴影部分即是接收端信元头变换模块。

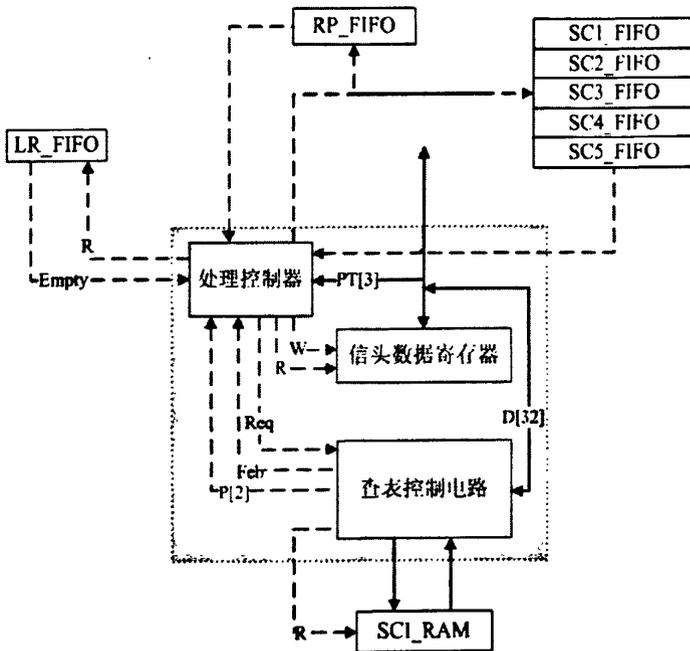


图 5.4 接收端信元头处理模块逻辑示意图

其工作的流程在于：从 LR\_FIFO 中读取保存有从 UTOPIA 接口传输过来的信元信息，并且判断是否为信令信元，如果是信令信元直接将其存储到 RP\_FIFO 中；如果是用户信元则将根据其信元头内容对存储有变换表的 SCI\_RAM 进行查表判断是 VP 交换还是 VC 交换，如果是 VP 交换则直接得到交换信息(内部信道号、交换优先级、TAG 值等)，将其变换成为内部信元(表 5.3 所示结构)，如果是 VC 交换则需要进行二次查表才能得到交换信息组成内部信元。组成的内部信元根据去往端口的不同分别存放到对应的 FIFO 中(即 SCn\_FIFO, n=1、2、3、4、5)，其中 SC1\_FIFO、SC2\_FIFO、SC3\_FIFO、SC4\_FIFO 分别存储去往端口 1、2、3、4 的信元，SC5\_FIFO 保存多播信元<sup>[23]</sup>。

按照上述描述，接收端信元头处理模块的功能采用状态机来实现，其状态转移的流程图如图 5.5 所示：

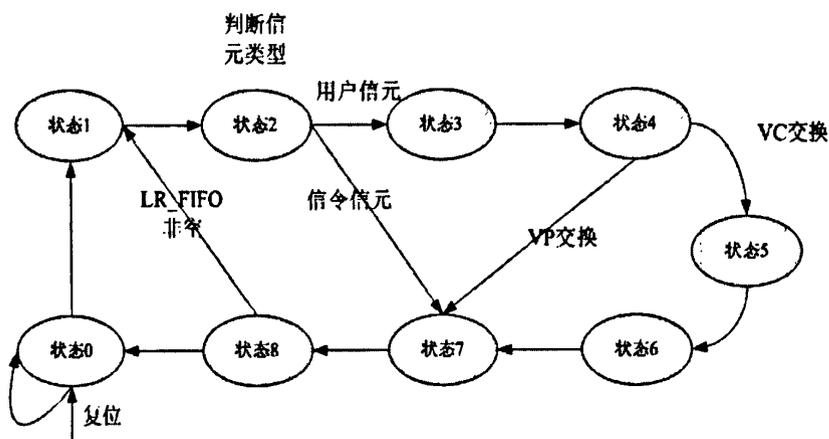


图 5.5 接收端信元头处理模块状态转移示意图

利用 VHDL 语言描述上述状态机原理<sup>[19]</sup>，进行仿真，如图 5.6 所示。图中对三种不同的信元进行了变换处理的仿真：信令信元、用户信元的 VP 交换、用户信元的 VC 交换。

图 5.6 中，第一个从 LR\_FIFO 中读出的(datain 所示数据)是信头(所有的读信号都是低电平有效)为 9FFFFFFF00BBBBBBB 的信元，由信元定义可知 PT[2:0]为 000 是一个信令信元，置位对应 RP\_FIFO 的写有效信号 rp\_fifo\_w，将其头部和载荷部分直接存入 RP\_FIFO 中。

第二个从 LR\_FIFO 中读出的是信头为 9FFFFFFF0FFFFFFF 的信元，这是一个用户信元，此信元的 VI[1:0]值为 01，利用公式  $VI * 2^2 + VPI \bmod 2^2$  可以得到 SCI\_RAM 的一次查表地址 7，通过对 SCI\_RAM 进行查找 VI\_VPI\_TABLE 得到地址 7 所对应的内容为 C0EEEEEE，由 VI\_VPI\_TABLE 的定义可知，这是一个 VP



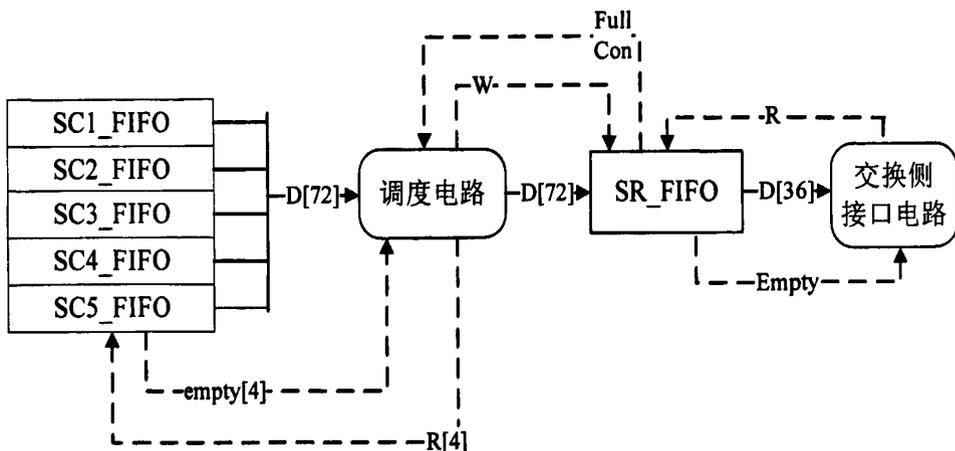
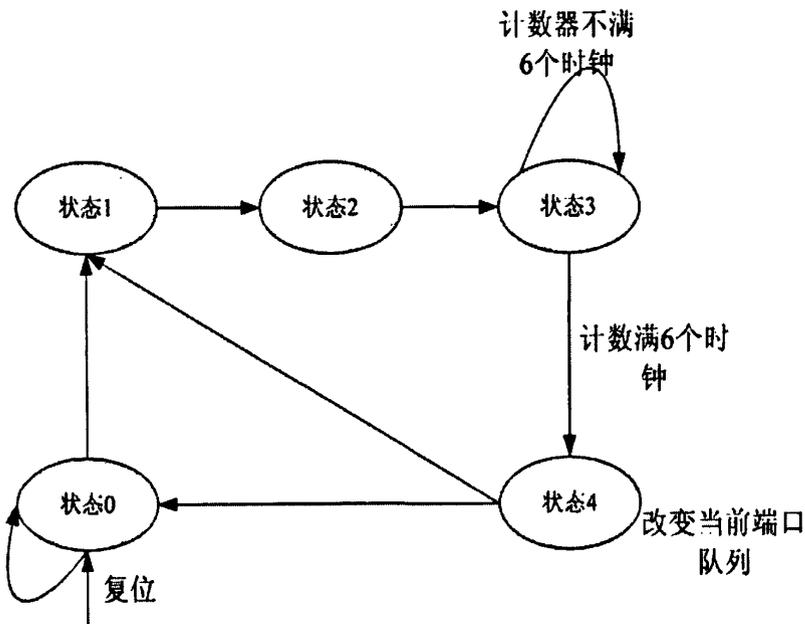


图 5.7 接收端调度模块逻辑示意图

在信元发送选择中，我们选择的调度策略是：采用队列轮询的策略，即五个队列轮流读取一个信元。

调度模块的上述功能可以使用状态机来实现<sup>[20]</sup>。其状态机工作示意图如图 5.8 所示：



按照上述状态及设计的思想，其 VHDL 语言实现的逻辑仿真如图 5.9 所示：

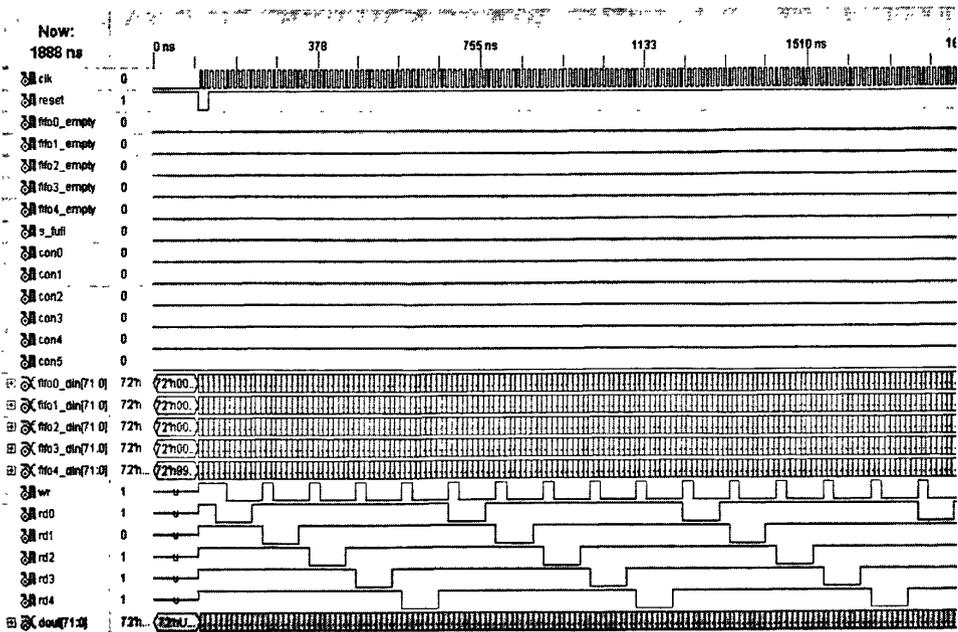


图 5.9 接收端调度模块逻辑示意图

从图中可以看出，五个队列 FIFO 的读信号(图 5.9 中 rd0~rd4 所示——低电平有效)依次有效，读 7 个时钟周期，输出一个完整的信元，从调度端口输出。

### 5.5 72bit 到 36bit 位宽转换 FIFO 的设计

为了增加信元预处理内部的处理速度，我们采取的内部数据总线为 72bit，但是在系统中我们设计的 ATM 标准接口 UTOPIO 接口的数据位宽为 36bit，这需要将 72bit 位宽的数据转换为 36bit 位宽，我们设计的 SR\_FIFO 即完成了这样的功能，其示意图如 5.10 所示。

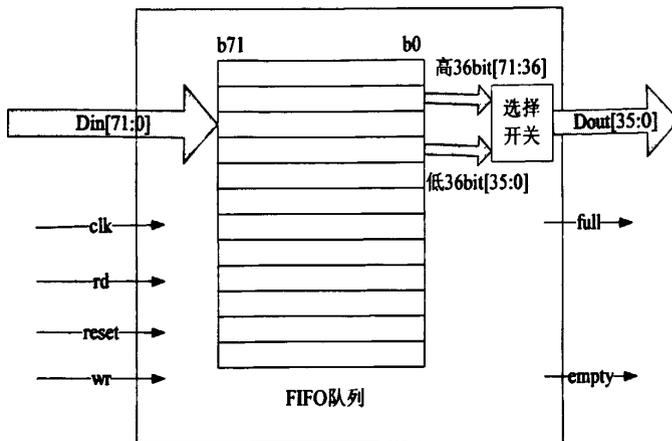


图 5.10 72bit 到 36bit 位宽转换 FIFO 示意图

完成位宽转换功能的 FIFO 设计思想是：在 FPGA 内部开辟位宽为 72bit，深

度为 14 的存储区,也即是缓存两个信元。在信元写入的时候和普通的 FIFO 操作相同,区别在于信元读出的时候,也即是同一个数据(7 和 14 地址空间中的数据除外,只读取高 36bit,因为低 36bit 为无效信息)读取两次:首先读出高 36bit 的数据,然后读取低 36bit 的数据,然后移动读指针,这样就完成了 72bit 到 36bit 的位宽转换。其仿真如图 5.11 所示:

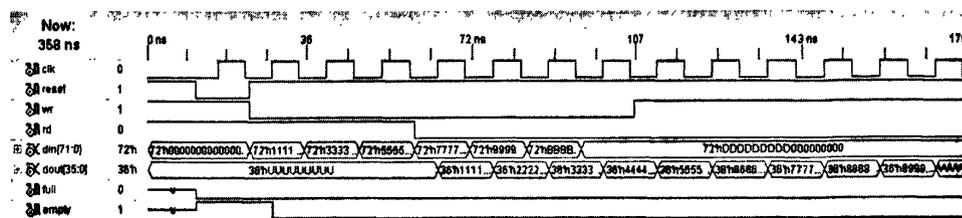


图 5.11 72bit 到 36bit 位宽转换 FIFO 仿真图

我们首先在位宽转换 FIFO 中写入一个信元(位宽为 72bit),其存储形式如表 5.2 所示。

表 5.2 72bit 到 36bit 位宽转换 FIFO 输入数据

地址	信元数据(十六进制形式)
1	111111111122222222
2	333333333344444444
3	555555555566666666
4	777777777788888888
5	9999999999AAAAAAAA
6	BBBBBBBBBCCCCCCCC
7	DDDDDDDD00000000

经过仿真可以看出输出的 36bit 信元流为:从 11111111 到 DDDDDDDD 共 13 个 36bit 依次输出,完成了数据宽度的变换,并且丢弃了原第 7 个地址的后 36bit(无用信息,填充比特)。

## 5.6 36bit 到 72bit 位宽转换模块的设计

从交换模块发送过来的数据经过 UTOPIA Level 1 接口发送到预处理模块需要将 36bit 的数据宽度增加到 72bit,以提高数据处理的速度。以一个信元的转换为例,其设计的思想是:一个信元有 13 个 36bit 的数据组成,利用状态机,首先将第一个 36bit 在寄存器中保存起来,然后将第二个 36bit 与寄存器中的 36bit 连接起来输出,这样就形成了 72bit 数据宽度的组合。把以上步骤重复 6 次,形成了 6 个 72bit 宽的数据,最后一个 36bit 数据与 36bit 的 0 连接输出,至此一个信元的转换完毕。

其设计状态转移示意图如 5.12 所示:

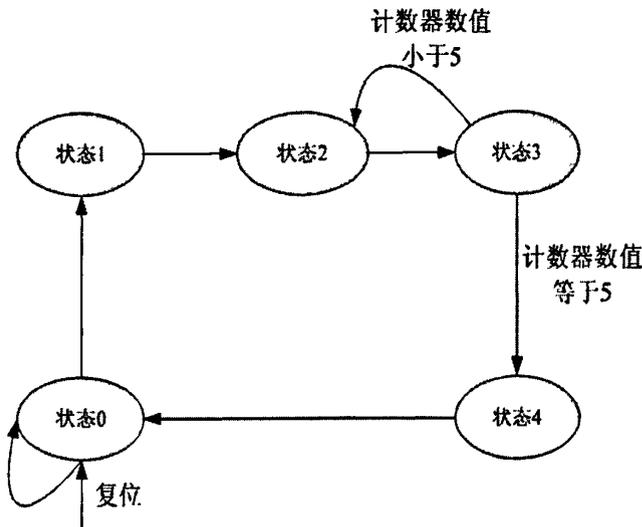


图 5.12 36bit 到 72bit 位宽转换 FIFO 状态转意图

按照以上设计思想, 利用 VHDL 语言进行描述, 其仿真结果如图 5.13 所示。

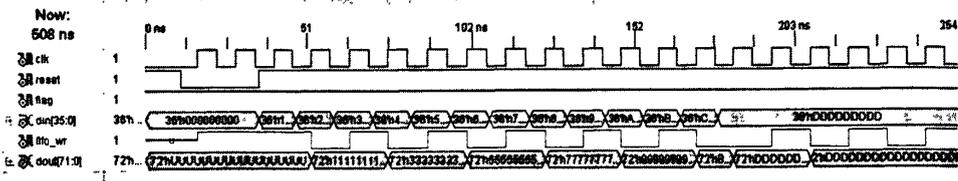


图 5.13 36bit 到 72bit 位宽转换 FIFO 仿真图

图 5.13 中仿真了一个信元即 13 个 36bit 数据到 7 个 72bit 数据的转换, 仿真中此模块完成了如图 5.14 所示的转换。

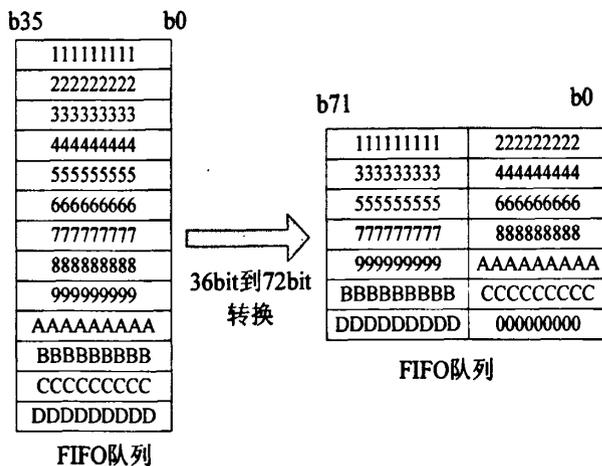


图 5.14 36bit 到 72bit 位宽转换示意图

## 5.7 发送端信头处理模块的设计实现

发送端信头处理模块是发送方向信元处理的核心模块，主要完成交换模块出来的内部信元到外部信元的转换，也即是用新的 VPI/VCI 值替换下内部信元的交换标签、内部信道号、优先级等信息生成外部信元。

发送端信头处理模块主要的工作流程是：从 ST\_FIFO 中读出数据，判断是否为信元的头部，如果不是信头则继续读出数据直到找到信头为止，如果是信元的头部则进行下述处理：首先根据信头信息得到信元变换查找表 CHE\_RAM 的查表地址，对 CHE\_RAM 进行查表得到新的 VPI/VCI 值替换原来的内部信道号等信息得到外部信元信头，连同载荷部分一起输出到多播 FIFO(MC\_FIFO)中，其工作示意图如图 5.15 所示：

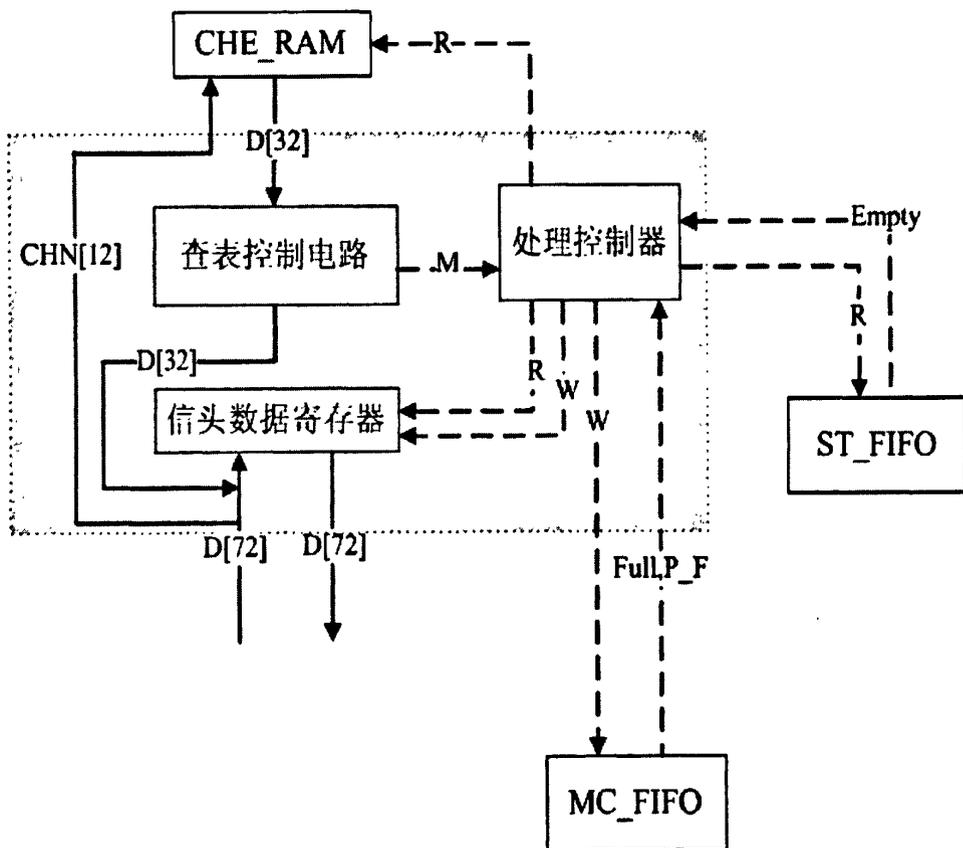


图 5.15 发送端信头处理模块逻辑示意图

其电路逻辑功能可以利用状态机来实现，其状态转移的流程图如图 5.16 所示：

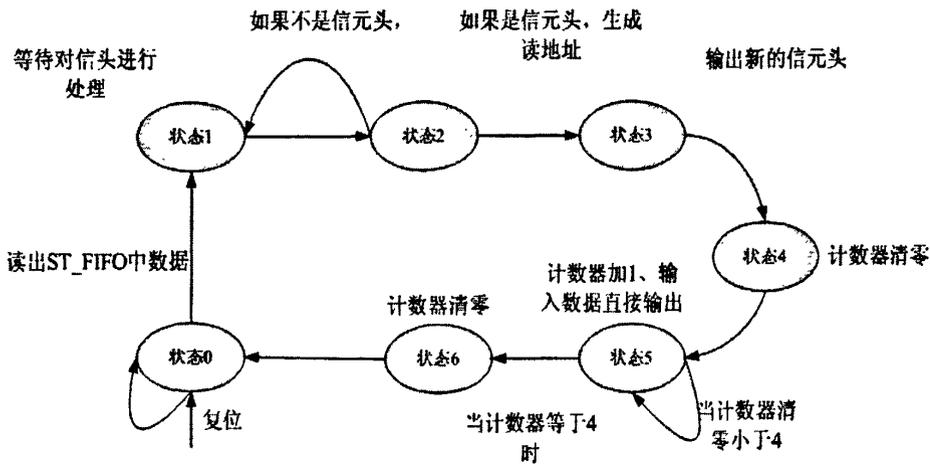


图 5.16 发送端信头处理模块状态转移图

按照上述设计思想，利用 VHDL 描述实现，在 ISE 平台上进行仿真如图 5.17 所示：

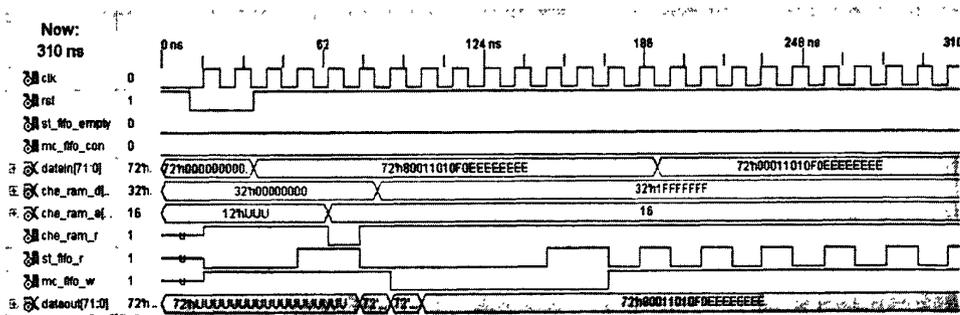


图 5.17 发送端信头处理模块仿真图

从图 5.17 可以看到，第一个输入的信元信头为 80011010F0EEEEEEEE，由定义可知对应的信道号为 1000(二进制形式)也即是查表的地址，通过查找 CHE\_RAM 得到变换信息：1FFFFFFF(出信道号和新的 VPI/VCI)，进而组成外部信元信头：8FFFFFFF1EEEEEEEE 并连同载荷一起输出，并且由图中所示当出现异常 ST\_FIFO 读出的数据经判断不是信头的时候则丢弃，读出下一个数据，直到判断出为信头为止。如图 5.17 中所示，当接收的信元是 00011010F0EEEEEEEE 时，第一比特为 0，则由定义可知不是信头，则予以丢弃重新读取下一个信元如图 5.17 中 st\_fifo\_r 所示。

### 5.8 多播模块的设计实现

多播模块(MC\_FIFO)完成信元的多播复制工作。由信元预处理的工作示意图 4.2 可知当从交换芯片发送来的信元经过信头变换以后得到了由新的 VPI/VCI 和输

出信道号组成的信元。由于一个信元预处理芯片连接四个物理端口，当信元需要进行多播发送的时候，多播模块即是要根据信元的输出信号产生去往不同端口的多播信元。输出信道号  $OUT\_NUM[3:0]$  由四比特表示，哪一位比特为 1 代表信元去往哪个端口，如果四比特都为 1 则表示信元将去往全部的四个端口也即是广播。所以多播模块的工作流程是：首先根据输入信元的输出信道号判断其多播属性产生每个去往每个端口的输出端口值  $VO$ ，替换输入信元的输出信道号组成等待输出的信元。

按照以上思想，利用状态机完成了此模块的 VHDL 语言描述，其仿真如图 5.18 所示。在仿真中我们首先输入的是信头为  $8FFFFFFFCFFFFFFF$  的信元，由输入信元的定义表 4.7 可知，此信元的输出信道号  $OUT\_NUM[3:0]$  二进制形式为 1100 代表这是一个要去往端口 4 和端口 3 的多播信元，根据多播的端口确定输出口值分别为 11 和 10，则分别产生了两个分别去往端口 3 和 4 的信头  $AFFFFFFF0FFFFFFF$  和  $BFFFFFFF0FFFFFFF$  加上相同的载荷从输出口输出，当一个多播完成时即输出一个高电平的多播完成信号  $complete$ 。第二个输出的是信头为  $8FFFFFFF0FFFFFFF$  的信元，由于其输出信道号的二进制形式为 1111，则这是一个广播信元，则同样按照上述思想复制四次完成多播复制，然后再输出一个高电平的多播完成信号  $complete$ ，如图 5.18 中所示。

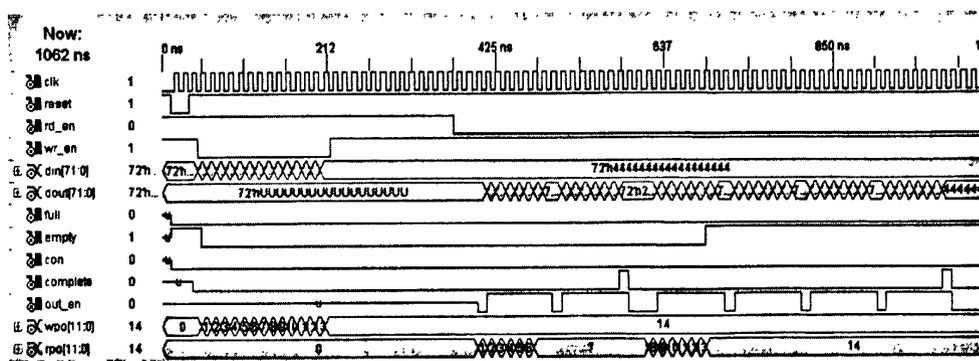


图 5.18 多播模块仿真图



## 结束语

卫星通信和 ATM 交换的结合是未来通信发展的重要方向,因此具有星上处理功能的交换机是当今的研究热点。本文在介绍星上 ATM 交换机整体设计的基础上,重点研究了信元预处理部分基于 FPGA 的设计实现,文中主要完成的工作为:

1. 在简述 ATM 交换基本原理基础上,按照技术指标给出了星上大容量 ATM 交换机的总体设计方案。该方案对整个交换机进行结构的划分,并且阐述了各个部分的具体功能。
2. 对信元预处理部分进行了详细的设计。信元预处理包括两个方向:接收方向和发送方向,接收方向主要完成了外部信元到内部信元的转换,发送方向主要完成了内部信元到外部信元的转换。文中对本部分的具体功能进行详细论述,并对具体子模块进行了信号的设计、参数的选择。
3. 在 ISE 仿真平台上实现了对信元预处理子模块基于 FPGA 的 VHDL 实现,并进行了各个模块的级联仿真,结果证明,它们可以很好地工作,实现了信元预处理的功能。

由于时间及其他各方面的因素限制,本文的研究工作中还存在一些不完善的地方。可以在以下方面进行更深入的研究:

1. 本文完成了星上 ATM 交换机信元预处理部分的基本功能,对于信令部分的考虑较少,可以进一步对 UNI、NNI 信令进行系统的设计。
2. 对信元预处理部分的功能进行了基于均匀业务下的测试,尚需对其他情况比如突发业务进行详细的测试。



## 致谢

首先衷心的感谢我的导师田斌副教授近三年来对我的精心培养和指导，田老师在通信领域深厚的学术造诣、广博的理论知识、活跃的学术思想和敏锐的洞察力使我对通信系统的知识有了更加深入的了解和认识；田老师严谨求索的治学理念、孜孜不倦的工作精神、宁静淡泊的生活态度一直激励着我，将永远是我做人的榜样。

感谢易克初教授在学习上给我们的诸多指导。他强烈的事业心、勇于创新的精神、饱满的工作热情和诲人不倦的师长风范深深的影响着我，是指引我人生道路的灯塔。

感谢田红心老师给我工作上的指导和生活上的关心。他对同学的关心和热情爽朗的性格，都给我留下了深刻印象。有了田老师对实验室的管理和对同学的严格要求，才有了实验室今天良好的学习和科研氛围。

感谢刘祖军老师在项目研究中给予的指导帮助，在刘老师耐心的指导下才使我解决了许多的问题，取得了些成绩，同时感谢王勇超老师和孙永军老师在我学习期间给予的悉心指导，他们耐心的态度、渊博的知识、谦虚的为人是我永远的榜样。

感谢焦毅博士在项目中给我的指导，感谢我的小师弟张聃的帮助，和你们的合作真的很愉快，因为有了你们的帮助才能使我顺利的完成了项目研究。

感谢与我同届的硕士生们，李怡、倪光华、王自保、左艳强、孙黎明、李德屹、安金坤、周振华、黄福林、郭永林、李保中、刘婷婷、王莎莎、高健、阎坤等，无论是专业知识的讨论还是生活上的交流，与他们的相处都使我获益很多。特别感谢同项目组的李怡，感谢你在生活上、学习上给我的诸多帮助，研究生能认识你是我一生最大的庆幸！

特别感谢好友杨治家、王自保在我最困难的时候予以的关心、照顾，是你们伴随我度过了那一段岁月，愿我们的友谊天长地久！

感谢下一届的师弟师妹：张聃、宋薇、许梅、安英杰等在论文中给予的帮助。

深深地感谢我的父母，正是他们的支持使我有机会实现读研的梦想，有勇气和责任去面对生活中的各种困难。我取得的每一点成绩和你们一直以来在学习和生活上给予的大力支持是密不可分的。你们的支持是我最坚强的后盾！



## 参考文献

- [1] 赵慧玲, 张国宏, 胡琳等. ATM、帧中继、IP 技术与应用. 北京: 电子工业出版社, 1998 年。
- [2] PryckerMde 著. 程时端, 刘斌译. 异步传递方式— 宽带 ISDN 技术. 北京: 人民邮电出版社, 1995 年。
- [3] 李生红, 刘泽民. ATM 网上一种简单通用的业务模型. 通信学报. 2000 年, 2 月, 21 卷第 1 期. pp.5-7。
- [4] 付立政, 冷晓冰. ATM 接入交换机上行链路性能的仿真研究. 通信学报. 2000 年, 4 月, 21 卷第 10 期. pp.6-18。
- [5] 李津生, 洪佩林, 陈意云. 宽带综合业务数字网与 ATM 局域网. 北京: 清华大学出版社, 1998 年。
- [6] 刘亚社, 刘增基. 中大规模 ATM 交换网络结构. 西安电子科技大学学报. 1997 年, 2 月, 第 24 期. pp.22-24。
- [7] 刘亚社, 刘增基, 胡征. 具有纵横输入互连方式和缓冲结构的递归 Knockout 交换网络. 电子学报. 1999 年, 4 月, 第 4 期. pp.13-14。
- [8] 刘亚社, 刘增基. CLOS 网络型大规模 ATM 交换结构. 通信技术. 1997 年, 3 月, 第 1 期. pp.25-27。
- [9] 刘亚社, 刘增基. CLOS 型 ATM 交换网络结构与路由选择算法. 通信技术, 1998 年, 6 月, 第 3 期. pp.19-21。
- [10] 刘亚社, 刘增基. CLOS 型大规模 ATM 交换网络结构. Radio Communications Technology. 1997 年, 6 月, 第 2 期. pp.25-26。
- [11] 戴礼森, 王大伟, 洪佩琳等. 8 端口 ATM 工作组交换机设计与性能分析. 通信学报. 2001 年, 4 月, 第 4 期. pp.29-31。
- [12] 侯伯亨, 顾新. VHDL 硬件描述语言与数字逻辑电路设计. 西安: 西安电子科技大学出版社, 1997 年。
- [13] 王小军著, 乔长阁等译. VHDL 简明教程. 北京: 清华大学出版社, 1997 年。
- [14] D.L.PERR 著. 周祖成译. 电子设计硬件描述语言 VHDL. 北京: 学苑出版社, 1994 年。
- [15] 黄建文. VHDL 语言及其应用. 北京: 中国铁道出版社. 1997 年。
- [16] 常青. 可编程专用集成电路及其应用与设计实践. 北京: 国防工业出版社, 1998 年。
- [17] 刘明业. 专用集成电路高级综合理论. 北京: 北京理工大学出版社, 1998 年。

- [18]汤继华.常用集成芯片使用.北京:北京理工大学出版社,1995年。
- [19]卢毅,赖杰.VHDL与数字电路设计.北京:科学出版社,2001年。
- [20]徐志军,徐光辉.CPLD/FPGA的开发与应用.北京:电子工业出版社,2002年。
- [21]陈锡生.ATM交换技术.北京:人民邮电出版社,2000年。
- [22]白耀术,吴家安.多优先级ATM交换机性能的模拟.电信交换.1998年,6月,第2期,pp.30-32。
- [23]H. Obara, S. Okamoto, Y Hamazumi. Input and output queueing ATM switch architecture with spatial and temporal slot reservation control. Electron. Lett. (January 1992).pp.22-24.
- [24]Ji-Hong Liaw, Cheng-Ming Weng. Modular Multicast ATM Switching Architecture. Computer Communications.Vol.19,No.12, Dec.1995.pp.32-33。
- [25]H.P.Kuhlen. An On-Board Processor for ISDN and ISDN-Compatible Applications. Elsevier Science Publishers B.V, 1988.

## 作者在读期间的研究成果

- [1] 吴威, 田斌 . ATM 交换机信元预处理的 FPGA 设计与实现. 西安电子科技大学通信工程学院 2007 年度研究生学术年会. 2007

