

# 硕士学位论文

## 时间触发 AFDX 终端系统的虚拟链路层 调度算法研究

### RESEARCH ON VIRTUAL LINK LAYER SCHEDULING ALGORITHM OF TIME TRIGGERED AFDX END SYSTEM

李莹雪

哈尔滨工业大学

2014 年 6 月

国内图书分类号：TP271+.5  
国际图书分类号：621. 3

学校代码：10213  
密级：公开

## 工学硕士学位论文

# 时间触发 AFDX 终端系统的虚拟链路层 调度算法研究

硕 士 研 究 生：李莹雪

导 师：吴辉 副教授

申 请 学 位：全日制工学硕士

学 科：电气工程

所 在 单 位：电气工程及自动化学院

答 辩 日 期：2014 年 6 月

授予学位单位：哈尔滨工业大学

Classified Index: TP271+.5

U.D.C: 621.3

Dissertation for the Master Degree in Engineering

**RESEARCH ON VIRTUAL LINK LAYER  
SCHEDULING ALGORITHM OF TIME  
TRIGGERED AFDX END SYSTEM**

<b>Candidate :</b>	Li Yingxue
<b>Supervisor :</b>	A.Prof. Wu Hui
<b>Academic Degree Applied for :</b>	Master of Engineering
<b>Specialty :</b>	Electrical Engineering
<b>Affiliation :</b>	Dept. of Electrical Engineering
<b>Date of Defence :</b>	June, 2014
<b>Degree-Conferring-Institution :</b>	Harbin Institute of Technology

## 摘 要

随着航空电子系统综合模块化发展，航空通信系统任务不断扩展，传统的数据总线已不能满足数据高速可靠传输的要求。目前先进的数据总线航空电子全双工交换式以太网（AFDX）在带宽、总线结构、数据传输机制等方面都作出很大改善，但是在保证消息传输的确定性方面仍存在有待改进之处。终端系统是 AFDX 网络中的重要组成部分，协议栈中各个不同的分层共同实现数据传输的功能。为更好满足航空电子通信系统高可靠性、硬实时性以及严格确定性的需要，本文的主要研究内容是为提高 AFDX 通信网络确定性能对终端系统虚拟链路层调度算法进行改进。

首先，介绍课题的研究背景和意义，论述航空数据总线的发展现状，比较传统航空数据总线 MIL-STD-1553B、ARINC 429 以及 ARINC 629 的优缺点，论证 AFDX 成为飞机主干通信网络的优越性。分析了 AFDX 终端系统常用到的调度算法以及保证数据传输确定性的时间触发通信协议，阐述了时间触发机制可借鉴之处，指出为提高网络通信确定性能需要做出改进的部分。

其次，结合时间触发通信机制，综合考虑航空电子通信任务的实时性和确定性要求，设计了时间触发 AFDX 终端系统虚拟链路层的调度算法。构建 AFDX 网络模型，分析时间触发调度策略的调度过程并与目前常用的调度策略进行比较，在理论计算与仿真实验两个方面说明时间触发调度策略的优越性能。

再次，分析 AFDX 终端系统的协议栈和虚拟链路层的功能，提出虚拟链路层总体结构，重点对发送和接收部分进行详细的模块化设计，完成基于 FPGA 的终端系统虚拟链路层的功能实现，并与 MAC 子层模块联调完成功能性仿真。

最后，构建终端系统通信性能的测试实验平台，验证终端系统正常情况下的数据交换功能，并在发送终端系统发送几种特殊情况数据帧的情况下，测试接收端对数据帧的处理功能。

**关键词：**AFDX；时间触发机制；调度算法；时间确定性；FPGA

## Abstract

With the development of integrated modular avionics system, expanding aviation communication system task, traditional data bus cannot meet the requirements of high speed and reliable data transmission. At present advanced data bus Avionics Full duplex Switched Ethernet has made great improvements in many aspects such as bandwidth, bus structure, data transmission mechanism and so on. But there are still some aspects to be improved in guarantee deterministic message transmission. End system is an important part of AFDX network, and its protocol stack various stratified joint can realize the function of data transmission. To meet the needs of avionics communication system with high reliability, real-time and strictly deterministic, the main research content of this article is to optimize scheduling algorithm to improve the network performance in AFDX end system virtual link layer.

Firstly, introduce the research background and significance. Discuss the current situation of the development of aviation data bus and argument the superiority of AFDX by comparing with several typical data bus such as MIL-STD-1553B, ARINC429 and ARINC629. Analysis scheduling algorithm used by AFDX end system at present and time triggered communication protocol, this paper uses the time triggered mechanism to improve the communication deterministic performance of AFDX network.

Secondly, consideration the demand of avionics communication, design time triggered scheduling algorithm of AFDX end system virtual link layer combining with the time triggered communication mechanism. Construct AFDX network model and analyze process of time triggered scheduling algorithm. Compare with commonly used scheduling policy in theoretical calculation and simulation experiment, to show the superior performance of time triggered scheduling policy in AFDX network.

Thirdly, analysing the protocol stack of AFDX end system and the function of the virtual link layer, the general structure of the virtual link layer is put forward. The key is to design transmitter and receive part of the modular in detail. Realize the function of the end system virtual link layer based on FPGA, and complete the functional simulation in alignment with the MAC sub-layer module.

Finally, based on the system platform, test end system the data exchange function under normal circumstances, and under the condition of the transmitter end system to send special data frame, to verify the data frame processing functions of

receiving end system.

**Keywords:** AFDX, Time-Triggered Mechanism, Scheduling Algorithm, Time Determinacy, FPGA

# 目 录

摘 要 .....	I
ABSTRACT .....	II
目 录 .....	IV
第 1 章 绪 论 .....	1
1.1 课题来源及研究的背景和意义 .....	1
1.2 本课题的国内外研究现状 .....	2
1.2.1 航空数据总线发展概况 .....	2
1.2.2 常用调度算法 .....	5
1.2.3 时间触发通信协议发展现状 .....	6
1.3 本课题的主要研究内容 .....	6
第 2 章 时间触发调度算法及其确定性 .....	8
2.1 时间触发调度算法 .....	8
2.1.1 时间触发 AFDX 终端系统协议栈 .....	8
2.1.2 时间触发调度算法设计 .....	9
2.2 时间触发 AFDX 网络建模及其确定性分析 .....	13
2.2.1 时间触发 AFDX 网络确定性分析 .....	13
2.2.2 时间触发 AFDX 网络建模及其延时计算 .....	14
2.3 静态优先级调度算法及其确定性分析 .....	19
2.3.1 静态优先级调度算法 .....	19
2.3.2 网络演算理论 .....	20
2.3.3 AFDX 网络在静态优先级调度下确定性分析 .....	21
2.4 本章小结 .....	23
第 3 章 基于 TRUETIME 平台的网络仿真分析 .....	24
3.1 TRUETIME 平台介绍 .....	24
3.2 TRUETIME 仿真模型构建 .....	25
3.2.1 终端系统仿真模型 .....	25
3.2.2 交换机仿真模型 .....	26
3.3 AFDX 网络调度算法仿真及分析 .....	27
3.4 本章小结 .....	30
第 4 章 时间触发 AFDX 虚拟链路层的 FPGA 实现 .....	31
4.1 虚拟链路层分析与 FPGA 实现的总体设计 .....	31

4.1.1 MAC 子层模块 .....	31
4.1.2 虚拟链路层发送模块 .....	32
4.1.3 虚拟链路层接收模块 .....	33
4.2 虚拟链路层发送模块实现 .....	34
4.2.1 发送模块结构 .....	36
4.2.2 存储器模块 .....	36
4.2.3 寄存器模块 .....	36
4.2.4 调度模块 .....	37
4.2.5 发送冗余管理模块 .....	40
4.2.6 终端系统发送模块仿真 .....	41
4.3 虚拟链路层接收模块实现 .....	42
4.3.1 接收模块结构 .....	42
4.3.2 完整性检查模块 .....	43
4.3.3 接收冗余管理模块 .....	44
4.3.4 终端系统接收模块仿真 .....	45
4.4 虚拟链路层全双工仿真 .....	47
4.5 本章小结 .....	48
第 5 章 时间触发 AFDX 终端系统性能测试 .....	49
5.1 测试系统实验平台 .....	49
5.2 终端系统通信演示 .....	50
5.3 终端系统冗余功能测试 .....	51
5.3.1 终端系统错误帧处理测试 .....	51
5.3.2 终端系统复位帧处理测试 .....	52
5.3.3 终端系统丢帧情况测试 .....	53
5.4 本章小结 .....	54
结 论 .....	55
参考文献 .....	56
攻读硕士学位期间发表的论文及其它成果 .....	60
哈尔滨工业大学学位论文原创性声明及使用授权说明 .....	61
致 谢 .....	62



# 第1章 绪 论

## 1.1 课题来源及研究的背景和意义

本课题来源于伊顿宇航大学计划项目《航空电子全双工交换式以太网终端系统及驾舱可编程触摸屏控制面板的设计研发》，根据 ARINC664 标准中 Part7 对航空电子全双工交换式以太网（Avionics Full Duplex Switched Ethernet, AFDX）的定义，开发 AFDX 的终端系统，本文从构建确定性网络任务出发，在调度算法的设计、网络性能分析以及终端系统的实现等方面展开研究，并进行了详细论述。

机载数据总线被认为是航空电子综合系统的“中枢神经”，用于机载设备和子系统之间的互连，承担着飞机上各个子系统以及模块之间信息交换的重要任务<sup>[1]</sup>，因而机载数据总线的发展成为航空电子技术发展的驱动力之一。随着航空电子系统的综合化、网络化和模块化发展，航空通信系统的通信任务扩展为高速数据、图像、多媒体以及语音等多种任务<sup>[2]</sup>，对航空电子机载总线网络的实时性和可靠性提出了更高的要求，存储器、微电子技术、逻辑器件的发展为数字数据总线的发展提供了技术支持，使数据总线的快速发展和航空电子系统的综合化变革成为可能。传统的 MIL-STD-1553B、ARINC429 及 ARINC629 等总线的数据传输网络虽然发展比较成熟，已经在一些军用飞机和民用飞机上有所应用，但是由于传输速率和工作方式等方面的限制不能成为飞机的主干网络。在以太网技术基础上发展起来的新型的航空数据总线 AFDX 是一种具有确定性的网络，AFDX 良好的数据传输机制减少机载负荷和系统布线，满足当前航空电子综合系统数据传输的容错性、实时性和可靠性的需求，为航空电子系统的发展奠定基础，作为飞机的主干网络已经成功应用在空客 A380 和波音 787 等民用飞机及 A400M 等军用飞机<sup>[3]</sup>。

终端系统是 AFDX 网络一个重要组成部分，其性能直接影响整个 AFDX 网络的通信性能。目前终端系统中用到的调度策略都会在多任务传输时给整个网络带来不同程度的不确定性，影响紧急消息传输的实时性。由 TTTech 公司提出的时间触发网络通信协议（Time-Triggered Protocol, TTP）采用时间触发通信机制，能够实现在规定时间序列内完成数据的收发，具有很高的确定性和可靠性<sup>[4]</sup>。时间触发架构下一切行为不受事件触发，完全由时间引导，解决消息对链路的共享冲突，消息的传输具有严格确定性，已有航空航天科研单位开始开发和预研<sup>[5]</sup>。

综合看来,时间触发机制可以成为解决 AFDX 网络中调度任务冲突的一项可行方案,为满足新一代航空电子数据总线的高实时性和确定性要求提出发展方向。

本课题将时间触发通信机制应用于 AFDX 终端系统的虚拟链路层调度策略设计中,设计终端系统虚拟链路层的时间触发调度算法;通过理论计算和仿真实验验证基于时间触发机制的调度策略在保证数据传输确定性方面与常用调度策略相比有很大的改善,并且对于紧急信息的传输有效地提高实时性;对 AFDX 协议和时间触发机制深入分析,实现时间触发调度策略下 AFDX 终端系统之间可靠地数据通信;另外,时间触发 AFDX 终端系统与普通 AFDX 终端系统的硬件设计完全一致,便于时间触发 AFDX 终端系统的实现,兼顾开发成本和终端系统之间的兼容性。综上所述,本课题对 AFDX 网络数据传输确定性的改善和时间触发 AFDX 网络的构建具有十分重要的意义。

## 1.2 本课题的国内外研究现状

航空电子系统结构经历了四代的不断演变,由早期的分立式结构、联合式结构向综合式结构以及先进的综合航空电子结构发展<sup>[6]</sup>,已使航电系统的体积和重量大大减少,满足可靠性、稳定性、易维护性等原则,综合模块化向着更广的方向发展<sup>[7]</sup>。新一代航空电子系统的实现取决于高效的数据传输机制,对数据总线提出了分布式处理、抗干扰、高可靠性等方面的要求,航空电子网络的互联技术也从低速的总线互联向高速的交换式网络方向转变,它是实现航空电子综合的一项核心支撑技术,数据总线的性能好坏直接影响到整个航电系统的数字化发展。

飞机任务需求不断扩展,各功能区的机载设备和航电子系统之间信息传输任务随之激增,对数据总线提出更高的要求,确定性与实时性成为衡量数据总线性能的关键指标,通信网络中数据帧的调度是影响通信性能的重要因素。在多任务的网络控制系统中,有限的带宽资源条件下,各类任务对实时性有不同的要求,多个任务同时访问网络时会在链路上产生共享冲突,多依赖于调度策略合理调度网络中各类任务以满足不同的实时性要求<sup>[8]</sup>。

### 1.2.1 航空数据总线发展概况

标准数字数据总线出现始于 1973 年美国空军 MIL-STD-1553 规范的出台,它是联合式航空电子系统的支柱,军用飞机采用这种总线连接机载电子设备,此项标准历经标准 A 版本和标准 B 版本的颁布,适应军事需要具有很高的可靠

性,今天的 1553 标准是在 B 版本基础上做出一些更改和补充<sup>[9]</sup>。MIL-STD-1553B 数据总线结构如图 1-1 所示,采用命令/响应式异步传输,利用一对屏蔽双绞线作为传输介质,半双工的传输方式传输速率为 1Mb/s,总线上挂接的终端分为总线控制器、远程终端和总线监控器,实际系统中应用两个总线控制器提供冗余<sup>[10]</sup>,每次数据传输受到总线控制器控制,总线控制器向数据总线发布指令,远程终端接收和解码来自总线控制器的命令,总线监控器用来监听数据总线上的信息交换<sup>[11]</sup>,若总线控制器发生故障会造成整个网络的瘫痪。带宽小、价格高、对故障敏感度强等缺点限制 MIL-STD-1553B 在民用飞机上的广泛应用。

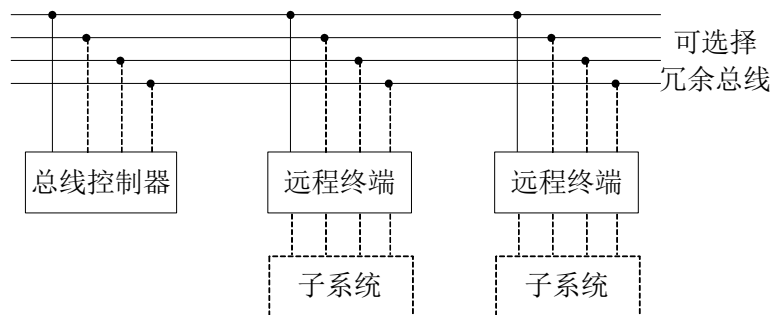


图 1-1 MIL-STD-1553B 总线结构

ARINC429 是第一个为民用飞机所使用的标准数据总线,由美国航空无线电公司于 1977 年发表并获得批准使用。ARINC429 是点对点的传输协议,最先应用于 Airbus 310 和 Boeing 767 等飞机上,因其高可靠性至今仍广泛应用到大多数的民用飞机中。如图 1-2 所示,ARINC429 数据传输由发送器单向发送信息经数据总线接收器接收并解码信号,每条总线单向传输但可以一发多收<sup>[12]</sup>,有 12.5kb/s 和 100kb/s 两种可选速率,结构简单,数据总线性能稳定。但是数据传输速率低,不能满足有大量数据传输或有多个数据源和接收设备在总线上互连的场合,每个信号源都连接一条 ARINC429 总线增加电缆重量。

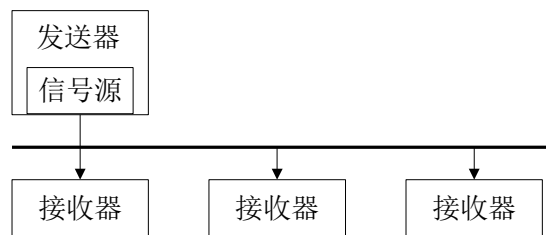


图 1-2 ARINC429 总线结构

为替代 ARINC429 开发了 ARINC629 数据总线,总线传输速率为 2Mb/s,拓扑结构与 MIL-STD-1553B 相同均为线型。总线上没有总线控制器,每个终端均为自主发送数据,数据在总线上实现双向传输。已经应用于 Boeing777 上,减少导线重量和连接器,终端数目也随之减少,增加系统的灵活性。但是较低

的传输速率仍不能满足作为飞机主干网络的要求。

航空电子全双工交换式以太网是由法国空客公司根据航空电子系统的需求建立起的确定性网络<sup>[13]</sup>，采用异步传输模式解决 IEEE802.3 以太网存在的延时缺点，实现数据传输的实时性与可靠性，在国外的大型飞机上如空客 A380 和波音 787 等民用飞机和 A400M 等军用飞机，采用 AFDX 为航电系统的主干网络，低数据量传输的机载设备利用 ARINC429 总线传输，两种总线数据间的转换通过无线电接口等设备完成。

由于 AFDX 的广泛应用前景，国内外各大公司和科研单位一直不断深入研究并开发出 AFDX 网络仿真、测试平台等相关产品。德国的 TechSAT 公司开发生产的产品针对 AFDX 终端系统的仿真、系统集成及压力测试与验证，德国 AIM 公司主要致力于开发分析 AFDX 的软件，可以实现测试、分析、仿真等功能，国内利用 AIM 公司的产品进行航空数据总线方面的测试。

AFDX 的数据传输速率可达 100Mb/s，采用光纤或铜质电缆为传输介质，星型拓扑结构，全双工工作方式，AFDX 的网络结构如图 1-3 所示，主要由航电子系统、终端系统和 AFDX 交换机组成。飞行控制计算机、监控系统、定位系统、娱乐系统、显示系统等机载设备组成航电子系统，AFDX 交换机对传输的数据过滤和监测<sup>[14]</sup>，终端系统作为将航电子系统和 AFDX 网络连接起来的接口进行安全和可靠地数据交换，是保证整个 AFDX 网络正常运行的关键环节。

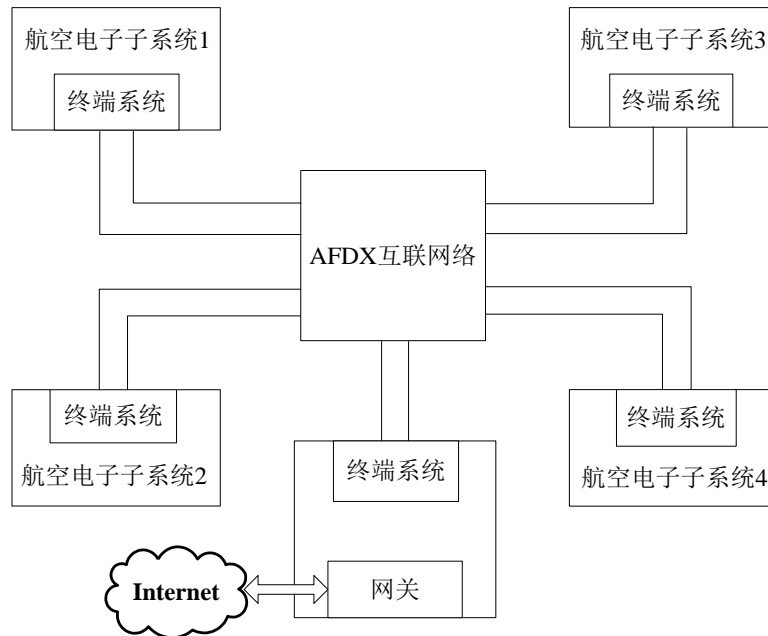


图 1-3 AFDX 网络结构

AFDX 应用于大飞机上作为主干网络与其它数据总线如 ARINC429、CAN

总线共同组成飞机上的整个通信系统，AFDX 表现出如下特点：

(1) AFDX 网络构建的航电系统遵循 OSI 分层模型协议，不同的层执行相应的功能，把协议规定需要实现的功能分散到不同的层中实现，更改某个层的功能不会影响到其它层的实现，使 AFDX 网络成为一个开放式的体系结构<sup>[15]</sup>。

(2) AFDX 利用交换机完成不同航电子系统之间的数据交换，电缆资源共享从而大量减小机载电缆重量，有利于实现飞机的大规模综合化。

(3) AFDX 提供冗余机制，数据帧在两条互为冗余的链路上传输同时发往两个网络，保证数据传输的可靠性。在终端系统采用有效的调度策略控制数据传输的延时上限，尽量避免在链路上发生共享冲突，保证数据传输的确定性。

综合上述分析，AFDX 网络上的数据传输体现很高的服务质量，满足新一代航空电子系统综合化发展需求。

### 1.2.2 常用调度算法

AFDX 终端系统作为连接航电子系统和互联网络的接口是 AFDX 网络数据传输中的重要环节，终端系统的调度算法会直接影响数据传输的实时性与确定性，目前的调度算法有先到先服务（First Come First Service, FCFS）、静态优先级（Static Priority, SP）和轮询（Round Robin, RR）等算法。

FCFS 是一种基本的调度算法，也是目前最常用的一种调度策略，不考虑数据的优先级，采用“先到先服务”原则对虚拟链路进行调度<sup>[16]</sup>。调度过程简单易行，数据队列按照先后顺序依次调度不需进行重新分组，调度产生的最大延时由最长队列的长度决定。但是不能保证对服务质量要求高的队列提供更好的服务，不利于紧急数据的实时传输。

RR 调度时对数据帧进行分组，按照一定顺序对各个分组进行轮询，每个数据分组都可以得到相同的机会被调度，体现了分组的公平性，但是每个分组的长度不同，导致服务时间不同，等待时间不可预测，有很大的不确定性。为了提高 RR 算法的实时性出现改进的 RR 算法，如 WRR、DRR 等。WRR 为加权轮询算法，调度过程中为每个数据分组各设置一个权值和计数器，分组每被调度一次计数器减 1，依次轮询计数器非零的分组。差额轮询 DRR 算法为队列分组设置服务额度，表示分组可被调度的限度，当有长度较大的分组超过服务额度时不能被服务，服务额度供下一轮使用<sup>[17]</sup>。

SP 调度策略下先将不同类型的数据按照紧急程度划分不同的优先级，高优先级的数据优先被调度，高优先级队列为空时查询下一优先级队列进行服务，有效地提高了数据传输时间的确定性，降低调度引入的延时<sup>[18]</sup>。SP 调度策略有

利于紧急数据的传输，但是逻辑设计比较复杂。

当网络中多任务传输竞争链路时，采用以上调度策略可以避免链路竞争冲突，确保消息传输的延时上限，但是有某个任务传输引起其他任务等待时都会给网络中的数据传输带来不确定性。

### 1.2.3 时间触发通信协议发展现状

由 TTTech 公司首先提出的时间触发通信协议（Time-Triggered Protocol, TTP）是一种高速、低成本、多端点传输的现场总线通信协议，确定性的通信机制为未来总线的发展提供了很好的思路，是当前所有基于时间触发的确定性网络通信技术中首个被 SAE 标准化的通信协议（SAE AS6003），可管理的模块化设计、严格确定性的通信机制和安全关键的容错系统使其在航空运输业、航天、铁路和汽车工业上都有广泛的应用。意大利 Aermacchi 公司 M-346 和洛克希德·马丁公司 F-16 飞机的全权限发动机数字控制系统已经成功应用 TTP<sup>[19]</sup>。TTP 为现代航空电子和航天控制系统开发的时间触发结构（Time-Triggered Architecture, TTA）在通信过程中系统的每个节点都在规定的时间内完成通信任务<sup>[20]</sup>，在物理上是分布的和分立的，但其综合程度可与综合模块化航空电子系统相比，用于全电飞机智能控制系统的开发。

基于 TTA，在 IEEE802.3 普通以太网标准之上实现的时间触发以太网（Time-Triggered Ethernet, TTE）引入了时间触发通信机制，实施时分多路访问（TDMA）方案实现无冲突、低延时的数据传输<sup>[21]</sup>。同时，TTE 网络的端系统和交换机与传统以太网的相兼容，降低系统开发的成本。TTE 的规范 SAE AS6802 已经发布，支持航空航天等广泛的行业应用。美国宇航局把 TTE 网络应用到猎户座载人飞行器上作为其主干网络，美国西科斯基飞机公司也选择 TTE 作为新一代模块化飞行管理系统的主干网络<sup>[22]</sup>，目前国内还处于开发推广阶段。另外一种基于 TTA 的协议 TTCAN（Time-Triggered Controller Area Network）是 CAN 总线的扩展，消息传输过程可预测，可扩展性和可靠性方面有所改善，主要应用在汽车通信系统中<sup>[23]</sup>。

## 1.3 本课题的主要研究内容

本课题基于对国内外机载数据总线的发展情况分析以及目前常用的调度算法特点比较，对 AFDX 总线的优越性能与终端系统调度策略方面存在的问题深入研究，根据 ARINC664 的 Part7 中对 AFDX 终端系统数据链路层的规定和时间触发通信机制，研究并设计了终端系统虚拟链路层的时间触发调度策略，从

而完成时间触发 AFDX 终端系统的构建，保证 AFDX 网络数据传输的确定性。

本论文的研究内容主要包括以下几个方面：

（1）探索时间触发通信协议，研究时间触发架构下的通信过程，以消除数据交换过程中产生的共享冲突提高通信系统的确定性为目标，借鉴时间触发通信机制，结合 AFDX 协议的规定，确定终端系统数据链路层的调度算法设计并详细规划数据帧调度的过程。

（2）建立 AFDX 网络模型分析时间触发调度策略下网络的确定性和实时性，与目前常用到的实时性能较好的静态优先级调度策略进行对比，通过理论计算和仿真实验验证时间触发调度算法在保证紧急消息传输实时性与通信网络确定性方面的优越性能。

（3）分析 AFDX 协议的主要内容，对终端系统虚拟链路层的功能需求进行详细设计，规划各部分功能的实现框图和设计流程，实现时间触发 AFDX 终端系统接收部分的冗余管理、完整性检查机制以及发送部分的流量整形、时间触发调度策略，确保数据传输的实时性和确定性。

（4）验证时间触发 AFDX 终端系统的通信性能，构建终端系统通信测试平台，对 AFDX 协议中提出的可能出现的特殊数据帧情况进行测试，验证终端系统对接收到数据帧的处理功能。

## 第 2 章 时间触发调度算法及其确定性

### 2.1 时间触发调度算法

AFDX 网络通信系统中终端系统是连接航空电子子系统与 AFDX 互连网络的接口，承担数据安全可靠交换的任务。作为 AFDX 网络数据传输的重要环节和关键部分，终端系统的性能直接影响整个 AFDX 网络的通信性能。AFDX 网络通信中多任务传输时会产生竞争链路的共享冲突，终端系统采用调度策略为不同任务分配合理的带宽和收发处理顺序解决冲突<sup>[24]</sup>，但仍会引起不确定因素影响数据传输的实时性和通信系统的确定性。时间触发通信机制满足硬实时性和确定性要求，数据交换过程可预测，具有完全确定性，AFDX 网络中引入时间触发机制有利于提高通信系统的确定性。

#### 2.1.1 时间触发 AFDX 终端系统协议栈

为满足综合式航空电子系统的数据传输技术要求，航空标准化机构于 2005 年制定 AFDX 总线的协议规范 ARINC664，协议中吸取了工业以太网的部分内容<sup>[25]</sup>，增加可靠性和确定性机制，定义一个用于飞行器的以太网数据传输网络。协议中 AFDX 终端系统引入的虚拟链路（Virtual Link，VL）是网络中的数据包从一个源端系统出发到达一个或多个预定目的终端系统的逻辑传输路径，与 ARINC429 总线的一点到多点传输相类似，但是多条虚拟链路共用一条物理链路<sup>[26]</sup>，避免使用过多的电缆和连接器增加机载负重。

ARINC664 共分为 8 部分，AFDX 主要与第 2 部分、第 3 部分和第 7 部分相关。ARINC 664 的第 2 部分规定了物理层和数据链路层，第 3 部分规定了基于 Internet 的协议与服务，第 7 部分在以太网协议 IEEE802.3 基础上规定了 AFDX 网络的确定性机制。AFDX 的终端系统按照 OSI 分层模型共有 7 层，其中网络层、传输层和应用层与普通以太网相符。AFDX 提供通信端口和服务接入点端口（SAP），通信端口分为采样端口和队列端口<sup>[27]</sup>。数据链路层中完成普通以太网的 MAC 层功能以及协议中规定的 AFDX 网络发送、接收数据的特殊处理。引入时间触发通信机制，改进虚拟链路的调度策略，在数据链路层中增加虚拟链路的时间触发调度，改进后的协议栈如图 2-1 所示，各个分层完成不同的功能，数据链路层修改不影响其他层的实现。

AFDX 终端系统的数据链路层实现发送数据帧的流量整形时间触发调度策



略、以及冗余管理，对接收数据帧进行完整性检查和冗余处理，保证通信网络中数据传输的实时性与确定性<sup>[28]</sup>。

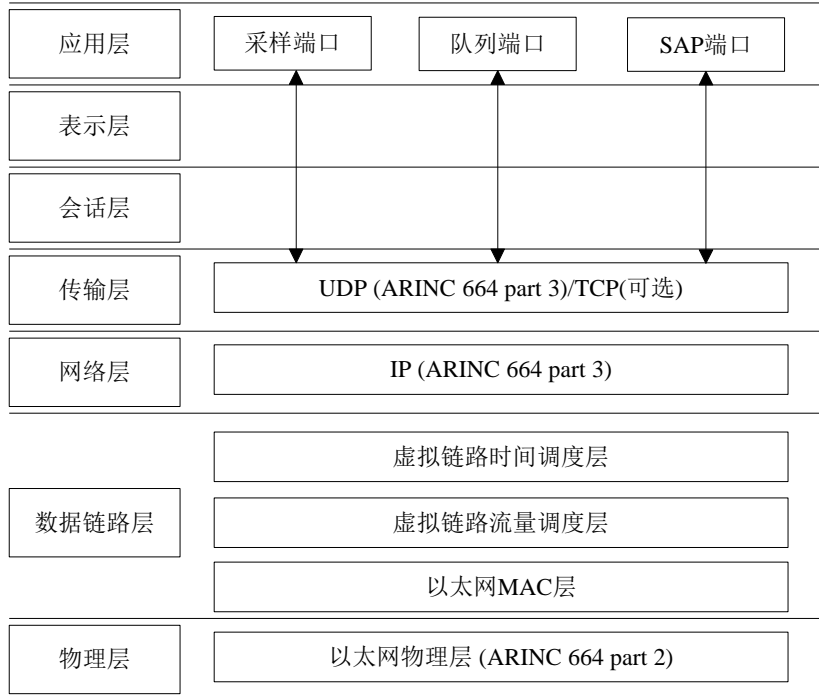


图 2-1 时间触发 AFDX 终端系统协议栈

## 2.1.2 时间触发调度算法设计

AFDX 终端系统的数据链路层引入虚拟链路保证数据传输的确定性，虚拟链路实质上是对物理链路的带宽分时复用，是 AFDX 网络通信的基础。每条 VL 都要指定两个重要参数：带宽分配间隔（Bandwidth Allocation Gap，BAG）和最大帧长度（ $L_{\max}$ ）。BAG 指一条 VL 上相邻两个数据帧的最小发送时间区间，取值为  $2^k$ ， $k$  取 0 到 7 的整数。BAG 的选择要满足 AFDX 的通信端口完成虚拟链路层的传输要求，调整网络中的数据流。 $L_{\max}$  为 VL 上可以传输的数据帧的最大帧长度<sup>[29]</sup>。VL 的最大可用带宽由公式（2-1）计算，单位为 bps。

$$C_{\max} = \frac{L_{\max} \times 8}{BAG \times 10^{-3}} \quad (2-1)$$

航电系统中根据紧急程度对数据分类，不同数据对时延的要求也不相同。警报、传感器等数据为有硬实时性要求的紧急数据，文件传输等数据为非紧急数据，但对延时有一定的要求<sup>[30]</sup>。时间触发 AFDX 终端系统中将承载有高时延要求的紧急数据的虚拟链路设计为时间触发虚拟链路（Time-Triggered

Virtual Link, TTVL), 其它传输有一定时延要求的非紧急数据的虚拟链路为常规 AFDX 网络中的虚拟链路, 是由事件触发的速率约束虚拟链路 (Rate-Constrained Virtual Link, RCVL)。数据传输过程中利用 MAC 目的地址当中的保留位设置虚拟链路的类型<sup>[31]</sup>, 速率约束虚拟链路设置保留位为 0, 时间触发虚拟链路的数据帧格式设置如图 2-2 所示。

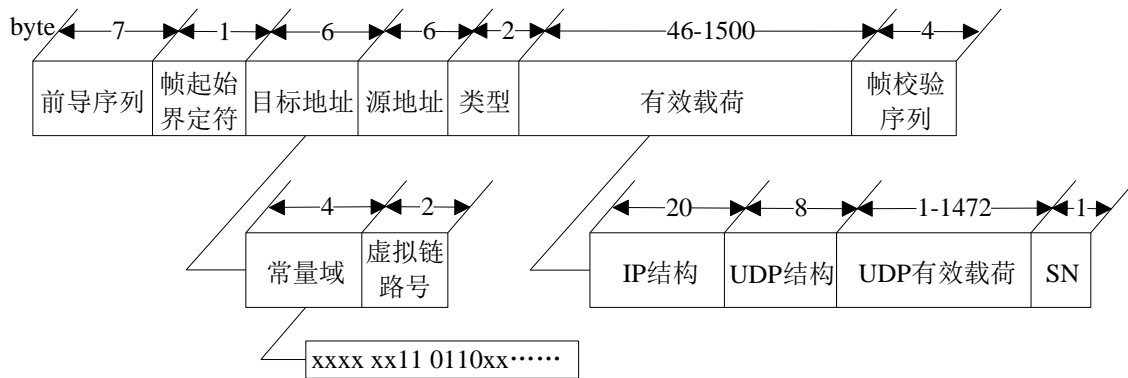


图 2-2 数据帧格式

当终端系统检测到数据帧的类型为 0110, 则数据帧所在的虚拟链路为时间触发虚拟链路, 采用时间触发调度策略, 否则为速率约束虚拟链路, 进行常规调度<sup>[32]</sup>。不需要改变 AFDX 终端系统的硬件结构就可以实现时间触发 AFDX 终端系统, 实现时间触发 AFDX 终端系统与普通 AFDX 终端系统的兼容。

为保证每条虚拟链路的带宽需求, 控制同一条虚拟链路上数据的突发传输, 使每条虚拟链路上的数据帧均匀出现, AFDX 终端系统的虚拟链路层在对虚拟链路调度前对数据源产生的不规律数据帧进行流量整形, 整形后的相邻两个数据帧按照所在虚拟链路的 BAG 时间均匀出现<sup>[33]</sup>, 如图 2-3 所示。

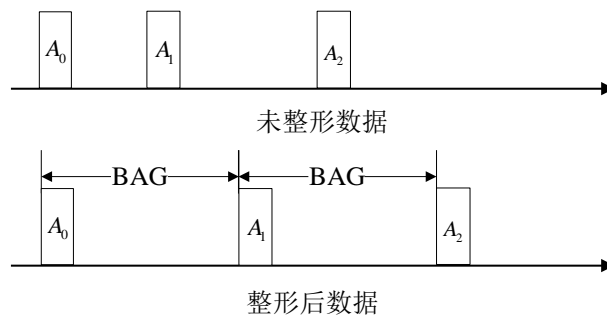


图 2-3 流量整形

AFDX 终端系统的调度策略引入时间触发通信机制, 时间触发通信协议实施时分多路访问 (TDMA) 方案, 避免数据在总线上传输产生冲突, 严格保证确定性通信。时间触发构架下数据传输过程如图 2-4 所示。数据的传输分配在相同长度的 TDMA 周期中进行, 若干 TDMA 周期组成一个循环序列, 在一个

TDMA 周期中具有若干个时隙，一个节点最多占用一个时隙，每个节点在它的时间片内带宽利用率为 100%。按计划的时隙和周期发送数据帧，每个 TDMA 周期中的一切行为都是唯一确定的，具有可预测性和完全确定性<sup>[34]</sup>。

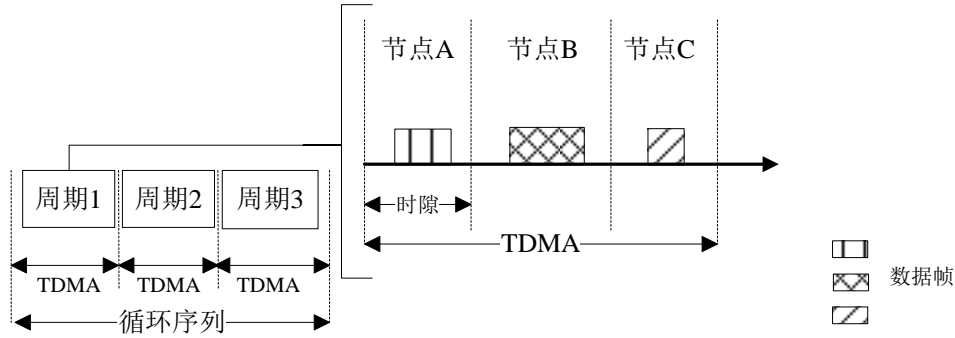


图 2-4 TTP 通信周期

时间触发 AFDX 终端系统的 TTVL 采用时间触发调度策略，调度过程与时间触发通信机制的过程相同，数据帧的传输都按照预先规划好的时刻进行。时间触发虚拟链路具有高优先级，根据调度时刻表对流量整形后的 TTVL 优先进行调度。借鉴时间触发以太网中调度时刻表的规划方法，根据协议中规定的 BAG 取值范围为 1-128ms，设定调度时刻表的矩阵周期（Matrix Cycle，MC）为所有 BAG 取值的最小公倍数即 128ms，表示为时间触发虚拟链路数据帧调度的任务周期，基本周期（Basic Cycle，BC）为所有 BAG 取值的最大公约数<sup>[35]</sup>，是时间触发虚拟链路数据帧传输的最小周期取值为 1ms。AFDX 网络数据传输速率为 100Mb/s，数据帧的最大长度为 1518 个字节，发送时间需要  $(1518+20) \times 8 / 100 = 123.04\mu\text{s}$ ，最小数据帧长度为 64 个字节，发送时间需要  $(64+20) \times 8 / 100 = 6.72\mu\text{s}$ ，则在基本周期 1ms 内可以最少可以发送 8 个数据帧，最多可发送 148 个数据帧。

图 2-5 所示的调度时刻表选取了 8 条虚拟链路，对应了 BAG 的所有取值情况。图中  $V_{m,n}$  表示第  $m$  条虚拟链路的第  $n$  个数据帧，在调度时刻表的位置是第  $[(n-1) \times 2^m + 1]$  行第  $(m+1)$  列，第  $m$  条虚拟链路的 BAG 取值为  $2^m \text{ms}$ 。计算可知若每条时间触发虚拟链路上的数据帧都达到最大长度，在第一个基本周期内发送完 8 个数据帧后还有  $15.68\mu\text{s}$ ，若数据帧长度均为最小值，则第一个基本周期的空余时间为  $946.24\mu\text{s}$ ，图 2-5 中的空余时间段范围为  $15.68\text{-}946.24\mu\text{s}$ ，其余的基本周期内不存在每条虚拟链路都被调度的情况，空余时间更长，保证每个基本周期之间的调度过程互不干扰。

基本周期0	$V_{0,1}$	$V_{1,1}$	$V_{2,1}$	...	$V_{7,1}$	空余时间	1ms	矩阵周期 128ms
基本周期1	$V_{0,2}$						1ms	
基本周期2	$V_{0,3}$	$V_{1,2}$					1ms	
基本周期3	$V_{0,4}$						1ms	
基本周期4	$V_{0,5}$	$V_{1,3}$	$V_{2,2}$				1ms	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
基本周期127	$V_{0,128}$						1ms	

图 2-5 调度时刻表

对 TTVL 时间触发调度的具体过程如下：

(1) 对需要被调度的 8 条时间触发虚拟链路分配优先级，BAG 值最小的虚拟链路优先级最高，图 2-5 中的第 0 条 TTVL 优先级最高，第 7 条 TTVL 优先级最低。若有相同的 BAG，则数据帧最大帧长度较大的 TTVL 优先级高<sup>[36]</sup>，调度过程中可以减小各个虚拟链路数据帧传输的延时差，达到均衡网络负载的目的。

(2) 准备对 TTVL 进行时间触发调度，初始化状态  $m=0$ ， $n=1$ 。

(3) 根据离线设计的时刻调度表，查找  $V_{m,n}$  被调度的时刻，严格按照时刻调度表发送数据帧。在第一个基本周期内按照优先级的高低每一条虚拟链路都先后被调度一次，即每条虚拟链路上的第一个数据帧在规定时刻被发送。由虚拟链路的 BAG 值可知在一个矩阵周期内 TTVL 被调度的次数为  $\frac{128}{2^m}$ ，一个矩阵周期中  $n$  的最大取值为  $\frac{128}{2^m}$ ，在调度时刻表中查找到每条 TTVL 的  $\frac{128}{2^m}$  个调度时刻， $V_{m,n}$  被调度的时刻  $t_{m,n}$  按公式 (2-2) 计算，单位为 ms，其中  $L_{p,1}$  为数据帧  $V_{p,1}$  的长度， $C$  为物理链路带宽 100Mb/s。

$$t_{m,n} = (n-1) \times 2^m + \frac{8 \times \sum_{p=0}^{m-1} L_{p,1} \times 1000}{C \times 10^6} \quad (2-2)$$

完成数据帧  $V_{m,n}$  的发送后， $m+1$ ，循环 (3) 的过程，直至  $m=7$  所有 TTVL 中的数据帧都按照调度时刻表发送，时间触发调度完成。

终端系统进行时间触发调度的同时交换机配置相应的调度时刻表，目的终端系统的接收部分功能与普通 AFDX 终端系统的相同，对接收到的数据帧进行完整性检查和冗余处理，完成时间触发虚拟链路数据帧的传输过程。整个过程

具有可预测性，解决多条虚拟链路需要被调度在数据链路层上产生冲突的问题。

若一个终端系统除时间触发虚拟链路外还有速率约束虚拟链路，两种虚拟链路的调度过程如图 2-6 所示。优先调度承载时间关键消息的时间触发虚拟链路，按照调度时刻表发送数据帧，标记出每个基本周期内的空余时间段，在标记出的时间段内采用常用的调度策略，本文中采用静态优先级调度策略（SP）调度速率约束虚拟链路，先将 RCVL 按照紧急程度划分不同的优先级，配置相应的 VL 调度索引表，根据调度索引表决定速率约束虚拟链路的调度顺序，有效地提高了数据传输时间的确定性，降低调度引入的延时。

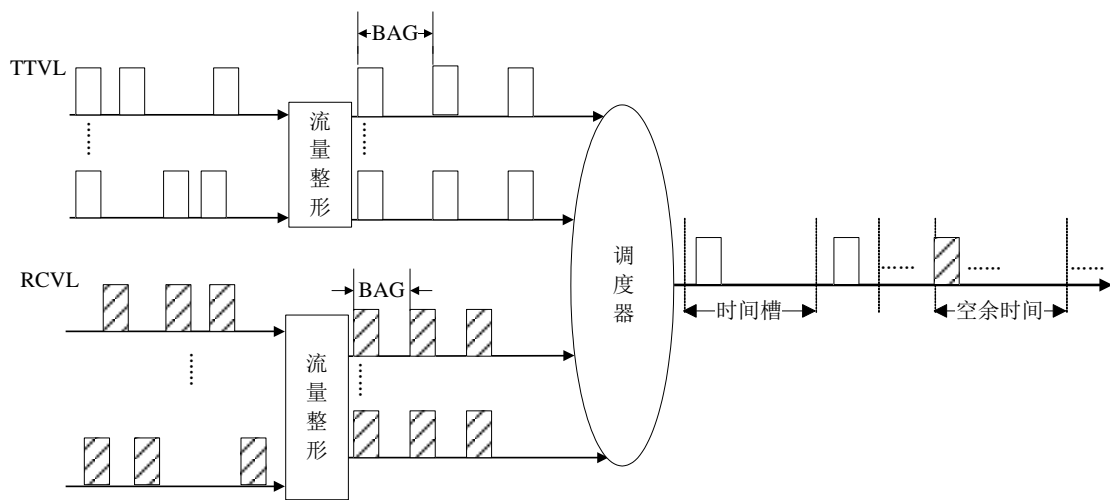


图 2-6 虚拟链路调度过程示意图

## 2.2 时间触发 AFDX 网络建模及其确定性分析

时间触发 AFDX 终端系统与普通 AFDX 终端系统的硬件结构和组网的拓扑结构都是相同的，分析时间触发 AFDX 网络的性能时与分析普通 AFDX 网络的过程相同，在终端系统调度虚拟链路部分的过程有所不同。建立与普通 AFDX 网络拓扑结构相同的网络模型，通过端到端延时这一性能指标分析网络确定性和实时性。

### 2.2.1 时间触发 AFDX 网络确定性分析

航电系统对数据传输的实时性和确定性要求很高，消息能否按时到达目的端系统关系到航电系统能否正常工作，网络的确定性关系到消息的可靠传输。AFDX 网络的端到端延时是分析 AFDX 网络确定性及时性的一个重要指标，表示为消息从源端系统出发到达目的端系统的时间，主要包括端系统的延时、链路上的传输延时和交换机的转发延时，累计计算结果为整个端到端延时<sup>[37]</sup>，

图 2-7 所示为端到端延时的各部分组成。

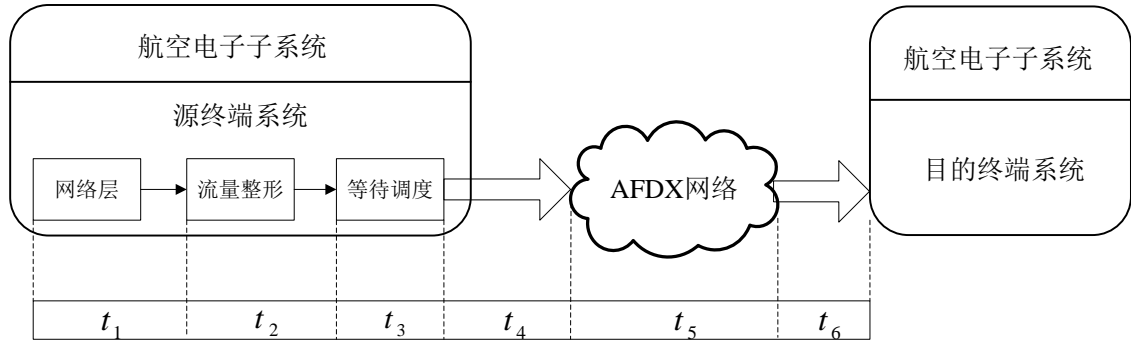


图 2-7 端到端延时组成

图 2-7 中  $t_1$  表示为消息经过 UDP、IP 模块封装成数据帧所用的时间，由端系统的技术性能决定，通常情况下认为处理速度足够快，该部分延时可以忽略不计。 $t_2$  为等待流量整形的时间，有需要流量整形的数据帧堆积时产生，由于时间触发调度策略严格按照调度时刻对数据帧转发，不会在调度器中出现积压的数据帧。 $t_3$  是数据帧在队列中等待调度的时间，主要取决于发送的帧个数和调度策略，影响网络的实时性能。本文中采用的时间触发调度策略离线设计调度时刻表，使每条时间触发虚拟链路的数据帧都严格按照调度时刻表安排的时间发送。 $t_3$  等于该数据帧在调度时刻表中规划的发送时刻与流量整形后时刻之差。 $t_4$  与  $t_6$  表示为数据帧在物理链路上的传输时延，可以认为该时间为一固定值。 $t_5$  是指数据帧在所经过的交换机转发延时，由于对交换机也配置相应的调度时刻表，数据帧在各个交换机的收发时间都是确定的。 $t_5$  可以由数据帧所经过的交换机调度时刻表中的时刻之差计算。

由上述分析可以看出时间触发 AFDX 网络中的端到端延时是一个固定值，可以预测，数据帧的整个传输过程具有完全确定性。

### 2.2.2 时间触发 AFDX 网络建模及其延时计算

时间触发 AFDX 网络结构与普通 AFDX 网络结构相同，本文建立的时间触发 AFDX 网络模型采用 AFDX 网络典型拓扑结构，网络模型如图 2-8 所示，模型中包括三个交换机 S1-S3，七个终端系统 e1-e7 和十条虚拟链路 V1-V10，配置各个虚拟链路参数如表 2-1，各个虚拟链路均为单播虚拟链路，计算时链路上的传输延时取典型值  $16 \mu\text{s}$ 。

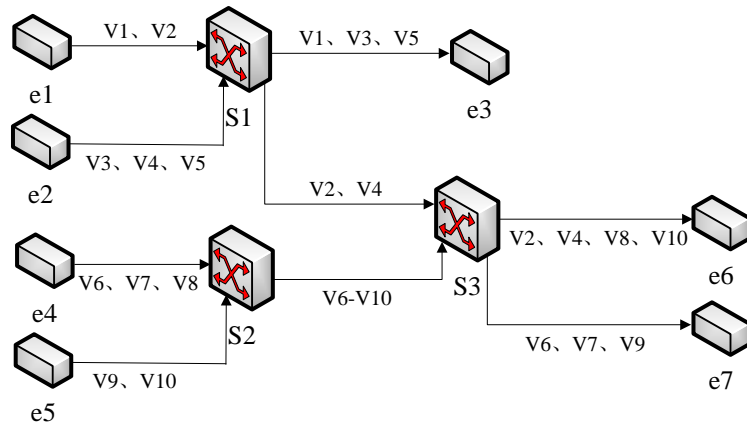


图 2-8 时间触发 AFDX 网络模型

表 2-1 虚拟链路配置表

VL	$L_{\max}$ (Bytes)	BAG(ms)
V1	512	16
V2	256	32
V3	1024	32
V4	1024	64
V5	1024	128
V6	512	32
V7	512	64
V8	128	32
V9	1024	128
V10	1024	32

图 2-8 中的虚拟链路均为时间触发虚拟链路，采用时间触发调度策略，对 V1、V2、V3、V4、V5 设计的调度时刻表如图 2-9 所示。

基本周期0	V <sub>1,1</sub>	V <sub>3,1</sub>	V <sub>2,1</sub>	V <sub>4,1</sub>	V <sub>5,1</sub>		1ms
基本周期16	V <sub>1,2</sub>						1ms
基本周期32	V <sub>1,3</sub>	V <sub>3,2</sub>	V <sub>2,2</sub>				1ms
基本周期48	V <sub>1,4</sub>						1ms
基本周期64	V <sub>1,5</sub>	V <sub>3,3</sub>	V <sub>2,3</sub>	V <sub>4,2</sub>			1ms
基本周期80	V <sub>1,6</sub>						1ms
基本周期96	V <sub>1,7</sub>	V <sub>3,4</sub>	V <sub>2,4</sub>				1ms
基本周期112	V <sub>1,8</sub>						1ms
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
基本周期127							1ms

矩阵周期  
128ms

图 2-9 V1-V5 的调度时刻表

每条时间触发虚拟链路按照各自的 BAG 值流量整形后划分不同的优先级，在交换机 S1 端口 V1 优先级最高，V3 次之，V2、V4、V5 的优先级依次递减。在一个矩阵周期 128ms 内 V1 被调度 8 次， $V_{1,1}$  在 0 时刻被调度， $V_{1,2}$ — $V_{1,8}$  分别间隔 16ms 被调度，V3 和 V2 被调度 4 次，V4 被调度 2 次，V5 被调度 1 次，根据公式 (2-2) 计算出  $V_{3,1}$ 、 $V_{2,1}$ 、 $V_{4,1}$ 、 $V_{5,1}$  的被调度时刻分别为：

$$t_{3,1} = \frac{8 \times 512 \times 1000}{100 \times 10^6} = 0.04096\text{ms}$$

$$t_{2,1} = \frac{8 \times (512 + 1024) \times 1000}{100 \times 10^6} = 0.12288\text{ms}$$

$$t_{4,1} = \frac{8 \times (512 + 256 + 1024) \times 1000}{100 \times 10^6} = 0.14336\text{ms}$$

$$t_{5,1} = \frac{8 \times (512 + 256 + 1024 + 1024) \times 1000}{100 \times 10^6} = 0.22528\text{ms}$$

其它数据帧被调度的时刻计算结果如表 2-2 所示。

表 2-2 调度时刻计算结果 (ms)

调度时刻	计算结果	调度时刻	计算结果
$t_{1,2}$	16	$t_{3,2}$	32.04096
$t_{1,3}$	32	$t_{3,3}$	64.04096
$t_{1,4}$	48	$t_{3,4}$	96.04096
$t_{1,5}$	64	$t_{2,2}$	32.1228
$t_{1,6}$	80	$t_{2,3}$	64.1228
$t_{1,7}$	96	$t_{2,4}$	96.1228
$t_{1,8}$	112	$t_{4,2}$	64.14336

$V_{1,1}$  在 0 时刻被调度，经过交换机 S1 到达目的端系统 e3，端到端延时包括链路的传输延时和数据帧发送时间，端到端延时  $D_{1,1}$  计算为：

$$D_{1,1} = 2 \times 16 + \frac{8 \times 512}{100} = 72.96\mu\text{s}$$

由于采用时间触发调度各个数据帧被调度的时刻已经固定，对于后面每个被调度数据帧都不存在有积压的数据帧传输过程未完成的情况，经过分析计算可知  $V_{1,2}$ — $V_{1,8}$  的端到端延时均为 72.96  $\mu\text{s}$ 。V3、V5 同样只经过一个交换机到达目的端系统 e3，V3 和 V5 的端到端延时  $D_3$ 、 $D_5$  分别计算为：



$$D_3 = 2 \times 16 + 40.96 + \frac{8 \times 1024}{100} = 154.88 \mu s$$

$$D_5 = 2 \times 16 + 225.28 + \frac{8 \times 1024}{100} = 339.2 \mu s$$

V6—V10 在交换机 S2 的发送调度时刻表如图 2-10 所示。

基本周期0	V <sub>10,1</sub>	V <sub>6,1</sub>	V <sub>8,1</sub>	V <sub>7,1</sub>	V <sub>9,1</sub>		1ms
基本周期32	V <sub>10,2</sub>	V <sub>6,2</sub>	V <sub>8,2</sub>				1ms
基本周期64	V <sub>10,3</sub>	V <sub>6,3</sub>	V <sub>8,3</sub>	V <sub>7,2</sub>			1ms
基本周期96	V <sub>10,4</sub>	V <sub>6,4</sub>	V <sub>8,4</sub>				1ms
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
基本周期127							1ms

矩阵周期  
128ms

图 2-10 V6-V10 的调度时刻表

优先级由高到低的顺序依次是 V10、V6、V8、V7、V9，在一个矩阵周期内被调度的次数分别为 4、2、4、1、4。各个数据帧的调度时刻计算结果如表 2-3 所示。

表 2-3 V6-V10 调度时刻计算结果 (ms)

调度时刻	计算结果	调度时刻	计算结果	调度时刻	计算结果
$t_{10,1}$	0	$t_{6,2}$	32.08192	$t_{8,3}$	64.12288
$t_{10,2}$	32	$t_{6,3}$	64.08192	$t_{8,4}$	96.12288
$t_{10,3}$	64	$t_{6,4}$	96.08192	$t_{7,1}$	0.13312
$t_{10,4}$	96	$t_{8,1}$	0.12288	$t_{7,2}$	64.13312
$t_{6,1}$	0.08192	$t_{8,2}$	32.12288	$t_{9,1}$	0.17408

V2、V4、V6—V10 在交换机 S3 的调度时刻表如图 2-11，各虚拟链路数据帧的调度时刻结算结果如表 2-4 所示。

基本周期0	V <sub>10,1</sub>	V <sub>6,1</sub>	V <sub>2,1</sub>	V <sub>8,1</sub>	V <sub>4,1</sub>	V <sub>7,1</sub>	V <sub>9,1</sub>		1 ms
基本周期32	V <sub>10,2</sub>	V <sub>6,2</sub>	V <sub>2,2</sub>	V <sub>8,2</sub>					1 ms
基本周期64	V <sub>10,3</sub>	V <sub>6,3</sub>	V <sub>2,3</sub>	V <sub>8,3</sub>	V <sub>4,2</sub>	V <sub>7,2</sub>			1 ms
基本周期96	V <sub>10,4</sub>	V <sub>6,4</sub>	V <sub>2,4</sub>	V <sub>8,4</sub>					1 ms
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
基本周期127									1 ms

矩阵周期  
128ms

图 2-11 V2、V4、V6—V10 的调度时刻表

表 2-4 V2、V4、V6—V10 调度时刻计算结果 (ms)

调度时刻	计算结果	调度时刻	计算结果	调度时刻	计算结果
$t_{10,1}$	0	$t_{6,4}$	96.08192	$t_{8,3}$	64.14336
$t_{10,2}$	32	$t_{2,1}$	0.12288	$t_{8,4}$	96.14336
$t_{10,3}$	64	$t_{2,2}$	32.12288	$t_{4,1}$	0.1536
$t_{10,4}$	96	$t_{2,3}$	64.12288	$t_{4,2}$	64.1536
$t_{6,1}$	0.08192	$t_{2,4}$	96.12288	$t_{7,1}$	0.23552
$t_{6,2}$	32.08192	$t_{8,1}$	0.14336	$t_{7,2}$	64.23552
$t_{6,3}$	64.08192	$t_{8,2}$	32.14336	$t_{9,1}$	0.27648

V10 和 V6 由源端系统出发，经过交换机 S2、S3 到达目的端系统，V10 和 V6 经过交换机 S2 到达交换机 S3 端口时的端到端延时  $D'_{10}$  和  $D'_6$  分别为：

$$D'_{10} = 2 \times 16 + \frac{8 \times 1024}{100} = 113.92 \mu s$$

$$D'_6 = 2 \times 16 + 81.92 + \frac{8 \times 512}{100} = 154.88 \mu s$$

根据两个调度时刻表的规划计算 V10 和 V6 的端到端延时  $D_{10}$ 、 $D_6$  分别为：

$$D_{10} = 3 \times 16 + \frac{8 \times 1024}{100} \times 2 = 211.84 \mu s$$

$$D_6 = 3 \times 16 + 81.92 + 81.92 + \frac{8 \times 512}{100} \times 2 = 293.76 \mu s$$

V2 和 V4 由源端系统出发经过交换机 S1 到达交换机 S3 端口，该过程内 V2 和 V4 的传输延时  $D'_2$  和  $D'_4$  分别计算为：

$$D'_2 = 2 \times 16 + 122.88 + \frac{8 \times 256}{100} = 175.36 \mu s$$

$$D'_4 = 2 \times 16 + 143.36 + \frac{8 \times 1024}{100} = 257.28 \mu s$$

由于  $D'_2 > D'_{10}$ ，V10 先到达交换机 S3 端口，按照时刻调度表的规划，V2 到达时调度时刻表已经开始，否则需要等待 V10 到达开始调度时刻表。

V2 由 e1 到达 e6 的端到端延时  $D_2$  为：

$$D_2 = D'_2 + 16 + 122.88 + \frac{8 \times 256}{100} = 334.72 \mu s$$

V8 由源端系统 e4 到达目的端系统 e6 的端到端延时  $D_8$  计算为：

$$D_8 = 3 \times 16 + 122.88 + 143.36 + \frac{8 \times 128}{100} \times 2 = 334.72 \mu s$$

V4 的端到端延时  $D_4$  为:

$$D_4 = D_4' + 16 + 153.6 + \frac{8 \times 1024}{100} = 385.92 \mu s$$

V7 和 V9 的端到端延时  $D_7$ 、 $D_9$  分别为:

$$D_7 = 3 \times 16 + 133.12 + 235.52 + \frac{8 \times 512}{100} \times 2 = 498.56 \mu s$$

$$D_9 = 3 \times 16 + 174.08 + 276.48 + \frac{8 \times 1024}{100} \times 2 = 662.4 \mu s$$

统计各虚拟链路的端到端延时计算结果如表 2-5 所示。

表 2-5 端到端延时计算结果 ( $\mu s$ )

虚拟链路号	延时	虚拟链路号	延时
V1	72.96	V6	293.76
V2	334.72	V7	498.56
V3	154.88	V8	334.72
V4	385.92	V9	662.4
V5	339.2	V10	211.84

## 2.3 静态优先级调度算法及其确定性分析

目前常用到的调度算法当中由陈昕提出的静态优先级队列调度算法为解决紧急数据的高效可靠传输提供了一种很好的解决方案,但是当终端系统有多条虚拟链路同时需要被调度时,会导致数据链路层上的冲突,某一条虚拟链路的调度使其它的虚拟链路等待而引入抖动(Jitter),抖动值的计算为一个可变值。可变延时的计算利用网络演算法,最后计算出网络的端到端延时结果为出现最坏情况的上限值<sup>[38]</sup>。

### 2.3.1 静态优先级调度算法

航空电子系统不断高度综合化发展,导航系统、飞行控制系统、警报系统等不断升级,通信网络中的传输任务紧急程度各不相同,按照其紧急程度对不同类型的数据赋予由高到低三个不同的优先级,每个优先级包含若干紧急程度相同的虚拟链路队列。

静态优先级队列调度过程如图 2-12 所示,对流量整形后的各个虚拟链路根据消息的紧急程度分配不同的优先级,优先级相同的虚拟链路进入相同的优先

级队列，如图中所示 V1、V2 为最高优先级，进入相同队列，V3、V4 和 V5、V6 分别属于第二优先级和低优先级，由每条虚拟链路的优先级标识与虚拟链路号映射对应关系形成优先级配置表。调度过程中查询优先级配置表按照优先级的高低与虚拟链路号的关系决定各个虚拟链路的调度顺序，同一优先级队列中的虚拟链路根据先到先服务的原则进行数据帧的转发，不同优先级队列采取非抢占式方式即后到达的高优先级数据帧不能抢占正在进行的低优先级数据帧的服务<sup>[39]</sup>。紧急数据的延时低于非紧急数据的传输延时，以确保紧急数据的实时性能要求。

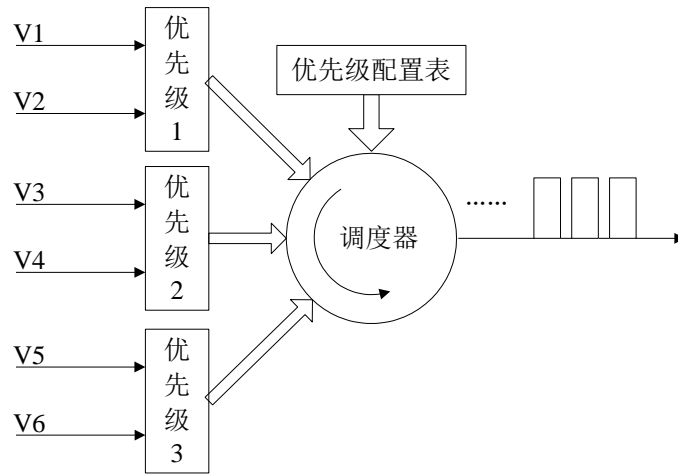


图 2-12 SP 调度过程

### 2.3.2 网络演算理论

终端系统的多条虚拟链路同时需要被调度时，常用的调度策略都会不同程度的引入抖动，ARINC 664 的 Part 7 规定抖动的上限值必须满足公式<sup>[40]</sup> (2-3)：

$$\begin{cases} \max\_jitter \leq 40\mu s + \frac{\sum_{i \in \{\text{set of VLs}\}} (20 + L_{\max j}) \times 8}{Nb w} \\ \max\_jitter \leq 500\mu s \end{cases} \quad (2-3)$$

其中  $L_{\max j}$  是第  $j$  条链路的最大帧长， $Nbw$  是物理链路的带宽，单位为 bits/s。

对单条 VL 规定的延时上限满足公式 (2-4)：

$$\max\_latency_i \leq BAG_i + \max\_jitter + latency\_in\_transmission \quad (2-4)$$

其中  $latency\_in\_transmission$  是发送的技术延时，是固定值，延时上限的确定主要取决于  $\max\_jitter$  的计算。交换机缓存和转发数据的延时是可变延时，利用网络演算法计算含有可变延时的端到端延时<sup>[41]</sup>。

网络演算法是一种基于最小加代数的分析网络延时及缓冲区调度的理论，

用于网络通信研究等领域<sup>[42]</sup>。网络演算理论分为确定性网络演算和随机网络演算，AFDX网络是确定性网络，利用确定网络演算理论计算端到端延时<sup>[43]</sup>。

网络演算法中根据数据流的到达曲线和交换机的服务曲线，计算出服务曲线和到达曲线之间的最大水平距离即为可变延时的上限值<sup>[44]</sup>。到达曲线 $\alpha(t)$ 与服务曲线 $\beta(t)$ 的计算公式分别为公式（2-5）和公式（2-6）：

$$\begin{cases} \alpha(t) = \sigma + \rho t \\ \sigma = L_{\max} \\ \rho = \frac{L_{\max}}{BAG} \end{cases} \quad (2-5)$$

$$\beta(t) = C(t - D_{tec})^+ \quad (2-6)$$

其中， $\sigma$ 为突发度， $\rho$ 为可持续流量， $L_{\max}$ 为最大帧长， $C$ 为服务速率在本文中为 100Mb/s， $(x)^+ = \max(0, x)$ ， $D_{tec}$ 是交换机技术延时，系统的可变延时上限为 $\alpha(t)$ 与 $\beta(t)$ 的最大水平距离 $d(\alpha, \beta)$ ，如图 2-13 所示。

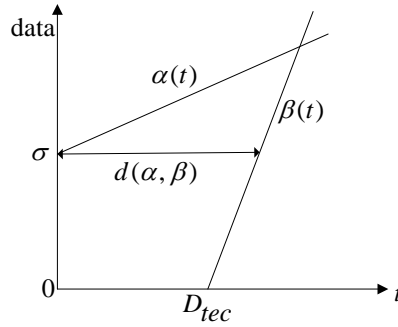


图 2-13 最大可变延时计算

计算中可以认为 $D_{tec}$ 是 0，则 $\beta(t)$ 可改写为：

$$\beta(t) = C(t)^+ \quad (2-7)$$

此时交换机转发延时 $d(\alpha, \beta)$ 计算为：

$$d = \frac{\sigma}{C} = \frac{L_{\max}}{C} \quad (2-8)$$

### 2.3.3 AFDX 网络在静态优先级调度下确定性分析

SP 调度策略中虚拟链路根据紧急程度分为三种不同的优先级，计算交换机转发延时上限的公式修改为<sup>[45]</sup>：

$$d_h = \frac{\max(\sigma_h, \sigma_m, \sigma_l) + \sum_{V_i \in V_h} \sigma_i}{C} \quad (2-9)$$

$$d_m = \frac{\max(\sigma_m, \sigma_l) + \sum_{V_i \in V_h} \sigma_i + \sum_{V_j \in V_m} \sigma_j}{C - \sum_{V_i \in V_h} \rho_i} \quad (2-10)$$

$$d_l = \frac{\sum_{V_i \in V_h} \sigma_i + \sum_{V_j \in V_m} \sigma_j + \sum_{V_k \in V_l} \sigma_k}{C - \sum_{V_i \in V_h} \rho_i - \sum_{V_j \in V_m} \rho_j} \quad (2-11)$$

公式 (2-9) 计算出最高优先级虚拟链路的可变延时, 其中  $\max(\sigma_h, \sigma_m, \sigma_l)$  表示三个不同优先级队列中的虚拟链路突发度的最大值,  $\sigma_h$  为最高优先级虚拟链路的突发度,  $\sigma_m$  和  $\sigma_l$  分别为高优先级和低优先级的突发度。公式 (2-10) 和 (2-11) 计算高优先级和低优先级虚拟链路的延时上限,  $\sum_{V_i \in V_h} \sigma_i$ 、 $\sum_{V_j \in V_m} \sigma_j$  和  $\sum_{V_k \in V_l} \sigma_k$  分别为所有最高优先级、高优先级和低优先级虚拟链路的突发度之和,  $\sum_{V_i \in V_h} \rho_i$  和  $\sum_{V_j \in V_m} \rho_j$  是所有最高优先级和高优先级虚拟链路的可持续流量之和, 也是虚拟链路的速率之和。由公式 (2-9)、(2-10) 和 (2-11) 分别计算出不同虚拟链路的可变延时上限值, 与虚拟链路经过的链路延时总和即为端到端延时的最终结果。

根据上述原理和公式对 2.2.2 中图 2-8 的网络模型计算 SP 调度策略下各个虚拟链路的端到端延时。

V1 优先级最高, 由 e1 出发经过一个交换机到达目的端系统, 可变延时部分计算为:

$$d_1 = \frac{\max(\sigma_h, \sigma_m, \sigma_l) + \sigma_h}{C} = \frac{(1024 + 512) \times 8}{100} = 122.88 \mu s$$

V3 优先级低于 V1, 高于 V2、V4 和 V5, 可变延时为:

$$d_3 = \frac{\max(\sigma_m, \sigma_l) + \sigma_1 + \sigma_3}{C - \rho_1} = \frac{(1024 + 1024 + 512) \times 8}{100 - \frac{512 \times 8}{16 \times 10^3}} = 205.326 \mu s$$

经过交换机 S1 的其它虚拟链路和经过交换机 S2 的虚拟链路转发延时计算结果如表 2-6 所示。表中结果为交换机转发延时上限值, 由调度器同时调度多条虚拟链路产生抖动造成, 计算值均不大于  $500 \mu s$ , 满足协议中规定的抖动延时上限值。

表 2-6 交换机转发延时计算结果 (  $\mu\text{s}$  )

虚拟链路号	计算结果	虚拟链路号	计算结果
V2	226.439	V7	257.069
V4	308.98	V8	215.869
V5	309.378	V9	257.235
V6	205.326	V10	163.84

V1、V3 和 V5 只经过一个交换机，计算的延时结果与链路延时之和即为最终延时。虚拟链路 V2、V4、V6—V10 经过第二个交换机时按照同样的方式计算经过 S3 的延时，将两个可变延时上限值结果与链路传输延时相加得到每条虚拟链路的最终延时结果，如表 2-7 所示。

表 2-7 端到端延时计算结果 (  $\mu\text{s}$  )

虚拟链路号	延时	虚拟链路号	延时
V1	154.88	V6	458.652
V2	500.587	V7	665.661
V3	237.326	V8	500.449
V4	675.951	V9	666.06
V5	341.378	V10	375.68

与 2.2.2 中时间触发调度策略下的端到端延时计算结果相比较，承载高延时要求的紧急消息的高优先级虚拟链路在时间触发调度策略下传输延时明显减小，通过理论计算说明时间触发虚拟链路调度算法对提高实时性能有了很大的改善。

## 2.4 本章小结

本章主要阐述了时间触发 AFDX 终端系统虚拟链路层调度算法的总体设计。首先了解时间触发通信机制，从消除 AFDX 网络虚拟链路调度的抖动目的出发，确定时间触发调度策略虚拟链路的调度过程。其次，构建 AFDX 网络模型分析并计算出时间触发调度策略下的端到端延时。最后，分析 AFDX 网络在静态优先级调度策略下的实时性能，对比理论计算结果，说明时间触发调度策略对网络通信性能的改善。

## 第 3 章 基于 TrueTime 平台的网络仿真分析

### 3.1 TrueTime 平台介绍

目前广泛应用的网络仿真软件主要有 NS2、OPNET 和 DRTSS 等，用于对网络路由协议的仿真和 TCP 层以上的模拟。对于网络延时和调度实时性能方面的研究以及数据链路层的仿真，本文选用由瑞典 Lund 大学于 2002 年推出的 MATLAB/TrueTime 工具箱，该工具箱针对特定的网络协议实现实时调度的综合仿真研究<sup>[46]</sup>。

如图 3-1 所示，TrueTime 工具箱提供了六个功能不同的模块：TrueTime Kernel、TrueTime Network、TrueTime Wireless、TrueTime Battery、ttSendMsg、ttGetMsg，仿真时可与 MATLAB 中的其他工具箱中的控制模块相结合构造网络控制系统。本文仿真实验中主要应用 TrueTime Kernel 和 TrueTime Network 模块。

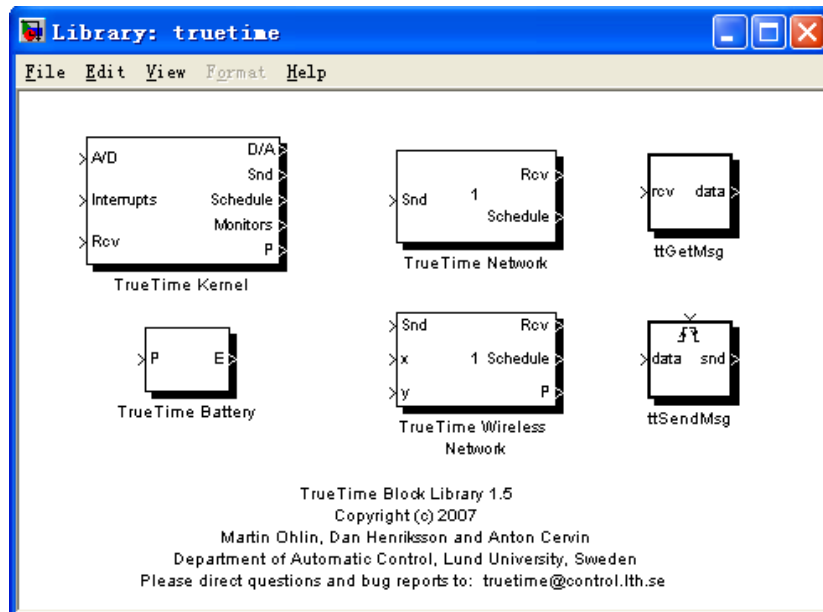


图 3-1 TrueTime 模块库

TrueTime Kernel 模块用作网络控制系统的网络节点，其内部嵌有 A/D、D/A 转换器接口，调度器与监控器的输出显示仿真过程中资源的分配情况<sup>[47]</sup>。TrueTime Kernel 模块按照用户自定义的任务工作，以中断形式产生的内部事件和外部事件决定任务是否工作，当发生事件中断时，调用用户自定义的中断句柄，执行中断服务<sup>[48]</sup>。中断柄函数代码可以采用 Matlab 或 C++ 语言编写，本文各个模块中的内嵌程序和函数采用 Matlab 语言编写。



TrueTime Network 模块提供了多种网络参数和网络模式等通信资源，共有六种网络协议可供选择，根据任务要求对网络参数进行设置，当有消息进出时网络模块工作并执行相应的功能。TrueTime Network 模块包括 Snd、Rcv 和 Schedule 接口，Snd 接口接收数据信息，Rcv 将接收到的信息发送到 TrueTime Kernel 模块的 Rcv 接口，Schedule 接口显示仿真过程中的网络资源分配情况<sup>[49]</sup>。TrueTime Network 模块中预定义了多种调度策略，根据需要可直接调用调度算法程序，也可修改调度程序后再调用。

## 3.2 TrueTime 仿真模型构建

TrueTime 平台下搭建 AFDX 网络模型，利用工具箱提供的模块和 MATLAB/Simulink 中的基本模块分别构造终端系统和交换机的仿真模型，分析 TrueTime 命令列表并解析各个命令的功能和用法，根据终端系统和交换机各自需要完成的任务，调用 TrueTime 命令利用 Matlab 语言编写各个模块中用到的程序和函数。将构建好的终端系统和交换机模型按照指定的 AFDX 网络结构连接，设置网络参数对 AFDX 网络不同调度算法进行仿真。仿真实验得到结果与理论计算结果比较，验证时间触发调度策略的优越性能。

### 3.2.1 终端系统仿真模型

AFDX 终端系统主要完成协议中规定的帧收发功能，仿真模型如图 3-2 所示，由 TrueTime Kernel 模块构成。模型中内核模块的输入输出端口与其它模块相连，示波器观察仿真过程中的资源分配情况。

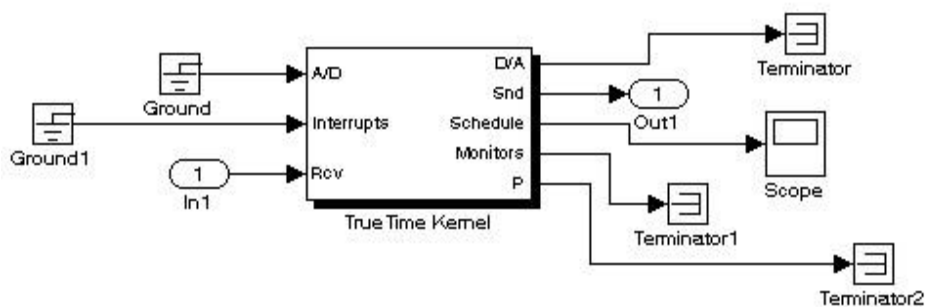


图 3-2 终端系统仿真模型

终端系统模型中的 TrueTime Kernel 模块配置如图 3-3，模块嵌入的初始化程序为 e1，时钟漂移和时钟偏移等初始化参数设置为 0，仿真过程中的时钟为标准时钟。初始化程序 e1 中设置了中断句柄函数和周期性数据流，初始化网络接口，利用周期性数据流模拟虚拟链路上的数据帧，中断柄函数中设置数据帧

的长度、优先级、发送端口等参数，设置输入与输出端口数目和调用相应的调度策略。初始化程序与内嵌的中断柄函数共同完成 AFDX 终端系统收发数据的功能。

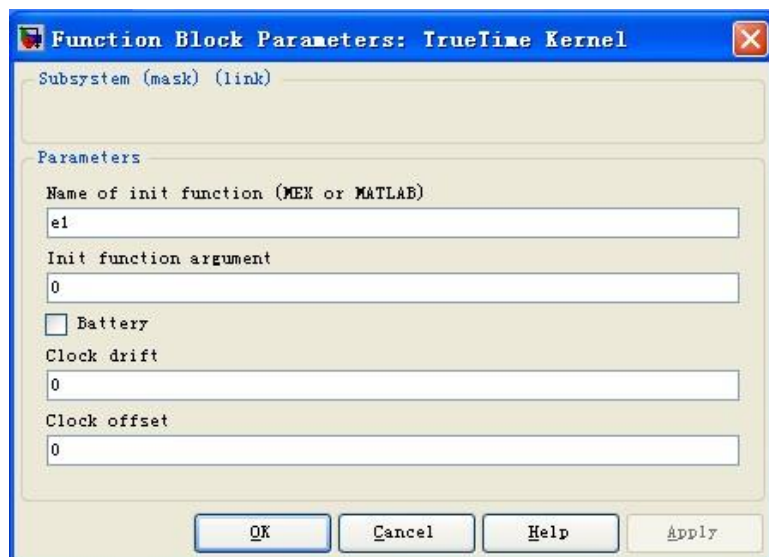


图 3-3 TrueTime Kernel 配置图

### 3.2.2 交换机仿真模型

AFDX 网络的交换机模型由 TrueTime Kernel 模块和 TrueTime Network 模块共同构成，交换机仿真模型示意图如图 3-4 所示。交换机模型中 TrueTime Network 模块之间不能直接相连，需要与 TrueTime Kernel 模块连接，内嵌函数初始化网络模块接口。

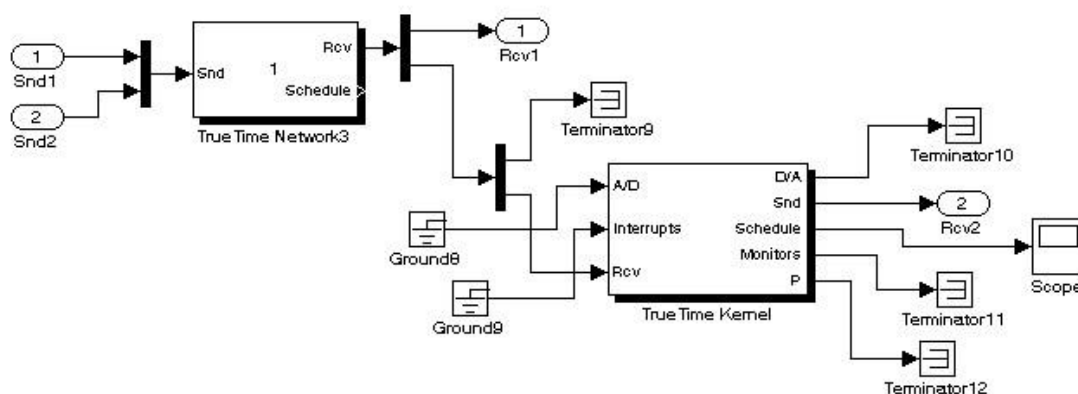


图 3-4 交换机仿真模型

TrueTime Network 模块配置图如图 3-5 所示，选择仿真实验所需的网络协议，根据网络结构设置连接到网络的节点数、数据的传输速率、最小帧长度和数据包丢失率等参数，网络模块输出端连接的 TrueTime Kernel 模块嵌入程序实

现用户自定义的功能，控制程序功能主要包括对连接的网络端口进行初始化并设置中断柄函数，中断柄函数将接受到的数据流分配到不同的网络模块，实现数据帧在特定调度策略下的传输过程。网络模块和内核模块共同完成交换机的功能，设置数据帧的传输路径并分配数据帧的发送网络节点。



图 3-5 网络模块配置

### 3.3 AFDX 网络调度算法仿真及分析

为了验证 AFDX 网络在时间触发调度策略下确定性能的提高，根据理论计算的 AFDX 网络结构建立仿真实验模型，按照 2.2.2 中图 2-8 建立网络仿真模型，根据表 2-1 的参数设置编写数据流模型程序，仿真模型如图 3-6。

图 3-6 中模型由 3.2 中介绍的三个交换机模型和七个终端系统模型与 MATLAB/Simulink 的基本模块共同构成，其中终端系统模型封装成子系统，只显示输入与输出端口，与交换机模型相连。

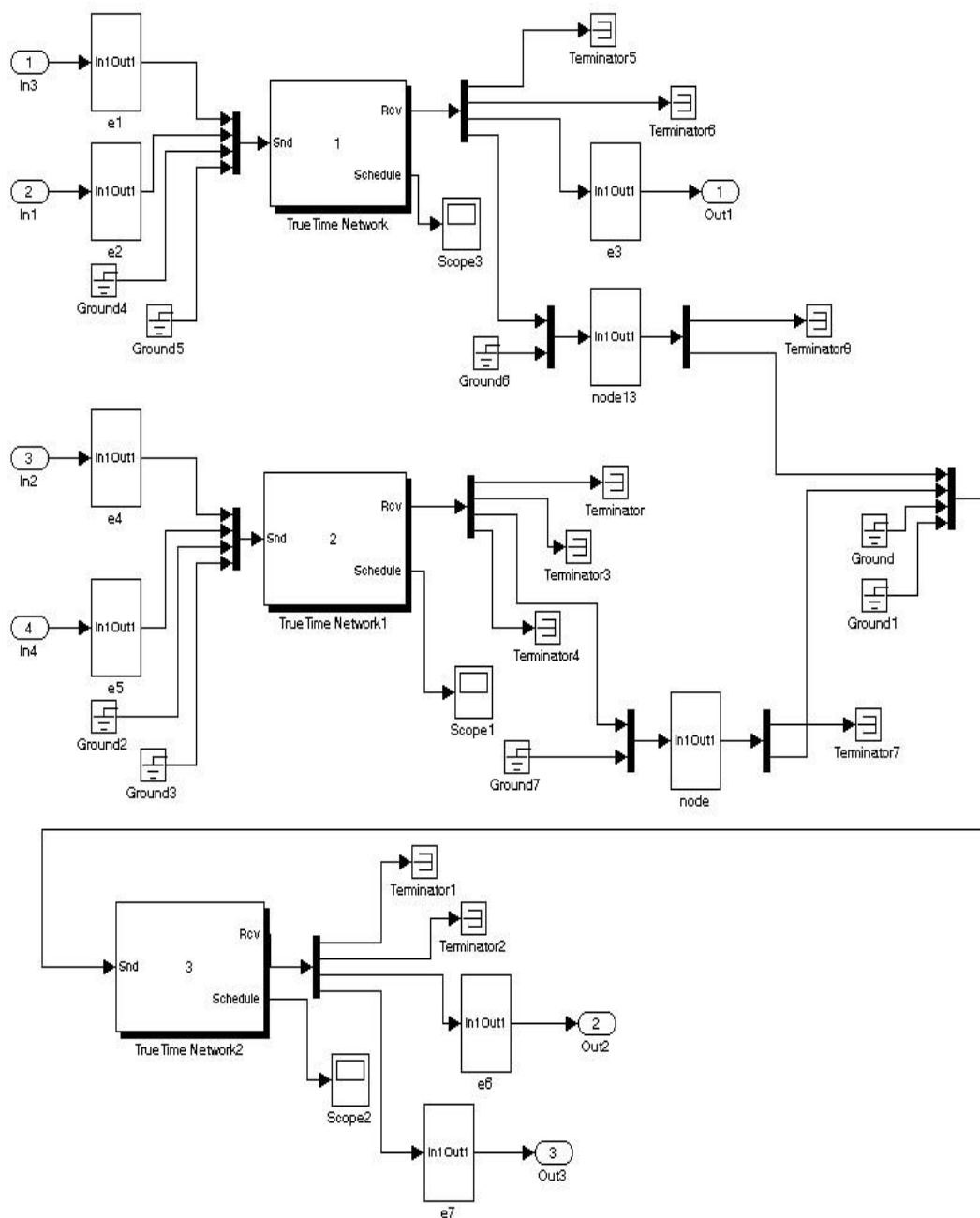


图 3-6 仿真模型

使用 TrueTime 进行仿真时，首先对网络模型中的各个模块初始化，包括指定输入和输出的端口数目、设置优先级、建立中断柄函数等。再根据仿真任务编写控制任务代码，主要由 TrueTime Kernel 模块调用。仿真模型中的程序和用到的主要函数说明如表 3-1 所示。其中网络模块配置选择交换式以太网，数据传输速率设置为 100Mb/s，最小帧长为 512bits，丢包率设定为 0，仿真时间选择为 6.4s，程序中调用静态优先级调度策略。最后按照图 3-6 搭建好的仿真模型开始仿真。

表 3-1 程序说明

源程序	函数	说明
e1、e2、e3、e4、 e5、e6、e7	ttInitKernel ttCreatePeriodicTask ttCreateInterruptHandler ttInitNetwork	初始化内核模块，调用静态优先级调度策略函数；建立周期数据流，设置周期和偏移；设置中断；初始化网络模块接口
S1、S2、S3、node13、 N14、SN31、node	ttGetMsg ttSendMsg	发送、接收数据并分配到下一个网络端口
flow1_1、flow2_2、flow3_3、 flow4_4、flow5_5、flow6_6、 flow7_7、flow8_8、flow9_9、 flow10_10	ttSendMsg ttCurrentTime	建立周期数据流，设定数据帧长度、优先级和接收端口，记录生成数据的时间
S13、S34	ttGetMsg ttCurrentTime	接收数据并计算延时

仿真结束后统计出延时结果，计算结果与仿真结果对比如图 3-7 所示。由图 3-7 的结果对比可以看出，仿真结果比计算值小，计算结果中包含了更多的悲观因素，与仿真结果差距较大。因为计算交换机转发延时这一可变延时要考虑到网络中数据转发过程中的最坏情况，突发度尽量选取最大值，网络演算法计算出的是端到端延时的上限值，包括不必要的悲观情况，而在实际网络数据帧的传输过程中不会出现<sup>[50]</sup>。

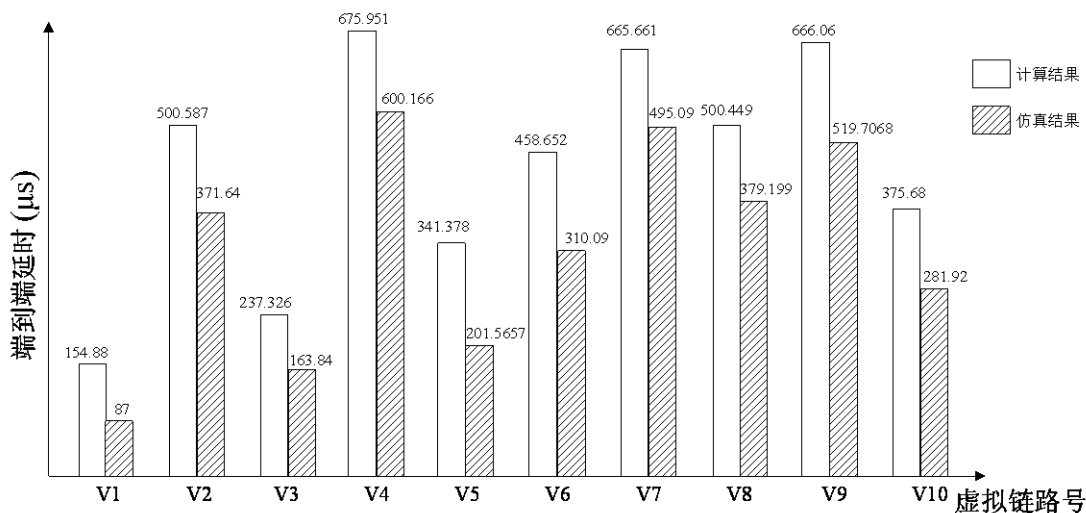


图 3-7 SP 调度策略计算结果与仿真结果对比

TrueTime Network 模块将网络协议配置成 TDMA，模拟时间触发通信协议，时分复用模式下修改交换机模型中的程序，完成时间触发调度，其他参数设置不变，统计仿真结果，将仿真结果与计算结果对比如图 3-8 所示。

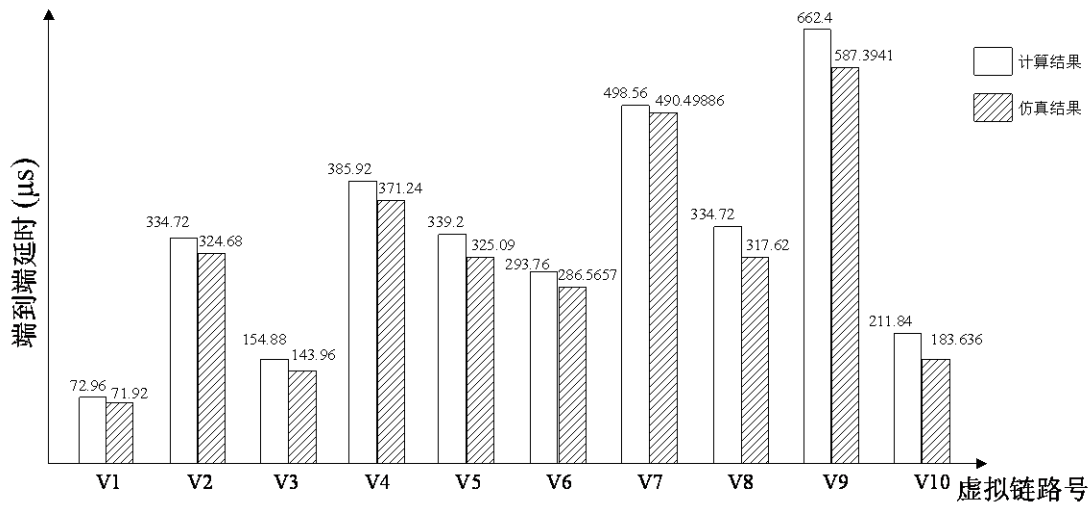


图 3-8 时间触发调度下的计算结果与仿真结果对比

图 3-8 中可以明显看出，计算结果与仿真结果相差很小，利用计算结果大致可以预先估计到实际情况下的端到端延时情况，说明时间触发调度策略在提升网络确定性方面有了很大改善。比较图 3-7 和图 3-8 中的两组仿真结果，在时间触发调度策略下对于实时性能要求较高的高优先级数据端到端延时减小，实时性能也有所改善。

### 3.4 本章小结

本章首先介绍了用于 AFDX 网络调度算法仿真的平台 TrueTime，阐述 AFDX 的终端系统和交换机模型构建和模块参数配置，再对理论计算的模型进行软件仿真，得到网络模型在不同调度策略下的延时结果，最后将仿真结果和理论计算结果对比分析，验证时间触发调度策略与常用调度策略相比在保证网络的确定性方面有很大的改善。

## 第 4 章 时间触发 AFDX 虚拟链路层的 FPGA 实现

### 4.1 虚拟链路层分析与 FPGA 实现的总体设计

时间触发 AFDX 终端系统的虚拟链路层与 ARINC664 Part7 中规定的终端系统虚拟链路层各部分功能相同，在发送模块的调度功能部分做出改进。协议中引入时间触发机制保证通信的确定性，利用两个互为冗余的网络传输增强通信的可靠性<sup>[51]</sup>。终端系统的数据链路层中 MAC 子层实现普通以太网的 MAC 层功能，虚拟链路层部分实现 AFDX 网络所特有的功能<sup>[52]</sup>。总体模块组成如图 4-1 所示。

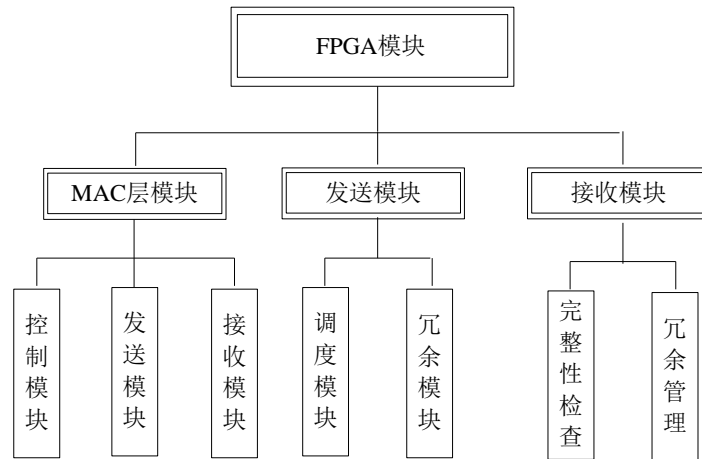


图 4-1 总体结构框图

#### 4.1.1 MAC 子层模块

根据 IEEE802.3 标准中对 MAC 层的规定，MAC 子层负责完成媒体访问管理，通过 MII 接口与以太网物理层芯片连接，将接收到的数据帧解封、进行 CRC 校验，对需要发送的数据帧封装成标准的以太网数据帧<sup>[53]</sup>。MAC 层模块包含数据发送、数据接收、控制模块、MII 管理模块和主机接口模块，如图 4-2 所示。

MAC 结构中各部分模块功能如下：

(1) 主机接口模块：MAC 模块与主机通过 wishbone 总线连接，接口模块内部包含主接口、辅助接口、接收和发送数据缓存的 FIFO 等，其中主接口通过地址线和数据线与外存相连，进行与外存之间的数据交互，辅助接口模块通过地址线和数据线连接 MAC 层模块的内部进行与数据缓冲区之间的数据交换。

主机要发送数据时通过 wishbone 总线将缓冲区中要发送的数据传送给 MAC 模块中的发送缓存，读取发送数据缓冲描述符，接收数据时将数据存储到接收 FIFO 中，发送和接收的状态分别写入发送、接收缓冲描述符中<sup>[54]</sup>。

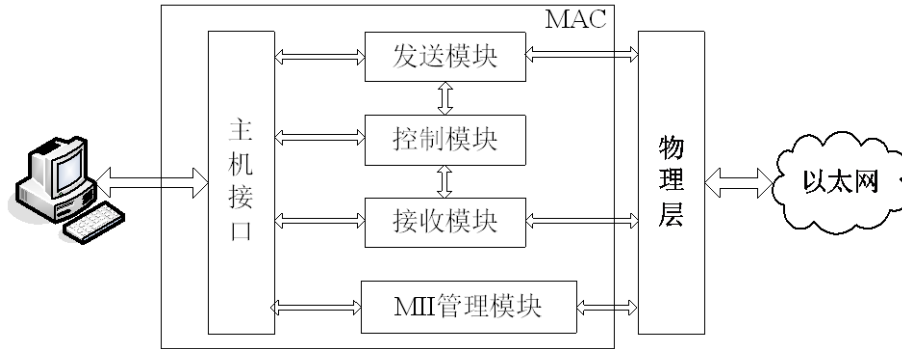


图 4-2 MAC 结构图

（2）发送模块：从发送缓冲区读取数据，对将要发送的数据帧添加前导符和帧起始界定符，为长度不足的数据帧进行填充，封装成标准以太网帧传送给物理层芯片，CRC 校验并在数据帧末端添加 CRC 校验码。发送模块具有冲突检测功能，将信号反馈给控制模块。

（3）接收模块：从物理层接收数据并存储在接收数据缓冲区内，对接收到的数据帧去除前导符，检查目的地址和 CRC 校验，决定是否接收存储数据帧，同时将信号传送给 wishbone 接口。

（4）控制模块：实现流量控制，当接收到流量控制请求时会发送控制帧，防止缓存中有数据帧溢出。发送数据帧时，控制模块读取发送缓冲区数据传递给发送模块；接收数据帧时，控制模块检测接收到的数据帧并输出给 wishbone 模块接收，缓存区存储数据帧<sup>[55]</sup>。

（5）MII 管理模块：提供与物理层芯片交互的接口，可以对物理层芯片进行配置或读取物理层芯片的配置信息。

#### 4.1.2 虚拟链路层发送模块

MAC 层内部的发送模块完成普通以太网对数据帧的发送功能，协议中规定的 AFDX 网络发送数据帧时的特殊处理在虚拟链路层的发送模块中完成。当有消息发往 AFDX 端口时，终端系统的传输层和网络层对消息添加 UDP 头和 IP 头完成封装后传送给虚拟链路层，虚拟链路层发送数据帧的过程如图 4-3 所示。

待发送各个虚拟链路中的数据帧经过流量整形后以该条虚拟链路的 BAG 时间均匀出现，进入调度器模块，按照特定的调度策略进行调度。本文中设计终端系统发送的虚拟链路均为时间触发虚拟链路，则发送部分的调度器模块完



成时间触发调度策略。完成调度后的数据帧进入冗余管理模块，在冗余管理模块中复制数据帧并添加序列号，分别发送到两个冗余网络中以保证 AFDX 通信网络数据发送的可靠性。

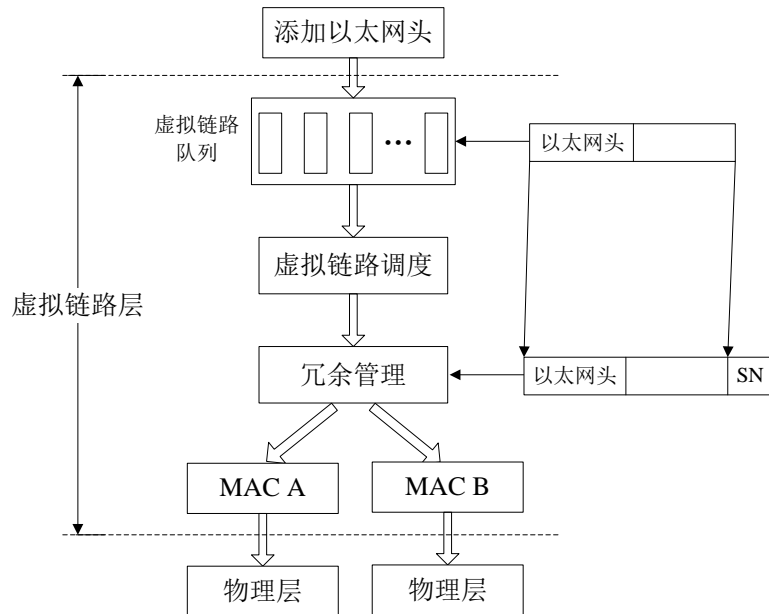


图 4-3 发送过程

### 4.1.3 虚拟链路层接收模块

MAC 层模块的接收部分完成普通以太网接收数据帧的基本功能，虚拟链路层中的接收模块根据协议规定对接收到的数据帧进行 AFDX 网络特有的处理，保证通信网络正确可靠地接收到数据帧。虚拟链路层接收到来自物理层的数据包，对数据帧处理的工作过程如图 4-4 所示。

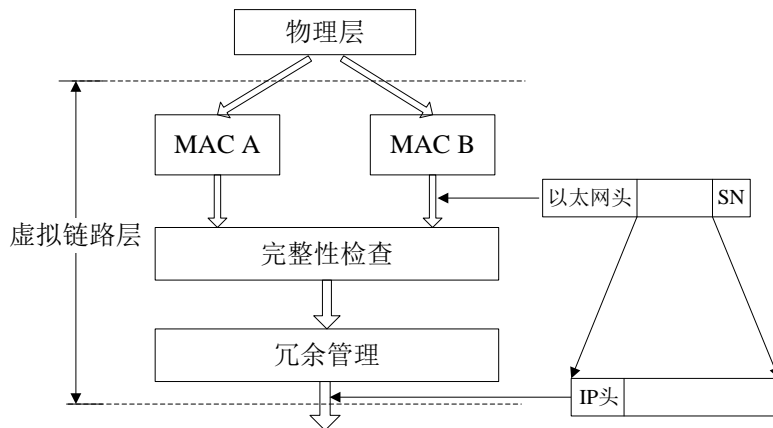


图 4-4 接收过程

进入到虚拟链路层的两组并行互为冗余的数据帧根据帧格式中的序列号位分别进行完整性检查，若序列号不正确将该数据帧丢弃。经过检查后的两组数

据帧进入冗余管理模块，采取“先到先有效”的原则，先接收到的数据帧视为有效帧，传输到上层网络进行处理，丢弃后到达的冗余帧。

## 4.2 虚拟链路层发送模块实现

时间触发 AFDX 终端系统硬件平台设计与普通 AFDX 硬件相同，以 FPGA 为核心，完成发送、接收等功能和 AFDX 协议栈的实现。AFDX 协议在以太网协议基础上修改而成，MAC 层功能与以太网的 MAC 相符，虚拟链路层的发送和接收部分完成 AFDX 网络为保证实时可靠通信所特有的功能。经过多年发展，以太网技术已经相当成熟并得到广泛应用，MAC 层 IP 核的设计已基本固定，本文主要阐述时间触发 AFDX 终端系统虚拟链路层的发送和接收模块的 FPGA 实现，以完成区别于以太网的特殊处理。

### 4.2.1 发送模块结构

发送模块根据 AFDX 网络的特殊要求，完成寄存器的配置、流量整形、虚拟链路调度和冗余处理的功能。时间触发 AFDX 终端系统在普通 AFDX 终端系统的基础上对发送部分的调度策略进行优化，终端系统虚拟链路层的发送模块内部结构如图 4-5 所示。

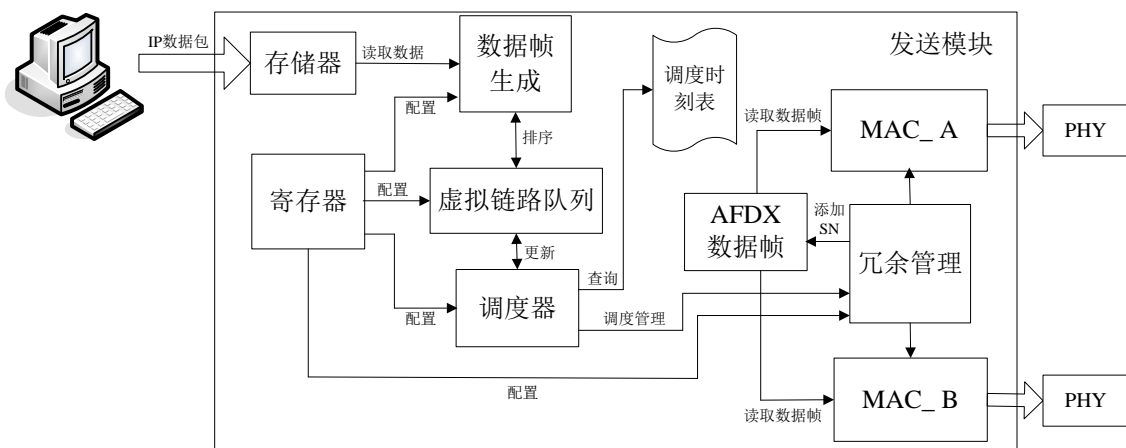


图 4-5 发送模块结构图

发送模块内部各部分简要介绍如下：

(1) 存储器模块：读取上位机待发送的数据包，存储在存储器当中等待读取数据处理。本文中设计 4 条虚拟链路，存储器模块中包含 4 组存储器缓存 4 条虚拟链路上的数据。

(2) 数据帧生成模块：取回 IP 数据包，添加虚拟链路号等 AFDX 数据帧信息，存储地址指针，等待流量整形后被调度。

(3) 寄存器模块：配置并存储各个模块的配置信息，如虚拟链路的 BAG 值、 $L_{\max}$  值、静态优先级等信息，设定相关模块的标志位控制数据帧的发送。

(4) 调度器模块：主要负责每个虚拟链路中的数据流量控制及多个虚拟链路间的调度管理。其中流量控制是根据每条虚拟链路 BAG 值的配置，设置每条虚拟链路的 BAG 定时器，控制虚拟链路上相邻数据帧的时间间隔。调度管理根据离线设计好的调度时刻表查询每条虚拟链路被调度的时刻，设计定时器，根据寄存器中的信息，优先调度优先级高的虚拟链路，查询是否到达下一条虚拟链路的调度时刻，若到达进行调度否则等待。

(5) 虚拟链路队列：包含每条虚拟链路的队列信息，设置标志位表明虚拟链路的发送状态，以便控制模块读取虚拟链路的发送信息。根据每条虚拟链路的不同优先级设定，对每条虚拟链路排序，存储优先级信息。

(6) 冗余管理模块：根据配置决定数据帧是否复制并同时发送到两个冗余网络，为发送的数据帧添加 SN 号，以便接收模块对数据帧进行完整性检查。

(7) MAC 模块：完成以太网 MAC 层功能，发送数据帧。

发送模块的流程图如图 4-6 所示。

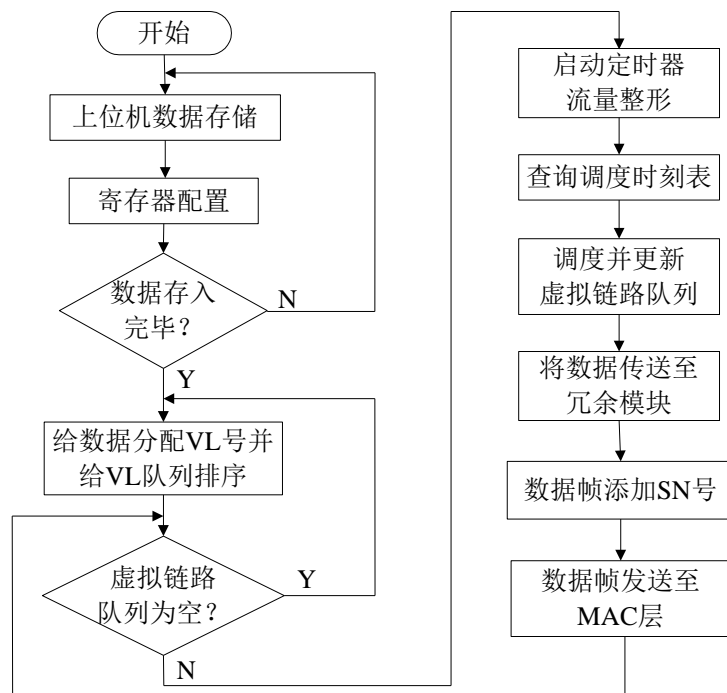


图 4-6 发送模块流程图

## 4.2.2 存储器模块

存储器模块读取主机发送的 IP 数据包，按照协议中规定的 AFDX 数据帧格式处理数据帧，存储虚拟链路信息完成对虚拟链路优先级排队。存储器模块的状态机设计如图 4-7 所示。

存储器模块处于空闲状态时，读取缓存在 fifo 中的数据包，为数据帧添加 MAC 源地址、目的地址和虚拟链路号，设置每条虚拟链路的地址指针，将数据帧映射到相应的虚拟链路，虚拟链路数据帧的地址指针、帧长度等信息存储到相应的虚拟链路对列当中。根据虚拟链路映射的地址，分配虚拟链路队列到不同的存储区域等待流量整形和调度处理。完成虚拟链路队列信息缓存，根据优先级的设置对虚拟链路排序，完成后进入存储器空闲状态，等待下一个数据包的到来。

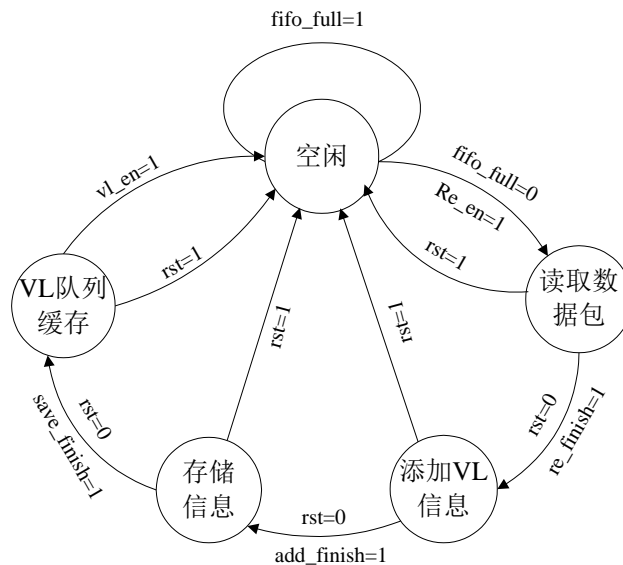


图 4-7 存储器模块状态机

## 4.2.3 寄存器模块

寄存器模块配置 MAC 层 IP 核的工作模式寄存器、中断源寄存器、发送缓冲描述符等信息，设置虚拟链路数据帧的最大长度和寄存器模块与各个模块相连的控制位与标志位。寄存器内部结构如图 4-8 所示。地址指针由 00 开始设置不同功能寄存器存储相应信息，控制虚拟链路数据帧的发送。同时在寄存器模块中产生握手信号等寄存器模块与其他模块的联络信号，保证各个模块之间进行正常的交换。

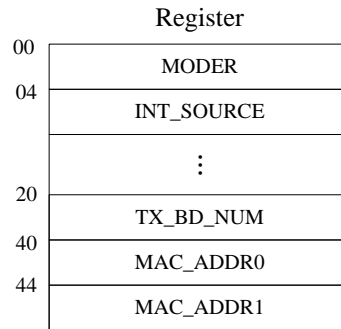


图 4-8 寄存器内部结构

#### 4.2.4 调度模块

时间触发 AFDX 终端系统的虚拟链路层采用时间触发调度策略，本文设计终端系统的虚拟链路为四条时间触发虚拟链路，每条虚拟链路的 BAG 均为 2ms，调度模块完成流量整形和时间触发调度功能。调度模块工作过程如图 4-9 所示。

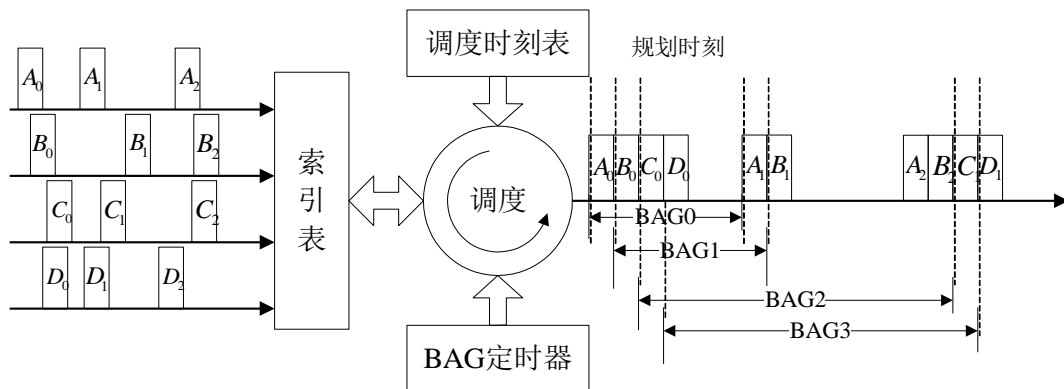


图 4-9 调度过程

##### 4.2.4.1 BAG 定时器模块

终端系统中的每一条虚拟链路都有各自的 BAG 值，BAG 定时器根据虚拟链路的 BAG 值设定定时时间为 2ms，每条虚拟链路 BAG 定时器由计数器完成，根据时钟周期可以计算出计数值 BAG\_value，每个 BAG 定时器独立运行。定时器程序流程图如图 4-10 所示。

由复位信号将定时完成标志 time\_up 和传输标志 FTT 置 0，查询虚拟链路队列上是否有数据帧到达，若有数据帧则启动 BAG 定时器，否则继续等待。当 BAG 的计数器计数值达到 BAG\_value 时，标志位 time\_up 和 FTT 分别置 1，计数器的计数初值置 0，重新计数，输出同一条虚拟链路上的下一个数据帧，使相邻的数据帧均以 BAG 时间间隔出现，完成流量整形。time\_up 和 FTT 与调度器的控制模块相连，根据 time\_up 状态判断虚拟链路是否流量整形，FTT 决定是否传输数据帧。

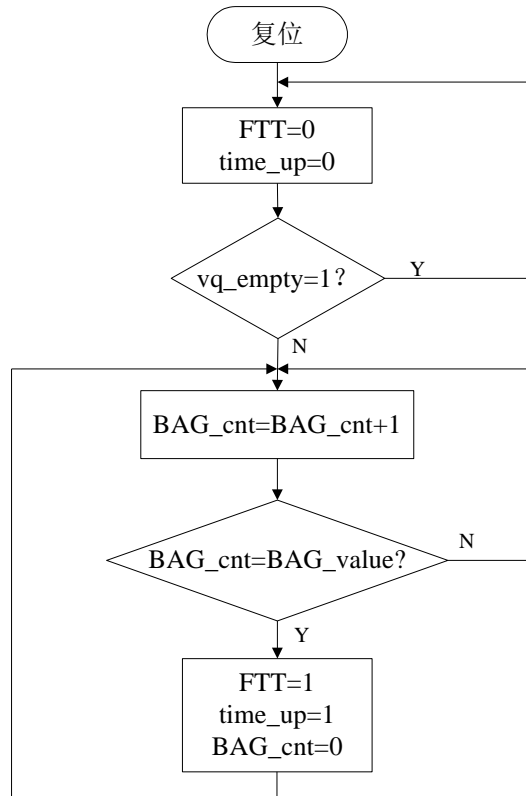


图 4-10 BAG 定时器流程图

#### 4.2.4.2 调度控制模块

时间触发调度策略是对虚拟链路数据帧发送过程的离线设计，根据虚拟链路的 BAG 值和最大帧长度预先计算好数据帧的发送时刻，严格按照调度时刻表调度数据帧。由于离线设计的可扩展性有限，在规划调度时刻表时，按照每条虚拟链路上的数据帧长度均为最大帧长 1538bytes 计算，虚拟链路 V0、V1、V2、V3 的优先级依次递减。设计调度时刻表如图 4-11 所示。

基本周期0	V <sub>0,1</sub>	V <sub>1,1</sub>	V <sub>2,1</sub>	V <sub>3,1</sub>	空余时间	1ms
基本周期2	V <sub>0,2</sub>	V <sub>1,2</sub>	V <sub>2,2</sub>	V <sub>3,2</sub>		1ms
基本周期4	V <sub>0,3</sub>	V <sub>1,3</sub>	V <sub>2,3</sub>	V <sub>3,3</sub>		1ms
基本周期6	V <sub>0,4</sub>	V <sub>1,4</sub>	V <sub>2,4</sub>	V <sub>3,4</sub>		1ms
基本周期8	V <sub>0,5</sub>	V <sub>1,5</sub>	V <sub>2,5</sub>	V <sub>3,5</sub>		1ms
基本周期10	V <sub>0,6</sub>	V <sub>1,6</sub>	V <sub>2,6</sub>	V <sub>3,6</sub>		1ms
⋮	⋮	⋮	⋮	⋮	⋮	⋮
基本周期127						1ms

}

矩阵周期  
128ms

图 4-11 调度时刻表

在基本周期 0 中高优先级的虚拟链路 V0 的第一个数据帧在 0 时刻被调度, 计算虚拟链路 V1 的第一个数据帧被调度时刻为:

$$t_{1,1} = \frac{8 \times 1538}{100} = 123.04 \mu\text{s}$$

同理可计算虚拟链路 V2 和 V3 第一次被调度的时刻分别为  $246.08 \mu\text{s}$  和  $369.12 \mu\text{s}$ , 完成第一次调度后四条虚拟链路分别以  $2\text{ms}$  为周期再次被调度。调度过程以 V0、V1 两条虚拟链路为例说明, 调度控制时序原理图如图 4-12 所示。

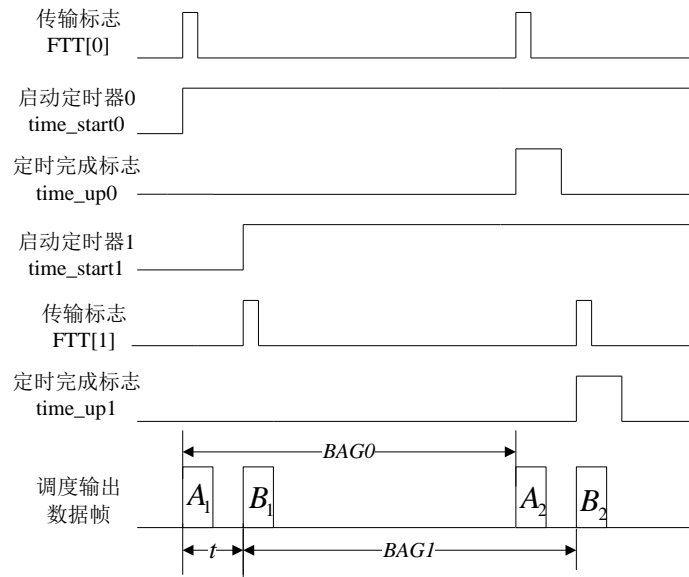


图 4-12 调度控制时序图

虚拟链路 V0 上的数据帧标记为  $A_1$ 、 $A_2$ , 虚拟链路 V1 上的数据帧为  $B_1$ 、 $B_2$ , 两条虚拟链路的 BAG 值分别为  $BAG0$  和  $BAG1$ 。首先查询高优先级的虚拟链路 V0 上是否有数据帧需要被调度, 检测到数据帧  $A_1$  时将 V0 的传输标志位 FTT[0] 置 1, 其它虚拟链路的传输标志位置 0, 此时只有 V0 占用总线, 直接输出  $A_1$ , 复位 FTT[0], 启动 V0 的 BAG 定时器和虚拟链路 V1 开始被调度时刻的定时器。根据调度时刻表的计算, V1 开始被调度时刻与 V0 第一次被调度相隔  $123.04 \mu\text{s}$ , 图 4-12 中表示为时间  $t$ , 则将 V1 初次调度时刻定时器设置为  $123.04 \mu\text{s}$ 。BAG 定时器定时标志 time\_up0 置 1 时发送 V0 的第二个数据帧  $A_2$ 。在 FTT[0] 第一次置 1 之后的  $123.04 \mu\text{s}$  即 V1 初次调度时刻定时器到时, V1 的传输标志 FTT[1] 被置位, 输出数据帧  $B_1$  并复位 FTT[1], 同时启动 V1 的 BAG 定时器, 定时时间到达时 time\_up1 置 1, 输出数据帧  $B_2$ 。

根据上述调度过程编写调度控制模块程序, 调度控制模块发送 V0、V1 的

第一个数据帧程序流程图如图 4-13 所示。

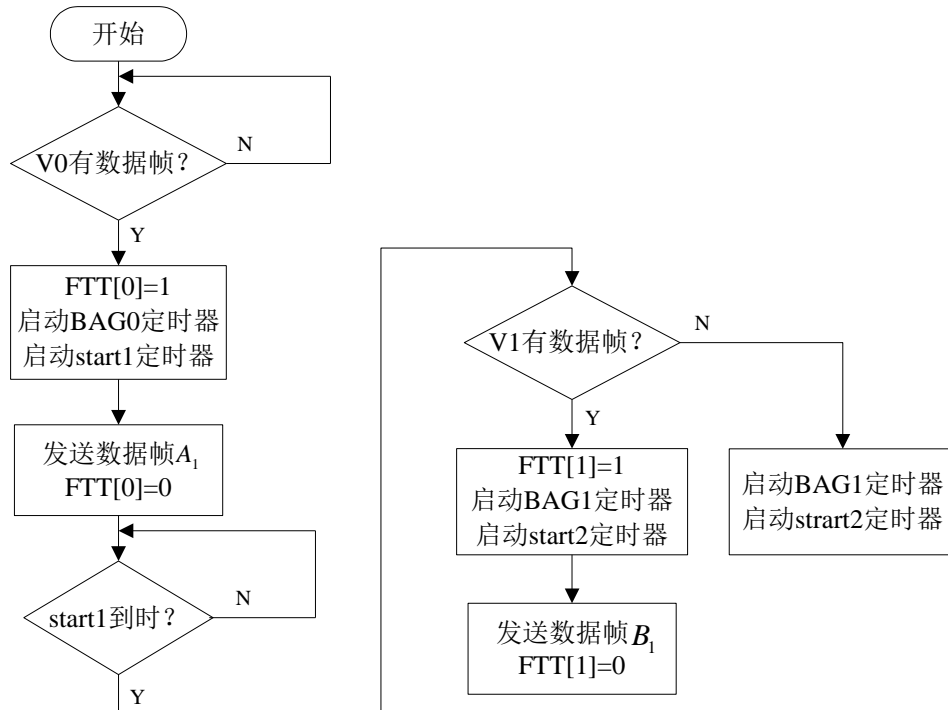


图 4-13 调度控制程序流程图

#### 4.2.5 发送冗余管理模块

发送部分的冗余管理模块接收调度模块的数据帧信息和控制信号，为虚拟链路上的数据帧添加序列号，以便接收模块进行完整性检查。根据寄存器模块的配置信息决定数据帧发往 A 网络还是 B 网络，或复制数据帧同时发送到两个网络。冗余管理模块的内部结构如图 4-14 所示。

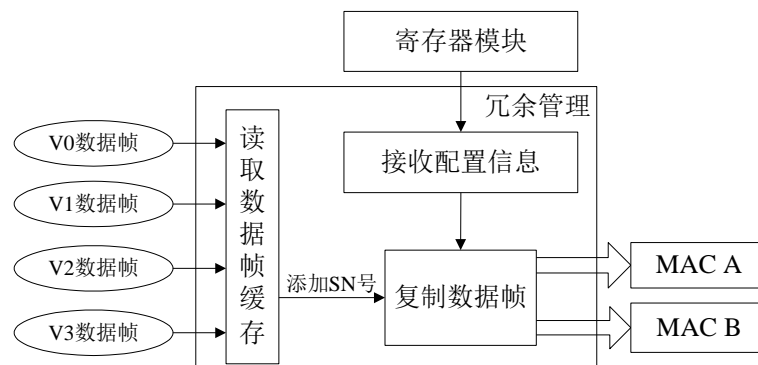


图 4-13 冗余管理模块结构



## 4.2.6 终端系统发送模块仿真

在 Xilinx ISE 环境下利用 Verilog 语言编程实现上述发送模块各部分的功能，虚拟链路层发送模块与 MAC 层模块共同构成时间触发 AFDX 终端系统的数据链路层发送部分，仿真测试原理图如图 4-14 所示。

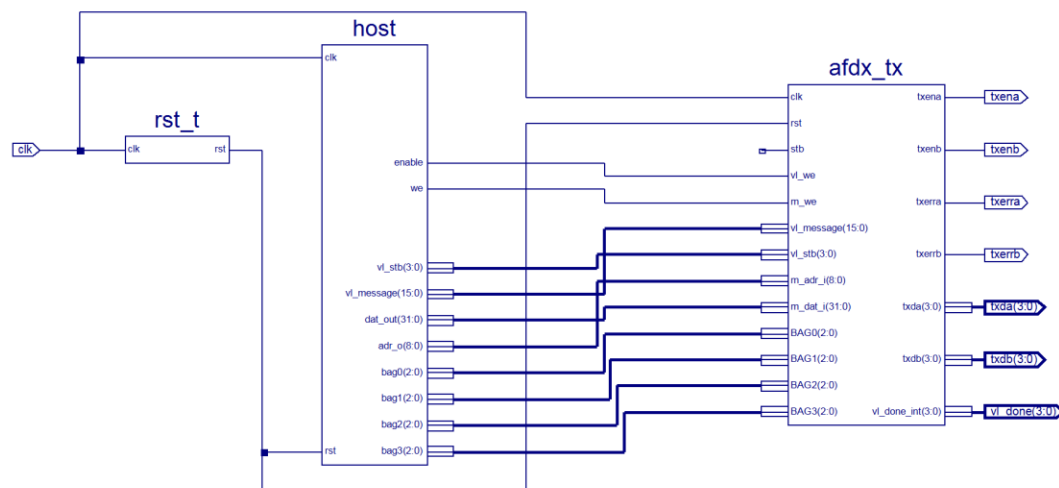


图 4-14 发送模块仿真测试原理图

仿真测试原理图中的 host 模块模拟主机，提供待发送的数据包和发送部分需要的信息，rst\_t 模块产生复位信号，afdx\_tx 模块中包含以太网 MAC 模块和虚拟链路层发送模块，包括调度模块、冗余管理模块等部分，共同实现时间触发终端系统虚拟链路层的发送功能。host 模块中设置四条虚拟链路，BAG 均为 2ms，数据帧长度为 60bytes，需要同时发送到两个网络，输出为四位数据。编写各个模块程序，通过 ISE 软件中的 XST 工具综合后，建立 testbench 文件波形仿真结果如图 4-15 所示。

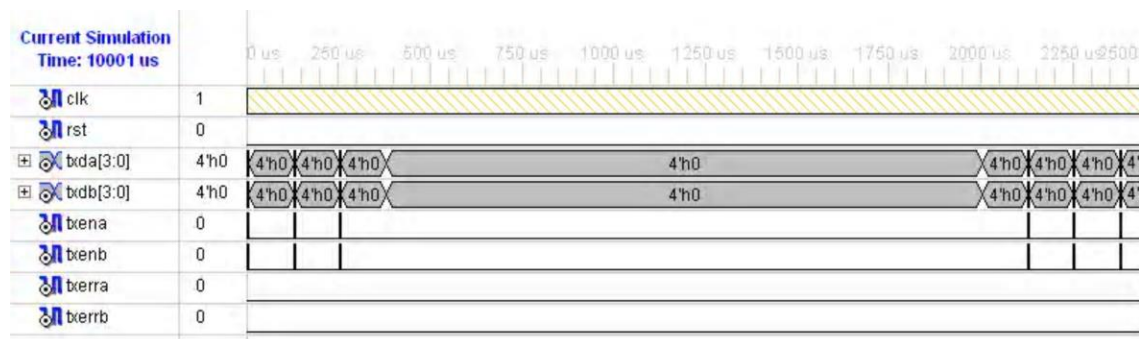


图 4-15(a) 仿真结果

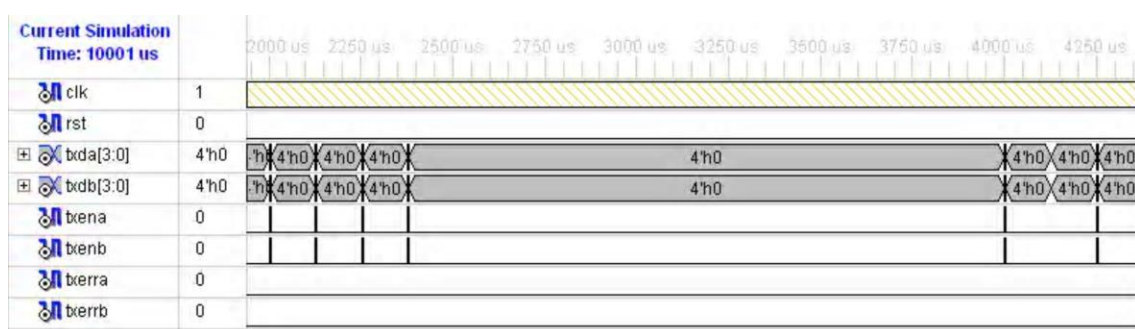


图 4-15(b) 仿真结果

程序中设计每条虚拟链路上各有三个数据帧，输出数据帧的示意图如图 4-16 所示。

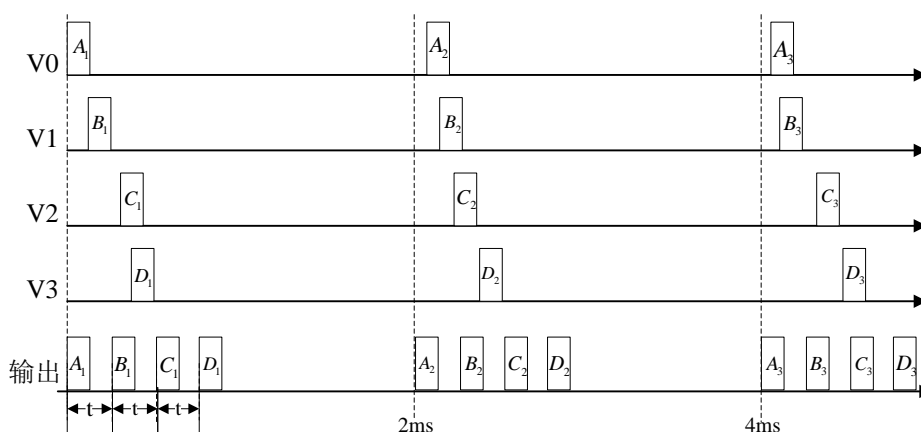


图 4-16 输出数据帧示意图

虚拟链路 V0-V3 依次发送数据帧，相邻两个数据帧间隔时间  $123.04\mu\text{s}$ ，完成第一次调度后，在 BAG 时间到达后进行第二次调度。由于数据帧生成到被调度中间有一段很小的时间，计划在 0 时刻被调度的  $A_1$  仿真结果中在  $3.2\mu\text{s}$  出现，其后被调度的数据帧都严格按照规划时刻出现。由仿真结果观察到每条虚拟链路都完成了流量整形和时间触发调度等功能。

### 4.3 虚拟链路层接收模块实现

时间触发 AFDX 终端系统的虚拟链路层接收模块与普通 AFDX 终端系统的接收功能相同，考虑到时间触发 AFDX 终端系统与普通 AFDX 终端系统之间的兼容性，接收部分也要考虑到接收到由普通 AFDX 终端系统发送的速率约束虚拟链路数据帧时出现的情况，并作出相应的处理。

#### 4.3.1 接收模块结构

时间触发 AFDX 终端系统的虚拟链路层接收模块功能与普通的 AFDX 终端

系统接收功能相同，根据协议规定需要完成完整性检查和冗余处理<sup>[56]</sup>。接收模块内部结构框图如图 4-17 所示。

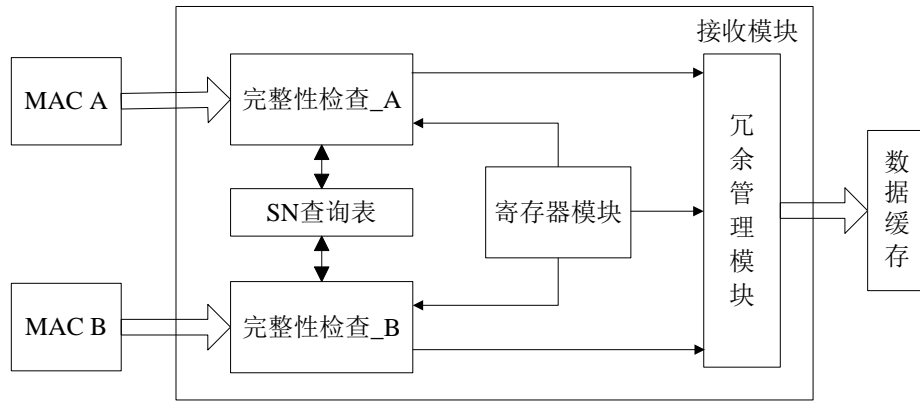


图 4-17 接收模块结构图

接收模块内部各部分介绍如下：

(1) 完整性检查：接收模块接收到来自两个冗余网络的数据帧，检查同一虚拟链路上的数据帧的序列号，根据 SN 查询表，按照协议规定去除无效帧。

(2) 寄存器模块：配置各部分的工作模式和状态，存储各个模块的配置信息和数据在各个模块间传输的标志位信息。

(3) 冗余管理模块：冗余管理模块接收来自两个经过完整性检查模块的同序列号的数据帧，根据“先到先有效”原则，对后到达的帧副本删除，保留先到达的有效帧输出。

### 4.3.2 完整性检查模块

完整性检查模块将接收到的来自两个网络的数据帧存储在内部的 fifo 中，读取 fifo 中数据帧的虚拟链路号和序列号存入到 SN 查询表，以便对数据帧进行 SN 号检查。完整性检查模块工作状态机如图 4-18 所示。

复位信号有效时为空闲状态，当复位信号失效并且读取数据信号有效时意味着有数据帧到达，读取数据帧后存储在 fifo 中同时将数据帧的虚拟链路号和 SN 号存储到 SN 查询表中。SN 查询表先接收到的 SN 号记为 PSN，接收到的同虚拟链路上数据帧的 SN 号与 PSN 比较，协议中规定 SN 需满足 $[PSN+1, PSN+2]$ ，保证接收到数据帧的连续性。SN 范围为 0~255，SN=0 为复位后发送的第一个数据帧，下一个数据帧的 SN 为 1。当无复位帧出现时 PSN=255 下一个接收到的 SN 号应为 1。不满足 SN 号规则的数据帧视为无效帧，对数据帧完成 SN 检查后，输出有效帧丢弃无效帧，完成数据帧传输进入空闲状态。

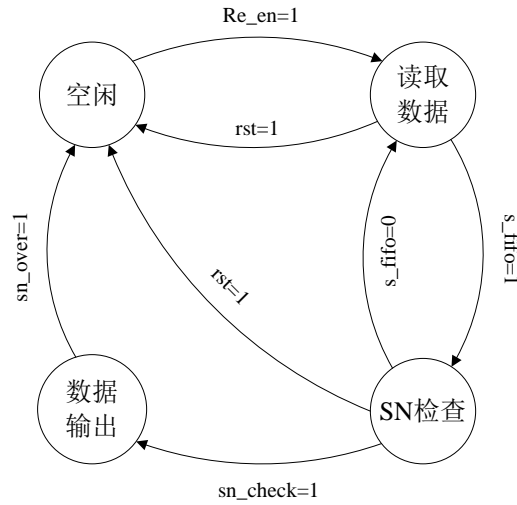


图 4-18 完整性检查模块状态机

### 4.3.3 接收冗余管理模块

由完整性检查模块输出的数据帧进入冗余管理模块，对两个冗余网络的有效数据帧采取“先到先有效”原则处理。时间触发虚拟链路采用时间触发调度策略，数据帧传输过程可预测，到达目的终端系统的时间根据计算结果可估计，速率约束虚拟链路采用常用的调度策略而在传输过程中产生抖动，交换机的调度会造成两个冗余网络数据帧的传输不同步，目的终端系统接收到的数据帧时刻存在偏差。冗余管理中 **SkewMax** 是一个重要参数，AFDX 协议中将 **SkewMax** 定义为接收到的同序列号两个互为冗余数据帧之间的时间间隔，根据通信网络中由发送终端系统和交换机调度引起的 **SkewMax** 判定无效的冗余帧<sup>[57]</sup>。数据帧到达接收端时可能出现数据帧丢失的情况，冗余处理结果如图 4-19 所示，AFDX 网络中允许丢失一帧的情况出现。

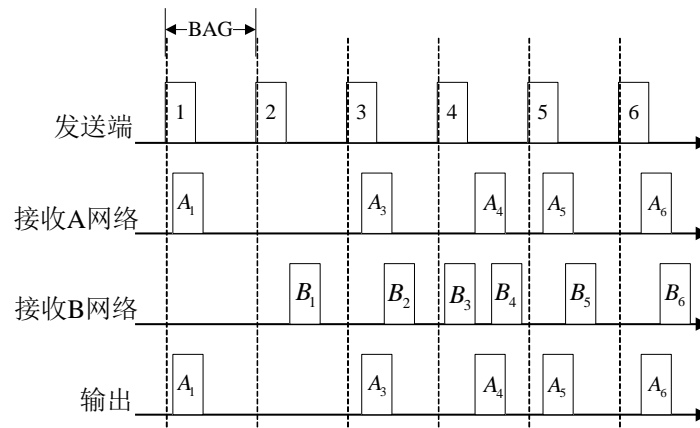


图 4-19 接收端丢帧情况

接收到的数据帧来自两个网络，冗余管理模块内部包含两个相同的处理单元分别对 A 网络和 B 网络的数据帧进行冗余处理，以 A 网络冗余处理为例，程序流程图如图 4-20 所示。

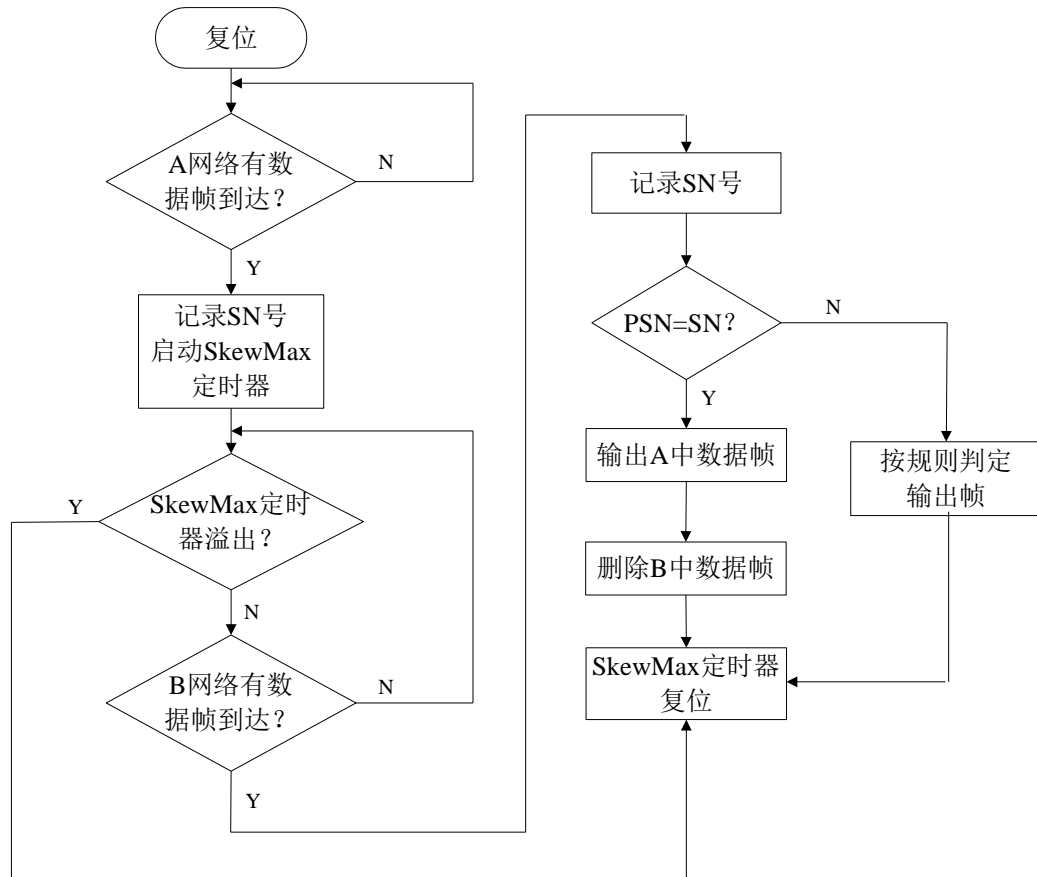


图 4-20 冗余处理流程图

系统复位后开始检测是否有 A 网络的数据帧到达，有数据帧到达记录 SN 号并启动 SkewMax 定时器，等待 B 网络相同 SN 号数据帧的到达。SkewMax 定时器溢出时，将定时器复位并比较接收到的 SN 与上一次接收的 PSN。在 SkewMax 定时器到时之前，若有 B 网络的数据帧到达，查询 SN 号，比较 PSN 与 SN。若 PSN 不小于 SN，则输出先到达的 A 网络数据帧，删除 B 网络的数据帧；若  $PSN < SN$ ，当 PSN 与上一个接收到的 SN 号连续且递增满足 SN 号范围时，输出 A 网络数据帧。完成一次数据帧冗余处理后 SkewMax 定时器清零。若 SkewMax 定时时间内没有收到备份帧，则无条件接收下一个数据帧。

#### 4.3.4 终端系统接收模块仿真

对上述接收模块的各部分编程，完成虚拟链路层接收模块功能。虚拟链路

层的接收模块与 MAC 子层模块相连，完成终端系统的接收功能。各部分模块编程后生成器件模块，仿真测试原理图如图 4-21 所示。

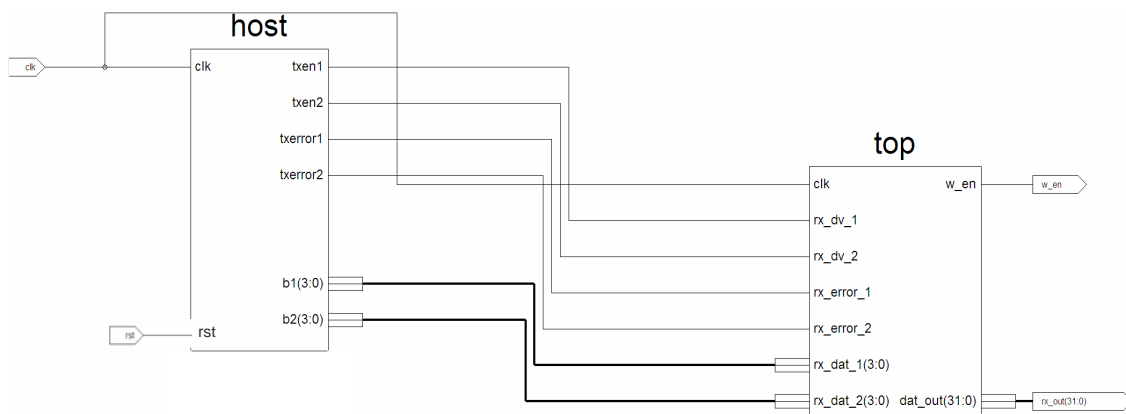


图 4-21 仿真测试原理图

模拟主机模块 **host** 提供待接收的数据帧和接收使能信号，**top** 模块中包括以太网 MAC 层模块、完整性检查模块和冗余管理模块。**host** 模块输出的两组四位数据模拟来自两个网络的数据帧，**top** 模块接收数据帧处理后输出一组 32 位数据。在 ISE 环境下对测试模块综合，添加波形仿真文件，运行仿真结果如图 4-22 所示，其中 **Operating\_Data\_Output** 为两组经过完整性检查的数据帧，经过冗余管理模块处理后的输出数据帧为 **rx\_out**。

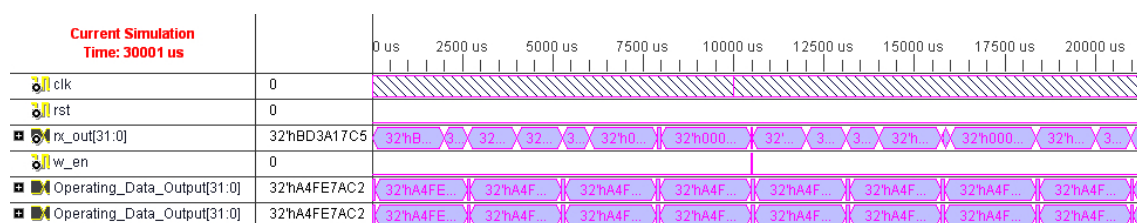


图 4-22 接收模块仿真结果

仿真模块中设计为正常情况下的虚拟链路上的数据帧，仿真结果的数据帧示意图如图 4-23 所示，最后输出的数据帧不存在丢帧和错误帧的特殊情况。

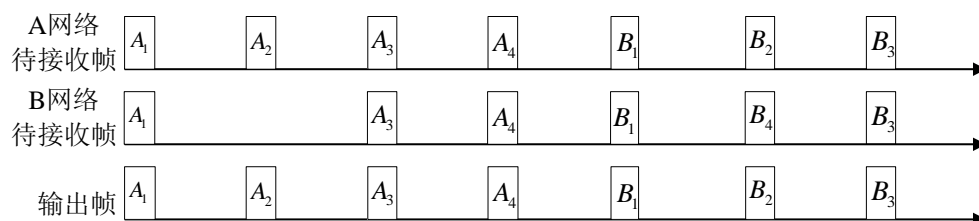


图 4-23 输出数据帧示意图

## 4.4 虚拟链路层全双工仿真

在寄存器模块设置终端系统工作在全双工方式，将终端系统虚拟链路层的发送模块和接收模块合并完成时间触发 AFDX 终端系统的收发功能，与以太网 MAC 层模块连接，实现终端系统的全双工数据传输。仿真测试原理图如图 4-24 所示。

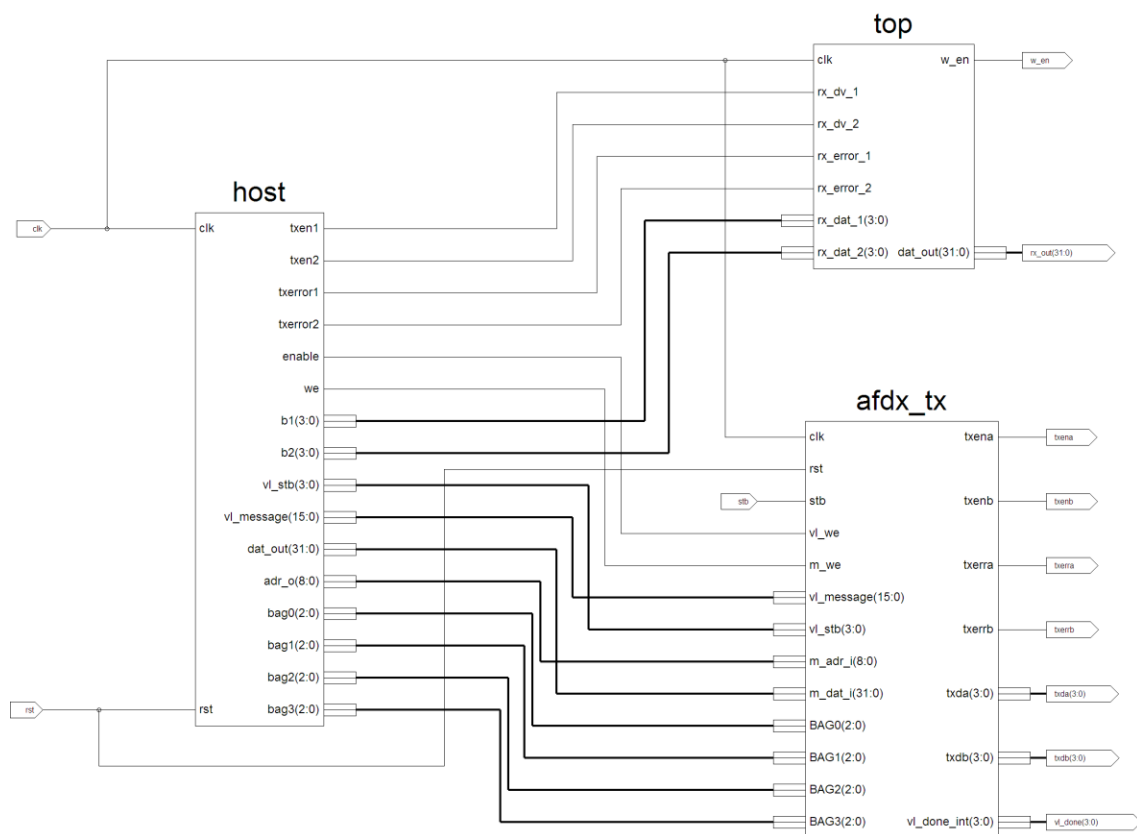


图 4-24 全双工仿真原理图

建立 testbench 文件，进行功能性仿真，仿真结果如图 4-25 所示，top 模块对接收到的数据帧进行处理，afdx\_tx 模块完成发送功能，host 模拟主机发出数据帧发送和接收命令，待发送和待接收数据帧同时传送给终端系统的发送和接收部分，发送和接收模块分别输出经过处理后的数据帧。仿真结果表明虚拟链路层可以运行在全双工的工作方式，同时进行数据的发送和接收。

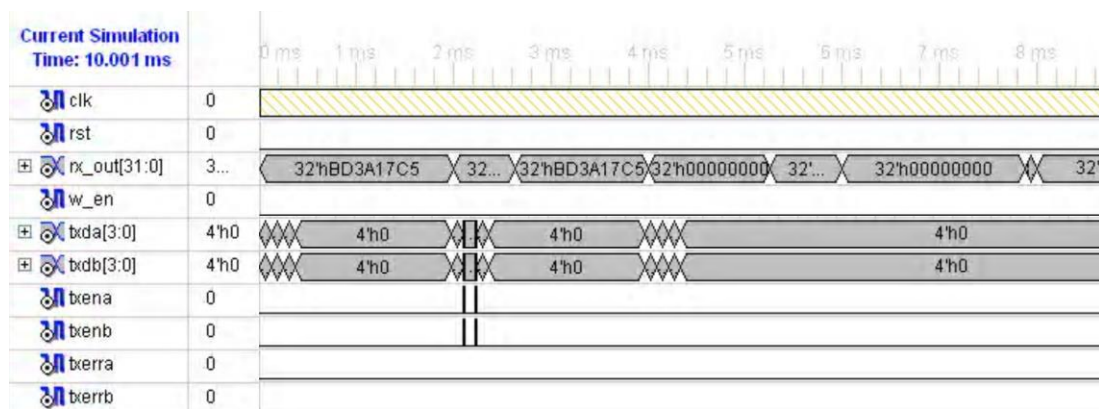


图 4-25 仿真结果

## 4.5 本章小结

本章主要阐述时间触发 AFDX 终端系统虚拟链路层的 FPGA 实现方案。首先从 AFDX 协议出发，分析协议规定终端系统所要实现的功能。再根据各部分功能需求，对各个功能模块进行设计，包括发送部分的虚拟链路调度、冗余管理，接收部分的完整性检查和冗余处理。最后分别将发送部分和接收部分与 MAC 模块联调，对其进行功能仿真，并对发送接收的合并结构进行全双工仿真，完成虚拟链路层功能的 FPGA 实现。



## 第 5 章 时间触发 AFDX 终端系统性能测试

### 5.1 测试系统实验平台

测试平台包括 AFDX 通信板、上位机和网络分析仪，实验平台如图 5-1 所示，验证时间触发 AFDX 终端系统的通信性能。



图 5-1 系统测试平台

实验中由上位机发送数据包到下载有全双工通信程序的 AFDX 通信板，网络分析仪在链路中捕捉观测传输的数据帧，通过网口连接到另一个 AFDX 通信板接收数据帧，同时连接上位机观测接收到的数据帧。AFDX 通信板完成终端系统中的虚拟链路层和物理层任务，终端系统中有两个 MAC 层模块与 AFDX 互连网络连接通信，根据协议规定需要完成调度、冗余管理等功能，由通信板上的 FPGA 芯片完成，对芯片的性能提出了很高的要求，不仅需要足够的逻辑处理单元还需要足够的存储空间。

AFDX 通信板上主要芯片为 FPGA 芯片 Virtex-4 系列的 XC4VSX25，具有低功耗、高速信号处理能力等特点，分布式 RAM 容量 160Kb，块 RAM 容量达到 2304Kb，扩展 FPGA 应用的灵活性，块 RAM 可被配置成单端口 RAM、双端口 RAM、FIFO 等存储结构，10240 个 Slice 提高逻辑运算效率。FPGA 芯片的可用 IO 被分成 10 个 BANK，分区后的 IO 设计为承担不同功能的系统配置 IO 和扩展 IO。FPGA 芯片利用 JTAG 接口下载调试和程序程序。FPGA 芯片的内部供电电压、JTAG 等外部连接器件的电压和 IO 口的电压由电源模块提供，电源模块部分采用 TPS54610 和两个 TPS54310 芯片将 5V 转换成 1.2V、2.5V 和 3.3V 供电电压。通信板上的物理层采用专用芯片 LXT973，兼容 IEEE802.3 协议的 LXT973 内置 100BASE-T 和 10BASE-T 高速以太网收发器。由于 AFDX 的高通信速率影响，为减少电磁干扰，保证终端系统的服务质量，在 LXT973 与接入的网络中间连接隔离变压器，主要隔离高频信号的干扰。由 RJ45 网口作为经过处理后的信号与上位机或通信板通信的端口，AFDX 冗余容错需求设计两个 RJ45 端口，以便终端系统之间的互为备份双网络通信连接。

数据帧在两个终端系统及链路上传输时，连入 AFDX 网络分析仪显示通信过程中的数据帧，监测网络通信情况，通过冗余网口连接接收端通信板。AFDX 网络分析仪选用北京华力创通公司的 AFDX 网络分析仪，专门用来对 AFDX 网络中数据帧的传输进行采集和分析。

## 5.2 终端系统通信演示

按照 5.1 中实验平台连接，终端系统间数据交换由发送端经过网络分析仪传输到接收端，连接上位机通过在线逻辑分析仪 ChipScope 观察接收到的数据帧。发送端发送数据帧为正常情况，虚拟链路 V1-V4 依次被调度，同一条虚拟链路上相邻数据帧的序列号连续递增，序列号的范围为 0~255。网络分析仪捕捉数据帧如图 5-2 所示。图中所示的每条虚拟链路无丢帧和错误帧出现，优先级相邻的两条虚拟链路相隔 123.04  $\mu$ s 被调度，由于时钟和链路的影响，观察到发送数据帧的时间间隔与预期值存在偏差，完成终端系统的发送功能。



No.	时间(月.天.时.分.秒.毫秒)	TAP端口	SN	VL	源MAC	目的MAC	源IP	目的IP	协议	源端口
2550	04.14.3.46.16.574.389	2	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.188.182.188	0.55.0.0	UDP	0
2551	04.14.3.46.16.574.421	3	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.188.182.188	0.55.0.0	UDP	0
2552	04.14.3.46.16.574.612	2	1	2	00-05-5d-61-58-a8	00-03-00-00-00-02	88.188.182.188	0.55.0.0	UDP	0
2553	04.14.3.46.16.574.837	3	1	2	00-05-5d-61-58-a8	00-03-00-00-00-02	88.188.182.188	0.55.0.0	UDP	0
2554	04.14.3.46.16.574.836	2	1	3	00-05-5d-61-58-a8	00-03-00-00-00-03	88.188.182.188	0.55.0.0	UDP	0
2555	04.14.3.46.16.574.858	3	1	3	00-05-5d-61-58-a8	00-03-00-00-00-03	88.188.182.188	0.55.0.0	UDP	0
2556	04.14.3.46.16.574.758	2	1	4	00-05-5d-61-58-a8	00-03-00-00-00-04	88.188.182.188	0.55.0.0	UDP	0
2557	04.14.3.46.16.574.776	3	1	4	00-05-5d-61-58-a8	00-03-00-00-00-04	88.188.182.188	0.55.0.0	UDP	0
2558	04.14.3.46.16.575.389	2	2	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.188.182.188	0.55.0.0	UDP	0
2559	04.14.3.46.16.575.433	3	2	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.188.182.188	0.55.0.0	UDP	0
2560	04.14.3.46.16.575.514	2	2	2	00-05-5d-61-58-a8	00-03-00-00-00-02	88.188.182.188	0.55.0.0	UDP	0
2561	04.14.3.46.16.575.541	3	2	2	00-05-5d-61-58-a8	00-03-00-00-00-02	88.188.182.188	0.55.0.0	UDP	0

图 5-2 网络分析仪捕捉数据

ChipScope 观测接收端数据帧如图 5-3 所示，其中 rx\_dat\_1 和 rx\_dat\_2 为接收到的来自两个网络的数据帧，rx\_out 为终端系统虚拟链路层接收部分进行完整性检查和冗余处理后的数据帧，接收到的处理数据帧示意图如图 5-4 所示。

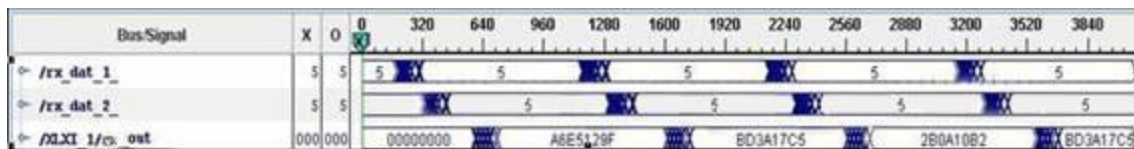


图 5-3 ChipScope 观测数据帧

由于两个网络传输数据帧的时间不同步，接收端接收到的互为备份的数据帧存在一定的时间间隔，经过完整性检查两个网络的数据帧均正常输出，到达冗余管理单元时先到达的数据帧为有效帧输出，后到达的数据帧丢弃，完成终端系统的接收功能。实验结果验证终端系统在正常工作情况下对数据帧进行发送和接收的功能。

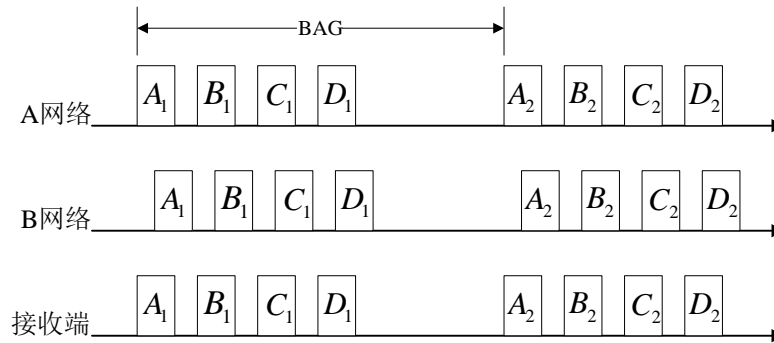


图 5-4 接收数据帧示意图

### 5.3 终端系统冗余功能测试

AFDX 网络中数据传输出现异常时，由冗余容错机制处理数据帧，保证传输的可靠性。实验中发送终端系统发送不同情况的数据帧，观测接收终端系统对数据帧的处理情况，验证时间触发终端系统冗余处理功能。

#### 5.3.1 终端系统错误帧处理测试

为了验证终端系统对错误帧的处理情况，发送终端系统发送的数据帧中含有序列号不连续的错误帧，处理过程示意图如图 5-5 所示。

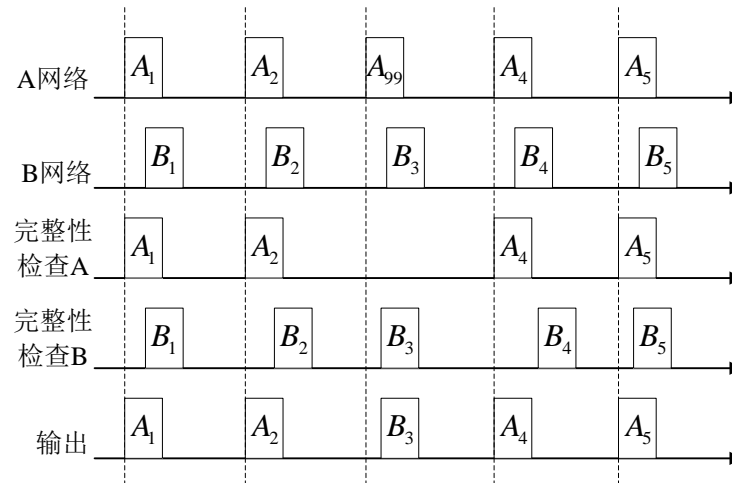


图 5-5 发送错误帧处理示意图

图中 A 网络发送的数据帧有不满足序列号连续规则的帧  $A_{99}$ ，经完整性检查时检查到错误帧并删除，B 网络中的数据帧均为正常情况下的连续数据帧，由于 B 网络数据帧后到达，当作冗余帧被丢弃，但是在应该输出数据帧  $A_3$  的时刻未检测到目标帧，规定时间范围内接受到 B 网络的备份帧  $B_3$ ，则被视为有效

帧输出，以确保通信网络中数据传输的可靠性。AFDX 网络分析仪观察到发送数据帧如图 5-6 所示。

No.	时间(月.天.时.分.秒.毫秒)	TAP端口	SN	VL	源MAC	目的MAC	源IP	目的IP	协议	源端口
1341312	04.22.24.11.46.729.524	1	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341313	04.22.24.11.46.729.559	4	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341314	04.22.24.11.46.729.563	1	2	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341315	04.22.24.11.46.729.584	4	2	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341316	04.22.24.11.46.729.602	1	99	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341317	04.22.24.11.46.729.604	4	3	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341318	04.22.24.11.46.729.642	1	4	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341319	04.22.24.11.46.729.680	4	4	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341320	04.22.24.11.46.729.682	1	5	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341321	04.22.24.11.46.733.524	4	5	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341322	04.22.24.11.46.733.559	1	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341323	04.22.24.11.46.733.563	4	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341324	04.22.24.11.46.733.584	1	2	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
1341325	04.22.24.11.46.733.602	4	2	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0

图 5-6 网络分析仪监测数据帧

实验中设计发送一条虚拟链路上的数据帧，一个网络发送正常数据帧，另一个网络发送含有错误帧的数据帧由通信板接收，接收到的数据帧利用 ChipScope 观察如图 5-7 所示。

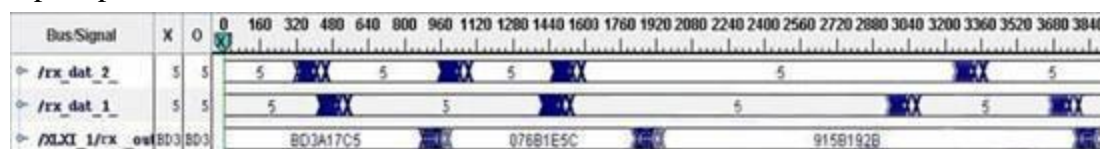


图 5-7 接收数据帧

图中显示经过冗余管理后输出的数据 rx\_out 为序列号连续帧，A 网络中的错误帧由 B 网络当中的数据帧替换，保证最后输出的数据帧是正确的。

### 5.3.2 终端系统复位帧处理测试

虚拟链路层传输的数据帧中 SN 号为 0 的数据帧为复位帧，当发送终端系统复位时发送出的第一个数据帧 SN 为 0。如果发送终端系统一直进行正常数据发送，则在接收到 SN 号为 255 的数据帧后下一个数据帧的 SN 号应为 1。终端系统处理复位帧的过程如图 5-8 所示。

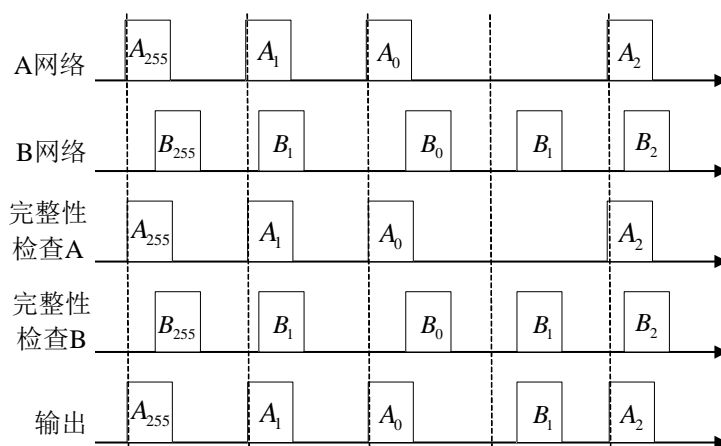


图 5-8 处理复位帧示意图

网络分析仪检测两个链路发送数据帧如图 5-9 所示。图中显示两个网络发送的数据帧与图 5-8 中数据帧出现情况相同。

No.	时间(月.天.时.分.秒.毫秒)	TAP端口	SN	VL	源MAC	目的MAC	源IP	目的IP	协议	源端口
724	04.25.29.21.18.595.347	2	255	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
725	04.25.29.21.18.595.363	3	255	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
726	04.25.29.21.18.595.367	2	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
727	04.25.29.21.18.595.408	3	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
728	04.25.29.21.18.595.426	2	0	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
729	04.25.29.21.18.595.428	3	0	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
730	04.25.29.21.18.595.466	2	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
731	04.25.29.21.18.595.504	3	2	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
732	04.25.29.21.18.595.506	2	2	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
733	04.25.29.21.18.593.347	2	255	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
734	04.25.29.21.18.593.363	3	255	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
735	04.25.29.21.18.593.367	2	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
736	04.25.29.21.18.593.408	3	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
737	04.25.29.21.18.593.426	2	0	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
738	04.25.29.21.18.593.428	3	0	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
739	04.25.29.21.18.593.466	2	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0
740	04.25.29.21.18.593.504	3	2	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0.0	UDP	0

图 5-9 网络分析仪采集数据

ChipScope 观察接收到的数据帧如图 5-10 所示，虚拟链路上 SN 号为 0 的数据帧为系统复位后发送的第一帧，第二帧的 SN 号为 1。不存在复位帧时 SN 范围为 1~255，发送数据帧中有 SN 号不连续情况，完整性检查时删除错误帧。经过冗余管理输出的数据帧为 rx\_out，删除错误帧后接收到复位帧，数据帧的序列号由 1~255 循环。实验结果验证终端系统对接收到的数据帧完整性检查满足序列号连续递增的变化规则，以及存在序列号为 0 时系统复位重新正确发送数据帧。

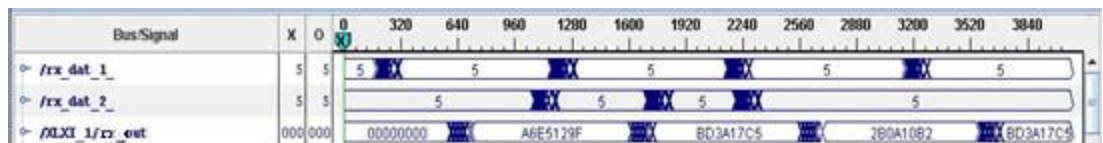


图 5-10 接收数据帧

### 5.3.3 终端系统丢帧情况测试

验证终端系统对数据帧丢失情况的处理，实验中数据帧传输情况如图 5-11 所示。

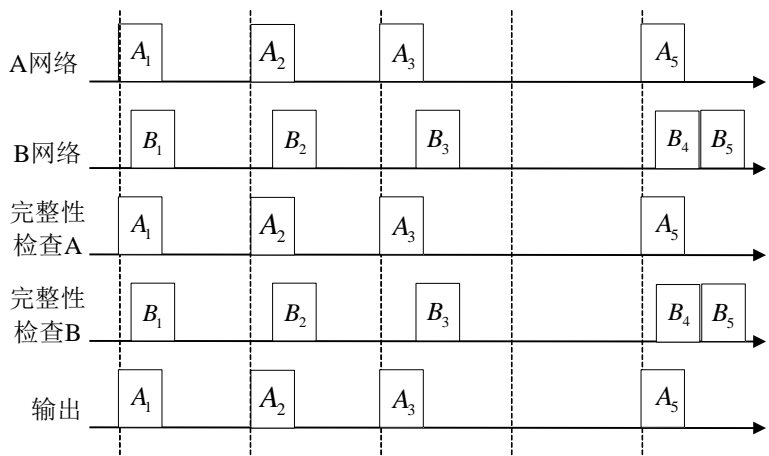


图 5-11 数据帧传输示意图



No.	时间(月.天.时.分.秒.毫秒)	TAP端口	SN	VL	源MAC	目的MAC	源IP	目的IP	协议	源端口
726	04/23/26:45:16.592.374	2	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
727	04/23/26:45:16.592.410	4	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
728	04/23/26:45:16.592.414	2	2	2	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
729	04/23/26:45:16.592.435	2	3	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
730	04/23/26:45:16.592.453	2	3	2	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
731	04/23/26:45:16.592.495	4	3	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
732	04/23/26:45:16.592.516	4	4	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
733	04/23/26:45:16.592.531	2	5	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
734	04/23/26:45:16.592.533	4	5	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
735	04/23/26:45:16.596.374	2	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
736	04/23/26:45:16.596.410	4	1	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
737	04/23/26:45:16.596.414	2	2	2	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
738	04/23/26:45:16.596.435	4	2	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
739	04/23/26:45:16.596.435	2	3	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0
740	04/23/26:45:16.596.455	4	3	1	00-05-5d-61-58-a8	00-03-00-00-00-01	88.168.192.168	0.55.0	UDP	0

接收终端系统接收数据帧后经过完整性检查和冗余管理输出数据帧结果如图 5-13 所示。

Bus:Signal	X	O	0	160	320	480	640	800	960	1120	1280	1440	1600	1760	1920	2080	2240	2400	2560	2720	2880	3040	3200	3360	3520	3680	3840	4000	
/rx_dat_2	5	5		5				5				5					5												
/rx_dat_1	5	5		5				5				5					5												
/ALXI 1/rx outB03	B03	B03						BD3A17C5				076B1E5C								915B192B									

数据帧传输时经过完整性检查，由冗余管理模块删除后到达的无效帧，最后输出结果 rx out 丢失一帧，与示意图 5-11 中情况相同。

## 5.4 本章小结

- 54 -

## 结 论

通过比较 AFDX 终端系统常用的调度策略，分析保证确定性方面存在的不足，本文将具有严格确定性的时间触发通信机制引入到 AFDX 终端系统的虚拟链路层中，设计时间触发调度策略并进行通信网络的确定性分析，实现基于 FPGA 的时间触发终端系统虚拟链路层设计，构建测试平台验证通信性能，完成以下工作：

(1) 对 AFDX 终端系统虚拟链路层协议栈分析并做出改进，引入时间触发调度层，详细设计时间触发调度算法，阐述调度过程并分析在时间触发调度策略下 AFDX 通信网络确定性能有很大改善，通过构建 AFDX 网络模型理论计算说明。

(2) 选取目前常用调度策略中保证 AFDX 网络实时性和确定性能方面具有优势的 SP 调度策略进行分析，利用网络演算法进行理论计算。在网络仿真平台 MATLAB/TrueTime 下，分别对时间触发调度和静态优先级调度策略在同一网络模型中进行仿真，比较理论计算值与仿真结果，充分说明时间触发调度算法的良好确定性能。

(3) 设计时间触发 AFDX 终端系统的虚拟链路层 FPGA 实现模块，根据协议中做出的规定，总体结构分为 MAC 层模块、发送模块和接收模块。其中 MAC 层模块与普通以太网功能相符，不做详细介绍，重点设计发送模块与接收模块。发送模块完成流量整形、时间触发调度和冗余管理功能，接收模块完成完整性检查和冗余处理功能，对发送和接收模块中各个功能单元详细设计。将 MAC 层模块与发送模块和接收模块联调，实现终端系统虚拟链路层的功能。

(4) 构建系统测试实验平台，分别在正常情况、发送数据帧中含有错误帧、复位帧以及丢帧情况下进行了测试，测试结果表明终端系统确定实时发送数据帧，对接收到的数据帧冗余容错处理，验证终端系统收发数据的良好性能。

本论文尚存在有待改进之处：

(1) 可进一步研究时间触发调度策略进行优化，保证规定调度时间的严格确定性和总线利用率的提高。

(2) 优化 FPGA 程序代码，减少 FPGA 资源利用率。

## 参考文献

- [1] 唐宁, 常青.航空数据总线技术分析研究[J].现代电子技术, 2014, 37(4):64-69.
- [2] 杨峰, 田泽, 夏大鹏, 等.AFDX 网络端系统确定性关键技术研究[J].计算机技术与发展, 2013(9).
- [3] 校莉.AFDX 在航空通信系统中的应用[J].电讯技术, 2010, 50(7):40-43.
- [4] 杨佳谔, 吴成富, 段晓军.时间触发架构总线分析[J].航空计算技术, 2013, 43(4):125-128.
- [5] 冯晓旺, 蓝海文.新一代航空航天总线技术[J].航空制造技术, 2012(3):98-99.
- [6] 彭菲.航空电子系统技术发展趋势研究[J].科技创新与应用, 2013(3):73.
- [7] 朱晓飞, 黄永葵.综合模块化航空电子系统标准分析及发展展望[J].航空电子技术, 2010, 41(4):17-22.
- [8] 王岩, 孙增圻.网络控制系统分析与设计[M].北京:清华大学出版社, 2009:52-53.
- [9] 刘士全, 隼扬, 蔡洁明, 等.1553B 总线应用发展研究[J].电子与封装, 2013, 13(12):12-15.
- [10] Lan Moir, Allan Seabridge. Civil Avionics Systems[M]. 范秋丽, 等译. 北京:航空工业出版社, 2009: 9-11.
- [11] Cary R.Spitzer.Digital Avionics Handbook[M].谢文涛, 等译.北京:航空工业出版社, 2010:3-8.
- [12] 夏志飞.大型飞机航电系统总线研究[J].航空维修与工程, 2012(6):58-60.
- [13] 马萌.航空专用数据总线研究[J].数字技术与应用, 2013(10):61-63.
- [14] 符富强.航空电子系统数据网络的要求与全双工交换以太网应用[J].科学之友, 2012(9):150-151.
- [15] 赵永库, 李贞, 唐来胜.AFDX 网络协议研究[J].计算机测量与控制, 2012, 20(1):8-11.
- [16] Hussein Charara, Jean-Luc Scharbarg, J.Ermont, et al. Methods for bounding end-to-end delays on an AFDX network[C]. ECRTS, 2006:18.
- [17] 李飞.AFDX 交换机调度算法的实现与验证[D].西安电子科技大学, 2013.
- [18] 贾卫松, 翟正军, 牛仕奇.AFDX 端系统设计中的发送调度方法研究与实现[J].计算机测量与控制, 2010, 18(11):2612-2615.
- [19] Cummings Rodney, Richter Kai, Ernst Rolf, et al. Exploring Use of Ethernet



- for In-vehicle Control Applications: AFDX, TTEthernet, EtherCAT, and AVB[C]. SAE 2012 World Congress and Exhibition, 2012, 5(1):72-88.
- [20] Li QingRu, Wang Hongchun, and Han Wei. Real-time Fault-tolerant Ethernet Technology-TTEthernet and Its Feature[C]. 2013 International Conference on Mechatronics and Industrial Informatics, 2013:2711-2714.
- [21] 刘帅, 张喜民, 郭鹏.TTE 通信技术在混合安全关键系统的应用[J].航空计算技术, 2013, 43(2):120-123.
- [22] 朱闻渊, 尹家伟, 蒋祺明.新型航空电子系统总线互连技术发展综述[J].计算机工程, 2011, 37[增刊]:398-402.
- [23] Li Xie. Research on the Real-time Property of TTCAN Protocol in Vehicle Communication Network[C]. 2013 International Conference on Mechanical Structures and Smart and Smart Materials, 2013:674-677.
- [24] 蒋菲怡, 吴勇, 张建东.AFDX 性能分析与调度算法[J].测控技术, 2013, 32(6):100-104.
- [25] 刘芸, 王红春, 王兵.AFDX 端系统协议软件设计与实现[J].电光与控制, 2012, 19(11):76-71.
- [26] 施太平, 姜莉, 田泽.AFDX 协议及关键技术的实现[J].测控技术, 2010, 31(10):81-84.
- [27] 任向隆, 马捷中, 翟正军.基于 FPGA 的 AFDX 端系统协议栈虚链路层的研究与实现[J].计算机测量与控制, 2010, 18(1):157-160.
- [28] 楼晓强, 田泽.AFDX 网络终端系统的确定性分析与实现[J].计算机技术与发展, 2010, 20(8):56-59.
- [29] Zhang Xueyuan, Wang Yonggang. Research of AFDX Network Delay Based on Modified Network Calculus[C]. IEEE International Conference on Network Infrastructure and Digital Content, 2012:178-181.
- [30] Ahlem Mifdaoui, Fabrice Frances, Christian Fraboul. Full Duplex Switched Ethernet for Next Generation 1553B-based Applications[C]. Proceedings of the 13<sup>th</sup> IEEE Real Time and Embedded Technology and Applications Symposium, 2007:45-56.
- [31] Wilfried Steiner, Gunther Bauer, Brendan Hall, et al. TTEthernet Dataflow Concept[C]. Eighth IEEE International Symposium on Network Computing and Applications, 2009:319-322.
- [32] Ekarin Suethanuwong. Scheduling Time-Triggered Traffic in TTEthernet Systems[C]. Proceeding of 2012 IEEE 17<sup>th</sup> International Conference on Emerging Technologies and Factory Automation, 2012.
- [33] 任向隆, 马捷中.航空电子全双工交换式以太网终端系统研究[J].计算机测量

- 与控制, 2009, 17(5):924-927.
- [34] 刘冬冬, 张天宏, 陈建, 等. TTP/C 协议的关键特性研究[J]. 计算机测量与控制, 2012, 20(10):2769-2772.
- [35] 刘晚春, 李峭, 何锋, 等. 时间触发以太网同步及调度机制的研究[J]. 航空计算技术, 2011, 41(4):122-127.
- [36] Liu Cheng, Wang Tong, Li Zheng, et al. Design and Delay Analysis of Time-triggered AFDX Network[J]. Journal of Beijing University of Aeronautics and Astronautics, 2013, 39(6):728-733.
- [37] Zhou Tianran, Zhang Yongtao, He Feng, et al. Schedulability Analysis for AFDX-based Avionics System[J]. Journal of Beijing University of Aeronautics and Astronautics, 2012, 38(1):75-79.
- [38] Melhem Tawk, Guchuan Zhu, Yvon Savaria, et al. A Tight End-to-End Delay Bound and Scheduling Optimization of an Avionics AFDX Network[C]. IEEE Digital Avionics System Conference, 2011:1-10.
- [39] Chen Xin, Zhou Yong-Jun, Jiang Wen-Bao, et al. Performance Analysis of AFDX Protocol and Scheduling Algorithm[J]. Tien Tzu Hsueh Pao/Acta Electronica Sinica, 2009, 37(5): 1000-1005.
- [40] ARINC Specification 664 P7-1[S]. AEEC, 2009.
- [41] H. Bauer, J.L. Scharbarg, and C. Fraboul. Worst-case end-to-end delay analysis of an avionics AFDX network[C]. Design, Automation and Test in Europe, 2010:1220-1224.
- [42] 蒋文婷, 高月红, 王广德, 等. 优先级网络性能的网络演算分析[J]. 无线电通信技术, 2013, 39(1):8-12.
- [43] Ren Yazhou, Hu Fei, and Li Jian. End to end jitter control on AFDX network[C]. IEEE International Conference on Transportation, Mechanical, and Electrical Engineering, 2011:515-518.
- [44] Henri Bauer, Jean-Luc Scharbarg, and Christian Fraboul. Improving the Worst-Case Delay Analysis of an AFDX Network Using an Optimized Trajectory Approach[J]. IEEE Transactions on Industrial Informatics, 2010, 6(4):521-533.
- [45] Xueyuan Zhang, Yonggang Wang. Research of AFDX Network Delay Based on Modified Network Calculus[C]. 2012 3<sup>rd</sup> IEEE International Conference on Network Infrastructure and Digital Content, 2012:178-181.
- [46] 周又玲. MATLAB 在电气信息类专业中的应用[M]. 北京:清华大学出版社, 2011:274-277.
- [47] 邓睿, 汤贤铭, 俞金寿. 基于 TrueTime 工具箱的网络控制系统时延分析[J].

- 工业控制计算机, 2010, 23(2):24-26.
- [48] 周艳, 陈永建.基于 TrueTime 的网络控制系统仿真平台的构建及其仿真分析[J].北京联合大学学报, 2012, 26(4):18-22.
- [49] Wang Y, Heb L. Analysis and Simulation of Networked Control Systems Delay Characteristics Based on Truetime[J]. Computer Modeling and New Technologies, 2013, 17(4):210-216.
- [50] Jean-Luc Scharbarg, Frederic Ridouard, and Christian Fraboul. A Probabilistic Analysis of End-To-End Delays on an AFDX Avionic Network[J]. IEEE Transactions on Industrial Informatics, 2009, 5(1):38-49.
- [51] Ahmad Al Sheikh, Olivier Brun, Maxime Cheramy, et al. Optimal Design of Virtual Links in AFDX Networks[J]. Real Time Syst, 2013(49):308-336.
- [52] 任向隆, 马捷中, 翟正军.基于 FPGA 的 AFDX 端系统协议栈虚链路层的研究与实现[J].计算机测量与控制, 2010, 18(1):157-160.
- [53] 王时雨, 贺占庄, 张盛兵.基于 FPGA 的以太网控制器的实现[J].微电子学与计算机, 2012, 29(2):171-175.
- [54] 丁世勇, 谭文文, 李桂英.基于 FPGA 的以太网 MAC 控制器的设计与实现[J].电子设计工程, 2011, 19(21):163-165.
- [55] 杨绍辉.AFDX 交换芯片中 MAC IP 核的设计与实现[D].西安电子科技大学, 2009.
- [56] 任向隆, 马捷中.航空电子全双工交换式以太网终端系统研究[J].计算机测量与控制, 2009,17(5):924-927.
- [57] 李哲, 田泽, 张荣华. AFDX 网络中 SkewMax 的研究[J]. 计算机技术与发展, 2010, 20(6): 249-253.

## 攻读硕士学位期间发表的论文及其它成果

- [1] 刘晓胜, 李莹雪, 张鹏宇, 郑检, 吴海涛. 基于虚拟时钟的 AFDX 调度算法研究及其延时计算[J]. 电气传动, 2013, 10[增刊]:119-122.

## 哈尔滨工业大学学位论文原创性声明及使用授权说明

### 学位论文原创性声明

本人郑重声明：此处所提交的学位论文《时间触发 AFDX 终端系统的虚拟链路层调度算法研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：李荣霄 日期：2014 年 7 月 7 日

### 学位论文使用授权说明

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

(1) 学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；(2) 学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；(3) 研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。  
本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：李荣霄 日期：2014 年 7 月 7 日  
导师签名：吴峰 日期：2014 年 7 月 7 日

## 致 谢

两年的研究生生活即将画上句号，走的辛苦却也收获满囊。有过困难无法解决时的困惑与烦躁，也有过问题迎刃而解时的欣喜与欢乐。感谢导师吴辉老师的支持和鼓励，刘晓胜教授的谆谆教导与悉心照顾，先生严谨进取的治学精神和豁达乐观的人生态度常常能够指引我走出低谷，顺利解决问题，并学到分析和解决问题的正确方法。从课题的选择到最终完成，刘老师都给予我细心的指导和不懈的支持

感谢在论文撰写过程中，为我提供帮助与支持的实验室博士研究生张良、庞继伟师兄以及刘博师姐，祝你们在各自的研究领域内硕果累累，为实验室多做贡献。

感谢已经毕业的硕士研究生张鹏宇、任会芬和赵振峰的帮助与鼓励，祝你们鹏程万里、前程似锦。

感谢同学任双雪和宋梦，在两年的生活中她们给予我鼓励和支持，陪我走过了愉快的两年。

特别感谢我的父母，感谢你们的辛苦培养，为我提供良好的学习和生活环境，感谢你们对我永远的支持与鼓励，祝你们健康长寿、幸福快乐。

由衷感谢对本论文提出宝贵意见的评审老师和答辩组老师，你们的意见和建议对本论文的修改与完善工作意义重大，谢谢你们。

最后，感谢所有关心、支持和帮助本论文撰写的老师和朋友！