

# 中国科学技术大学

# 博士学位论文



## 实时以太网 POWERLINK 在加速器控 制系统中的应用研究

作者姓名： 孙晓康

学科专业： 核科学与技术

导师姓名： 刘功发教授级高工

完成时间： 二〇二〇年五月二十八日



University of Science and Technology of China  
A dissertation for doctor's degree



**Application and Research of  
Ethernet POWERLINK in Accelerator  
Control System**

Author: Xiaokang Sun

Speciality: Nuclear Science and Technology

Supervisor: Prof. Gongfa Liu

Finished time: May 28, 2020



## 中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名：\_\_\_\_\_

签字日期：\_\_\_\_\_

## 中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入《中国学位论文全文数据库》等有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

保密的学位论文在解密后也遵守此规定。

公开  保密（\_\_\_\_ 年）

作者签名：\_\_\_\_\_

导师签名：\_\_\_\_\_

签字日期：\_\_\_\_\_

签字日期：\_\_\_\_\_



## 目 录

第 1 章 绪论 ······	1
1.1 课题研究背景 ······	1
1.1.1 加速器控制系统简介 ······	1
1.1.2 实时性分类 ······	3
1.1.3 加速器控制系统中的实时性需求 ······	4
1.2 国内外研究现状 ······	5
1.2.1 基于 POWERLINK 的 ALBA 设备保护系统 ······	5
1.2.2 CERN 在辐射区域关于 POWERLINK 的应用研究 ······	6
1.2.3 上海光源的光束线前端真空泄漏快保护系统 ······	8
1.3 论文工作的主要内容及创新点 ······	8
第 2 章 POWERLINK 通信协议研究 ······	10
2.1 实时以太网介绍 ······	10
2.2 POWERLINK 协议介绍 ······	12
2.2.1 POWERLINK 协议的基本特性 ······	12
2.2.2 POWERLINK 协议的网络模型 ······	14
2.3 POWERLINK 协议的实现 ······	17
2.3.1 基于 Linux 系统实现 POWERLINK 协议 ······	19
2.3.2 基于 FPGA 实现 POWERLINK 协议 ······	20
2.3.3 测试小结 ······	26
2.4 POWERLINK 通信周期的理论计算 ······	27
2.5 POWERLINK 通信协议的仿真模拟 ······	29
2.5.1 OMNeT++ 仿真器 ······	29
2.5.2 POWERLINK 通信节点建模 ······	31
第 3 章 EPICS 环境下基于 POWERLINK 的分布式 IO 系统 ······	35
3.1 主站 PC 方案的系统设计与开发 ······	35
3.1.1 系统架构设计 ······	35
3.1.2 EPICS 设备驱动程序的开发 ······	35
3.1.3 从站控制器的设计与开发 ······	37
3.1.4 测试系统搭建 ······	43
3.1.5 系统性能测试与分析 ······	45

## 目 录

---

3.2 全站 FPGA 方案的系统设计与开发 ······	49
3.2.1 系统架构设计 ······	50
3.2.2 从站控制器的设计与开发 ······	50
3.2.3 EPICS 设备驱动程序的开发 ······	53
3.2.4 测试系统搭建 ······	54
3.2.5 系统性能测试与分析 ······	55
3.3 全站 FPGA 方案通信周期的理论计算 ······	59
3.4 全站 FPGA 方案的仿真模拟 ······	62
<b>第 4 章 HALF 设备保护系统的设计 ······</b>	<b>64</b>
4.1 HALF 预研工程 ······	64
4.2 加速器中的设备保护系统 ······	65
4.2.1 设备保护系统的任务 ······	65
4.2.2 国内外加速器的机器保护系统调研 ······	66
4.3 HALF 设备保护系统设计 ······	70
4.3.1 HALF 设备保护系统任务 ······	70
4.3.2 HALF 设备保护系统设计原则 ······	71
4.3.3 HALF 设备保护系统运行模式 ······	71
4.3.4 HALF 设备保护系统总体结构 ······	71
4.3.5 联锁输入信号的预处理 ······	73
4.4 注入器 EPS 设计 ······	74
4.4.1 电子枪联锁系统 ······	76
4.4.2 真空联锁系统 ······	77
4.4.3 冷却水联锁系统 ······	79
4.4.4 注入器分总体 EPS 联锁信号总结 ······	79
4.4.5 注入器设备保护系统实时性能评估 ······	80
4.5 储存环分总体 EPS 设计 ······	84
4.5.1 真空联锁系统 ······	85
4.5.2 冷却水联锁系统 ······	87
4.5.3 真空部件温度联锁系统 ······	88
4.5.4 高频联锁系统 ······	89
4.5.5 注入联锁系统 ······	91
4.5.6 储存环分总体 EPS 联锁信号总结 ······	92
4.5.7 储存环设备保护系统实时性能评估 ······	92
4.6 HALF 设备保护系统的信息报警 ······	97

## 目 录

---

4.7 HALF 设备保护系统的数据存档与查询 ······	98
第 5 章 总结与展望 ······	100
5.1 总结 ······	100
5.2 展望 ······	101
参考文献 ······	103
补充材料 ······	108
致谢 ······	109
在读期间发表的学术论文与取得的研究成果 ······	110



# 第1章 绪论

## 1.1 课题研究背景

### 1.1.1 加速器控制系统简介

粒子加速器是利用电磁场不断加速真空环境下的带电粒子，使之达到很高能量的装置。粒子加速器广泛应用于基础科学、生产生活、医疗卫生、科学技术等各个领域<sup>[1-2]</sup>。其中，同步辐射光源是粒子加速器在科研领域的重要应用之一。同步辐射是速度接近光速的带电粒子在电磁场的作用下发生弯转时，沿弯转轨道的切线方向发出的电磁辐射，最早是在电子同步加速器上发现的。同步辐射光源是指产生同步辐射的物理装置，它的主要设备包括注入器和电子储存环，注入器产生电子束并将其加速到所需能量，电子储存环使得电子发生弯转并产生同步辐射光。同步辐射光源具有亮度高、强波长范围宽且连续可调、方向性及偏振性好、超纯净等优异特性，广泛应用于物理、化学、材料科学、生命科学、环境科学、微电子等众多基础研究和应用研究领域。合肥光源（Hefei Light Source, HLS）由中国科学技术大学国家同步辐射实验室承建，它是我国第一台以真空紫外和软X射线为主的专用同步辐射光源，在1989年建成出光。合肥光源的主体设备是一台能量为800MeV、平均流强为360mA的电子储存环。在2009至2014年间，合肥光源进行了重大维修改造，改造后的合肥光源称为HLS-II，HLS-II的储存环束流发射度显著降低，装置稳定性改善，接近三代同步辐射光源的水平<sup>[2]</sup>。为了建设基于衍射极限储存环的第四代同步辐射光源，国家同步辐射实验室提出了合肥先进光源（Hefei Advanced Light Source, HALS）的构想并开展了HALS预研工程，HALS于2019年下半年开始改称为Hefei Advanced Light Facility (HALF)，HALF是一个位于中低能区，覆盖中红外至软X射线波段，具有高亮度、低发射度、全空间相干的最先进的衍射极限光源<sup>[3]</sup>。

加速器是复杂系统的精密集成，通常加速器会包括电子枪、微波加速、磁铁偏转与聚焦、束流测量、束流注入、束流反馈、高频加速、插入元件等多个系统，系统之间协调工作，才能保证加速器的正常工作，这是加速器控制系统首先要实现的功能。加速器运行期间，需要密切监测各个系统的工作状态，进行必要的联锁和保护，才能够保证加速器安全、稳定和高效的运行。加速器运行状态监测和联锁保护，也是加速器控制系统必需的功能。综上所述，加速器控制系统承担着加速器“大脑”的功能，直接指挥着加速器的调试、运行。一般来说，粒子加速器控制系统的具体任务如下<sup>[4-5]</sup>：

- (1) 对电子枪、磁铁电源、真空、高频、微波、束调管、注入、冷却水、插

- 入件以及束测等系统的设备进行监测和控制；
- (2) 建立报警和联锁保护机制，确保操作人员和设备的安全；
  - (3) 提供定时及同步系统，实现相关设备间协调工作及信号同步；
  - (4) 建立上层物理应用软件，实现轨道校正、BBA、线性参数测量与校正、响应矩阵测量、局部凸轨、Lattice 标定、束流光学参数测量及校正、插入，元件补偿、装置状态和参数分析等功能；
  - (5) 建立数据存档和查询系统，以网页方式提供历史数据查询与分析等功能。

粒子加速器往往体积庞大、设备众多，其控制系统一般是基于网络的分布式控制系统，目前国内外应用最为广泛的控制系统开发工具是 EPICS。EPICS 是实验物理和工业控制系统 (Experimental Physics and Industrial Control System) 的缩写，是 20 世纪 90 年代初发展起来的大型分布式控制系统的软件运行环境和开发平台，已成功应用于全世界近百个大科学装置，涉及大型粒子加速器、高能物理探测器、大型天文望远镜等多种类型装置<sup>[6]</sup>。国际上，European XFEL、NSLS-II、ITER、LCLS-II、FRIB、ESS、ILC 等工程项目都采用 EPICS 作为控制系统的软件开发平台<sup>[7-13]</sup>。在国内，EPICS 已成功应用于近年来相继完成的 HLS-II、北京正负电子对撞机重大改造项目 (BEPC-II)、上海同步辐射光源 (SSRF)、中国散裂中子源 (CSNS) 等大型科学工程项目中<sup>[14-17]</sup>，目前正在建设中的上海硬 X 射线自由电子激光装置 (SHINE) 和高能同步辐射光源 (HEPS) 等项目也采用 EPICS 作为控制系统的软件开发平台<sup>[18-19]</sup>。基于 EPICS 的控制系统是基于网络的分布式控制系统，其架构如图 1.1 所示，遵循所谓的“标准模型” (Standard Models)，从上到下分为三层，分别为管理层 (Supervisory controls)、前端控制层 (Front End Controls) 和设备控制层 (Devices Controls)<sup>[20]</sup>。

#### • 管理层 (Supervisory controls)

位于系统最上层是管理层，由 OPI 和服务器系统组成。其中，OPI 给操作人员提供了交互界面，操作人员通过 OPI 监控系统状态，发送控制命令。服务器系统包括数据库、文件服务器等，以提供系统初始化配置、历史数据存档和查询等功能。

#### • 前端控制层 (Front End Controls)

前端控制层在 EPICS 控制系统中处于关键位置，它与管理层构成客户端/服务器 (Client/Server) 模式，前端控制层为服务端，管理层为客户端，二者之间通过 Channel Access (CA) 通信协议进行通信。前端控制器（也被称为 IOC）负责输入输出控制、保存动态数据和控制算法，是控制任务的主要执行者。同时，通过 CA 协议将相关信息发布至管理层。

#### • 设备控制层 (Devices Controls)

位于系统最底层的是设备控制层，是 EPICS 控制系统的被控对象。被控设

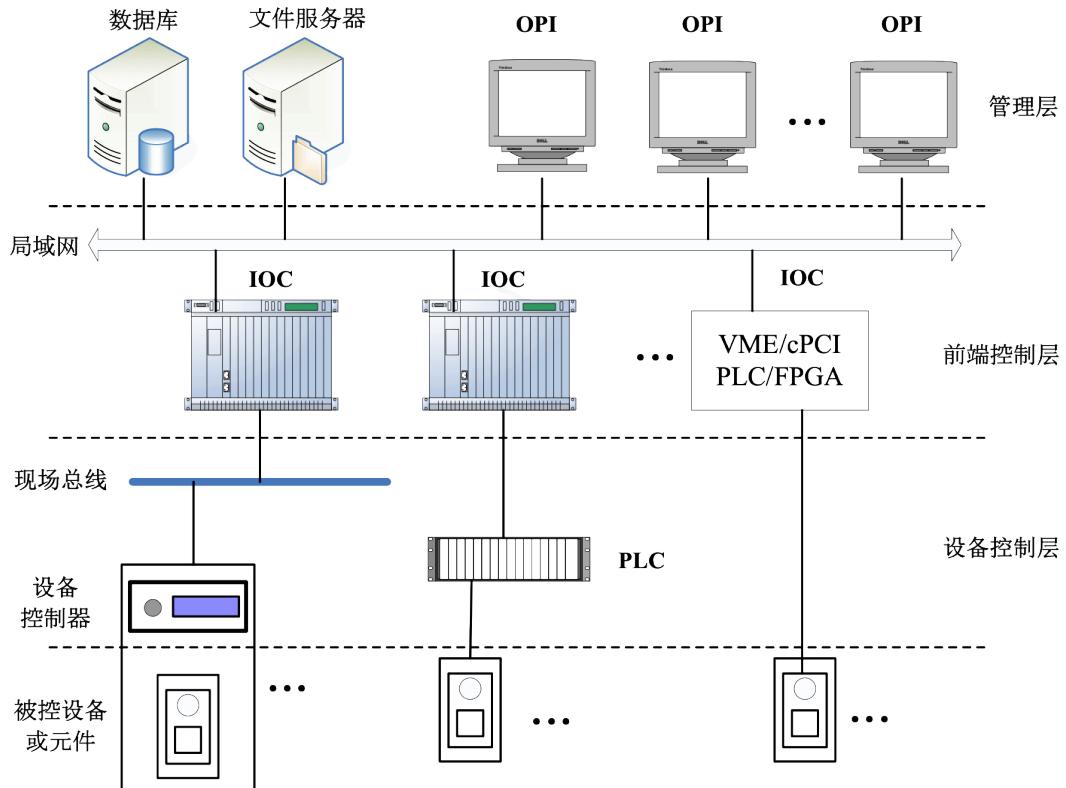


图 1.1 基于 EPICS 的控制系统基本结构图

备种类多样，有嵌入式智能设备控制器，负责控制功能复杂的设备，它们可以通过现场总线、以太网与 IOC 通信；也可以是 PLC，用来对下层设备进行过程控制的状态监测，PLC 可以通过网络或串口的方式与 IOC 通信。除此之外，IOC 也可以通过 IO 模块直接对设备和元件进行控制。

### 1.1.2 实时性分类

实时性的含义是一个事件发生后，在确定时间内系统做出反应的能力。对于工业自动化系统来说，实时性是工业控制网络的基本要求，根据不同的应用场合，将实时性要求划分为三个范围，分别为：

- (1) 信息集成和较低要求的过程自动化应用场合，实时响应时间的要求是 100ms 或更长；
- (2) 绝大多数的工厂自动化应用场合，实时响应时间的要求最少为 5~10ms；
- (3) 对于高性能的同步运动控制应用，特别是在 100 个节点下的伺服运动控制应用场合，实时响应时间要求低于 1ms，同步传送和抖动时间小于  $1\mu s$ <sup>[21]</sup>。

工业以太网是指用于工业自动化环境，符合 IEEE 802.3 标准，按照 IEEE 802.1D 和 IEEE 802.1Q 规范，并对其进行实时性扩展而实现的以太网。工

业以太网的实时响应时间可以达到 5~10ms，相当于现有的现场总线。对于一些实时性能要求更高的应用场合，各标准组织和公司基于工业以太网提出了各种提升实时性能的技术方案。这些方案都在 IEEE 802.3 标准的基础上对其进行实时性扩展从而提高实时性能，这就是实时以太网（Real Time Ethernet, RTE）。

Ethernet POWERLINK 是一种完全开源的实时以太网技术，最初由奥地利 B&R 公司于 2001 年 11 月创立和开发，自 2003 年以来，由独立的用户组织 EPSG 负责该技术的进一步发展。POWERLINK 完全建立在标准以太网（IEEE802.3）的基础之上，采用纯软件方式实现的协议，可达到硬实时的性能。EPSG 在 2008 年发布了该技术的开源版本 openPOWERLINK<sup>[22]</sup>。

### 1.1.3 加速器控制系统中的实时性需求

数据通信在粒子加速器控制系统中起着纽带的作用。随着加速器装置规模的增大和复杂度的提高，对数据通信性能的要求越来越高，而实时性是影响系统性能的关键因素，开展这方面的应用研究具有非常重要的工程应用价值。

对于加速器控制系统来说，大部分系统对实时性的要求并不高，相当于工业自动化系统的第一类场合，实时响应时间的要求是 100ms 或更长，如恒温水系统的监控、真空状态的监测、慢速联锁保护等；对实时性要求较高的系统则与工业自动化系统第三类场合的要求类似，如快速联锁保护系统和电子储存环轨道快反馈系统（Fast Orbit Feedback, FOBB）等。

慢速联锁保护系统的实时响应时间要求是 100ms 或更长，实现方式大都采用 PLC 作为控制器，PLC 间的通信采用标准以太网或实时以太网的软实时模式，如 FERMI@Elettra FEL Facility、TPS、ESS<sup>[23-25]</sup>。

快速联锁保护系统的实时性要求在 ms 量级，一般用于束流丢失的机器保护或真空泄露的快保护等，其实现方式因应用场合的不同而有所差异。例如，上海光源的光束线前端真空泄露快保护系统采用基于 FPGA 和 ARM 的控制器，控制器间的通信参照实时以太网 POWERLINK 的通信方式来自主设计；西班牙 ALBA 的设备快速联锁系统控制器采用贝加莱公司（B&R）的 PLC，PLC 间的通信采用实时以太网 POWERLINK；美国 NSLS-II 的设备快速联锁系统控制器采用 Allen-Bradley 公司的 PLC，PLC 间的通信采用实时以太网 Ethernet/IP；上海光源的束流丢失保护系统采用集中控制方式，将所有 140 个 BPM 信号汇总到 NI PXI 的 FPGA 板卡中，以提高系统的实时性；西班牙 ALBA 的束流丢失保护系统采用了专门设计的技术，将 MRF 定时技术的单向数据传输改进为双向数据传输，以实现高的实时性。

电子储存环轨道快反馈频率达 KHz 量级，反馈周期为 ms 量级，不同的装置所采用的方法差异较大。例如，上海光源轨道快反馈系统采用反射内存方式

在 10 个站点间共享数据，每个站点采用 2 台 VME 计算机，分别用作轨道数据的采集和反馈校正<sup>[26]</sup>；英国 Diamond 轨道快反馈系统采用 RocketIO 来构建数据传输网络，通信速度是 2.12Gbps<sup>[27]</sup>；美国 NSLS-II 轨道快反馈系统采用自主研发的串行通信接口 SDI 在 30 个计算节点间传输数据，系统快反馈频率可以达到 10KHz<sup>[28]</sup>。

从上述的解决加速器装置控制中高实时性问题的技术路线来看，有的采用自主设计的专用通信方案，有的采用实时以太网等通用技术。一般来说，通用技术较专用技术具有更好的开放性和灵活性，并可以节省产品的开发成本，缩短开发周期。因此，应该在粒子加速器控制领域中加强对实时以太网技术的研究和应用。

Ethernet POWERLINK 作为一种开源的实时以太网技术，其开源性对加速器装置控制系统来说具有特别重要的意义。在加速器装置控制技术的发展过程中，开放和共享一直是其发展的方向，现在流行的控制系统开发平台无一例外都是开源的，如 EPICS、TANGO、DOOCS 等。

## 1.2 国内外研究现状

POWERLINK 已在工业控制领域得到了广泛的研究和应用<sup>[29-34]</sup>，目前在加速器控制领域，与 POWERLINK 相关的研究和应用还很少。国际上只有西班牙 ALBA 将 POWERLINK 用于设备保护系统，国内只有上海光源在开发光束线前端真空泄露快保护系统时，控制器间的通信设计参考了 POWERLINK 的通信方式。而在应用最广泛的 EPICS 系统中，目前还未见与 POWERLINK 相关的文献。

### 1.2.1 基于 POWERLINK 的 ALBA 设备保护系统

ALBA 是一台三代同步辐射光源，建造于西班牙的巴塞罗那，于 2012 年建成投入使用。ALBA 由直线加速器、满能量增强器和储存环组成，其中电子储存环的能量为 3GeV，周长为 270 米，直线加速器产生并加速电子束流至 100MeV，低发射度的增强器加速电子束流至满能量 3Gev。ALBA 目前拥有八条光束线，包括软 X 射线和硬 X 射线，主要用于生物科学、凝聚态物质、纳米科学和材料科学的研究。ALBA 的设备保护系统（Equipment Protection System, EPS）的任务是避免装置组件受到损害。EPS 保护范围广，不仅涵盖了 ALBA 加速器及其光束线，还有例如射频，光学，真空等实验室。除此之外，EPS 还备份了快速联锁系统的所有信号。目前，EPS 共管理数千个联锁信号，其中对数百个联锁信号需要提供毫秒量级的保护动作。

ALBA 的 EPS 采用 B&R 公司的 X20CP1484 型号 PLC 作为控制器，采用

分布式 I/O 模块作为现场设备，其硬件架构如图 1.2 所示。负责加速器设备保护的 PLC 共有 56 台，PLC 之间的通信周期要求达到 20ms 以下，采用百兆以太网 POWERLINK 技术实现。分布式 I/O 模块安装在加速器隧道内的铅屏蔽箱中，共有 110 个 I/O 模块通过 X2X 总线与 PLC 通信<sup>[35]</sup>。PLC 通过以太网接入基于 Tango 的 ALBA 主控制系统局域网中，一台 Tango 的设备服务器通过基于以太网 TCP/IP 的 Modbus 协议来轮询 PLC，从而监控 EPS 的状态。

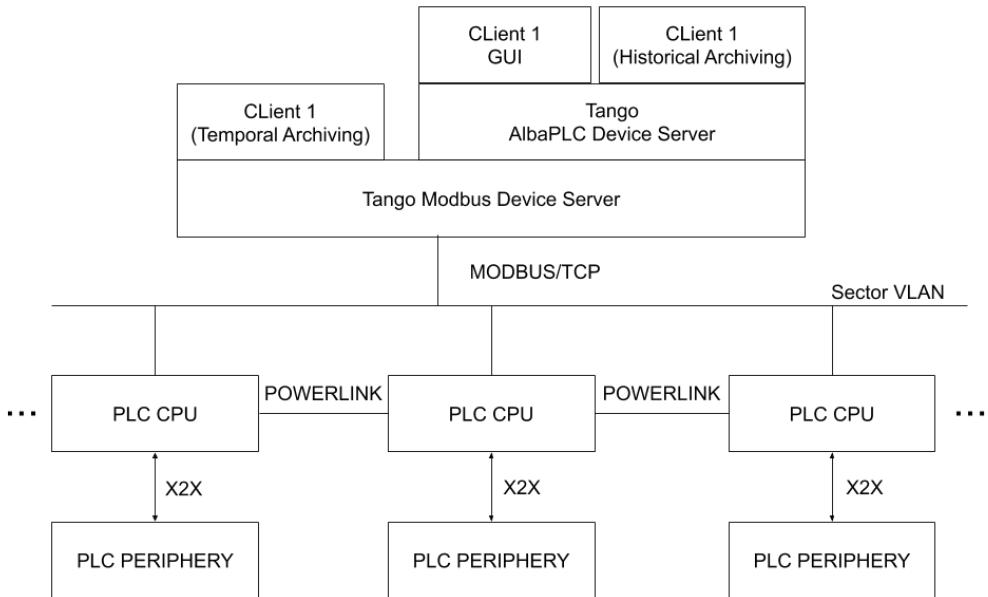


图 1.2 ALBA 设备联锁保护系统架构图

EPS 主要包括六个子系统，分别是：真空，磁铁，射频，插入件、前端设备和光束线。每条光束线都有一个独立的 EPS 系统，并全部接入到加速器的 EPS 中。每个子系统除了本地保护逻辑之外，当子系统中某些设备状态异常时，EPS 应立即通知其他相关子系统做出保护动作。例如，当真空阀由于真空管道中的压力超过阈值而关闭，EPS 应立即关闭高频系统并切断束流。每个子系统由相关 PLC 负责，子系统之间的联锁信号都是通过 POWERLINK 来进行传输。

### 1.2.2 CERN 在辐射区域关于 POWERLINK 的应用研究

欧洲核子中心 (Conseil Européen pour la Recherche Nucléaire, CERN) 位于瑞士日内瓦，是世界上最大的粒子物理实验室，建设了包括周长 26.7 千米的大型强子对撞机 (Large Hadron Collider, LHC)、周长 6.9 千米的超级质子同步加速器 (Super Proton Synchrotron, SPS) 等多个加速器和一个反质子减速器 (Antiproton Decelerator, AD)。

CERN 加速器装置控制系统的硬件架构可以分成三层，分别是前端层、现场

总线层和分布式 I/O 层，如图 1.3 所示。其中，前端层是 PLC、VME、PICMG 1.3、MTCA.4 等总线控制器，控制器通过现场总线和工业以太网来对分布式 I/O 层的不同设备进行控制。CERN 现场总线层的通讯需求如表 1.1 所示，CERN 在不同区域内采用了不同的通信技术，在无辐射区域采用 Profibus 总线和 Profinet 工业以太网技术，对靠近 LHC 装置的辐射区域则采用耐辐射的 WorldFIP 总线技术<sup>[36]</sup>。

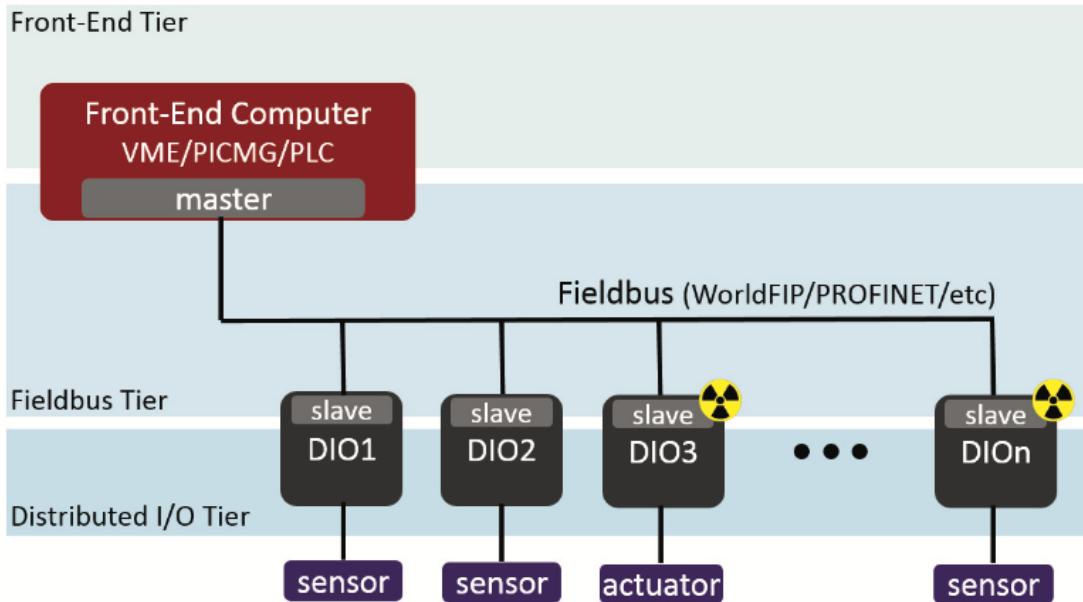


图 1.3 CERN 的硬件架构图

表 1.1 CERN 现场总线层的通讯需求

characteristic	value
topology	mainly daisy chain
nodes	100
cycle time	1ms
synchronization	100μs
data-per-cycle	1Kbyte

在 CERN 的最新研究计划中，在现场总线层加入了 POWERLINK 实时以太网技术。POWERLINK 作为唯一开源的可支持多种平台实时以太网技术，完全满足表 1.1 所示的通讯需求。目前，CERN 已在现有芯片如 NetX52 上实现了 Profinet、POWERLINK 等四种实时以太网协议，但是现有芯片不能工作在辐射区域内。CERN 计划在辐射区域内采用 Microsemi 公司的耐辐射 FPGA 芯片来实现 POWERLINK 协议，直接可以与无辐射区域的支持 POWERLINK 的现有芯片直接通信。这种升级不需要替换所有已安装的设备，极大节省了人力物力。

### 1.2.3 上海光源的光束线前端真空泄漏快保护系统

上海光源是我国建造的第一台三代中能同步辐射光源，由一台 150MeV 的直线加速器，输运线，将电子束加速到 3.5GeV 的满能量增强器，周长 432 米的 3.5GeV 电子储存环、八条光束线和数个实验站组成。其中光束线由安装在真空中的一系列光学元件组成，在加速器正常运行期间，光束线的真空中室是对储存环开放的。当光束线发生真空快泄漏事故时，为防止真空泄漏扩散，真空中保护快阀必须立即关闭。由于同步辐射光的直接照射会打坏真空中保护快阀，因此必须在真空中快阀关闭前，切断储存环的束流。位于光束线接口处的真空中泄漏快保护系统作为整个光源机器保护系统的一部分，需要在真空中泄漏信号给出后 1ms 内切断储存环束流。

上海光源的光束线前端真空中泄漏快保护系统的 IO 站点数为 20，时间响应需求是在 1ms 内，完成所有 IO 站点之间的实时数据传输<sup>[26]</sup>。系统内各节点的硬件平台是 FPGA 和实时以太网，通讯协议的设计参考了实时以太网 POWERLINK 的 PRC (PollResponse Chaining) 通讯模式，按照时间分片的原理来传输实时数据，各从站在各自的时间片内传输实时数据，避免两个或者多个从站同时发送数据，从而避免数据冲突，通信机制如图 1.4 所示。

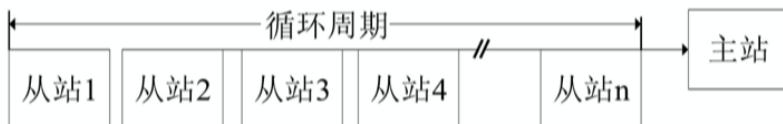


图 1.4 时间分片原理图

第一阶段，主站的时钟与所有从站时钟同步。第二阶段，在传输数据的第一个循环周期内，先进行主时钟与第一个节点时钟的同步，再按照时间分片原理进行数据传输；在第二个循环周期内，先进行主时钟与第二个节点时钟的同步，再按照时间分片原理进行数据传输；在  $n+1$  个循环周期内，先进行主时钟与第一个节点时钟的同步，再按照时间分片原理进行数据传输，依次类推。

## 1.3 论文工作的主要内容及创新点

本论文的主要研究内容为实时以太网 POWERLINK 在粒子加速器控制系统中的应用研究。首先，我们将对加速器控制系统中实时性需求进行详细分析，并调研 POWERLINK 在国内外加速器控制系统中的应用，然后开展对 POWERLINK 通信协议的研究，解析 POWERLINK 协议并进行 POWERLINK 性能测试；其次在 EPICS 环境下设计基于 POWERLINK 设计分布式 IO 控制系统，设计并开发基于 Zynq 的前端控制器以及 EPICS 设备驱动程序；然后我们将搭建原型系统并

进行相关实时性能测试，并且采用理论计算和仿真建模分析两种方法来对多节点的分布式控制系统解决方案的通信周期进行对比研究，确定方案可行性；最后，我们基于 POWERLINK 和 FPGA 控制器设计 HALF 设备保护系统，并搭建相关原型系统验证了技术方案的可行性。

本文分为五个章节，每章节的主要内容的简介如下：

第一章为绪论。本章介绍了课题研究背景，包括加速器控制系统简介、实时性分类以及加速器控制系统中的实时性分类。然后进行了国内外研究现状调研，包括基于 POWERLINK 的 ALBA 设备保护系统、CERN 在辐射区域关于 POWERLINK 的应用研究和上海光源的光束线前端真空泄漏快保护系统。最后阐明了本课题的研究意义。

第二章为 POWERLINK 通信协议研究。本章首先介绍了实时以太网的概念，并对比了 5 种主流实时以太网的性能。随后详细介绍了 POWERLINK 实时以太网技术，包括 POWERLINK 协议的通信机制、网络模型和数据帧结构。然后介绍了 POWERLINK 的实现方式，包括基于 Linux 实现 POWERLINK 协议和基于 FPGA 实现 POWERLINK 协议。最后提出了理论分析和仿真建模两种方法对 POWERLINK 通信周期进行分析和评估。

第三章为 EPICS 环境下基于 POWERLINK 的分布式 IO 系统。本章提出了两套基于 POWERLINK 的 EPICS 分布式 IO 控制系统方案，其中第二套方案是第一套方案的升级版本。本章分别介绍了两套方案的系统架构设计、软硬件设计与开发，并分别搭建了相应的系统进行了相关性能测试。然后，基于系统的具体测试情况，我们进一步完善了理论计算方法，并验证了仿真建模方法的可行性。

第四章为 HLAf 设备联锁系统的设计。本章首先介绍了课题研究背景，并对国内外设备保护系统进行了调研，基于千兆 POWERLINK 技术和 Zynq 控制器设计了 HALF 设备保护系统。随后阐述了 HLAf 设备联锁保护系统的联锁保护逻辑。然后通过理论计算和网络模拟两种方法评估了设计方案的实时性能。最后我们设计了 HALF EPS 报警系统和 HALF EPS 的历史数据存档与查询系统。

第五章为论文的工作总结与展望。

本文的主要创新点为：

1. 通过自主开发的驱动程序，首次将 POWERLINK 集成 EPICS 架构中，拓展了 EPICS 的应用范围。
2. 基于千兆 POWERLINK 技术和 Zynq 控制器设计了 HALF 设备保护系统，并通过理论计算和网络模拟两种方法评估了方案的实时性能。

## 第 2 章 POWERLINK 通信协议研究

实时以太网凭借其性能优势，不仅在工业领域得到广泛应用，在加速器控制领域的应用也越来越广泛。本章首先简介了实时以太网的发展情况，随后详细介绍了 POWERLINK 协议的通信机制、网络模型和数据帧结构，然后介绍了 POWERLINK 协议的基于 PC 和基于 FPGA 的两种实现方式，并搭建了相应的测试系统进行了性能测试，最后提出了理论分析和仿真建模两种方法对 POWERLINK 通信周期进行分析和评估。

### 2.1 实时以太网介绍

实时以太网是指在不改变 ISO/IEC8802-3 的通信特征、相关网络组件或 IEC1588 的总体行为基础上，通过一定程度的修改，使之满足实时行为。实时以太网既保持了普通以太网高通信速率等优势，又提高了其实时性、确定性、可靠性<sup>[37]</sup>。根据实现机制和工业控制领域的实时性等级，实时以太网可以划分为 class A、class B 和 class C 三类<sup>[38]</sup>，如图 2.1 所示，一种实时以太网可以同时支持不同的技术实现方案，例如 Ethernet/IP 既有支持 Class A 的方案，也有支持 Class C 的方案。Class A 类别的实时以太网保持 TCP/IP 协议簇不变，保留传输层及以下的协议，只在应用层做了实时性修改，所以这种协议虽然可以直接接入普通以太网，但是并未克服普通以太网的原有缺陷，实时性不高，Modbus/TCP 和 Ethernet/IP 实时以太网属于 Class A 类型的协议。Class B 类别实时以太网弃用了 TCP/IP 的传输协议，在数据链路层及以上使用自定义的数据通信机制，物理层使用标准以太网的通信硬件，该类别实时以太网的实时性较强，一般可满足高性能的同步运动控制应用的实时需求，Ethernet POWERLINK、EPA 和 Profinet RT 等实时以太网属于 Class B 类型的协议。Class C 类别的实时以太网在数据链路层使用自定义的数据通信机制的同时，在物理层使用了专用的网络通信硬件，该类别实时性能最好，EtherCAT 和 Profinet IRT 等实时以太网属于这类协议，Ethernet/IP 为了提升性能，也有支持 Class C 类的技术方案。综合来看，在实时性能上，Class C 优于 Class B，Class B 优于 Class A；在协议通用性上，Class A 优于 Class B，Class B 优于 Class C。目前，在工业自动化领域有 PROFINET、Ethernet POWERLINK、SERCOS III、EtherCAT 及 EtherNet/IP 5 个主流的实时以太网技术，以下分别做简单介绍。

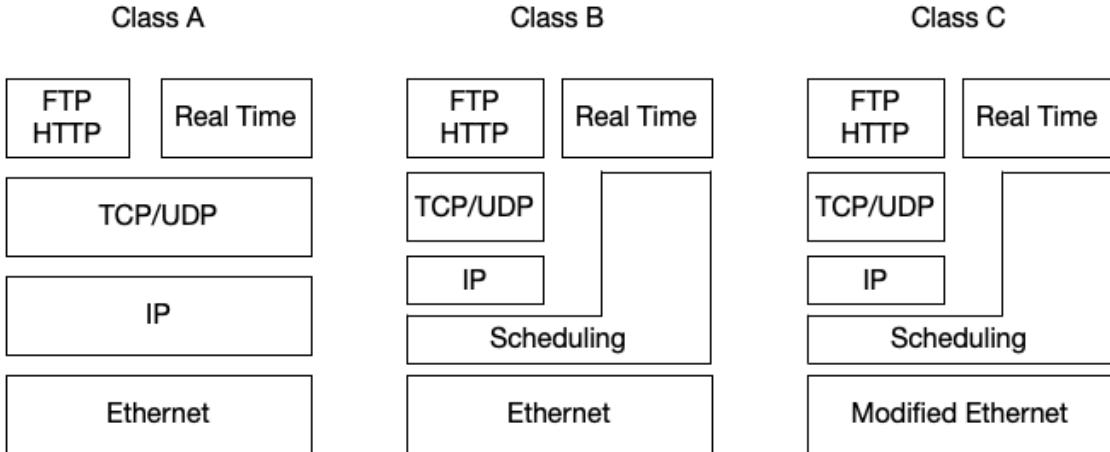


图 2.1 实时以太网分级

### 1. Ethernet POWERLINK

Ethernet POWERLINK (EPL) 最初由奥地利 B&R 公司于 2001 年 11 月创立和开发，自 2003 年以来，由独立的用户组织 EPSG 负责该技术的进一步发展<sup>[39]</sup>，为了表述方便，本文将 Ethernet POWERLINK 简称为 POWERLINK。EPSG 在 2008 年发布了 POWERLINK 技术的开源版本 openPOWERLINK，可在 PC、PLC、FPGA 等硬件平台上实现<sup>[40]</sup>。POWERLINK 遵循 IEEE1588 分布式时钟系统标准，在数据链路层采用 SCNM (Slot Communication Network Management) 时间分片网络管理机制避免数据冲突从而进行高效的数据传输。具体来说，在 POWERLINK 通信网络中，只能指定一个设备为管理节点，其他设备为受控节点。管理节点为每个受控节点分配相应的通信时隙，每个管理节点只在自己的时隙内传输数据，互不干扰，这样可以保证在特定的时间内完成各节点的数据传输，满足实时性需求。

### 2. PROFINET

PROFINET 是由 Profibus International 组织提出的实时以太网通信协议，PROFINET 是基于 IEEE 802.1D 和 IEEE 1588 标准进行实时扩展<sup>[41]</sup>。为满足不同工业场景下实时性需求，PROFINET 提供了一个标准通信协议和 SRT、IRT 两种实时通信协议。标准通信协议是基于 TCP/IP 的协议，主要用于设备的参数化配置、组态和诊断数据的读取。SRT 是软实时协议，SRT 根据 IEEE 802.1p 标准定义了报文的优先级，主要用于数据的高性能循环传输。IRT 协议的实现需要专用的等时同步 ASIC 芯片，适用于高性能数据传输和同步运动控制运用等场合。

### 3. EtherCAT

EtherCAT 由德国 Beckhoff 公司研发，由 ETG 组织维护发展。EtherCAT 采用 IEEE 1588 标准实现分布式时钟精确同步，用于现场级的高速 I/O 网络，它的物理层使用标准的以太网物理层，可为双绞线和光纤<sup>[42]</sup>。EtherCAT 采用集总束帧

等时通信的模式，主站负责把数据帧发送给各个从站，每个从站从该帧中取走属于自己的数据，或者写入该从站要输出的数据。从站在处理完成后会修改标志位，表明已完成数据帧的处理。

#### 4. SERCOS III

SERCOS III 是串行实时通信系统（Serial Real-time Communication System, SERCOS）的第三代协议，主要用于实现工控机与数字伺服系统、传感器和可编程控制器 I/O 口之间的实时数据通讯<sup>[43]</sup>。与 EtherCAT 相同，SERCOS III 采用集总束帧等时通信模式。虽然 SERCOS III 协议的通信循环周期最快可达到至  $31.25\mu\text{s}$ ，但 SERCOS III 协议的物理层和数据链路层必须通过专用的 SERCOS 接口控制芯片实现。

#### 5. EtherNet/IP

Ethernet/IP (Ethernet/Industrial Protocol) 是由 ControlNet 组织 CI、工业以太网协会 IEA 和供应商协会 ODVA 共同研发的，在标准的以太网硬件上运行，在 TCP/IP 和 UDP/IP 协议的基础上，在应用层采用 CIP (Common Industrial Protocol) 协议来进行实时数据交换<sup>[44]</sup>。EtherNet/IP 为了提升性能，也有支持 Class C 类的技术方案，2003 年，ODVA 通过制定 CIPsync 标准并推出专用以太网芯片，将 EtherNet/IP 的循环周期改善到百微秒量级，可用于伺服电机控制。

表 2.1 列出了上述 5 种主流工业以太网协议的性能。5 种主流实时以太网技术在实时性方面的性能相当，循环周期都达到了  $100\mu\text{s}$  量级，可以满足加速器装置控制系统中对实时性的大部分需求。值得指出的是，在这 5 种主流实时以太网技术中，只有 POWERLINK 是完全开源的技术。POWERLINK 的开源性对粒子加速器控制系统来说具有特别重要的意义，开放和共享一直是粒子加速器控制技术的发展方向，所以本文采用 POWERLINK 实时以太网作为研究对象。

## 2.2 POWERLINK 协议介绍

### 2.2.1 POWERLINK 协议的基本特性

Ethernet POWERLINK 协议遵循标准的 OSI 参考模型，与标准的以太网 7 层 OSI 参考模型不同的是 POWERLINK 仅定义了物理层、数据链路层和应用层三层协议，其中物理层和介质访问控制层与标准以太网相同，应用层采用工业控制领域常用的 CANopen 通信协议，为不同的应用提供兼容性接口。数据链路层采用 SCNM 机制来确保数据的实时性传输，SCNM 机制即是时间片通信网络管理机制。SCNM 的工作机制是将通信周期划分成不同的时间片段，并把这些时间段分配给等时同步数据和异步数据的传输，在等时同步数据传输阶段，网络中各节点只能在被分配的时间片段内传输数据，互不干扰。这种机制能够严格按照精

表 2.1 5 种主流工业以太网性能比较

	POWERLINK	PROFINET	EtherCAT	SERCOS III	EtherNet/IP
发布时间	2001	2006	2003	2008	2006
冗余支持	是	是	否	是	是
应用层协议	CANopen	Profibus	CANopen	SERCOS	DeviceNet
交叉通信	是	是	否	否	是
循环周期	100μs	100μs	100μs	100μs	100μs
抖动	50-80ns	20ns	20ns	50-80ns	100ns
技术实现	无 ASIC	ASIC	ASIC	ASIC	无 ASIC
开放性	主从开放	从站开放	需购买授权	主从开放	主从开放
通信模式	轮询机制	轮询机制	集束帧	集束帧	轮询机制
时钟标准	IEEE1588	IEEE1588	IEEE1588	IEEE1588	IEEE1588
安全支持	openSAFETY	ProfiSafe	FEoE	SERCOS Safe	CIP Safety

确的时间片段来进行数据的传输，从而达到高实时性<sup>[22]</sup>。

Ethernet POWERLINK 网络一般由一个主站和若干个从站组成，主站负责为从站分配时间片段和管理网络，又被称为管理节点（Management Node， MN），从站又被称为受控节点（Controlled Node， CN）。一个完整的 EPL 通信周期包括了等时同步通信阶段、异步通信阶段和空闲阶段。在等时同步通信阶段，主站会对各从站进行高精度的时钟同步，然后主站按照网络配置逐个轮询从站，从站接收到轮询请求后，才可以在属于自己的时间片段内传输数据。在异步通信阶段会进行一些非实时性数据或者其他协议的数据传输，包括基本的 TCP/IP 协议以及 HTTP 等应用层协议<sup>[22]</sup>。

基于 POWERLINK 协议的通信机制和网络结构，POWERLINK 协议具有如下特性<sup>[39]</sup>：

- (1) 开放性：POWERLINK 协议代码开源，可在 PC、PLC、FPGA、ARM 等硬件平台上实现；
- (2) 实时性：在 FPGA 硬件平台下，最快通信循环周期 <100μs；
- (3) 抖动：网络抖动在 50-80ns 范围内；
- (4) 兼容性：物理层基于 IEEE 802.3 标准以太网，支持标准以太网介质传输；应用层的 CANopen 协议支持大部分 CANopen 设备；
- (5) 网络容量：一个 EPL 网络可以支持最多 240 个站点，每个站点可以支持最多 1500 个字节的吞吐量；
- (6) 支持总线型、树型、星型、菊花链型等多种拓扑结构；
- (7) 支持 POWERLINK 网络中各节点之间通信（交叉通信）；
- (8) 支持多主冗余、环网冗余和双网冗余等多种冗余方案；

(9) 引入 openSAFETY 技术，可应用于工业安全领域。

## 2.2.2 POWERLINK 协议的网络模型

Ethernet POWERLINK 协议的网络参考模型如图 2.2 所示，主要有物理层、数据链路层和应用层组成。在添加了虚拟以太网接口之后，POWERLINK 也可以支持 TCP/IP 协议簇，应用层也可以运行 HTTP、FTP 等协议。

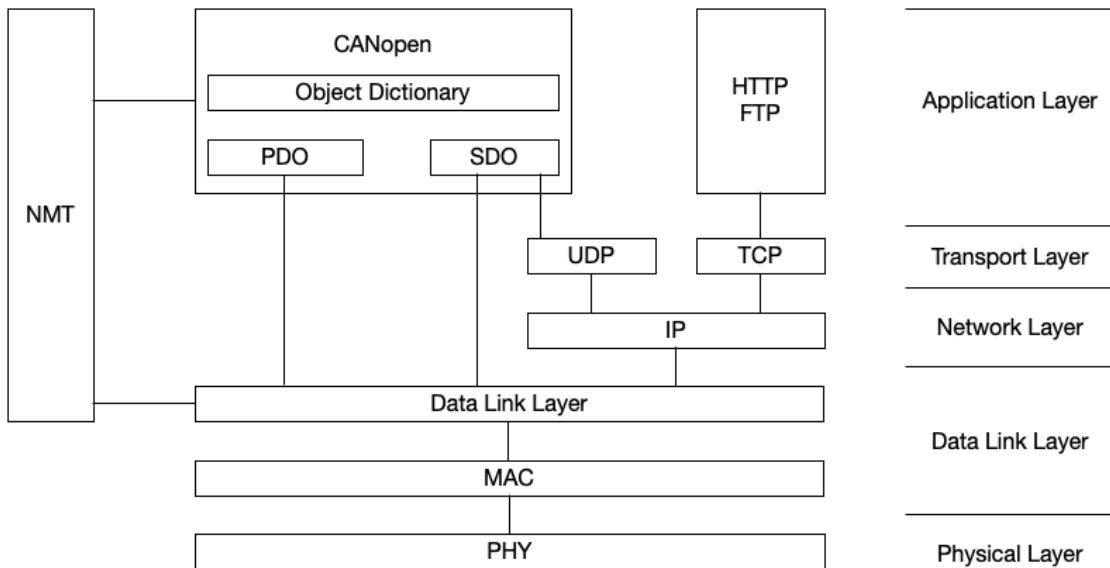


图 2.2 POWERLINK 协议网络参考模型

### • 物理层 (PHY)

POWERLINK 协议的物理层和介质访问控制子层 (MAC) 是基于标准以太网的 IEEE 802.3 标准，工业领域较为常用的为百兆以太网，最新版本支持千兆以太网。POWERLINK 网络支持总线型、树型、星型、菊花链型等多种拓扑结构，组网时为了保证数据的实时性，推荐使用集线器。集线器工作在物理层，抖动仅为几十纳秒，转发延时在亚微秒级别。网络连接器支持 RJ45 和 M12 两种类型，对于较恶劣的环境，可以使用 M12 接口和超 6 类线缆。

### • 数据链路层 (Data Link Layer)

POWERLINK 网络中的通讯周期管理、时间片分配、时钟同步、数据报文的传输都是在数据链路层完成的。数据链路层定义了 5 种类型的协议帧，如表 2.2 所示，分别是 SoC、PReq、PRes、SoA、Asnd。

POWERLINK 数据帧封装在标准以太网的数据段中，与标准以太网一致，POWERLINK 数据帧长度最长为 1518Byte，最短为 64Byte。POWERLINK 数据帧的结构如图 2.3 所示，其中 0~13 字节是标准以太网的帧头，n~n+4 为 CRC 校验序列，14~n 字节是 POWERLINK 数据帧信息。

表2.2 Ethernet POWERLINK协议帧类型

协议帧类型	协议帧名称
周期开始	SoC
主站轮询请求	PReq
从站轮询响应	PRes
异步通信开始	SoA
从站异步响应	Asnd

字节偏移	比特偏移								数据出处	
	7	6	5	4	3	2	1	0		
0~5	目的MAC地址								Ethernet II	
6~11	源MAC地址									
12~13	以太网报文类型(0x88AB)									
14	保留	POWERLINK报文类型识别								
15	目的POWERLINK站ID号								POWERLINK	
16	源POWERLINK站ID号									
17~n	数据									
n+1~n+4	CRC校验和								Ethernet II	

图2.3 POWERLINK协议帧结构

下面我们以 SoC 帧和 PReq/PRes 数据帧为例介绍 POWERLINK 数据帧的结构。SoC 帧作为同步数据帧，帧长度固定为 45Byte，其结构如图 2.4 所示，SoC 数据帧中包含了网络绝对时间和相对时间，当主站为 NMT\_GS\_INITIALIZING 状态时，相对时间清零，然后每产生一个 SOC，该数值就累加一个循环周期，其单位为  $\mu\text{s}$ 。

PReq 作为主站发送给从站的请求数据帧，其结构如图 2.5 所示，其中包含了源站点站点号和用来寻址的目的站点号，从第 10 个字节开始是主站发送给从站的用户数据。PRes 作为从站的响应数据帧通信数据帧，其结构和长度均与 PReq 数据帧类似，不同的是第 5 个字节中包含了异步发送队列中数据帧的优先级和队列中挂起的数据帧数目。

一个基本的 POWERLINK 通信周期分为三个阶段，按发生顺序分别是等时同步阶段、异步通信阶段和空闲阶段，通信过程如图 2.6 所示。等时同步阶段用来传输周期性的实时数据，主站首先通过以太网广播发送 SoC 帧对所

字节偏移	比特偏移													
	7	6	5	4	3	2	1	0						
0	res	报文类型												
1	目的地													
2	源													
3	保留													
4	MC	PS	res	res	res	res	res	res						
5	res	res	res			res								
6...13	NETtime/保留													
14...21	Relative Time/保留													
22...45	保留													

图 2.4 SoA 协议帧结构

字节偏移	比特偏移													
	7	6	5	4	3	2	1	0						
0	res	报文类型												
1	目的地													
2	源													
3	保留													
4	MC	PS	res	res	res	res	res	res						
5	res	res	PR			RS								
6	PDOVersion													
7	res													
8..9	大小													
10..n	载荷													

图 2.5 PReq 协议帧结构

有从站进行时钟同步，然后通过 PReq 帧依次轮询每个已配置的处于活动状态的从站。PReq 帧仅被节点号对应的从站接收，对应的从站通过 PRes 广播帧响应此请求。若有从站在一定时间内未做出响应，则主站不会因此中断通信，会继续轮询之后的从站。在异步通信阶段，仅一个被授权的从站可以在此阶段传输非实时数据。主站首先广播 SoA 帧，被授权的从站向网络中发送 Asnd 帧回应。异步通信阶段不是必须的，没有异步通信阶段的情况下，等时同步阶段之后会直接进入空闲阶段。空闲阶段是本周期数据传输完成后与下一个周期开始之间的时间间隔。

- **应用层 (Application Layer)**

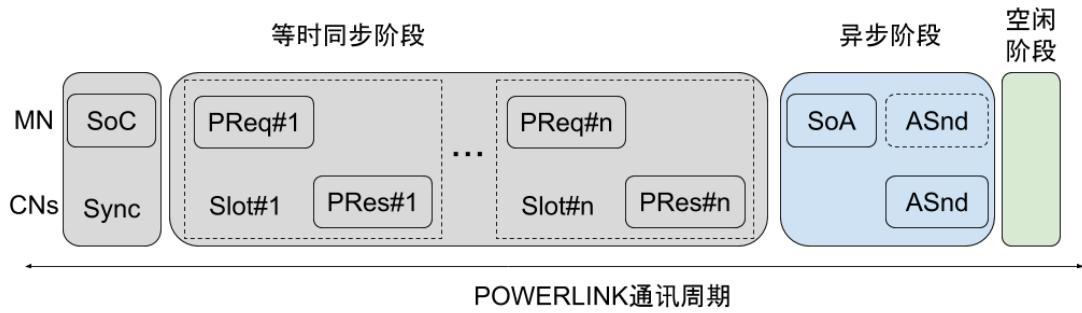


图 2.6 POWERLINK 通讯过程

POWERLINK 的应用层时遵循 CANOpen 标准，其核心是对象字典 (OBD)<sup>[45]</sup>。OBD 是一种数据结构，其中包含与通信协议相关的所有参数以及通信中的过程变量。在 POWERLINK 网络中，周期性实时传输的 OBD 对象被映射为过程数据对象 (PDO)，PDO 分为两类：发送过程数据对象 (TPDO) 和接收过程数据对象 (RPDO)，分别是该站的发送数据和接收数据的载体。非实时性的 OBD 对象（例如网络配置命令）被映射为服务数据对象 (SDO)。

## 2.3 POWERLINK 协议的实现

Ethernet POWERLINK 协议作为一种开源的实时以太网技术可以在多种平台上实现，本章重点研究了 POWERLINK 协议实现方案之一的 openPOWERLINK。openPOWERLINK 是由德国 SYSTEC 电子、B&R 自动化和 Kalcito 公司联合开发并维护的开源 POWERLINK 协议栈，在遵循 BSD 许可证的前提下，用户可以在 <https://sourceforge.net/projects/openpowerlink/> 免费获取。openPOWERLINK 作为一种纯软件实现方案，它实现了 POWERLINK 协议栈的所有重要功能，例如标准的轮询通信模式，动态和静态 PDO 映射，通过 ASnd 异步数据帧传输的 SDO，以及通过虚拟以太网接口实现的异步通信<sup>[40]</sup>。openPOWERLINK 的软件结构分为三层，分别是用户应用层、通信抽象层和底层协议栈层，其中用户应用层和底层协议栈层分别对应 POWERLINK 协议的应用层和数据链路层，通信抽象层采用共享缓存区的方式连接应用层和协议栈层。openPOWERLINK 支持 Intel X86、FPGA 等多种硬件平台和 Linux、Windows 等操作系统，本章重点研究了 openPOWERLINK 在 Linux 环境下和 FPGA 平台下的实现，并进行了性能测试。

在组建 POWERLINK 网络之前，需要进行网络配置。openCONFIGURATOR 是一个开源网络配置工具<sup>[46]</sup>，可以方便快速的组建一个 POWERLINK 网络、轻松地配置各个节点的网络参数和映射参数。图 2.7 为 openCONFIGURATOR 的主

界面，我们可以进行通信周期的配置以及各从站的通信参数配置。openCONFIGURATOR 的输出文件主要是后缀名.cdc 的网络配置文件。cdc 文件是一个二进制文件，保存了整个网络的配置信息。主站会读取 cdc 文件并根据 cdc 文件来配置循环周期、主站以及各个从站的网络参数和映射参数。

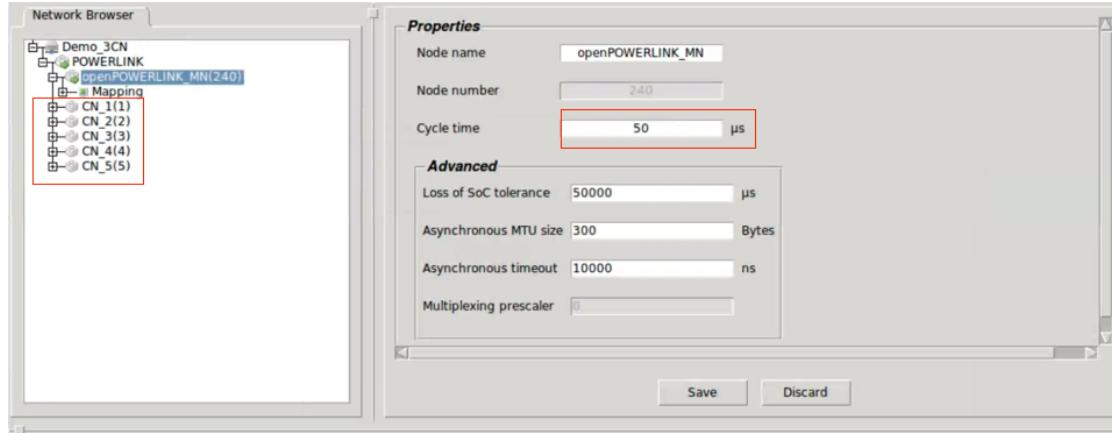


图 2.7 openconfigurator 主界面

图 2.8 为 openCONFIGURATOR 中 1 号从站的配置界面，值得注意的是从站的 PollResponse Timeout 网络参数，这个是指主站接收该从站 PRes 数据桢的超时时间。网络中的从站有可能发生故障，导致从站无法发送数据或者发送的数据无法到达主站，如果主站在 PollResponse Timeout 这个时间段内没有收到该从站的回复数据时，就会将该节点略过，和下一个节点通信。此外，如果从站设备的反应时间很长，那么就需要将 PollResponse Timeout 参数设置的大一些，例如如果该从站基于 FPGA，这个参数设置为 2~20 $\mu$ s 均可，如果该从站基于 windows，这个参数可能要设置为几个 ms。

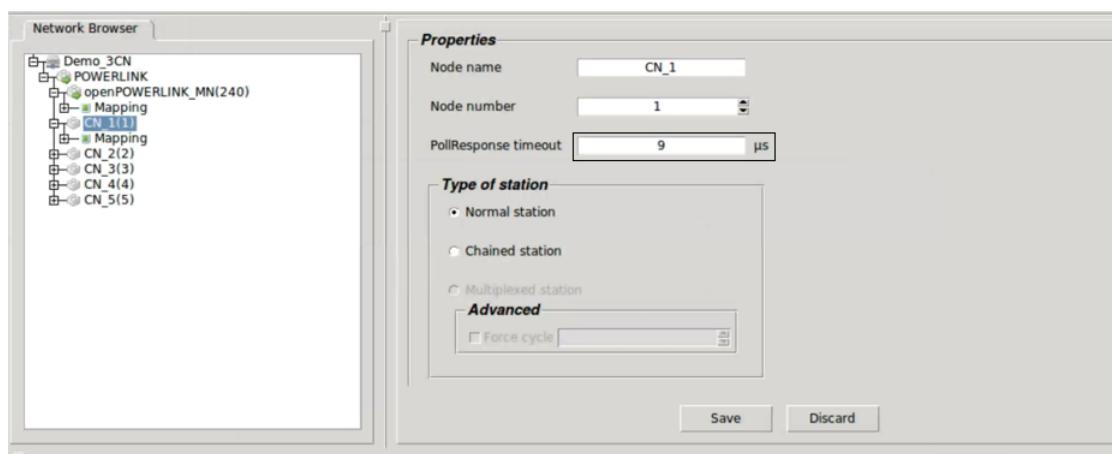


图 2.8 1 号从站的配置界面

### 2.3.1 基于 Linux 系统实现 POWERLINK 协议

openPOWERLINK 不需要额外硬件就可以直接运行在基于 x86 架构的 Linux 系统平台下，为了减小 Linux 系统对 openPOWERLINK 实时性能的影响，我们为 linux 系统安装了 RT-PREEMPT 实时补丁。根据协议栈运行位置的不同，openPOWERLINK 在实时 Linux 系统下有两种工作模式<sup>[47]</sup>：

- (1) 协议栈工作在用户空间：协议栈被封装成一个运行于用户空间的用户代码库，通过调用 libcap library 来访问以太网接口。这种工作模式的优点是支持所有标准以太网的网卡芯片。这种工作模式的实时性能在 1ms 左右，可以满足大多数应用场景。
- (2) 协议栈工作在内核空间：协议栈直接工作在 Linux 内核空间下，直接访问以太网硬件接口，实时性高。这种工作模式的缺点是对以太网硬件接口依赖严重，需要针对不同的网口类型开发相应的驱动程序，限制了 openPOWERLINK 的应用。

本小节对工作在用户空间的 POWERLINK 协议栈进行了性能测试，内核模式下的高性能方案在第 3.1 节中详细介绍。

我们建立由 2 台 RT-Linux PC 构成的简单系统，系统架构如图 2.9 所示，一台用作 POWERLINK 主站，一台是节点号为 1 的从站，主站和从站均运行 V2.2.2 版本的 openPOWERLINK 开源程序，节点之间采用千兆 POWERLINK 通信。

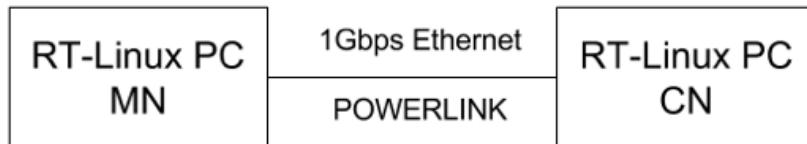


图 2.9 基于 RT-Linux PC 的 POWERLINK 通信测试系统

openPOWERLINK 程序的安装与启动过程如下所示：

```

1 1. 安装 cmake 编译工具
2 下载 cmake-3.5.0-rc3.tar.gz 并解压
3 > cd <cmake_dir>
4 > ./bootstrap
5 > make
6 > make install
7
8 2. 编译 openPOWERLINK 协议栈
9 下载 openPOWERLINK_V2.2.2.tar.gz 并解压
10 Create debug libraries
11 > cd <openPOWERLINK_dir>/stack/build/linux
  
```

```

12 > cmake -DCMAKE_BUILD_TYPE=Debug ../..
13 > make
14 > make install
15 Create release libraries
16 > cd <openPOWERLINK_dir>/stack/build/linux
17 > cmake -DCMAKE_BUILD_TYPE=Release ../..
18 > make
19 > make install
20
21
22 3. 编译 openPOWERLINK 应用层程序
23 > cd <openPOWERLINK_dir>/apps/<demo_dir>/build/linux
24 > cmake ../..
25 > make
26 > make install
27
28
29 4. 将配置文件 mnobd.cdc 拷贝至主站的启动目录下
30 [test@powerlinkTest1 demo_mn_console]$ ls
31 demo_mn_console mnobd.cdc mnobd.cdc-bk set_prio
32
33 5. 启动 openPOWERLINK
34 主站启动：
35 [test@powerlinkTest1 demo_mn_console]$ sudo ./demo_mn_console
36 从站启动：
37 [test@powerlinkTest2 demo_cn_console]$ sudo ./demo_mcn_console

```

我们将测试系统通信周期配置为 1ms，系统可以稳定运行，图 2.10 是抓取的 POWERLINK 协议帧数据。如图 2.10 所示，在 1ms 的通信周期内，主站和从站严格按照 PReq/PRes 的轮询响应模式进行通信，通过两个连续 SoC 帧之间的间隔可以计算得到通信周期。我们抓取了 25 万个连续的 POWERLINK 数据帧，并绘制了如图 2.11 所示的概率密度分布图，该图表明，大多数的通信周期接近 1ms，实时性一般，抖动较大。

### 2.3.2 基于 FPGA 实现 POWERLINK 协议

基于 FPGA 实现 POWERLINK 协议的方案有两种<sup>[48]</sup>：

- (1) 软核方案：在 FPGA 中的 MCU 上运行 POWERLINK 协议栈，MCU 为软核（例如：Altera Nios，Xilinx Zynq），软核中运行的是 openPOWERLINK 的协议栈，循环周期可以达到 100~200μs。
- (2) HDL 方案：采用纯硬件描述语言 HDL 来实现 POWERLINK 的协议栈，这种方案不需要 MCU，实时性能较高，循环周期可以达到数十 μs。

本小节主要介绍软核方案的技术细节和测试结果，HDL 方案在第 3.2 中详细

No.	Time	Source	Destination	Protocol	Lan Info
1	0.000000000	Dell_a1:e6:ba	EPLv2_SoC	POWERLINK	60 240-> 1 PReq [ 1 ] F:RD=1 V:0.0
2 <b>0.000053046</b>	<b>Dell_a1:e6:ba</b>	<b>HewlettP_5c:30:28</b>	<b>POWERLINK</b>	<b>60 240-&gt; 1 PReq [ 1 ] F:RD=1,RS=0,PR=0 V=0.0 NMT_CS_OPERATIONAL</b>	
3 0.000089927	HewlettP_5c:30:28	EPLv2_PRes	POWERLINK	60 1->255 PRes [ 1 ] F:RD=1,RS=0,PR=0 V=0.0 NMT_MS_OPERATIONAL	
4 0.000281205	Dell_a1:e6:ba	EPLv2_SoA	POWERLINK	60 240->255 SoA (IDENT_REQUEST)->253 NMT_MS_OPERATIONAL	
5 0.001001998	Dell_a1:e6:ba	EPLv2_SoC	POWERLINK	60 240->255 SoC	
6 <b>0.001053872</b>	<b>Dell_a1:e6:ba</b>	<b>HewlettP_5c:30:28</b>	<b>POWERLINK</b>	<b>60 240-&gt; 1 PReq [ 1 ] F:RD=1 V:0.0</b>	
7 0.001088974	HewlettP_5c:30:28	EPLv2_PRes	POWERLINK	60 1->255 PRes [ 1 ] F:RD=1,RS=0,PR=0 V=0.0 NMT_MS_OPERATIONAL	
8 0.001290314	Dell_a1:e6:ba	EPLv2_SoA	POWERLINK	60 240->255 SoA (NO_SERVICE)-> 0 NMT_MS_OPERATIONAL	
9 0.001983038	Dell_a1:e6:ba	EPLv2_SoC	POWERLINK	60 240->255 SoC	
10 <b>0.002048175</b>	<b>Dell_a1:e6:ba</b>	<b>HewlettP_5c:30:28</b>	<b>POWERLINK</b>	<b>60 240-&gt; 1 PReq [ 1 ] F:RD=1 V:0.0</b>	
11 0.002076496	HewlettP_5c:30:28	EPLv2_PRes	POWERLINK	60 1->255 PRes [ 1 ] F:RD=1,RS=0,PR=0 V=0.0 NMT_CS_OPERATIONAL	
12 0.002277695	Dell_a1:e6:ba	EPLv2_SoA	POWERLINK	60 240->255 SoA (NO_SERVICE)-> 0 NMT_MS_OPERATIONAL	
13 0.002996693	Dell_a1:e6:ba	EPLv2_SoC	POWERLINK	60 240->255 SoC	
14 <b>0.003048393</b>	<b>Dell_a1:e6:ba</b>	<b>HewlettP_5c:30:28</b>	<b>POWERLINK</b>	<b>60 240-&gt; 1 PReq [ 1 ] F:RD=1 V:0.0</b>	
15 0.003097355	HewlettP_5c:30:28	EPLv2_PRes	POWERLINK	60 1->255 PRes [ 1 ] F:RD=1,RS=0,PR=0 V=0.0 NMT_CS_OPERATIONAL	
16 0.003282814	Dell_a1:e6:ba	EPLv2_SoA	POWERLINK	60 240->255 SoA (NO_SERVICE)-> 0 NMT_MS_OPERATIONAL	
17 0.004002169	Dell_a1:e6:ba	EPLv2_SoC	POWERLINK	60 240->255 SoC	
18 <b>0.004053125</b>	<b>Dell_a1:e6:ba</b>	<b>HewlettP_5c:30:28</b>	<b>POWERLINK</b>	<b>60 240-&gt; 1 PReq [ 1 ] F:RD=1 V:0.0</b>	
19 0.004096991	HewlettP_5c:30:28	EPLv2_PRes	POWERLINK	60 1->255 PRes [ 1 ] F:RD=1,RS=0,PR=0 V=0.0 NMT_CS_OPERATIONAL	
20 0.004292116	Dell_a1:e6:ba	EPLv2_SoA	POWERLINK	60 240->255 SoA (NO_SERVICE)-> 0 NMT_MS_OPERATIONAL	
21 0.004985183	Dell_a1:e6:ba	EPLv2_SoC	POWERLINK	60 240->255 SoC	
22 <b>0.005041175</b>	<b>Dell_a1:e6:ba</b>	<b>HewlettP_5c:30:28</b>	<b>POWERLINK</b>	<b>60 240-&gt; 1 PReq [ 1 ] F:RD=1 V:0.0</b>	
23 0.005065053	HewlettP_5c:30:28	EPLv2_PRes	POWERLINK	60 1->255 PRes [ 1 ] F:RD=1,RS=0,PR=0 V=0.0 NMT_CS_OPERATIONAL	
24 0.005270030	Dell_a1:e6:ba	EPLv2_SoA	POWERLINK	60 240->255 SoA (NO_SERVICE)-> 0 NMT_MS_OPERATIONAL	

> Frame 1: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0  
> Ethernet II, Src: Dell\_a1:e6:ba (78:2b:cb:a1:e6:ba), Dst: EPLv2\_SoC (01:11:1e:00:00:01)  
> Ethernet POWERLINK

```

0000 01 11 1e 00 00 01 78 2b cb a1 e6 ba 88 ab 01 ff .....X* ....Y
0010 f0 00 00 00 00 00 00 00 00 c8 59 af 01 .....X* ....Y
0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....X* ....Y
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....X* ....Y

```

图 2.10 基于两台 PC 的 wireshark 抓包结果

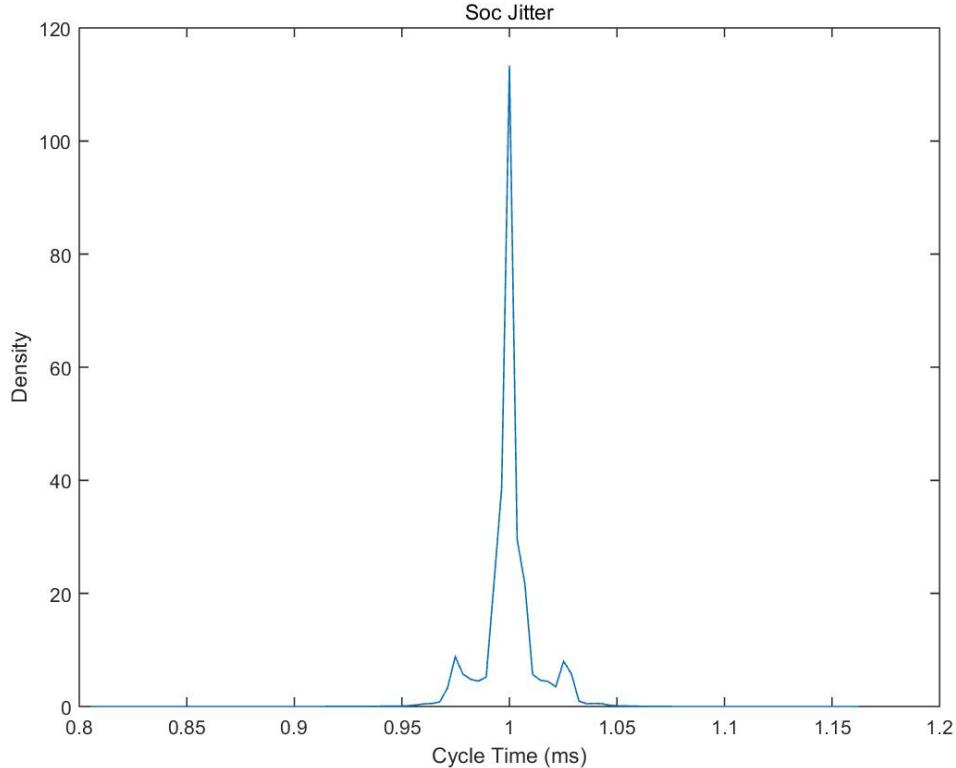


图 2.11 基于两台 PC 的通信周期概率密度分布图

介绍。软核方案实现的硬件平台是友晶科技的 DE2-115 的 FPGA 开发板，DE2-115 开发板具有充足的逻辑处理单元和存储单元，同时兼具低功耗、低成本的优点，具体配置如下：

- (1) 核心的 FPGA 芯片：FPGACyclone IV 4CE115F29，包含有 115k 个逻辑单元，低成本，低功耗。

(2) 存储单元: 2-Mbyte SRAM, 64-Mbyte SDRAM, 8-Mbyte Flash memory。(3) 时钟: 50M 晶振, 支持外部时钟。(4) 标准接口: 2 个 100M 自适应以太网络适配器, 80 针带保护电路的外接 IO, 通用串行总线 USB 接口, SD Card 接口, RS-232 标准串口。

软件基于 Altera Nios II 软核来实现, Nios II 是由 Altera 公司推出的嵌入式软核处理器, 它采用 32 位精简指令集, 32 位数据通道, 可在一个时钟周期内完成一条指令的处理。作为目前最流行的软核处理器 Nios II 核拥有高计算性能, 但占用不到 50% 的 FPGA 资源。同时 Altera 公司提供了与 Nios II 配套的软件集成开发环境 Quartus II, 可以在该环境下完成包括编辑、编译、调试程序和下载在内的软件开发任务。

基于 Nios II 软核的 POWERLINK 协议软件架构如图 2.12 所示。软核处理器 Altera Nios II 被实例化为 POWERLINK 通信处理器 (Communication Processor) 和主机处理器 (host processor), 这两个软核分别用来执行 openPOWERLINK 内核层和用户层的代码。处理器之间通过 Altera Avalon 总线连接。以太网接口是通过优化控制器 openMAC 构建的, 此外, 还实现了一个集线器 (HUB) 软核, 用来组建菊花链拓扑结构的网络。

采用软核方案来实现 POWERLINK, 通信过程如下:

第一步: 主站的 PReq 数据帧通过以太网的 openMAC 发送到从站。

第二步: 从站的以太网 openMAC 收到数据帧以后, 向 POWERLINK 通信处理器和主机处理器发起中断, 请求处理器处理收到的数据帧。

第三步: 处理器响应中断请求, 处理 PReq 数据帧, 并回复 PRes 数据帧。在这个过程中, 处理器响应中断请求并处理中断的时间比较长, 往往在几十个  $\mu s$ 。

基于软核方案, 我们设计了由 1 个主站和 6 个从站组成的系统, 系统架构如图 2.13 所示。该系统是线性拓扑结构, 从站按照节点号从小到大的顺序依次连接。主站是 RT-Linux PC, 从站是 DE2-115 开发板, 节点之间基于百兆 POWERLINK 通信<sup>[49]</sup>。

对于 POWERLINK 测试系统, 我们采用 Hilscher netAnalyzer 以太网分析仪用来抓取 POWERLINK 数据帧。以太网分析仪 netANALYZER 是用于对实时以太网传输数据进行监控和分析的工具, 由德国赫优讯公司生产。Hilscher netANALYZER 由于其内置的 ASIC 专用芯片, 测量分辨率可达到 10ns, 可以用来测量各种高精度时间同步协议, 如: IEEE1588、PROFINET IRT、EtherNet/IP 集成 CIP Sync、EtherCAT 或 Sercos、Ethernet POWERLINK。Hilscher netANALYZER 在硬件上集成的测试访问点 Tap, 保证了测试过程中以太网通信不会受到影响, 既不会带来额外的数据帧延时, 也不影响时间抖动。测试访问点 Tap 还保证了即使在错误状态下数据内容也不会被改变。网络上发生的所有报文数据及报文错

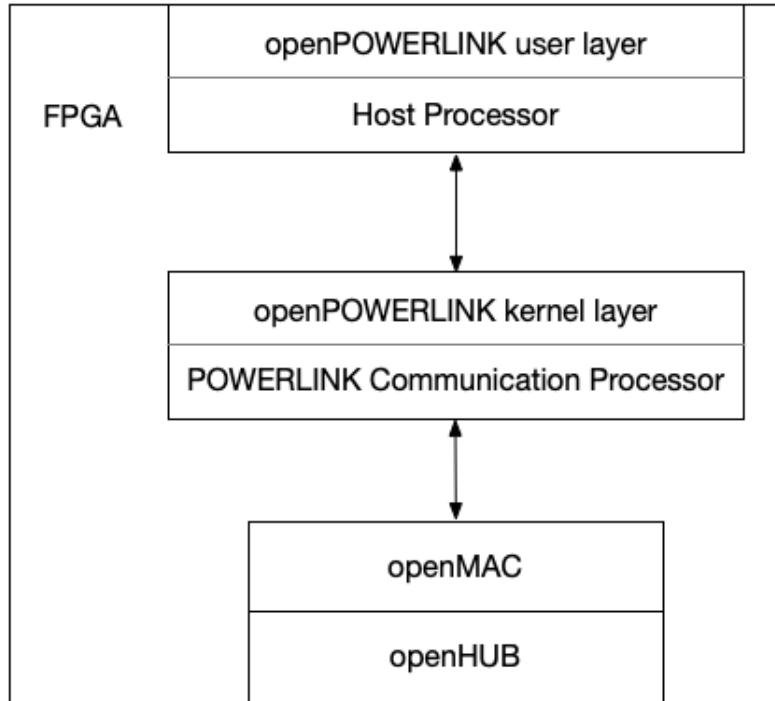


图 2.12 基于 DE2-115 开发板的 POWERLINK 软件架构图

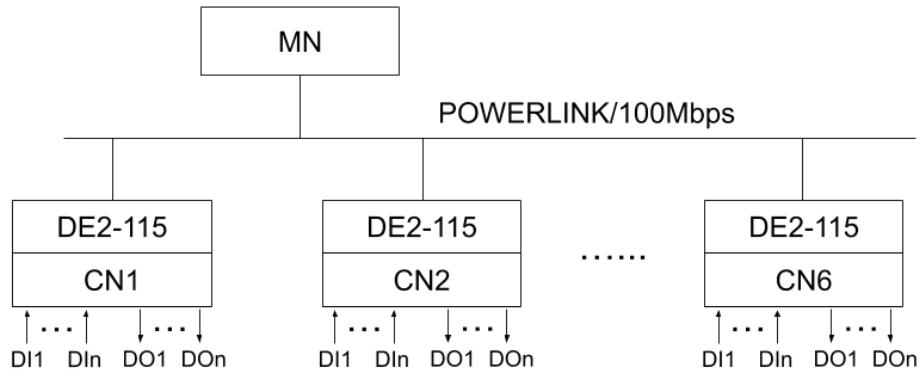


图 2.13 基于 DE2-115 开发板的 POWERLINK 通信测试系统

误都毫无改变地传输至现场设备。如图 2.14 所示的以太网分析仪 netANALYZER 包含 5 个以太网端口，其中 2 个“Tap”端口组成“TapA”，2 个“Tap”端口组成“TapB”，可以同时抓取 2 条通信网络上的数据帧。另一个端口作为“镜像端口”负责接收来自 TapA/TapB 端口上的转发数据，同时“镜像端口”负责将数据传输至与 PC，PC 上运行了专用软件 netANALYZER Scope2，支持将抓包结果输出为可被 Wireshark 解析的 pcap 格式文件，从而可以进行更详细的分析。这里我们仅用到了“TapA”端口进行 POWERLINK 通信测试。

基于此设计，我们搭建了如图 2.15 所示的测试系统。其中，PC 的操作系统是带有实时补丁 rtai-3.10.75 的 Centos7.4.1708，PC 上的 openPOWERLINK 程序工作在用户空间。示波器用来测试系统响应时间，我们采用 Hilscher netAnalyzer



图 2.14 以太网分析仪 netANALYZER 正面（左图）和背面（右图）

以太网分析仪用来抓取 POWERLINK 数据帧，这里我们仅用到了“TapA”端口进行 POWERLINK 通信测试。

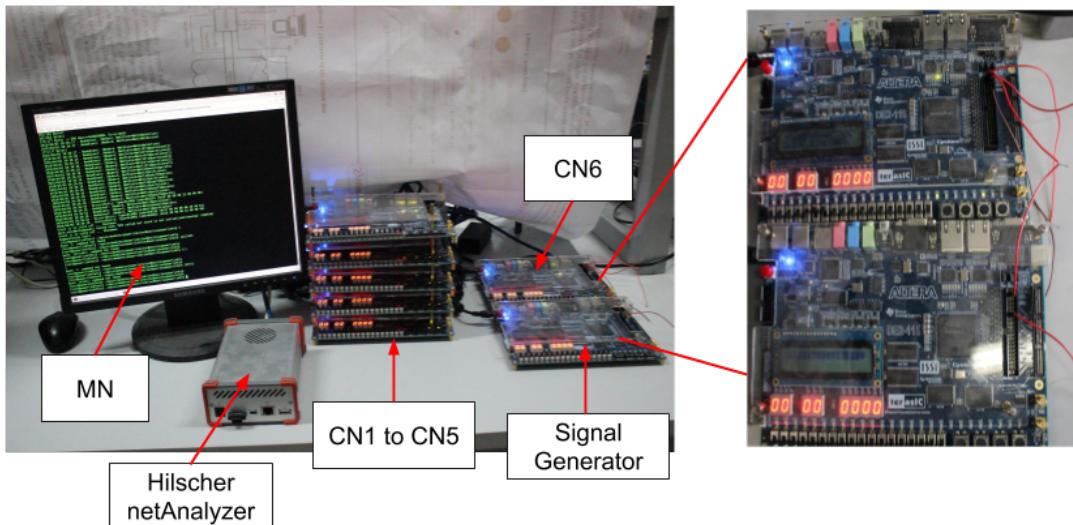


图 2.15 基于 DE2-115 开发板的 POWERLINK 通信系统实物图

基于此测试系统，我们进行了通信周期和响应时间的性能测试。经过多次测试，该系统可以在  $700\mu\text{s}$  的通信周期下的稳定运行，Hilscher netAnalyzer 以太网分析仪的抓包结果如图 2.16 所示。我们测试了大约 27000 个连续的 POWERLINK 通信周期，循环周期的平均值约为  $703\mu\text{s}$ ，图 2.17 是 python 程序拟合的通信周期的概率密度分布曲线，密度的峰值在通信周期为  $700\mu\text{s}$  附近，但是抖动较大。

响应时间是输入信号到达系统和系统回复输出信号之间的时间差。它包括 DE2-115 开发板上输入/输出信号的处理时间，数据通信时间，网线延迟和集线器延迟。数据通信时间是主要部分，其他时间在  $1\sim10\mu\text{s}$  的水平。对于此测试系统，在一个通信周期中，从站 1 是第一个开始通信的从站，从站 6 是最后一个结束通信的从站，所以从站 1 输入信号到从站 6 输出信号的时间差是最长的通信

88 0 .004495280	IntelCor_03:e3:2d	EPLv2_SoC	POWERLINK	64 240->255 SoC
89 0 .004531440	IntelCor_03:e3:2d	02:88:ab:00:00:01	POWERLINK	64 240-> 1 PReq
90 0 .004540310	02:88:ab:00:00:01	EPLv2_PRes	POWERLINK	64 1->255 PRes
91 0 .004574800	IntelCor_03:e3:2d	02:88:ab:00:00:02	POWERLINK	64 240-> 2 PReq
92 0 .004587910	02:88:ab:00:00:02	EPLv2_PRes	POWERLINK	64 2->255 PRes
93 0 .004618400	IntelCor_03:e3:2d	02:88:ab:00:00:03	POWERLINK	64 240-> 3 PReq
94 0 .004635710	02:88:ab:00:00:03	EPLv2_PRes	POWERLINK	64 3->255 PRes
95 0 .004691360	IntelCor_03:e3:2d	02:88:ab:00:00:04	POWERLINK	64 240-> 4 PReq
96 0 .004712870	02:88:ab:00:00:04	EPLv2_PRes	POWERLINK	64 4->255 PRes
97 0 .004759520	IntelCor_03:e3:2d	02:88:ab:00:00:05	POWERLINK	64 240-> 5 PReq
98 0 .004785270	02:88:ab:00:00:05	EPLv2_PRes	POWERLINK	64 5->255 PRes
99 0 .004830880	IntelCor_03:e3:2d	02:88:ab:00:00:06	POWERLINK	64 240-> 6 PReq
100 0 .004860870	02:88:ab:00:00:06	EPLv2_PRes	POWERLINK	64 6->255 PRes
101 0 .004894720	IntelCor_03:e3:2d	EPLv2_SoA	POWERLINK	64 340->255 SoA
102 0 .005192400	IntelCor_03:e3:2d	EPLv2_SoC	POWERLINK	64 240->255 SoC

图 2.16 基于 DE2-115 开发板的 wireshark 抓包结果

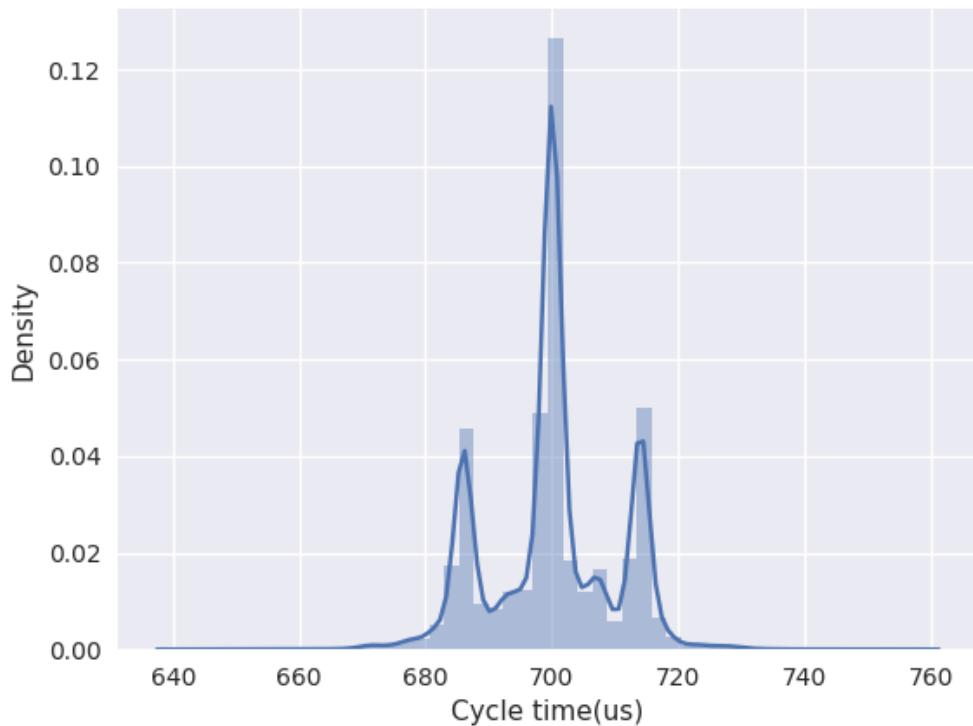


图 2.17 基于 DE2-115 开发板的通信周期概率密度分布图

时间，即是系统最长响应时间。完成的响应过程如下：

第一步：在第 n 个周期中，从站 1 的输入信号通过 PRes 数据帧进行 POWERLINK 网络。

第二步：主站收到从站 1 的 PRes 数据帧后，对输入信号进行相应的处理。

第三步：在第 n+1 个周期中，主站通过 PReq 数据帧将输出数据传输给从站 6，从站 6 输出相应信号。

我们使用示波器测试了该测试系统的最长响应时间，图 2.18 是示波器在“单触发”模式下的测试结果，通道 2 (ch2) 是从站 1 的输入信号，通道 1 (ch1) 是从站 6 的输出信号。响应时间约为 1070μs，

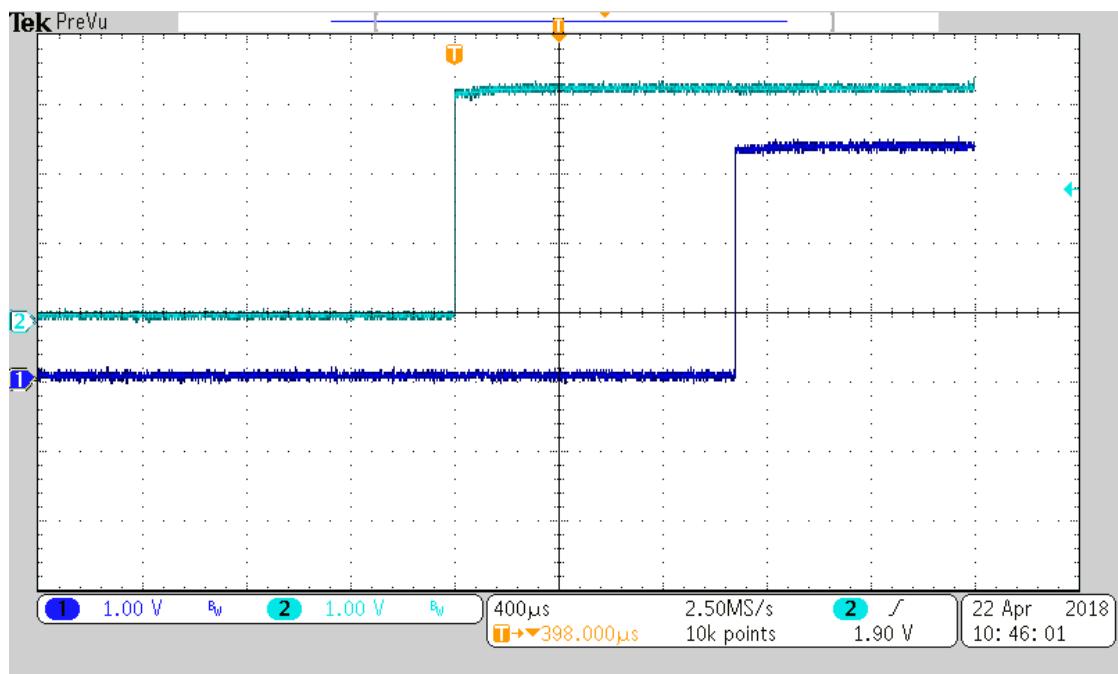


图 2.18 基于 DE2-115 开发板的测试系统响应时间测试结果

### 2.3.3 测试小结

本节分析了 openPOWERLINK 的两种实现方式，分别是 Linux 用户空间方案和基于 FPGA 的软核方案。针对两种实现方式我们分别搭建了测试系统并进行相关性能测试，根据测试结果，总结如下：

- (1) 在 Linux 用户空间方案中，通信周期最快为 1ms；在基于 FPGA 的软核方案中，七个节点的系统通信周期最快为  $700\mu s$ 。POWERLINK 协议在 FPGA 硬件平台下的实时性能明显优于在 x86 Linux 平台下的性能。
- (2) 对于基于 FPGA 的软核方案，如图 2.16 所示，相邻 PRes 数据帧和 PReq 数据帧为 FPGA 从站响应时间，反应时间约为  $10\mu s$  左右，这是由于软核处理 PReq 数据帧并回复 PRes 数据帧而导致的延时。若采用纯硬件描述语言 HDL 来实现 POWERLINK 的协议栈可缩短从站的响应时间，进而提高实时性能。
- (3) 在数据传输速度上，百兆以太网比千兆以太网低一个量级，所以 POWERLINK 物理层采用千兆以太网可以减少数据传输的延时。
- (4) 通过以下三种方式可以进一步提高实时性：
  - openPOWERLINK 工作在 Linux 系统的内核空间下。
  - 在 FPGA 平台上，采用纯硬件描述语言 HDL 来实现 POWERLINK 的协议栈。
  - POWERLINK 物理层采用千兆以太网。

## 2.4 POWERLINK 通信周期的理论计算

通信周期直接反应系统的实时性，决定了系统的响应时间。我们提出了理论分析和仿真模拟两种方法，同时结合系统的实际通信参数，来预估多节点 POWERLINK 系统的通信周期，从而为多节点大规模的 POWERLINK 方案设计提供依据。本节首先详细分析 POWERLINK 通信周期的时间组成，最后推导出通信周期的计算公式<sup>[50]</sup>。

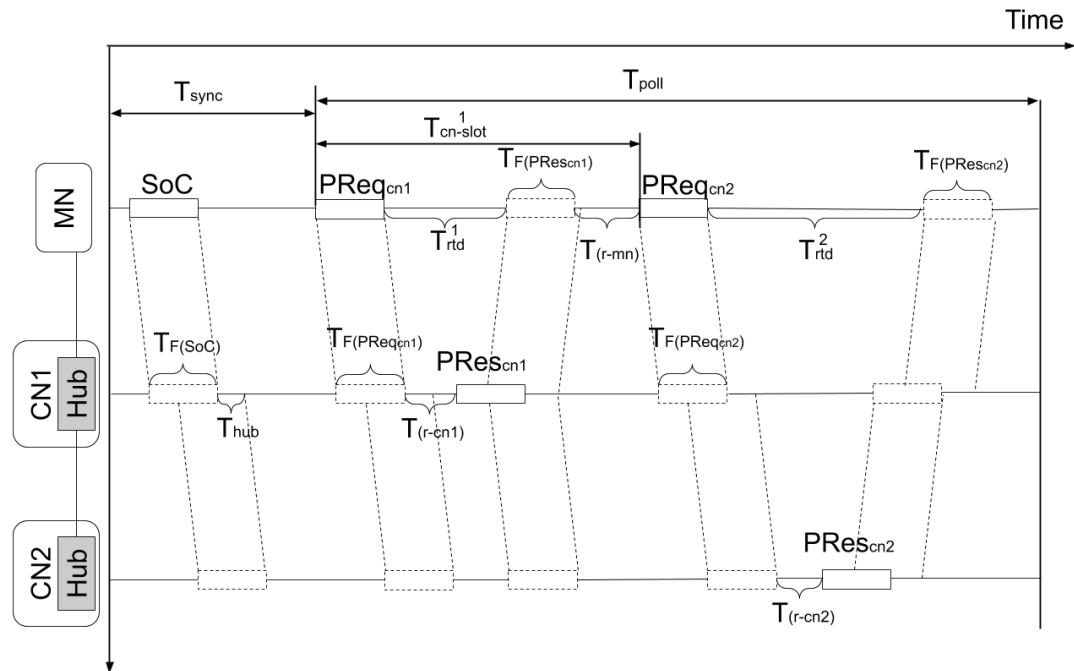


图 2.19 powerlink 等时同步阶段通信过程

在 POWERLINK 通信周期中，等时同步阶段用来传输周期性的实时数据，占据了绝大部分的通信周期。图 2.19 表示了一个基本 POWERLINK 网络的等时同步通信过程，该 POWERLINK 网络由一个主站和两个从站组成，各站点按照线型拓扑连接。等时同步阶段由同步阶段和轮询阶段组成，如公式 2.1 所示，其中  $T_{ip}$  代表等时同步阶段时间， $T_{sync}$  和  $T_{poll}$  分别代表同步阶段和轮询阶段的时间。 $T_{sync}$  由公式 2.2 表示， $T_{F(SoC)}$  为同步数据帧传输的时间。根据第 2.2.2 中对 POWERLINK 数据帧的分析，SoC 帧的大小为 64Byte，采用千兆以太网传输， $T_{F(SoC)}$  仅为  $0.512\mu s$ ， $T_W$  为等待从站接收并处理 SoC 帧的时间，一般来说  $T_W$  的大小取决于从站的配置。

$$T_{ip} = T_{sync} + T_{poll} \quad (2.1)$$

$$T_{sync} = T_{F(SoC)} + T_W \quad (2.2)$$

轮询阶段时长 ( $T_{poll}$ ) 可由公式 2.3 表示,  $T_{frames}$  为轮询阶段中所有数据帧的传输时间,  $T_{net}$  为由所有网络组件造成的延时, 网络组件包括主站、从站、HUB 和网线。如图 2.19 所示, PReq 和 PRes 数据帧在每个从站时隙都会被发送,  $T_{frames}$  可以用公式 2.4 表示。

$$T_{poll} = T_{frames} + T_{net} \quad (2.3)$$

$$T_{frames} = \sum_{i=1}^n (T_{F(PReq)}^i + T_{F(PRes)}^i) \quad (2.4)$$

$$T_{net} = \sum_{i=1}^n T_{rtd}^i + nT_{r-mn} \quad (2.5)$$

$T_{net}$  由公式 2.5 表示, 如图 2.19 所示,  $T_{rtd}^i$  为主站在第  $i$  个从站时间间隙内的往返延时,  $T_{r-mn}$  为主站的响应时间。对于图 2.19 所示的 POWERLINK 网络结构,  $T_{rtd}^1$  和  $T_{rtd}^2$  分别由公式 2.6 和 2.7 表示, 其中  $T_c$  为网线传播延时,  $T_{hub}$  为从站控制器的 HUB 延时,  $T_{r-cni}$  为第  $i$  个从站的处理时间。根据公式 2.6 和 2.7,  $T_{rtd}^i$  可用公式 2.8 所示, 当从站设备类型相同,  $T_{r-cni}$  都等于  $T_{r-cn}$ 。

$$T_{rtd}^1 = 2T_c + T_{hub} + T_{r-cn1} \quad (2.6)$$

$$T_{rtd}^2 = 4T_c + 3T_{hub} + T_{r-cn2} \quad (2.7)$$

$$T_{rtd}^i = 2iT_c + (2i - 1)T_{hub} + T_{r-cn} \quad (2.8)$$

$$\sum_{i=1}^n i = n(n + 1)/2 \quad (2.9)$$

我们假设所有的从站设备类型相同且使用等长网线连接, 即对于每个从站来说, 参数  $T_c$ ,  $T_{hub}$  和  $T_{r-cn}$  均相等。将公式 2.8 和公式 2.9 代入公式 2.5, 可得公式 2.10,  $T_{net}^{line}$  为线性拓扑结构下所有网络组件造成的延时。

$$\begin{aligned} T_{net}^{line} &= \sum_{i=1}^n [2iT_c + (2i - 1)T_{hub} + T_{r-cn}] + nT_{r-mn} \\ &= n(n + 1)T_c + n^2T_{hub} + n(T_{r-cn} + T_{r-mn}) \end{aligned} \quad (2.10)$$

将公式 2.4 的值和公式 2.10 代入公式 2.3, 可得到轮询阶段时长 ( $T_{poll}$ ) 的值, 如公式 2.11 所示:

$$T_{poll} = \sum_{i=1}^n (T_{F(PReq)}^i + T_{F(PRes)}^i) + n(n+1)T_c + n^2T_{hub} + n(T_{r-cn} + T_{r-mn}) \quad (2.11)$$

考虑到同步阶段时长和空闲阶时长  $T_{idle}$ , 可得循环周期  $T_{cycle\_time}$ , 如公式 2.12 所示:

$$T_{cycle\_time} = T_{F(SoC)} + T_W + \sum_{i=1}^n (T_{F(PReq)}^i + T_{F(PRes)}^i) + n(n+1)T_c + n^2T_{hub} + n(T_{r-cn} + T_{r-mn}) + T_{idle} \quad (2.12)$$

这里  $T_c$ ,  $T_{hub}$ 、 $T_{r-cn}$  和  $T_{r-mn}$  等参数与网线长度、主站从站的配置有关, 需要根据具体情况来确定。

同时需要另外注意的一个参数是各从站的通信时隙  $T_{cn-slot}^i$ , 我们可以根据  $T_{cn-slot}^i$  的值来配置 openCONFIGURATOR 来配置 PollResponse Timeout 参数, PollResponse Timeout 参数的值要大于  $T_{cn-slot}^i$  的值。如图 2.19 所示,  $T_{poll}$  为各从站的通信时隙  $T_{cn-slot}^i$  之和,  $T_{poll}$  也可以用公式 2.13 表示。根据图 2.19 所示,  $T_{cn-slot}^i$  可用公式 2.14 表示。

$$T_{poll}^i = \sum_{i=1}^n T_{cn-slot}^i \quad (2.13)$$

$$T_{cn-slot}^i = T_{F(PReq)}^i + T_{F(PRes)}^i + T_{rtd}^i + T_{r-mn} \quad (2.14)$$

## 2.5 POWERLINK 通信协议的仿真模拟

### 2.5.1 OMNeT++ 仿真器

通过对 POWERLINK 系统的通信过程理论分析, 我们得到通信周期的计算公式, 可以对 POWERLINK 系统的实时性能进行初步估算。为了进一步准确可靠的评估 POWERLINK 通信系统性能, 我们还发展了仿真建模的分析方法。POWERLINK 网络仿真基于 OMNet++ 网络模拟器展开, OMNeT++ (Objective Modular Network Testbed in C++) 是一个面向对象的模块化离散事件网络仿真框架<sup>[51]</sup>。作为面向对象的离散事件仿真器, OMNeT++ 主要用于通信网络和分布式系统的仿真, 具体可以应用于以下领域<sup>[52-54]</sup>:

- (1) 协议建模;
- (2) 有线和无线通信网络的建模;
- (3) 排队网络建模;

- (4) 多处理器等分布式硬件系统的建模;
- (5) 验证硬件架构;
- (6) 评估复杂软件系统的性能。

OMNeT++ 模型的基本元素是基于 C++ 实现的简单模块，简单模块中可以定义消息格式和消息传播的行为，简单模块之间通过门和连接相互通信，从而形成复杂模块，复杂模块可以由无限个简单模块构成，多个复杂模块组成整个 OMNeT++ 模型。在下面的图 2.20 中，方框代表简单的模块（灰色背景）和复合模块，小方块表示门，箭头表示连接<sup>[55]</sup>。

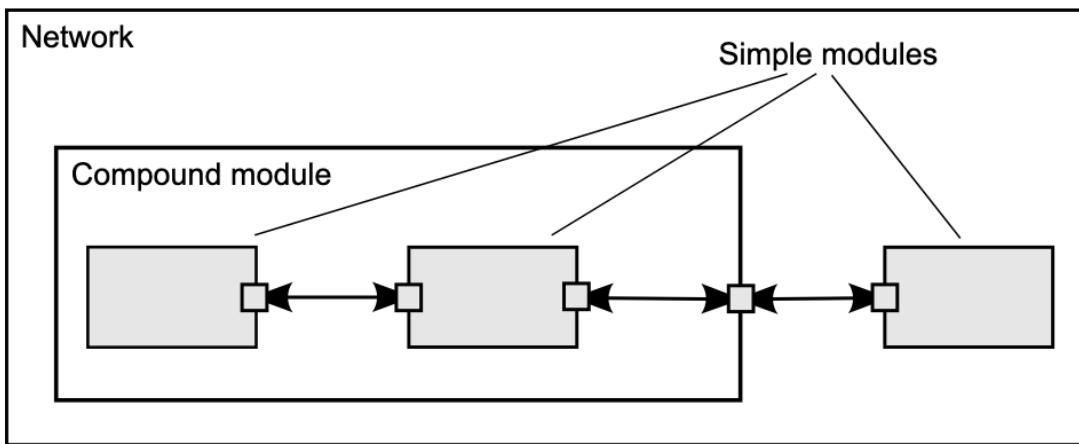


图 2.20 omnet 简单模块与复杂模块

OMNeT++ 的主要组成部分如下：

- (1) 仿真内核库：OMNeT++ 提供的仿真类库代码；
- (2) 网络拓扑描述文件：由 NED 语言编写的文件，既可以用来描述网络拓扑结构，又可以通过参数、门、信道连接、简单模块等来定义复杂模块；
- (3) 消息定义文件：用于定义数据报文格式；
- (4) 简单模块定义文件：简单模块的行为定义文件，包括 C++ 编写的 \*.cc 文件和 \*.h
- (5) 用户接口：该接口用于仿真运行时的测试，演示等工作。

在使用 OMNeT++ 建模中，最常使用的就是 INET 框架。INET 框架是开源的 OMNeT++ 的标准协议模型库，对应 OMNeT 的各种新版本 INET 框架还在不断更新中，INET 框架包含了常用的多种有线和无线协议的模型，具体到各种物理层模型、应用层模型等。在 POWERLINK 仿真建模中，我们使用的是 OMNeT++4.6 版本搭配 inet-2.6-b673687 版本。

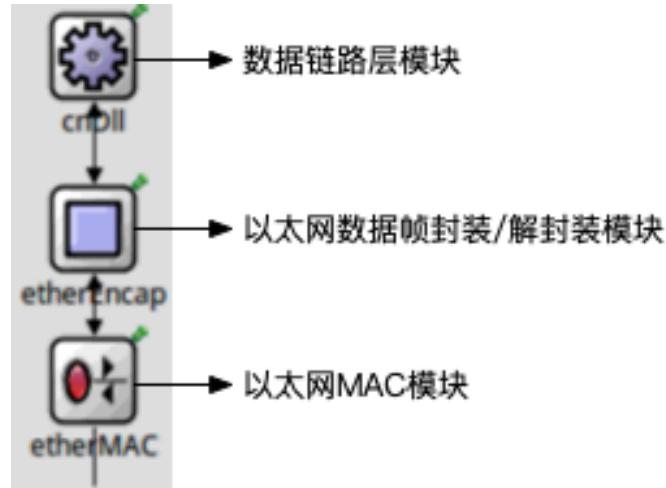


图 2.21 POWERLINK 通信节点模型

### 2.5.2 POWERLINK 通信节点建模

基于 OMNeT++ 仿真平台，我们对 POWERLINK 主站和从站进行了建模。节点模型如图 2.21 所示，由以下三个模块组成：

- (1) 以太网 MAC 模块：采用 INET 框架下的标准以太网 MAC 模块，支持千兆以太网通信；
- (2) 以太网数据帧封装/解封装模块：采用 INET 框架下的 EtherEncap 模块，用于 EthernetII 数据帧的封装和解封装；
- (3) 数据链路层模块：按照 POWERLINK 的 SCNM 机制定义了模块行为。

我们在配置文件中定义了各节点模型的通信参数，主要的配置文件包括 omnetpp.ini 和 xdc\_cn\_std.xml 文件，其中 omnetpp.ini 文件中定义了 POWERLINK 模型的通信周期、主站从站的响应时间以及从站的节点号和响应数据大小，具体如下所示：

```

1 # The configuration of the MN
2 POWERLINK 模型的通信周期：
3 **.MN.mnDll.cycleLenMean = 500us
4 主站响应时间：
5 **.MN.mnDll.responseTimeMean = 1us
6
7 # The configuration of the CNs
8 从站响应时间：
9 **.CN.cnDll.responseTimeMean = 1us
10 1号从站的节点号以及响应数据大小：
11 **.CN1.cnDll.nodeId = 1
12 **.CN1.cnDll.presActPayload = 1
13
14 2号从站的节点号以及响应数据大小：
15 **.CN2.cnDll.nodeId = 2

```

```

16 **.CN2.cnDll.presActPayload = 1
17
18 # The configuration of the hub
19 hub 延时：
20 **.hubDelayMean = 50ns

```

xdc\_cn\_std.xml 文件中则定义了各从站的通信间隙 (PResTimeout) 以及接收来自主站请求数据帧的大小 (PReqActPayload)，这里“PResTimeout”参数的意义与 openCONFiGURATOR 中设置的 PollResponse Timeout 网络参数意义相同。

```

1 <Nodes>
2   <Node NodeId="1" PResTimeout="6000ns" PReqActPayload="1" />
3   <Node NodeId="2" PResTimeout="9000ns" PReqActPayload="1" />
4 </Nodes>

```

除了对通信节点建模之外，在.msg 文件中定义四种 POWERLINK 数据帧结构。

```

1 cplusplus {{
2 // POWERLINK node reserved addresses
3 const int C_ADR_INVALID = 0;
4 const int C_ADR_MN_DEF_NODE_ID = 240;
5 const int C_ADR_DUMMY_NODE_ID = 252;
6 const int C_ADR_DIAG_DEF_NODE_ID = 253;
7 const int C_ADR_RT1_DEF_NODE_ID = 254;
8 const int C_ADR_BROADCAST = 255;
9 // POWERLINK messages header size (in bytes)
10 const int EPL_SOC_HEADER_BYTES = 46;
11 const int EPL_PREQ_HEADER_BYTES = 10;
12 const int EPL_PRES_HEADER_BYTES = 10;
13 const int EPL_SOA_HEADER_BYTES = 46;
14 const int EPL_ASND_HEADER_BYTES = 4;
15 } }
16
17 enum eMsgType
18 {
19     MsgTypeNonPowerlink = 0x00;
20     MsgTypeSoc = 0x01;
21     MsgTypePreq = 0x03;
22     MsgTypePres = 0x04;
23     MsgTypeSoa = 0x05;
24 }
25
26 packet POWERLINKHdr {
27     int MsgType enum(eMsgType);
28     int Dst;
29     int Src;

```

```

30 }
31
32 packet SocFrame extends POWERLINKHdr {
33     MsgType = MsgTypeSoc;
34     Dst = C_ADR_BROADCAST;
35     Src = C_ADR_MN_DEF_NODE_ID;
36     bool MC;
37     bool PS;
38     long NetTime;
39     long RelativeTime;
40 }
41
42 packet SoaFrame extends POWERLINKHdr {
43     MsgType = MsgTypeSoa;
44     Dst = C_ADR_BROADCAST;
45     Src = C_ADR_MN_DEF_NODE_ID;
46     int NMTStatus;
47     int RequestedServiceID;
48     int RequestedServiceTarget;
49     int EPLVersion;
50 }
51
52 packet PreqFrame extends POWERLINKHdr {
53     MsgType = MsgTypePreq;
54     Src = C_ADR_MN_DEF_NODE_ID;
55     bool MS;
56     bool EA;
57     bool RD;
58     int PDOVersion;
59     int Size;
60 }
61
62 packet PresFrame extends POWERLINKHdr {
63     MsgType = MsgTypePres;
64     Dst = C_ADR_BROADCAST;
65     int NMTStatus;
66     bool MS;
67     bool EN;
68     bool RD;
69     int PR;
70     int RS;
71     int PDOVersion;
72     int Size;
73 }

```

我们只需要通过编写.ned 文件来定义相应系统拓扑结构，代入主从节点模

型以及相应参数即可对相应的 POWERLINK 通信系统进行模拟。

# 第3章 EPICS环境下基于POWERLINK的分布式IO系统

在粒子加速器控制领域，EPICS是目前国际上应用最广泛的控制系统开发工具包，而开源实时以太网POWERLINK则广泛应用于工业控制领域，将POWERLINK和EPICS结合起来，可拓宽EPICS的应用范围。我们基于FPGA开发了支持千兆POWERLINK通信协议的控制器，开发了相关的EPICS设备驱动程序，提出了两套EPICS环境下基于POWERLINK的分布式I/O系统方案，第一套方案的主站采用PC实现，我们称为主站PC方案，其中第二套方案是在主站PC方案基础上的升级版本，主站从站均基于FPGA实现，我们称之为全站FPGA方案。本章将分别介绍两套方案的系统架构设计、软硬件设计与开发，并分别搭建了相应的测试系统。然后，基于两套测试系统的具体情况，我们进一步完善了理论计算方法，并验证了仿真模拟方法的可行性。

## 3.1 主站PC方案的系统设计与开发

主站PC方案的主体部分是基于POWERLINK协议的控制网络，POWERLINK网络负责收集各节点的I/O信号，在主站经过算法处理后，通过相应的从站输出信号。EPICS的核心输入输出控制器（Input/Output Controller, IOC）应用负责监控POWERLINK网络中各站点的状态，并通过Channel Access协议将收集的信息发布至控制系统局域网中。

### 3.1.1 系统架构设计

主站PC方案的系统架构如3.1图所示，系统的网络拓扑为菊花链结构。该系统中有1个主站（MN）和10个从站（CN）组成，所有站点按照站点号从小到大的顺序依次连接，站点之间的通信基于千兆POWERLINK。主站是运行RT-Linux操作系统的PC，openPOWERLINK（oplk）程序和IOC应用程序也在运行在此PC上，其中openPOWERLINK工作在内核空间。从站是基于FPGA的控制器，支持千兆POWERLINK通信和集线器（HUB）功能。每个从站控制器都有多个DI和DO通道，可用于信号采集和输出<sup>[56]</sup>。

### 3.1.2 EPICS设备驱动程序的开发

EPICS设备驱动程序的功能是将POWERLINK网络中传输的数据转换为EPICS IOC中各种记录（Record）。EPICS IOC的软件结构如图3.2中“IOC Ap-

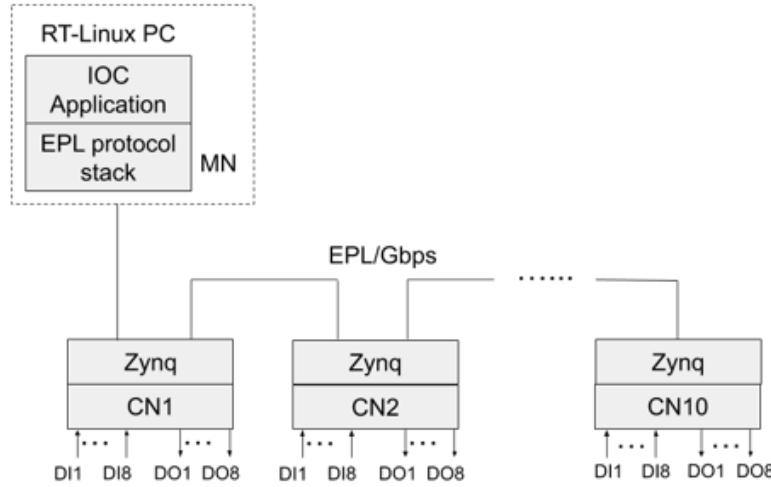


图3.1 方案1 系统硬件架构图

plication”部分所示，IOC以运行时数据库（Runtime Database）为核心，运行时数据库由记录（Record）组成，记录中保存着与IO信号有关的所有数据，如信号的值、工程单位、处理时间、报警范围等。记录通过Record Support、Device Support及Driver Support来与POWERLINK网络交换信息。

驱动程序的开发工作主要是按照EPICS的接口规范来编写Device Support，而Driver Support负责与POWERLINK网络通信。具体来说，IOC应用程序与POWERLINK网络中的主站建立通信，以获取POWERLINK网络中各从站的IO数据。IOC的任务是监控POWERLINK网络中各节点的状态，所以IOC与POWERLINK主站之间的通讯实时性要求并不高。在主站PC方案所设计的系统里，oplk和IOC作为两个应用程序运行在同一台PC上，所以我们采用基于进程间Socket（IPC Socket）的通信方式来实现IOC应用程序与POWERLINK主站程序的通信。

主站的软件架构如图3.2所示，由oplk和IOC应用程序组成。oplk通过“Data Link Layer”收集各从站的DI信号，这些信号被发送到“Application Layer”中名为“Input Data”缓存区中。同时作为Socket服务端，“Application Layer”会将“Input Data”缓存区中的数据传输到IOC应用程序。“Application Layer”中名为“Output Data”的缓存区用于接收来自IOC应用程序的控制参数等命令，并向下将控制参数发送到相应的从站。

IOC中的EPICS驱动程序由“Driver Support”和“Device Support”组成。作为Socket客户端，“Driver support”中有两个缓存区，用于存储与oplk程序交换的数据。在这些缓存区中，“Recv Buf”缓存区用于接收来自oplk程序的数据，“Send Buf”缓存区用于存储发送至oplk程序的控制参数。“Device Support”的功能是将两个缓存区中的数据转化成EPICS中的记录，目前“Device Support”

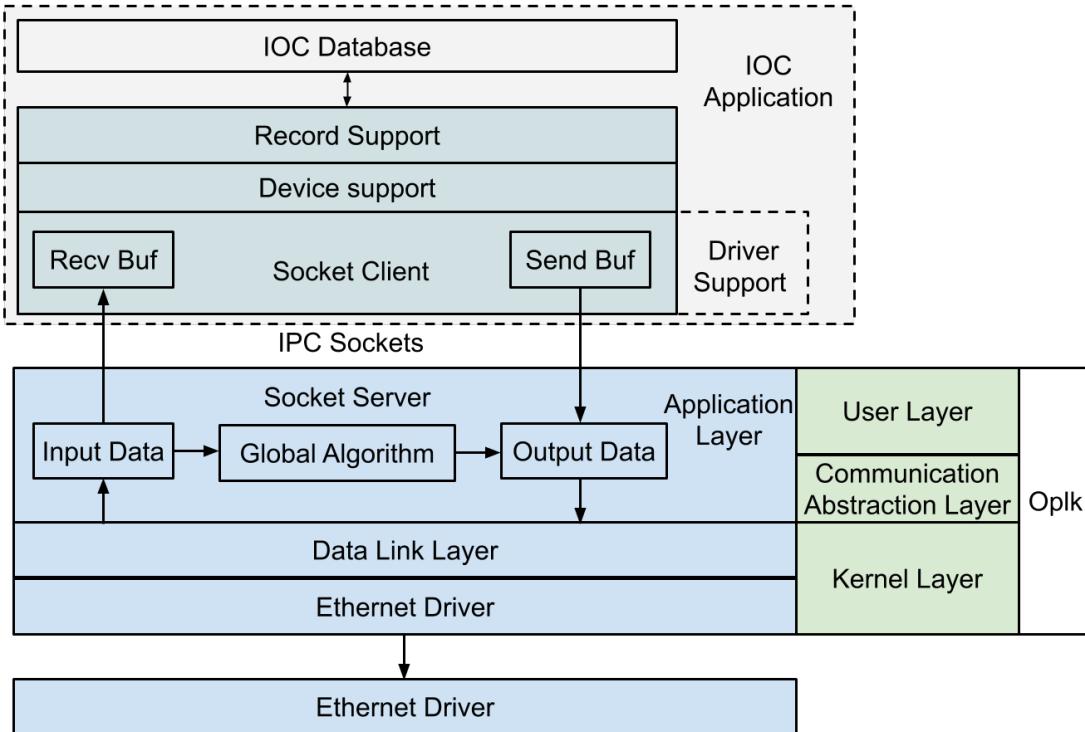


图 3.2 主站的软件架构图

支持标准的 EPICS DI 和 DO 记录。

### 3.1.3 从站控制器的设计与开发

FPGA 平台具有较高的实时性和灵活性，是实现 POWERLINK 协议较为理想的平台，也是目前普遍采用的方式。基于 FPGA 实现 POWERLINK 的方案有两种，软核方案是在 FPGA 中通过软核的形式来运行 openPOWERLINK 程序，HDL 方案是采用纯硬件描述语言 Verilog HDL 来实现 POWERLINK 的协议栈。具体来说，HDL 方案可以采用 POWERLINK 协议分离的方式来实现，数据链路层、物理层基于 FPGA 来实现，应用层基于 ARM 处理器来实现。这种设计既可以利用 FPGA 来达到较高的硬实时性，又可以利用 ARM 处理器来兼容各种应用层协议。

软核方案的技术细节和性能测试已在第 2.3.2 节中详细介绍，对比软核方案，HDL 方案有以下特点和性能优势：

- (1) 采用硬件语言 Verilog HDL 来实现，无需软核，无需外扩 RAM，即可以达到硬实时的性能。
- (2) 实时性高，最短通信周期可以达到几十  $\mu\text{s}$ ，抖动可以达到数十  $\text{ns}$ 。若采用 MCU 软核的方式，很难达到这样的性能指标。
- (3) 对比第 2.3.2 节中软核方案的通信过程，HDL 方案使用硬件加速，在 FPGA 中实现自动回复功能。当 PReq 数据帧进入到 FPGA 中的 MAC 时，MAC 会将准备

好的上一周期的PRes数据快速发送到网络上。当使用了自动回复功能后，PReq数据帧和PRes数据帧之间的时间间隔可以达到几个 $\mu\text{s}$ ，从而缩短通信周期，提高实时性。

(4) 兼容不同的应用层，虽然标准POWERLINK协议的应用层为CANopen，但是不同行业有不同的标准和习惯。采用此方案，可以仅使用Verilog HDL实现的数据链路层，而自定义应用层的协议。

(5) 兼容不同的物理层，在保持数据链路层一致的情况下，此方案可以满足100Mbps、1000Mbps以太网。

## 1. 硬件介绍

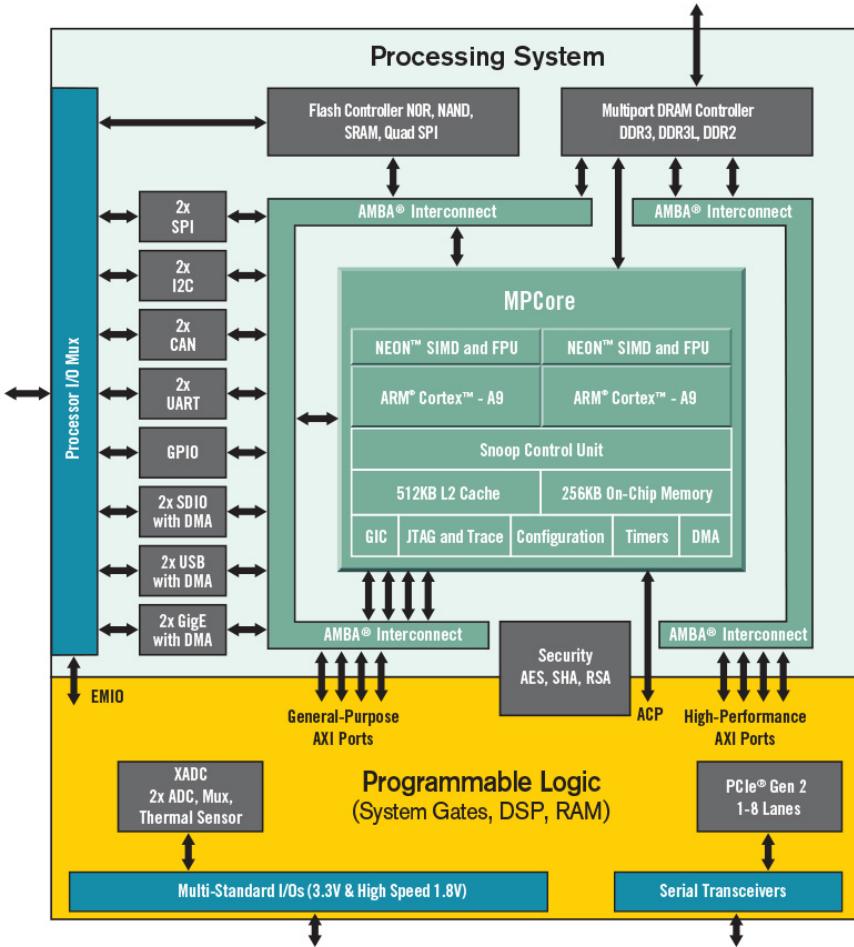


图3.3 Xilinx Zynq-7000芯片的架构图

在核心片的选择上，采用Xilinx Zynq-7000型号的芯片来实现POWERLINK协议。该芯片的结构如图3.3所示，Zynq-7000器件在硬件上配备了双核ARM Cortex-A9处理器，该处理器与基于28nm Artix-7的可编程逻辑集成，在软件上兼具ARM处理器的软件可编程性与FPGA的硬件可编程性。由于其高度的灵活性和较小的硬件占用空间，Zynq是自动化项目中小型高性能设备的理想平台<sup>[57]</sup>。

我们可以利用Zynq芯片的ARM处理器实现POWERLINK协议的应用层，利用FPGA实现POWERLINK协议的数据链路层和物理层，从而达到硬实时的性能。

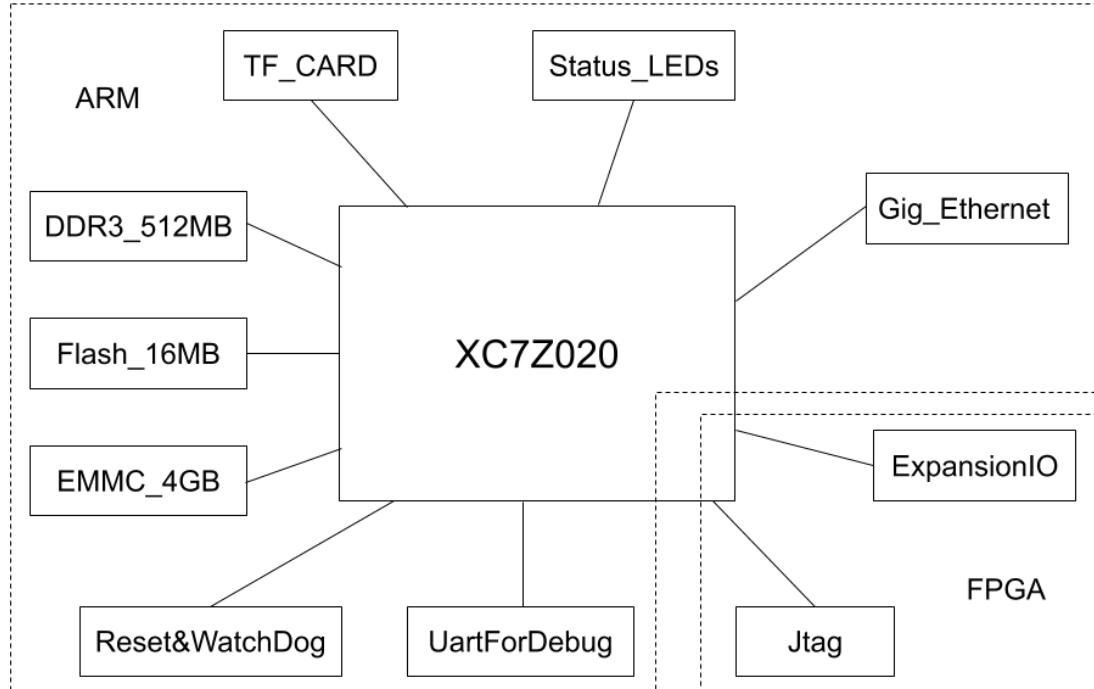


图3.4 基于Zynq的核心板硬件架构图

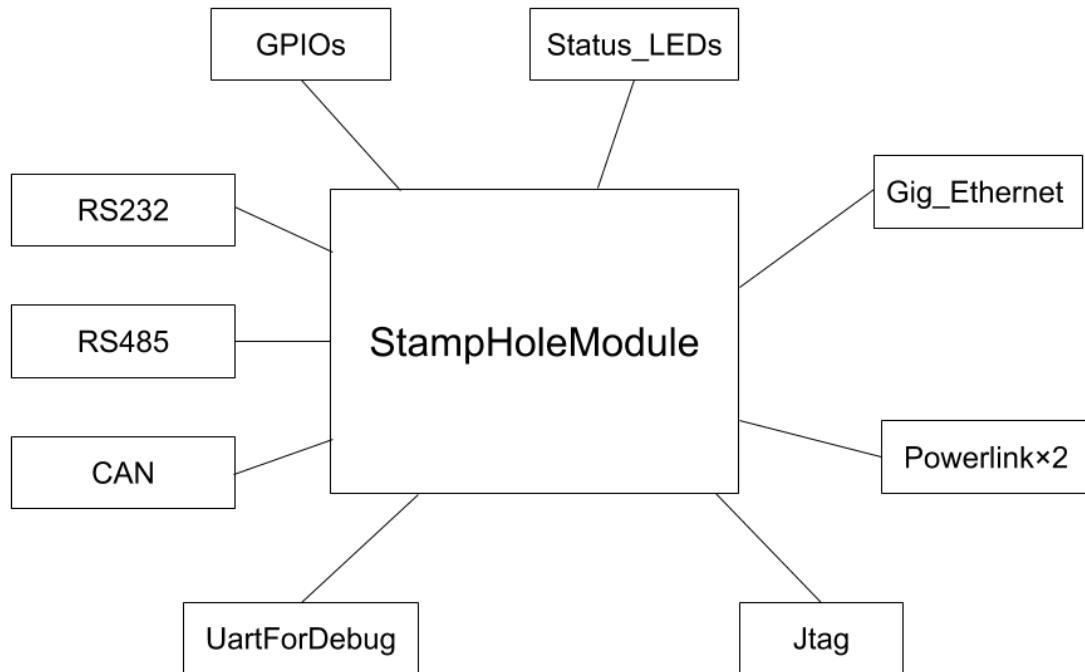


图3.5 基于Zynq的底板硬件架构图

基于核心片Zynq，设计了由核心板和底板组成的控制器，核心板通过相应引脚与底板相连接，核心板的硬件架构如图3.4所示，底板的硬件架构如图3.5所

示，核心板硬件系统的各部分为：

- (1) 核心片：POWERLINK 网络通信的核心部件，选用 Xilinx Zynq-7000 家族的芯片，具体型号为 XC7Z020，逻辑单元数 85k。
- (2) 外部存储：外部存储连接到 ARM 硬核上，包括 512MB 的 DDR3、16MB 的 Qspi\_Flash、4GB 的 EMMC 和 TF\_CARD，用来提供 POWERLINK 程序的运行空间和存储空间。
- (3) Reset&WatchDog：外置看门狗计时器，用来复位程序。
- (4) Gig\_Ethernet：千兆以太网 PHY 芯片，通过此以太网接口与 EPICS IOC 应用程序程序通信。
- (5) JTAG 接口：POWERLINK 协议程序的下载和调试接口。
- (6) Expansion IO：扩展 IO 引脚，用于连接底板。

底板硬件系统的各部分为：

- (1) POWERLINK×2：用于 POWERLINK 通信的 HUB 的两个千兆以太网接口。
- (2) GPIOs：提供 DI、DO 接口。
- (3) 本地通讯接口：提供了 RS232、RS484 以及 CAN 本地通讯接口。

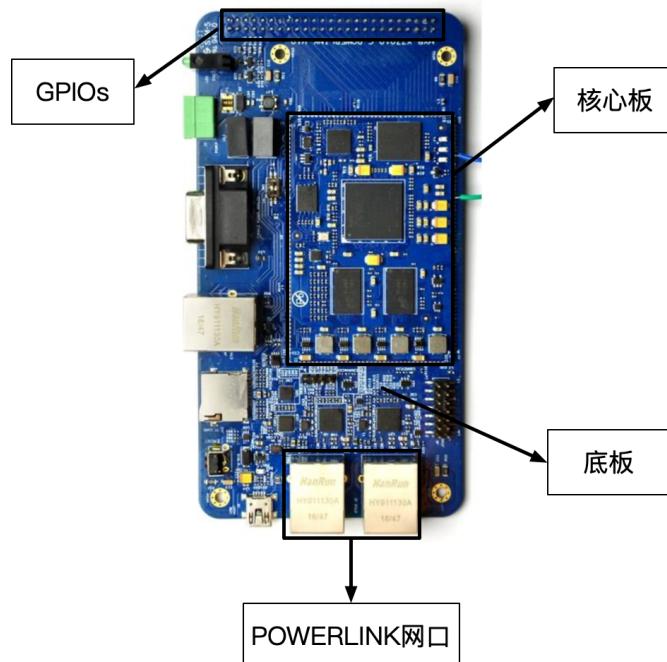


图 3.6 控制器电路板照片

图 3.6 所示的是控制器电路板实物，可从照片中看出 GPIO 接口、两个 POWERLINK 网口。

## 2. 软件设计

基于Zynq的硬件架构，设计了如图3.7所示的软件架构。首先介绍与POWERLINK通信有关的软件设计。从站POWERLINK协议的软件架构由三部分组成：基于ARM处理器的POWERLINK应用层、基于FPGA的POWERLINK数据链路层和物理层。

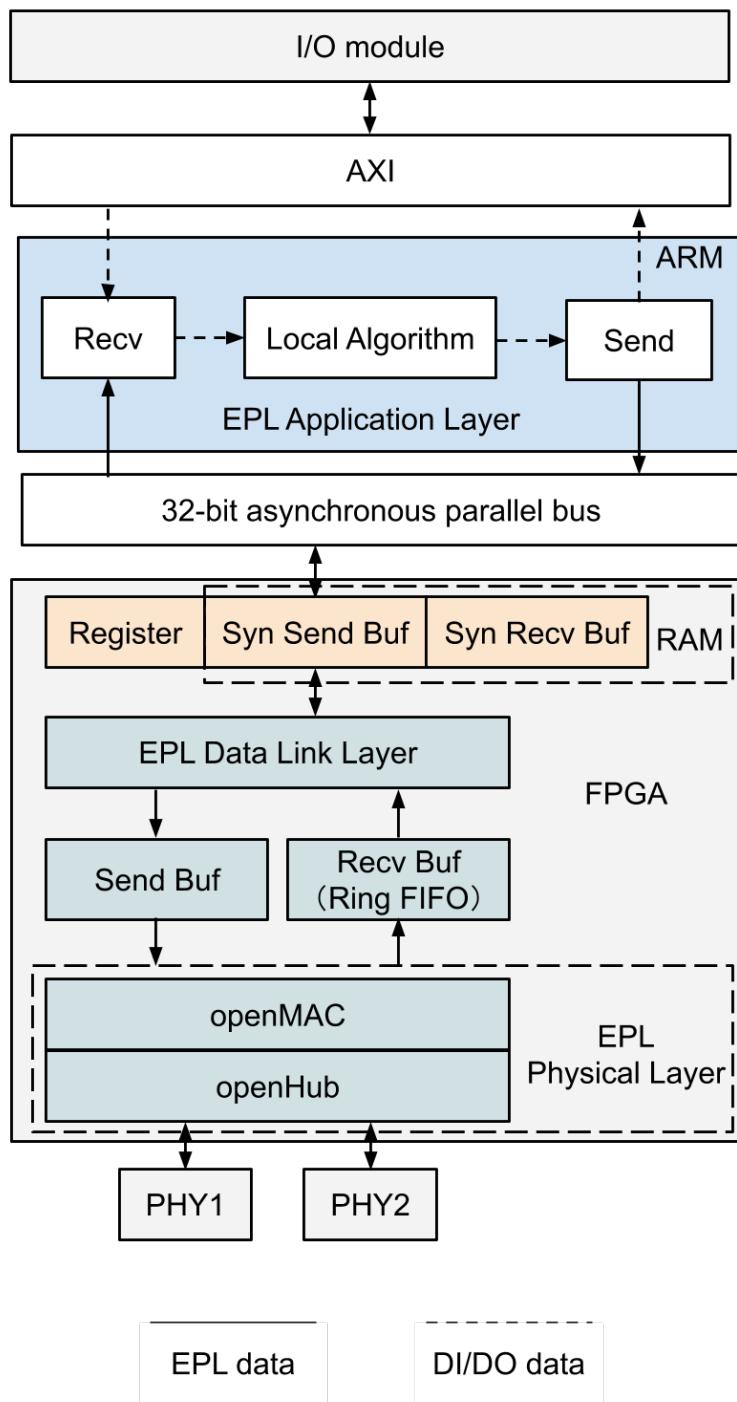


图3.7 基于Zynq的控制器软件架构图

- **物理层**

POWERLINK 物理层采用 Verilog HDL 实现，由 openHub 和 openMAC 两部分组成。openMAC 支持千兆以太网 PHY 芯片，双口 openHUB 可以以菊花链的形式来连接两个相邻的从站。

- **数据链路层**

POWERLINK 的数据链路层状态机和 NMT 网络管理等采用 Verilog HDL 来实现。

- **应用层**

POWERLINK 协议的应用层基于 ARM 架构采用 C 语言实现。除了可以使用标准 POWERLINK 的 CANopen 应用层以外，还支持其他任意的应用层。

- **API 接口**

应用层与数据链路层之间的接口为 API 接口。该接口采用寄存器和双口 RAM 实现，应用层通过 32 位并行总线来访问寄存器和双口 RAM 中的数据。

数据传输的流程和细节如下：

- (1) POWERLINK 应用层通过 32 位并行总线将数据写入寄存器和“Syn Buf”缓存区，缓存区采用双口 RAM 实现从而避免数据链路层和应用层同时对 RAM 操作造成的冲突。
- (2) 在数据链路层和应用层之间，有“Sync Recv Buf”和“Sync Send Buf”两个缓存区，分别用于保存数据链路层收到的同步数据和下个周期要发送的同步数据。同步数据是周期性传输的数据，在每个周期开始的时候，数据链路层都会给应用层一个同步中断信号，触发应用层的同步回调函数对“Sync Send Buf”和“Sync Recv Buf”两个缓存区中的数据进行读写操作。
- (3) 数据链路层组建要发送的数据包，并写入 MAC 层的“Send Buf”缓存区中，然后启动发送。发送完成，通过寄存器的标志位来反馈发送是否正确。
- (4) MAC 层接收到数据时，将其放到 MAC 层的“Recv Buf”缓存区中，该缓存区为环形，可以存放 2 个最大以太网的帧。当收到一个完整且正确的 PReq 数据帧，即该数据帧的目标节点号与本节点相同，则通知数据链路层进行处理，将数据帧的有效数据部分从 MAC 的“Recv Buf”拷贝到“Sync Recv Buf”中。如果接收到的数据有问题，则丢弃本帧数据，并将重新设置 MAC 的接收设置，为下一次接收做准备。当处理完毕，通知 MAC 层释放接收缓冲区“Recv Buf”，用来保存后面的数据帧。

除了完成 POWERLINK 协议通信，从站还需要完成 IO 信号的采集、本地处理和输出。如图 3.7 所示，IO 模块的数据存储在相应的外部 RAM 中，应用层通过 AXI 总线来读写 RAM 中的 IO 数据，应用层的本地算法模块用来对 IO 数据进行本地处理。

### 3.1.4 测试系统搭建

为测试主站PC方案设计系统的实时性能，基于第3.1.1节设计的系统架构，我们搭建了相应的测试系统，以下将分别介绍系统的硬件组成和软件开发。

#### 1. 测试系统的硬件组成



图 3.8 主站 PC 方案的系统照片

按照第3.1.1节设计的主站PC方案，我们搭建了如图3.8所示的测试系统。照片上可以看到1台PC和10台FPGA控制器。EPICS IOC应用程序和oplk主站程序运行在在同一台PC上，PC上安装了带有内核实时补丁3.10的Centos7系统。该PC的网卡型号为intel 82573，oplk主站程序通过该网卡驱动，运行在在内核模式下。

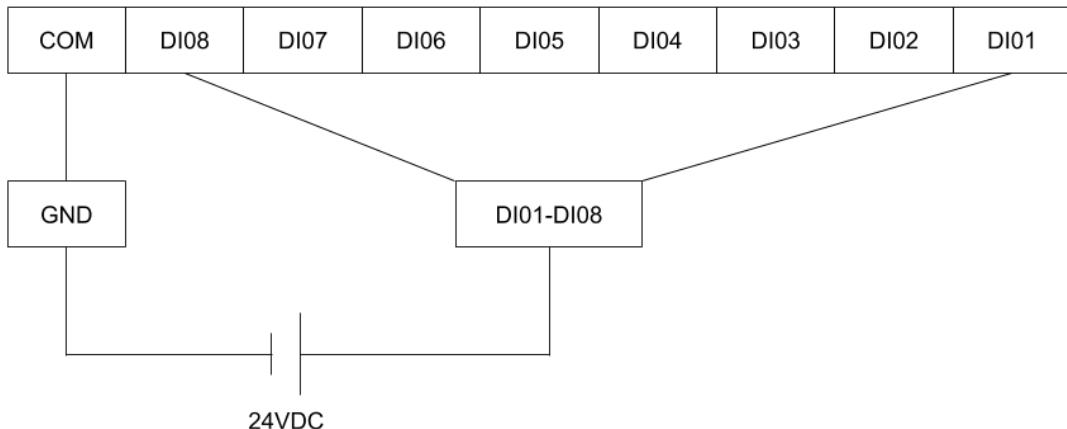


图 3.9 从站控制器 DI 接口

测试系统共有10个从站，每个从站都是基于Zynq的控制器。控制器对外提供了8个DI和8个DO通道，DI接口如图3.9所示，公共端（COM0端口供DI输入使用，输入设备的地需要与此处共地形成回路；DO接口如图3.10所示，为

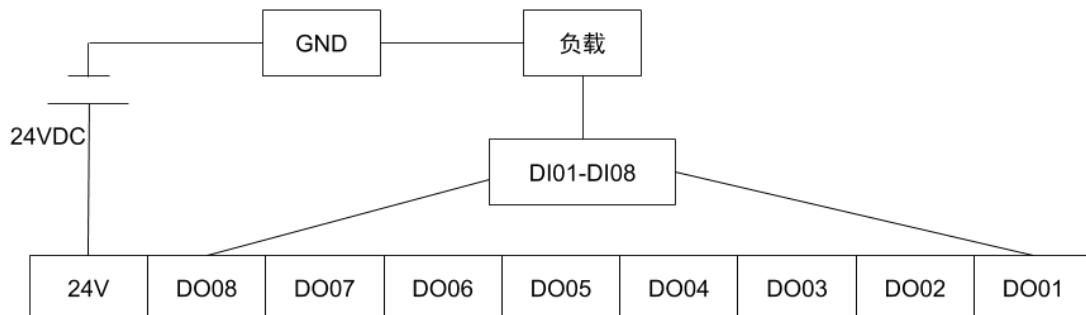


图 3.10 从站控制器 DO 接口

源型输出，24V 输入端为 DO 输出负载所接的正 24V 电压。另外，2 个千兆以太网端口用于连接相邻的从站，3 个状态指示灯分别显示控制器的电源状态，错误状态和操作状态，灯亮表示状态正常。

## 2. 测试系统的软件开发

测试系统的软件开发包括主站程序和从站程序的开发。基于 Zynq 的从站控制器程序按照图 3.7 所示的软件架构开发，除了完成 POWERLINK 通信以外，在 POWERLINK 应用层加入了本地算法功能，从站控制器可以直接对 DI 信号进行处理，并输出相应 DO 信号。主站程序包括来 EPICS IOC 应用程序和 oplk 主站程序，IOC 应用程序的开发按照图 3.2 所示的软件架构展开，oplk 主站程序的应用层加入了全局算法功能，主站通过全局算法处理来自各从站控制器的 DI 信号，并通过相应的从站输出相应 DO 信号。oplk 主站应用层程序和 Zynq 控制器应用层程序类似，程序框架如下：

```

1 while(1)
2 {
3     // 处理出来 POWERLINK 网络状态机跳转， 网络配置等
4     ret = oplk_process();
5
6     // 主站 UDP Server 通信程序
7     udp_send_data((const char *) UDP_TX_Buffer, int data_len);
8     udp_recieve_data(const char *frame, int data_len);
9
10    // 从站读写 DI/DO 和本地算法程序
11    CHA_DI_read(u32 Input_Pin);
12    local_process_data ();
13    FPGA_DO_write(u32 Output_Pin, u32 Data);
14 }
15
16 // 同步回调函数，由 SoC 触发
17 void syncCb(void)
18 {
19     // 负责读写同步数据和主站全局算法程序

```

```

20 pdou_copyRxPdoToPi();
21 global_process_data();
22 pdou_copyTxPdoFromPi();
23 }

```

程序主要由 while 循环程序和 POWERLINK 通信同步回调函数组成。while 循环程序一直运行，用来处理 POWERLINK 异步事件，状态机跳转，网络配置等，对于主站我们在 while 循环程序里添加了 UDP Server 程序；对于从站，我们在 while 循环程序里添加了 DI/DO 读写和本地算法程序。通信同步回调函数“synCb ()”由 SoC 触发，用于处理周期性同步数据，我们在此添加了全局算法程序。

### 3.1.5 系统性能测试与分析

主站 PC 方案中的 IOC 负责监控 POWERLINK 网络中各节点状态，并不直接参与 POWERLINK 通信，所以系统实时性能即是系统中 POWERLINK 网络部分的实时性能。我们对测试系统的 POWERLINK 网络进行了一系列性能测试，其中包括通信周期、响应时间的实时性能和主站响应时间，下面将详细介绍测试细节和测试结果。

#### 1. 系统通信周期测试

通信周期是衡量实时系统性能的重要参数，短时间周期内持续稳定运行对于实时性系统至关重要。我们使用 openCONFIGURATOR 网络组态工具来配置不同的 POWERLINK 通信周期，利用 ProfiShark 1G Ethernet Troubleshooter 以太网诊断工具来测试原型系统在不同通信周期下的运行情况，从而确定系统最短通信周期。ProfiShark 1G 是一款专用于千兆以太网监测的故障诊断仪，硬件时间戳仅为 8ns，可以抓取高实时性的 POWERLINK 通信数据，将抓包结果输出为可被 Wireshark 解析的 pcap 格式文件。

我们对主站 PC 方案系统进行了通信周期测试，测试结果如下：图 3.11 所示的是主站 PC 方案系统的数据帧抓取结果，通信周期可根据两个连续 SoC 帧之间的时间隔计算得出的，如图 3.11 所示。我们抓取来超过 12k 个连续的 POWERLINK 帧，并将通信周期的概率密度分布绘制在图 3.12 中。图 3.12 表明，大多数情况下的通信周期都非常接近系统的通信周期设定值  $275\mu s$ ，通信周期周期的最大偏移约为  $12\mu s$ 。

#### 2. 响应时间测试

响应时间是衡量系统实时性能的重要参数。在我们的测试中，响应时间是指 DI 信号到达系统与系统输出 DO 信号之间的时间间隔。测试系统中具有本地响应和全局响应两种响应机制，本地响应指是从站本身对于 DI 信号直接做出的响

0.450670000	IntelCor_03:e3:2d	EPLv2_SoC	POWERLINK	64 240->255 SoC		
0.45067128	IntelCor_03:e3:2d	CamilleB_56:78:01	POWERLINK	64 240-> 1 PReq		
0.450678272	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 1->255 PRes		
0.45069864	IntelCor_03:e3:2d	CamilleB_56:78:01	POWERLINK	64 240-> 2 PReq		
0.450697992	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 2->255 PRes		
0.450716784	IntelCor_03:e3:2d	CamilleB_56:78:01	POWERLINK	64 240-> 3 PReq	$T_{(r-mn)}+T_f(PReq)$	
0.450717920	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 3->255 PRes		
0.450738104	IntelCor_03:e3:2d	CamilleB_56:78:01	POWERLINK	64 240-> 4 PReq		
0.450739232	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 4->255 PRes		
0.450754880	IntelCor_03:e3:2d	CamilleB_56:78:01	POWERLINK	64 240-> 5 PReq		
0.450756008	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 5->255 PRes		
0.450779056	IntelCor_03:e3:2d	CamilleB_56:78:01	POWERLINK	64 240-> 6 PReq		
0.450780192	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 6->255 PRes		
0.450804264	IntelCor_03:e3:2d	CamilleB_56:78:01	POWERLINK	64 240-> 7 PReq		
0.450805400	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 7->255 PRes		
0.450825032	IntelCor_03:e3:2d	CamilleB_56:78:01	POWERLINK	64 240-> 8 PReq		
0.450826168	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 8->255 PRes		
0.450847248	IntelCor_03:e3:2d	CamilleB_56:78:01	POWERLINK	64 240-> 9 PReq		
0.450848384	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 9->255 PRes		
0.450868352	IntelCor_03:e3:2d	CamilleB_56:78:01	POWERLINK	64 240-> 10 PReq		
0.450870656	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 10->255 PRes		
0.450889288	IntelCor_03:e3:2d	EPLv2_SoA	POWERLINK	64 240->255 SoA		
0.450945080	IntelCor_03:e3:2d	EPLv2_SoC	POWERLINK	64 240->255 SoC		

图3.11 测试系统的POWERLINK数据帧抓取结果

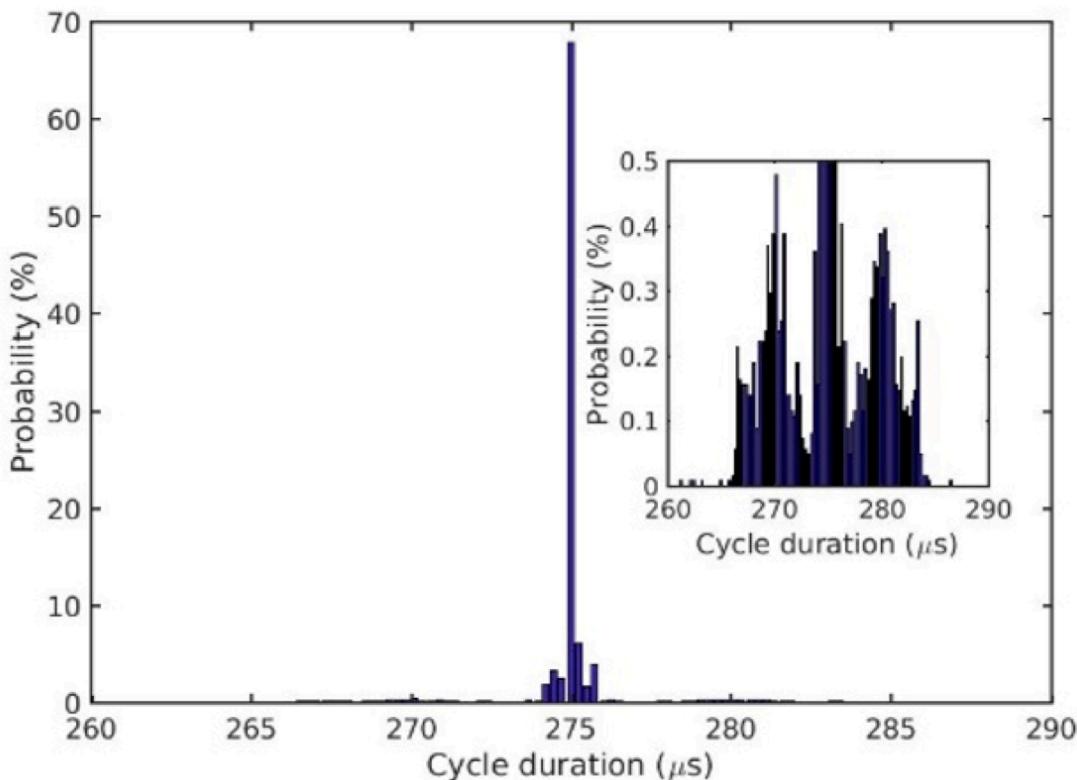


图3.12 测试系统的POWERLINK通信周期概率密度分布

应，全局响应是指DI信号经过POWERLINK网络传输，被主站处理之后，系统做出的响应。我们对系统进行了本地响应时间和全局响应时间的测试，测试结果和分析如下：

我们首先对从站控制器进行本地响应时间的测量，本地响应的过程如下：

1. 系统从站的GPIO读取输入信号，并将输入值以二进制形式存储在外部RAM中；

2. 运行在 ARM 上的应用层程序通过 AXI 总线读取并处理 RAM 中的输入数据；
3. 处理结果通过 AXI 总线写入相应的外部 RAM 中并通过 GPIO 输出信号。

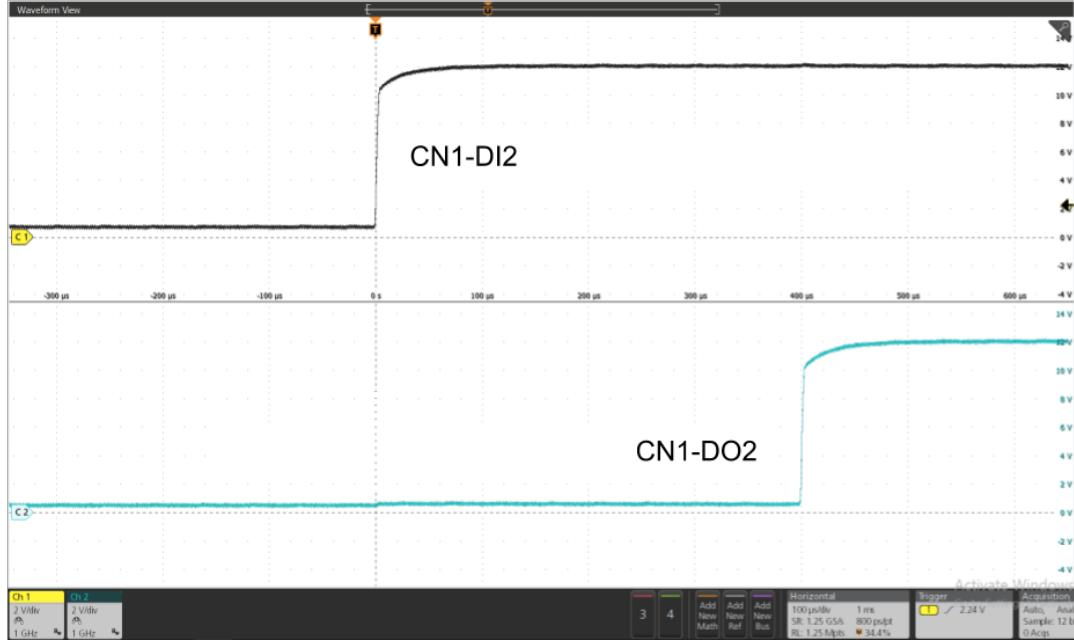


图 3.13 本地响应时间测试结果

我们将 1 号从站控制器的输入和输出信号连接到示波器上，以输入和输出信号的上升沿时间差来表示本地响应时间。测试结果如图 3.13 所示，Ch1（黑线）代表 1 号从站控制器的输入信号；Ch2（青色线）代表 1 号从站控制器的输出信号，响应时间约为  $400\mu\text{s}$ 。这个时间延迟包括从站控制器输入/输出接口电路中的光电耦合器的延时和 ARM 读取处理数据的时间。由于 ARM 的时钟频率为 125MHz，AXI 总线传输的数据宽度为 32bit，所以运行在 ARM 上的应用层程序通过 AXI 总线读取处理数据的时间不到  $1\mu\text{s}$ ，所以本地响应时间主要是光电耦合器导致的延时。

全局响应时间 ( $T_{global-response}$ ) 由 POWERLINK 网络通信时间 ( $T_{com}$ ) 和从站处理输入/输出信号的时间 ( $T_{CN-process}$ ) 组成，可以用公式 3.1 来表示。POWERLINK 网络通信时间包括用于封装和解封装数据帧的时间、主站处理数据的时间、电缆中数据传输的时间以及 HUB 中的时间延迟。从站处理输入/输出信号的时间可以用公式 3.2 来表示，包括了光耦合器的延时 ( $T_{coupler}$ ) 和输入信号等待进入 POWERLINK 通信的时间  $T_{wait}$ )。

$$T_{global-response} = T_{com} + T_{CN-process} \quad (3.1)$$

$$T_{CN-process} = T_{coupler} + T_{wait} \quad (3.2)$$

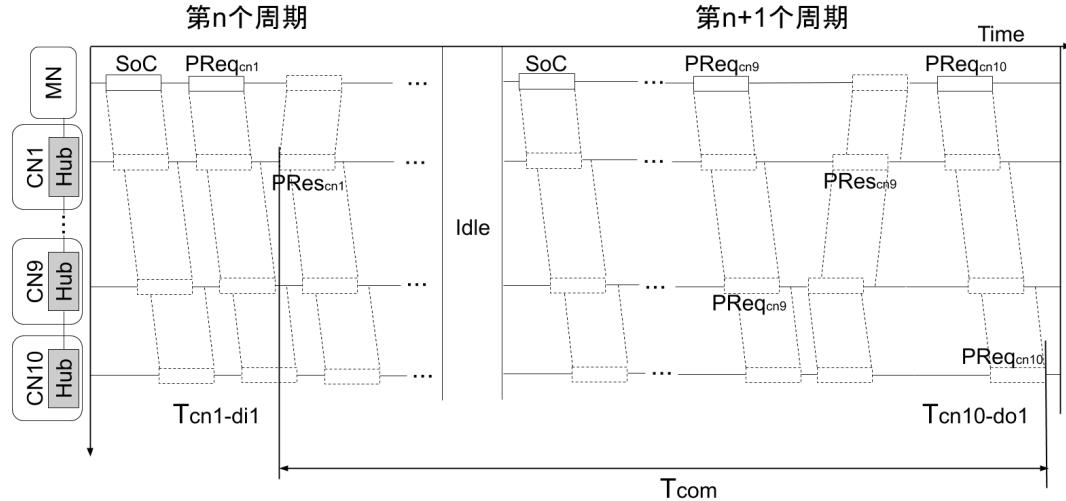


图 3.14 全局响应过程

在全局响应时间的测试中，我们设计了如图 3.14 所示的响应过程。在第 n 个周期，1 号从站的输入信号通过 PRes 数据帧进入 POWERLINK 网络， $T_{cn1-di1}$  为 1 号从站接收到输入信号的时刻，该信号经过主站处理之后，处理结果在第 n+1 个周期通过 PReq 数据帧从 10 号从站输出， $T_{cn10-do1}$  为 10 号从站输出信号的时刻。该响应过程经历两个连续周期， $T_{com}$  是此过程的通信时间。根据 POWERLINK 通信顺序，10 号从站是通信周期中的最后一个参与通信的。因此， $T_{com}$  也是该系统全局响应的最长通信时间。如图 3.11 所示，可以使用 ProfiShark 1G 以太网分析仪捕获数据帧来计算  $T_{com}$ ，大约为  $457\mu s$ 。示波器测得的总响应时间如图 3.15 所示，Ch1（黑线）代表 1 号从站控制器的输入信号；Ch3（红线）代表 10 号从站控制器的输出信号，响应时间约为  $870\mu s$ 。考虑到在本地响应过程光电耦合器引起的延迟时间 ( $T_{coupler}$ ) 约为  $400\mu s$ ，输入信号在“Sync Send Buf”缓存区等待被发送的时间 ( $T_{wait}$ )，输入信号等待进入 POWERLINK 通信的时间 ( $T_{wait}$ ) 约为  $21\mu s$ 。



图 3.15 全局响应时间测试结果

主站的响应时间是指当主站收到来自从站的PRes数据帧到回复PReq数据帧的时间。主站在每个从站通信时隙中都需要响应PReq数据帧，所以主站响应时间直接影响系统的通信周期，进而影响到系统响应时间，而且随着从站数量的增加，影响会加大。主站响应时间与设备类型有关，主站PC方案的主站采用的是基于x86运行Linux的PC。如图3.11所示，相邻PRes数据帧和PReq数据帧之间时间差为主站响应时间( $T_{r-mn}$ )和PReq数据帧的传输时间( $T_{F(PReq)}$ )之和。PReq数据帧长度为64Byte，采用千兆以太网传输，则 $T_{F(PReq)}$ 为 $0.512\mu s$ 。由此计算得到主站PC方案运行Linux PC的主站平均响应时间约为 $16.296\mu s$ 。

### 3. 测试小结

本节对主站PC方案的测试系统进行了包括通信周期、系统响应时间和主站响应时间的性能测试，根据测试结果，总结如下：

- (1) 主站PC方案的测试系统在10节点情况下，可以在 $275\mu s$ 通信周期下运行，系统全局响应时间为 $870\mu s$ 。
- (2) 测试系统的本地响应时间为 $400\mu s$ ，此延时较大，主要是从站输入/输出接口电路中的光耦合器的延时造成的。
- (3) 从站控制器的DI/DO通道数量较少，难满足真实的工程应用需求，可采用数据选择器来扩展DI/DO通道数量。
- (4) 主站是运行Linux系统的PC，较之FPGA，PC的响应时间较慢，响应时间约为 $16.296\mu s$ 。主站的openPOWERLINK程序运行在内核空间下，对以太网网卡类型依赖严重，需要针对不同的网卡类型开发相应的驱动程序，程序可移植性差。主站可采用Zynq控制器来实现，提高反应时间，缩短通信周期。

## 3.2 全站FPGA方案的系统设计与开发

基于主站PC方案的系统测试结果，我们提出了改进版本全站FPGA方案，全站FPGA方案的设计指标如下：

- (1) 从站控制器的输入/输出接口电路采用高速数字隔离器(Digital Isolator)，使本地响应时间 $<10\mu s$ 。
- (2) 从站控制器采用数据选择器将DI通道数量扩展至128路，DO通道数量扩展至16路。
- (3) 主站采用Zynq控制器来实现，在5个节点的系统中，通信周期达到 $50\mu ss$ ，全局响应时间小于 $200\mu s$ 。
- (4) 运行在PC上的IOC应用程序采用UDP Socket的方式与主站通信，UDP协议帧的长度较小，数据传输速度快，IOC应用程序可以尽可能实时监控到POWERLINK网络中各从站状态。

### 3.2.1 系统架构设计

基于全站 FPGA 方案的设计指标，设计了如图 3.16 所示的系统架构，该系统由 1 个主站和 5 个从站组成，网络拓扑为线型拓扑结构，所有站点均按站点号从小到大顺序依次连接，站点之间的通信基于千兆 POWERLINK。主站是基于 Zynq 的控制器，IOC 应用程序运行在 PC 上，IOC 应用程序通过千兆以太网与主站通信。从站是基于 Zynq 的控制器，支持千兆 POWERLINK 通信和集线器 (HUB) 功能。每个从站控制器都有 128 路 DI 和 16 路 DO 通道，用于信号采集和输出。

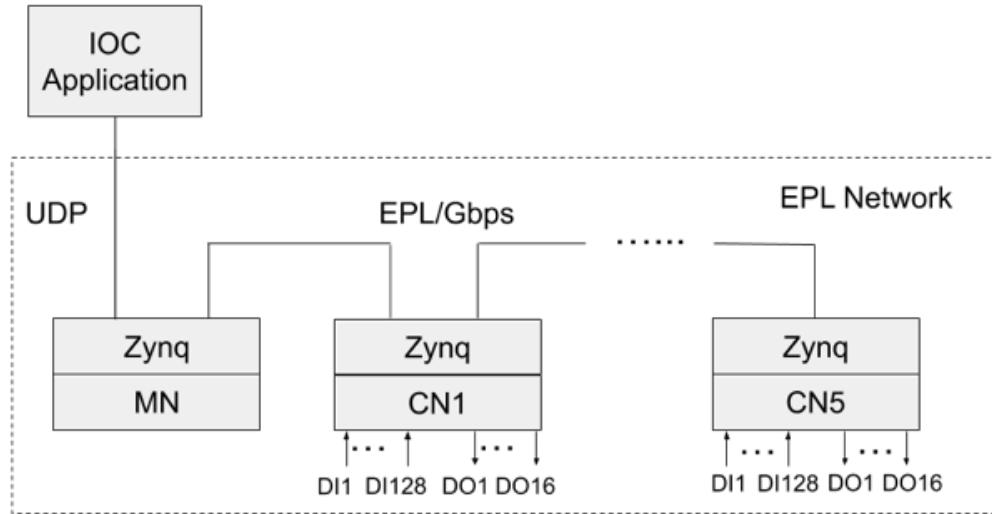


图 3.16 全站 FPGA 方案的系统硬件架构图

### 3.2.2 从站控制器的设计与开发

#### 1. 硬件介绍

全站 FPGA 方案中主站控制器的硬件选型与设计和主站 PC 方案前端控制器基本相同，不同的是主站控制器增加了一个千兆以太网口用于与 IOC 应用程序通信，具体可以参见第 3.1.3 节中硬件介绍部分，在此不再赘述。从站控制器按照设计指标需要提高本地响应速度和扩展 DI 通道数，所以从站控制器采用主板、DI 扩展板和背板相结合的方式来实现，主板和 DI 扩展板为独立板卡，背板为母板，主板和 DI 扩展板均插接与背板上。图 3.17 所示的是方案 2 从站控制器主板实物，主板由核心板和底板组成，硬件设计与主站 PC 方案的从站控制器基本相同。不同的是，全站 FPGA 方案主板增加了背板扩展 DI 接口用于连接背板，主板输出接口电路采用了高速数字隔离器，数字隔离器的型号是 ADuM1400，主板采用 4 选 1 数据选择器来扩展 DI 通道数量，数据选择器型号为 SN54HC153。图 3.18 所示的是 DI 扩展板实物，DI 扩展板输入接口电路同样采用了 ADuM1400

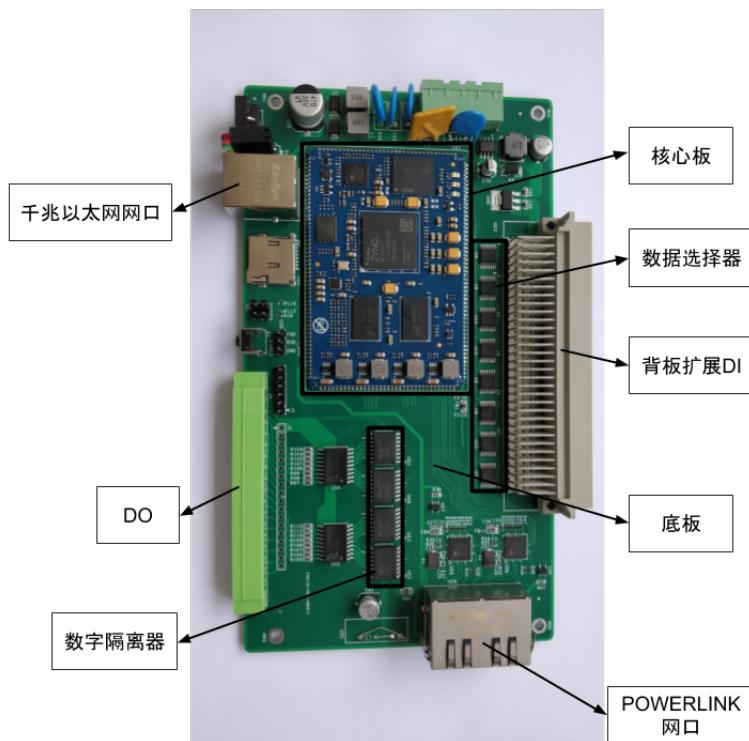


图 3.17 从站控制器电路板照片

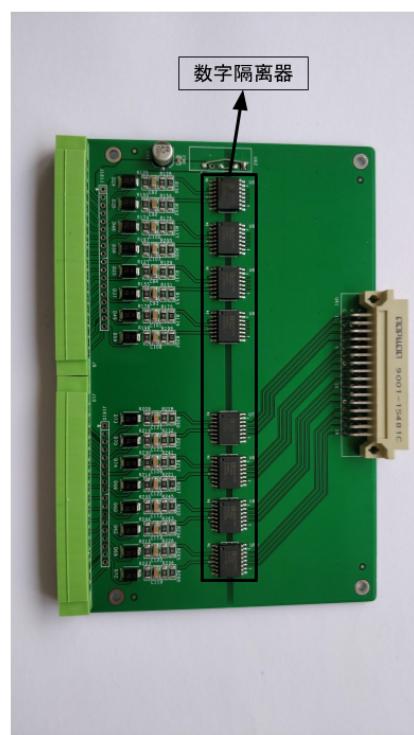


图 3.18 从站控制器 DI 扩展板照片

数字隔离器，4块DI扩展板共提供了128路DI通道。

## 2. 软件设计

从站控制器的软件架构如图3.19，由三部分组成：基于ARM处理器的POWERLINK应用层、基于FPGA的数据链路层和物理层。实现方式与主站PC方案基本相同，具体可以参见第3.1.3节中软件设计部分，在此不再赘述。不同的是，IO模块的数据直接存储在FPGA内部双口RAM中，较之主站PC方案，这种设计减少了读写的延时，使得数据读写速度更快，运行在ARM上的应用层程序通过32位并行总线来读写RAM中的数据。

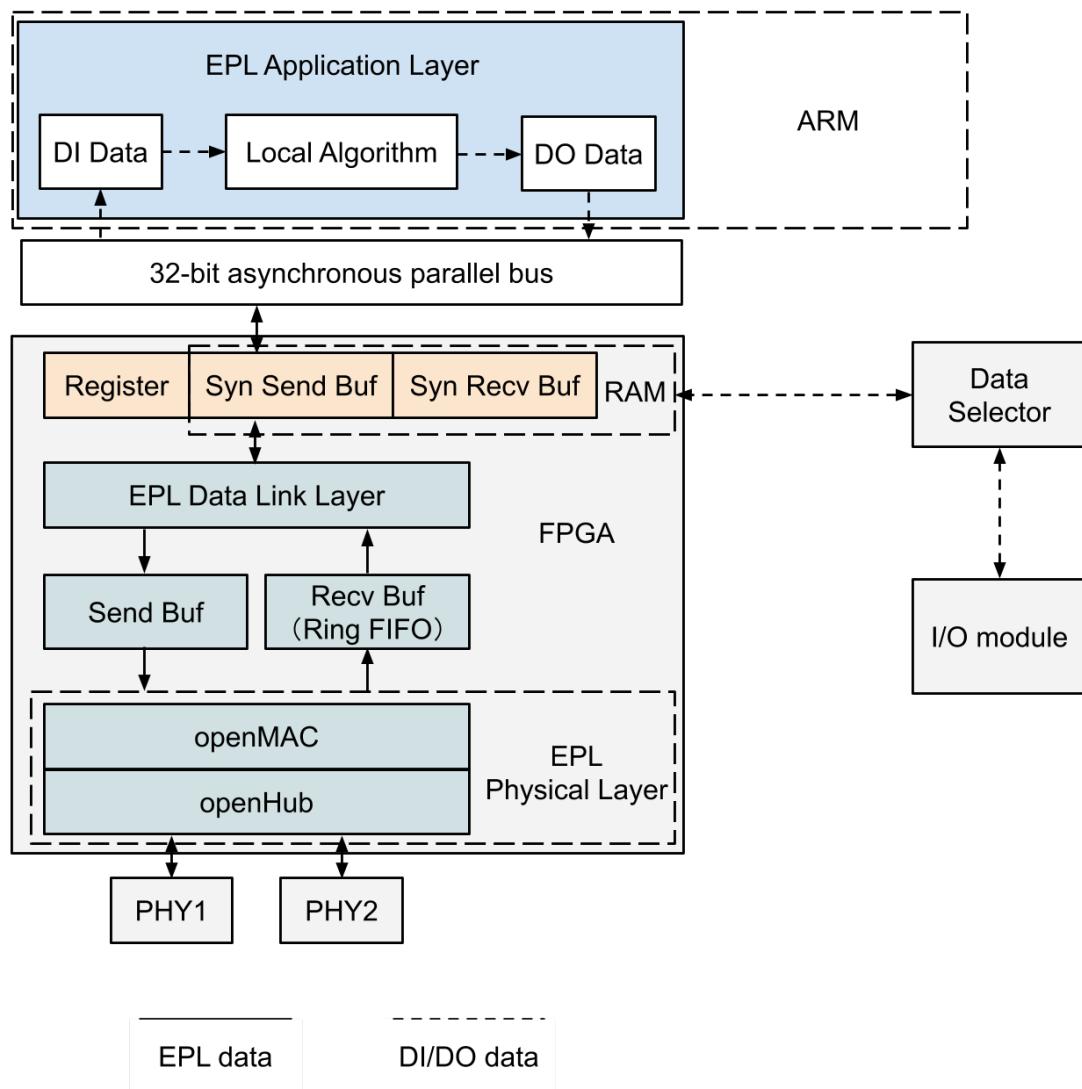


图3.19 基于Zynq的从站控制器软件架构图

主站控制器的软件架构与从站类似，主站应用层保持数据链路层和物理层不变的同时，在应用层添加了UDP客户端通信模块和全局算法模块，如图3.20所示的MN部分。

### 3.2.3 EPICS设备驱动程序的开发

IOC 应用程序运行在 Linux PC 上，主站是基于 Zynq 的控制器，我们采用基于 UDP Socket 的通信方式来实现二者之间的通信。具体实现方式如图 3.20 所示，主站（MN）的“Data Link Layer”收集各从站的 DI 信号，这些信号被发送到“Application Layer”中名为“Input Data”缓存区中。同时作为 UDP 客户端，“Application Layer”会将“Input Data”缓存区中的数据传输到 IOC 应用程序。“Application Layer”中名为“Output Data”的缓存区用于接收来自 IOC 应用程序的控制参数等命令，并向下将控制参数发送到相应的从站。“Application Layer”加入了全局算法功能，主站通过全局算法处理来自各从站控制器的 DI 信号，并通过相应的从站输出相应 DO 信号。

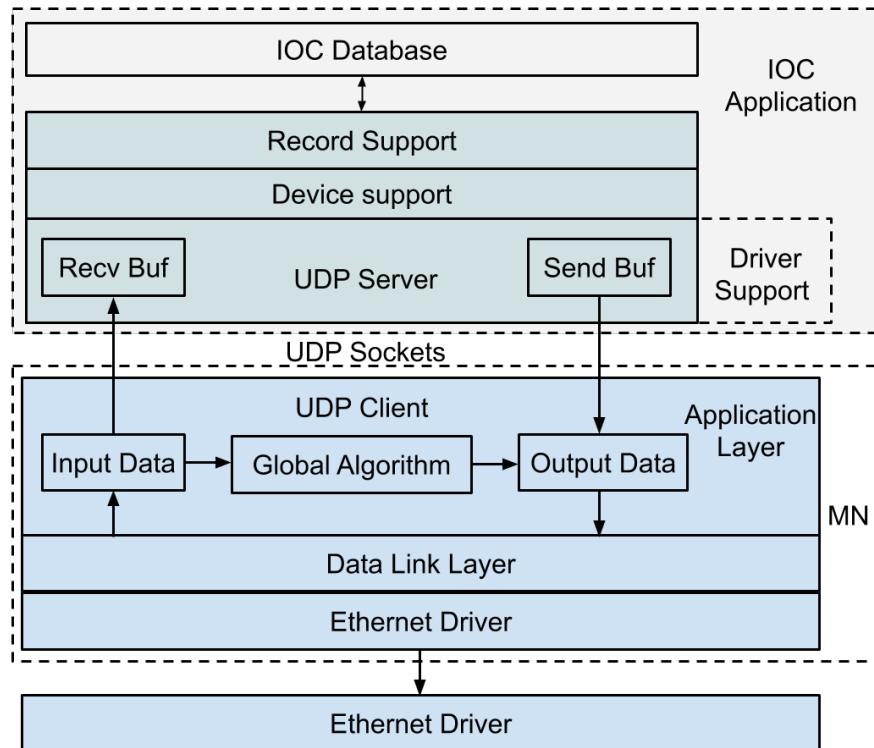


图 3.20 EPICS 设备驱动软件架构图

IOC 应用程序负责监控 POWERLINK 网络中各从站状态，IOC 应用程序的软件结构如图 3.20 所示。作为 UDP 服务端，“Driver support”中有两个缓存区，用于存储与主站交换的数据。在这些缓存区中，“Recv Buf”用于接收来自主站的数据，“Send Buf”用于存储发送至主站的控制参数。“Device Support”的功能是将两个缓存区中的数据转化成 EPICS 中的记录，目前“Device Support”支持标准的 EPICS DI 和 DO 记录。

### 3.2.4 测试系统搭建

基于第3.2.1节设计的全站FPGA方案系统架构，我们搭建了相应的测试系统，图3.21为全站FPGA方案测试系统的照片。照片上可以看到1台PC和6台Zynq控制器，EPICS IOC应用程序运行在PC上，6台FPGA控制器组成了千兆POWERLINK网络，其中1台为主站，5台为从站，主站与PC通过千兆以太网通信，各从站按照节点号从小到大的顺序依次相连成直线拓扑。每个从站都是基于Zynq的控制器，控制器对外提供128个DI和32个DO通道，DI接口如图3.22所示，为漏型输入；DO接口如图3.23所示，为源型输出。

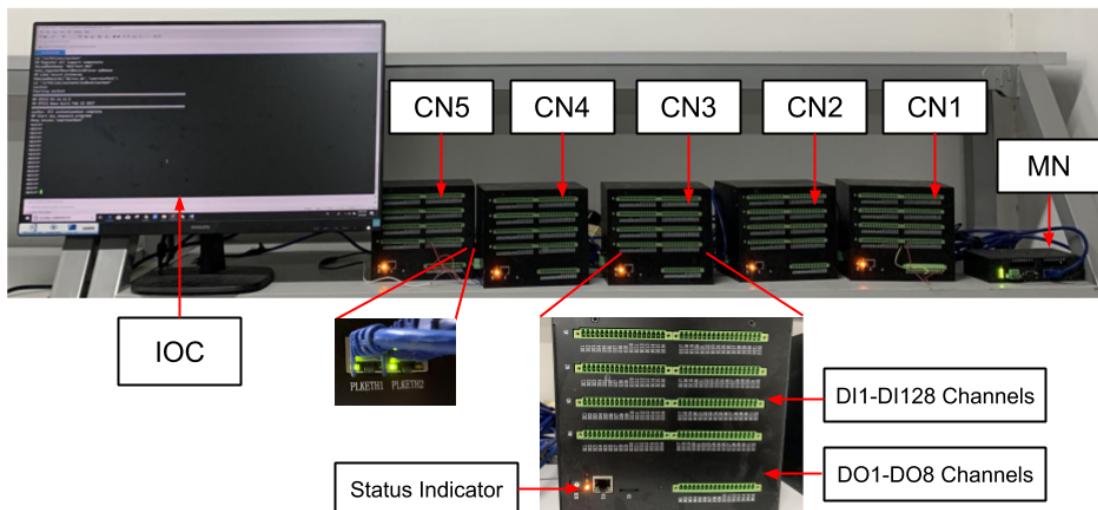


图3.21 全站FPGA方案测试系统照片

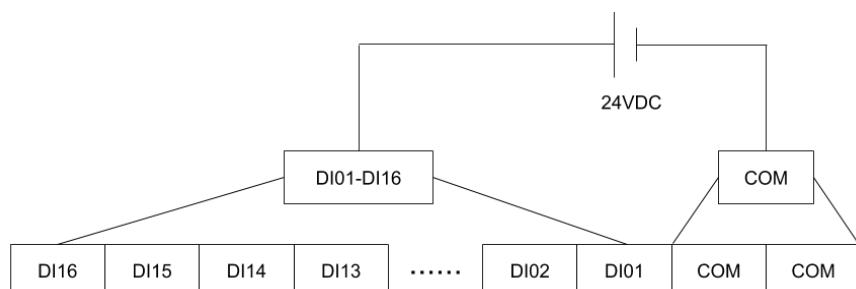


图3.22 从站控制器DI接口

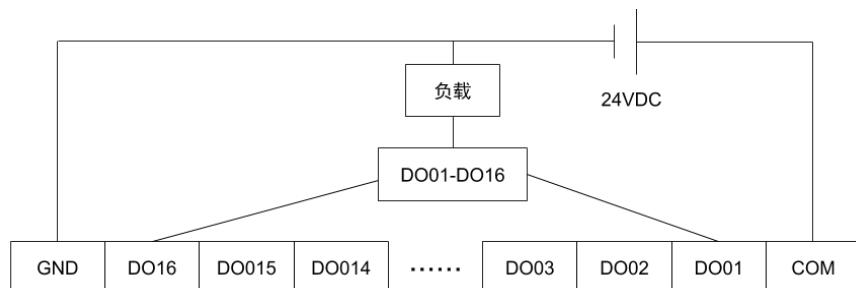


图3.23 从站控制器DO接口

### 3.2.5 系统性能测试与分析

#### 1. 通信周期测试

全站 FPGA 方案系统的通信周期测试方式与主站 PC 方案相同，同样是使用 ProfiShark 1G Ethernet Troubleshooter 以太网诊断工具来测试原型系统在不同通信周期下的运行情况，从而确定系统最短通信周期。

4 0.000022096	CamilleB_56:78:f0	EPLv2_SoC	POWERLINK 64 240->255 SoA		
5 0.000023688	CamilleB_56:78:f0	CamilleB_56:78:01	POWERLINK 64 240-> 1 PReq	→	T(sync)
6 0.000025928	CamilleB_56:78:01	EPLv2_PRes	POWERLINK 64 1->255 PRes		
7 0.000028264	CamilleB_56:78:f0	CamilleB_56:78:01	POWERLINK 64 240-> 2 PReq	→	T(r-mn)+T <sub>F(PReq)</sub>
8 0.000031856	CamilleB_56:78:01	EPLv2_PRes	POWERLINK 64 2->255 PRes		
9 0.000034192	CamilleB_56:78:f0	CamilleB_56:78:01	POWERLINK 64 240-> 3 PReq		
10 0.000039128	CamilleB_56:78:01	EPLv2_PRes	POWERLINK 64 3->255 PRes		
11 0.000041464	CamilleB_56:78:f0	CamilleB_56:78:01	POWERLINK 64 240-> 4 PReq		
12 0.000047736	CamilleB_56:78:01	EPLv2_PRes	POWERLINK 64 4->255 PRes		
13 0.000050072	CamilleB_56:78:f0	CamilleB_56:78:01	POWERLINK 64 240-> 5 PReq		
14 0.000057688	CamilleB_56:78:01	EPLv2_PRes	POWERLINK 64 5->255 PRes		
15 0.000060000	CamilleB_56:78:f0	EPLv2_SoA	POWERLINK 64 240->255 SoA		
16 0.000072920	CamilleB_56:78:f0	EPLv2_SoC	POWERLINK 64 240->255 SoC	→	T(idle)

图 3.24 测试系统 POWERLINK 数据帧抓取结果

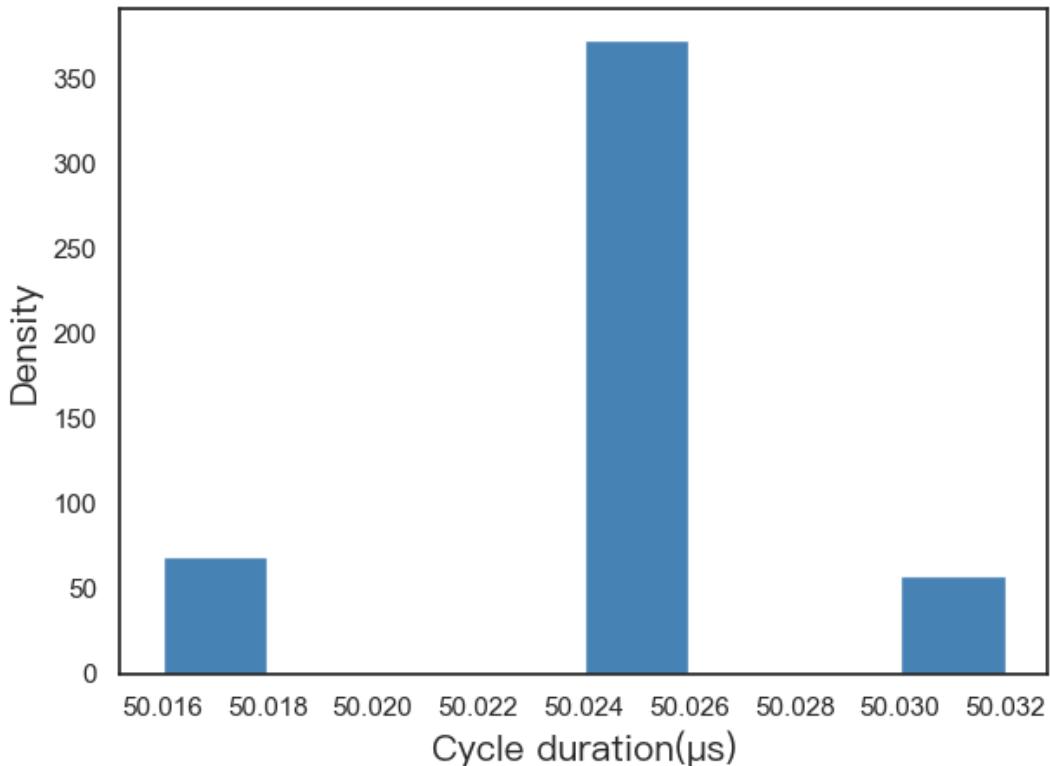


图 3.25 测试系统 POWERLINK 通信周期概率密度分布

全站 FPGA 方案的测试系统通信周期最快可配置为  $50\mu s$ ， $50\mu s$  通信周期下的通信过程如图 3.24 所示。我们抓取了超过 20k 个连续的 POWERLINK 数据帧，并将通信周期的概率密度分布绘制在图 3.25 中。该图显示大多数情况下的通信周期为  $50.024\mu s$ ，少数情况为  $50.016\mu s$  和  $50.032\mu s$ ，此结果是由基于 Zynq 的控

制器和ProfiShark 1G以太网分析仪的125Mhz时钟频率引起的。通信周期周期的最大偏移仅为 $32\mu s$ ，抖动非常小。

## 2. 响应时间测试

我们同样对全站FPGA方案的测试系统进行了本地响应时间测试、全局响应时间测试和主站响应时间测试，测试结果和分析如下：

我们首先对从站控制器进行本地响应时间的测量，本地响应的过程如下：

1. 系统从站的GPIO读取输入信号，并将输入值以二进制形式存储在FPGA的双口RAM中；
2. ARM通过32bit并行总线读取并处理RAM中的输入数据；
3. 处理结果通过32bit并行总线写入相应的RAM中并通过GPIO输出信号。

我们首先选择测试系统中的5号从站控制器进行了本地响应时间测量，同样以输入和输出信号的上升沿时间差来表示本地响应时间。测试结果如图3.26所示，Ch2（青色线）代表5号从站控制器的输入信号；Ch3（红色线）代表5号从站控制器的输出信号，响应时间为 $5\mu s$ 。这个时间延迟包括从站控制器输入/输出接口电路中的数字隔离器的延时和ARM读取处理数据的时间。由于ARM的时钟频率为125MHz，ARM读取处理数据的时间仅在 $\mu s$ 量级，所以本地响应时间主要是由从站控制器的数字隔离器造成的，较之主站PC方案中的光耦合器，型号为ADuM1400的数字隔离器大大提高了信号传输速率。

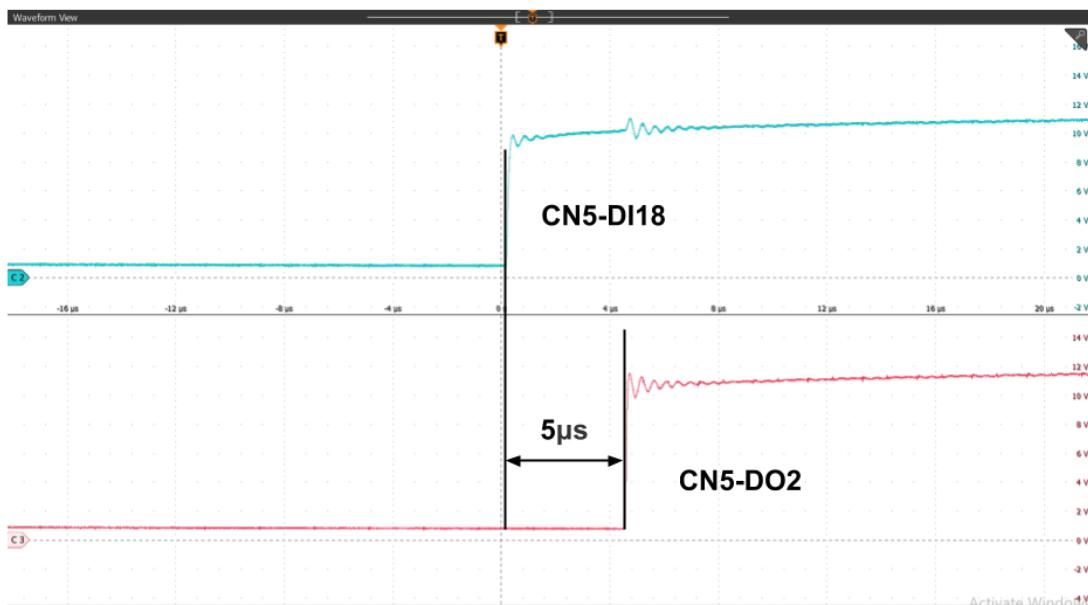


图3.26 本地响应时间测试结果

全站FPGA方案的测试系统的全局响应时间( $T_{global-response}$ )由POWERLINK网络通信时间( $T_{powerlink}$ )和从站处理输入信号( $T_{CN-process}$ )的时间组成，见公式3.3。

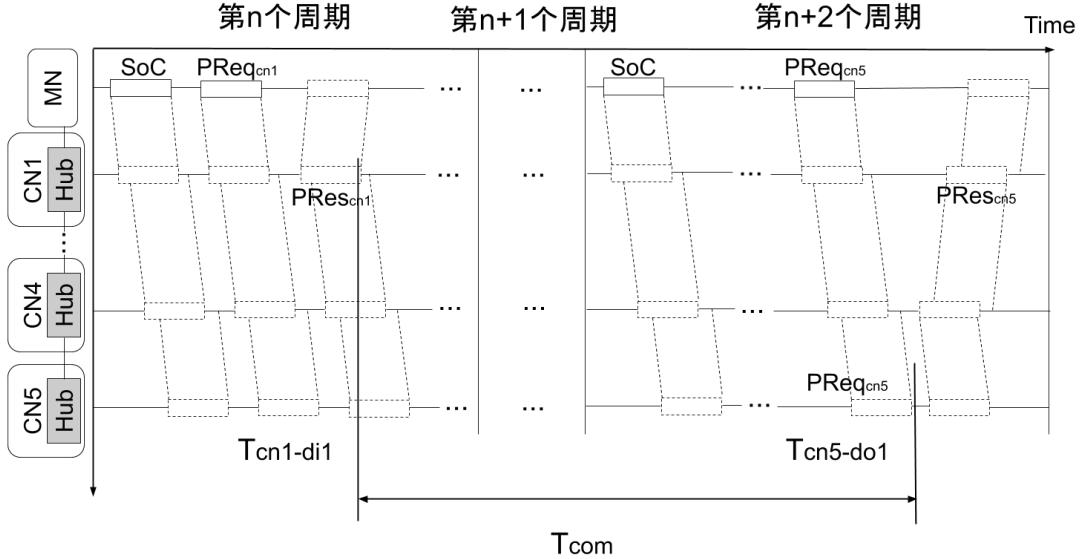


图3.27 系统全局响应过程

$$T_{global-response} = T_{powerlink} + T_{CN-process} \quad (3.3)$$

在全局响应时间的测试中，我们设计了如图3.27所示的响应过程。在第n个周期，1号从站的输入信号通过PRes数据帧进入POWERLINK网络， $T_{cn1-di1}$ 为1号从站接收到输入信号的时刻，在第n+1个周期主站对该输入信号进行处理，处理结果在第n+2个周期通过PReq数据帧从5号从站输出， $T_{cn5-do1}$ 为5号从站输出信号的时刻。与主站PC方案通信过程不同的是，该响应过程需要经历三个连续周期，原因是主站对输入信号的处理时间为数 $\mu s$ ，无法在本周期将处理结果传输出去。具体解释如下：在第n+2的周期开始的时候，主站发送SoC数据帧后产生同步中断信号，中断信号会触发应用层的同步回调函数“synCb ()”对“Sync Recv Buf”缓存区中的输入数据进行处理，这一过程的耗时为数 $\mu s$ ，包括响应中断信号的时间、32bit并行总线读写的时间和数据处理的时间，所以处理结果只能在缓存区中等待下一个周期被传输出去。如图3.27所示， $T_{com}$ 是此过程的POWERLINK通信时间。根据POWERLINK通信顺序，5号从站是通信周期中的最后一个参与通信的，因此 $T_{com}$ 也是该系统全局响应的最长通信时间。我们可以根据Wireshark抓包结果来计算第三个周期5号从站的PReq数据帧与第一个周期1号从站的PRes数据帧之间的时间差，从而得到 $T_{com}$ ，大约为125 $\mu s$ 。

从站处理输入信号的时间( $T_{CN-process}$ )包括磁耦合器的延时( $T_{coupler}$ )和输入信号在“Sync Send Buf”缓存区等待被发送的时间( $T_{wait}$ )，见公式3.4，其中磁耦合器的延时约为5 $\mu s$ 。输入信号在进入从站控制器后并不会立刻进入POWERLINK网络，输入信号值需要在“Sync Send Buf”缓存区中等待回调函数“synCb ()”读取后才可以传输到POWERLINK网络中，而“synCb ()”需要SoC

帧的触发才可以执行，所以输入信号的等待时间为信号进入系统到从站接收到SoC数据帧的时间差，最长为一个通信周期 $50\mu s$ 。由公式3.3和3.4得，全局响应时间最长为 $180\mu s$ 。

示波器测得的总响应时间如图3.28所示，Ch2（青色线）代表1号从站控制器的输入信号；Ch1（黑色线）代表5号从站控制器的输出信号，响应时间为 $160\mu s$ 。

$$T_{CN-process} = T_{coupler} + T_{wait} \quad (3.4)$$

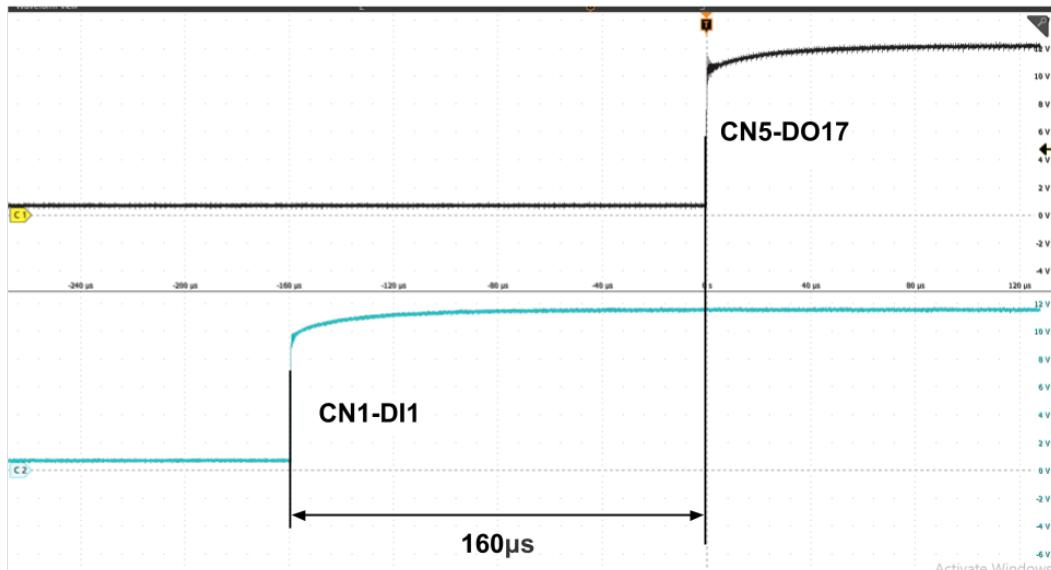


图3.28 全局响应时间测试结果

全站FPGA方案的测试系统采用的基于Zynq的主站控制器。如图3.24所示，相邻PRes数据帧和PReq数据帧之间时间差为主站响应时间( $T_{r-mn}$ )和PReq数据帧的传输时间( $T_{F(PReq)}$ )之和。PReq数据帧长度为64Byte，采用千兆以太网传输，则 $T_{F(PReq)}$ 为 $0.512\mu s$ 。可得 $T_{r-mn}$ 的平均值为 $1.865\mu s$ ，基于Zynq控制器的响应时间平均值为 $1.865\mu s$ 小于基于x86的pc $16.296\mu s$ 的响应时间，主站采用Zynq控制器来实现可以显著提升系统实时性能。

### 3. 测试小结

本节对全站FPGA方案的测试系统进行了包括通信周期、系统响应时间和主站响应时间的性能测试，根据测试结果，总结如下：

- (1) 全站FPGA方案的测试系统在5个节点情况下，可以在 $50\mu s$ 通信周期下运行，系统全局响应时间为 $160\mu s$ ，满足小于 $200\mu s$ 的设计指标。
- (2) 全站FPGA方案的测试系统主站采用Zynq控制器来实现，将主站响应时间缩短至 $1.865\mu s$ ，较之主站PC方案中的pc $16.296\mu s$ 的响应时间，缩短了约 $14.4\mu s$ 的响应时间。

- (3) 全站 FPGA 方案的测试系统从站控制器的输入输出电路使用高速数字隔离器后，将本地响应时间缩短至  $5\mu s$ ，较之主站 PC 方案的光耦合器，显著提高了信号传输的速率。
- (4) 全站 FPGA 方案的测试系统从站控制器使用 4 选 1 数据选择器，将每个从站控制器的 DI 通道扩展至 128 路。DO 通道数扩展至 16 路。
- (5) 全站 FPGA 方案的测试系统在通信周期和主站响应时间等性能参数上优于主站 PC 方案，所以我们选择对基于全站 FPGA 方案的多节点 POWERLINK 系统进行研究。我们将采用第 2.4 节和第 2.5 节提出的理论计算和仿真模拟两种方法，结合全站 FPGA 方案测试系统的实测参数，对多节点 POWERLINK 系统的通信周期进行评估。

### 3.3 全站 FPGA 方案通信周期的理论计算

本节根据第 2.4 节中提出的 POWERLINK 通信周期的理论计算方法并结合全站 FPGA 方案测试系统的实测数据，推导出全站 FPGA 方案的多节点 POWERLINK 系统的通信周期计算公式。

The diagram illustrates the communication cycle  $T_{poll}$  for the full FPGA scheme. It is divided into two main components:  $T_{cn-slot}^2$  and  $T_{rtd4} + T_{F(PRes)}$ . The  $T_{cn-slot}^2$  component is shown as a bracketed pair of slots. The  $T_{rtd4} + T_{F(PRes)}$  component is also bracketed and positioned below the first slot of the  $T_{cn-slot}^2$  pair.

0.000072920	CamilleB_56:78:f0	EPLv2_SoC	POWERLINK	64 240->255 SoC
0.000073712	CamilleB_56:78:f0	CamilleB_56:78:01	POWERLINK	64 240-> 1 PReq
0.000075952	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 1->255 PRes
0.000078296	CamilleB_56:78:f0	CamilleB_56:78:01	POWERLINK	64 240-> 2 PReq
0.000081880	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 2->255 PRes
0.000084224	CamilleB_56:78:f0	CamilleB_56:78:01	POWERLINK	64 240-> 3 PReq
0.000089152	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 3->255 PRes
0.000091496	CamilleB_56:78:f0	CamilleB_56:78:01	POWERLINK	64 240-> 4 PReq
0.000097768	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 4->255 PRes
0.000100112	CamilleB_56:78:f0	CamilleB_56:78:01	POWERLINK	64 240-> 5 PReq
0.000107736	CamilleB_56:78:01	EPLv2_PRes	POWERLINK	64 5->255 PRes
0.000118056	CamilleB_56:78:f0	EPLv2_SoA	POWERLINK	64 240->255 SoA
0.000122944	CamilleB_56:78:f0	EPLv2_SoC	POWERLINK	64 240->255 SoC

图 3.29 全站 FPGA 方案测试系统抓包结果

通过第 2 节中对全站 FPGA 方案测试系统实测所得的数据，结合第 2.4 节中对等时同步阶段的理论分析，计算出基于 Zynq 控制器的  $T_{hub}$ 、 $T_{r-cn}$  等参数，并推导出轮询阶段时长的计算公式。

对于全站 FPGA 方案测试系统，主站和每个从站的用户数据均为 2Byte 时，PReq 和 PRes 数据帧的大小均为 64Byte， $T_{F(PReq)}$  等于  $T_{F(PRes)}$ ，综合第 2.4 节中的从站通信间隙的相关公式 3.5 和 3.6，我们可以得到公式 3.7，即相邻从站时隙的差值为  $2T_c + 2T_{hub}$ ，是个定值，表示从站时间间隙随着节点号的变大呈线性增长。

$$T_{rtd}^i = 2iT_c + (2i - 1)T_{hub} + T_{r-cn} \quad (3.5)$$

$$T_{cn-slot}^i = T_{F(PReq)}^i + T_{F(PRes)}^i + T_{rtd}^i + T_{r-mn} \quad (3.6)$$

$$\begin{aligned} T_{cn-slot}^{i+1} - T_{cn-slot}^i &= T_{rtd}^{i+1} - T_{rtd}^i \\ &= 2T_c + 2T_{hub} \end{aligned} \quad (3.7)$$

如图3.29所示，相邻两个PReq数据帧之间的时间差等于该从站时间间隙( $T_{cn-slot}$ )，我们可以计算出每个从站的时间间隙，结果列于表3.1。

表3.1 从站时间间隙统计表：

	Min	Max	Mean	Std.Dev
$T_{cn-slot}^1$	4.56	4.584	4.572	0.0072
$T_{cn-slot}^2$	5.90	5.94	5.91	0.0091
$T_{cn-slot}^3$	7.25	7.3	7.26	0.0069
$T_{cn-slot}^4$	8.66	8.59	8.607	0.0099
$T_{cn-slot}^5$	9.91	9.98	9.929	0.0102

所有数值均以 $\mu s$ 表示。

图3.30为从站平均时间间隙散点图，我们添加了线性拟合趋势线，R平方值为1，表示拟合程度高，趋势线的可靠性高。拟合直线的函数斜率为1.3411等于 $2T_c + 2T_{hub}$ ， $T_c$ 为网线的延时，网线长度为2m，延时为(5ns/m)，则 $T_{hub}$ 为0.66 $\mu s$ 。如图3.29所示，相邻的PRes数据帧和PReq数据帧之间的时间差为 $T_{rtd}^i$ 和 $T_{F(PRes)}$ 之和，结合公式3.5，可以计算得到 $T_{r-cn}$ 的平均值为1.048 $\mu s$ 。我们将一些重要通信参数的计算结果汇总列入表3.2中。

表3.2 通信参数统计

主站响应时间 ( $T_{r-mn}$ )	从站响应时间 ( $T_{r-cn}$ )	HUB 延时 ( $T_{hub}$ )
1.865 $\mu s$	1.048 $\mu s$	0.66 $\mu s$

根据第2.4节中的公式3.8、公式3.9和公式3.10，并代入 $T_{hub}=0.66\mu s$ ， $T_{r-cn}=1.048\mu s$ ， $T_{r-mn}=1.865\mu s$ ，我们可以推导出全站FPGA方案的等时同步阶段时长公式为3.11。

$$T_{poll} = T_{frames} + T_{net} \quad (3.8)$$

$$T_{frames} = \sum_{i=1}^n (T_{F(PReq)}^i + T_{F(PRes)}^i) \quad (3.9)$$

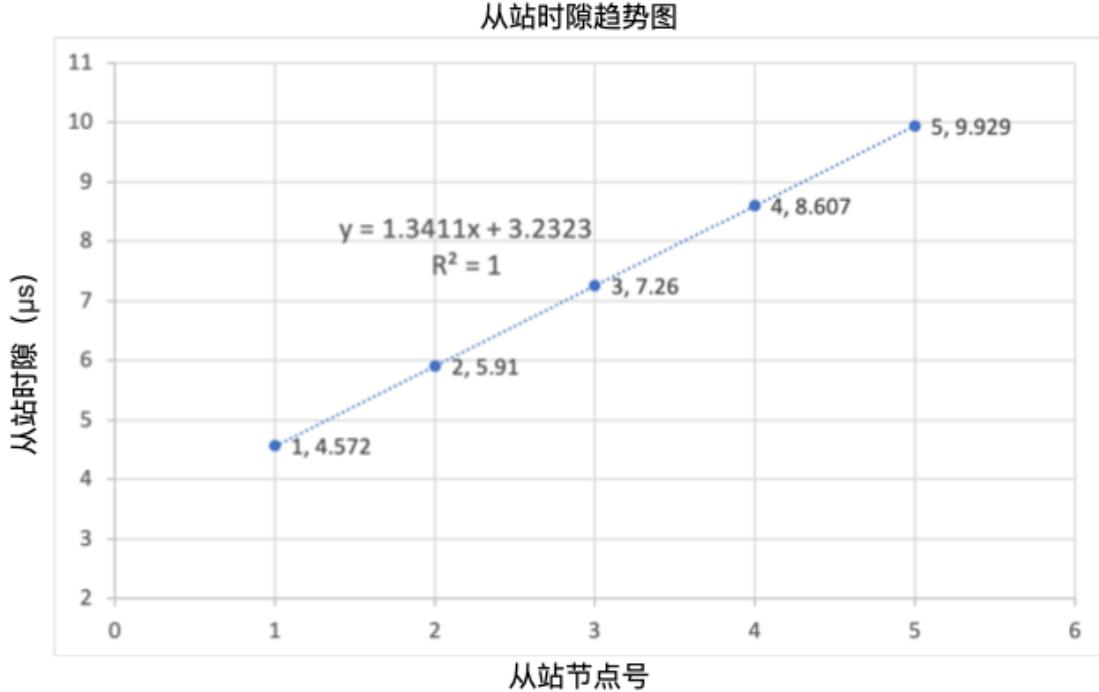


图 3.30 从站时隙趋势图

$$T_{net} = \sum_{i=1}^n T_{rtd}^i + nT_{r-mn} \quad (3.10)$$

$$\begin{aligned} T_{poll} &= \sum_{i=1}^n (T_{F(PReq)}^i + T_{F(PRes)}^i) + T_{net}^{line} \\ &= \sum_{i=1}^n (T_{F(PReq)}^i + T_{F(PRes)}^i) + 0.67n^2 + 2.923n \end{aligned} \quad (3.11)$$

对于全站 FPGA 方案系统，SoC 帧的大小为 64Byte，采用千兆以太网传输， $T_{F(SoC)}$  仅为  $0.512\mu s$ ， $T_W$  为等待从站接收并处理 SoC 帧的时间，从站是基于 Zynq 的控制器，处理速度极快。由图 3.24 所示，同步过程 ( $T_{sync}$ ) 在  $1\mu s$  以内可以完成。考虑到空闲时间 ( $T_{idle}$ ) 为传输最大异步数据预留的时间，根据图 3.24 所示实测结果  $T_{idle}$  约为  $12.9\mu s$ ，可得通信周期的公式 3.12，单位为  $\mu s$ 。

$$\begin{aligned} T_{cycle\_time} &= T_{F(SoC)} + T_W + \sum_{i=1}^n (T_{F(PReq)}^i + T_{F(PRes)}^i) + n(n+1)T_c + n^2T_{hub} + n(T_{r-cn} + T_{r-mn}) + T_{idle} \\ &= \sum_{i=1}^n (T_{F(PReq)}^i + T_{F(PRes)}^i) + 0.67n^2 + 2.923n + 13.9 \end{aligned} \quad (3.12)$$

### 3.4 全站FPGA方案的仿真模拟

全站FPGA方案的仿真模拟基于OMNet++网络模拟器展开，我们可以直接使用第2.5建立了支持千兆POWERLINK通信的节点模型，然后结合实测参数对全站FPGA方案测试系统进行建模，将仿真结果和实测结果进行对比，来验证仿真建模方法的可行性，从而可将仿真建模方法应用于多节点POWERLINK系统通信周期的评估。

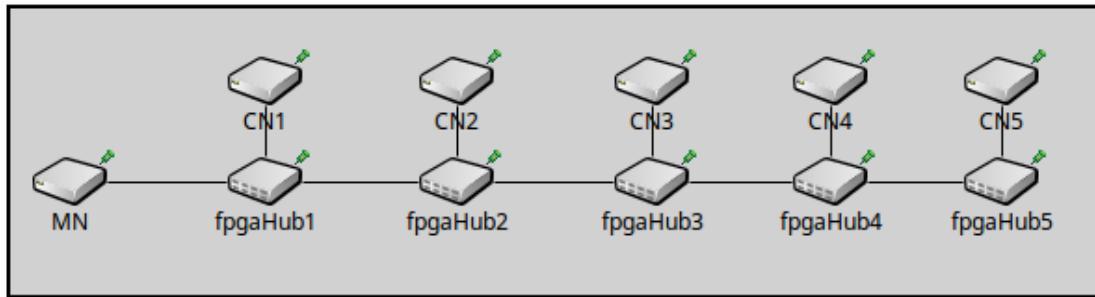


图3.31 测试系统模型

我们首先对全站FPGA方案测试系统进行了建立了如图3.31所示的模型，节点数和拓扑结构均与原型系统保持一致，相邻节点之间通过2m长的网线相连，节点之间的通信基于千兆POWERLINK。对于原型系统中从站控制器的模拟，我们采用从站通信节点模块加上FPGA HUB模块来实现，两模块间传输延时设置为 $0\mu s$ 。除此之外，为了准确模拟原型系统，我们采用原型系统中的实测参数对系统模型进行了配置，配置文件包括`xdc_cn_std.xml`和`omnetpp.ini`。`xdc_cn_std.xml`文件中配置了主站对各从站发送数据的大小(`PReqActPayload`)和各从站被分配的通信时隙(`PResTimeout`)，其中各从站的通信时隙通过公式3.5和公式3.6得到， $T_{r-mn}$ 、 $T_{r-cn}$ 、 $T_{hub}$ 参数采用表3.2中的实测数据，具体配置如下：

```

1 <Nodes>
2   <Node NodeId="1" PResTimeout="4617ns" PReqActPayload="2byte" />
3   <Node NodeId="2" PResTimeout="5958ns" PReqActPayload="2byte" />
4   <Node NodeId="3" PResTimeout="7299ns" PReqActPayload="2byte" />
5   <Node NodeId="4" PResTimeout="8640ns" PReqActPayload="2byte" />
6   <Node NodeId="5" PResTimeout="9981ns" PReqActPayload="2byte" />
7 </Nodes>

```

`omnetpp.ini`仿真配置文件包括了对系统通信周期、各从站节点号和响应数据大小、HUB延时的设定，具体内容如下：

```

1 [Config EPLBase]
2 # 通信周期设置
3 **.MN.mnDll.cycleLenMean = 50us
4
5 # 各从站节点号和pres数据大小设定

```

```

6 **.CN1.cnDll.nodeId = 1
7 **.CN1.cnDll.presActPayload = 2
8 **.CN2.cnDll.nodeId = 2
9 **.CN2.cnDll.presActPayload = 2
10 **.CN3.cnDll.nodeId = 3
11 **.CN3.cnDll.presActPayload = 2
12 **.CN4.cnDll.nodeId = 4
13 **.CN4.cnDll.presActPayload = 2
14 **.CN5.cnDll.nodeId = 5
15 **.CN5.cnDll.presActPayload = 2
16
17 # FPGA HUB 延时设定
18 **.hubDelayMean = 0.66us

```

模拟过程持续了大约 3000 个 POWERLINK 通信周期。因为同步阶段和同步阶段时长为定值，所以我们统计了等时同步阶段的时长，并将等时同步阶段概率密度分布绘制在图 3.32 中，该图表明大多数情况下的等时同步阶段时长为  $37.2\mu s$ ，最大抖动不超过 30ns。实测的等时同步阶段时长平均值为  $36.312\mu s$ ，较之实测值，仿真模拟方法对轮询阶段时长的计算误差在  $1\mu s$  以内， $1\mu s$  误差对于  $50\mu s$  的通信周期来说影响非常小，所以仿真模拟方法可以准确的估算轮询阶段的时长。考虑到同步过程 ( $T_{sync}$ ) 在  $1\mu s$  以内可以完成以及  $12.9\mu s$  的空闲时间，通过仿真模拟得到的通信周期为  $51.1\mu s$ 。

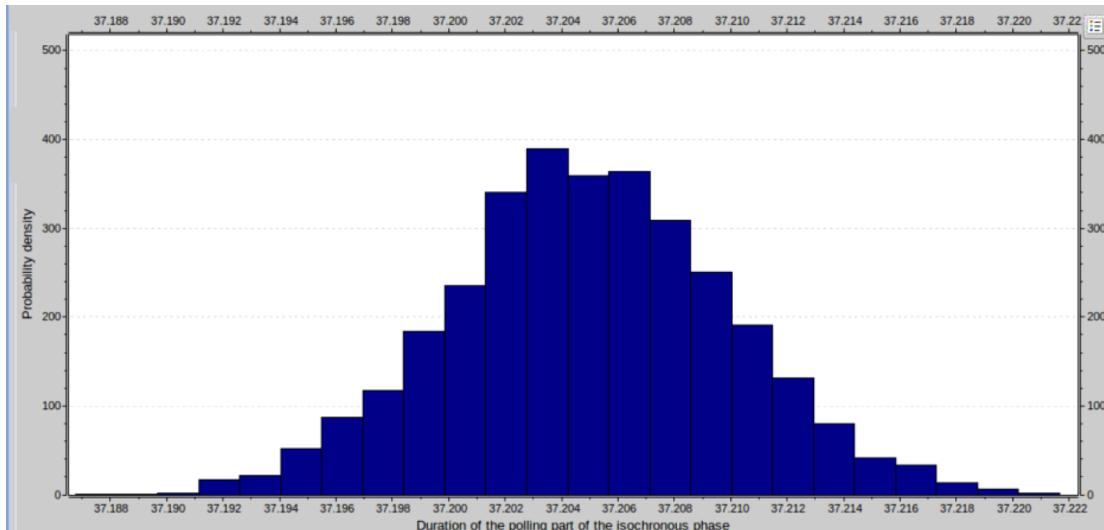


图 3.32 原型系统模型轮询阶段的概率密度分布

## 第4章 HALF设备保护系统的设计

### 4.1 HALF预研工程

合肥先进光源（Hefei Advanced Light Facility，HALF）是由国家同步辐射实验室提出的第四代基于衍射极限储存环（Diffraction Limited Storage Ring，DLSR）的同步辐射光源，HALF是一个位于中低能区，以真空紫外和软X射线为主的，具有高亮度、低发射度和唯一全辐射谱段空间相干性最先进的衍射极限光源。HALF将集成与相干性密切关联的系列先进测量技术，具有超快时间分辨、超高空间分辨和超高能量分辨的能力，为量子信息与量子材料、能源与环境、物质科学、生命科学等前沿研究领域提供强有力的研究平台。

目前国家同步辐射实验室在中国科学院和地方政府的共同支持下开展了HALF预研工程，预研工程将对加速器、光束线站的核心关键技术进行攻关及样机研制，为未来HALF的建设打好技术基础。HALF预研工程计划于2020年底完成，目前最新设计的合肥先进光源的结构如图4.1所示，HALF加速器总体由注入器和存储环组成。

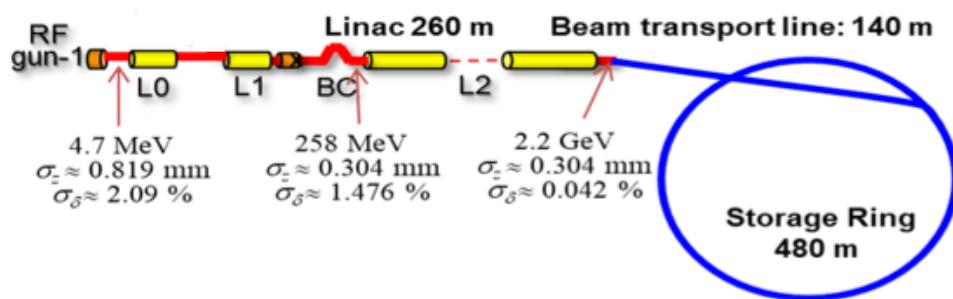


图4.1 HALF总体结构图

注入器由满能量直线加速器（Linac）和束流输运线（Beam transport line）组成，其中直线加速器的长度为260米，输运线长度为140米。直线加速器用于产生电子束，将其能量加速到2.2GeV。输运线负责将直线加速器的电子束传输到储存环，沿储存环切线方向产生的同步辐射光通过光束线传输到实验站进行各种科学实验。

HALF储存环的光束发射度非常小，因此光束线可以将一定波长范围内的光束聚焦到其衍射极限，这种类型的储存环称为衍射极限储存环，HALF衍射极限储存环的最新设计参数如表4.1所示。

表4.1 HALF储存环主要参数值

参数	数值	单位
能量	2.2	GeV
周长	480	m
周期数	20	-
自然发射度	85.1	pm · rad
工作点 (H/V)	48.175/17.175	-
自然色品	-75, -79	-
动量紧缩因子	$6.3 \times 10^{-5}$	-
阻尼分配数 (H/V/L)	1.475/1.0/1.525	-
自然阻尼时间 (H/V/L)	22/32.4/21.2	ms
电子单圈弯铁辐射能量损失	217.5	keV
自然能散	$0.66 \times 10^{-3}$	-
直线节数目	40个(20个长直线节+20个中直线节)	-
长直线节长度	5.5	m
长直线节中点处的 $\beta_x/\beta_y$ /色散函数	5.445/2.533/0.0	m
中直线节长度	2.2	m
中直线节中点处的 $\beta_x/\beta_y$ /色散函数	2.842/1.954/0.028	m

## 4.2 加速器中的设备保护系统

### 4.2.1 设备保护系统的任务

机器保护系统 (Machine Protection System, MPS) 是加速器控制系统的重要组成部分，负责保护加速器机器组件的安全。MPS 在加速器各分总体、系统之间建立联锁保护逻辑，当设备出现故障或者束流丢失时，MPS 能够快速地对加速器重要设备实施保护，并向中央控制系统报告设备故障信息，对故障数据进行存档和分析。根据响应时间的不同，MPS 可分为慢联锁保护系统 (Slow Protection System, SPS) 和快联锁保护系统 (Fast Protection System, FPS)，SPS 和 FPS 在不同加速器装置中的名称有所不同，例如 ALBA 光源将 SPS 称为设备保护系统 (Equipment Protection System, EPS)，将 FPS 称为 FIS (Fast Interlock System)<sup>[35]</sup>。本文将 HALF MPS 中的慢联锁保护系统称为 EPS，快联锁保护系统称为 FPS。从系统响应时间的角度来看，EPS 的响应时间一般在 10ms 量级，而 FPS 的响应时间一般要求在 10μs 量级。EPS 和 FPS 的具体应用场合如下：

- (1) EPS 监测的设备状态信号主要是真空度、温度、冷却水流量等与设备运行安全相关的信号，经过 EPS 联锁逻辑判断后，迅速实施相应的设备保护措施。
- (2) FPS 的应用场合是在加速器运行过程中，如果发生了束流偏离轨道达到一定阈值或者某些设备出现故障，导致电子束流可能会损毁机器，此时必须采取保护

措施及时切断束流。

### 4.2.2 国内外加速器的机器保护系统调研

本节对国外加速器装置机器保护系统的结构和硬件组成进行了调研，具体情况如下：

#### 1. LCLS 的机器保护系统

直线加速器相干光源 (Linac Coherent Light Source, LCLS) 是世界上第一个硬 X 射线自由电子激光装置，其永磁波荡器长度为 130 米，位于美国斯坦福直线加速器中心。LCLS 机器保护系统的任务是避免波荡器等重要设备受到束流照射而损坏，要求 MPS 在一个束流脉冲周期 8.33ms 内切断束流。MPS 监测的信号包括束流损失信号、束流位置信号、束流强度和插入件状态。

LCLS 机器保护系统的系统架构如图 4.2 所示，机器保护系统由 32 个从节点 (Link Node) 和一个主节点 (Link Processor) 组成，主节点通过核心交换机与从节点构成星形拓扑结构，节点间通过专用的千兆以太网通信，通信协议是 LCLS 自主设计的实时协议。从节点分布于从激光注入器到 X 射线实验线站的整个装置，负责接收束流故障信号。主节点按照联锁保护逻辑处理来自从节点的束流故障信号，通过相应的控制设备 (Mitigation device) 切断束流。主节点和从节点同时接入到 EPICS 控制网络，用于中控系统监控机器保护系统的状态<sup>[58]</sup>。

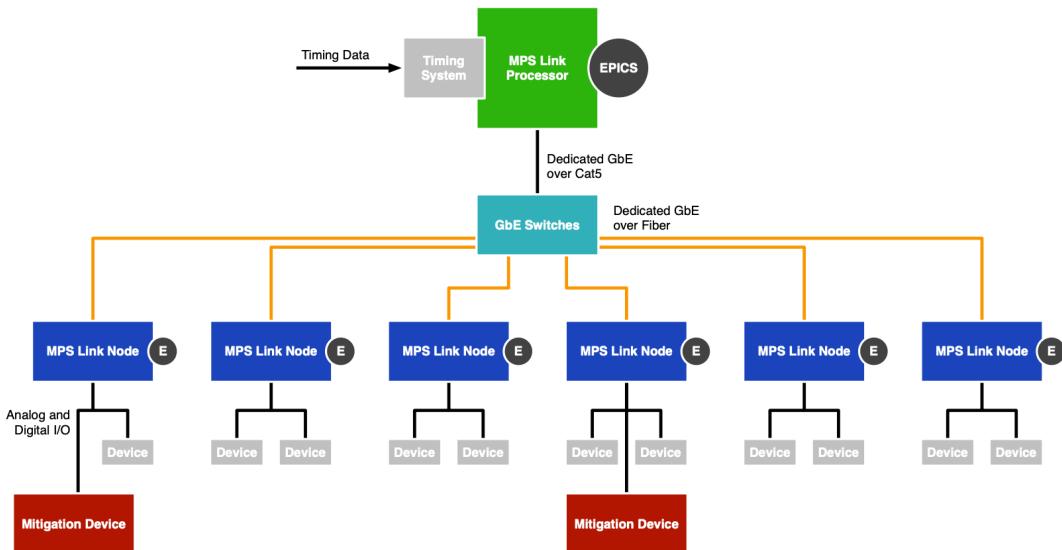


图 4.2 LCLS 机器保护系统结构图

机器保护系统的从节点基于型号为 Xilinx Vertex4 的 FPGA 芯片实现，每个连接节点可支持多达 96 路数字输入，8 路固态继电器输出，4 路 TTL 兼容逻辑电平触发输入和 4 路触发输出，同时配备了单独的串口与 EPICS IOC 通信。

## 2. RAON 的机器联锁保护系统

Rare isotope Accelerator complex for ON-line experiment (RAON) 是一台正在建设中的重离子加速器，为在线同位素分离实验和飞行时间型碎片分离实验提供平台。RAON 由韩国基础科学研究所 (IBS) 负责，CERN, Fermilab, TRIUMF 等实验室也共同参与到 RAON 项目建设中。

RAON 机器保护系统的结构如图 4.3 所示，由 Slow Interlock System (SIS)、Fast Protection System (FPS)、Beam Permit System (BPS) 和 Post-mortem System (PMS) 组成<sup>[59]</sup>，

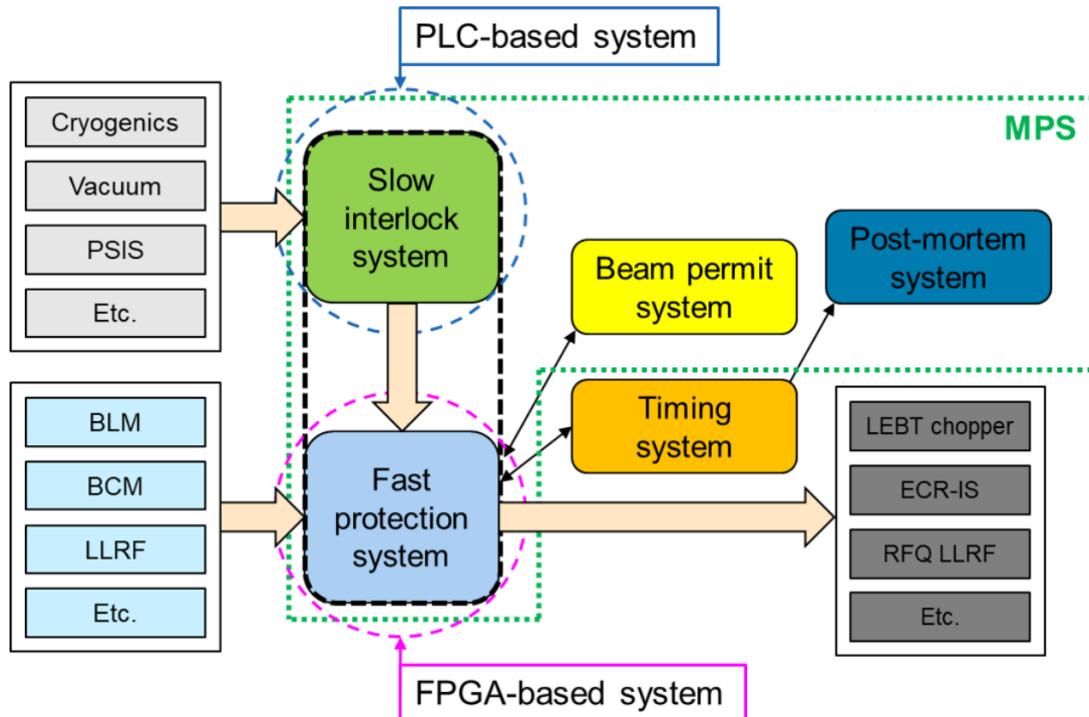


图 4.3 RAON 机器保护系统结构图

各系统具体设计如下：

- (1) SIS 基于 PLC 实现，监测低温，真空等系统中的设备信号，一旦设备出现故障，SIS 要在几十 ms 内通过 FPS 实施切断束流的设备保护措施。
- (2) FPS 各节点基于 FPGA 实现，FPS 节点监测的信号来自束流丢失监测器 (BLM)、束流流强监控器 (BCM)、低电平 (LLRF) 系统设备等，FPS 需要在几十  $\mu$ s 内通过禁止高频功率输出等措施切断束流。
- (4) BPS 负责根据加速器不同的运行模式来切换 MPS 运行状态。
- (5) PMS 是用来对设备故障信号进行存档和分析的系统。

## 3. 上海光源储存环的机器保护系统

文献<sup>[60]</sup> 中所描述的机器保护系统是指上海光源的设备保护系统，基于 PLC 实现，其总体结构如图 4.4 所示，具体设计如下：原有的上海光源储存环机器保

护系统由1个储存环分总体机器保护系统控制器和20个机器保护系统单元控制器组成，控制器采用日本Yokogawa FA-M3系列PLC，机器保护系统单元负责监控本单元的真空相关信号和冷却水状态信号，机器保护系统单元之间通过电缆相连。20个机器保护系统单元将各单元的联锁信号传输给储存环分总体级联锁控制器，经过储存环分总体联锁控制器逻辑判断后，给出禁止高频功率输出、禁止储存环注入等保护信号。

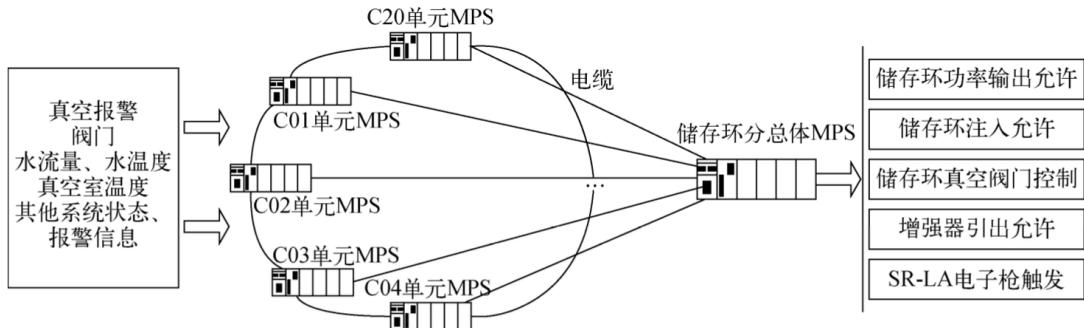


图4.4 原有的的储存环机器保护系统结构图

上海光源线站工程（二期）为了满足超高磁场二极铁、超导扭摆器、低温波荡器等新增设备的联锁需求，在保证机器保护系统原联锁功能运行正常的情况下，对机器保护系统进行了升级。升级后的机器保护系统架构如图4.5所示，各机器保护系统单元增加了FL-net专用网络通信模块，各单元通过FL-net专用网络连接成菊花链结构，MPS通过FL-net通信解决了联锁系统跨单元通信的问题，满足了低温波荡器真空联锁系统获取相邻单元间阀门信号的联锁需求。经过在线测试，基于FL-net的MPS响应时间小于21ms，满足上海光源加速器运行的需求。

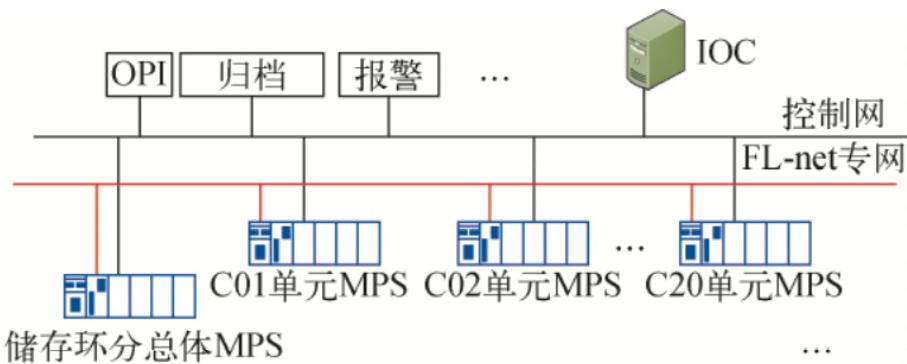


图4.5 改造后的储存环机器联锁保护系统结构图

#### 4. CSNS 漂移管直线加速器水冷联锁系统

中国散裂中子源(China Spallation Neutron Source, CSNS)是我国在2018年建成的全世界第四台脉冲型散裂中子源，包括一台80MeV负氢漂移管直线加速

器 (Drift Tube Linac, DTL)、一台 1.6GeV 快循环质子同步加速器、两条束流输运线、一个靶站以及一期 3 台中子散射谱仪, CSNS 的质子束功率达到 10 kW, 脉冲重复频率达到 25Hz。

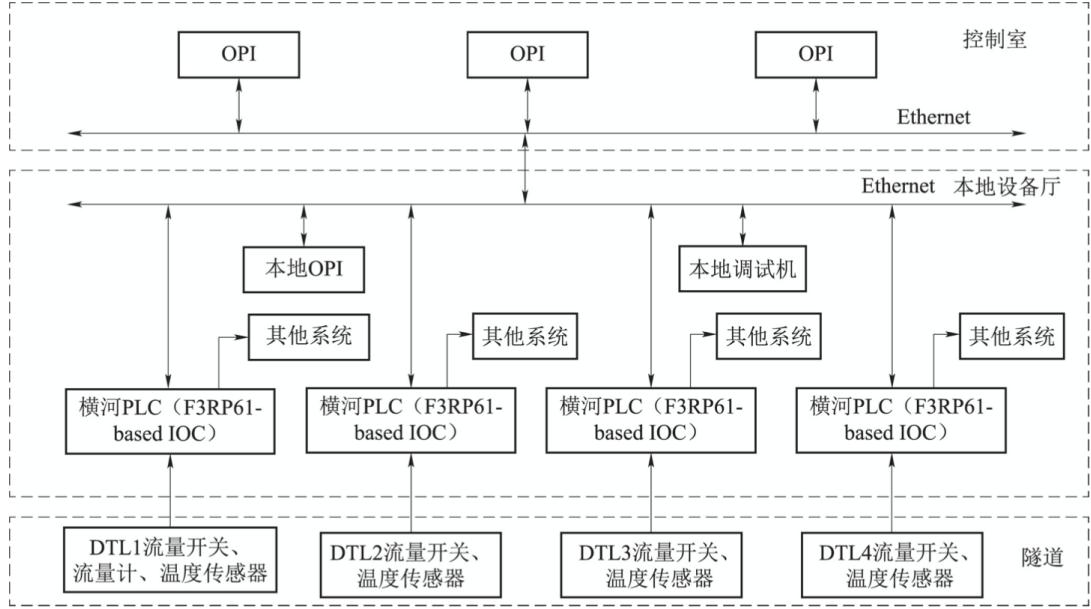


图 4.6 CSNS 漂移管直线加速器水冷联锁系统结构图

DTL 水冷联锁系统负责监测 DTL 的冷却水流量和温度信号, 通过控制器逻辑处理后, 为磁铁电源和射频系统提供联锁保护信号, 系统架构如图 4.6 所示。DTL 水冷联锁系统可分为 4 套子系统, 子系统控制器为横河 FA-M3 PLC, 负责监测冷却水状态信号, 经过 PLC 中 CPU 模块的联锁处理后, 为磁铁电源和射频系统提供保护。EPICS IOC 运行在 PLC 中的嵌入式 CPU 模块中, 负责监控水冷联锁系统的状态<sup>[61]</sup>。

## 5. 调研总结

一般来说, 设备保护系统和快保护系统是相对独立的两个系统。其中, 大部分加速器装置选择采用 PLC 作为设备保护系统的联锁控制器, 例如上述中的 RAON 重离子加速器和上海光源, 除此之外还有西班牙的 ALBA 光源等。PLC 之间有的采用硬件联锁信号相连, 例如改造前的上海光源储存环机器保护系统, 上海软 X 射线自由电子激光装置联锁保护系统<sup>[62]</sup>; 有的采用以太网相连, 例如 ALBA 光源 PLC 之间的通信是基于 POWERLINK 实现<sup>[35]</sup>, 升级后的上海光源储存环机器保护系统控制器也增加了 FL-net 专用网络通信模块<sup>[60]</sup>。部分加速器装置选择采用 FPGA 控制器作为设备保护系统的联锁控制器, 例如 LCLS 的联锁控制器, 控制器之间采用专用的千兆以太网通信, 采用自主设计的通信协议<sup>[58]</sup>。对于 FPS 的快联锁保护场合, 一般采用 FPGA 控制器来实现, 快联锁信号直接连接到联锁控制器中进行处理, 例如 RAON 和 J-Parc 的快保护系统都采用 FPGA

控制器来实现快联锁响应<sup>[63]</sup>。

对于 HALF 设备保护系统，我们计划采用前面研究过的 Zynq 控制器作为联锁控制器，控制器之间的通信通过千兆 POWERLINK 实现。同时 Zynq 控制器也可以作为快保护系统的联锁控制器，只是 Zynq 控制器之间只能采用联锁信号直接相连的方式。这样，设备保护系统和快保护系统的联锁控制器可以采用同样类型的硬件，为以后的系统运行和维护带来便利。由于本论文主要是研究 POWERLINK 在加速器装置中的应用研究，所以下文将描述基于 POWERLINK 的 HALF 设备保护系统设计。

## 4.3 HALF 设备保护系统设计

### 4.3.1 HALF 设备保护系统任务

作为 HALF 控制系统的重要组成部分，HALF EPS 的任务是在 HALF 不同的运行模式下保证机器设备的安全。HALF EPS 在 HALF 各分总体、系统间建立联锁保护逻辑，监测的设备状态主要包括真空度、温度、冷却水流量等，在经过系统联锁逻辑判断后，给出禁止电子枪触发、禁止 RF 功率输出允许、禁止注入等联锁保护信号。HALF EPS 遍布整个装置，联锁输入信号分布广泛，数量众多，包括各设备状态信号和来自其他系统的联锁信号，具体可分为以下 11 类信号：

- (1) 电子枪状态信号；
- (2) 真空相关信号；
- (3) 冷却水状态信号；
- (4) 磁铁电源状态信号；
- (5) 微波设备状态信号；
- (6) 调制器状态信号；
- (7) 注入系统状态信号；
- (8) 储存环真空部件温度信号；
- (9) 插入件状态信号；
- (10) 来自光束线站的联锁信号；
- (11) 来自人身安全系统的联锁信号。

EPS 通过给出某些设备的禁止运行信号从而达到保护重要设备的目的，保护措施具体如下：

- (1) 关闭电子枪驱动激光的光闸；
- (2) 禁止电子枪触发；
- (3) 关闭真空阀门；
- (4) 禁止固态放大器触发；

- (5) 禁止调制器触发；
- (6) 切断高频功率输出；
- (7) 禁止储存环注入。

### 4.3.2 HALF设备保护系统设计原则

HALF设备保护系统应遵循以下设计原则：

- (1) 独立性：设备保护系统应具有独立性，体现在采用专用控制器，控制器之间通过专用网络进行通信；
- (2) 失效安全性：当控制器出现故障或信号失联时，仍可以最大程度上保障 HALF 装置的安全；
- (3) 可靠性：系统应该具有高可靠性，可以采取硬件冗余设计、自检功能、Watchdog、Heartbeat 等保证系统的可靠性；
- (4) 实时性：对于设备故障信号，设备保护系统要在确定的时间内实施保护动作。
- (5) 灵活性：设备保护系统应采用模块化设计并且预留出足够数量的 IO 点，以方便扩展。

### 4.3.3 HALF设备保护系统运行模式

EPS 需要接受来自 HALF 运行管理系统的命令，切换自身的运行模式以满足 HALF 的运行管理要求。HALF EPS 设计了 5 种运行模式，分别如下：

- (1) 停机模式：此模式下 HALF 处于停机状态，EPS 需要确保所有真空阀门均处于关闭状态，并停止所有系统设备的定时触发信号。
- (2) 直线加速器调试模式：此模式下仅进行注入器分总体中直线加速器的束流调试，直线加速器的 EPS 正常开启，旁路掉输运线、储存环 EPS 的联锁输入信号和来自光束线站的联锁信号。
- (3) 注入器调试模式：此模式下仅进行注入器分总体的束流调试，注入器 EPS 正常开启，旁路掉储存环 EPS 的联锁输入信号和来自光束线站的联锁信号。
- (4) HALF 加速器总体调试模式：此模式下进行注入器和储存环的束流调试，注入器 EPS 和储存环 EPS 正常开启，旁路来自光束线站的联锁信号。
- (5) 运行模式：HALF 处于正常的运行状态，此时 EPS 开启全部联锁保护功能。

### 4.3.4 HALF设备保护系统总体结构

根据 HALF 的体系架构，我们可将 HALF EPS 由上至下分为总体级联锁、分总体级联锁、系统级联锁三个联锁层次，其中，下层系统为上层联锁提供信号，并接收来自上层的联锁保护命令。三个层次的联锁功能具体如下：

- (1) 总体级联锁：根据 HALF 的不同运行模式切换 EPS 的运行模式；处理 HALF

各分总体之间的联锁关系，以及接收来自人身安全系统等外部系统的联锁信号，确保整个 HALF 的运行安全。

(2) 分总体级联锁：负责分总体内各个系统之间设备的联锁保护，保证分总体的安全运行，同时向总体级联锁提供联锁输入信号。

(3) 系统级联锁：负责各系统内部设备的联锁保护，保证分系统运行正常，同时向分总体级联锁系统提供联锁输入信号。

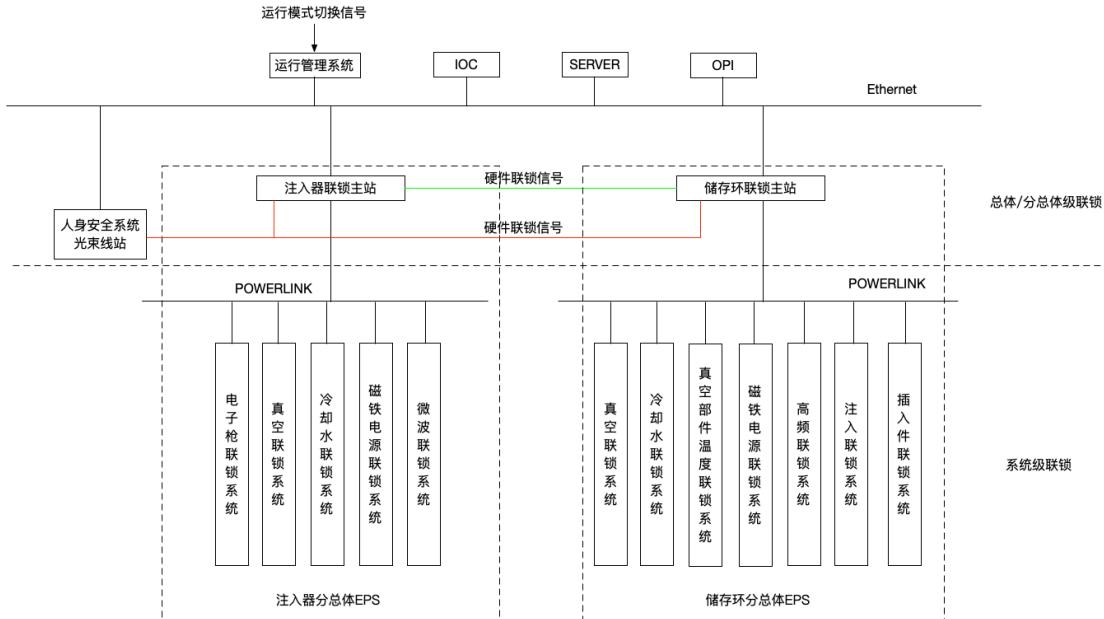


图 4.7 HALF EPS 总体结构图

根据 HALF 设备保护系统层次化的结构特点，我们设计了如图 4.7 所示的 HALF EPS 总体结构，可以概括如下：

(1) HALF EPS 分为注入器分总体 EPS 和储存环分总体 EPS 两部分，分总体 EPS 负责本分总体范围内的设备安全。按照不同的联锁保护逻辑，我们将注入器分总体 EPS 分为电子枪联锁系统、真空联锁系统、磁铁电源联锁系统、微波联锁系统、冷却水联锁系统，储存环分总体 EPS 可分为真空联锁系统、冷却水联锁系统、真空部件温度联锁系统、磁铁电源联锁系统、高频联锁系统、注入联锁系统、插入件联锁系统。

(2) 各分总体 EPS 基于独立的千兆 POWERLINK 网络设计，由一个主站控制器和多个从站控制器组成，控制器均采用 Zynq 控制器。主站控制器负责处理本分总体内各个系统之间的联锁保护关系，从站作为分总体内各系统的 EPS 控制器，负责收集并处理系统内的设备联锁信号。从站将设备联锁信号通过 POWERLINK 上传至主站控制器，经过主站控制器的联锁逻辑处理之后，主站通过 POWERLINK 网络将保护命令传输至相应的从站实施保护。

(3) 注入器 EPS 主站和储存环 EPS 主站之间通过硬件信号相连，处理分总体之间

的联锁保护关系。

- (4) 来自人身安全系统和光束线站的联锁信号采用硬件信号的形式接入分总体 EPS 主站，通过分总体内相应从站实施保护动作。
- (5) EPS 主站通过千兆以太网接入 HALF 控制系统网络中，将设备故障信息和联锁保护信息通过 IOC 上传至 HALF 控制系统。同时 EPS 主站通过控制系统网络接受来自运行管理系统的模式切换命令，切换自身的运行模式以满足 HALF 的运行管理要求。

#### 4.3.5 联锁输入信号的预处理

HALF 设备保护系统运行在不同模式下时，需要对联锁输入信号进行不同的预处理，具体包括旁路、锁存和复位三种处理<sup>[64]</sup>：

- (1) 旁路：设备保护系统处于调试模式时，需要临时解除对部分联锁输入信号的联锁响应，通过将输入信号设置成旁路状态来实现这种需求。旁路的原理：输入信号被设置成旁路状态后，输入信号的实时状态仍可以正常显示，但当输入信号处于报警状态时不会引起系统的联锁保护响应。
- (2) 锁存：设备保护系统处于调试模式或者运行模式时，当联锁输入信号处于报警状态时，需要将输入信号设置成锁存状态，保持输入信号的报警状态不变。当输入信号从报警状态恢复到正常状态后，被锁存的输入信号将继续处于故障状态，输入信号引起的联锁输出信号也保持为保护状态，当操作人员解除该输入信号锁存时，才能取消信号的故障状态，从而解除相应的联锁保护。
- (3) 复位：当输入信号从故障状态恢复到正常状态时，需要通过复位操作解除信号的锁存状态，复位可分为手动复位和自动复位两种方式。手动复位是操作人员对故障信息进行判断后，通过手动复位操作撤消故障信号的锁存状态；对于部分联锁输入信号，可能需要操作员进行频繁的复位操作，我们在采用自动复位的方式解除联锁信号的锁存状态。

联锁输入信号的预处理流程如图 4.8 所示，具体流程如下：

1. 联锁输入信号首先经过模式旁路和调试旁路的判断，模式旁路指在停机或调试模式下无需对任何联锁输入信号进行处理；调试旁路是指在加速器调试过程中，操作人员对部分输入信号进行的旁路设置，以暂时去掉对这些联锁输入信号的保护响应；
2. 对联锁输入信号进行锁存处理，对于未被锁存并处于故障状态下的联锁输入信号进行锁存处理；
3. 对于恢复正常状态的联锁输入信号，通过自动复位或操作人员手动复位的处理对该信号锁存状态进行解除。
4. 预处理之后的联锁输入信号，再进行联锁逻辑的处理。

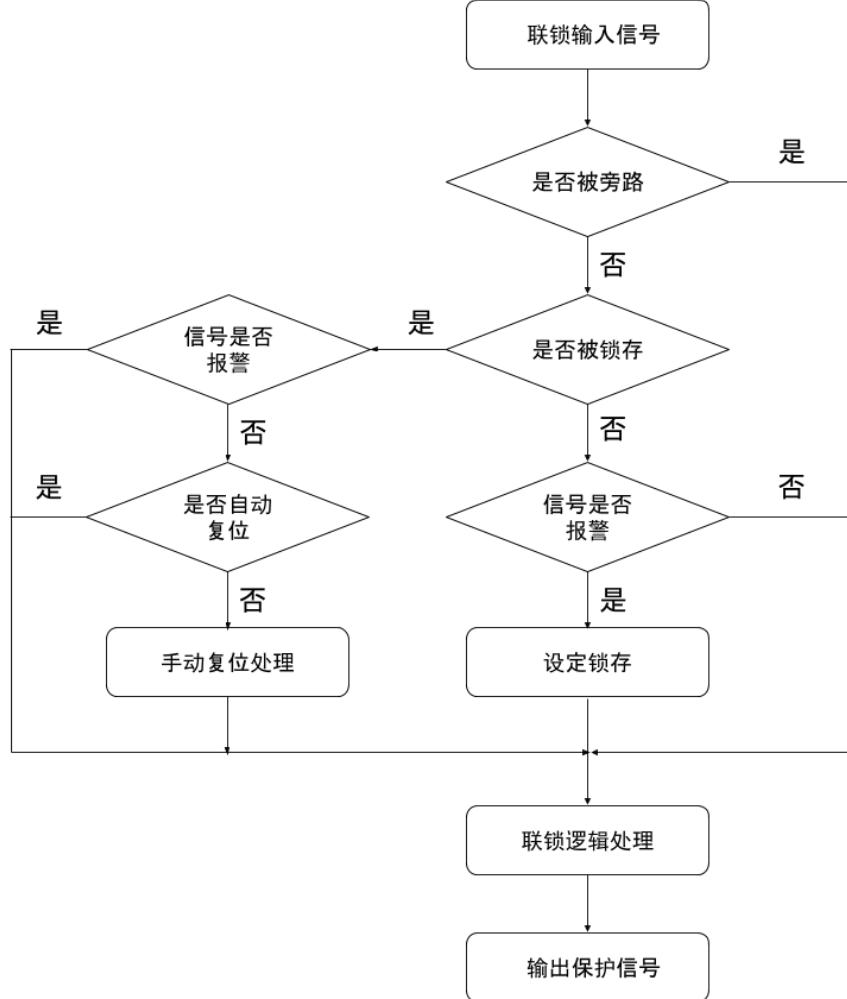


图 4.8 输入信号预处理流程图

下面分别对 HALF 注入器和储存环分总体 EPS 的设计进行阐述。

#### 4.4 注入器 EPS 设计

注入器由直线加速器和输运线组成，直线加速器包括 1 台光阴极微波电子枪和主加速器，其中光阴极微波电子枪由 1 台 80MW 速调管通过波导功分器提供微波功率，主加速器中一共有 20 个 S 波段 6m 等梯度加速段（每个 6m 等梯度加速段均由 2 根 3m 等梯度加速管组合而成），使用 20 套 80MW 速调管为主加速器提供微波功率。

注入器分总体 EPS 的任务是当注入器分总体内各系统设备出现故障或者接收到来自注入器分总体外部的联锁故障信号时，经过注入器联锁主站的联锁逻辑判断后，通过联锁从站实施相应的保护动作。图 4.9 描述了注入器 EPS 的联锁保护功能。

注入器 EPS 的输入联锁信号较多，包括运行管理系统的联锁模式切换信号；

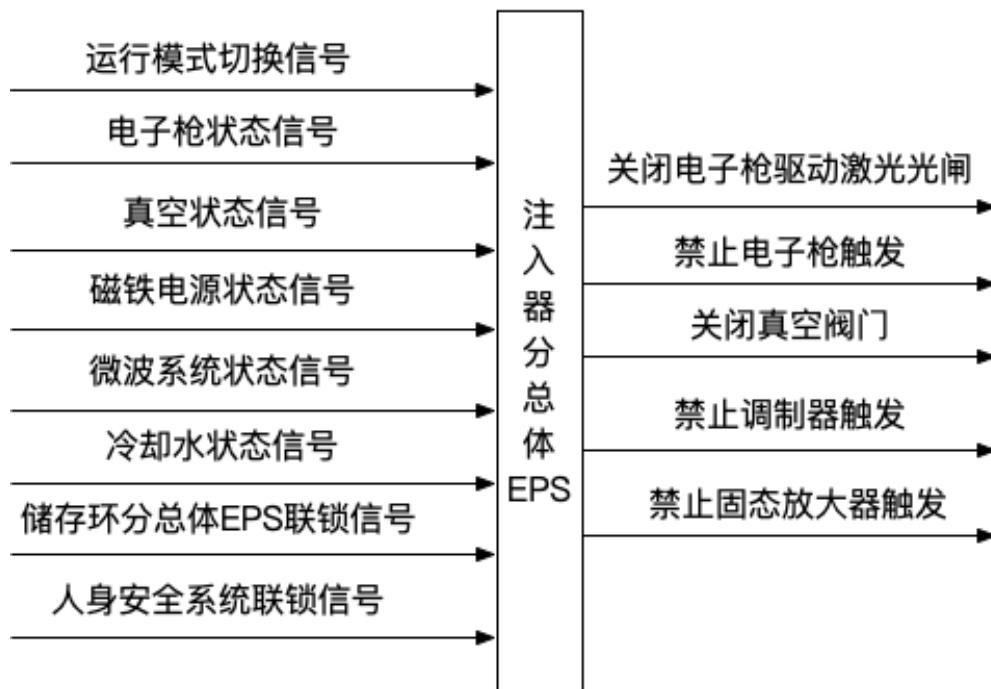


图 4.9 注入器 EPS 联锁保护功能图

注入器分总体范围内的真空度信号、冷却水流量和温度信号、电子枪系统状态信号、磁铁电源系统状态信号和微波系统状态信号；来自储存环分总体 EPS 的联锁信号；来自人身安全系统的联锁信号。注入器 EPS 提供的保护包括关闭电子枪的驱动激光光闸、关闭真空调门、禁止电子枪、调制器和固态放大器的时序触发信号。

根据目前 HALF 注入器的设计参数，注入器分总体 EPS 可由 1 台联锁主站和 13 台联锁从站组成，13 台从站按照束流的走向进行部署。直线加速器范围内部署了 11 台联锁从站，其中电子枪区域内的设备安全由 1 台从站负责，另外 10 台从站负责主加速器区域的设备安全，每台从站负责监测连续两个加速段区域内的设备状态信号。输运线范围内的设备较少，部署了 2 台联锁从站。

联锁主站具有运行模式切换的功能，综合当前运行模式和联锁输入信号，做出相应的逻辑判断，通过相应的从站实施保护。联锁从站负责所在区域内设备的安全，从站的联锁输入信号包括所在区域内的真空度信号、冷却水流量与温度信号以及相应的系统状态信号。联锁输入信号经过从站的联锁逻辑处理后，从站直接实施相应的本地保护，同时将联锁输入信号提供给主站进行联锁逻辑处理。

由于 HALF 工程还处于预研阶段，注入器各系统的设计细节尚未明确，所以注入器 EPS 各联锁从站的部署位置以及输入/输出信号数量还未确定，无法描述每个从站的联锁逻辑。下面我们从联锁保护逻辑的角度，以电子枪联锁系统、真空联锁系统和冷却水联锁系统为例，对这些系统相关联锁信号的保护逻辑进行描述。

#### 4.4.1 电子枪联锁系统

电子枪联锁系统的保护任务是对电子枪系统内的联锁信号提供本地保护，同时接收来自主站的保护命令为其他系统的联锁故障信号提供保护。图 4.10 所示的是电子枪联锁系统的结构图。电子枪联锁系统由电子枪联锁从站负责，输入信号包括电子枪冷却水温度/流量信号、电子枪真空度预警/报警信号、屏蔽间门状态信号、电子枪相应的速调管真空度预警/报警信号和冷却水温度/流量信号、相应的波导真空度报警/预警信号和冷却水温度/流量信号，经过电子枪联锁从站的逻辑判断后，电子枪联锁从站直接实施相应的的保护动作。

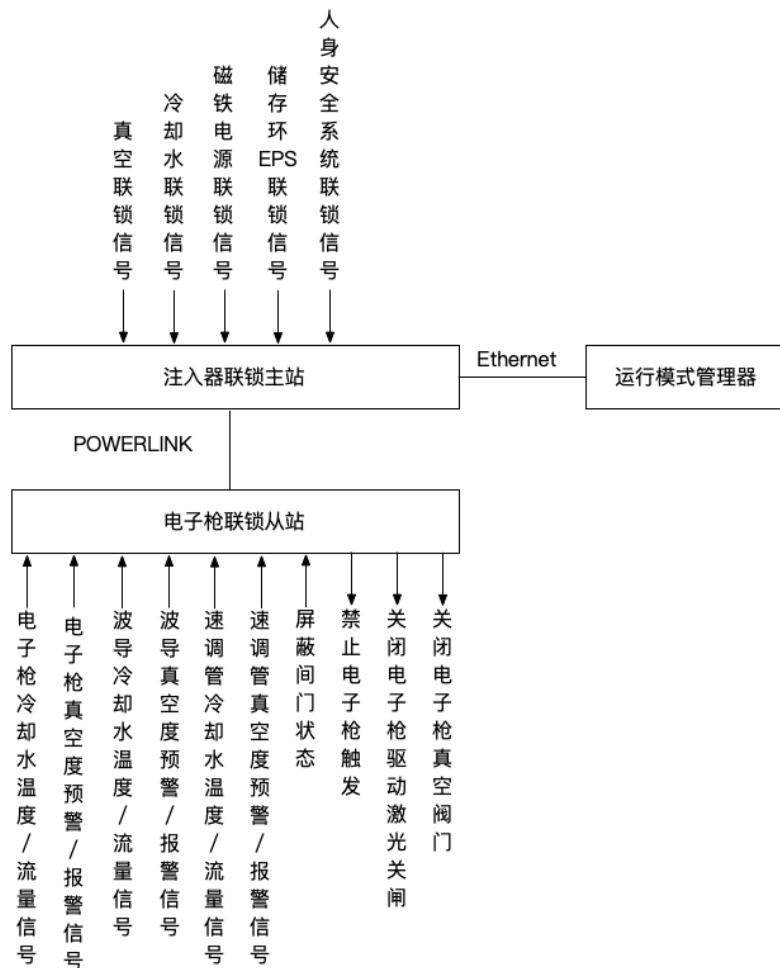


图 4.10 电子枪系统联锁功能图

电子枪联锁系统的本地保护逻辑如下：

- 当电子枪系统内以下联锁输入信号故障时，包括屏蔽间门状态信号、电子枪冷却水温度/流量信号和真空度预警信号、电子枪相应的速调管冷却水温度/流量信号和真空度预警信号、相应波导的冷却水温度/流量信号状态和真空度预警信号，电子枪联锁从站均会关闭驱动激光的光闸、禁止电子枪及其相应的调制器和固态放大器的时序触发信号以停止束流。

2. 当电子枪、相应的速调管和波导真空度报警时且束流已被打掉，电子枪联锁从站要立即关闭电子枪真空阀门以防止真空泄漏扩散，同时电子枪联锁从站还需要将电子枪真空度故障信号提供给联锁主站，通过相应的从站关闭其余真空阀门。

除了本地保护之外，当其他系统联锁信号出现故障时，注入器联锁主站需要通过电子枪联锁从站实施相应保护，具体的联锁保护逻辑如下：

1. 当注入器分总统范围内的真空度信号出现故障时，电子枪联锁从站会关闭电子枪真空阀门以防止真空泄漏扩散，同时关闭驱动激光的光闸和切断电子枪的时序触发信号以停止束流。
2. 当注入器分总体内冷却水温度/流量信号、磁铁电源状态信号、来自储存环分总体和人身安全系统的联锁信号出现故障时，电子枪联锁从站实施关闭驱动激光的光闸和切断电子枪时序触发信号的保护动作。

#### 4.4.2 真空联锁系统

真空联锁系统的任务是对注入器范围内的真空度联锁信号提供相应的保护措施，联锁输入信号包括真空度预警信号和报警信号，其中真空度预警值小于真空度报警值，图 4.11 所示的是真空联锁系统结构图，联锁输入信号具体包括电子枪区域内的电子枪、波导管、速调管的真空度预警/报警信号；主加速器区域内 20 个加速段的真空度预警/报警信号、40 根波导管的真空度预警/报警信号、20 台速调管的真空度预警/报警信号；输运线真空度预警/报警信号。真空联锁系统的输出保护信号包括禁止电子枪、21 台固态放大器和 21 台调制器的时序触发信号和关闭注入器范围内相应的真空阀门，其中注入器范围内的真空阀门共 22 台，分别位于电子枪出口处、每个加速段出口处和输运线出口处<sup>[?]</sup>。

电子枪的真空联锁保护逻辑已在电子枪联锁系统中进行了描述，其余设备的真空联锁保护逻辑具体如下：

1. 加速段真空保护：当加速段的真空度处于预警值和报警值之间时，负责监测此加速段的从站直接切断相应的固态放大器和调制器时序触发信号，同时将预警信息上传到联锁主站，通过电子枪从站实施关闭激光光闸、切断电子枪和电子枪对应的固态放大器和调制器时序触发信号。  
当加速段真空度大于报警值且束流已被打掉后，负责监测此加速段的从站直接关闭此区域的真空阀门，同时将报警信息上传到联锁主站，主站通过相应的从站关闭相应的真空阀门以防止真空进一步泄漏。
2. 速调管真空保护：当速调管的真空度处于预警值和报警值之间时，负责监测此速调管的从站直接切断相应的固态放大器和调制器时序触发信号，同时将预警信息上传到联锁主站，通过电子枪从站实施关闭激光光闸、切断

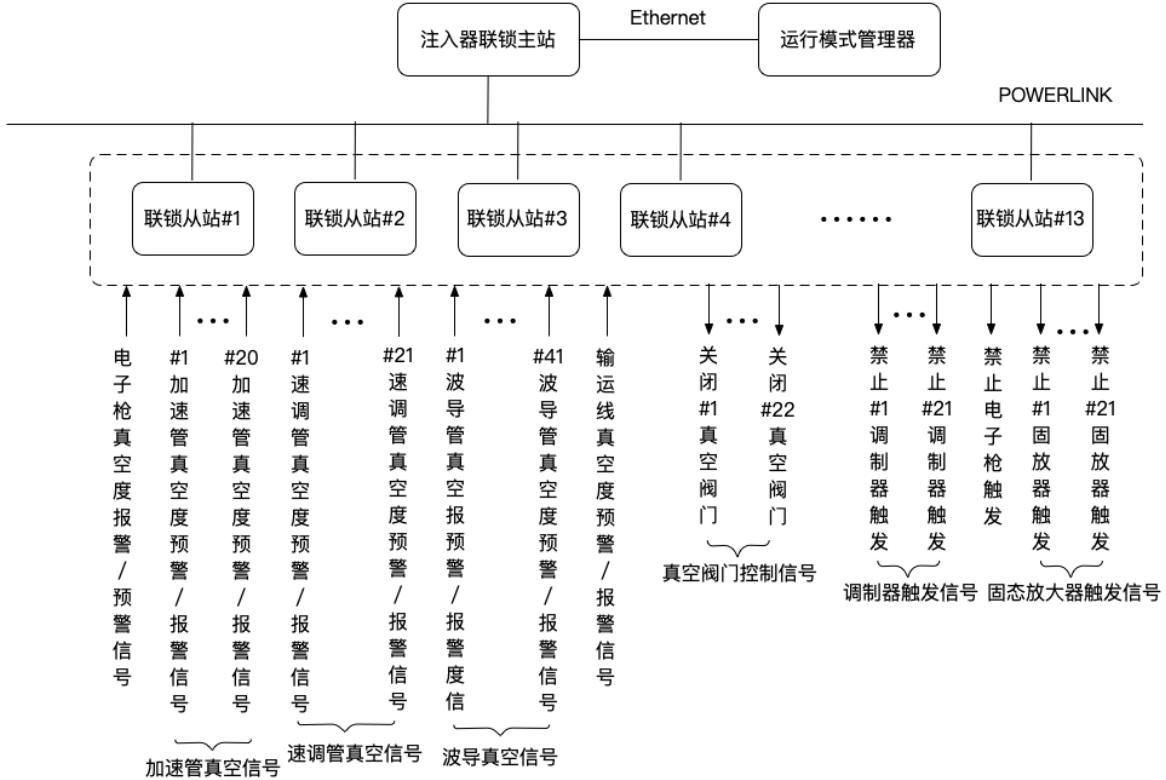


图 4.11 真空联锁系统结构图

电子枪和电子枪对应的固态放大器和调制器时序触发信号。

当速调管真空度大于报警值且束流已被打掉后，负责监测此速调管的从站直接该区域的真空阀门，同时将报警信息上传到联锁主站，主站通过相应的从站关闭相应的真空阀门以防止真空进一步泄漏。

3. 波导真空保护：当波导的真空度处于预警值和报警值之间时，负责监测此波导的从站直接切断相应的固态放大器和调制器时序触发信号，同时将预警信息上传到联锁主站，通过电子枪从站实施关闭激光光闸、切断电子枪和电子枪对应的固态放大器和调制器时序触发信号。

当波导真空度大于报警值且束流已被打掉后，负责监测此速调管的从站直接关闭该区域的真空阀门，同时将报警信息上传到联锁主站，主站通过相应的从站关闭相应的真空阀门以防止真空进一步泄漏。

4. 当输运线的真空度处于预警值和报警值之间时，输运线从站将预警信息上传到联锁主站，通过电子枪从站实施关闭激光光闸、切断电子枪和电子枪对应的固态放大器和调制器时序触发信号。注入器联锁主站还会通过将故障信息通过硬件信号传输给储存环联锁主站，储存环 EPS 实施禁止高频功率输出和禁止储存环注入的保护动作。

当输运线真空度大于报警值且束流已被打掉后，负责监测此输运线的从站直接关闭该区域的真空阀门，同时将报警信息上传到联锁主站，主站通过

相应的从站关闭相应的真空阀门以防止真空进一步泄漏。

#### 4.4.3 冷却水联锁系统

磁铁线圈与磁铁电源的冷却水温度/流量信号直接与其磁铁电源相连，综合成为一路电源状态信号输入到相应联锁从站，

冷却水联锁系统的任务是对注入器范围内的冷却水温度/流量信号提供相应的保护措施。冷却水联锁系统的结构如图 4.12 所示，冷却水联锁系统的输入信号是注入器各区域的冷却水温度/流量信号，具体包括电子枪及其速调管和波导的冷却水温度/流量信号、主加速器区域内 40 根加速管的冷却水温度/流量信号、40 根波导管的冷却水温度/流量信号和 20 台速调管的冷却水温度/流量信号，除此以外磁铁线圈与磁铁电源的冷却水温度/流量信号直接与其磁铁电源相连，综合成为一路电源状态信号输入到相应联锁从站。注入器电源分为直线段和输运线段 2 个单元，直线段预估包括中功率电源 3 台、小功率电源 84 台，输运线段预估包括大中功率电源 11 台、小功率电源 89 台，注入器的磁铁电源状态信号一共是 187 路<sup>[65]</sup>。

冷却水 EPS 的输出保护信号包括禁止电子枪、22 台固态放大器和 22 台调制器的时序触发信号。

电子枪冷却水联锁保护逻辑已在电子枪联锁系统中进行了描述，其余设备的冷却水联锁保护逻辑具体如下：

1. 磁铁及磁铁电源磁铁的冷却水联锁保护：当磁铁电源状态信号信号出现故障时，经过联锁主站逻辑处理后，通过电子枪从站实施关闭驱动激光的光闸和切断电子枪、调制器和固态放大器时序触发信号的保护动作。
2. 加速管冷却水联锁保护：当加速管的冷却水温度/流量信号出现故障时，负责此区域的从站直接切断此加速管对应的固态放大器和调制器时序触发信号；同时将故障信息上传到联锁主站，主站通过电子枪从站关闭驱动激光的光闸和切断电子枪、调制器和固态放大器时序触发信号。
3. 速调管冷却水保护：当速调管的冷却水温度/流量信号出现故障时，负责此区域的从站直接切断此速调管对应的固态放大器和调制器时序触发信号。
4. 波导冷却水保护：当波导的冷却水温度/流量信号出现故障时，负责此区域的从站直接切断此波导对应的固态放大器和调制器时序触发信号。

#### 4.4.4 注入器分总体 EPS 联锁信号总结

根据最新的 HALF 设计参数，我们对注入器 EPS 分总体的联锁信号进行了初步统计，各系统联锁信号数量如表 4.2 所示。

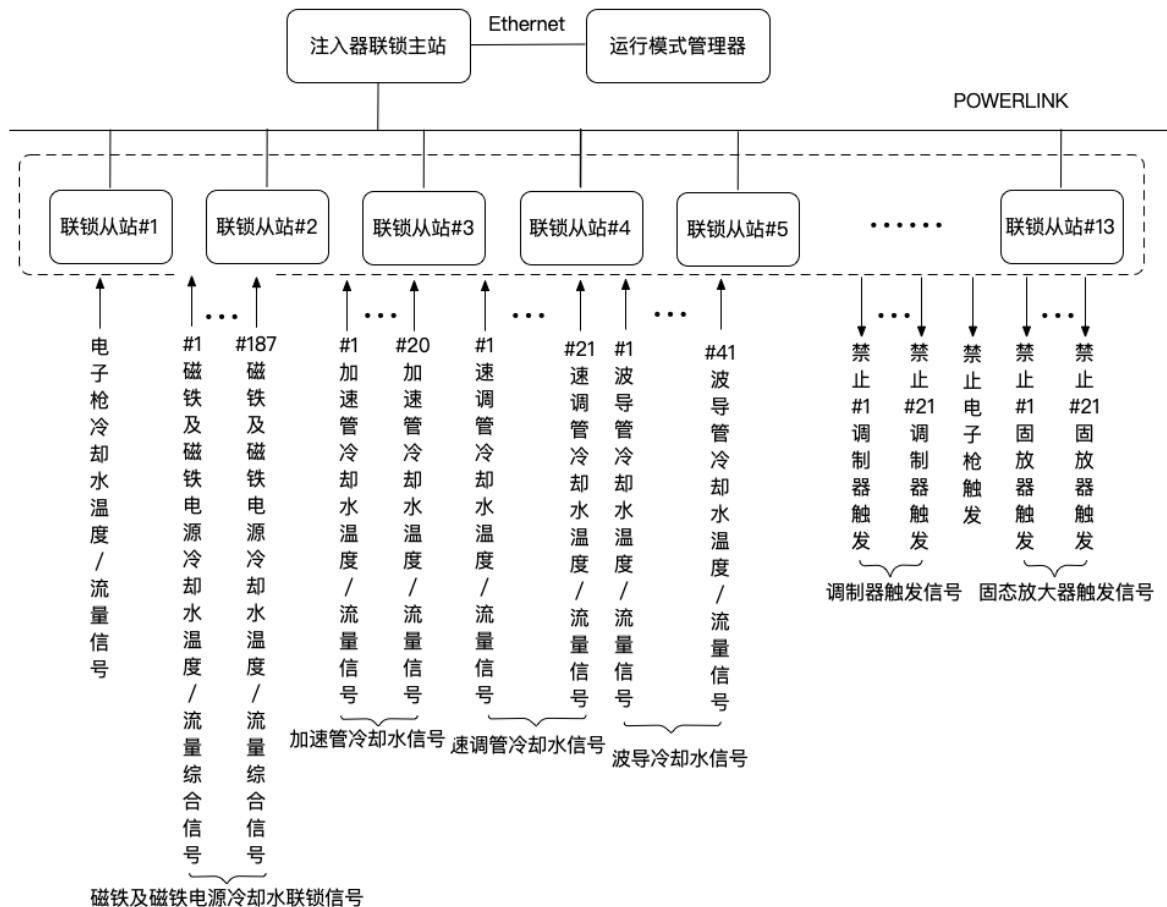


图 4.12 冷却水 EPS 联锁保护功能图

表 4.2 注入器 EPS 的联锁信号数量统计表

系统名称	联锁输入信号	联锁保护信号
电子枪联锁系统	1	2
真空联锁系统	168	22
磁铁电源联锁系统	187	0
微波联锁系统	21	42
冷却水联锁系统	166	0
储存环分总体 EPS	1	0
人身安全系统	1	0
总计	545	66

#### 4.4.5 注入器设备保护系统实时性能评估

根据注入器分总体设备保护系统的设计方案，我们可以将第 3.2 节中全站 FPGA 方案的测试系统作为注入器分总体 EPS 的原型系统，但是由于原型系统的从站数量仅为 5 台，无法满足注入器设备保护系统 13 台联锁从站的规模。下

面我们将根据注入器设备保护系统的设计规模和具体联锁信号数量，采用理论计算和网络模拟两种方法对注入器设备保护系统的实时性能进行评估。

### 1. 理论计算

我们首先对注入器设备保护系统的通信周期进行分析计算。通信周期由等时同步阶段和空闲阶段组成，等时同步阶段的时长公式为 4.1，其中  $T_{sync}$  为 SoC 数据帧同步各从站的时间， $T_{sync}$  由 SoC 数据帧的传输时间和各从站接收并处理 SoC 帧的时间组成，根据第 3.3 节中对同步阶段的理论分析， $T_{sync}$  的最大值一般不超过  $1\mu s$ 。

$$T_{ip} = T_{sync} + T_{poll} \quad (4.1)$$

轮询阶段时长  $T_{poll}$  按照公式 4.2 来计算，其中  $\sum_{i=1}^{13} (T_{F(PRes)}^i + T_{F(PReq)}^i)$  为 13 个联锁从站的 PReq/PRes 数据帧的传输时间之和。注入器设备保护系统系统中所有 PReq/PRes 帧的总数据量可通过公式 4.3 进行计算，总数据量 ( $\sum_{i=1}^{13} (DataSize_{PReq}^i + DataSize_{PRes}^i)$ ) 由 PReq/PRes 数据帧的帧头帧尾总数据量 ( $DataSize_{total}^{frame}$ ) 和 PReq/PRes 数据帧的应用层总数据量 ( $DataSize_{total}^{user\_data}$ ) 组成。根据第 2.2.2 节中对 POWERLINK 数据帧结构的分析，PReq 数据帧和 PRes 数据帧的帧头帧尾大小 ( $DataSize_{format}^{PReq}$ ,  $DataSize_{format}^{PRes}$ ) 均为 21Byte，根据公式 4.4，可得 13 个联锁从站的 PReq/PRes 数据帧的帧头帧尾总数据量 ( $DataSize_{total}^{format}$ ) 等于 546Byte。

PReq 和 PRes 数据帧中的应用层数据分别对应于该从站的输出信号和输入信号，考虑到与联锁信号相关的锁存、旁路和复位预处理状态信号也需要通过 PReq/PRes 数据帧参与通信，我们按照一个输入/输出信号的相关数据大小为 1Byte 来进行计算，根据公式 4.5，PReq 和 PRes 数据帧中的应用层数据即等于系统联锁信号的总数量。根据表 4.2 的统计结果，注入器设备保护系统中联锁信号的总数量为 611，考虑到保护设备的状态回读信号，我们在计算时将联锁信号数量翻倍，即按照 1222 个总联锁信号数量来进行计算。

我们将  $DataSize_{total}^{format\_data}$  和  $DataSize_{total}^{user\_data}$  的数值代入公式 4.3，得到系统中所有 PReq/PRes 数据帧的总大小为 1768Byte，按照千兆以太网的速率进行数据传输，可计算出 PReq/PRes 数据帧传输时间之和 ( $\sum_{i=1}^{13} (T_{F(PReq)}^i + T_{F(PRes)}^i)$ ) 为  $14.144\mu s$ 。

$$T_{poll} = \sum_{i=1}^{13} (T_{F(PReq)}^i + T_{F(PRes)}^i) + T_{net}^{line} \quad (4.2)$$

$$\sum_{i=1}^{13} (DataSize_{PReq}^i + DataSize_{PRes}^i) = DataSize_{total}^{format} + DataSize_{total}^{user\_data} \quad (4.3)$$

$$DataSize_{total}^{format} = 13(DataSize_{format}^{PReq} + DataSize_{format}^{PRes}) \quad (4.4)$$

$$D_{total}^{user\_data} = \sum_{i=1}^{13} (D_{InputSignal}^i + D_{outputSignal}^i) \quad (4.5)$$

公式 4.2 中的  $T_{net}^{line}$  为注入器设备保护系统的网络组件延时，各联锁从站按照线型拓扑结构相连， $T_{net}^{line}$  可按照公式 4.6 进行计算，其中  $T_c$  为相邻两节点之间网线的传输延时，注入器 260m 长度，这里假定各联锁从站按照等距离部署，相邻从站之间的距离为 22m，按照 5ns/m 的传输延时计算， $T_c$  为  $0.11\mu s$ 。 $T_{hub}$ 、 $T_{r-cn}$ 、 $T_{r-mn}$  参数根据第 3.3 节中系统的实测结果，分别为  $1.865\mu s$ 、 $1.048\mu s$  和  $0.66\mu s$ ，代入公式 4.3，可得  $T_{net}^{line}$  为  $169.429\mu s$ 。

$$T_{net}^{line} = \sum_{i=1}^{13} [2iT_c + (2i - 1)T_{hub} + T_{r-cn}] + nT_{r-mn} \quad (4.6)$$

将  $T_{net}^{line}$  和  $\sum_{i=1}^{13} (T_{F(PReq)}^i + T_{F(PRes)}^i)$  代入公式 4.2，可得轮询阶段时长  $T_{poll}$  为  $193.349\mu s$ ，加上同步阶段的时长  $T_{sync}$ ，则等时同步阶段的时长  $T_{ip}$  为  $194.349\mu s$ 。

考虑到空闲阶段时长为定值，为传输最大异步数据预留的时间，根据第 3.3 节中系统的实测结果为  $12.9\mu s$ ，可得注入器设备保护系统通信周期为  $206.249\mu s$ 。

根据第 3.2.5 节中对系统响应时间的分析，耗时最长的响应过程为联锁输入信号从 1 号从站输入，经过主站处理后，通过 13 号从站输出保护动作，此过程跨越 3 个周期，考虑到输入信号的通信等待时间和从站数字隔离器  $5\mu s$  的信号处理时间，注入器设备保护系统的最长响应时间为  $829.996\mu s$ 。

## 2. 网络模拟

基于 OMNeT++ 仿真平台，我们对注入器设备保护系统进行了建模，模型如图 4.13 所示。

根据第 3.3 对通信周期的理论分析，各联锁从站被分配的通信时隙 ( $T_{cn-slot}^i$ ) 可通过公式 4.7 和公式 4.8 计算得到，其中各从站的 PReq 数据帧和 PRes 数据帧的传输时间  $T_{F(PReq)}^i$  和  $T_{F(PRes)}^i$  跟该从站的输入和输出信号数量有关，我们根据第 4.4 节中注入器设备保护系统各从站的部署情况，可以对各从站的联锁信

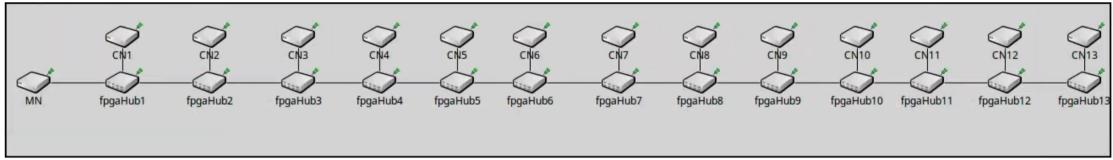


图 4.13 注入器设备保护系统模型

号数量进行估算，考虑到保护的回读状态信号，这里同样将联锁信号数量翻倍来进行计算。 $T_{hub}$ 、 $T_{r-cn}$ 、 $T_{r-mn}$  参数按照第 3.3 节中系统的实测结果，分别为  $0.66\mu s$ 、 $1.865\mu s$  和  $1.048\mu s$ ，根据公式 4.8 可以计算出各联锁从站被配置的通信时隙 (PResTimeout)。

$$T_{rtd}^i = 2iT_c + (2i - 1)T_{hub} + T_{r-cn} \quad (4.7)$$

$$T_{cn-slot}^i = T_{F(PReq)}^i + T_{F(PRes)}^i + T_{rtd}^i + T_{r-mn} \quad (4.8)$$

各从站具体配置如下所示，其中 PResTimeout 为各联锁从站的通信时隙，PReqActPayload 为各从站的输出信号大小。

```

1 <Nodes>
2   <Node NodeId="1" PResTimeout="6129ns" PReqActPayload="5byte" />
3   <Node NodeId="2" PResTimeout="6998ns" PReqActPayload="6byte" />
4   <Node NodeId="3" PResTimeout="8538ns" PReqActPayload="6byte" />
5   <Node NodeId="4" PResTimeout="10078ns" PReqActPayload="6byte" />
6   <Node NodeId="5" PResTimeout="11618ns" PReqActPayload="6byte" />
7   <Node NodeId="6" PResTimeout="13158ns" PReqActPayload="6byte" />
8   <Node NodeId="7" PResTimeout="14698ns" PReqActPayload="6byte" />
9   <Node NodeId="8" PResTimeout="16238ns" PReqActPayload="6byte" />
10  <Node NodeId="9" PResTimeout="17778ns" PReqActPayload="6byte" />
11  <Node NodeId="10" PResTimeout="19318ns" PReqActPayload="6byte" />
12  <Node NodeId="11" PResTimeout="20858ns" PReqActPayload="6byte" />
13  <Node NodeId="12" PResTimeout="24749ns" PReqActPayload="1byte" />
14  <Node NodeId="13" PResTimeout="26289ns" PReqActPayload="1byte" />
15 </Nodes>
```

模拟过程持续了大约 2000 个 POWERLINK 通信周期。等时同步阶段的概率密度分布绘制在图 4.14 中，该图表明注入器设备保护系统轮询平均时长为  $189.6047\mu s$ ，与通过理论计算得到的轮询阶段时长  $193.349\mu s$  相比，网络模拟得到的结果略低，原因是大多数从站在分配的通信时隙内可提前完成相应的通信。考虑到同步阶段时长  $T_{sync}$  最大为  $1\mu s$  和  $12.9\mu s$  的空闲时长，可得注入器设备保护系统通信周期为  $203.5047\mu s$ 。考虑到输入信号的通信等待时间和从站数字隔离器  $5\mu s$  的信号处理时间，注入器设备保护系统的最长响应时间为  $819.0188\mu s$ 。

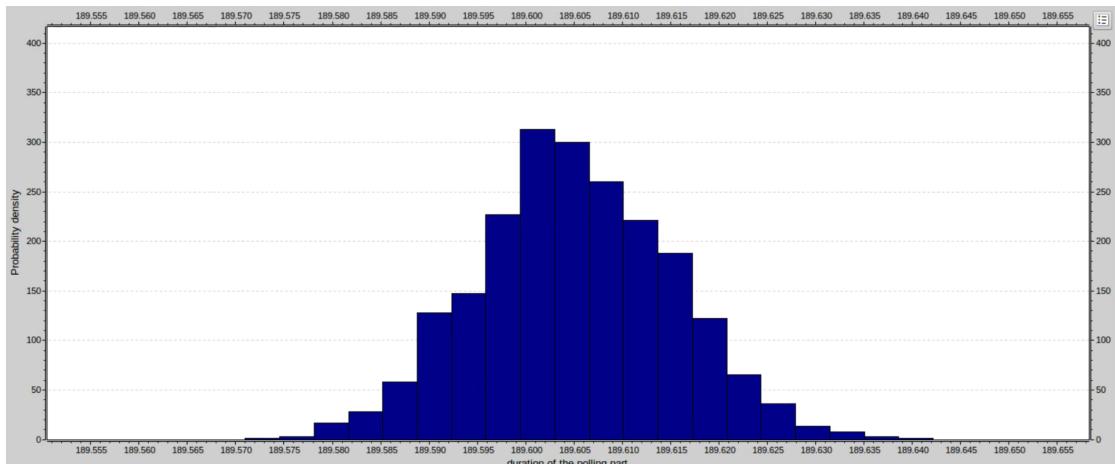


图 4.14 注入器设备保护系统模型轮询阶段时长的概率密度分布

### 3. 小结

我们通过理论计算和仿真模拟的方法两种方法对注入器设备保护系统的实时性能进行了估算，两种方法的计算结果比较相近，通信周期的估算结果分别为  $206.249\mu s$  和  $203.5047\mu s$ ，最大响应时间的估算结果分别为  $829.996\mu s$  和  $819.0188\mu s$ ，通过理论计算和仿真模拟的方法两种方法得到的结果较接近，均完全满足设备保护系统响应时间 10ms 响应时间的需求。

## 4.5 储存环分总体 EPS 设计

HALF 储存环由 20 个全同的多弯铁消色散结构 (MBA) 的聚焦单元组成，每个单元长度为 24m，可提供一个 5.5m 的长直线节和一个 2.2m 的中直线节。储存环周长为 480m，可以提供 20 个长直线节和 20 个中直线节，除了两个直线节分别安装注入系统和高频系统外，其中至少 35 个直线节可用于安装插入元件，目前一期 HALF 工程计划建设 10 条插入件光束线。

储存环分总体 EPS 的任务是当储存环分总体内各系统设备出现故障或者接收到来自储存环分总体外部的联锁故障信号时，经过储存环联锁主站的逻辑处理后，通过联锁从站实施相应的保护动作。图 4.15 描述了储存环分总体 EPS 的联锁保护功能。

储存环 EPS 的输入联锁信号较多，包括来自运行管理系统的联锁模式切换信号；储存环分总体范围内的真空度信号、冷却水温度/流量信号、真空部件温度信号、磁铁电源状态信号、高频系统状态信号、注入系统状态信号、插入件系统状态信号；来自光束线站和人身安全系统的联锁信号。储存环 EPS 提供的保护措施具体如下：关闭储存环范围内的真空阀门、禁止高频功率输出、禁止储存环注入和通过注入器联锁主站停止电子枪出束。

根据目前 HALF 储存环的设计参数，储存环分总体 EPS 可由 1 台联锁主站

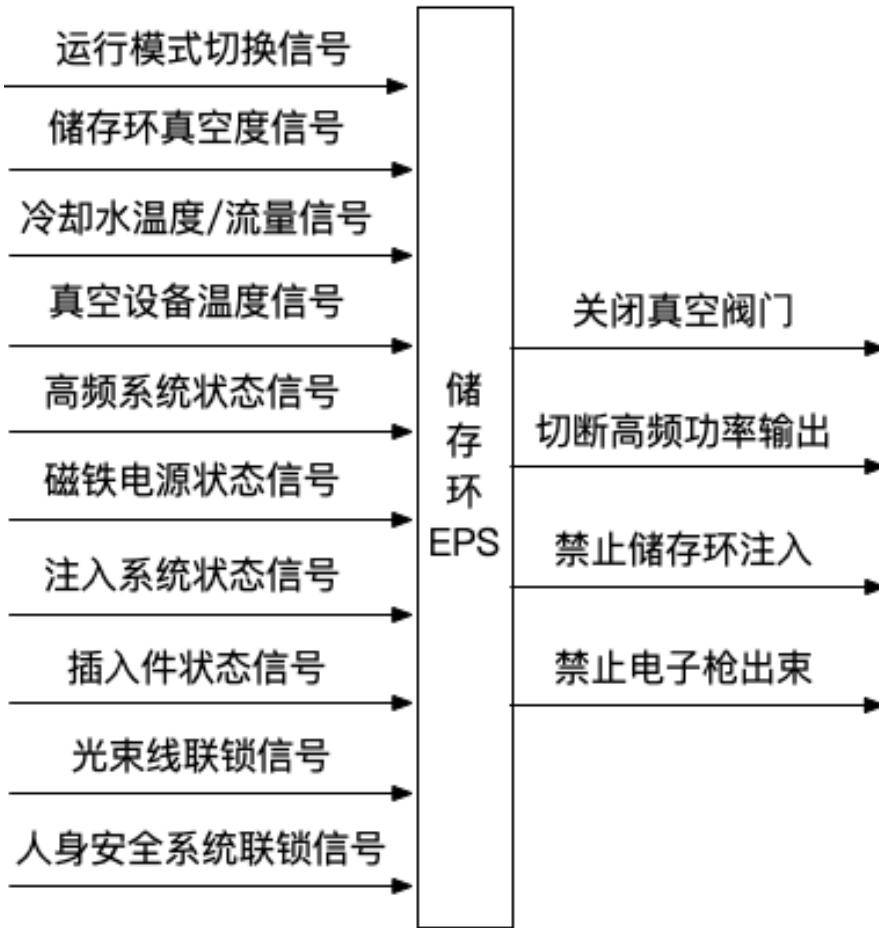


图 4.15 储存环 EPS 联锁保护功能图

和 20 台联锁从站组成，20 台从站按照 20 个 MBA 单元的位置部署在整个储存环分总体范围内。联锁主站具有运行模式切换的功能，综合当前运行模式和联锁输入信号，做出相应的逻辑判断，通过相应的从站实施保护措施。20 台联锁从站负责所在区域内设备的安全，从站的联锁输入信号包括所在区域内的真空度信号、冷却水流量/温度信号、真空设备温度信号以及相应的系统状态信号。联锁输入信号经过从站的联锁逻辑处理后，从站直接实施相应的本地保护，同时将联锁输入信号提供给主站进行逻辑处理。

由于 HALF 工程还处于预研阶段，储存环各系统的设计细节尚未明确，所以储存环 EPS 各联锁从站的部署位置以及输入/输出信号数量还未确定，无法描述每个从站的联锁逻辑。下面我们从联锁保护逻辑的角度，以真空联锁系统、冷却水联锁系统、真空部件温度联锁系统、高频联锁系统和注入联锁系统为例，对这些系统相关联锁信号的保护逻辑进行描述。

#### 4.5.1 真空联锁系统

真空联锁系统的输入信号是储存环范围内的真空度信号，包括真空度预警信号和报警信号，其中真空度预警值小于真空度报警值。全环一共 120 个真空度

联锁输入信号，由20个联锁从站负责，每个从站负责相应区域内的3个真空度预警信号和3个真空度报警信号，这里暂时不考虑光束线前端区和光束线站的真空度联锁信号，光束线的真空保护由光束线联锁保护系统负责。联锁从站的输出信号是真空阀门控制信号，全环共20台真空阀门，分别位于相邻两个MBA单元之间。

图4.16所示的是真空联锁系统结构图。

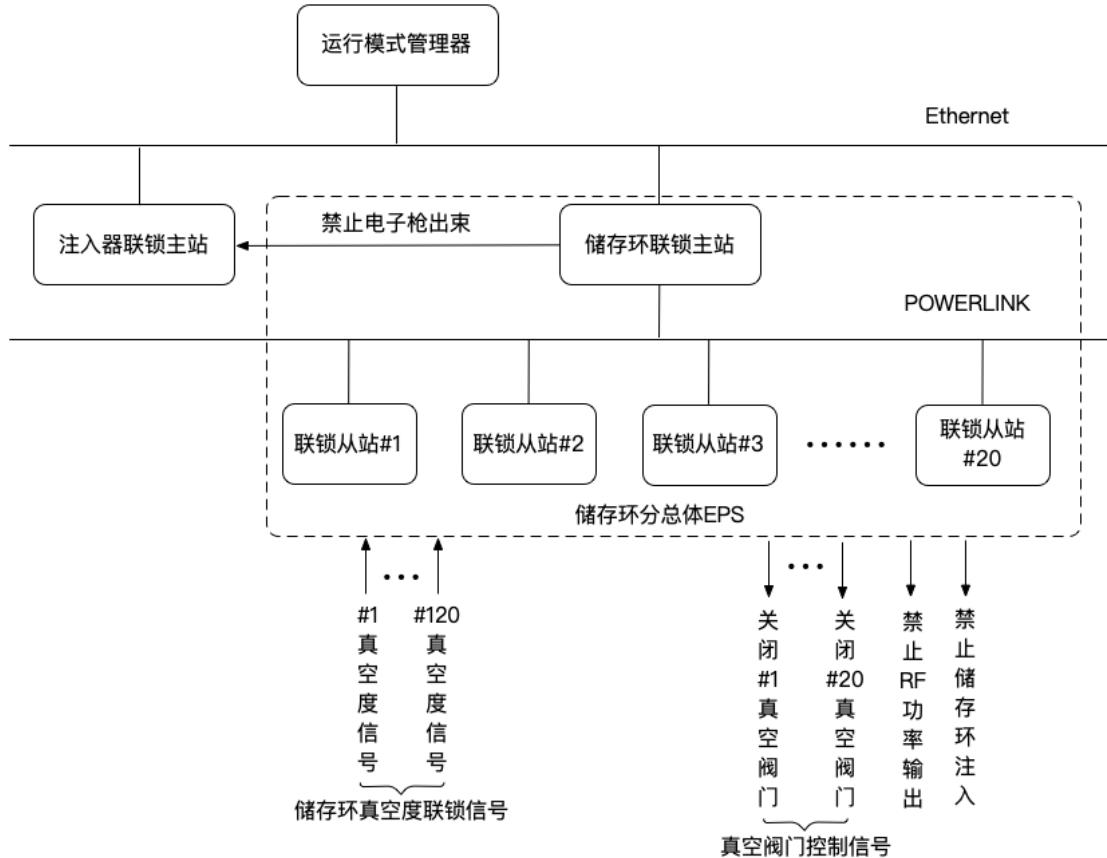


图4.16 真空联锁系统结构图

对于真空度预警信号和真空度报警信号，真空联锁系统的具体保护措施如下：

1. 当储存环真空度处于预警值和报警值之间时，负责此区域的联锁从站将故障信息上传到联锁主站，主站通过相应的从站实施禁止高频功率输出和储存环注入的保护动作，同时将联锁故障信号传输给注入器联锁主站，通过电子枪从站实施关闭激光光闸和切断电子枪时序触发信号的保护动作。
2. 当储存环真空度高于报警值且束流已被打掉后，负责监测此真空度信号的从站直接关闭此单元的真空阀门，同时将报警信息上传到联锁主站，主站通过相应的从站关闭相应的真空阀门以防止真空进一步泄漏。

### 4.5.2 冷却水联锁系统

冷却水联锁系统的输入信号是储存环各区域的冷却水温度/流量信号，号具体如下：

1. 真空部件的冷却水温度/流量信号：每个联锁从站负责相应区域的弯段、光子吸收器和波纹管共 24 个冷却水温度/流量信号，全环共 480 个真空部件的冷却水温度/流量信号；
2. 磁铁及磁铁电源的冷却水温度/流量信号：储存环磁铁电源系统设计为 20 个单元，每个单元磁铁包括 6 块纵向梯度二极铁、4 块二四极组合铁、16 块四极铁、6 块六极铁、2 台八极铁和 12 块校正铁，其中四极铁和校正铁均单独供电，其他磁铁是分成若干组串联供电。各单元中的磁铁线圈与磁铁电源的冷却水温度/流量信号直接与其磁铁电源相连，综合成为一路电源状态信号输入到相应联锁从站，每个从站的联锁输入信号具体包括 16 路四级铁电源状态信号、12 路校正铁电源状态信号、1 路二级铁电源状态信号、1 路六级铁电源状态信号和 1 路八级铁电源状态信号，共 31 路联锁输入信号，全环共 620 路磁铁电源状态信号。
3. 10 个插入件的冷却水温度/流量信号，由相应的联锁从站负责。

图 4.17 所示的是冷却水联锁系统功能图，

冷却水联锁系统的具体联锁保护逻辑如下：

1. 真空部件冷却水联锁保护：当真空盒、光子吸收器和波纹管等真空部件的冷却水温度高于阈值，或这些这些真空部件的冷却水流量低于阈值时，相应的联锁从站将故障信息提供给联锁主站，经过主站的逻辑处理后，通过高频系统从站切断高频功率输出；通过注入系统联锁从站实施禁止储存环注入的保护动作；同时将联锁故障信号传输给注入器联锁主站，通过电子枪从站关闭激光光闸并切断电子枪、调制器和固态放大器时序触发信号。
2. 高频腔冷却水联锁保护：当高频腔冷却水温度/流量信号出现故障时，高频腔联锁从站实施切断高频功率输出的本地保护措施；同时从站将故障信息上传到联锁主站，经过联锁主站逻辑处理后，通过注入系统联锁从站实施禁止储存环注入的保护动作；同时将联锁故障信号传输给注入器联锁主站，通过电子枪从站关闭激光光闸并切断电子枪、调制器和固态放大器时序触发信号。
3. 磁铁冷却水联锁保护：当磁铁线圈的冷却水温度/流量信号出现故障时，相应的联锁从站将故障信息提供给联锁主站，经过主站逻辑处理后，通过高频系统从站切断高频功率输出；通过注入系统联锁从站实施禁止储存环注入的保护动作；同时将联锁故障信号传输给注入器联锁主站，通过电子枪

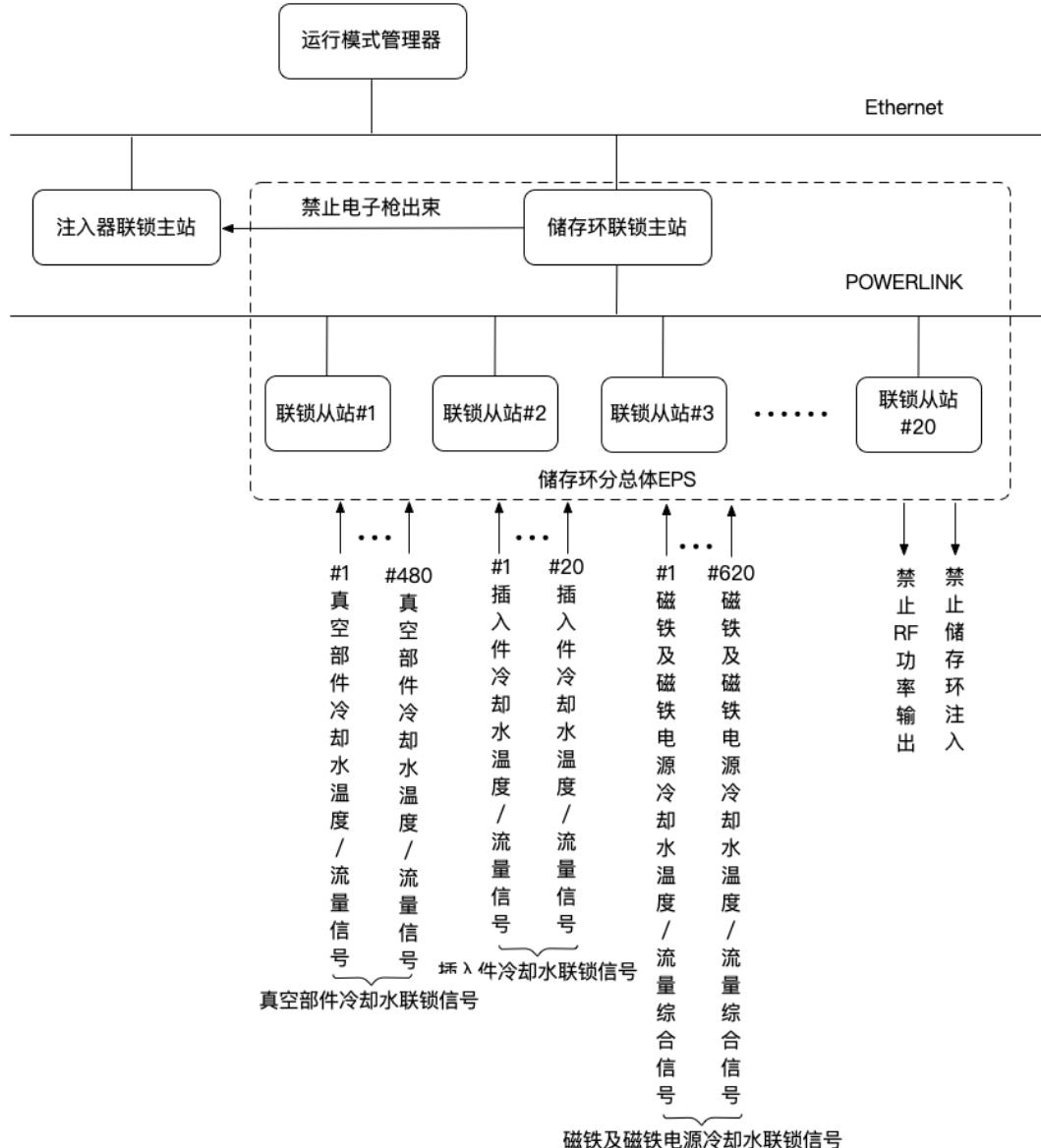


图 4.17 冷却水联锁系统结构图

从站关闭激光光闸并切断电子枪、调制器和固态放大器时序触发信号。

4. 插入件冷却水联锁保护：当插入件的冷却水温度/流量信号出现故障时，相应的联锁从站将故障信息提供给联锁主站，经过联锁主站逻辑处理后，通过高频系统从站切断高频功率输出；通过注入系统联锁从站实施禁止储存环注入的保护动作；同时将联锁故障信号传输给注入器联锁主站，通过电子枪从站关闭激光光闸并切断电子枪、调制器和固态放大器时序触发信号。

### 4.5.3 真空部件温度联锁系统

真空部件温度联锁系统的输入信号是储存环范围内真空盒、光子吸收器和波纹管等真空部件的外壁温度信号，每个联锁从站包括相应区域的真空部件共 30 个温度信号，全环共 600 个真空部件温度联锁信号，

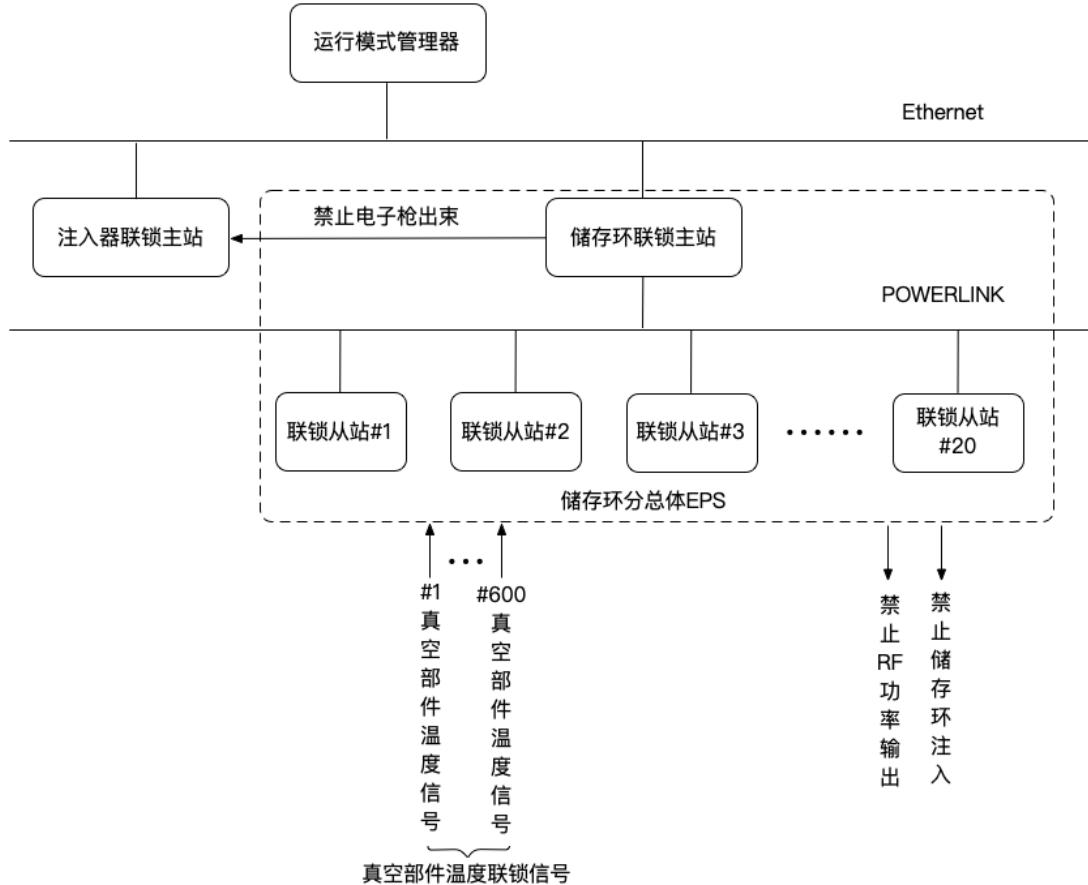


图 4.18 真空部件温度联锁系统结构图

图 4.11 所示的是真空部件温度联锁系统结构图。当真空盒、光子吸收器和波纹管等部件的外壁温度超过阈值，相应的联锁从站将故障信息提供给联锁主站，经过主站逻辑处理后，通过高频系统从站切断高频功率输出；通过注入系统联锁从站实施禁止储存环注入的保护动作；同时将联锁故障信号传输给注入器联锁主站，通过电子枪从站关闭激光光闸并切断电子枪、调制器和固态放大器时序触发信号。

#### 4.5.4 高频联锁系统

高频联锁系统的任务是对高频系统内的联锁输入信号提供本地保护措施，同时接收来自主站的保护命令为其他系统的联锁故障信号实施保护措施。图 4.19 所示的是高频联锁系统结构图。高频联锁系统由高频从站负责，从站的输入信号包括高频腔冷却水温度/流量信号、高频腔真空度预警/报警信号、高频腔温度信号和高频腔的输入和反射功率，输出保护动作是切断高频功率的输出。

当高频系统内的联锁信号出现故障时，高频腔联锁从站直接实施切断高频功率输出的本地保护动作；同时将联锁故障信号传输给注入器联锁主站，经过联锁主站逻辑处理后通过注入系统联锁从站禁止储存环注入，将联锁故障信号传

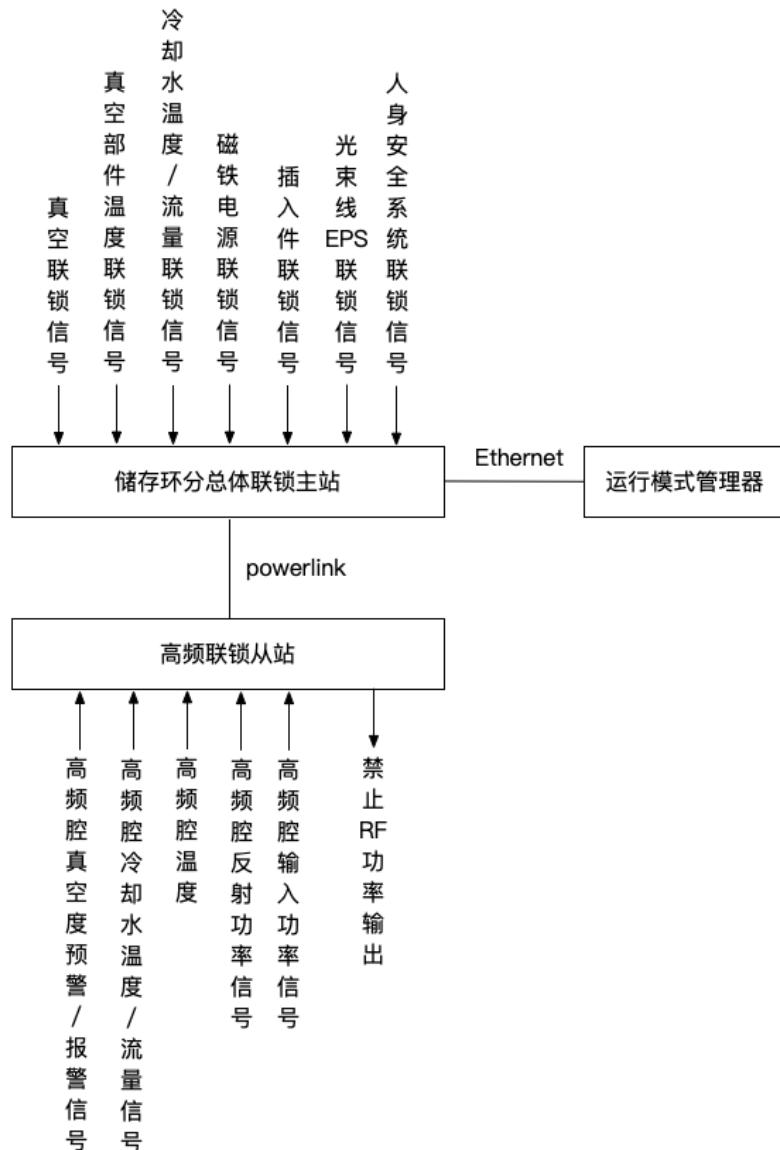


图 4.19 高频联锁系统结构图

输给注入器联锁主站，通过电子枪从站实施关闭激光光闸和切断电子枪、调制器和固态放大器序触发信号的保护动作。

当其他系统联锁信号出现故障时，联锁主站需要通过高频系统联锁从站实施相应保护，具体的联锁保护逻辑如下：当储存环分总体内真空度信号、真空部件温度信号、冷却水温度/流量信号、磁铁电源状态信号、来自注入器分总体联锁信号、人身安全系统联锁信号出现故障时，经过主站联锁逻辑处理后，通过高频系统从站实施切断高频功率输出的保护动作。

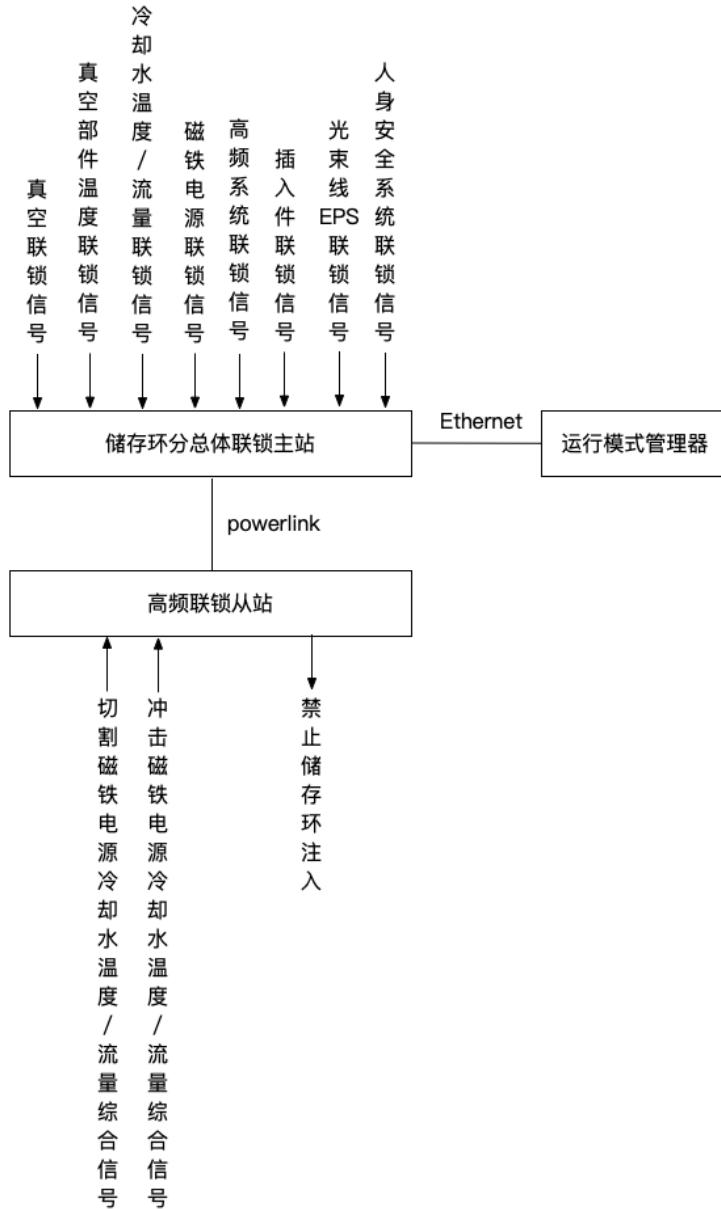


图 4.20 注入联锁系统功能图

#### 4.5.5 注入联锁系统

注入联锁系统是对注入系统内的联锁输入信号提供本地保护措施，同时接收来自主站的保护命令为其他系统的联锁故障信号实施保护措施。图 4.20 所示的是注入联锁系统结构图。注入联锁系统由注入从站负责，从站的输入信号包括冲击磁铁和切割磁铁的磁铁电源综合状态信号，输出保护动作是禁止储存环注入。

当注入系统内的磁铁电源综合状态信号出现故障时，注入联锁从站直接实施禁止储存环注入的本地保护动作；同时将联锁故障信号传输给储存环联锁主站，经过主站逻辑处理后，通过高频系统联锁从站切断高频功率输出；同时将联

锁故障信号传输给注入器联锁主站，通过电子枪从站实施关闭激光光闸和切断电子枪触发信号的保护动作。

当其他系统联锁信号出现故障时，联锁主站需要通过注入系统联锁从站实施相应保护，具体的联锁保护逻辑如下：当储存环分总体内真空度信号、真空部件温度信号、冷却水温度/流量信号、磁铁电源状态信号、来自注入器分总体联锁信号、人身安全系统联锁信号出现故障时，经过主站联锁逻辑处理后，通过高频系统实施禁止储存环注入的保护动作。

#### 4.5.6 储存环分总体 EPS 联锁信号总结

根据最新的 HALF 设计参数，我们对储存环分总体 EPS 的联锁信号进行了初步统计，各系统联锁信号数量如表 4.3 所示。

表 4.3 储存环分总体 EPS 的联锁信号数量统计表

系统名称	联锁输入信号	联锁保护信号
真空联锁系统	120	30
冷却水联锁系统	500	0
真空部件温度联锁系统	600	0
磁铁电源联锁系统	620	0
高频联锁系统	2	1
注入联锁系统	2	1
插入件联锁系统	20	0
光束线联锁系统	10	0
人身安全系统	1	0
总计	1875	32

#### 4.5.7 储存环设备保护系统实时性能评估

根据储存环分总体设备保护系统的设计方案，我们同样可以将第 3.2 节中基于 POWERLINK 的分布式 IO 系统作为储存环分总体 EPS 的原型系统，结合储存环设备保护系统的设计规模和具体联锁信号数量，采用理论计算和网络模拟两种方法对储存环设备保护系统的实时性能进行评估。

##### 1. 理论计算

我们首先对储存环分总体的设备保护系统的通信周期进行分析计算。通信周期由等时同步阶段和空闲阶段组成，等时同步阶段的时长可按照公式为 4.9，其中  $T_{sync}$  为 SoC 数据帧同步各从站的时间，根据第 3.3 节中对同步阶段的理论

分析,  $T_{sync}$  的最大值一般不超过  $1\mu s$ 。

$$T_{ip} = T_{sync} + T_{poll} \quad (4.9)$$

轮询阶段时长  $T_{poll}$  按照公式 4.10 来计算, 其中  $\sum_{i=1}^{20}(T_{F(PReq)}^i + T_{F(PRes)}^i)$  为 20 个联锁从站的 PReq/PRes 数据帧的传输时间之和。储存环设备保护系统系统中所有 PReq/PRes 帧的总数据量 ( $DataSize_{total}^{PReq/PRes}$ ) 可通过公式 4.11 进行计算, 总数据量由 PReq/PRes 数据帧的帧头帧尾总数据量 ( $DataSize_{total}^{format}$ ) 和 PReq/PRes 数据帧的应用层总数据量 ( $DataSize_{total}^{user\_data}$ ) 组成。根据第 2.2.2 节中对 POWERLINK 数据帧结构的分析, PReq 数据帧和 PRes 数据帧的帧头帧尾大小 ( $DataSize_{format}^{PReq}$ ,  $DataSize_{format}^{PRes}$ ) 均为 21Byte, 根据公式 4.12, 可得 20 个联锁从站的 PReq/PRes 数据帧的帧头帧尾总数据量 ( $DataSize_{total}^{format}$ ) 等于 840Byte。

PReq 和 PRes 数据帧中的应用层数据分别对应于该从站的输出信号和输入信号, 考虑到与联锁信号相关的锁存、旁路等预处理状态信号, 我们按照一个输入/输出信号的相关数据大小为 1Byte 来进行计算, 根据公式 4.13, PReq 和 PRes 数据帧中的应用层总数据量即等于系统联锁信号的总数量。根据表 4.2 的统计结果, 储存环设备保护系统中联锁信号的总数量为 1907, 考虑到保护设备的回读状态信号, 这里我们将联锁信号数量翻倍, 即按照 3814 个总联锁信号数量来进行计算。

我们将  $DataSize_{total}^{format}$  和  $DataSize_{total}^{user\_data}$  的数值代入公式 4.11, 得到系统中所有 PReq/PRes 数据帧的总大小为 4654Byte, 按照千兆以太网的速率进行数据传输, 可计算出 PReq/PRes 数据帧传输时间之和 ( $\sum_{i=1}^{20}(T_{F(PReq)}^i + T_{F(PRes)}^i)$ ) 为  $37.232\mu s$ 。

$$T_{poll} = \sum_{i=1}^{20}(T_{F(PReq)}^i + T_{F(PRes)}^i) + T_{net}^{line} \quad (4.10)$$

$$DataSize_{total}^{PReq/PRes} = DataSize_{total}^{format} + DataSize_{total}^{user\_data} \quad (4.11)$$

$$DataSize_{total}^{format} = 20(DataSize_{format}^{PReq} + DataSize_{format}^{PRes}) \quad (4.12)$$

$$DataSize_{total}^{user\_data} = \sum_{i=1}^{13}(DataSize_{InputSignal}^i + DataSize_{outputSignal}^i) \quad (4.13)$$

公式 4.10 中的  $T_{net}^{line}$  为储存环设备保护系统的网络组件延时, 各联锁从站按照线型拓扑结构相连,  $T_{net}^{line}$  可按照公式 4.14 进行计算, 其中  $T_c$  为相邻两节点之

间网线的传输延时，储存环周长为 480m，这里假定各联锁从站按照等距离部署，相邻从站之间的距离为 24m，按照 5ns/m 的传输延时计算， $T_c$  为  $0.12\mu s$ 。 $T_{hub}$ 、 $T_{r-cn}$ 、 $T_{r-mn}$  参数根据第 3.3 节中系统的实测结果，分别为  $0.66\mu s$ 、 $1.048\mu s$  和  $1.865\mu s$ ，代入公式 4.14，可得  $T_{net}^{line}$  为  $372.66\mu s$ 。

$$T_{net}^{line} = \sum_{i=1}^{20} [2iT_c + (2i - 1)T_{hub} + T_{r-cn}] + nT_{r-mn} \quad (4.14)$$

将  $T_{net}^{line}$  和  $\sum_{i=1}^{20} (T_{F(PReq)}^i + T_{F(PRes)}^i)$  的值代入公式 4.10，可得轮询阶段时长  $T_{poll}$  为  $409.892\mu s$ ，加上同步阶段的时长  $T_{sync}$ ，则等时同步阶段的时长  $T_{ip}$  为  $410.892\mu s$ 。

考虑到空闲阶段时长为定值，为传输最大异步数据预留的时间，根据第 3.3 节中系统的实测结果为  $12.9\mu s$ ，可得储存环设备保护系统通信周期为  $423.792\mu s$ 。

根据第 3.2.5 节中对测试系统响应时间的分析，耗时最长的响应过程为联锁输入信号从 1 号从站输入，经过主站处理后，通过 20 号从站输出保护动作，此过程跨越 3 个周期，考虑到输入信号的通信等待时间和从站数字隔离器  $5\mu s$  的信号处理时间，储存环设备保护系统的最长响应时间约为  $1.7ms$ 。

## 2. 网络模拟

基于 OMNeT++ 仿真平台，我们对储存环设备保护系统进行了建模，模型如图 4.21 所示。

根据第 3.3 中对通信周期的理论分析，各联锁从站被分配的通信时隙 ( $T_{cn-slot}^i$ ) 可通过公式 4.16 和公式 4.17 计算得到，其中各从站的 PReq 数据帧和 PRes 数据帧的传输时间  $T_{F(PReq)}$  和  $T_{F(PRes)}$  跟该从站的输入和输出信号数量有关，各从站较为均匀的部署在储存环范围内，这里我们将储存环联锁信号总数量按照各从站进行均分计算，根据表 4.2 的统计结果，储存环设备保护系统中联锁信号的总数量为 1907，考虑到输出信号的回读状态信号、运行模式切换信号等相关信号，这里同样将联锁信号数量翻倍，即按照 3814 个总联锁信号数量来进行计算，则每个从站的联锁信号数量为 191。根据公式 4.18，PReq/PRes 帧的帧头帧尾 ( $DataSize_{PReq/PRes}^i$ ) 均为 21Byte，可得每个从站的 PReq/PRes 帧的数据量 ( $DataSize_{PReq/PRes}^i$ ) 为 233Byte。 $T_{hub}$ 、 $T_{r-cn}$ 、 $T_{r-mn}$  参数按照第 3.3 节中系统的实测结果，分别为  $0.66\mu s$ 、 $1.048\mu s$  和  $1.865\mu s$ ，可以计算出各联锁从站被配置的通信时隙 (PResTimeout)。各从站具体配置如下所示：

$$DataSize_{PReq/PRes}^i = DataSize_{format}^i + DataSize_{user\_data}^i \quad (4.15)$$

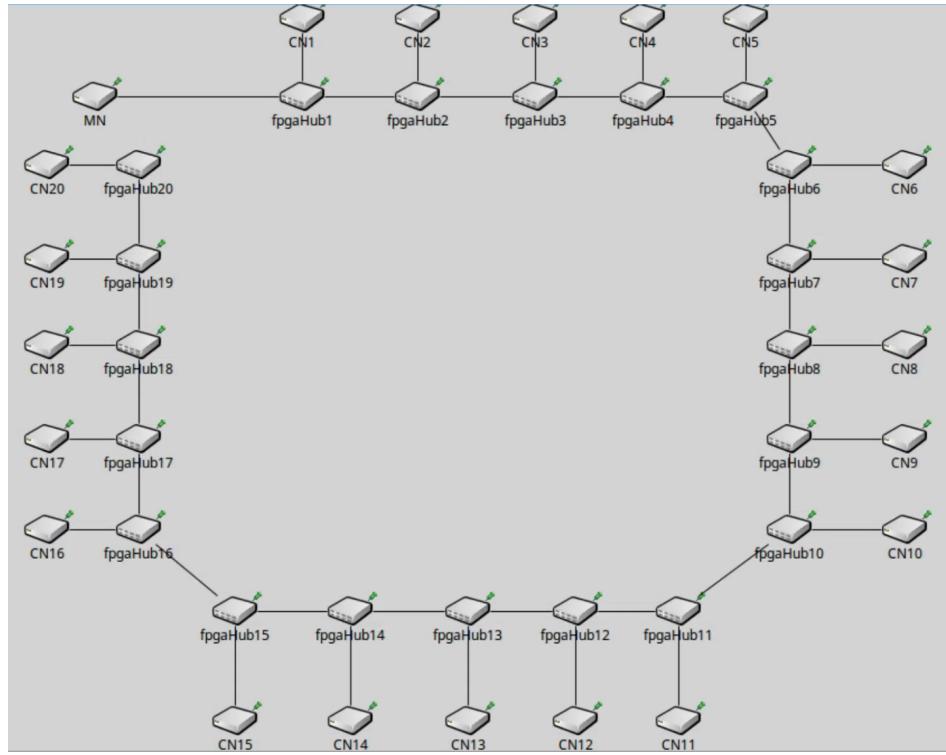


图 4.21 储存环设备保护系统模型

$$T_{rtd}^i = 2iT_c + (2i - 1)T_{hub} + T_{r-mn} \quad (4.16)$$

$$T_{cn-slot}^i = T_{F(PReq)}^i + T_{F(PRes)}^i + T_{rtd}^i + T_{r-mn} \quad (4.17)$$

$$DataSize_{PReq/PRes}^i = DataSize_{format}^i + DataSize_{user\_data}^i \quad (4.18)$$

各从站具体配置如下所示，其中 PResTimeout 为各联锁从站的通信时隙，PReqActPayload 为各从站的输出信号大小。

```

1 <Nodes>
2   <Node NodeId="1" PResTimeout="5677ns" PReqActPayload="5byte" />
3   <Node NodeId="2" PResTimeout="7237ns" PReqActPayload="6byte" />
4   <Node NodeId="3" PResTimeout="8797ns" PReqActPayload="6byte" />
5   <Node NodeId="4" PResTimeout="10357ns" PReqActPayload="6byte" />
6   <Node NodeId="5" PResTimeout="11917ns" PReqActPayload="6byte" />
7   <Node NodeId="6" PResTimeout="13477ns" PReqActPayload="6byte" />
8   <Node NodeId="7" PResTimeout="15037ns" PReqActPayload="6byte" />
9   <Node NodeId="8" PResTimeout="16597ns" PReqActPayload="6byte" />
10  <Node NodeId="9" PResTimeout="18157ns" PReqActPayload="6byte" />
11  <Node NodeId="10" PResTimeout="19717ns" PReqActPayload="6byte" />
12  <Node NodeId="11" PResTimeout="21277ns" PReqActPayload="6byte" />
13  <Node NodeId="12" PResTimeout="22837ns" PReqActPayload="1byte" />
```

```

14 <Node NodeId="13" PResTimeout="24397ns" PReqActPayload="1byte" />
15 <Node NodeId="14" PResTimeout="25957ns" PReqActPayload="6byte" />
16 <Node NodeId="15" PResTimeout="27517ns" PReqActPayload="6byte" />
17 <Node NodeId="16" PResTimeout="29077ns" PReqActPayload="6byte" />
18 <Node NodeId="17" PResTimeout="30637ns" PReqActPayload="6byte" />
19 <Node NodeId="18" PResTimeout="32197ns" PReqActPayload="6byte" />
20 <Node NodeId="19" PResTimeout="33757ns" PReqActPayload="6byte" />
21 <Node NodeId="20" PResTimeout="35317ns" PReqActPayload="1byte" />
22 </Nodes>

```

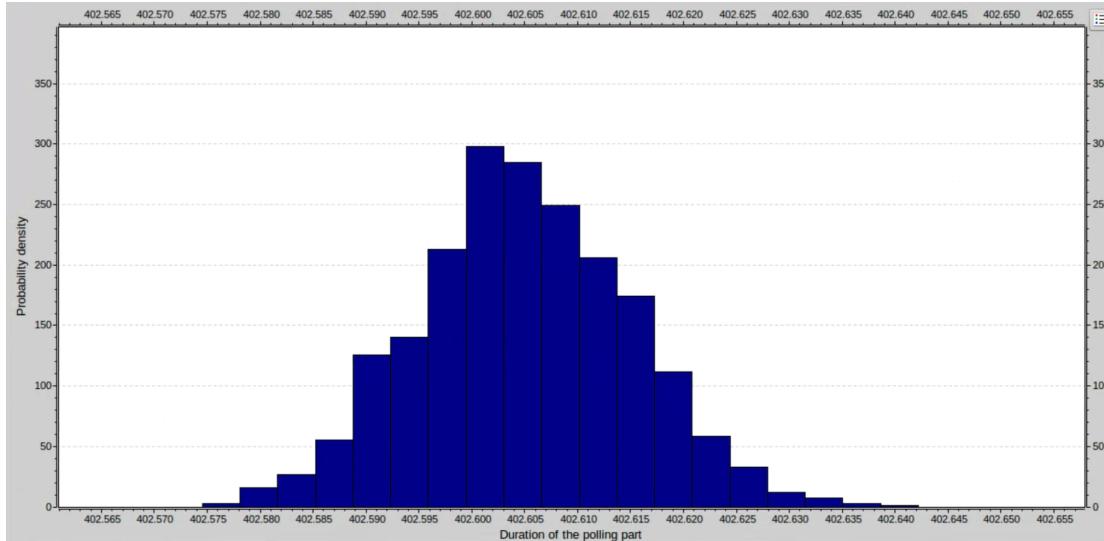


图 4.22 储存环设备保护系统模型轮询阶段时长的概率密度分布

模拟过程持续了大约 2000 个 POWERLINK 通信周期。等时同步阶段的概率密度分布绘制在图 4.22 中，该图表明储存环设备保护系统轮询平均时长为  $402.6024\mu s$ ，与通过理论计算得到的轮询阶段时长  $409.892\mu s$  相比，网络模拟得到的结果略低，原因是大多数从站在分配的通信时隙内可提前完成相应的通信。考虑到同步阶段时长  $T_{sync}$  最大为  $1\mu s$  和  $12.9\mu s$  的空闲时长，可得储存环设备保护系统通信周期为  $416.5024\mu s$ 。考虑到输入信号的通信等待时间和从站数字隔离器  $5\mu s$  的信号处理时间，储存环设备保护系统的最长响应时间为  $1.671ms$ 。

### 3. 小结

我们通过理论计算和仿真模拟的方法两种方法对储存环设备保护系统的实时性能进行了估算，两种方法的计算结果比较相近，通信周期的估算结果分别为  $423.792\mu s$  和  $416.5024\mu s$ ，最大响应时间的估算结果分别为  $1.7ms$  和  $1.7671ms$ ，通过理论计算和仿真模拟的方法两种方法得到的结果较接近，均完全满足设备保护系统响应时间  $10ms$  响应时间的需求。

## 4.6 HALF设备保护系统的信息报警

系统运行过程中出现异常情况时，及时将报警信息发布出来，让工作人员第一时间进行故障维修或隐患排除，对提高系统的可用性是非常重要的。HALF EPS 是在 EPICS 架构下设计的，如图 4.7 所示。EPICS 社区先后发布过多款报警软件，如 ALH、BEAST 和 Phoebus/Alarms 等<sup>[66-68]</sup>。Phoebus/Alarms 是最新发布的报警软件，由美国橡树岭国家实验室（Oak Ridge National Laboratory，ORNL）的工作人员开发。国家同步辐射实验室在 HALF 预研工程中对 Phoebus/Alarms 报警软件进行了研究，并在其基础上根据实际需求进行了二次开发，增加了报警信息网页查询、微信和短信 3 种报警信息发布方式<sup>[69]</sup>。

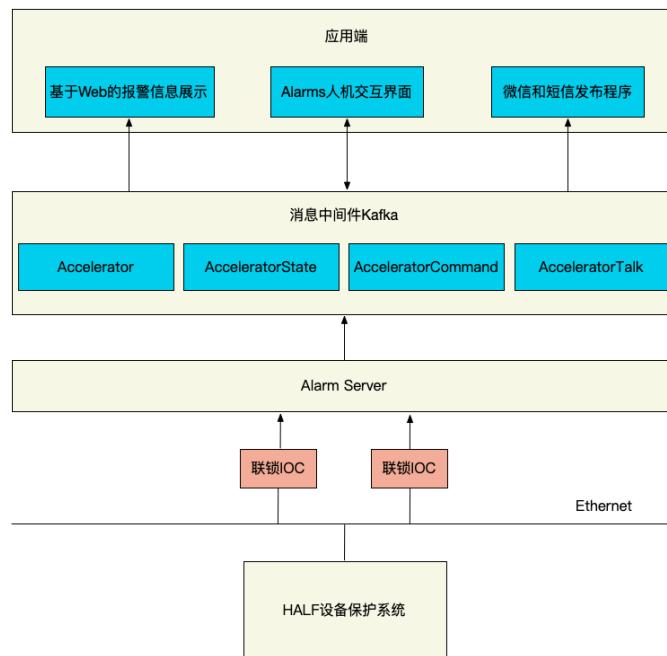


图 4.23 报警系统架构图

图 4.23 是 HALF EPS 报警系统的架构图，分为服务端、Kafka、上层应用端等 3 层。服务端是 Alarms 的组件 Alarm Server。Alarm Server 从 Kafka 中读取报警系统配置文件，然后通过 CA 协议从联锁 IOC 中获得相关联锁信号的报警信息，并将相报警信息发送 Kafka 分布式消息中间件。上层应用端主要由 3 类模块组成，分别是 Phoebus/Alarms 中的客户端组件、报警信息查询网页、微信和短信发布程序。

报警信息根据其严重程度进行分为 3 个等级：严重故障、一般故障和正常状态。严重故障会通过短信和微信及时发送给相关工作人员，在控制界面上以红色标记，可以启动自动保护程序或由工作人员决定处理措施；一般故障不发送短信和微信，仅在控制界面上以黄色标记，给予提醒；正常状态的信息用于描述设备的运行状态，不使用标记颜色。

## 4.7 HALF设备保护系统的历史数据存档与查询

HALF设备保护系统在运行的过程中，会产生大量与联锁事件相关的数据，包括设备故障数据以及相关的联锁保护数据等，这些存档数据可用于HALF装置故障诊断，对装置的运行维护具有重要意义。HALF装置计划采用Archiver Appliance作为历史数据存档工具，Archiver Appliance是EPICS社区最新发布的数据存档软件<sup>[70]</sup>，已应用在国内外多个加速器装置中，如SLAC、NSLS-II、ESS、HLS-II<sup>[71-73]</sup>等，HALF设备保护系统的联锁事件相关历史数据也会存入基于Archiver Appliance的HALF数据存档与查询系统中。

Archiver Appliance设计了三个数据存储级别，即根据数据产生时间将历史数据划分为短期存储、中期存储和长期存储，如表所示4.4。数据首先存入位于读写速度最快的内存中（短期存储），随着时间的推移，逐步迁移至位于读写速度稍慢的本地硬盘（SSD）中（中期存储），最后数据全部存入读写速度最慢的网络存储（SAN）中（长期存储）。一般来说，距离越近的数据检索越频繁，因此这样的设计可有效提高工作人员查询联锁事件相关历史数据的效率。

表4.4 Archiver Appliance存储级别

存储级别	数据产生时间	存储介质
短期存储	几个小时内	内存
中期存储	几天内	本地硬盘（SSD）
长期存储	长期	网络存储（SAN）

基于Archive Appliance，我们设计了HALF设备保护系统的历史数据存档系统，架构如图4.24所示。系统分为数据存档软件和数据可视化应用两个部分，Archive Appliance作为历史数据存档工具负责收集来自联锁IOC的数据，Archiver Appliance中数据采集引擎通过EPICS Channel Access与联锁IOC中的每个PV建立连接，监测PV值的变化，并将其写入短期存储中。Archiver Appliance内置的迁移模块完成了不同存储级别之间的数据自动迁移，数据检索模块提供了用于数据查询的HTTP/JSON接口。数据可视化应用采用基于web的网页图形界面为操作人员提供历史数据查询功能。

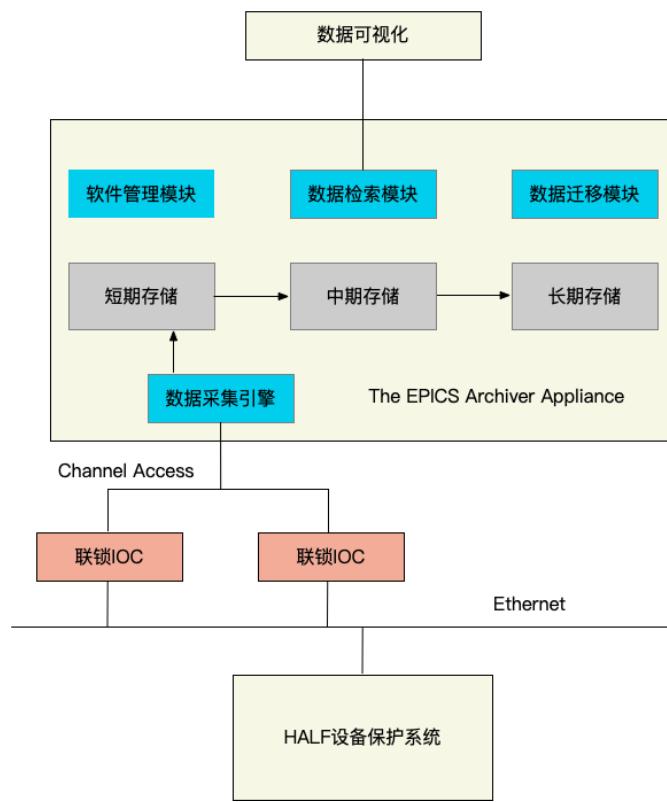


图 4.24 历史数据存档与查询系统图

## 第 5 章 总结与展望

### 5.1 总结

Ethernet POWERLINK 作为一种开源实时以太网技术广泛应用于工业控制领域, 特别是有高实时性需求的场合, 例如高性能的同步运动控制应用。在粒子加速器控制领域, EPICS 是目前国际上应用最广泛的控制系统。论文将 POWERLINK 以太网技术和 EPICS 环境结合起来, 开展了一系列的应用研究工作, 现将研究工作总结如下:

(1) 对 POWERLINK 通信协议进行了分析和性能测试。openPOWERLINK 是 POWERLINK 协议栈的开源实现版本, 我们基于 openPOWERLINK 搭建了 2 个由不同硬件组成的测试系统, 测试系统的硬件组成分别是 2 台 RT-Linux PC、1 台 RT-Linux PC 和 6 块 DE2-115 FPGA 开发板。在这 2 个测试平台上分别进行了 POWERLINK 通信实验, 采用网络分析仪 netANALYZER 抓取了 POWERLINK 数据帧, 并利用 Wireshark 软件对通信过程进行诊断分析, 掌握了 POWERLINK 协议的数据帧结构和通信机制。同时测试了不同硬件平台下 POWERLINK 协议的实时性能, 2 台 RT-Linux PC 的千兆 POWERLINK 通信周期最快为 1ms, 由 RT-Linux PC 和 DE2-115 FPGA 开发板组成的百兆 POWERLINK 系统通信周期最快为  $700\mu\text{s}$ , POWERLINK 协议在 FPGA 硬件平台下的实时性能明显优于其在 RT-Linux PC 平台下的性能。采用理论分析的方法, 推导出了线型拓扑结构下 POWERLINK 多节点系统的通信周期计算公式。网络模拟基于 OMNet++ 仿真器展开, 建立了支持 POWERLINK 通信的主从节点模型。

(3) 设计了 EPICS 环境下基于 POWERLINK 的分布式 IO 系统。我们提出了两套系统方案, 分别是主站 PC 方案和全站 FPGA 方案。主站 PC 方案的系统通信网络是千兆 POWERLINK, 各节点依次相连成菊花链拓扑结构, 从站采用基于 Zynq 的控制器, 负责信号采集和输出, 主站负责收集处理各从站 IO 数据并传输至 EPICS IOC。我们开发了基于 Zynq 芯片的从站控制器, Zynq 的 FPGA 部分实现了 POWERLINK 协议的物理层和数据链路层, 并且实现了 HUB 的功能, ARM 部分实现了 POWERLINK 协议的应用层。主站 PC 方案采用一台 RT-Linux PC 作为系统主站, PC 上运行了 IOC 应用程序和内核空间下的 openPOWERLINK 程序, 我们基于进程间 Socket 开发了 EPICS 设备驱动程序。为测试系统性能, 我们搭建了 10 个节点组成的测试系统。测试系统的通信周期最快可到  $275\mu\text{s}$ , 控制器本地响应时间约为  $400\mu\text{s}$ , 系统全局响应时间为  $870\mu\text{s}$ 。经研究发现, 本地响应时间主要是由从站输入/输出接口电路中的光耦合器延时; 同时作为主站的 RT-Linux PC 的响应时间较慢, 约为  $16.296\mu\text{s}$ 。

基于主站 PC 方案系统的测试结果，我们提出了全站 FPGA 方案。在全站 FPGA 方案中，从站控制器的输入/输出接口电路采用高速磁隔离器，主站采用 Zynq 控制器来实现，主站与运行在 PC 上的 IOC 应用程序通过 UDP Socket 的方式通信。为测试改进方案的系统性能，我们搭建了 5 个节点组成的系统，通信周期最快可到  $50\mu\text{s}$ ，控制器本地响应时间为  $5\mu\text{s}$ ，系统全局响应时间为  $160\mu\text{s}$ ，全站 FPGA 方案系统的实时性能明显优于主站 PC 方案系统。我们基于全站 FPGA 方案系统的测试结果，进一步完善了理论计算方法，然后通过对全站 FPGA 方案系统进行仿真模拟，模拟得到的系统通信周期  $51.1\mu\text{s}$ ，接近实际测试系统的  $50\mu\text{s}$  通信周期，验证了仿真建模分析方法的可行性，

(4) 目前国家同步辐射实验室正在进行 HALF 预研工程，我们基于 POWERLINK 设计了 HALF 设备保护系统 (EPS)。HALF EPS 由注入器分总体 EPS 和储存环分总体 EPS 组成，各分总体 EPS 基于独立的千兆 POWERLINK 网络设计，主站与从站均采用 Zynq 控制器，从站将设备联锁信号通过 POWERLINK 上传至主站，经过主站的联锁逻辑处理之后，主站通过 POWERLINK 网络将保护命令传输至相应的从站实施保护。注入器分总体 EPS 可由 1 台联锁主站和 13 台联锁从站组成，我们详细分析注入器 EPS 的联锁保护逻辑，并统计了注入器 EPS 的联锁信号数量。根据注入器分总体 EPS 的设计规模，采用理论分析和网络仿真两种方法对其实时性能进行了评估，得到最大响应时间的估算结果分别为  $829.996\mu\text{s}$  和  $819.0188\mu\text{s}$ ，均完全满足设备保护系统响应时间  $10\text{ms}$  响应时间的需求。储存环分总体 EPS 可由 1 台联锁主站和 20 台联锁从站组成，我们详细分析储存环 EPS 各系统的联锁保护逻辑，并统计了储存环分总体 EPS 的联锁信号数量。根据储存环分总体 EPS 的设计规模，采用理论分析和网络仿真两种方法对其实时性能进行了评估，得到最大响应时间的估算结果分别为  $1.7\text{ms}$  和  $1.671\text{ms}$ ，也完全满足  $10\text{ms}$  响应时间的需求。最后基于 Archive Appliance 设计了 HALF EPS 的历史数据存档与查询系统，基于 Phoebus/Alarms 设计了 HALF EPS 报警系统。

## 5.2 展望

本论文的研究内容还有一些有待完善和扩展的地方，具体如下：

(1) 除了实时性，还需要开展 POWERLINK 冗余技术的研究，增加 HALF 设备保护系统的可靠性。POWERLINK 技术提供了三种冗余实现方式：双网冗余、环型冗余和多主冗余。双网冗余是在系统中采用两个独立的网络，当一个网络出现故障，另一个网络依然可以工作，不会丢失数据帧。环型冗余是一种常用的冗余，将菊花链拓扑结构的最后一个节点再与主站相连接以构成了一个环。当其中

一根线缆出现问题，这个系统依然可以继续工作。多主冗余就是在系统中，存在多个主站，仅有主站一个处于活动状态，其他的主站处于备用状态，当活动主站出现故障时，备用主站接替其工作，继续维持网络的稳定运行。

(2) 在加速器控制领域，除了设备联锁保护系统，POWERLINK 技术还可应用于电子储存环轨道快反馈 (Fast Orbit Feedback, FOFB) 系统。FOFB 系统负责对储存环的束流闭轨畸变进行校正，快反馈校正频率要求达到 KHz 量级。在一个闭轨校正周期中，系统需要完成束流位置数据的传输、轨道校正的计算和校正铁电源电流值的设定。FOFB 系统由若干个遍布储存环的校正单元组成，我们可以采用 Zynq 控制器作为 FOFB 系统的校正单元控制器，控制器负责接收来自全环的束流位置数据，按照闭轨校正算法计算校正铁电流值，电流值通过 POWERLINK 传输至各校正铁电源控制器<sup>[74]</sup>。目前正在与电源组合作共同开展此项工作<sup>[75]</sup>。

## 参 考 文 献

- [1] CLAYTON J, SHEDLOCK D, VANDERET S, et al. Prototype 1.75 MV X-band linear accelerator testing for medical CT and industrial nondestructive testing applications[J/OL]. Health Monitoring of Structural and Biological Systems 2015, 2015, 9438:943813. DOI: 10.1117/12.2084394.
- [2] LI J, XU W, XUAN K, et al. OPERATION STATUS OF HLS-II[C]//Proceedings of IPAC2016: number 11375177. Busan, Korea, 2016: 4155-4158.
- [3] BAI Z H, LIU G W, LUO C W, et al. Initial Lattice Design for Hefei Advanced Light Source : a Vuv and Soft X-Ray Diffraction-Limited Storage Ring Hals Storage Ring Lattice[C]// Proceedings of IPAC2016. Busan, Korea, 2016: 2889-2891.
- [4] 赵籍九, 尹兆升. 粒子加速器技术[M]. 1 版. 北京: 高等教育出版社, 2006.
- [5] 刘松强. 计算机控制系统的原理与方法[M]. 北京: 科学出版社, 2006.
- [6] EPICS homepage[EB/OL]. <https://epics.anl.gov/>.
- [7] AGHABABYAN A, BACHER R, BARTKIEWICZ P, et al. The large scale european xfel control system: Overview and status of the commissioning[C]//2015: 5-8.
- [8] CARCASSI G, DALESIO L R, DOHAN D, et al. NSLS II Control System[C]//Proceedings of ICALEPCS2009. Kobe, Japan, 2009: 319-321.
- [9] WALLANDER A, ABADIE L, MAIO F D, et al. News from ITER Controls –a Status Report [C]//Proceedings of ICALEPCS2011. 2001: 1-4.
- [10] FLATH D. Recent and Future Upgrades to the Control Systems of LCLS and LCLS-II Scientific Instruments[R]. ICALEPCS2017, Barcelona, Spain, 2017.
- [11] SHEN G, BERRYMAN E, CHABOT D, et al. Status and Progress of FRIB High Level Controls[C]//Proceedings of LINAC2016. East Lansing, MI, USA, 2016: 885-887.
- [12] ARREDONDO I, Del Campo M, ECHEVARRIA P, et al. Multipurpose Controller with EPICS integration and data logging: BPM application for ESS Bilbao[J/OL]. Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2013, 726:127-138. <http://dx.doi.org/10.1016/j.nima.2013.05.152>.
- [13] CARWARDINE J, ARNOLD N, LENKSZUS F, et al. The ILC Global Control System[C]// Proceedings of PAC2007. Albuquerque, New Mexico, USA, 2007: 868-872.
- [14] 李为民, 刘功发, 李川, 等. NSRL 二期工程中加速器控制系统的改造[J]. 中国科学技术大学学报, 2007, 37(45):359-363.
- [15] ZHAO J. Experience of Developing BEPCII Control System[C]//Proceedings of ICALEPCS07. Knoxville, Tennessee, USA, 2007: 511-513.

- [16] SHEN L R, LIU D K, SHANGHAI S. The SSRF Control System[C]//Proceedings of IPAC 2010. Kyoto, Japan, 2010: 2704-2706.
- [17] C.H.WANG. Status of the CSNS Control System[C]//Proceedings of ICAL EPCS2011. Grenoble, France, 2011: 1341-1344.
- [18] LV H, LENG Y, YAN Y, et al. The high level application architecture of the control system for SHINE[J/OL]. Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2018, 908(May):167-171. <https://doi.org/10.1016/j.nima.2018.08.047>.
- [19] CHU P, QIAO Y S, WANG C H, et al. HEPS High-Level Software Architecture Plan [C/OL]//Proceedings of IPAC 2018. Vancouver, BC, Canada, 2018: 1884-1887. DOI: 10.18429/JACoW-IPAC2018-WEPAF030.
- [20] KUIPER B. Issues in Accelerator Controls[C]//Proceedings of ICAL EPCS1991. Tsukuba, Japan, 1991: 602-611.
- [21] 缪学勤. 实时以太网技术现状与发展[J]. 自动化博览, 2005, (02):21-24+26.
- [22] Ethernet POWERLINK Communication Profile Specification EPSG DS 301 V1.4.0 Ethernet POWERLINK Communication Profile Specification[M/OL]. 2018. [www.ethernet-powerlink.org](http://www.ethernet-powerlink.org).
- [23] GIACUZZO F, BATTISTELLO L, FRÖHLICH L, et al. EQUIPMENT AND MACHINE PROTECTION SYSTEMS FOR THE FERMI@Elettra FEL FACILITY[C]//Proceedings of ICAL EPCS2011. 2011: 1119-1121.
- [24] LIAO C Y, WU C Y, CHENG Y S, et al. Implementation of machine protection system for the taiwan photon source[C/OL]//Proceedings of IPAC2014. 2014: 3189-3191. DOI: 10.18429/JACoW-IPAC2014-THPRO126.
- [25] ESS Bilbao Interlock System Approach[C]//Proceedings of ICAL EPCS2013. JACoW, 2013: 205-209.
- [26] 刘鑫, 蒋舸扬, 沈立人, 等. 基于 FPGA 和 ARM 的控制器设计及在快联锁保护系统中的应用[J]. 核技术, 2010, (33):653-656.
- [27] UZUN I S, BARTOLINI R, REHM G, et al. INITIAL DESIGN OF THE FAST ORBIT FEEDBACK SYSTEM FOR DIAMOND LIGHT SOURCE[C]//Proceedings of ICAL EPCS2013. 2013.
- [28] SINGH O, BACHA B, BLEDNYKH A, et al. NSLS-II BPM and Fast Orbit Feedback System Design and Implementation[C]//Proceedings of ICAL EPCS2013. 2013: 316-322.
- [29] 徐海亚, 赵增基, 朱波, 等. 基于 POWERLINK 技术标准的交叉通信研究[J]. 控制工程, 2015, 22:115-120.
- [30] 朱晓枫, 甘屹, 陶益民. 基于 Ethernet Powerlink 多轴伺服运动控制系统的同步性能研究

- [J]. 现代制造工程, 2014, 11:64-68.
- [31] 史贵林, 刘克敏, 朱波, 等. POWERLINK 的火指控一体化综合交换设备设计[J]. 火力与指挥控制, 2012, 37:112-114.
- [32] 黄苏平, 施霞. Ethernet-Powerlink 通信技术在列车控制与监测系统中的应用[J]. 数据采集与处理, 2012, 27:411-413.
- [33] SENO L, VITTURI S. A Simulation Study of Ethernet Powerlink Networks[C]//Proceeding of ETFA. 2007: 740-743.
- [34] CENA G, SENO L, VALENZANO A, et al. Performance analysis of Ethernet Powerlink networks for distributed control and automation systems[J]. Computer Standards & Interfaces, 2009, 31:566-572.
- [35] FERNÁNDEZ-CARREIRAS D, BELTRAN D, KLORA J, et al. PERSONNEL PROTECTION, EQUIPMENT PROTECTION AND FAST INTERLOCK SYSTEMS: THREE DIFFERENT TECHNOLOGIES TO PROVIDE PROTECTION AT THREE DIFFERENT LEVELS[C]//Proceedings of ICALEPCS2011. 1055-1057.
- [36] DANILUK G, GOUSIOU E, SWITZERLAND G. Plans at CERN for Electronics and Communication in the Distributed I/O Tier[C/OL]//Proceedings of ICALEPCS2017. 2017: 1552-1556. DOI: 10.18429/JACoW-ICALEPCS2017-THPHA071.
- [37] 郝晓俊. 工业以太网实时通讯技术分析[J]. 现代工业经济和信息化, 2018, 12:90-91.
- [38] 魏亚鹏, 韩卫光. 实时工业以太网技术的研究[J]. 组合机床与自动化加工技术, 2013, 07: 49-53.
- [39] Ethernet Powerlink Standardization Group [Online][EB/OL]. <https://www.ethernet-powerlink.org/>.
- [40] openPOWERLINK homepage[EB/OL]. <http://openpowerlink.sourceforge.net/web/>.
- [41] FELD J. PROFINET - scalable factory communication for all applications[C/OL]//IEEE International Workshop on Factory Communication Systems, 2004. Lemgo/Detmold, Germany, 2005: 33-38. DOI: 10.1109/wfcs.2004.1377673.
- [42] 王磊, 李木国, 王静, 等. 基于 EtherCAT 协议现场级实时以太网控制系统研究[J/OL]. 2011, 32(7):2294-2297. DOI: 10.16208/j.issn1000-7024.2011.07.079.
- [43] 王凯. 面向工业实时以太网协议 Sercos III 的分析研究[D]. 机械科学研究院总院, 2017.
- [44] PAUL B. EthernetIP - Industrial Protocol[C]//8th International Conference on Emerging Technologies and Factory Automation: volume 2. Zaragoza, Spain, 2001: 505-514.
- [45] 金超, 夏继强, 满庆丰. CANopen 对象字典的分析与具体实现[J]. 单片机与嵌入式系统应用, 2012, 12:12-14.
- [46] openCONFIGURATOR[EB/OL]. <https://sourceforge.net/projects/openconf/>.
- [47] WOLFGANG WALLNER J B. openPOWERLINK in Linux Userspace: Implementation and

- Performance Evaluation of the Real-Time Ethernet Protocol Stack[C]//Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.
- [48] 肖维荣, 王谨秋. 开源实时以太网 POWERLINK 详解[M]. 机械工业出版社, 2015.
- [49] X. SUN G L, Y. Song. DISTRIBUTED IO SYSTEM BASED ON ETHERNET POWERLINK UNDER THE EPICS ARCHITECTURE[C]//Proceedings of IPAC2018. Vancouver, BC, Canada, 2018: 4917-4919.
- [50] KNEZIC M, DOKIC B, IVANOVIC Z. Theoretical and experimental evaluation of ethernet powerlink pollresponse chaining mechanism[J/OL]. IEEE Transactions on Industrial Informatics, 2017, 13:923-933. DOI: 10.1109/TII.2016.2634554.
- [51] OMNeT++ Simulation Manual[M/OL]. <https://doc.omnetpp.org/omnetpp/manual/>.
- [52] 高承志, 范贤学. 基于 OMNeT++ 的无线传感器网络通信协议仿真[J]. 自动化与信息工程, 2015, 36:29-34.
- [53] 许期长, 刘湃. 基于 OMNeT++ 的 6LoWPAN 协议设计与实现[J]. 数字技术与应用, 2014, (11):144-145.
- [54] 刘敏. 基于 OMNeT++4.0 的 IEEE802.11 无线网路仿真[J]. 电子制作, 2015, (09):10-11.
- [55] 吴剑锋, 郭英, 范海宁. OMNeT++ 网络仿真器的设计原理分析[J]. 微计算机应用, 2006, (05):550-552.
- [56] SUN X K, SONG Y F, XUAN K, et al. The prototype of equipment protection system based on Ethernet POWERLINK for HALS[J]. Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 2019, 946.
- [57] zynq-7000[EB/OL]. <https://china.xilinx.com/products/silicon-devices/soc/zynq-7000.html>.
- [58] NORUM S, ALLISON S, BROWNE M, et al. THE MACHINE PROTECTION SYSTEM FOR THE LINAC COHERENT LIGHT SOURCE[C]//Proceedings of PAC09. Vancouver, BC, Canada, 2009: 4856-4858.
- [59] JIN H, CHOI Y, LEE S. CURRENT STATUS OF THE RAON MACHINE PROTECTION SYSTEM DEVELOPMENT[C]//PCaPAC2018. 2018: 160-162.
- [60] 于春蕾, 陈建锋, 莹张, 等. 上海光源储存环机器保护系统升级[J]. 原子能科学技术, 2020, 54:572-576.
- [61] 何泳成, 李刚, 金大鹏, 等. CSNS 漂移管直线加速器水冷联锁系统设计[J]. 核电子学与探测技术, 2017, 37:585-589.
- [62] 于春蕾, 欢赵, 丁建国. 上海软 X 射线自由电子激光装置联锁保护系统[J]. 原子能科学技术, 2018, 52:757-761.
- [63] Machine protection system for J-parc[C]//Accelerator Technical Advisory Committee. 2004.
- [64] 仇文君, 周大勇, 陈建锋, 等. 基于 PLC 的直线加速器联锁保护系统[J]. 核技术, 2008, 07: 506-510.

## 参 考 文 献

---

- [65] 孙晓康, 宋一凡, 李川, 等. HLS- 恒温冷却水控制系统的设计与实现[J]. 原子能科学技术, 2018, 52:1139-1143.
- [66] Alarm Handler User's Guide[M/OL]. <http://www.aps.anl.gov/epics/EpicsDocumentation/ExtensionsManuals/AlarmHandler-alhUserGuide-1.2>.
- [67] K K, X C, E D. The best ever alarm system toolkit[C]//Proceedings of ICALEPCS2009. 2009: 46-48.
- [68] ROSATI C, SHROF K, KASEMIR K. JavaFX AND CS-STUDIO: BENEFITS AND DISADVANTAGES IN DEVELOPING THE NEXT GENERATION OF CONTROL SYSTEM SOFTWARE[C]//Proceedings of ICALEPCS2017. 2017: 770-774.
- [69] X C, Y S, SHAO Z E A. ALARM SYSTEM OF IRFEL AT NSRL[C]//Proceedings of IPAC2019. 2019: 3896-3899.
- [70] Archiver Appliance homepage[EB/OL]. <https://github.com/slacmshankar/epicsarchiverap>.
- [71] M S, L L, M P, et al. The EPICS Archiver Appliance[C]//Proceedings of ICALEPCS2015. Melbourne, 2015: 761-764.
- [72] T K, S G, L F, et al. Status of the European spallation source control system[C]//Proceedings of ICALEPCS2015. Melbourne, 2015: 177-181.
- [73] YIFAN S, GONGFA L. Automatic data archiving and visualization at HLS-II[J]. Nuclear Science and Techniques, 2018, 29:129-135.
- [74] TIAN Y, HA K, YU L, et al. NSLS-II FAST ORBIT FEEDBACK SYSTEM [C]//Proceedings of ICALEPCS2015. 2015: 34-37.
- [75] SHAO Z, GAO H, ZHANG H, et al. Research and design of digital magnet power supply controller for hals[C/OL]//Proceedings of IPAC2019: volume 1350. 2019. DOI: 10.1088/1742-6596/1350/1/012174.

## 补充材料

补充内容。

**致 谢**

## 在读期间发表的学术论文与取得的研究成果

### 已发表论文

1. A A A A A A A A A
2. A A A A A A A A A
3. A A A A A A A A A

### 待发表论文

1. A A A A A A A A A
2. A A A A A A A A A
3. A A A A A A A A A

### 研究报告

1. A A A A A A A A A
2. A A A A A A A A A
3. A A A A A A A A A