电子科技大学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

硕士学位论文

MASTER THESIS

论文题目 InfiniBand 接口通信控制器的研究与 FPGA 实现

学科专业 通信与信息系统

学　　号 201421010421

作者姓名 岳 杰

指导教师 杨海芬　副教授

分类号 _____ 密级 _____

UDC [注1] _____

学 位 论 文

InfiniBand 接口通信控制器的研究与 FPGA 实现

（题名和副题名）

岳 杰

（作者姓名）

指导教师 杨海芬 副教授

电子科技大学 成 都

（姓名、职称、单位名称）

申请学位级别 硕士 专业学位类别 通信与信息系统

提交论文日期 2017.3.29 论文答辩日期 2017.5.19

学位授予单位和日期 电子科技大学 2017 年 6 月

答辩委员会主席 _____

评阅人 _____

注 1：注明《国际十进分类法 UDC》的类号。

Research and FPGA Implementation of InfiniBand Interface Communication Controller

A Master Thesis Submitted to

University of Electronic Science and Technology of China

Major: **Communication and Information System**

Author: **Jie Yue**

Supervisor: **Associate Prof. Haifen Yang**

School: **School of Communication & Information Engineering**

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名：_____ 日期：　　年　月　日

论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将学位论文的全部或部分内容编入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。

（保密的学位论文在解密后应遵守此规定）

作者签名：_____ 导师签名：_____

日期：　　年　月　日

摘　要

随着计算能力逐步向数据中心的集中，数据传输的高效性和实时性变得比以往更加重要。InfiniBand 作为统一的网络互联结构，可以提供超高带宽和超低延迟的数据传输能力，广泛应用于集群服务器、磁盘阵列和企业数据中心。在下一代无线通信系统中，采用 InfiniBand 作为集中式基站互联技术，可有效改善传统互联的带宽低和实时性较差等问题。针对此背景，本文给出了一种用于集中式基站数据传输的 InfiniBand 接口解决方案，接口通过 FPGA+主机通道适配器（Host Channel Adapter，HCA）实现。

首先，本文在对 InfiniBand 协议进行了深入研究的基础上，对其拓扑结构、层次结构、控制机制和通信机制进行了分析，设计了 FPGA+HCA 的接口实现方案。接口中采用 Xilinx Virtex7 系列 FPGA 和型号为 Mellanox ConnectX-3 Pro 的 HCA。本文同时对接口中 FPGA 的功能进行了划分，作为可编程逻辑器件，FPGA 功能可根据实际需求灵活调整。

其次，本文在分析了通信管理协议的基础之上，给出了 InfiniBand 接口中通信控制器 FPGA 实现的总体方案。通信控制器由接入层管理模块、通信管理模块、GMP（General Management Packet，通用管理报文）封装与解析模块、GMP 发送与接收模块和 QP Verb 命令模块组成。通信控制器不仅能满足 InfiniBand 协议中通信链路建立与拆除的功能需求，而且具备差错控制功能。本文设计的报文封装与解析模块能支持多种不同格式管理数据包的封装与解析。本文还针对多个普通“队列对”和特殊“队列对”同时有共用资源使用请求这一现象制定了仲裁策略。

最后，本文选择 Vivado2015.2 软件作为开发工具和仿真工具，给出了功能仿真测试方案和结果。并在 Xilinx 的 VC707 开发板上对 InfiniBand 接口通信控制器进行了实现。FPGA 验证结果表明，系统能正常控制 InfiniBand 链路的建立和拆除，各个模块工作正常，满足设计要求。时序报告表明通信控制器在传输数据位宽为 256 bit 时，运行速率超过 250 MHz，吞吐量满足通道适配器 40 Gb/s 需求。

关键词：InfiniBand，通道适配器，通信控制器，FPGA

ABSTRACT

As computing power is gradually concentrated to the data centers, the efficiency and real-time nature of data transmission become more important than ever. InfiniBand as a unified interconnect structure, can provide ultra-high bandwidth and ultra-low latency data transmission capabilities, widely used in cluster servers, disk arrays and enterprise data centers. In the next generation of wireless communication systems, using the InfiniBand as the interconnect technology of centralized base station can effectively improve some problems which caused by the tranditional interconnect technology, such as the low bandwidth and poor real-time nature. Based on this background, this thesis presents a solution of InfiniBand interface for data transmission in the centralized base station, which is implemented by FPGA+HCA.

First of all, based on the deep research of InfiniBand protocol, this thesis analyzes the topology, the hierarchical structure, the control mechanism and the communication mechanism of InfiniBand, the interface scheme of FPGA + HCA is designed. Interface using Xilinx Virtex7 series FPGA and Mellanox ConnectX-3 Pro HCA. At the same time, the function of FPGA in the interface is divided. As a programmable logic device, the FPGA function can be adjusted flexibly according to the actual demand.

Secondly, based on the analysis of communication management protocol, this thesis gives the overall scheme of FPGA implementation of communication controller in IB interface. The communication controller is composed of an access layer management module, a communication management module, a GMP encapsulation and parse module, a GMP sending and receiving module and a QP Verb command module. The communication controller not only has the basic function of establishing and dismantling the communication link in the InfiniBand protocol, but also has the function of error control. In this thesis, the packet encapsulation and parse module can support the encapsulation and analysis of many different formats of management packets. This thesis also propose an arbitration strategy for the common "queue pair" and the special "queue pair", which have shared resource using requests.

Finally, this thesis chooses the Vivado2015.2 software as the development tool and the simulation tool, and gives the function simulation test scheme and results. The IB interface controller is implemented on the Xilinx VC707 development board. The FPGA

verification results show that the system can control the establishment and removal of the InfiniBand link, and each module works well to meet the design requirements. The timing report indicates that the communication controller is capable of running at more than 250MHz when the transfer data bit is 256. The throughput meets the HCA 40 Gb/s requirements.

Keywords：infiniband，channel adapter，communication control，FPGA

目 录

第一章 绪 论 1
1.1 论文研究背景及意义 1
1.2 国内外发展及研究现状 2
1.3 论文的组织章节 4
第二章 InfiniBand 协议分析 6
2.1 InfiniBand 体系结构 6
2.1.1 InfiniBand 系统拓扑结构 6
2.1.2 InfiniBand 协议层次结构 7
2.2 通道适配器控制机制 9
2.2.1 InfiniBand Verb 机制 9
2.2.2 通道适配器命令控制机制 10
2.3 InfiniBand 通信机制 12
2.3.1 InfiniBand 数据包格式 12
2.3.2 基于队列对的数据传输模型 13
2.3.3 通用管理报文传输过程 15
2.4 本章小结 16
第三章 IB 接口通信控制器分析与设计 17
3.1 IB 接口规划 17
3.1.1 IB 接口分析 17
3.1.2 IB 接口设计 18
3.2 通信管理协议 20
3.2.1 通信管理报文格式 20
3.2.2 通信管理实体 22
3.2.3 通信链路的建立过程 23
3.2.4 通信链路的终止过程 25
3.2.5 超时重传机制分析 26
3.3 IB 接口通信控制器总体设计 28
3.4 本章小结 30
第四章 通信控制器的 FPGA 实现 31
4.1 通信管理模块的实现 31

4.1.1 服务仲裁模块 32
4.1.2 CM Client 模块 32
4.1.3 CM Server 模块 35
4.1.4 超时与计数模块 37
4.2 GMP 封装与解析模块的实现 39
4.2.1 GMP 封装模块 39
4.2.2 GMP 解析模块 41
4.3 数据发送与接收模块的实现 44
4.3.1 数据发送模块 44
4.3.2 数据接收模块 50
4.4 QP Verb 命令模块的实现 50
4.5 本章小结 52
第五章 通信控制器仿真与验证 53
5.1 验证环境介绍 53
5.2 通信控制器功能仿真 54
5.2.1 功能仿真测试方案 54
5.2.2 功能仿真结果 55
5.3 通信控制器 FPGA 验证 58
5.3.1 时序报告和资源消耗 58
5.3.2 FPGA 验证方案 59
5.3.3 FPGA 验证结果分析 60
5.4 本章小结 62
第六章 总结与展望 63
6.1 工作总结 63
6.2 研究展望 63
致 谢 65
参考文献 66
个人简历及攻读硕士期间的研究成果 69

图目录

图 1-1 PCI 和 IB 结构对比 1
图 1-2 超级计算机 TOP500 互联协议使用数量对比 3
图 2-1 InfiniBand 系统拓扑结构 6
图 2-2 InfiniBand 协议层次 7
图 2-3 HCR 寄存器格式 10
图 2-4 Verb 命令执行流程图 11
图 2-5 InfiniBand 数据包格式 12
图 2-6 数据传输示意图 13
图 2-7 QP 状态转换及对应 Verb 命令 14
图 2-8 WQE 格式 14
图 3-1 IB 接口应用场景示意图 17
图 3-2 IB 接口 FPGA 与 HCA 对应协议层次 18
图 3-3 InfiniBand 接口总框图 19
图 3-4 节点通信管理实体 23
图 3-5 链路建立数据包交互和 QP 状态转换过程 24
图 3-6 链路拆除数据包交互和 QP 状态转换过程 25
图 3-7 链路建立数据传输定时模式示意图 27
图 3-8 IB 接口通信控制器设计框图 29
图 4-1 通信管理模块结构框图 31
图 4-2 CM Client 工作状态转移图 33
图 4-3 CM Server 工作状态转移图 36
图 4-4 系统定时模块结构图 38
图 4-5 GMP 封装模块结构框图 39
图 4-6 GMP 组成部分与数据长度 40
图 4-7 REQ 封装模块结构框图 41
图 4-8 REQ 八个 256bit 数据组成部分 41
图 4-9 GMP 解析模块结构框图 42
图 4-10 GMP 解析流程图 43
图 4-11 GMP 发送模块结构框图 44
图 4-12 QP_station 状态转换图 45

图 4-13 仲裁调度示意图 46
图 4-14 普通 QP 调度模型 47
图 4-15 GMP 发送 WQE 格式 47
图 4-16 发送 WQE Data 部分格式 48
图 4-17 UAR 寄存器格式 48
图 4-18 CQ Processor 模块结构框图 49
图 4-19 CQE 数据格式 49
图 4-20 QP Verb 命令模块结构框图 50
图 4-21 命令控制模块状态图 51
图 5-1 VC707 结构框图 53
图 5-2 功能测试平台框图 54
图 5-3 通信链路建立测试结果 56
图 5-4 超时重传测试结果 56
图 5-5 REJ 包处理测试结果 57
图 5-6 链路拆除仿真结果 57
图 5-7 仲裁测试结果 58
图 5-8 综合实现后时序报告 59
图 5-9 资源消耗报告 59
图 5-10 IB 接口通信控制器 FPGA 验证方案 59
图 5-11 FPGA 测试 Client 方结果 60
图 5-12 FPGA 测试 Server 方结果 60
图 5-13 链路信息存储和删除测试数据 61
图 5-14 FPGA 测试结果部分关键数据 61

表目录

表 2-1 部分 Verb 命令类集与释义 9
表 2-2 HCR 寄存器关键字段释义 11
表 3-1 GMP 包格式 20
表 3-2 GMP 首部字段含义 21
表 3-3 通信管理 GMP 包属性 22
表 3-4 REQ 数据部分关键信息 22
表 5-1 部分功能仿真测试用例 55
表 5-2 双方节点部分配置信息 55
表 5-3 REQ 部分关键信息 62

缩略词表

英文缩写	英文全称	中文释义
BBU	Building Base band Unit	室内基带处理单元
BTH	Base Transport Header	基本传输包头
CM	Communication Management	通信管理
CQ	Competition Queue	完成队列
FPGA	Field Programmable Gate Array	现场可编程门阵列
HDR	High Data Rate	高倍数据速率
GMP	General Management Packet	通用管理报文
GSI	General Service Interface	通用服务接口
HCA	Host Channel Adapter	主机通道适配器
HCR	HCA Command Register	通道适配器命令寄存器
LID	Local Identification	本地身份信息
LRH	Local Route Header	本地路由包头
MAD	Management Datagram	管理报文
MTU	Max Transmission Unit	最大传输单元
QoS	Quality of Service	传输服务质量
QP	Queue Pair	队列对
QPN	Queue Pair Number	队列对序列号
REQ	Request	链路建立请求数据包
RRU	Remote Radio Unit	远端射频单元
RTL	Register Transfer Level	寄存器转换级电路
RTR	Ready to Receive	准备好接收
RTS	Ready to Send	准备好发送
RTU	Ready To Use	准备好使用
SCSI	Small Computer System Interface	小型计算机系统接口
WQE	Work Queue Entry	工作队列单元

第一章 绪 论

1.1 论文研究背景及意义

随着信息化，数字化社会的飞速发展，人们对于数据传输带宽的需求与日俱长。现有的互联技术已经不能跟上计算机高速发展的步伐，并愈发成为了数据服务、应用进程以及企业计算的瓶颈[1-4]。一些高端计算新兴概念，比如集群、超算、全时服务等，需要在服务器节点之间，服务器和 I/O（Input/Output，输入输出）设备间进行超高带宽和低延迟的数据传输，以便于将更多的功能交由 I/O 设备完成[5]。为了满足需求，The InfiniBand Trade Association（IBTA）于 1999 年提出了 InfiniBand 这一新的互联技术。相比较于 PCI（Peripheral Component Interconnect，外设部件互联标准），以太网等传统互联技术，InfiniBand（IB）可以提供更高的带宽、更低的延迟以及更高的安全保护[1]。

PCI 总线作为一种高性能局部总线，广泛应用于工作站以及个人电脑中，但作为一种共享式总线，PCI 工作效率渐渐不能满足当前传输需求[6]。如图 1-1 所示，由于采用了树型拓扑结构，PCI 对网络化支持较差。虽然现在的 PCIe 总线属于高速串行点对点双向传输，总线上每个设备分配有独享通道带宽，但是在网络化支持上，仍需要大量的虚拟化开发工作，增大了研发成本[7]。

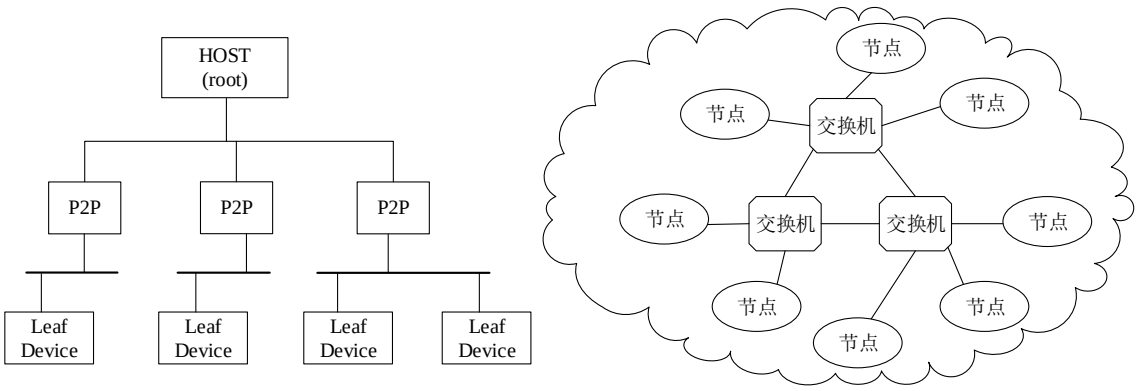


图 1-1 PCI 和 IB 结构对比

InfiniBand 作为一种网络通信协议，可以为网络各节点之间提供基于交换机的点对点双向串行链路支持。InfiniBand 采取以应用程序为中心的消息传递方法，通过通信节点间阻力最小的路径来传输数据[8]。应用最新的高倍数据速率（High Data Rate，HDR）技术，其传输速率高达 200 Gbps。同时支持远程直接访问机制，有效

的减少了操作系统开销，数据可以以更低的延迟通过网络。基于以上这些优点，IB在超算、集群、企业数据中心、数据分析以及存储等领域得到了广泛应用[9]。

目前，同样以高速率、大容量、低延迟等为其标签的下一代无线通信系统—5G，正处于火热的研发之中[10]。在 5G 背景下，移动数据流量的暴涨将给网络带来严峻的挑战。目前，蜂窝网络采用各基站独立的分布式架构，每一个基站由基带处理单元（Building Base band Unit，BBU）和远端射频单元（Remote Radio Unit，RRU）组成。BBU 负责基带数据处理，RRU 负责接收和发送射频数据以及数字上下变频操作，RRU 和 BBU 之间基于通用公共无线电接口（CPRI）进行数据传输[11]。但是由于通信网络负载的潮汐效应现象和基站数目的增加，带来了资源利用率差，能源消耗高的负面效应。针对这一现象，研究人员提出了集中式蜂窝网络架构[12]。集中式基站结构中，所有基站的 RRU 单元留在原站址上，BBU 单元则集中成一个数据处理中心。各基站接收数据并进行下变频处理，然后通过高性能互联网络将数据传至数据处理中心。采取何种互联网络将直接决定架构的工作效率和成本。

在集中式蜂窝网络架构中，带宽和时延是重要的性能指标，在采用 4*4 MIMO 和 3 扇区容量配置下，需要 40G 的传输网络才能保证数据传输的带宽要求。同时，网络传输引入的时延将影响接入性能和信号覆盖。目前 IP 网络或 SDH 网络难以完美满足需求。而 InfiniBand 作为一种高带宽低延迟的互联网络，其特性完美的契合了集中式基站中对于高性能网络互联结构的需求。使用 InfiniBand 作为新一代移动通信集中式基站互联技术，具有以下几个优点：（1）IB 最高可以提供 200 Gbps 的传输速率，可有效应对 5G 带来的大数据流量的冲击，满足通信传输带宽要求。（2）其纳秒级的应用延迟可有效提高数据集中化的实时性与效率，进而增强无线网络接入性能和信号覆盖。（3）组网自由，可根据应用场景大小灵活增加或减少传输节点。综上所述，InfiniBand 适合作为集中式基站方案互联网络。同时 InfiniBand 是针对硬件实现而设计的架构，通过 FPGA 设计实现高速传输接口，符合协议全硬件实现的初衷。利用 FPGA 高速和可编程特性[13]，其功能可定制，集中式基站处无需高性能服务器，即可实现数据通过 IB 网络进行传输。

1.2 国内外发展及研究现状

为解决共享式总线结构在海量数据传输应用中的瓶颈问题，由戴尔、Compaq、英特尔、微软、惠普、IBM 和 Sun 七家公司发起，共同研究新一代 I/O 标准。最初将研发成果命名为 System I/O，自 1999 年 10 月起，正式改名为 InfiniBand[2]。InfiniBand 作为一种长缆线的连接方式，具有高速、低延迟的高效传输特性。至今，该技术一直由 IBTA 进行更新，维护和发展。但是，在 2002 年，由于英特尔、微

软相继退出 IBTA，加上当时各国经济的衰退和 IT 预算紧缩，该技术进入冰河期，技术发展和应用都迟滞不前。

而随着网络技术的高速发展，集群计算、存储区域网、内部处理器通信等高端领域对带宽、扩展性、QoS 以及 RAS 等指标有更高的需求。各大公司纷纷开始将目光重新转向 InfiniBand。在 2004 年，IBM 开始将 InfiniBand 应用于其集群技术解决方案[14]。随后，惠普的 InfiniBand 产品在市场上出现，包含交换机和通道适配器。2016 年，世界卫生组织选择了 InfiniBand 作为其网络解决方案，以改善患者护理，帮助研究人员优化治疗。2004 年 11 月，美国国家航空航天局，完成了基于 InfiniBand 互联技术的超级计算机，运算速度当时位居世界第二。自此以后，超级计算领域开始大面积的将 InfiniBand 作为互联解决方案。如图 1-2 所示，截止到目前，全球排名前 500 的超级计算机中有 42%使用了 InfiniBand 作为其互联协议[15]，在 TOP100 中更是占到了 50%。

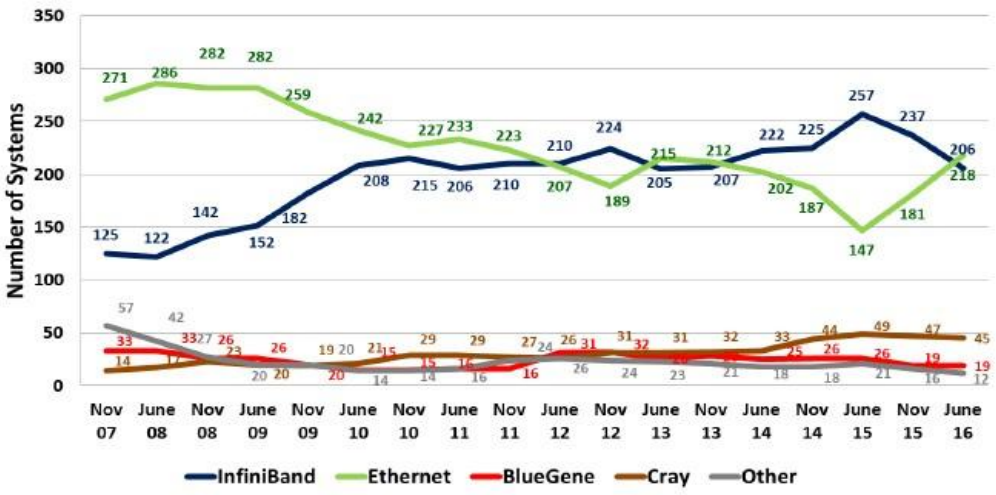


图 1-2 超级计算机 TOP500 互联协议使用数量对比[15]

在 InfiniBand 产品和解决方案研发方面，迈络思公司作为该领域的行业领导者和技术核心推动者，为客户提供完善的 IB 解决方案以及覆盖 IB 架构中从线缆到路由器、交换机和通道适配器的全线产品。在 2016 年超算会议 SC16 上，迈络思公司展示了基于 200 Gb/s InfiniBand 网络的最新 Quantum QM8790 交换机和 ConnectX-6 的通道适配器。该交换机具备 40 个端口，每个端口可提供 200 Gb/s 的传输能力，延迟为 90ns，吞吐量高达 16 Tb/s。2015 年，迈络思公司推出了首款可编程通道适配器。该通道适配器上额外搭载一块 Xilinx 公司 Kintex XCKU060 FPGA 芯片。用户可通过将部分数据迁移到 FPGA 进行加速处理，进而提高工作效率和降低延迟。

在 InfiniBand 结合 FPGA 实现方面，国外有公司提供 InfiniBand 协议层次的 IP 核。比如 Silicon Cores 提供 Link Protocol Engine IP 核，该 IP 支持全双工的独立发送和接收链路，支持所有的数据和管理报文的处理。IOWA 州立大学某团队采用 Xilinx Virtex-5 FPGA 芯片，实现了 InfiniBand 物理层和链路层，并应用于项目研究中 IBM 集群。文献[16]中，PolyBus 推出了 IB 传输层 IP，该 IP 符合 1.2 版本 IB 协议，支持可靠数据传输和不可靠数据传输两种方式，最大可建立 1024 个”队列对”，支持流量控制。文献[17]中，作者设计了一种同时支持 InfiniBand 和以太网的高速收发器。该收发器通过 FPGA 实现，FPGA 负责协议切换控制、数据包的封装和解析。该收发器可作为 IB 网络接口，提供高达 54 Gbps 的数据接收和发送能力。

相比于国外，InfiniBand 在国内的研究稍显滞后，并且少有公司进行 InfiniBand 产品的研发生产。但是在超级计算领域相关应用的研究上，国内取得了很大的成果。在最新的世界超级计算机性能评比中，来自中国的神威太湖之光和天河二号分别获得一二名。其中，神威太湖之光采用了 PCIe 与 InfiniBand 混和网络结构[18]。InfiniBand 用于 160 个 Supernode 之间以及外部存储和中央网络的连接。天河二号采用了私有协议（TH Express），但该协议中借鉴了很多 InfiniBand 技术，例如 RDMA 的读写机制、胖树网络拓扑、路由芯片独立等。

在理论研究方面，文献[19]中国防科技大学的曹光权针对高速传输下的拥塞问题，在文章“InfiniBand 网络拥塞控制的实现和观测”中对 IB 网络拥塞控制机制进行了深入分析和研究。根据 IB 网络拥塞产生原因，提出了拥塞反向传播机制、交换机拥塞标记算法以及适配器基于拥塞标记的速率调整机制等解决方案。

文献[20]中国防科技大学结合 Mellanox 的芯片提出了一种新的用户和通道适配器的连接架构。他们以 IB 网络与处理器节点之间接口实现模型作为研究点，针对现有接口的 PCI Express 总线存在传输延时高和安全隔离机制差等问题，提出了 CI2 这一新的 IB 网络接口的实现模型，该模型实现了 IB 网络与 CPU 直连，有效的减少了端系统延时，进一步提高了 IB 网络的高速传输性能。

文献[21]中作者针对分布式内存文件系统，设计了一种超高带宽、低延迟的互联方案。该方案采用 InfiniBand 互联技术，并在 Xilinx VC709 开发平台上实现了 Infiniband 精简接口。该接口通过层次结构精简、报文格式优化以及通信接口直连等途径，在 16 B 数据负载的情况下可以获得 1.31 us 的低延迟。

1.3 论文的组织章节

本文内容主要分为以下几个章节：

第一章绪论。介绍了本文中 InfiniBand 技术的发展历程和发展趋势，并对本文

的研究背景做了介绍以及对论文内容进行了安排。

第二章 InfiniBand 协议分析。首先对 InfiniBand 的拓扑结构和层次结构进行了介绍；其次对通道适配器的控制机制以及数据通信机制进行了分析与研究。

第三章通信控制器分析与设计。首先给出了 InfiniBand 接口设计方案，进行了 FPGA 功能模块划分。针对本文研究内容，对通信管理协议做出了分析并对 IB 接口中通信控制器部分进行了整体设计。

第四章通信控制器模块实现。采用模块化设计思路，对通信控制器总体框图中各个功能模块进行了设计与实现，包含模块结构、原理图和状态机。

第五章通信控制器仿真与验证。对接口通信控制器进行了功能仿真测试方案制订、RTL 级功能仿真和 FPGA 验证。并对验证结果进行了分析。

第六章总结与展望。总结全文工作，给出了当前研究的不足以及对未来的研究提出了建议和展望。

第二章 InfiniBand 协议分析

InfiniBand 以其高带宽和低延迟的特性而受到广泛应用和研究。本章将对 IB 协议进行介绍与分析，包括其拓扑结构、层次结构、通道适配器控制机制和 IB 通信机制。

2.1 InfiniBand 体系结构

2.1.1 InfiniBand 系统拓扑结构

InfiniBand 系统主要由四个部分组成，分别是通道适配器(Channel Adapter，CA)、IB 链路、交换机和路由器[1,22]。图 2-1 为 InfiniBand 系统拓扑结构。

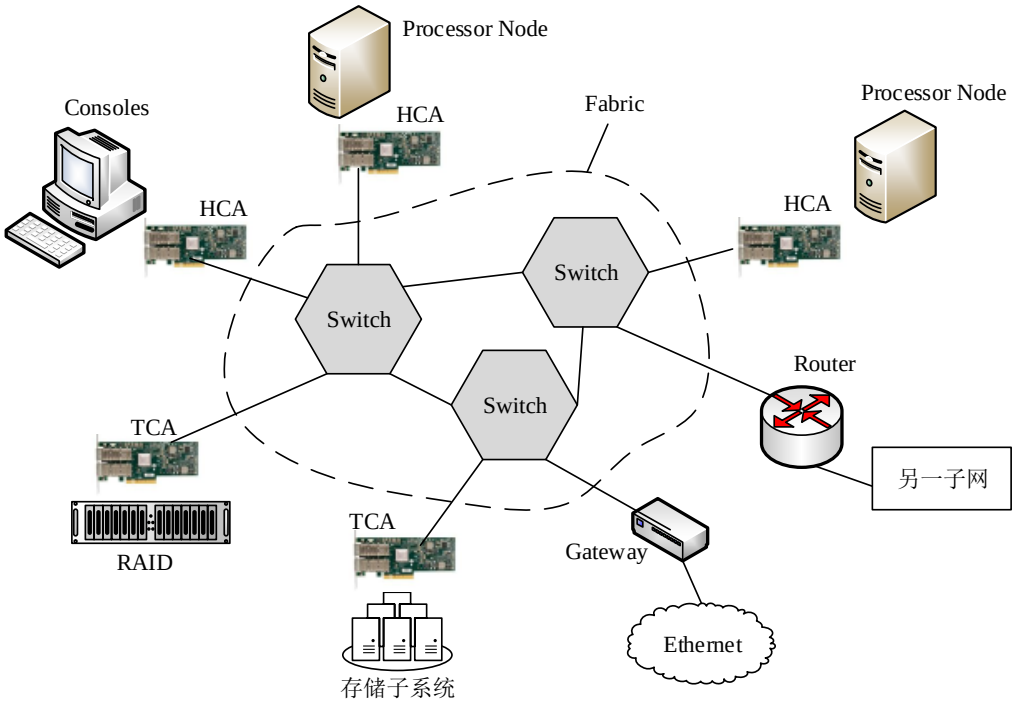


图 2-1 InfiniBand 系统拓扑结构[1]

通道适配器用于连接节点和链路。基于接入的节点种类，CA 分为目标通道适配器（Target Channel Adapter，TCA）和主机通道适配器（Host Channel Adapter，HCA）。目标通道适配器用于接入 I/O 节点，比如磁盘阵列和存储子系统。主机通道适配器支持接入层，用于接入用户节点。不同于 I/O 节点，用户节点可以通过 Verbs 命令对系统进行配置和管理以及发起数据通信。

IB 链路是双向点对点通信信道，传输介质既可以是铜线也可以是光纤。当前协议版本规定单条链路上的信令速率为 2.5 Gbaud。同时物理链路可以并行使用以实现更大的带宽。常见有 1X、4X、8X 和 12X 链路。链路最大支持 10Km 的连接距离。

交换机用于扩大 InfiniBand 网络，为子网中提供更多的连接端口，以便连接更多的端节点。交换机作为子网内数据包的基本转发组件，基于子网管理单元配置的路由表对数据包进行转发。同时，交换机既可以转发单播数据包，也支持转发广播数据包。

路由器用于互联多个子网络，负责将数据包从一个子网转发到另一个子网，而不消耗或生成数据包。与交换机不同，路由器读取全局路由头，根据其 IPv6 网络层地址转发数据包。因此，路由器的存在增加了网络的扩展性。

2.1.2 InfiniBand 协议层次结构

InfiniBand 架构分为多个层，每个层彼此独立，下层为上层提供服务。如图 2-2 所示，InfiniBand 协议由物理层、链路层、网络层、传输层、接入层、上层协议组成[1,23-24]。

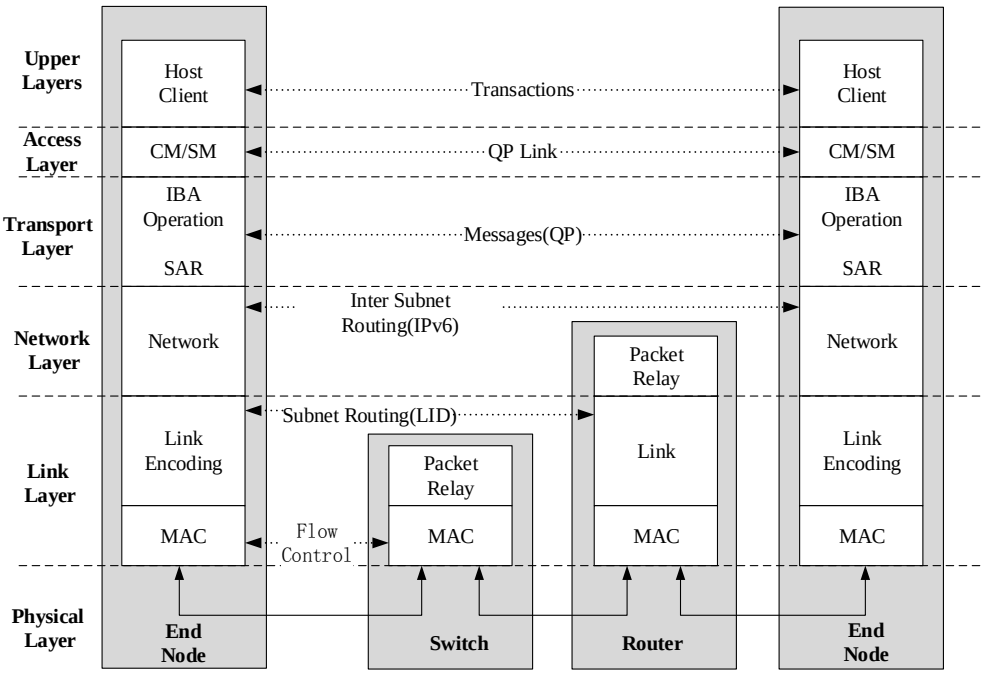


图 2-2 InfiniBand 协议层次[1]

（1）物理层

物理层定义了系统的机械和电气特性，包括背板端口、电缆端口和光缆端口三

种物理端口的特性和信号组成。物理层同时定义了链路层数据包字节流和物理介质的串行比特流之间的接口。接口使用行业标准 8B/10B 编码方式对数据包字节流进行编码，通过物理层制定的比特成帧方式将串行比特流放到物理链路上。除了编码和解码，物理层还包含链路培训、链路初始化、时钟容限补偿和错误检测等逻辑。

（2）链路层

链路层定义了数据包结构和数据包在本地子网内的交换。

链路层中有两种类型的数据包：通信数据包和管理数据包。通信数据包用于常规的数据传输，一个数据包可以携带多达 4K 字节的有效数据载荷。管理数据包用于数据传输链路信息配置和维护，比如虚拟通道支持等设备信息。

在同一个子网内，数据包的转发和交换在链路层处理。子网内所有节点具有子网管理器分配的 16 位本地 ID（Local ID，LID）。发送节点数据包经过链路层时，通道适配器将本地端口 LID 地址写入数据包的本地路由包头（Local Route Header，LRH）内 SLID（Source LID，源 LID）字段内，将目的节点 LID 信息写入数据包的 LRH 内 DLID（Destination LID，目的 LID）字段。交换机将通过数据包中 LRH 中 DLID 字段查询转发表，将数据包正确的转发到目的节点。目的节点通过验证 DLID，对数据包进行接收或丢弃处理。

（3）网络层

网络层处理子网间的数据包路由，数据传输双方节点在同一个子网内，则不需要网络层。子网之间发送的数据包在经过网络层时，节点将会对数据包全局路由包头（Global Route Header，GRH）进行添加或解析。GRH 中包含数据包的源和目的 128 位全球 ID（Globle ID，GID）。

（4）传输层

传输层负责数据包的通道复用、有序传输和传输服务（包含可靠连接、可靠数据报连接、不可靠连接、不可靠数据报连接和原始数据报连接）。传输层还负责事务数据发送时的分段以及接收时的重新组合。基于通信双方协定的数据通道最大传输单元（Max Transmission Unit，MTU），发送设备传输层将数据消息划分为适当大小的数据包，在数据前添加基本传输包头（Base Transmission Header，BTH），BTH 中包含传输层为每个数据包分配的分组序号。接收设备接收到数据包后进行接入权限验证，通过则基于数据包 BTH 中目的队列对信息和分组序列号进行数据重组。

（5）接入层

接入层为用户提供 InfiniBand 网络的传输级访问。该层次将传输层实现的全部功能展现给上层用户，增强了 verbs 命令传输接口。接入层包含子网管理单元和通

信管理单元，支持大多数 InfiniBand 用户组网、通信建立等请求。协调多个上层组件对共享的 InfiniBand 资源的访问[24]。

（6）上层协议

上层协议是指用户在满足 InfiniBand 架构规定下，为了兼顾应用软件所采用的一些通信协议。比如双方应用可通过 IPoIB（IP over IB）协议，利用物理 IB 网络通过 IP 协议进行连接，并进行数据传输。或者可以是用户利用 IB 网络提供的 Verb 机制和通信机制所自定义的一些满足自身需求的功能。

2.2 通道适配器控制机制

2.2.1 InfiniBand Verb 机制

在 InfiniBand 网络结构中，用户通过主机通道适配器接入交换网络。为了描述主机通道适配器和用户节点之间的服务接口，协议制定了 Verb 机制[1,25]。

InfiniBand 作为一个行业标准互联协议，不绑定到任何给定的操作系统或服务平台。因此，协议中没有直接规定任何应用程序编程接口（Application Programming Interface，API），而是通过 Verbs 来完全确定 APIs 的所需行为。这样既满足了用户针对特定操作系统的 API 开发，又确保了跨平台和跨供应商的互操作性[25]。Verbs 是 HCA 所有功能的一个抽象展示。Verbs 描述了配置、管理和操作 HCA 所需的全部功能以及定义了调用特定功能所需要的参数和格式。用户不会直接和 HCA 接口进行信息互动，而是通过调用这些 Verbs 来实现管理功能和网络通信等。

表 2-1 部分 Verb 命令类集与释义

Verb 命令	类集	释义
Open HCA	强制	启动通道适配器
Modify HCA Attributes	强制	配置通道适配器
Close HCA	强制	关闭通道适配器
Create Queue Pair	强制	创建队列对
Create EE Context	可选	创建可靠数据报服务
Post Send Request	强制	发起发送请求
Create Shared Receive Queue	可选	共享接收队列创建

Verbs 被分为强制和可选两大类集[1]：强制类为保证系统正常工作所必须的基本操作，所有供应商生产的主机通道适配器必须支持此类 Verbs；可选类是指在完

成某些可选功能时被应用的命令。表 2-1 中罗列了部分 Verb 命令。

强制类 Verbs 包含 HCA 控制、内存注册、队列对处理、通信请求和完成事务处理几个方面。其中 HCA 控制包含打开、初始化、配置和关闭通道适配器。内存注册包含向通道适配器内存注册、虚实地址映射、注册内存释放、内存区域和缓存窗的绑定等操作。队列对处理包含创建、查询、调整和释放队列对等操作。通信请求则是系统初始化后用户发起数据交换的请求，包含发送和接收操作。完成事务处理则包括事件轮询、同步和异步事件处理操作。可选类则是基于额外的功能而设定的操作，比如基于共享接收队列的创建、修改和释放操作。

2.2.2 通道适配器命令控制机制

上一小节讲述了 IB 协议中用于控制 HCA 的 Verb 命令机制。为适应多平台，协议中仅仅定义了每一个 Verb 命令的输入参数、输出参数和该条命令的操作意义。在实际运用中，具体用户何时调用 Verb 命令由上层协议决定，每条命令如何和 HCA 进行命令数据交互由供应商决定。本文以 Mellanox 公司的 ConnectX-3pro 主机通道适配器作为研究对象，对通道适配器命令控制机制进行说明。

ConnectX-3pro 通道适配器中有两块供用户读取和配置的寄存器区域[26]，一块名为全局设备配置寄存器（Global Device Configuration Registers，GDCR），用于设备初始化配置；另一块名为用户访问区域（User Access Region，UAR），用于通信请求和事务处理等操作。在用户节点初始化阶段，系统将为 HCA 所有寄存器区域进行地址分配，通过区域初始地址和偏移量，用户可以对该区域的每一个寄存器进行数据的读取与配置。

在 GDCR 区域中有一块 HCA 命令寄存器（HCA Command Register，HCR）用于命令控制，该寄存器块布局如图 2-3 所示。

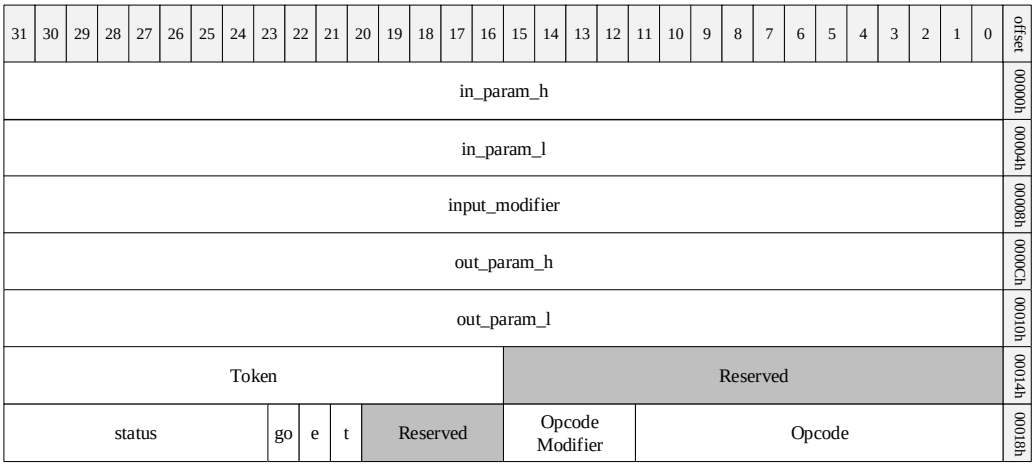


图 2-3 HCR 寄存器格式

其中，一些关键寄存器的释义如表 2-2 所示。

表 2-2 HCR 寄存器关键字段释义

Name	释义
Opcode	指示此次命令模式操作编码，对应不同 Verb 命令种类
Out_param	输出参数或系统用于存放输出参数的地址，共 64 位，分为高 32 位和低 32 位
In_param	输入参数或系统用于存放输入参数的地址，共 64 位，分为高 32 位和低 32 位
T	首条命令该值为 1，以后每次该值翻转指示新一条命令
Go	指示 HCA 工作状态，为 1 指示正在工作
Status	命令执行状态报告

用户对 HCA 进行命令控制，就是将 Verb 命令的输入数据写入 HCR 寄存器区域 In_param 寄存器。如果某次命令有通道适配器返回数据，则用户可通过读取 Out_param 寄存器获得。一次完整的命令执行流程如图 2-4 所示。

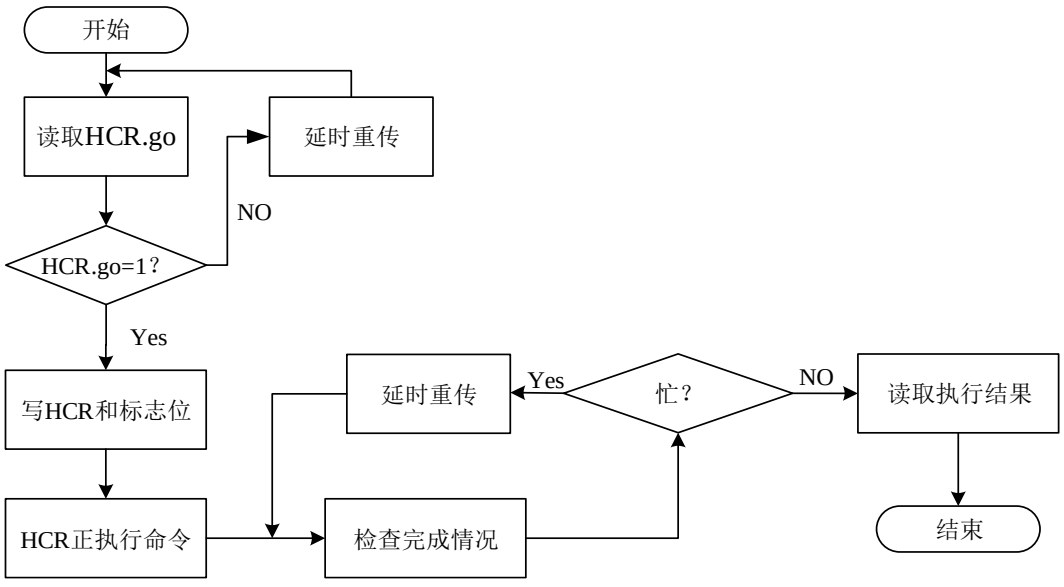


图 2-4 Verb 命令执行流程图

当用户有 Verb 命令执行请求时，系统首先会通过查询通道适配器 HCR.go 字段来判断是否空闲：如忙，则根据配置的时间进行重查操作；如空闲，则用户封装 Verb 命令和更新标志位，并写入系统为 HCR 寄存器分配的地址。然后通过查询通道适配器忙闲状态判断是否执行完成，完成则读取执行结果，否则等待重传。

2.3 InfiniBand 通信机制

2.3.1 InfiniBand 数据包格式

在数据通信过程中，通信发送方上层应用数据在经过各层协议时，业务数据被映射到封装协议的净荷中，同时填充对应层次的包头数据。最终形成 IB 协议的数据包，并在物理层经过封装成比特流送到物理链路中[1,4]。接收方则按照数据封装的逆过程，逐层拆解数据包，提取包头中的有效信息，判断数据包下一步处理过程，最终获得发送方应用数据。图 2-5 为 InfiniBand 数据包格式。

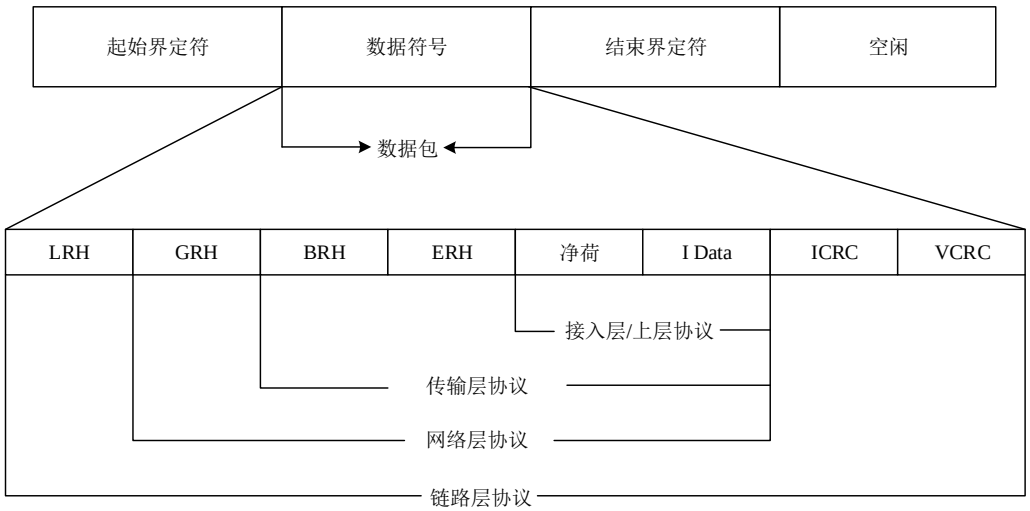


图 2-5 InfiniBand 数据包格式[1]

图 2-5 中，各个数据部分意义和应用如下：

LRH：本地路由包头，包含发送节点和目的节点在子网中的地址信息，子网内的交换机通过 LRH 中地址信息对数据包进行路由。

GRH：全局路由包头，通信双方不在同一个子网内时，路由器通过 GRH 中 GID 信息路由数据包至发送方本地子网以外的目标。

BTH：基本传输包头，包含目的地用于接收该数据包的队列对信息，并且指示是否为可靠数据传输数据包，如若为可靠传输，目的节点需为该数据包回复一个确认数据包。当数据块超过系统最大传输单元时，传输层会对数据进行分段处理，并在 BTH 中添加数据序列号，接收方通过该序列号对数据进行重组。

ERH：扩展传送包头，在特定传输类型下的扩展补充信息。比如 RDMA 操作下的虚拟地址，数据块大小等信息。

I data：即时数据，用户传递的一些较小数据块。

ICRC：恒定循环冗余校验。

VCRC：可变循环冗余校验。

2.3.2 基于队列对的数据传输模型

在 InfiniBand 系统拓扑结构中，各节点通过通道适配器接入网络。而在逻辑层面上，IBA 定义了队列对（Queue Pair，QP）作为数据通信链路的终端[1,3,24-25]。队列是指用户向队列对传递的通信请求将按照传入顺序排队等候通道适配器执行；而每个 QP 由一个发送队列和一个接收队列成对组成，因此被称为队列对。图 2-6 为数据传输模型示意图。

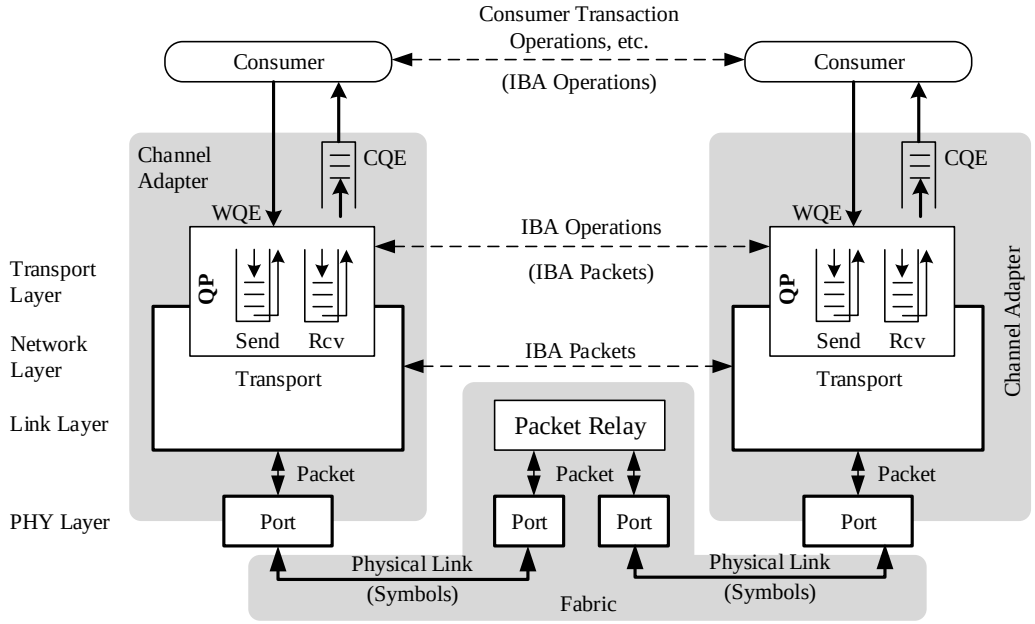


图 2-6 数据传输示意图[1]

在通道适配器初始化阶段，用户根据连接需求通过 INIT HCA 命令进行 QP 的创建，并为每个 QP 分配 24 位的 QP 序列号。每个用户最多可以创建 16M 个 QP 用作数据链路终端。初始化创建的 QP 将处于复位（reset，RST）状态，除了 RST 状态外，QP 还可能处于初始化（initial，INIT）状态、可接收（ready to receive，RTR）状态和可发送（ready to receive，RTS）状态[26-28]。其中 RTR 状态下 QP 可以进行接收数据操作，RTS 状态下 QP 既可发送又可接收数据。图 2-7 为 QP 状态转换图和引起状态转换的 Verb 命令。

当不同用户之间有数据通信需求时，通信双方将按照通信建立协议，将用于此次通信的双方 QP 连接形成逻辑上的双向数据传输通道。该数据传输通道支持 Send/Receive（发送/接收）、RDMA（Remote Direct Memory Access，远程内存访问）和 Bind（绑定）这三类协议规定的基本传输类型。

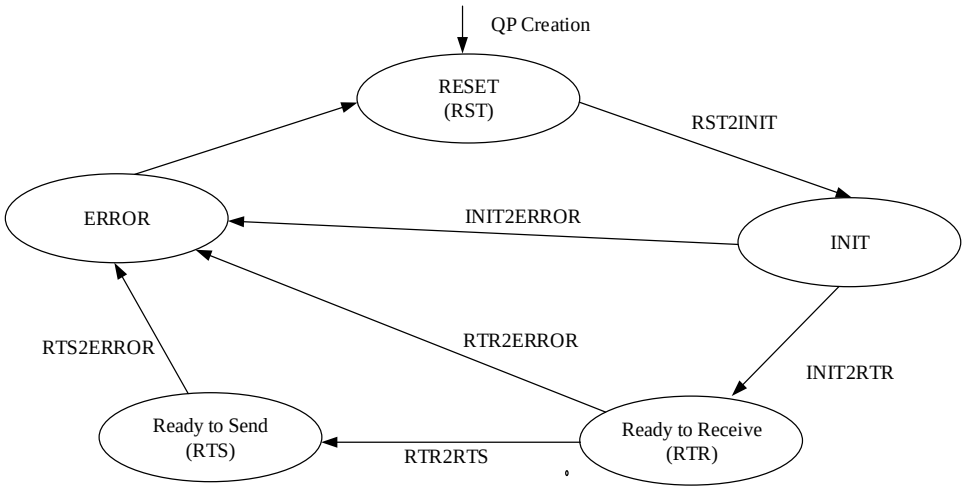


图 2-7 QP 状态转换及对应 Verb 命令

一旦用户间的数据传输通道建立完成，QP 将成为用户双向数据传输引擎。引起数据传输的用户指令被称作工作队列元素（Work Queue Entry，WQE）。多个 WQE 将在队列中按照写入顺序排队等候硬件执行。图 2-8 为 WQE 格式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ctrl Segment																															
Datagram Segment																															
FC Address Vector Segement																															
FC Init segment																															
Large Send Offload Segment																															
Memory Management(Bind,FRWR,Local invalidate																															
Remote Address																															
Atomic/Extended atomic																															
Data Segments(Pointer & Data)																															

图 2-8 WQE 格式[26]

如图 2-8 所示，每个 WQE 可包含多个片段，按照操作类型，WQE 携带不同的有效片段。在发送/接收操作中，数据发送方将发送 WQE 写入数据链路本方 QP 发送队列，该 WQE 由控制片段和数据片段组成。其中控制片段包含一些控制信息。数据片段携带本地数据起始地址、数据长度和安全 Key 等信息。当队列执行顺序轮到该 WQE 时，通道适配器读取该指令，并按照 IB 协议逐层添加包头信息将待发送数据以数据包形式发送。对于接收方，用户将接收 WQE 写入数据链路对应 QP 接收队列，接收队列[29]仅包含数据片段，数据片段地址信息指示了用户用于存放接收数据的内存位置。对于 RDMA 读写操作，发起方中 RDMA WQE 则由控制片段和远端地址片段组成。远端地址片段指定了写操作的目的地址或读操作的接收

存储区域地址，数据接收方无需消耗接收队列中 WQE。

为了反馈用户指令的完成情况信息，协议制定了完成队列（Completion Queue，CQ）机制[30]。完成队列是指一块用于存储用户指令完成情况的存储空间。在初始化阶段可以创立多个 CQ，用户可通过 INIT CQ 命令将 CQ 和 QP 进行映射并绑定。例如，将 CQ1 和 QP1 进行绑定操作后，通道适配器将会把 QP1 队列中用户指令完成情况以完成队列元素（Completion Queue element，CQE）形式存入 CQ 中。用户通过轮询完成队列，读取 CQE 并解析其中有效信息，用以指示下一步操作。

2.3.3 通用管理报文传输过程

在 IB 网络架构中，按照数据包净荷的不同，节点间传递的数据包分为两类[1,4]：一类用于通信数据传输；一类用于管理数据传输。用于管理数据传输的包被称为管理数据包（Mangement Datagram，MAD）。MAD 是管理消息传递的基本元素。网络结构中各节点间通过各种 MAD 消息的交互，完成对网络的管理以及保证通信的顺利进行。用于通信管理的 MAD 叫做通用管理数据包（General Management Packet，GMP），用于子网管理的 MAD 叫子网管理数据包（Subnet Management Packet，SMP）。在本文研究背景下，系统主要涉及 GMP 的传输。

GMP 传输作为数据传输的一种，通信链路端口同样为 QP。按照协议规定，每个 IB 用户节点必须存在序列号为 0 和 1 的队列对[31]，QP0 和 QP1 被称为特殊队列对。特殊队列队在适配器初始化完成后，用于传递管理数据报文。其中 QP0 用来传递 SMP，QP1 用来传递 GMP。在传输方式上，特殊队列对采取了发送/接收（Send/Receive）数据传输方式，该传输方式更加适合传输数据量小的管理控制数据。特殊队列对和完成队列绑定设计时，本文中，系统采用一一对应方式，即通过绑定 CQ1 和 QP1 来指示 GMP 的发送与接收完成情况。

在网络初始化完成，子网内每个用户被分配本地 ID 后，某用户有 GMP 发送需求时，发送方需向 QP1 发送队列存储区间写入发送 WQE，该 WQE 指定了通用管理报文数据在本地的存储地址以及访问该段数据的权限信息，数据长度固定为 256Bytes。随后向通道适配器用户接入寄存器区域发送一个“门铃信息”，该门铃信息用于指示 QP1 中有 WQE 正在队列中等待执行。通道适配器接收到门铃信息并读取队列中 WQE，按照指定的操作方式对数据进行处理，处理包括数据的读取，数据的包头逐层添加，最终发往数据链路。当 GMP 发送完成后，通道适配器向 CQ1 中写入一个包含发送完成信息的 CQE，用户轮询 CQ1，读取和解析该 CQE，判断是否发送成功并继续后续操作。

GMP 接收方首先在内存区域划分一段地址空间用于存放 GMP，区间起始地址

和长度将写入接收队列中接收 WQE 数据片段。通道适配器接收到数据包将按照协议层次逐层解析包头信息，当数据包 BTH 中 QPN 字段指示了本地用于接收此数据的队列对为 QP1，通道适配器将读取 QP1 接收队列 WQE，并按照 WQE 中指定的地址信息将数据存入相应存储区间。完成数据接收后，通道适配器向 CQ1 中写入一个包含接收完成信息的 CQE。用户轮询 CQ1，读取和解析该 CQE，如若接收成功，则读取该存储区域获得 GMP。

2.4 本章小结

本章节首先介绍了 InfiniBand 系统的体系结构，包含拓扑结构以及协议层次结构。紧接着针对 InfiniBand 的 Verb 命令控制机制以及迈络思公司研发的通道适配器对应的控制机制做了详细介绍和分析。包括命令分类、命令的格式和命令的执行流程。最后研究了 InfiniBand 的通信机制，并针对本文研究内容，给出了 GMP 传输过程。本章是后续设计的基础，设计需满足 InfiniBand 协议中的具体规范。

第三章 IB 接口通信控制器分析与设计

在对 IB 协议（特别是通道适配器控制机制和 IB 通信机制）进行了研究与分析基础上，本章将针对实际应用场景，完成 IB 接口的设计方案和硬件功能划分。并着重对 IB 接口中通信控制器部分进行了设计。

3.1 IB 接口规划

3.1.1 IB 接口分析

本文在第二章节中针对 InfiniBand 体系结构做了介绍，从图 2-1 可以看出，处理器节点通过通道适配器接入 InfiniBand 网络。在集中式基站应用场景中，我们采用 FPGA+HCA 的接口设计方案，用于建立各个基站和基带处理中心的高效数据交换网络。图 3-1 为接口应用场景图。

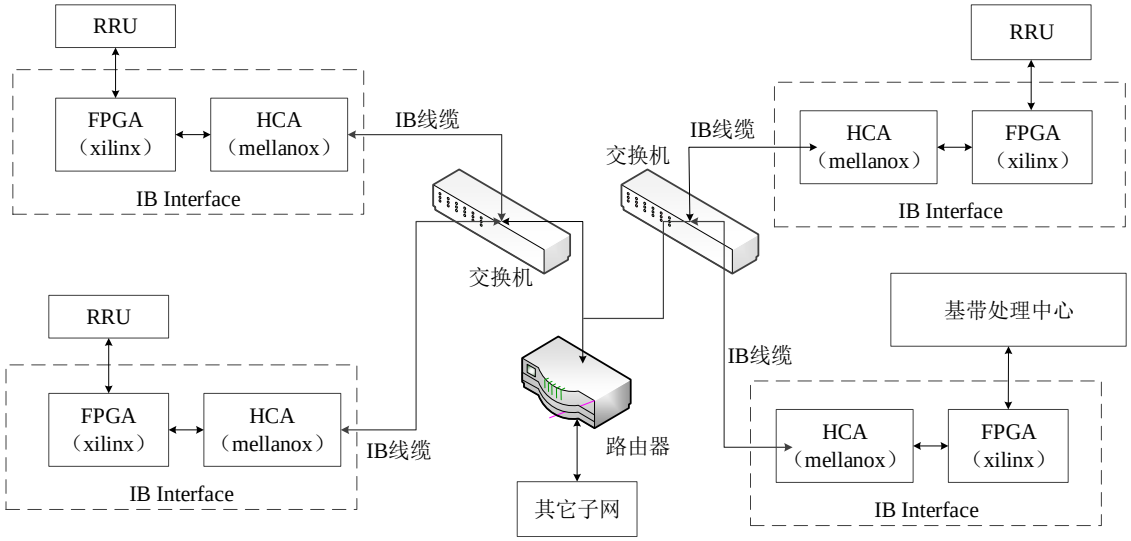


图 3-1 IB 接口应用场景示意图

图 3-1 中，虚线部分 FPGA 和 HCA 一起构成 IB 接口，并通过 IB 线缆接入交换网络。接口采用了 Xilinx Virtex7 系列 FPGA 和 Mellanox ConnectX-3 Pro 通道适配器。在集中式基站布局中，由于基站基带数据处理集中化，各个基站中将不再配置高性能服务器用于控制数据转发，FPGA 将作为 IB 接口的核心控制部分负责基站上行链路基带数据向处理中心的集中以及下行链路数据的缓存。

结合 2.1 小节的分析，IB 接口的主要功能是能按照 IB 协议的子网管理和通信管理进行网络拓扑和数据链路的建立，以及按照 IB 协议层次产生和处理 IB 数据

包。即 FPGA 逻辑能和主机通道适配器一起协同完成 IB 协议工作。图 3-2 为二者协同工作中 FPGA 与 HCA 对应协议层次。

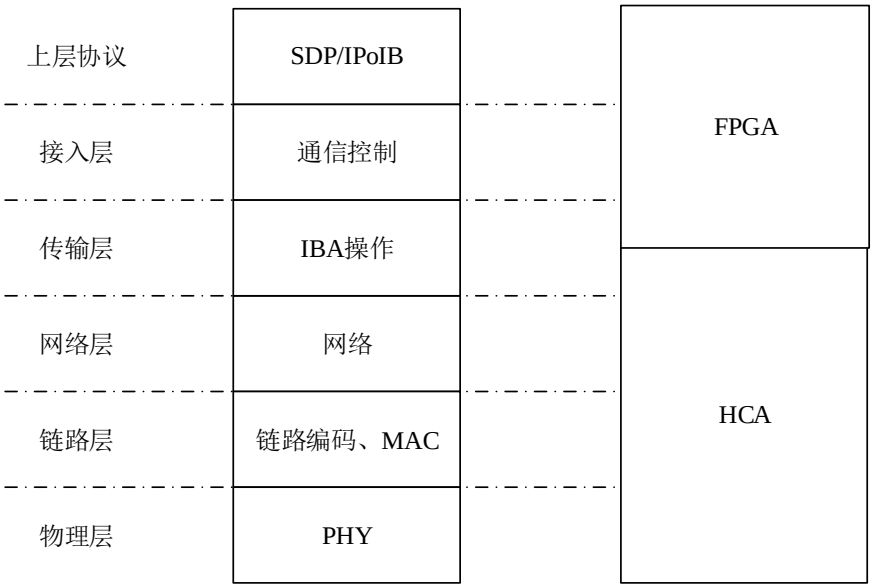


图 3-2 IB 接口 FPGA 与 HCA 对应协议层次

在 IB 接口中，HCA 已经实现了 IB 协议层次中物理层、链路层、网络层和部分传输层的功能，因此 FPGA 内部控制器的主要功能有两个：一是完成协议层次中剩余传输层部分以及传输层以上的层次。结合 2.1.2 小节介绍，FPGA 将提供传输层的数据传输服务、接入层的通信管理和子网管理功能以及用户自定义的上层功能或 SDP 协议；二是按照通道适配器控制机制对 HCA 进行数据交互，使二者能够共同完成协议层次工作。

3.1.2 IB 接口设计

结合本文研究背景、IB 协议的体系结构和 IB 的工作机制，图 3-3 中给出了 IB 接口的硬件设计框图，其中深色部分为作者负责部分，名为通信控制器。

从图 3-3 可以看出，IB 接口主要由 FPGA 和 HCA 两大硬件部分组成。IB 接口各模块采用 AXI4[33-34]总线进行互联。各模块主从角色通过连接线 M(Master)和 S（Slave）区分。因为通道适配器既会主动对从设备模块进行数据读写，又会被主设备模块数据配置，因此在接口中同时具有主从设备角色。按照图 3-2 协议工作层次划分及各层次功能定义，FPGA 主要完成部分传输层、接入层和上层等功能。

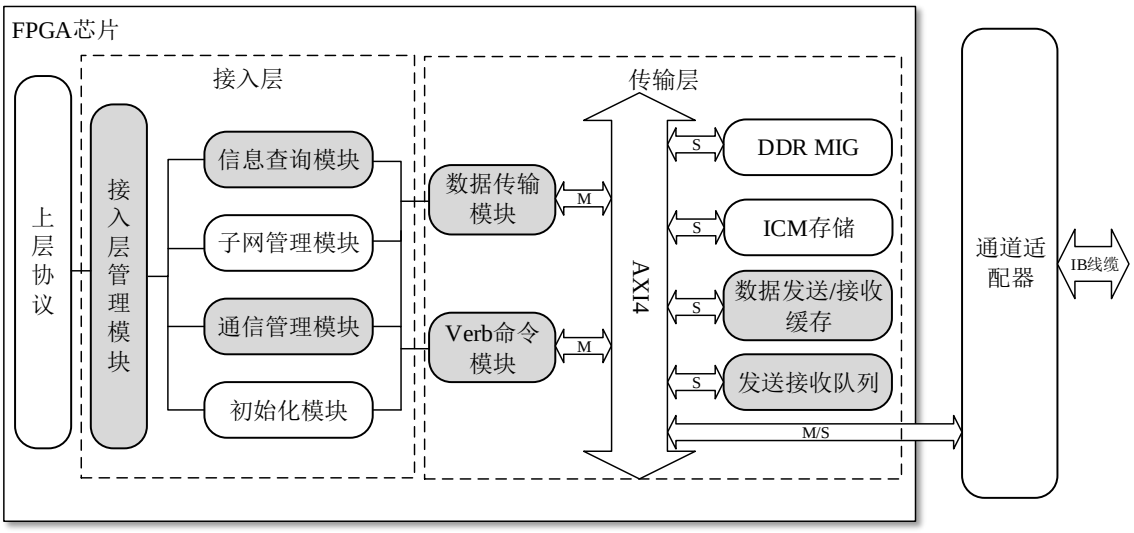


图 3-3 InfiniBand 接口总框图

图 3-3 中数据传输模块、Verb 命令模块、DDR MIG IP 核、ICM 存储模块、数据发送/接收缓存模块和发送接收队列模块对应 IB 协议传输层。数据传输模块按照 IB 通信机制，接收数据传输请求，生成 WQE 并存入发送接收队列模块并接受通道适配器读取。Verb 命令模块接收接入层 Verb 命令请求，完成对通道适配器的控制。ICM 存储模块用于存储通道适配器控制字。数据发送/接收缓存模块用于待发送数据和接收数据的缓存。传输层部分模块和通道适配器协作完成接入层以下层次协议功能，可为上层提供数据传输和 Verb 命令执行等服务。

初始化模块、信息查询模块、子网管理模块、通信管理模块和接入层管理模块为接入层组成部分。各模块接受传输层提供的服务，共同完成系统初始化、子网管理和通信控制等功能。其中通信管理模块和下层功能模块一起组成通信控制器，完成通信管理协议的硬件描述与实现。IB 接口的工作流程如下：

（1）系统上电复位后，初始化模块将按照启动序列向 Verb 命令模块发送初始化 Verb 命令，命令包含 HCA 设备信息查询、HCA 寄存器地址初始化、QP 的创建、CQ 的创建、QP 和 CQ 的映射关系等命令，Verb 命令模块将按照通道适配器命令控制机制与 HCA 进行信息交互，并将返回数据和完成情况进行反馈。

（2）当本地节点设备初始化完成后，节点设备按照子网管理协议主动探索子网或被子网所发现，此时子网管理模块将负责整个子网的地址分配或接受子网对本地的地址分配信息。同时该模块还将负责网络拓扑异常变动的监控和处理。

（3）子网建立完成，各个节点将被分配独一无二的子网地址。用户节点间如有通信需求，用户将向通信管理模块传递待建立通道属性和对方地址信息。通信管理模块通过调用数据传输模块和 Verb 命令模块完成链路的建立。

（4）数据传输通道建立完成，用户向队列对通信指令，通信指令可以是发送、接收、RDMA 写或读数据操作。HCA 对通信指令进行解析，结合数据传输通道信息，数据将按照 IB 协议层次逐层添加包头，最终经过 IB 线缆发出。相反，接收方则逐层解析包头信息，重组有效数据，保证数据被用户正确接收。

（5）数据传输完毕，用户可以选择保留数据传输通道或者关闭。如需要关闭通道，通信双方通信管理模块将负责通信链路的拆除

在 IB 接口中，接入层管理模块、通信管理模块、数据传输模块、Verb 命令模块和队列对模块互相协作，负责完成通信链路的建立与拆除过程，以及数据传输的控制，被称为通信控制器，是本文主要研究对象，接下来文章将对通信管理协议做相关研究和分析并对接口框图中通信控制器部分进行详细介绍。

3.2 通信管理协议

通信管理协议作为 IB 的重要组成部分，位于接入层，定义了通信链路建立和拆除过程，包含过程中通信管理报文格式、数据包交互、QP 状态转换和超时重传机制。该协议是通信控制器设计的基础。

3.2.1 通信管理报文格式

用于通信管理的数据包 GMP 由首部和数据两部分组成。报文总长度为 256 字节，其中前 24 字节为首部，后 232 字节为数据部分[35]。GMP 格式如表 3-1 所示

表 3-1 GMP 包格式

<table>
<tr><td>Bytes</td><td>Bits 31-24</td><td>Bits 23-26</td><td>Bits 15-8</td><td colspan="2">Bits 7-0</td></tr>
<tr><td>0</td><td>BaseVersion</td><td>MgmtClass</td><td>ClassVersion</td><td>R</td><td>Method</td></tr>
<tr><td>4</td><td colspan="2">Status</td><td colspan="3">ClassSpecific</td></tr>
<tr><td>8</td><td colspan="4" rowspan="2">Transaction ID</td></tr>
<tr><td>12</td></tr>
<tr><td>16</td><td colspan="2">Attribute ID</td><td colspan="3">Reserved</td></tr>
<tr><td>20</td><td colspan="4">Attribute Mod</td></tr>
<tr><td>24</td><td colspan="4" rowspan="3">Data</td></tr>
<tr><td>…</td></tr>
<tr><td>252</td></tr>
</table>

首部中各个字段的具体含义如表 3-2 所示，其中 MgmtClass、Method 和 Attribute ID 三个字段共同决定了包种类。

表 3-2 GMP 首部字段含义

名称	长度	含义
BaseVersion	8 比特	GMP 格式版本信息，当前版本该值为 1
MgmtClass	8 比特	指定操作集
ClassVersion	8 比特	操作集特定版本，当前版本该值为 2
R	1 比特	指示 GMP 接收方是否需要回应一个数据包
Method	7 比特	基于某一操作集的具体手段
Status	16 比特	当前操作状态编码
ClassSpecific	16 比特	用于子网管理，在通信管理报文中无效
Transaction ID	32 比特	如使用，则表示交易指示符
Attribute ID	16 比特	决定了该包的具体操作类型。包含建立请求、建立回复、拒绝、终止请求、终止回应和建立完成等
Attribute Mod	32 比特	用于子网管理中

在 InfiniBand 系统中，基于产生 GMP 的管理单元实体，GMP 分为设备管理集、基板管理集、通信管理集以及部分用户自定义的管理集，在 GMP 首部中通过 MgmtClass 字段来予以区分。用户在接收 GMP 过程中，需通过该字段正确路由到对应管理实体中。本文 GMP 主要用于通信管理，属于通信管理集，MgmtClass 字段值为 0x07。

在通信管理集下，首部 Method 字段可配置为四种操作模式：属性获取、属性配置、请求回应和管理信息发送。对于本设计，通信管理报文主要用于向对方节点传输链路信息，因此报文中 Method 字段配置为管理信息发送，其 Method 字段值为 0x03。

Attribute ID 字段则用于区分发送的不同链接管理信息。对于通信管理单元，在运行期间会涉及到 REQ、REP、RTU、REJ、DREQ、DREP 六种通信管理报文的发送与接收。表 3-3 为每种通信管理报文的 Attribute ID 字段值和具体含义。

其中，作为客户端的用户节点 CM 需要支持发送 REQ、REJ、RTU、DREQ 和 DREP 数据包，响应 REP、REJ、DREQ 和 DREP 数据包。接收通信建立请求节点 CM 需要支持响应 REQ、REJ、RTU 和 DREQ 数据包，发送 REP、REJ、DREP 数据包。

表 3-3 通信管理 GMP 包属性

Name	Attribute ID	含义
REQ	0x0010	通信建立请求
REP	0x0013	通信建立请求回复
RTU	0x0014	通信建立完成
REJ	0x0012	拒绝
DREQ	0x0015	链路拆除请求
DREP	0x0016	链路拆除请求回复

由于不同种类的数据包，其数据部分格式不一，且每种数据包总长度为256Bytes。因此，本文将以 REQ 为例，给出 REQ 数据包部分关键信息，如表 3-4 所示。

表 3-4 REQ 数据部分关键信息

Name	Byte[Bit]偏移	长度（bits）	含义
Local_ComID	24	32	本地通信 ID
Local_GUID	40	64	本地全球唯一标识符
Local_QPN	56	24	此次链路建立的 QP 序列号
Service TYPE	27[5]	2	链路服务类型
Timeout	71	5	超时重传机制中时间指示信号
Max_RETRY	75	4	最大重传次数
Remote_LID	78	16	远端节点地址信息
Private Data	164	736	携带的私有数据

3.2.2 通信管理实体

通信管理协议定义了在 IB 网络节点间建立、维护和释放数据传输信道的机制。支持建立节点间可靠连接、不可靠连接和可靠数据报等传输服务类型通道[35]。在网络系统中，通信管理实体如图 3-4 所示。

从图 3-4 可以看出，在通信双方节点 A 和 B 中，均存在一个通信管理模块（Communication Management，CM）。通信管理模块的作用为接收以及处理通信链路建立或拆除请求。通过 CM 模块，应用之间可以建立起基于 QP 的可靠连接通道

或可靠数据报通道。可靠连接通道为 QP 间一一对应关系。可靠数据报通道，则可映射多个通信 QP 到一个 EEC 上，用于一些广播信息传输。每个应用支持建立多个数据传输通道。

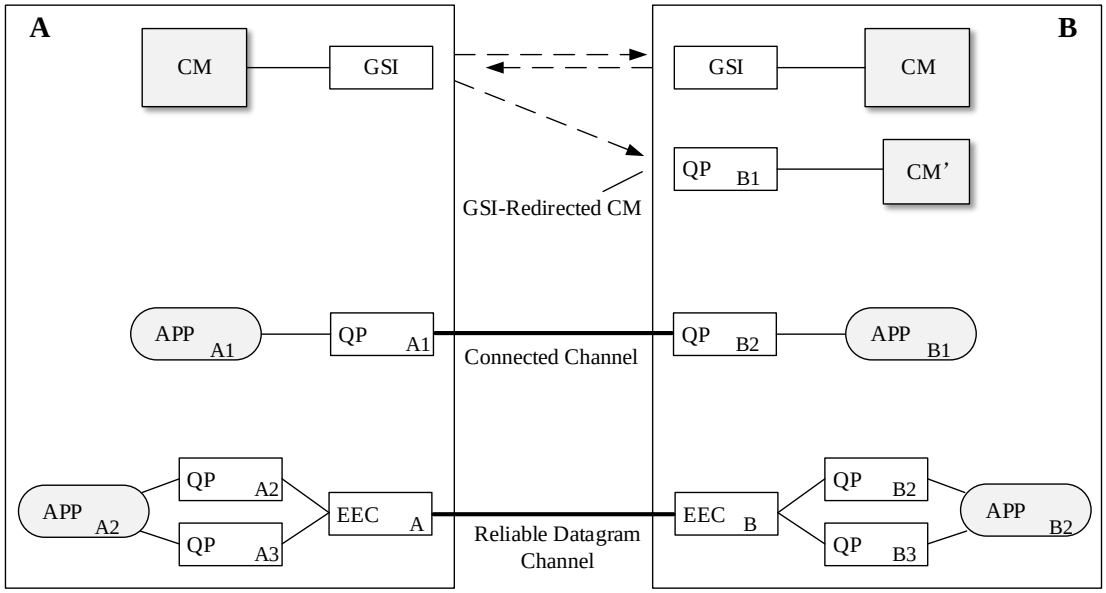


图 3-4 节点通信管理实体

通信管理模块之间通过通用服务接口（Globle Service Interface，GSI），也就是 QP1 交互 GMP 进行数据通信。作为全双工的数据传输协议，各节点 CM 能独立的收发通信管理报文。由于接入层还存在基板管理、性能管理和设备管理等通用模块对 QP1 有使用请求，为缓解数据传输拥堵，通信管理协议支持 CM 接口迁移，通过使用普通 QP 来代替 QP1 完成通信管理报文的收发，如图 3-4 中 B 节点通过 B1 进行 GMP 包的接收。该方案可有效提升 CM 之间的数据传输效率。

3.2.3 通信链路的建立过程

同 TCP 协议类似，InfiniBand 通信管理协议也是面向连接的协议。通信双方节点用户在进行数据传输之前，需要先在二者之间建立一条数据链路，该条链路以通信双方 QP 作为端点。IB 链路建立过程需要三次通信管理报文的交换[1]和双方 QP 三次状态转换。通常，我们将主动发起连接的节点用户称为客户端（Client），被动接收连接的用户称为服务器端（Server）。图 3-5 为客户端和服务器端数据链路建立过程示意图。

从图 3-5 中可以看出，数据链路建立的过程如下：

（1）客户端节点 A 在初始化后将创建本地 QP。在接收到通信建立请求后将向服务器节点 B 发送操作类型为 REQ（Request）的通信管理报文，该报文符合标

准 GMP 格式。REQ 首部信息中 Method 标识字段为 0x03，Attribute 字段为 0x0010。数据部分由 140 字节的信息数据和 92 字节的私有数据组成。其中信息数据包含网络为 A 节点分配的地址信息、待通信节点 B 的网络地址信息、A 节点用于此次连接的 QP 序列号、待建立数据通道的服务类型以及为了 B 节点能可靠接收到请求信息所设定的超时重传信息等。私有数据则是 B 节点用于判断是否接受此次通信建立请求的关键信息，通信管理协议未对此段数据做格式定义，通信管理模块将应用传输的数据放入该部分。并将 QP 的状态从 RST 转换到 INIT 状态。

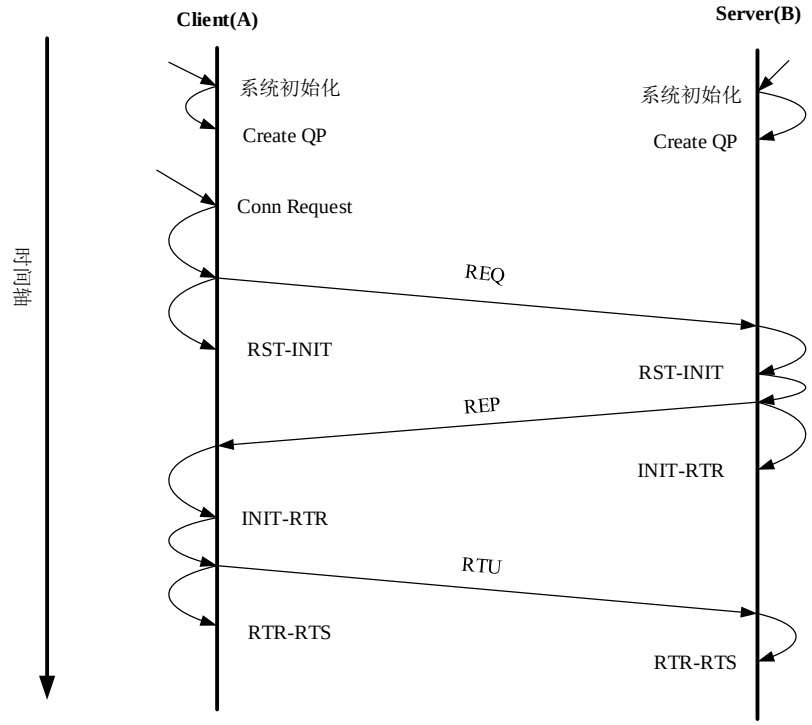


图 3-5 链路建立数据包交互和 QP 状态转换过程

（2）服务器节点 B 接收到节点 A 的 REQ 报文，接受此次建立请求，将本地 QP 的状态从 RST 转为 INIT 状态，并向客户端节点 A 发送操作类型为 REP 的通信管理报文，发送成功后将 QP 状态转为 RTR 状态。REP 数据部分包含节点 B 用于响应此次数据链路请求的 QP 序列号、重传计数和 B 节点传递给 A 的私有数据。B 节点拒绝此次连接请求情况将在后面设计中阐述。

（3）客户端节点 A 接收到 REP，向服务器端节点 B 发送操作类型为 RTU（Ready to Use）的通信管理报文并将本地 QP 状态转换为 RTS 状态。RTU 作为链路建立确认报文，数据部分仅包含通信双方通信 ID 和私有数据。客户端节点 B 在接收到 RTU 后将本地 QP 状态转换为 RTS 状态。

经过以上三次数据交换后，两个节点之间的数据传输链路就建立完成，同时双

方链路端点 QP 也将处于 RTS 状态，双方节点可按照协议通信方式通过此链路进行全双工的数据传输。

3.2.4 通信链路的终止过程

由于通信双方建立的是可靠地全双工数据传输通道，通信双方任意一方都可以主动关闭连接[36]。通信管理协议规定双方通过交换 DREQ（Disconnect Request，拆除请求）和 DREP（Disconnect Reply，拆除请求回复）来终止连接，同时将本地 QP 转换为初始化状态。图 3-6 为一次链路拆除过程

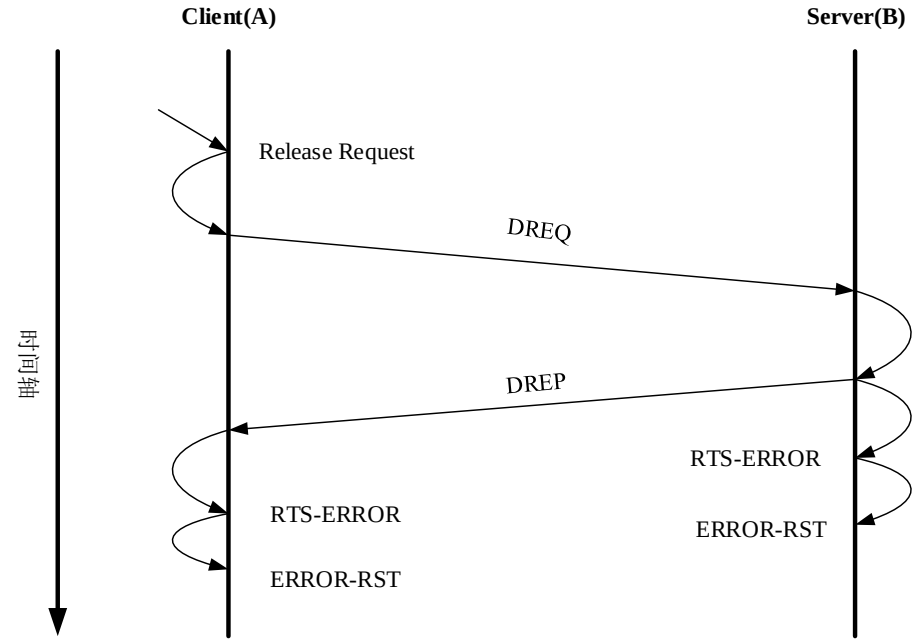


图 3-6 链路拆除数据包交互和 QP 状态转换过程

当某一方准备终止某条数据传输通道时，将向另一方发送一个操作类型为 DREQ 的通信管理报文。数据部分包含该条链路另一端 QP 序列号，通信双方通信 ID 和私有数据。接收到 DREQ 的节点将回复一个操作类型为 DREP 管理报文作为确认并将本地 QP 状态转换为初始化状态。DREP 数据部分包含终止链路双方通信 ID 和私有数据。链路终止发起方接收到 DREP 后将 QP 转换为初始化状态，链路拆除完成。

在终止数据传输通道过程中，需注意失效连接（Stale Connection）的产生[37]。失效连接是指由于某些原因通信双方仅有一方存有连接信息。原因可能是远端通信管理单元已经发送 DREQ 并终止链路，但是未被本地通信管理单元所接收到。这样本地节点将会仍然保存连接信息。同样，当远端节点突然崩溃或者重启，而本

地节点未获得消息而将保存连接信息。当出现失效连接，用户通过此通道进行数据传输时，通道适配器将通过事件报告机制予以提醒。用户可根据需求删除连接信息或重连。

3.2.5 超时重传机制分析

通信管理协议作为一种可靠的传输协议，制定了超时重传机制用于差错控制。目前，在重传计数小于最大重传次数情况下，主要有两种情景会触发重传机制：一是发送端节点在发送数据包后，在规定时间内未接收到远端节点的回复；二是发送端节点发送数据包后，接收到了远端节点的 REJ 数据包，且 REJ 数据包中的原因符合重传规则。

在实际运行过程中，客户端用户通信管理实体在进行数据链路建立过程中，通过 REQ 携带最大重传次数信息。服务器端接收并解析 REQ 获得最大重传次数。通信双方每次触发重传机制后，将重传管理数据并更新重传计数器。此机制将避免双方重传次数超过最大重传次数。

在通信管理数据交互过程中，通信管理实体通过请求数据包携带了本方节点等待回复的最大时间，即接收方响应时间，超时将触发重传机制。当接收方不能在规定响应时间内完成回复，可通过 MRA 数据包携带本方服务时间进行二次时间协商。此定时工作机制既保证系统能正常运行，又具备可调节性。定时工作模式如图 3-7 所示。

图 3-7 中：

Remote CM Response Timeout:该字段用于指示客户端规定服务器端响应 REQ 所用的时间。

Remote CM Service Timeout:指示服务器端完成响应所需时间字段

Local CM Response Timeout:用于指示客户端用于响应 REP 所需时间的字段。该字段由客户端用户配置。

Local CM Service Timeout:用于指示客户端完成响应所需时间字段。

t_{min}、t_{max}：管理报文在链路中传输所需最小和最大时间。

需要解释的是，在 B 节点接收到 REQ 后，解析得到 A 节点指示 B 完成 REQ 响应时间字段 Remote_CM_Response_Timeout，B 节点用户判断自身能否在规定时间内回复 REP：如果不能则回复 MRA（REQ），数据包用于确认 REQ 已被接收和传递 Remote_CM_Service_Timeout 字段告知 A 节点自己需要处理 REQ 和回复 REP 的时间，A 接收到 MRA 后将按照此时间重新计时；如果 B 能准时响应 REQ 则不需回复 MRA，A 节点等待时间超过最长时间将使能超时指示信号。同理，B 节点

超时指示信号会在 B 节点等待时间超过最长等待时间后被使能。

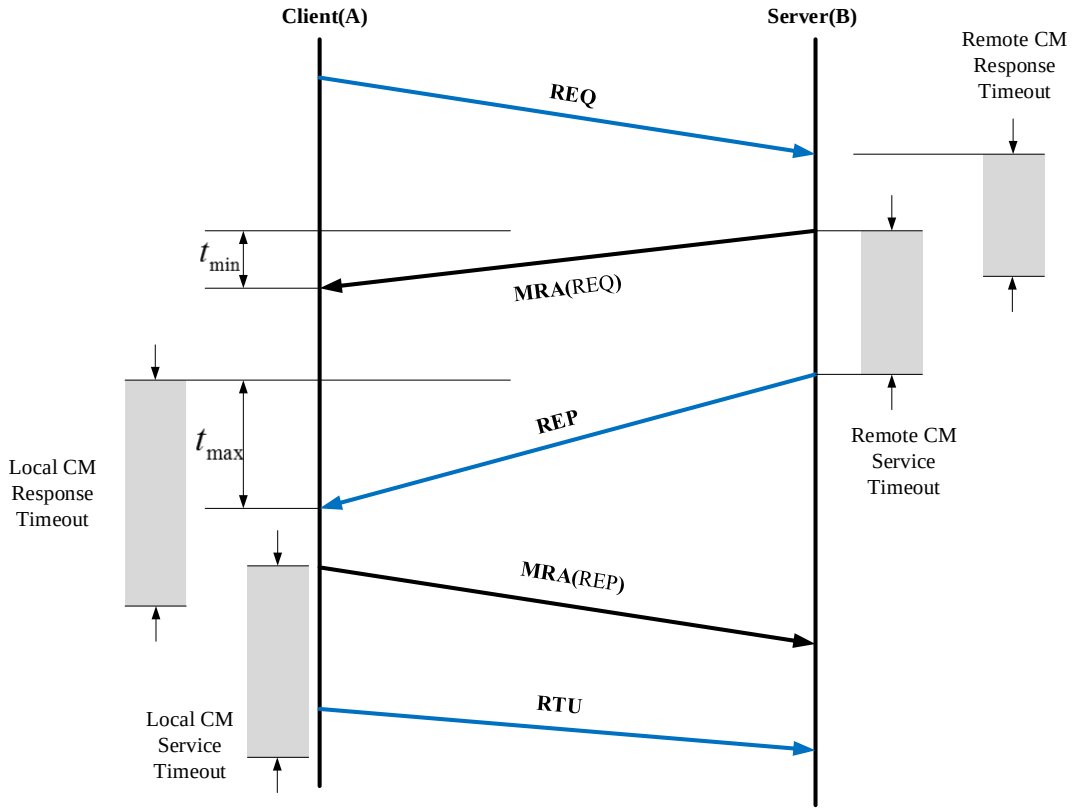


图 3-7 链路建立数据传输定时模式示意图

经过以上分析，我们可以得到 A 节点发送 REQ 后等待 REP 最长时间和 B 节点发送 REP 后等待 RTU 的最长时间。首先，协议规定了数据中各个时间字段数值和实际时间的换算关系为：

$$t = 4.096us \bullet 2^{Timeout} \tag{3-1}$$

其中，t 为实际时间，$Timeout$ 为数据包中各个指示时间字段的值。

所以，Client 方 REQ 发送后超时指示信号实际使能时间为：

如未接收到 MRA：

$$t\_client\_wait = 4.096us \bullet 2^{Remote\_CM\_Response\_Timeout} + 2 \bullet t_{max} \tag{3-2}$$

接收到 MRA 后，计时重启，需等待时间：

$$t\_client\_mra = 4.096us \bullet 2^{Remote\_CM\_Service\_Timeout} + t_{max} \tag{3-3}$$

Server 方 REQ 发送后超时指示信号使能时间为：

未接收到 MRA：

$$t\_server\_wait = 4.096us \bullet 2^{Local\_CM\_Response\_Timeout} + 2 \bullet t_{max} \tag{3-4}$$

接收到 MRA 后：

$$t\_server\_mra = 4.096us \bullet 2^{Local\_CM\_Service\_Timeout} + t_{\max} \quad (3\text{-}5)$$

3.3 IB 接口通信控制器总体设计

在整个接口框图中，通信控制器作为数据通信前通道建立与通信后信道拆除的关键处理单元，为保证系统正常运行，通信控制器需具有以下功能：

（1）通信管理协议的硬件描述和设计：设计完全符合通信管理协议，按照协议规定的通信数据链路建立和终止过程运行。能正确封装和解析运行过程中涉及的各种通信管理数据包，能使通信链路双方 QP 按照图 2-7 所示的规则进行状态的正确转换。

（2）正确处理通信建立请求：建立请求不仅可能来自本地上游用户，而且可能来自远端节点用户，因此要求我们 FPGA 设备能同时作为 InfiniBand 传输中的客户端和服务器端设备进行工作。同时需保存和维护所有连接信息，以及无效连接信息的清除。

（3）收发 GMP：通信控制器能按照 IB 数据通信机制，对运行过程中产生的 GMP 通过 QP1 正确进行发送和接收。

（4）Verb 命令处理：设备能按照 Verb 命令机制，对工作过程中必要的 Verb 命令进行处理，按照 HCA 控制机制和通道适配器进行命令数据交互[39]。

（5）可靠性：为保证通信管理报文的正确收发，以及通信链路的建立顺利，系统需具有重传和超时机制，提高系统的容错率。

（6）吞吐量匹配：本设计采用的通道适配器速率达到 40 Gbps。为充分利用通道适配器速度优势，通信控制器采用 256 bit 数据位宽，运行频率需达到 250 MHz。

在硬件框图设计中，结合实际功能需求以及遵循模块化的设计方法，通信控制器整体设计的框图为图 3-8：

如图 3-8 所示，通信控制器主要包含以下模块：

（1）接入层管理模块

作为接入层的中心模块，负责层次的管理和调度。由路由模块，连接信息管理模块和私有数据缓存模块组成。

在接入层中，除了通信管理单元外，还存在子网管理单元、通用服务管理单元和子网信息查询单元等在为上层用户提供服务，路由模块负责路由上层用户请求到对应服务单元并上报服务单元完成信息。

连接信息管理模块主要负责存储节点 QP 连接信息，包含 QP 连接状态，如处于已连接状态，则还包含链路对方 QPN 和通信 ID 等信息。在链路建立或终止前，

均需先读取本地 QP 连接信息，用以避免连接通道的重复建立和无效连接通道的产生。

私有数据缓存模块负责缓存通信管理过程中通信双方的私有数据。该模块通过两个 FIFO 组成，一个用于存储本地用户下传的私有数据，一个用于存储已接收的远端用户私有数据。

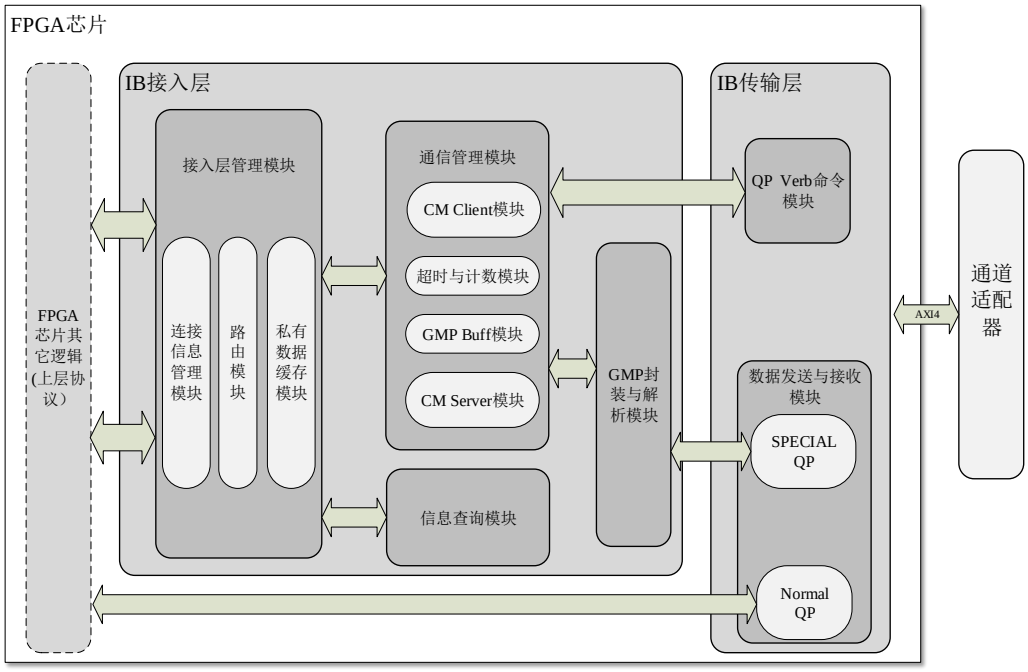


图 3-8 IB 接口通信控制器设计框图

（2）通信管理模块

作为通信管理协议的硬件描述和设计，模块由 CM Client 模块、超时与计数模块、GMP Buff 模块和 CM Server 模块组成。

CM Client 模块是用户作为客户端接收并处理用户通信建立请求的模块。即扮演图 3-4 里 A 节点中 CM 角色。模块负责接收请求，按照握手规则，调用 GMP 封装模块产生待发送管理报文，接收 GMP 解析模块数据。调用 HCR 控制模块，控制通道适配器对 QP 工作状态进行转换。

超时与计数模块和 GMP Buff 模块是通信管理模块重要组成部分。超时与计数模块保证了通信建立可靠性。根据超时重传机制，判断是否有超时现象，并在最大重传次数范围内指导 Server 和 Client 模块进行重传操作。GMP Buff 模块由 FIFO（先入先出）构成，对 GMP 解析模块数据做缓冲处理。防止接收和处理数据速率不匹配。

CM Server 模块是用户作为服务器端接收并处理远端用户通信建立请求的模

块。即扮演图 3-1 里 B 节点中 CM 角色。和 CM Client 模块一样，该模块调用 GMP 产生模块和 HCR 控制模块完成对应工作请求。同时二者还将处理通信链路的终止请求。

（3）GMP 生成与解析模块

GMP 生成模块用于通信建立过程中各种操作类型的通信管理报文的封装。该模块按照协议格式生成首部和数据部分，并将 256 Byte 的数据以 256 bit 的数据流存入缓存模块。生成模块支持 REQ、REP、RTU、REJ、DREQ 和 DREP 六种必要的管理报文封装。

GMP 解析模块的主要功能是实现通信管理报文的解析。虽然所有报文首部格式固定，但不同报文数据部分携带信息不一。模块采用多个解析模块并行设计，同时支持 REQ、REP、REJ、RTU、DREQ 和 DREP 等管理报文的解析，获取其携带的远端节点通信信息。

（4）QP Verb 命令模块

QP Verb 命令模块的主要功能是按照通道适配器命令控制机制实现 Verb 命令的执行与数据交互。在通信控制器中主要涉及到 QP 状态转换相关 Verb 命令的产生和执行。

（5）数据发送与接收模块

数据发送与接收模块负责接收和发送数据，由图 3-3 中数据传输模块、数据发送/接收模块和发送/接收队列组成。按照用于通信的 QP 序列号，管理数据报通过 QP0 和 QP1 两个 Special QP 传输，而业务数据则通过 Normal QP 传输。通信控制器则主要通过 QP1 进行通信管理报文的传输。

3.4 本章小结

本章结合第二章分析，首先根据应用场景，对 IB 接口进行了功能分析，给出了 FPGA+HCA 的接口硬件设计方案，并对接口的结构以及接口的工作流程进行了介绍。其次，本章研究了通信控制器相关的通信管理协议，包括通信管理协议中的通信管理报文格式、链路建立过程、链路拆除过程和超时重传机制。最后，根据通信控制器的功能需求，给出了 IB 接口的通信控制器硬件设计框图。

第四章 通信控制器的 FPGA 实现

在 IB 接口中，通信控制器主要负责节点间数据传输链路的建立和拆除。本章将对通信控制器中部分关键模块进行 FPGA 实现。

4.1 通信管理模块的实现

通信管理模块作为通信控制器的核心模块，控制着整个系统的工作状态和流程。结合 3.3 节对通信管理模块的功能介绍，该模块的内部结构和外部连接如图 4-1 所示。

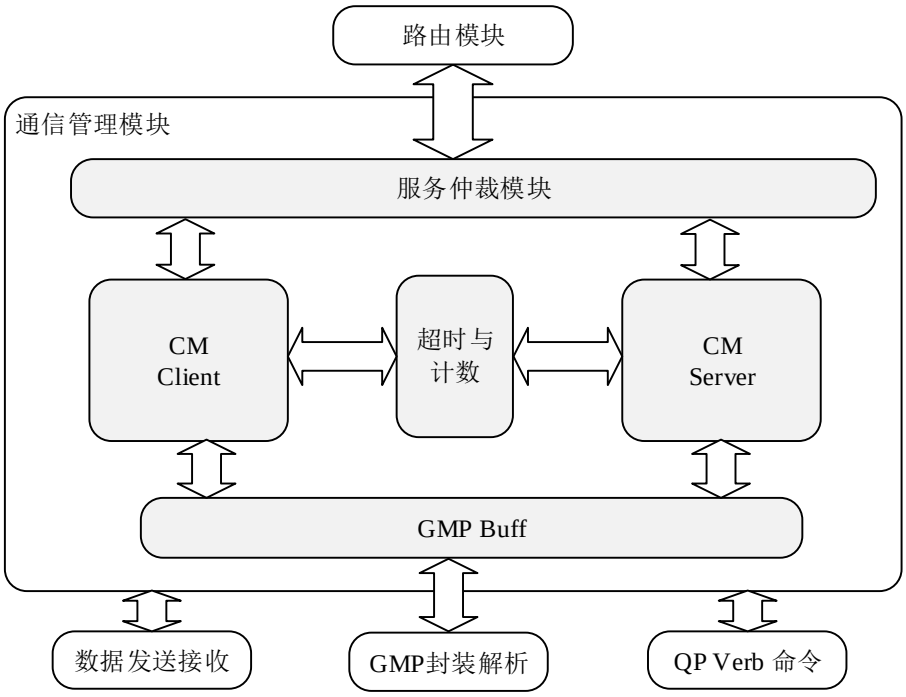


图 4-1 通信管理模块结构框图

通信管理模块内部由服务仲裁模块、CM Client 模块、CM Server 模块、超时与计数模块和 GMP Buff 模块组成。

从图 4-1 可以看出，通信管理模块向上和接入层路由模块进行数据交互，接收用户通信建立或拆除请求以及反馈链路建立或拆除过程中的信息数据，比如私有数据和链路节点信息。向下调用数据发送模块传输 GMP、调用 QP Verb 命令模块进行 QP 状态控制和接收 GMP 解析模块上传的关键数据并进行缓存。

4.1.1 服务仲裁模块

服务仲裁模块用于和接入层管理模块连接，接收用户配置数据和上传数据。主要负责两个功能：

（1）接收上层配置数据。该数据既可能用于本方主动建立数据链路，也可能用于回复远端节点建立请求。因此该模块需根据数据类型指示符号，判断该数据是传递到 CM Client 模块还是 CM Server 模块，正确路由数据到指定模块。

（2）在通信建立过程中 CM Client 模块和 CM Server 模块需将远端节点的交互数据上传，比如通信管理数据包携带的链路信息和私有数据等，用户根据此类数据决定下一步操作行为。在两个模块同时有数据上传请求时，会发生权限仲裁问题，此时服务仲裁模块将按照 CM Client 汇报数据优先级更高原则，优先上传 CM Client 模块的数据。

4.1.2 CM Client 模块

4.1.2.1 模块功能介绍

CM Client 模块是通信控制器处理用户主动发起的通信建立请求的核心模块，其工作行为完全符合 InfiniBand 通信管理协议对客户端通信管理实体的行为要求。该模块的主要功能是：

（1）接收服务仲裁模块路由的用户通信建立请求，按照通信链路建立过程，调用 GMP 封装模块，封装 REQ、RTU 等管理数据包，并向数据发送模块发起数据包传输请求。接收 REP 数据包中关键信息。通信建立完成后将完成信息上传。

（2）接收服务仲裁模块路由的用户通信拆解请求，按照通信链路终止过程，调用 GMP 封装模块，封装 DREQ 数据包，并向数据发送模块发起数据包传输请求。接收 DREP 数据包中关键信息。将已拆除的连接通道信息上传。

（3）在传输通道建立或拆除过程中，调用 HCR 命令模块，转换传输通道对应 QP 的状态。

（4）处理同为 Client 的对等通信双方之间均发送 REQ 数据包的情景，按照协议相关规定进行角色转换。

（5）根据超时重传机制，对满足要求的数据包进行重传。

该模块在设计中通过有限状态机实现，接收传输层模块的服务，完成上述功能。

4.1.2.2 CM Client 状态机

CM Client 模块工作运行状态转移图如图 4-2 所示，总共 9 个状态，从图中可

以直观明显的看出各个状态跳转条件和下一状态。

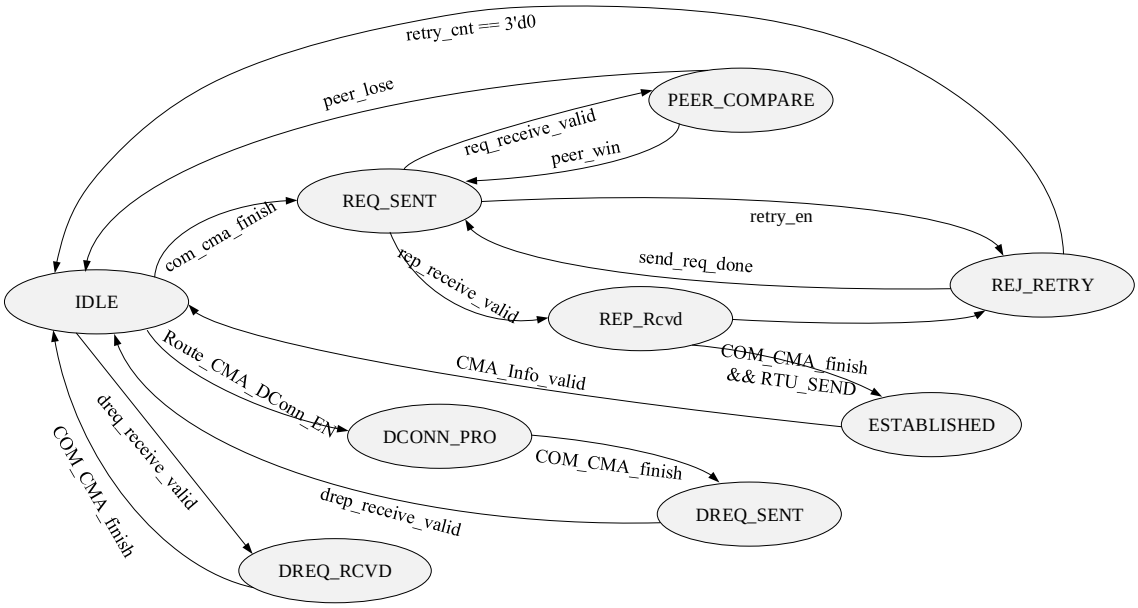


图 4-2 CM Client 工作状态转移图

各个状态功能、跳转条件以及状态变化如下所示：

（1）IDLE：初始状态

功能描述：系统上电复位后将自动进入初始状态。当路由模块传来通信建立请求时，组织用于封装 REQ 所必需的数据，向 GMP 封装模块发起 REQ 封装请求。REQ 封装完成存入对应存储空间后，向数据发送模块传递 REQ 包发送请求。当系统接收到 DREP 数据包时，将删除数据包指定数据链路信息。

跳转条件：REQ 发送完成指示信号有效时，跳入 REQ_SENT 状态；用户链路拆除请求指示信号有效时则进入 DCONN_PRO 状态；接收到通信对方节点 DREQ 时进入 DREQ_RCVD；均无效时则保持 IDLE 状态。

次态：IDLE，REQ_SENT，DCONN_PRO，DREQ_RCVD。

（2）REQ_SENT：REQ 已发送状态

功能描述：通信链路本地端口的 QP 状态转换为初始化状态，转换通过发送 RST2INIT_QP 命令完成。启动超时与计数模块，记录此次连接过程等待时间和已发送 REQ 次数。

跳转条件：REP 接收完成指示信号有效，则进入 REP_RCVD 状态；REQ 接收完成指示信号有效，则进入 PEER_COMPARE 状态；超时信号有效，则进入 REJ_RETRY 状态；以上信号均无效时保持 REQ_SENT 状态。

次态：REQ_SENT、REP_RCVD、PEER_COMPARE、REJ_RETRY。

（3）PEER_COMPARE：对等请求比较状态

功能描述：在本方节点发送 REQ 期间，接收到对方节点链路建立请求时，将比较 REQ 中携带的 GUID 和本地节点 GUID 大小，更新结果标志位寄存器。

跳转条件：结果标志位信号指示本地节点继续作为客户端则转为 REQ_SENT 状态；指示本地退为服务端，则转为 IDLE 状态。

次态：IDLE、REQ_SENT

（4）REP_RCVD：已接收到 REP 状态

功能描述：验证 REP 数据部分 Remote_COMID 字段，与本地信息一致则向 GMP 封装模块发起 RTU 封装请求，封装结束后向数据发送模块使能 RTU 发送请求。通过 INIT2RTR_QP 命令转换 QP 至可接收状态;字段信息与本地信息不一致，则向 GMP 封装模块发起 REJ 封装请求，请求中包含拒绝原因。封装结束后则向 COM 模块传递发送 REJ 请求。

跳转条件：RTU 发送完成指示信号有效时，跳入 ESTABLISHED 状态；REJ 发送完成指示信号有效时，跳入 REJ_RETRY 状态；均无效时保持此状态。

次态：ESTABLISHED、REJ_RETRY、REP_RCVD。

（5）REJ_RETRY：重传状态

功能描述：重传次数低于最大重传次数时，重新向数据发送模块发起传输请求。重传次数等于最大重传次数时，放弃此次请求，并汇报信息。

跳转条件：REQ 发送完成指示信号有效时，跳入 REQ_SENT 状态；重传次数耗尽则进入 IDLE 状态；否则保持此状态。

次态：IDLE、REQ_SENT、REJ_RETRY。

（6）ESTABLISHED：建立完成状态

功能描述：将已建立链路信息写入连接信息管理模块。通过 RTR2RTS_QP 命令将 QP 转换为可发送状态。

跳转条件：命令执行完成信号有效则跳入 IDLE 状态；否则保持此状态。

次态：IDLE、ESTABLISHED。

（7）DCONN_PRO：链路拆除请求处理状态

功能描述：处理用户链路拆除请求。向远方节点发送 DREQ。

跳转条件：DREQ 发送完成指示信号有效时，跳入 DREQ_SENT 状态；否则保持此状态。

次态：DCONN_PRO、DREQ_SENT。

（8）DREQ_SENT：DREQ 已发送状态

功能描述：通过 2ERR_QP 命令将本地 QP 转为错误状态。启动超时与计数模

块，进行定时。

跳转条件：DREP 接收完成指示信号有效，则进入 IDLE 状态；等待回复超时进入 IDLE 状态。否则保持此状态

次态：IDLE、DREQ_SENT。

（9）DREQ_RCVD：DREQ 已接收状态

功能描述：向对方节点回复 DREP，并通过 2ERR_QP 命令将本地 QP 转为错误状态。

跳转条件：DREP 发送完成指示信号有效则进入 IDLE 状态，否则保持此状态。

次态：IDLE、DREQ_RCVD。

4.1.3 CM Server 模块

4.1.3.1 模块功能介绍

对于以主机通道适配器为网络接口的用户来说，不仅能主动与远端节点来建立数据传输通道，还能作为服务器节点响应远端节点发起的通信建立请求。CM Server 模块作为系统中回应通信建立与拆除请求的处理单元，主要完成以下功能：

（1）接收远端通信节点的通信建立 REQ 数据包，启动通信建立响应流程，调用 GMP 封装模块，封装 REP 数据包，并向数据发送模块发起数据传输请求。接收 REP 数据包中关键信息。通信建立完成后将链路信息存入连接信息管理模块。

（2）和 CM Client 模块类似，处理通信链路双方任意一方发起的链路拆除请求。

（3）根据接收的数据包转换数据链路本地 QP 的状态。

4.1.3.2 CM Server 状态机

CM Server 工作运行状态同样通过状态机来进行控制，CM Server 状态机如图 4-3 所示，总共 9 个状态。

各个状态功能、跳转条件以及状态变化如下所示：

（1）LISENT：监听状态

功能描述：系统上电复位后将自动进入监听状态。当监听到远端用户发送的 REQ 时，将上传 REQ 包中关键信息和私有数据，同时通过发送 RST2INIT_QP 命令将 QP 转换为初始化状态；当监听到链路拆除请求时，封装并发送 DREQ 数据包。当监听到远端用户发送的 DREQ 时，回复 DREP。

跳转条件：QP 状态转换命令完成指示信号有效则转为 REQ_RCVD 状态，链路拆除请求有效时，跳为 DCONN_PRO 状态。DREQ 接收完成指示信号有效时跳

转到 DREQ_RCVD 状态。均无效时则保持监听状态。

次态：IDLE，REQ_RCVD，DCONN_PRO，DREQ_RCVD。

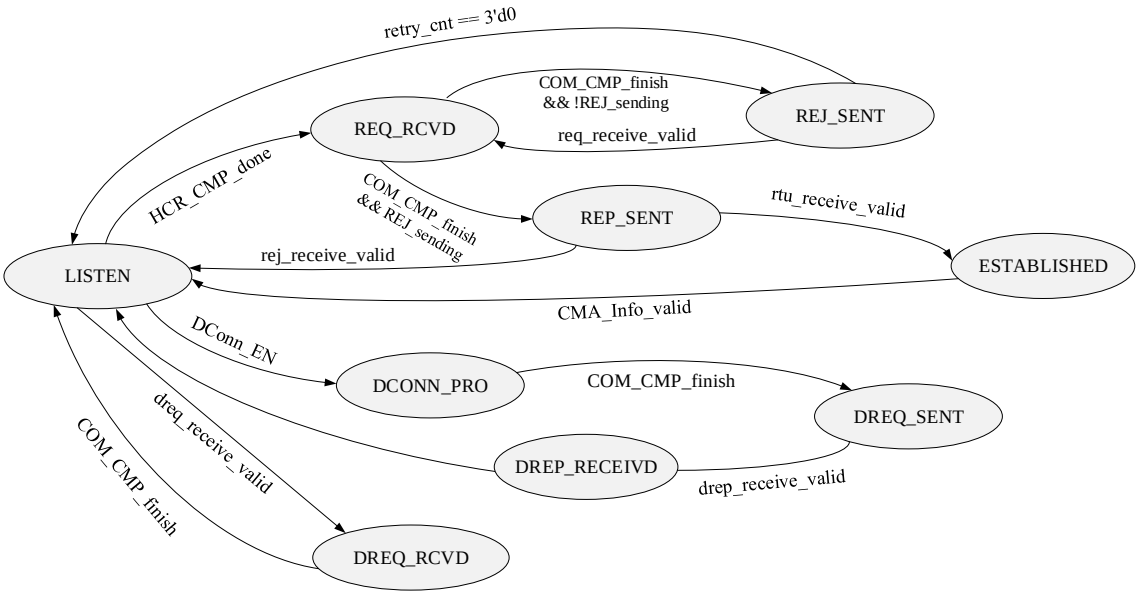


图 4-3 CM Server 工作状态转移图

（2）REQ_RCVD：REQ 已接收状态

功能描述：用户根据 REQ 携带的数据，做出是否建立连接决定。当应答信号指示同意此次连接请求则封装并发送 REP。然后通过 INIT2RTR_QP 命令将通信链路本地 QP 转换为可接收状态。应答信号指示拒绝此次连接请求则封装并发送 REJ，数据包携带有拒绝原因，然后通过 2ERROR_QP 命令将 QP 转为错误状态。

跳转条件：REP 发送完成指示信号有效，则进入 REP_SENT 状态。REJ 发送完成指示信号有效时，则进入 REJ_SENT 状态。否则就保持 REQ_RCVD 状态。

次态：REQ_Rcvd、REP_Sent、REJ_Sent。

（3）REJ_SENT：REJ 已发送状态

功能描述：更新本地已重传次数。

跳转条件：指示重传次数耗尽信号有效进入 LISTEN 状态。重传次数未耗尽且重新接收到 REQ，则进入 REQ_RCVD 状态。否则保持此状态。

次态：REJ_SENT、LISTEN、REQ_RCVD。

（4）REP_SENT：REP 已发送状态

功能描述：启动计时单元，超时则更新超时指示信号。

跳转条件：超时指示信号有效前接收到 RTU 数据包则进入 ESTABLISHED 状态；接收到 REJ 或超时指示有效则进入监听状态。

次态：ESTABLISHED、LISTEN。

（5）ESTABLISHED：建立完成状态

功能描述：向链路信息管理模块写入已建立链路信息，并通过 RTR2RTS_QP 命令将 QP 转换为可发送状态。

跳转条件：链路信息上传完成则跳入 IDLE 状态。

次态：IDLE、ESTABLISHED。

（6）DCONN_PRO：链路拆除请求处理状态

功能描述：用于处理用户链路拆除请求。封装和发送 DREQ。

跳转条件：DREQ 发送完成指示信号有效时，跳入 DREQ_SENT 状态。否则保持此状态。

次态：DCONN_PRO、DREQ_SENT。

（7）DREQ_SENT：DREQ 已发送状态

功能描述：通过 2ERR_QP 命令将本地 QP 转为错误状态。调用超时与计数模块进行定时。

跳转条件：接收到对方节点回复的 DREP，则进入 LISTEN 状态。等待回复超时进入 LISTEN 状态。否则保持此状态

次态：LISTEN、DREQ_SENT。

（8）DREQ_RCVD：DREQ 已接收状态

功能描述：向对方节点回复 DREP，通过 2ERR_QP 命令将本地 QP 转为错误状态。

跳转条件：DREP 发送完成指示信号有效则进入 LISTEN 状态；否则保持此状态。

次态：LISTEN、DREQ_RCVD。

（9）DREP_RCVD：DREP 已接收状态

功能描述：通过 2ERR_QP 命令将本地 QP 转为错误状态。

跳转条件：命令执行完成指示信号有效则进入 LISTEN 状态；否则保持此状态。

次态：LISTEN、DREP_RCVD。

4.1.4 超时与计数模块

在小节 3.2.5 中，本文通过分析超时重传机制得到了通信控制器运行过程中，通信管理实体对远端节点回复的等待时间。本模块将按照式(3-1)到(3-5)给出的各种情景下超时信息为 CM Client 和 CM Server 做定时服务，在超过规定时间时给出

指示信号。通过公式(3-1)到(3-5)可以看出，最大等待时间由两部分组成，一部分为基于 4.096 us 的多重计时；一部分为数据在链路传输时间。因此在设计中我们将以一个 4.096 us 的计时器作为基础计时器，并通过输入 Timeout 字段值重复计时，达到计时可配置目的。最后再通过单独的一个计时单元来对链路传输时间定时。图 4-4 为系统定时模块。

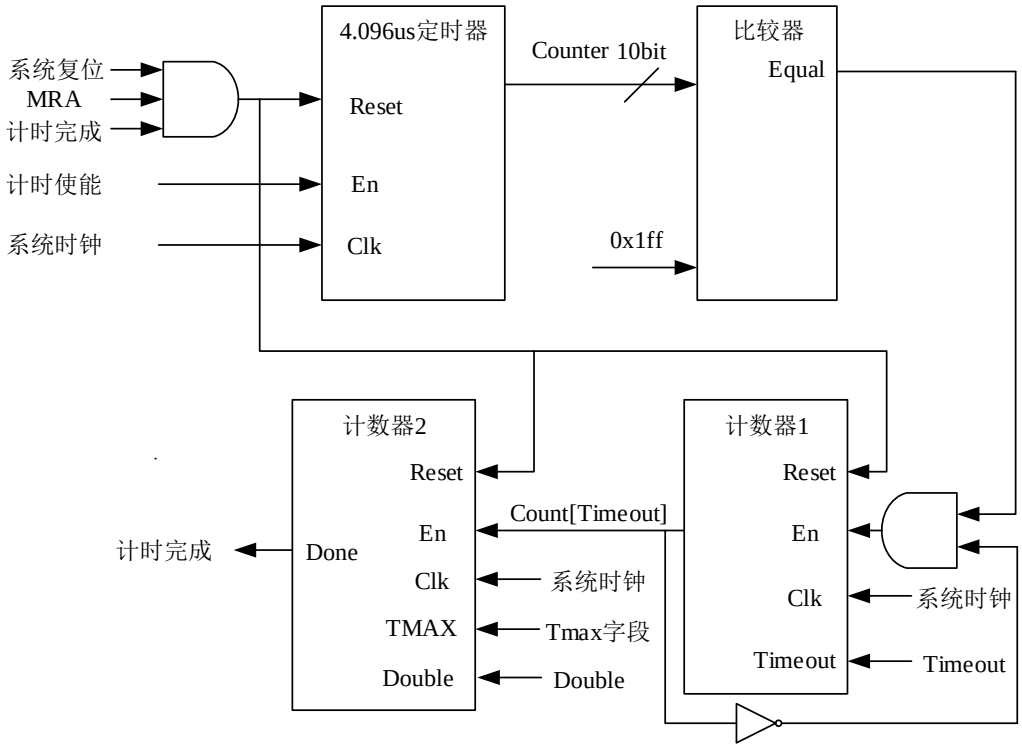


图 4-4 系统定时模块结构图

本设计系统时钟频率为 250 MHz，时钟周期为 4 ns。在进行 4.096 us 定时器设计时，计数寄存器数据位宽为 10。当 counter=10’d1023 时，比较器会输出一个脉冲信号，信号有效间隔为 4.096 us。计数器 1 在比较器输出有效时进行计数加一操作。每自加一意味时间经过了 4.069 us。当 count[Timeout]位为 1 时表示时间已经过 $4.096 \cdot 2^{Timeout}$ us。此时计数器 1 将不再继续计数。启动计数器 2，并通过 Double 指示信号判断计时时间为 T_{max} 还是 $2 \cdot T_{max}$。计数器 2 计时完成后将使能超时表示符，定时结束。

在运行过程中，未超时阶段接收到 MRA，系统将对计数器做一次清零处理，后续重新以式(3-3)或式(3-5)为超时判断时间点。同时将 Timeout 输入字段更新为 MRA 中接收到的服务时间字段。

4.2 GMP 封装与解析模块的实现

4.2.1 GMP 封装模块

4.2.1.1 设计概览

GMP 封装模块主要功能是对 CM Client 模块和 CM Server 模块通信管理数据包封装请求进行处理。无论待封装的 GMP 是用于通信建立握手包（REQ、REP、REJ、RTU、MRA）还是用于链路拆除的握手包（DREQ、DREP），都要根据通信管理模块的配置数据来对 GMP 进行封装使之符合通信管理报文格式规范。

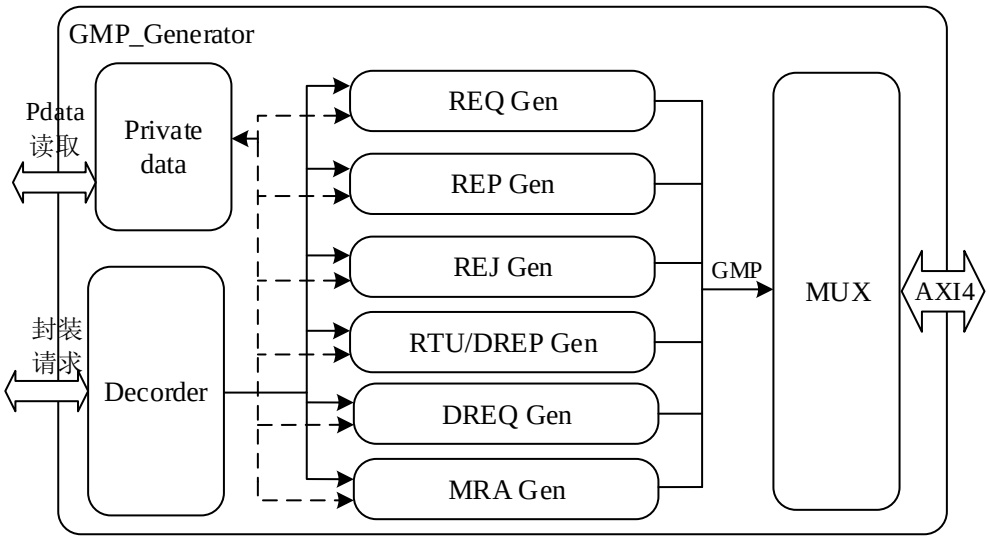


图 4-5 GMP 封装模块结构框图

图 4-5 为 GMP 封装模块内部结构图，其中，Decorder 模块负责接收来自通信管理模块的 GMP 封装请求，并通过对 CMA_Gen_attrID 信号进行译码，判断待封装 GMP 类型，使能对应封装模块工作请求信号。接收各个模块封装完成信号，向通信管理模块汇报完成情况。

Private Data 模块负责读取用户通过该 GMP 携带的私有数据。由 3.2.1 小节介绍可知，私有数据是 GMP 包数据部分的重要组成块。设计中采用深度为 1024 的 FIFO 存储用户配置的私有数据，并接受各个封装模块对私有数据的提取。因为数据采用双字（Double Word，DW）对齐，所以 FIFO 数据位宽为 32 bit。

REQ Gen、REP Gen、REJ Gen、RTU/DREP Gen、DREQ Gen 和 MRA Gen 模块则是分别用于封装 REQ、REP、REJ、RTU、DREQ、DREP 和 MRA。其中由于 RTU 和 DREP 的数据部分字段完全一样，因此共用一个封装模块。

在设计中，通过 3.2.1 小节的介绍，我们已经知道各种数据包由首部和数据两

部分组成，而不同数据包的有效数据部分无论长度和组成字段均不一样。如图 4-6 所示，GMP 首部长度固定为 24 Bytes，数据部分为 232 Bytes。数据部分由有效数据和私有数据组成，有效数据和私有数据长度不一，且有效数据的组成字段不一，所以在设计中，GMP 封装模块需要多个模块用以支持七种数据包封装。单个封装模块在工作请求指示信号有效时将启动封装，封装过程在后面将详细介绍。因为 IB 接口整体采用 256 bit 数据位宽设计，因此封装完成的长度为 256 Byte GMP 将存入位宽为 256 bit 的存储空间中，每个数据包占用 8 个存储单元。

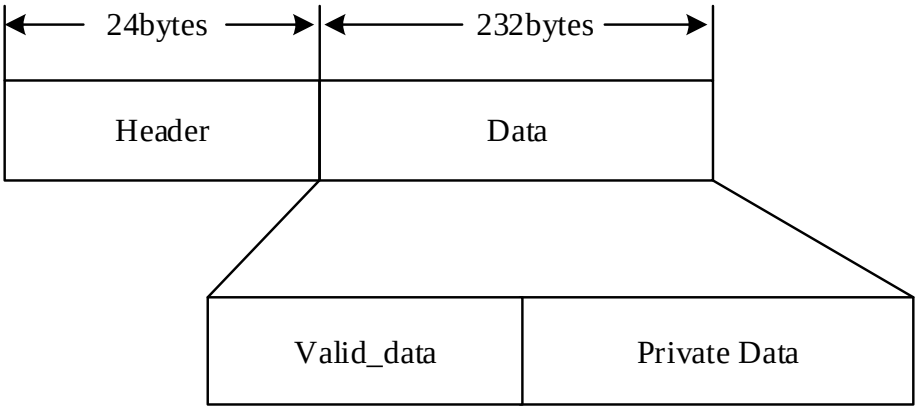


图 4-6 GMP 组成部分与数据长度

为了方便后续 GMP 发送，系统划分了 7 个存储空间，分别用于存储七种数据包。MUX 模块用于接收 6 个封装模块的输出数据，判断数据包类型，存入系统为此类型数据包分配的存储空间。

4.2.1.2 REQ_Gen 模块

六个封装模块工作模式和流程类似，功能均包含构造 GMP 首部，生成有效数据段，读取私有数据块和数据存储。本节将以 REQ 封装为例，介绍 REQ_Gen 模块。图 4-7 为 REQ_Gen 内部模块。

REQ_Head_data 模块负责接收 REQ 封装请求并产生 REQ 首部和有效数据，其中 REQ 有效数据长度为 140 Bytes，需占用 35 个 32 bit 数据位宽 RAM 的存储地址空间。该模块采用的是有 11 个状态的有限状态机完成，每个状态负责一个字段的写入并且可以通过添加或减少状态灵活配置有效数据字段。因为有效数据长度较长，可配置字段较多，本文主要根据需求，对表 3-4 中字段进行了配置。

REQ_Reassemble 模块主要有以下功能：（1）接收 REQ_Head_data 模块产生的 REQ 首部数据。（2）读取有效数据 RAM 中的有效数据部分，共 35 个数据。（3）读取私有数据，共 23 个 32 bit 数据。（4）按照图 4-6 格式进行 REQ 封装，以数据

位宽为 256 bit 存入 REQ 存储空间。

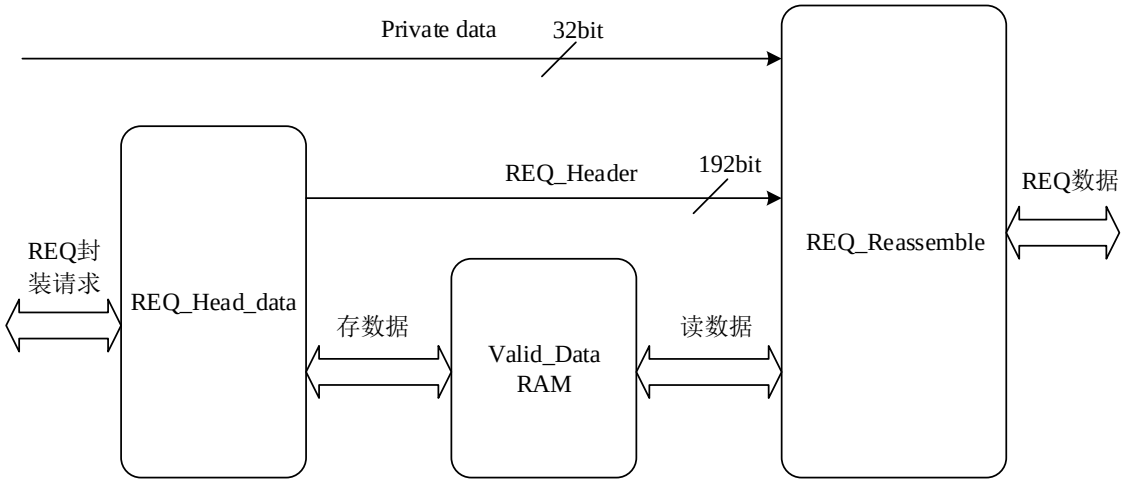


图 4-7 REQ 封装模块结构框图

最终，REQ 的八个 256 bit 数据组成部分如图 4-8 所示。

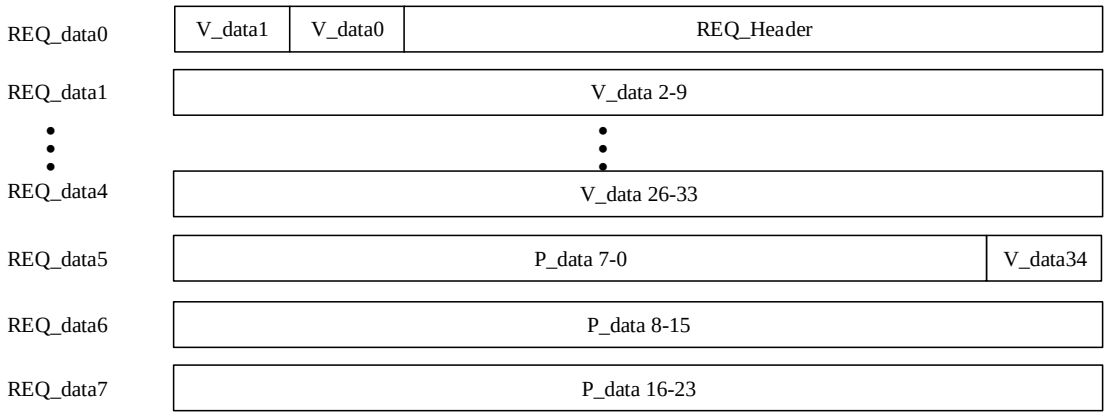


图 4-8 REQ 八个 256bit 数据组成部分

图 4-8 中 REQ_data（n）为组成 REQ 的第 n 个 256bit 数据，V_data（n）为第 n 个 32 bit 有效数据，REQ_Header 为 REQ 首部数据，P_data（n）为第 n 个 32 bit 私有数据。

4.2.2 GMP 解析模块

作为 GMP 封装模块工作的逆过程，GMP 解析模块主要负责对经过 QP1 接收的 GMP 包进行解析，获取其携带的关键信息。当数据包首部字段指示该 GMP 为通信管理数据包时，数据将被路由到 CM_Client 模块或 CM_Server 模块。图 4-9 为 GMP 解析模块的内部框图。

解析控制模块功能为接收完成队列轮询信息、读取接收存储空间数据、调用首部验证模块以及根据首部验证模块信息调用对应 GMP 包解析模块。

首部验证模块将对首部数据进行解析，获得首部关键字段，并指示控制模块是否进行 GMP 数据部分解析。

六个解析模块接收控制模块解析指令，对接收 GMP 数据部分进行解析。

MUX 模块主要负责上传解析模块获得的关键数据，包含首部指示包属性字段、GMP 有效数据和私有数据。

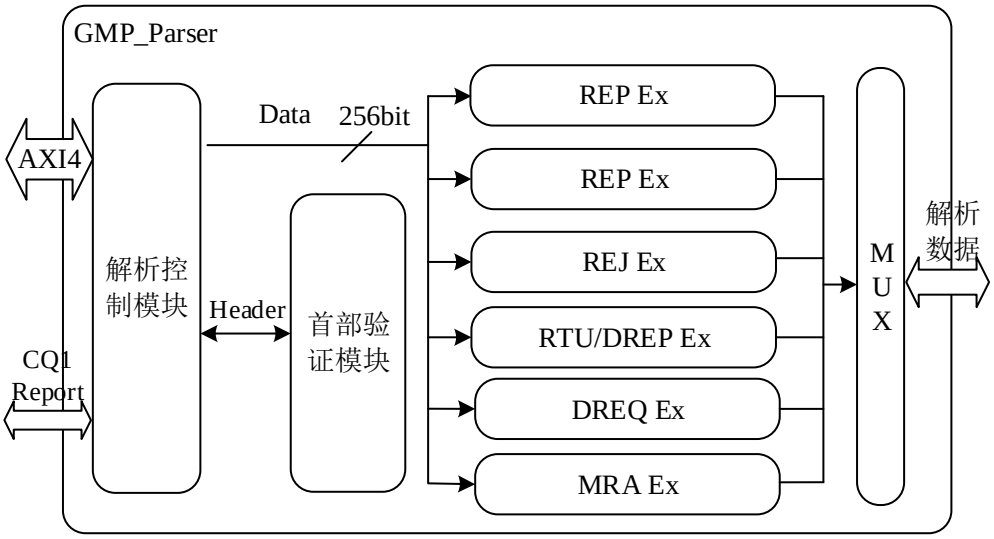


图 4-9 GMP 解析模块结构框图

GMP 解析模块工作流程如图 4-10 所示。

（1）解析控制模块接收 CQ1 上传的接收数据相关信息，如果信息指示接收数据长度为 256 Bytes 且数据传输方式为 01010（Send 传输方式），则进行下一步解析操作，否则接收数据将不被处理。

（2）读取接收数据缓存模块首地址数据，该数据为 256 bit，从图 4-8 可以得到，该数据由首部和前 64 位有效数据组成。首部数据将会被首部验证模块处理，64 位有效数据将会被寄存器缓存。

（3）首部验证模块将对 GMP 首部进行解析，检查 BaseVersion 字段版本信息是否被本地支持，检查 MgmtClass 字段是否是 0x07（通信管理集），Method 字段是否为 0x03（Send 操作）。以上信息均匹配，则将接收数据包 Attribute ID 字段发往解析控制模块。

（4）解析控制模块在接收到首部验证模块发送的信息后，继续读取余下 GMP 数据，根据 Attribute ID 字段信息调用对应 GMP Ex 模块对余下数据进行解析。解析模块将按照封装模块封装过程的逆过程，获取其中关键字段信息并通过寄存器

缓存。

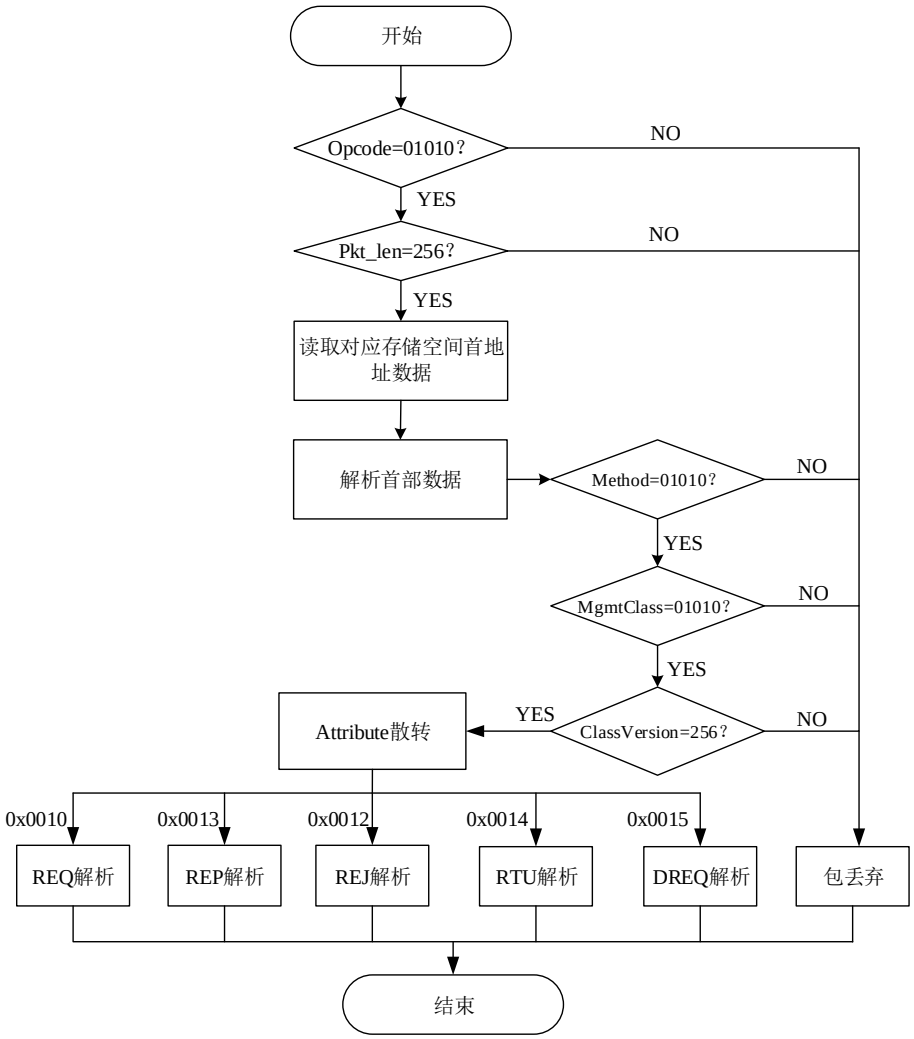


图 4-10 GMP 解析流程图

（5）最后对私有数据进行提取和缓存。由于接收数据包以 256 位数据位宽存入接收缓存空间，而提供给用户读取的数据位宽为 32 位，因此还需要对数据进行位宽变换处理。同时由于每种报文的私有数据长度和私有数据在 256 位数据中的起始位置不一，因此需要做额外的处理。以 REQ 数据为例，从图 4-8 可以看出，REQ 数据的私有数据从第五个数据的第二个 DW 数据开始，私有数据总长度为 23 DW。对于所有数据包，私有数据处理方式为：首部处理过程中，判断数据包类别，通过两个偏移数据指示该种报文私有数据在整个数据中的起始位置，然后读取包含私有数据的报文片段，并从偏移数据指示位置起，提取每一个 32 位数据并依此存入 FIFO 中。

4.3 数据发送与接收模块的实现

4.3.1 数据发送模块

4.3.1.1 设计概览

数据发送模块主要按照 InfiniBand 通信机制，将通信管理模块缓存在发送缓冲区的数据发往目的地址。经过 2.3 小节的介绍，GMP 主要通过 QP1 进行发送，但是在数据发送过程中还涉及到 QP0，QP1 和普通 QP 之间通信优先级的问题，因此设计中需包含多个 QP 工作处理过程。本文在设计中采用了 8 QP 设计，QP0 和 QP1 为特殊 QP，QP2 到 QP7 为普通 QP，数据发送模块的整体设计如图 4-11 所示。

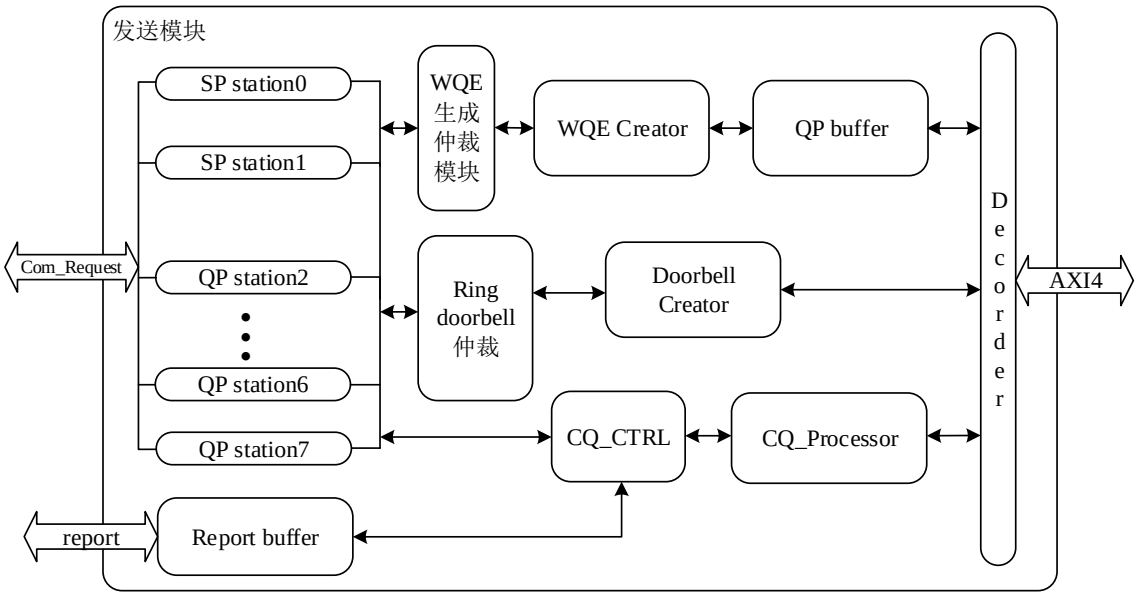


图 4-11 GMP 发送模块结构框图

其中 SP station 模块是特殊 QP0 和 QP1 的工作状态控制模块，QP station2 至 QP station7 模块是普通 QP 的工作状态控制模块。按照协议规定，所有 QP 都是彼此独立的，各个 QP 之间工作状态不会互相干扰。

WQE 生成仲裁模块和 Ring Doorbell 仲裁模块将按照调度策略实现对多 QP 通道对 WQE Creator 和 Doorbell Creator 的使用仲裁。保证系统工作的高效性。

WQE Creator 模块和 Doorbell Creator 模块分别用于生成 WQE 和 Doorbell。通过前面章节介绍可知，WQE 是队列对的基本单元，携带有此次通信操作的指示信息。Doorbell 则是知晓 HCA 队列对中有待处理单元的门铃信息，通道适配器接收到门铃信息后，将依次读取 WQE，并执行 WQE 所指定的数据通信行为。

CQ_Ctrl 模块负责接收 QP 指定的轮询信息，对通道适配器回馈的完成信息进

行轮询并将结果存入 Report buffer，等待用户查询。

CQ_Processor 模块负责对接收的 CQE 进行解析，并将解析结果汇报给绑定到该 CQ 的 QP。

Decorder 模块按照系统模块地址分配规则对读或者写地址进行译码，正确路由读写请求到对应存储空间。

4.3.1.2 QP Station 模块

按照各个 QP 独立的原则，设计中不论是特殊 QP 还是普通 QP，工作状态均由 QP_station 模块状态机控制。该状态机由五个状态组成，其状态转移如图 4-12 所示。

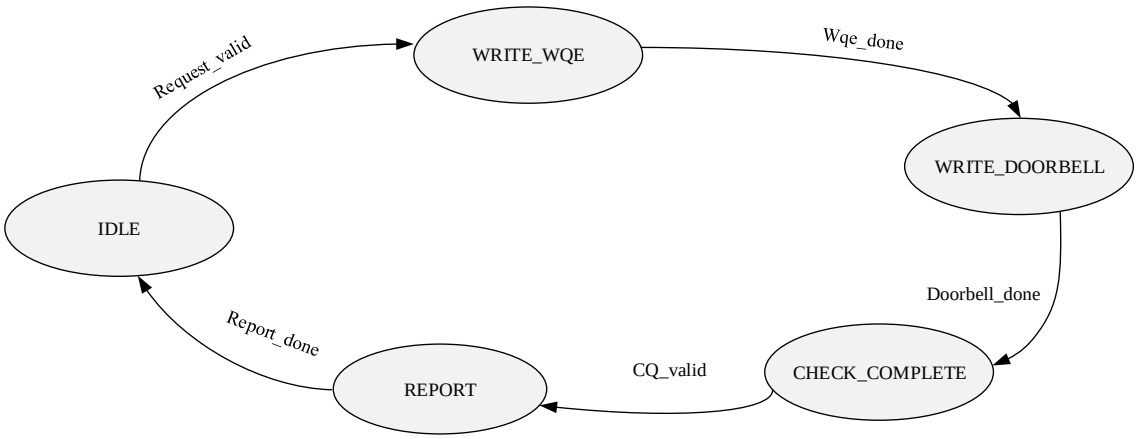


图 4-12 QP_station 状态转换图

IDLE:空闲状态，系统上电复位后将进入该状态。当通信请求信号有效时，将进入 WRITE_WQE 状态。

WRITE_WQE:写 WQE 状态，根据通信请求操作类型，向 WQE 产生仲裁模块发起形成 WQE 的请求。指示 WQE 产生完成信息有效时，将进入 WRITE_DOORBELL 状态。

WRITE_DOORBELL:写门铃状态，根据队列对信息，向 Doorbell Creator 模块发起形成门铃的请求。指示门铃产生完成并已发送至通道适配器信息有效时，将进入 CHECK_COMPLETE 状态。

CHECK_COMPLETE:检查完成状态，发送门铃后，通道适配器读取 WQE 并执行指定操作后，将通过 CQE 反馈完成情况，对于每一个 QP，将通过指定轮询绑定的完成队列获得完成信息。接收到轮询汇报信息后，将进入 REPORT_TIME 状态。

REPORT_TIME:汇报状态，回复用户此次通信请求的完成情况，并进入空闲状态，等待新的通信请求。

4.3.1.3 仲裁模块

仲裁模块分为 WQE 生成仲裁模块和 Ring Doorbell 仲裁模块，前者用于 8 个 QP 中有多个通道同时有写 WQE 请求的情景，后者用于 8 个 QP 中有多个通道同时有 Ring Doorbell 请求的情景。二者均采用相同的调度策略。调度策略是：

- QP0 的请求优先级最高，QP1 的优先级次之，普通 QP 的优先级最低。
- 6 个普通 QP 的优先级一致，普通 QP 的请求将会按照先来先服务（FCFS）算法按照请求时间顺序依次执行。
- 经过仲裁，某请求获得服务后，只有该服务结束完成后才会进行新一轮的仲裁。即在为普通 QP 某请求服务过程中，即使高优先级 QP 有请求，也将等到普通 QP 请求被完成后才会得到服务

图 4-13 为一次仲裁调度示例。

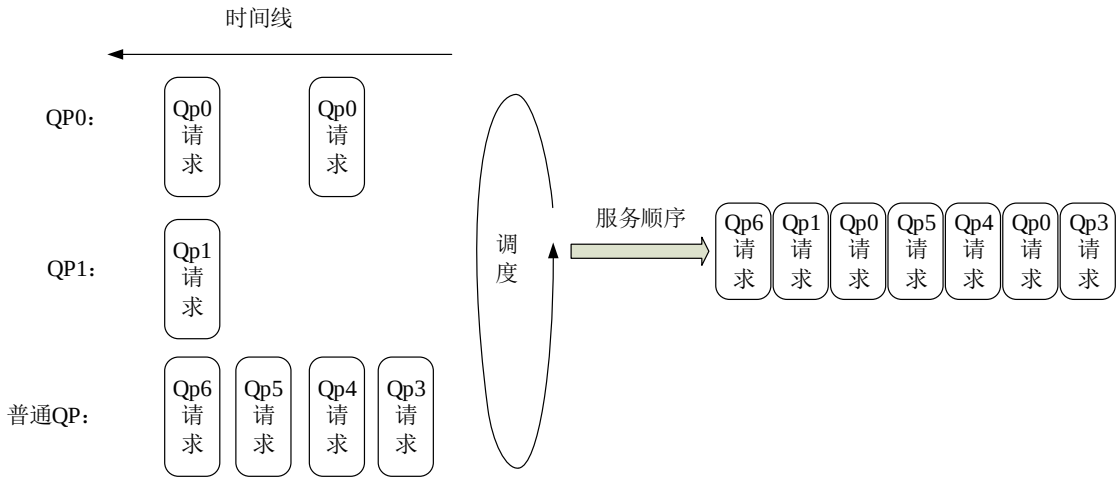


图 4-13 仲裁调度示意图

在实际运行过程中，下一次的仲裁发生于上一次的工作完成后，因此图 4-13 只是给出了理想状态的仲裁，实际中仲裁结果根据上一次服务时间而有所不同，但是策略不变。在硬件代码设计中，通过设置指示正在工作的标志位，防止因高优先级 QP 的工作请求导致低优先级通道服务中断。整个接口系统工作过程中，QP0 和 QP1 大部分工作在普通 QP 工作之前，普通 QP 工作时候 QP0 和 QP1 的请求次数较少，这样的策略既保证了特殊 QP 的优先服务，又避免了低优先级的请求被“饿死”的现象。

对于普通 QP 间的仲裁，设计遵循普通 QP 相互平等独立的原则，采用了先来

先服务（FCFS）算法，该算法简单易于实现。图 4-14 为普通 QP 调度模型。

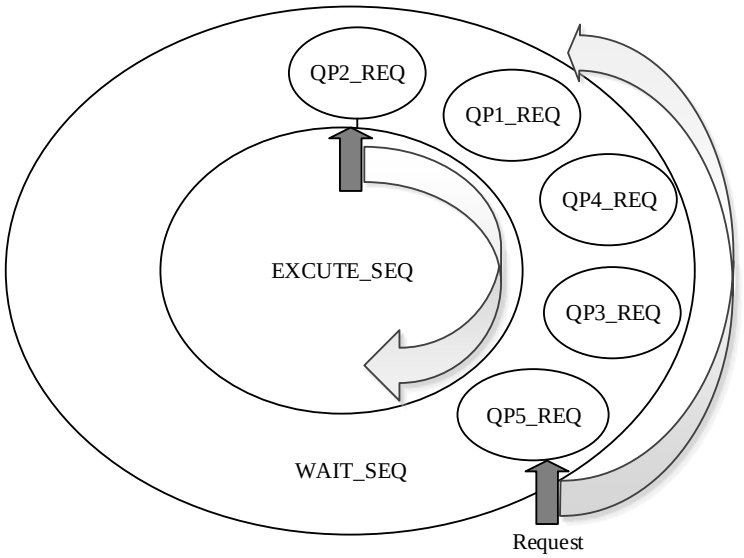


图 4-14 普通 QP 调度模型

图 4-14 中 EXCUTE_SEQ 为执行序列计数器，WAIT_SEQ 为等待序列计数器。当普通 QP 有仲裁请求时，仲裁模块会按照请求先后顺序为每一个普通 QP 按照 1-6 的顺序编号，并往复循环。执行顺序也将按照 1-7 的顺序执行。在更高优先级队列没有请求且编号数和执行序列计数器一致的时候，持有该编号数的 QP 请求将会被服务。

4.3.1.4 WQE Creator 模块

该模块主要为通过仲裁的 WQE 生成请求进行服务，包括 WQE 的产生和存储。通过 2.3 小节的介绍可以知道，WQE 是 IB 协议中引起数据传输的基本单元。用于传输 GMP 数据的 WQE 格式如图 4-15 所示。

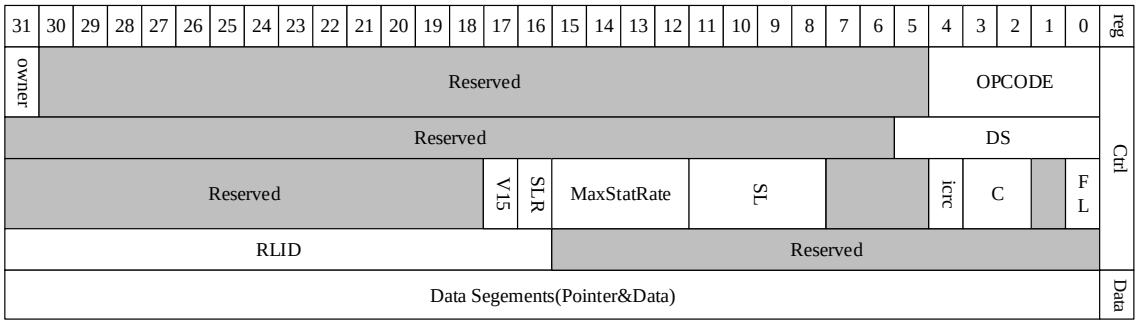


图 4-15 GMP 发送 WQE 格式

其中部分关键字段为：

（1）OPCODE：指示操作类型，在 GMP 传输中，值为 01010。

（2）RLID：远端节点在网络中的地址信息。

（3）MAXStatRate：最大传输速率，由通道适配器决定。

用于指示待传输数据大小和位置信息的 Data Segements 格式如图 4-16 所示。

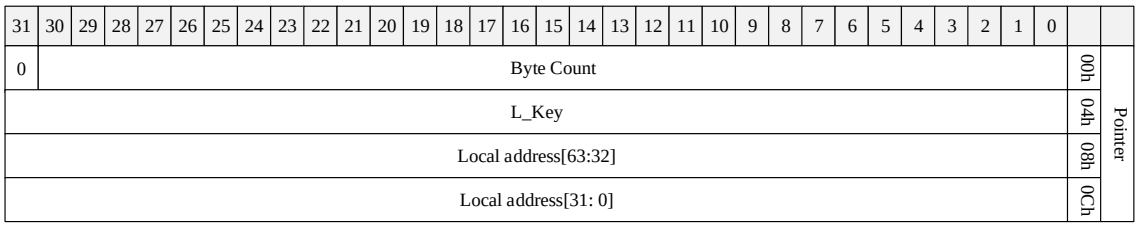


图 4-16 发送 WQE Data 部分格式

其中 Byte Count 字段为传输数据长度，GMP 为 256 Bytes。L_key 为用户为本地存储空间设置的接入权限，通道适配器凭该字段读取本地数据。Local address 为待发送数据起始地址，在 GMP 发送过程中由 GMP 类型决定，系统为每种 GMP 数据分配了存储空间。

WQE Creator 模块将接收用户相关配置数据信息，并按照图 4-15 和图 4-16 给出的 WQE 格式生成 WQE 并存放在发送或接收队列中。

4.3.1.5 Doorbell Creator 模块

和 WQE 产生模块类似，该模块用于 Doorbell 信息的生成，并将 Doorbell 数据写入通道适配器中对应的用户接入区域。用户接入区域的地址在初始化中被映射到了系统地址空间，用户接入区域中用于 QP 进行 ring doorbell（写门铃信息）动作的寄存器格式和偏移量如图 4-17 所示。

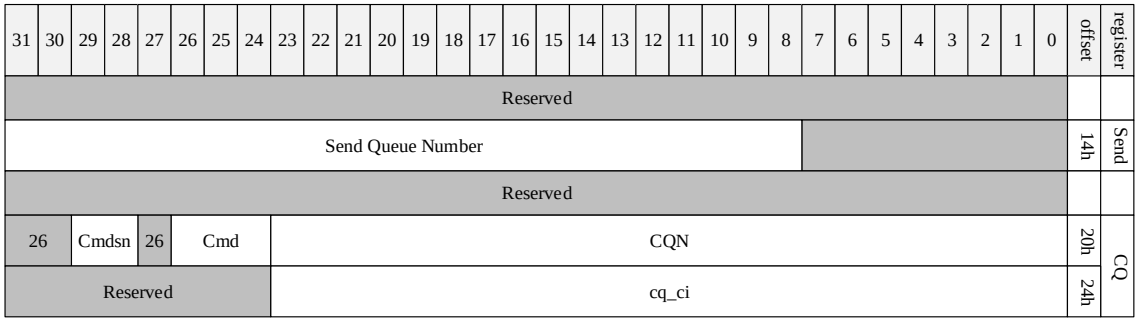


图 4-17 UAR 寄存器格式

Doorbell Creator 模块将按照图 4-17 中格式形成数据，并写入地址为用户接入区域初始地址加偏移量对应寄存器中。

4.3.1.6 CQ Processor 模块

CQ Processor 模块主要负责缓存和解析通道适配器写入完成队列的 CQE。因为在运行过程中会有两种 CQE 产生：一种用于指示通道适配器完成一次接收过程，数据已被 HCA 写入系统指定的存储空间，解析结果将用于数据接收模块；一种用于指示通道适配器对发送队列的 WQE 执行情况，解析结果将用于发送模块。本小节主要讨论发送操作对应的 CQE 处理。

图 4-18 为 CQ Processor 模块内部框图，在设计中，采用了 8 个完成队列和 QP 一一对应的方案。其中 CQ1 用于接收通道适配器对 QP1 中 WQE 执行情况的反馈。CQ_buffer 模块用于存储完成队列元素 CQE。CQ Decorder 模块接收传输接口的写请求，并根据写地址信号译码，将数据存入正确的完成队列中。CQ_Extractor 模块在接收到解析指令后，读取完成队列 CQE，进行解析并将解析数据上传。图 4-19 为 CQE 的数据格式。

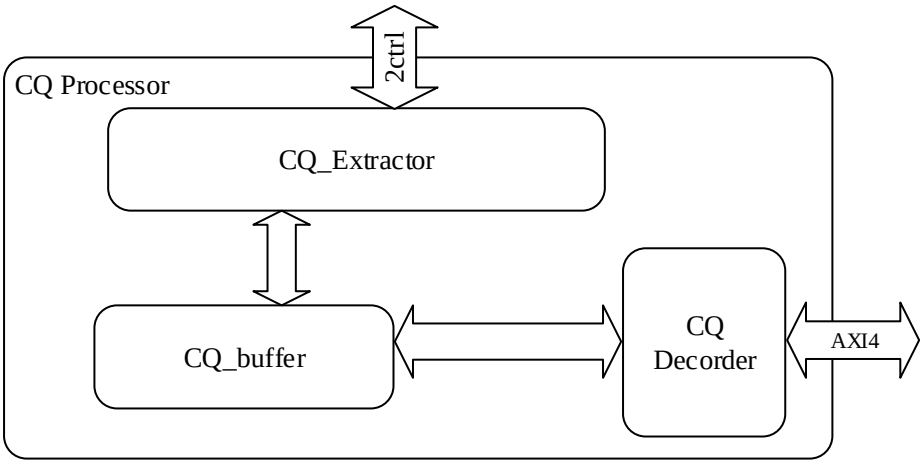


图 4-18 CQ Processor 模块结构框图

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
SMAC[31:0]																																10h
Byte_cnt																																14h
WQE_counter																Checksum																18h
Reserved								Timestamp[15:0]																Owner	S/R	IS	OPCODE					1Ch

图 4-19 CQE 数据格式

当 Opcode 字段为 11110 时指示发送操作执行失败；为其它编码时，指示编码对应发送操作执行成功。

4.3.2 数据接收模块

在接口通信控制器系统中，接收模块较为简单，因为数据从网络接收到写入用户指定接收缓存地址由通道适配器完成，系统只需划分一块用于接收的缓存空间，将首地址和接入权限通过接收 WQE 放入接收队列中。通道适配器接收数据完成后将形成 CQE 并写入完成队列中用于知晓用户。CQE 的解析将由 4.3.1.6 小节介绍的 CQ Processor 模块完成。

CQ Processor 模块解析获得的信息将上传给图 4-11 中 CQ_CTRL 模块，信息包含图 4-19 中指示接收数据长度 Byte_cnt 字段和指示操作类型的 Opcode 字段。并通过 S/R（发送/接收）信号线指示信息对应数据接收还是发送。当 S/R 信号指示操作为接收时，CQ_CTRL 将解析数据上传给 GMP 解析模块；为发送时，将解析信号存入 CQ_buffer 供用户查询。

4.4 QP Verb 命令模块的实现

通信管理协议不仅规定了待建立数据链路双方用户的握手规则，还规定了通信双方用作数据链路端点的 QP 状态转换过程。通信链路建立的最终结果就是通信双方的节点互有对方用于通信的 QP 信息，且队列对已转换为可发送状态。QP Verb 命令模块的功能就是用于 QP 状态转换命令的处理。图 4-20 为 QP Verb 命令模块的内部框图。

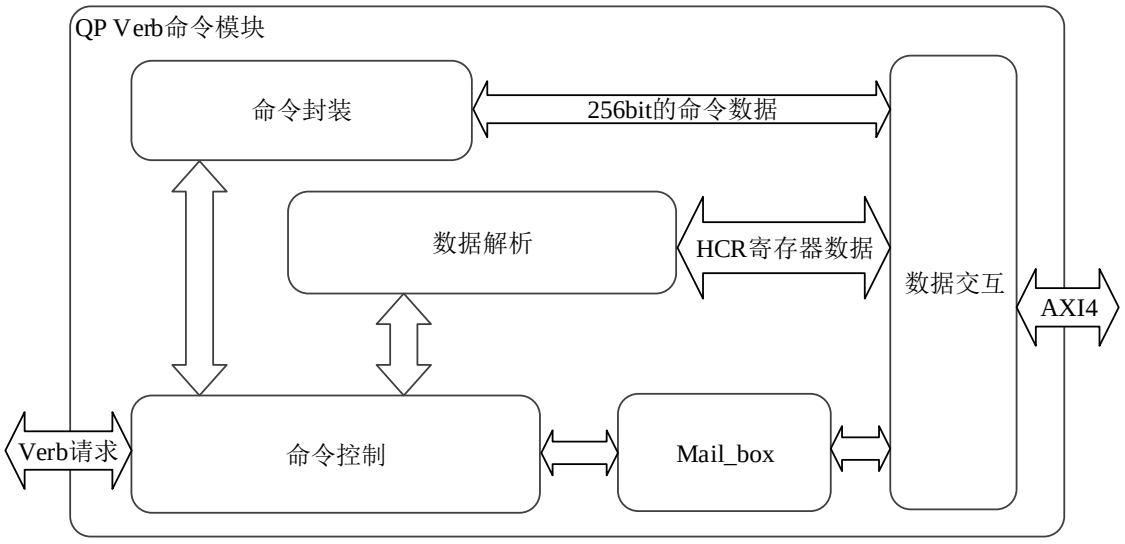


图 4-20 QP Verb 命令模块结构框图

命令模块由命令控制、命令封装、数据解析、数据交互和 Mail_box（邮箱机制）四部分组成，整个模块工作流程和图 2-4 完全一致。

（1）命令控制模块

控制整个模块的工作过程，该模块通过有限状态机实现，状态转移过程如图 4-21 所示

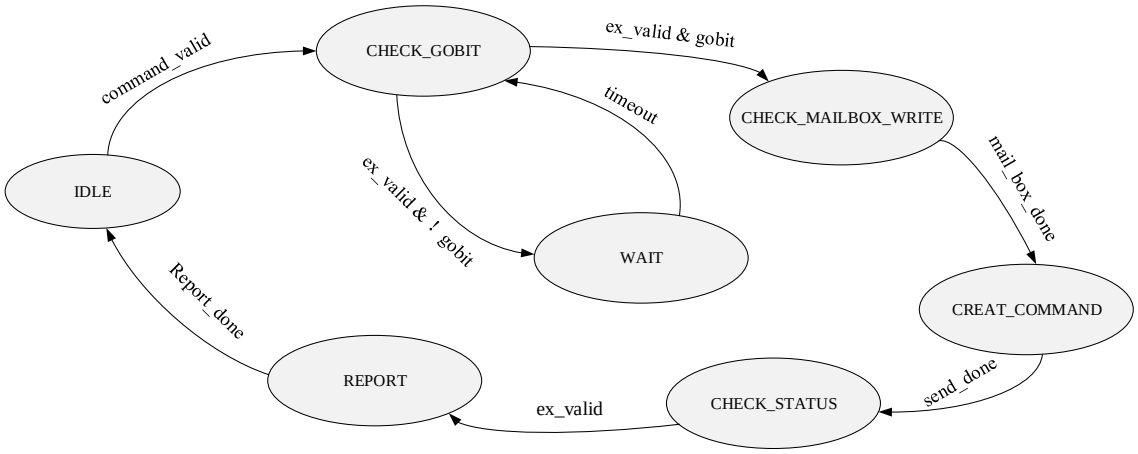


图 4-21 命令控制模块状态图

- IDLE：空闲状态。
- CHECK_GOBIT：查询 HCR.go 字段，判断 HCA 是否空闲。
- WAIT：系统忙时等待重传状态，等待时间可配置。
- CHECK_MAILBOX_WRITE：检查命令 Mail_box 数据是否配置结束。
- CREAT_COMMAND：命令封装状态，按照 HCR 寄存器格式配置控制命令数据。
- CHECK_STATUS：检查命令完成情况。
- REPORT：汇报命令完成情况。

（2）命令封装模块

命令封装模块根据命令请求，按照图 2-3 中 HCR 格式产生待发送的 HCR 命令。并向 HCR 寄存器在系统中映射地址发起写操作。

（3）数据解析模块

数据解析模块负责对读取的 HCR 数据进行解析，在一次命令处理过程中，模块需对 HCR 进行两次读取操作：第一次读取用于查询 HCR.go 信息，判断通道适配器是否处于空闲状态；第二次读取用于查询 HCR.status 字段信息，该字段指示了命令的完成状态。

（4）Mail_box

为了解决 QP 状态转换 Verb 命令的输入参数过大问题，系统制订了邮箱（Mail_box）机制。即系统需划分长度为 8 KB 的地址空间用于存放输入输出参数，

其中 4 KB 用于存放输入参数，4 KB 用于存放输出参数。在命令产生时，图 2-3 中寄存器输入输出参数字段将被写入输入输出存放空间的起始地址。对于四种 QP 状态转换命令，输入参数均为指示 QP 属性的控制字数据。而控制字数据中部分字段数据是在通信建立过程中通过管理数据包交互得到的，比如：控制字数据中远端 QPN 字段数据来源于远端通信管理模块回复的 REP 数据包。所以在发送 Verb 命令前需对 mail_box 输入参数进行复写操作，对部分控制字字段数据进行更新操作。

（5）数据交互模块

负责通过传输接口和通道适配器进行互相的数据传输。

4.5 本章小结

本章节主要对接口通信控制器中的通信管理模块、GMP 封装和解析模块、GMP 发送与接收模块以及 QP Verb 命令模块进行了 FPGA 实现。通信管理模块控制 IB 接口通信控制器工作状态，其中超时和计数模块提供可配置定时服务。GMP 封装与解析模块和数据发送与接收模块共同完成通信管理数据的交互，工作过程满足 IB 通信机制。QP Verb 命令模块可对 QP 状态转换 Verb 命令进行处理。

第五章 通信控制器仿真与验证

本文已对 InfiniBand 接口通信控制器的总体设计以及各个模块实现进行了详细的介绍。本章节将对通信控制器的功能正确性进行验证。

5.1 验证环境介绍

（1）硬件平台

本文设计中所采用的硬件平台为 Xilinx Virtex-7 FPGA VC707 开发套件，FPGA 芯片型号为 XC7VX485T。该开发套件是一款速率为 40 Gb/s 的高速平台，非常适合本设计的带宽需求。包含丰富的 IP 软核和硬核，为高带宽高性能的设计实现提供强大的支持。同时该套件包含 PCI Express、AXI、HDMI、UART、4 个 SFP+以及板载 DDR3 内存，可以为后续设计提供丰富的外部接口以及数据存储空间。图 5-1 为 VC707 的连接框图。

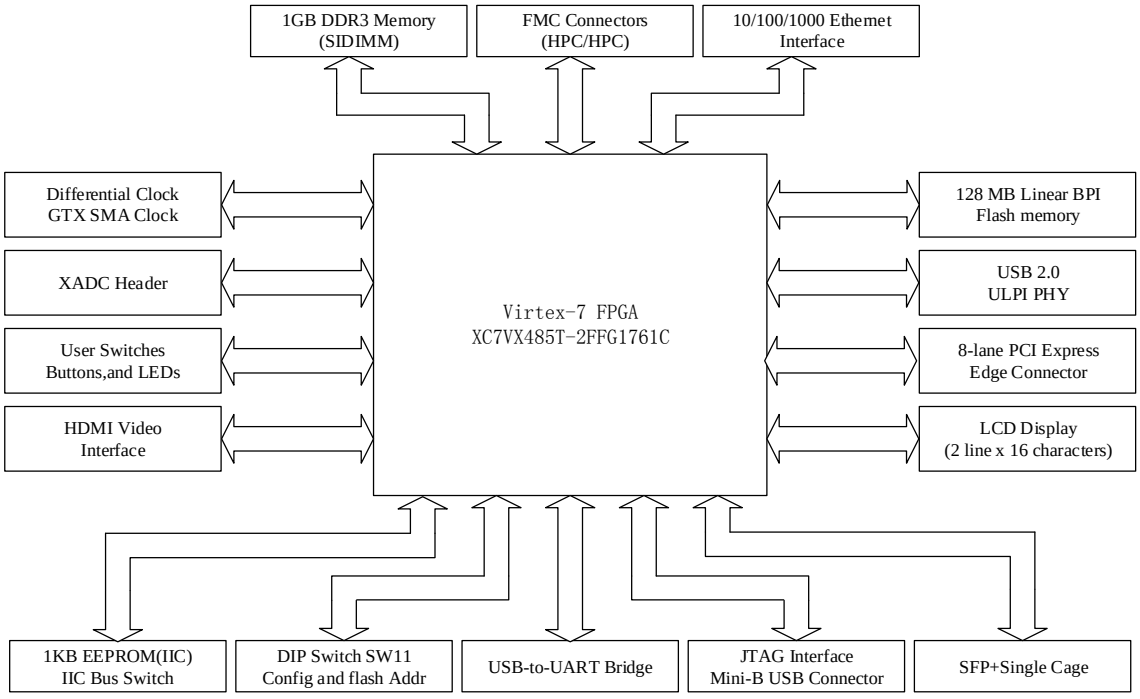


图 5-1 VC707 结构框图[40]

（2）开发与仿真工具

本文设计的开发工具为 Vivado，版本为 2015.2。Vivado 作为赛灵思公司在 ISE 之后推出的新一代 EDA 工具，提供全新构建的 SoC 增强型、以 IP 和系统为中心

的开发环境，以解决系统级集成和实现的生产力瓶颈。在模块设计（Block Design）下，用户可以将自己设计的模块封装为 IP 核，方便设计集成和重复利用。同时 Vivado 开发工具集成的 Vivado 仿真器，可提供行为级的逻辑仿真和综合实现后仿真，方便功能测试。本设计的模块设计、功能仿真、综合、实现以及比特流的生成[41]均在 Vivado 中完成。

5.2 通信控制器功能仿真

5.2.1 功能仿真测试方案

通过前面章节介绍可知，InfiniBand 网络中各个节点通过通道适配器接入网络进行数据交互。本文中，两个节点通信控制器将按照图 2-1 通过 HCA 接入交换网络。

每个节点通信控制器中均包含 CM Client 和 CM Server 模块，用于同时实现用户节点和服务器节点的功能。而在实际仿真过程中，为了重点仿真通信控制器通信建立和拆除功能的正确性，系统功能测试平台如图 5-2 所示。

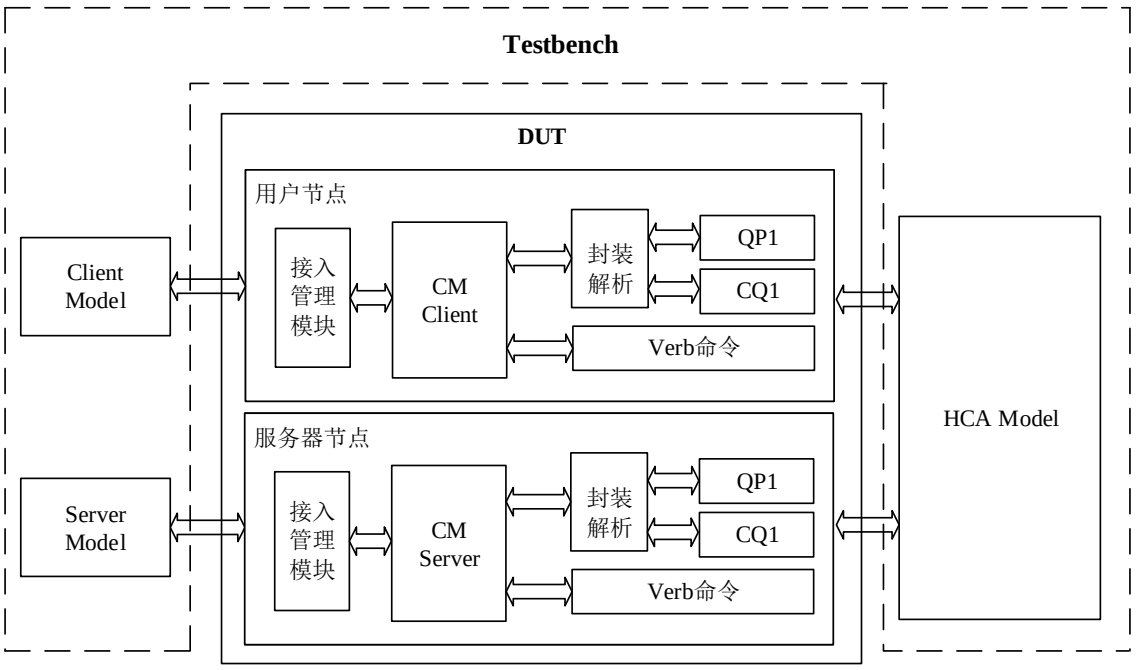


图 5-2 功能测试平台框图

通信控制器系统主要功能为建立或者拆除用户节点和服务器节点间的数据通信链路，因此，在测试平台中编写了 Client Model 和 Server Model 模块用于模拟用户节点和服务器节点中上层对于通信建立或拆除的处理流程，包含通信链路建立

或拆除请求和配置数据的产生以及节点对部分通信管理数据包的发送指示。节点通信控制器通过通道适配器接入 InfiniBand 网络，向通道适配器发起数据通信请求和控制命令完成和远端节点的数据通信。所以测试平台中 HCA Model 模块用于模拟通道适配器的操作，包含通信请求的接受、队列对中 WQE 的读取、WQE 指示操作的完成和 CQE 的产生。

测试平台设计完成后，便开始进行测试向量的设计。按照通信控制器的两大主要功能，测试向量主要分为通信链路建立测试向量和通信链路拆除测试向量。其中通信链路建立测试向量又分为正常建立测试向量，差错控制，仲裁策略测试向量等。通信链路拆除测试向量又分为用户节点请求拆除测试向量和服务器节点请求拆除测试向量。通信控制器的每一个功能都必须能在测试向量中找到。表 5-1 为针对通信控制器功能所设计的一些关键测试用例[42-43]。

表 5-1 部分功能仿真测试用例

测试用例名称	测试用例验证功能
通信链路正常建立测试	通信控制器完成用户节点和服务器节点间链路建立功能
通信建立差错控制测试	通信控制器超时与重传机制测试
Client 端请求链路拆除测试	Client Model 模拟用户节点上层发起链路拆除请求，用户节点通信控制器的链路拆除功能验证
Server 端请求链路拆除测试	Server Model 模拟服务器节点上层发起链路拆除请求，服务器节点通信控制器的链路拆除功能验证
仲裁策略测试	测试在多个 QP 有工作请求时的仲裁调度功能

5.2.2 功能仿真结果

针对表 5-1 中给出的测试用例，通过添加相应的测试激励，得到了仿真功能图，接下来将对每种测试用例的仿真结果进行分析。在结果分析前首先对测试中通信链路双方节点关键配置信息进行规定。节点配置信息如表 5-2 所示

表 5-2 双方节点部分配置信息

节点	QPN	LID	COM_ID	GUID	Retry	TimeOut
Client 节点	0x2	0x0001	0xaaaabbbb	0xccccddddeeeeffff	0x2	0x1
Server 节点	0x8	0x0004	0x11112222	0x3333444455556666	0x2	0x1

（1）通信链路正常建立测试

工作过程：通信双方按照 3.2.3 小节链路建立过程进行工作。

预期结果：通信双方正确进行 REQ、REP、RTU 数据包交互。接收方能正确解析出发送方的配置数据和私有数据。QP 状态转换和数据包发送接收按照第二章节描述机制进行。

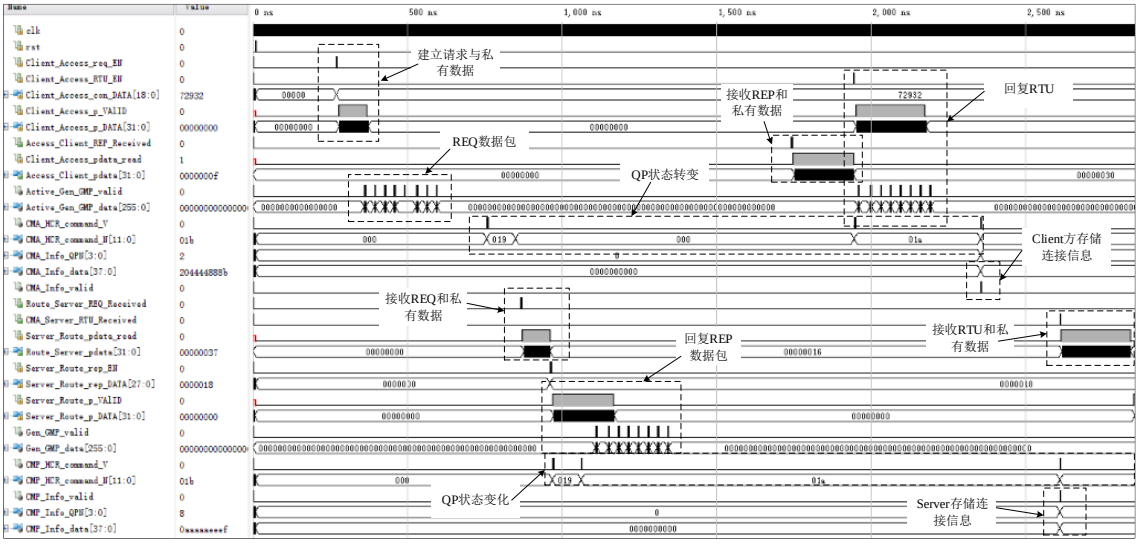


图 5-3 通信链路建立测试结果

结果分析：图 5-3 中列出了工作过程中关键信号，信号时序图展示了通信链路建立过程中数据交换过程和 QP 状态转换过程，仿真结果满足预期，进而证明功能正确。

（2）通信建立差错控制测试

工作过程：主要测试超时重传和通信控制器对 REJ 包的处理。

预期结果：按照配置的重传次数进行重传处理，或接收到 REJ 时能正确的上传错误信息。

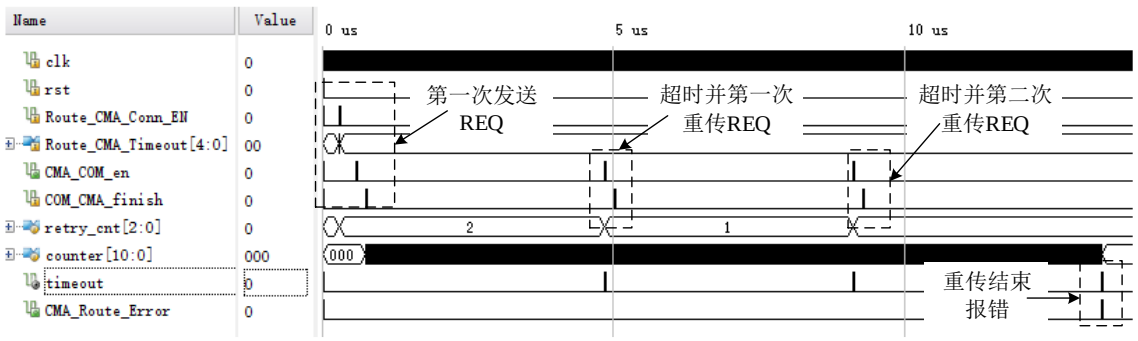


图 5-4 超时重传测试结果

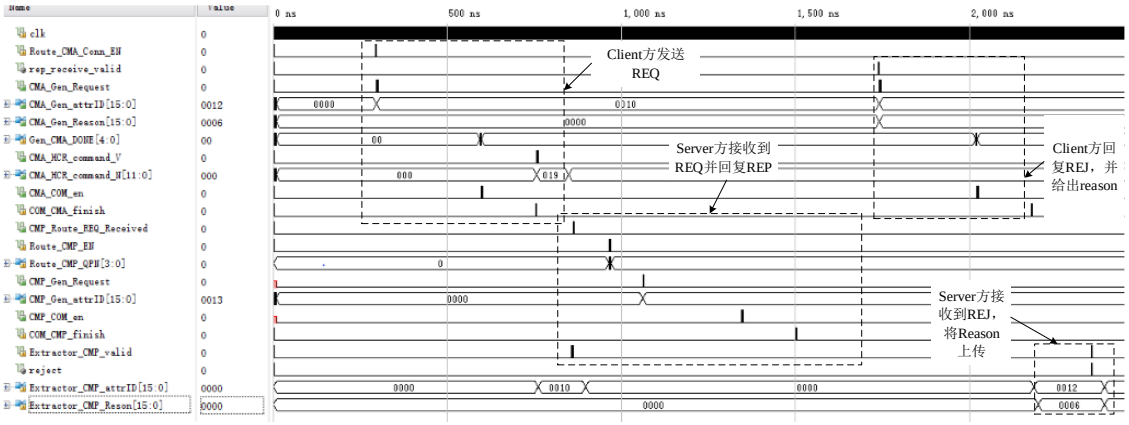


图 5-5 REJ 包处理测试结果

结果分析：如图 5-4 在超时重传测试中，Client 方一直未接收到 Server 方回复的 REP。在等待时间达到用户设定的服务时间后，将发起消息重传。重传次数达到了此处用户设定的两次后，仍未收到 REP，系统进行了报错处理。其工作方式符合超时重传机制。在 REJ 处理测试中，结果如图 5-5 所示，Client 收到了 Server 回复的 REP，但在校验过程中，REP 中包含无效 COMID，Client 方将回复 REJ 包，REJ.Reason 值为 0x6。Server 方解析收到的数据包，指示为 REJ，并将 Reason 上传。上传值为 0x6，满足设计要求。

（3）拆除测试

工作过程：通信双方均能发起链路拆除请求，本处以 Server 发起链路拆除请求，Client 方进行回应为例。

预期结果：通信双方正确的进行 DREQ 和 DREP 包交互，同时将本地 QP 转为初始化状态，将本地保存的链路信息删除。

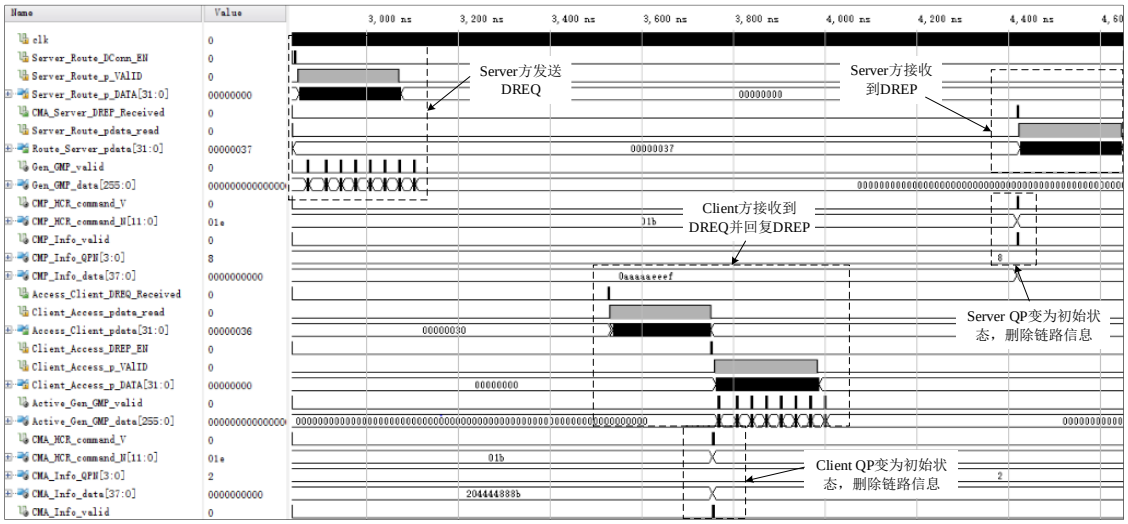


图 5-6 链路拆除仿真结果

结果分析：系统能正确的进行 DREQ 和 DREP 交互，并将本地 QP 状态转换为 RST 状态。并将本地对应连接信息删除，防止 Stale Connection 的产生。系统工作符合设计要求。

（4）仲裁策略

工作过程：IB 接口设计包含 6 个普通 QP 和 2 个特殊 QP，八个 QP 同时有多个 QP 对 WQE Creator 模块有工作请求。

预期结果：有通信请求 QP 按照图 4-13 仲裁方式得到 WQE Creator 模块的服务。

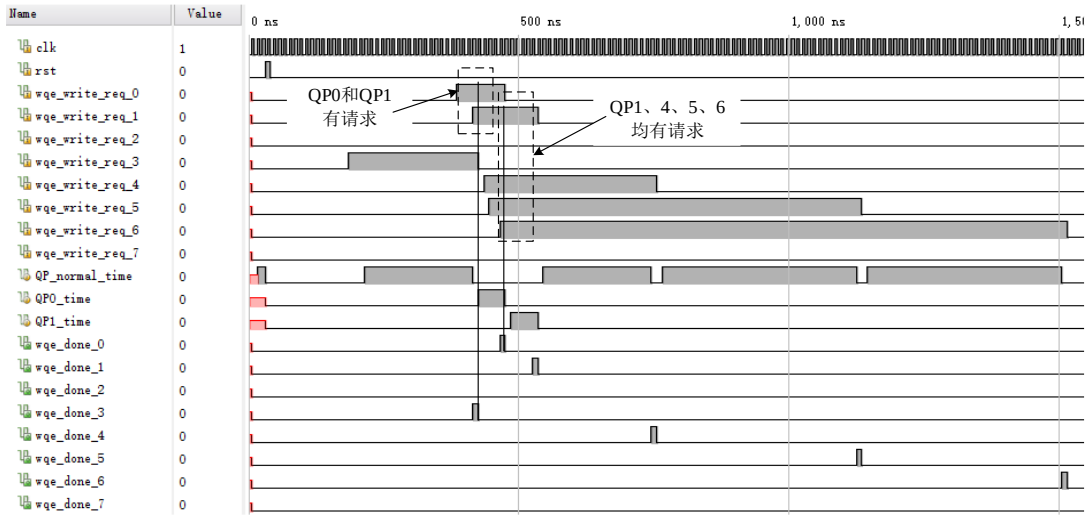


图 5-7 仲裁测试结果

结果分析：在仿真初始阶段，高优先级通道请求的到来会导致低优先级通道服务中断，经分析，是由于在为通道服务过程中发生了新的仲裁导致此问题，最后通过添加通道工作标志寄存器保证了仲裁正确性。经过修改后仿真结果如图 5-7 所示，从图 5-7 可以看出，在 QP3 正在接受服务时，QP0 有通信请求，但并未立即得到服务，而是此时仍需等到 QP3 请求完成后，才会执行新一轮仲裁，在 QP3 完成时，此时 QP0 和 QP1 均有请求，QP0 得到了服务。在 QP0 请求完成时，QP1、QP4 和 QP5 均有请求，QP1 得到服务。后面 QP4，QP5 和 QP6 按照请求顺序依次得到了服务，结果表明，仲裁模块按照仲裁策略正常运行。

5.3 通信控制器 FPGA 验证

5.3.1 时序报告和资源消耗

系统功能仿真通过后，就可以通过 Vivado 工具对系统设计进行综合、实现和比特流生成，并进行 FPGA 验证。在验证过程中，通过插入集成逻辑分析仪（ILA，

Integrate Logic Analyzer）来观察内部信号。其基本原理是通过在设计中标记待观测信号和触发条件，利用 FPGA 中未使用的分布式 RAM 将信号实时地保存，然后通过 JTAG 口传送到计算机并在屏幕上显示出时序波形。因此，通过该方法可以展示硬件内部工作状况，验证系统工作的正确性。

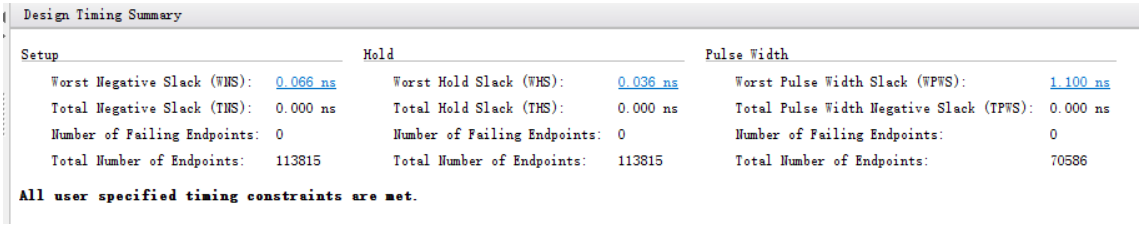


图 5-8 综合实现后时序报告

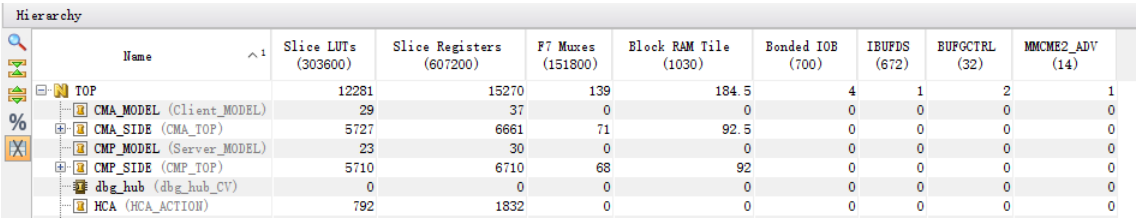


图 5-9 资源消耗报告

图 5-8 和图 5-9 为 InfiniBand 接口通信控制器综合实现后时序报告和资源消耗情况。在 Vivado 中，软件不再给出系统能运行的最大频率，取而代之系统时钟约束是否得到满足。本设计模块时钟约束条件为 250 MHz。图 5-8 中报告表明，系统能在 250 MHz 下正常运行，满足设计要求。在资源消耗上，通信控制器因为需要缓存不同的通信管理数据包，所以使用了较多的块状 RAM。

5.3.2 FPGA 验证方案

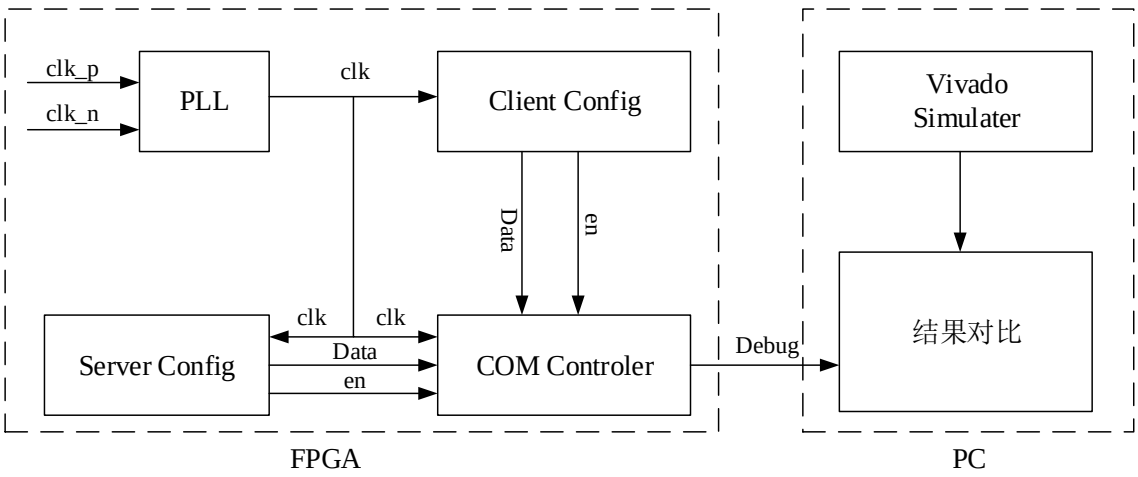


图 5-10 IB 接口通信控制器 FPGA 验证方案

本次下板验证方案如图 5-10 所示，系统 250 MHz 时钟源由 FPGA 芯片内部 200 MHz 差分时钟经时钟管理单元得到。Client Config 和 Server Config 模块用来模拟用户和服务器端配置链路建立和拆除数据。在系统启动后，Client Config 会发起一次通信链路建立请求，在通信链路建立完成后，Server Config 会对本次建立链路做拆除请求。PC 端获取工作过程中数据包的交互情况和工作流程，和 RTL 功能仿真以及协议中数据包格式做对比，判断运行结果是否正确。

5.3.3 FPGA 验证结果分析

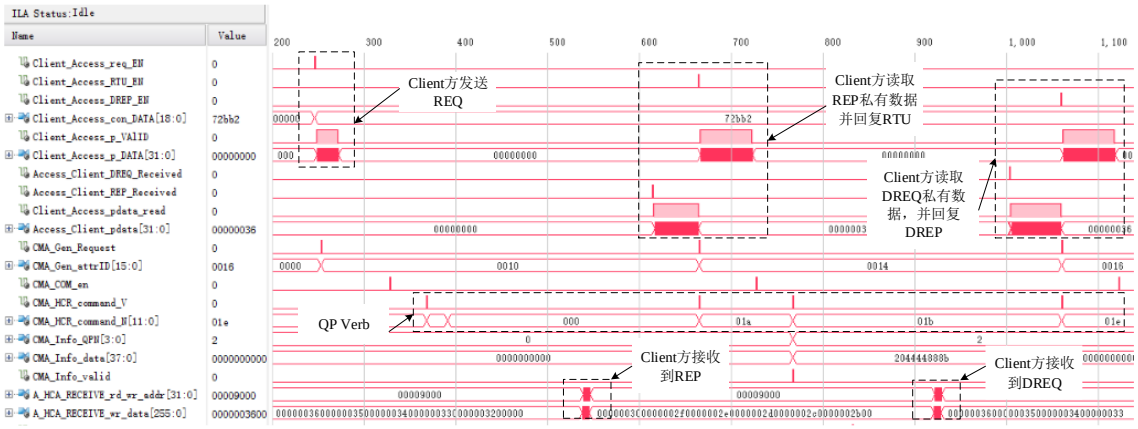


图 5-11 FPGA 测试 Client 方结果

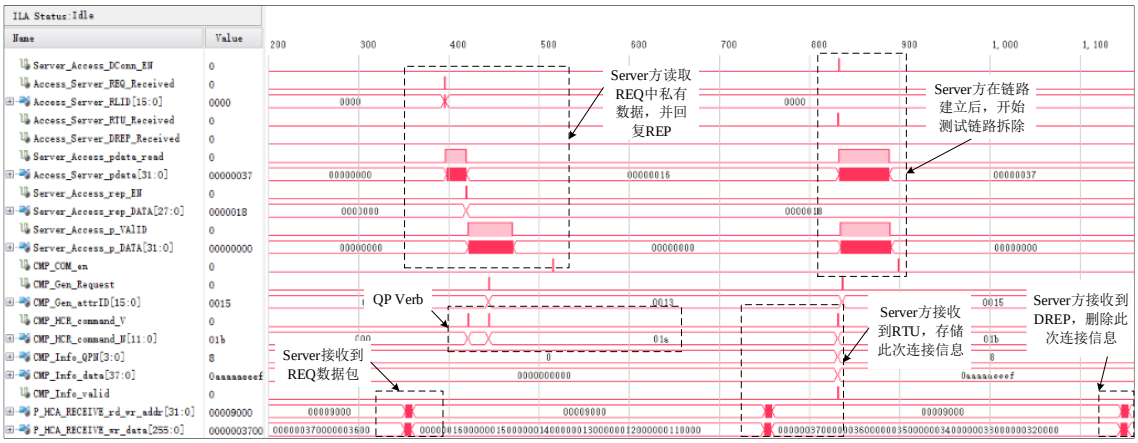


图 5-12 FPGA 测试 Server 方结果

图 5-11 和图 5-12 是通过 debug 抓取的 FPGA 运行数据波形，包含一次通信链路和通信拆除过程。结果显示通信控制器工作过程中通信双方数据包的发送与接收过程、QP Verb 命令的发送过程，以及链路信息的存储和删除均符合设计，并与系统 RTL 级仿真结果一致。下板过程中存在未知原因导致一块深度为 32 的 RAM 在例化操作已加载全零初始化文件情况下，首次进行读数据操作时读出非零值，导

致后续工作异常。多次测试后通过复位时首先对该存储空间进行写零操作得以解决此问题。接下来我们通过对其中部分数据进行抓取，证明其数据结果正确性。

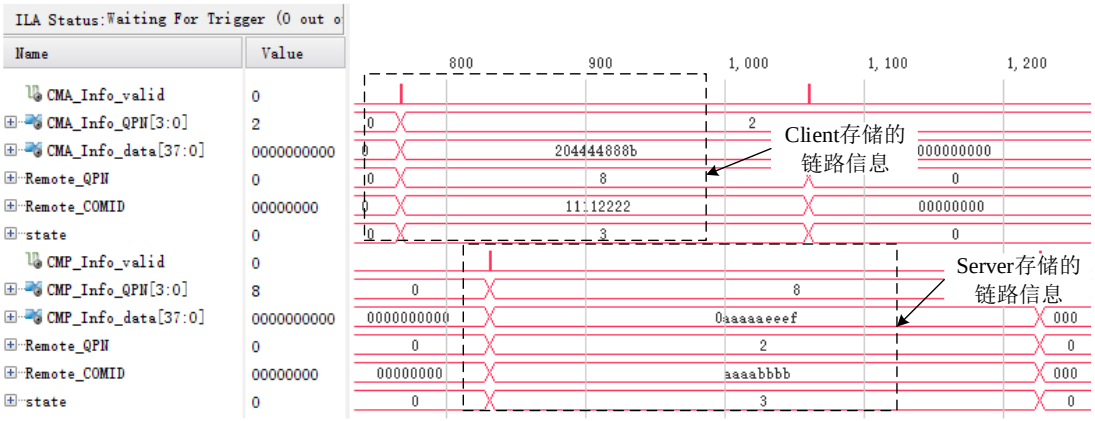
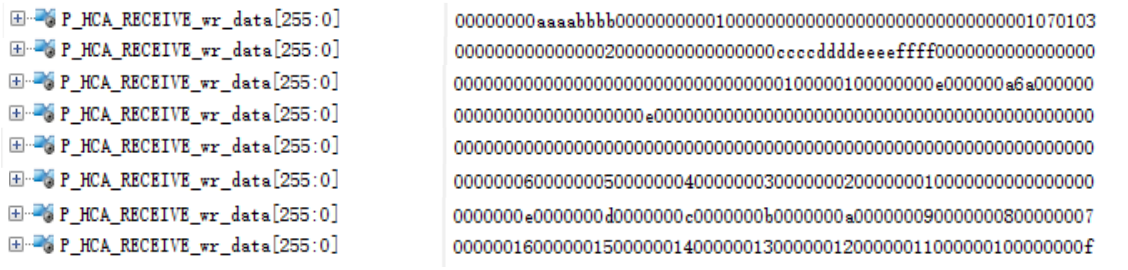
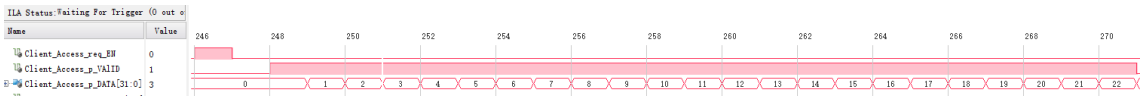


图 5-13 链路信息存储和删除测试数据

图 5-13 中 38 位 Info_data 包含链路对方 QPN、COMID 信息以及 2 位的链路状态信息。链路双方在此次 FPGA 测试中对链路信息进行了两次更新：第一次 State 字段为 0x3，指示链路建立完成，同时对比表 5-2 中通信双方配置信息，通信控制器正确存储了链路信息；第二次 State 字段为 0x0，指示链路被拆除。



(a)



(b)

(c)

图 5-14 FPGA 测试结果部分关键数据

(a)Client 封装的 REQ 数据;(b)Client 配置的 REQ 私有数据;(c)Server 解析的 REQ 私有数据

在工作过程中，通信控制器涉及到多种数据包交互，本文以 REQ 为例，验证

FPGA 测试中数据包正确性。图 5-14 为 debug 获得的 REQ 数据。其中，图 5-14（a）为 Server 接收的 REQ 数据。图 5-14（b）为 Client 用户通过 REQ 携带的 23 个 32 bit 私有数据，图 5-14（c）为 Server 用户通过解析 REQ 得到的 23 个 32 bit 数据。图 5-14（b）和（c）中二者数据一致，证明控制器能正确的封装和解析通信管理数据包中私有数据。

对于图 5-14（a），Server 接收的数据共 8 个 256 bit 数据，总长度为 256 Bytes，其数据组成如图 4-8 所示。对比表 3-2 和表 3-4 给出的 REQ 数据包格式，对数据进行分析可以获得表 5-3。

表 5-3 REQ 部分关键信息

信号	数值/Hex
MgmtClass	07
Method	03
Attribute ID	10
Local_ComID	aaaabbbb
Local_GUID	ccccddddeeeeffff
Local_QPN	2
Service TYPE	0
Timeout	01
Max_RETRY	02
Remote_LID	0004

通过对比 IB 协议对 GMP 的格式规定，表 5-3 中 MgmtClass、Method 和 Attribute ID 三个字段数值指示了数据包为 REQ。同时根据表 5-2 中双方配置信息，REQ 数据包中携带的 Client 方信息和配置信息一致。从而证明了系统工作的正确性。由于 REQ 数据包携带的数据量较大，此处仅配置了部分关键数据字段用以验证封装解析正确性，后续可通过对封装解析模块增减状态达到更多数据字段可配置效果。

5.4 本章小结

本章节首先对验证环境进行了介绍，然后给出了 IB 通信控制器的功能仿真方案，仿真结果证明了本文设计的正确性。最后给出了 IB 通信控制器的 FPGA 验证方案。验证结果进一步证明了本设计的正确性。

第六章 总结与展望

6.1 工作总结

集中式蜂窝网络架构作为下一代无线通信一种解决方案，可以有效降低网络的能耗以及维护费用，但同时也带来了高带宽低延迟的数据传输挑战。针对此应用背景，本文给出了以 InfiniBand 作为集中式基站间传输网络的解决方案。在各个基站中，基带数据将通过 FPGA+HCA 实现的 IB 接口接入 IB 交换网络。

本文主要针对 IB 接口中通信控制器进行了研究与 FPGA 实现，主要工作包括：

（1）在对 InfiniBand 协议进行了深入学习和研究基础上，结合 Mellanox ConnectX-3 Pro 通道适配器，对通道适配器控制机制和 IB 通信机制进行了分析，并以 GMP 传输为例，给出了 IB 管理数据包传输过程。

（2）本文以实际应用为背景，给出了 FPGA+HCA 的 IB 接口实现方案，并进行了 FPGA 模块划分和模块功能介绍。通过对通信管理协议中数据包结构、通信链路建立与终止过程、超时重传机制的分析，结合功能需求，给出了 IB 接口中通信控制器的总体设计框图，通信控制器不仅能让 IB 接口具有处理网络远端节点的链路建立与拆除请求的能力，而且能接受用户配置主动建立与拆除数据链路。

（3）对本文提出的通信控制器进行了 FPGA 实现，详细给出了 FPGA 内各模块的结构和设计思路。设计中采用模块复用，节约了逻辑资源。本文还针对多个 QP 同时有共用资源使用请求这一现象制定了仲裁策略。通过对各模块 RTL 级代码编写和级联，完成通信控制器实现。

（4）本文对通信控制器制定了功能仿真方案，设计了多个测试用例用以保证系统工作的健壮性和功能正确性，并对各测试用例仿真结果进行了分析。最后，本设计在 Xilinx VC709 开发板上进行了 FPGA 验证，通过 Debug 抓取数据波形并和协议规定的工作过程和数据包格式进行对比来判断设计功能正确性。时序报告表明通信控制器在传输数据位宽为 256 bit 时，运行速率超过 250 MHz，其 40Gb/s 的吞吐量满足 IB 接口中 HCA 需求。

6.2 研究展望

由于 InfiniBand 协议过于庞大与复杂，在有限的时间和硬件基础上，本文设计的接口控制器系统在功能和差错控制上仅完成了基本功能，后续还可以继续深入研究与实现如下内容：

（1）完成 FPGA 与主机通道适配器转接板，实现 FPGA 通过通道适配器接入 InfiniBand 网络。当前设计中，FPGA 部分已能按照 IB 协议生成通信管理报文，并按照通道适配器相关控制机制发起通信请求与命令发送，但由于两大硬件模块之间的转接板尚未完成，因此还不能真正接入 IB 网络。

（2）在通信链路建立与拆除过程中，数据包中有效数据部分可携带的链路信息多达 232 Bytes，本文在设计中可配置字段仅保留部分重要字段，后续设计中可以通过在封装模块中添加状态以及解析中添加寄存器以丰富可配置数据。

（3）完善错误应对机制。除了超时重传机制外，协议还制定了备用路径机制。在后续设计中，可以对该部分进行实现。

（4）将用户接口标准化。目前工作重点在于设计通信控制器，在用户接口设计方面考虑不够，以后可以考虑制定标准统一的用户接口。

致　谢

转眼间，到了学生生涯即将结束的日子，回首七年来一次次漫步在银杏大道、湖边和一次次奋斗在教研室、图书馆的日子，不禁心生感慨。虽然七年间充满酸甜苦辣，但是总体是美好、快乐和充实的。科大不仅给我们提供了优美的校园环境，其严谨的学习科研氛围更是为我以后的工作和生活打下坚实基础，在此即将毕业之际，对所有老师和同学们致以最衷心的感谢。

首先，感谢我的导师杨海芬老师，是杨老师领我进入研究生的大门，并一直给我提供了莫大的帮助。科研中，杨老师总是保持一丝不苟的态度，热情、认真。生活上杨老师更是品德高尚、待人和善，深受同学们的欢迎和喜欢。

感谢项目组指导老师阎波老师。阎老师热情开朗，学识渊博，拥有丰富的科研经验，指导我们一次次解决项目遇到的问题。每次项目会议时候，阎老师总是认真的聆听我们的问题，并耐心告诉我们应该怎么分析和处理，让我研究生生涯无论是在科研上还是在生活态度上都受益匪浅。

其次还要感谢李广军老师，李老师学识和做人方面均让我们深深敬佩。感谢教研室林水生老师、郭志勇老师、郑植老师、黄乐天老师和姚毅老师，是你们一直持有的求真求实科研态度和日以夜继的辛勤奋斗，给我们提供了如此优良的科研氛围与硬件条件。

感谢项目组的杨廷坚、谢林甫、李应谦和石叶明师兄，你们平时的耐心指导使我能快速融入科研生活。感谢项目组的王海崧和董玉停同学，大家一起完成科研任务，互相学习，一起进步，希望以后工作的日子大家继续一起努力。感谢解国强同学，和你一起交流问题，让我收获颇多。感谢李志文师弟，望你前程似锦。

感谢我的父母，在我成功时，你们比我更激动；失意时，你们鼓舞我前进。同时感谢本文的各位评审老师，你们辛苦了。

参考文献

[1] InfiniBand Trade Association. InfiniBand Architecture Specification: Release 1.0[S]. Beaverton: InfiniBand Trade Association,2000:62-1078

[2] P.Grun. Introduction to InfiniBand for end users[J]. White paper, InfiniBand Trade Association, 2010:2-53

[3] 温建伟,管剑波,孙志刚. InfiniBand 技术现状与发展趋势[J]. 2008 年中国高校通信类院系学术研讨会论文集 (下册), 2009:1-3

[4] T. Shanley, J. Winkles. InfiniBand Network Architecture[M]. Addison-Wesley Professional, 2003:1-521

[5] G. F. Pfister. An introduction to the InfiniBand architecture[J]. High Performance Mass Storage and Parallel I/O, 2001, 42: 617-632

[6] 王强,林小莉,曾繁泰.PCI 总线数据传输瓶颈分析及其解决方案[J]. 高性能计算技术, 2003 (004): 34-37

[7] 徐君明,裴先登,王海卫,等. 高性能计算机 I/O 技术 PCI Express 分析[J]. 计算机工程, 2004, 30(12): 6-7

[8] S. Coulter, J. Martinez. Introduction to InfiniBand[R]. Los Alamos National Laboratory (LANL), 2015:1-3

[9] A. R. Mamidala. Scalable and high performance collective communication for next generation multicore InfiniBand clusters[D].USA The Ohio State University, 2008:1-3

[10] C. X. Wang, F. Haider, X. Gao, et al. Cellular architecture and key technologies for 5G wireless communication networks[J]. IEEE Communications Magazine, 2014, 52(2): 122-130

[11] Specification I. Common Public Radio Interface (CPRI); Interface Specification[S]. China, Huawei,2004:1-5

[12] 周一青,潘振岗,翟国伟,等. 第五代移动通信系统 5G 标准化展望与关键技术研究[J]. 数据采集与处理, 2015, 30(4): 714-724

[13] 振勇,褚,木云,等. FPGA 设计及应用[M]. 西安：西安电子科技大学出版社, 2002：36-45

[14] Y. Lin, L. Shao, Z. Zhu, etal. Wireless network cloud: Architecture and system requirements[J]. IBM Journal of Research and Development, 2010, 54(1): 4: 1-4

[15] P. Timothy. Competition Heats Up In Cluster Interconnects[J]. USA: The Next Platform,2016:1-5

[16] R. Männer, B. Deluigi, Saaler W, et al. The POLYBUS: a flexible and fault-tolerant multiprocessor interconnection[J]. Interfaces in Computing, 1984, 2(1): 45-68

[17] N. Woods. Integrating FPGAs in high-performance computing: the architecture and implementation perspective[C].Proceedings of the 2007 ACM/SIGDA 15th international symposium on Field programmable gate arrays. ACM, 2007: 132-132

[18] 蒋芳,蔡玉高.揭秘世界最快超算“神威太湖之光”[J]. 半月谈, 2016 (13): 73-75

[19] 曹光权.InfiniBand 网络拥塞控制的实现和观测[D].湖南,国防科技大学, 2013:35-55

[20] 沈利.InfiniBand 网络接口的研究与实现[D].湖南,国防科学技术大学, 2010:32-74

[21] 刘英文,唐玉华,易晓东.面向分布式内存访问的低延迟 InfiniBand 接口[J]. 计算机研究与发展, 2014 (S1): 123-129

[22] A. R. Mamidala. Scalable and high performance collective communication for next generation multicore InfiniBand clusters[D]. The Ohio State University, 2008:2-3

[23] B. Woodruff, S. Hefty, R. Dreier, et al. Introduction to the InfiniBand core software[C].Linux symposium. 2005, 2: 271-282

[24] Intel Corporation. InfiniBand™ Linux Operating System Software Access Layer[M].USA Intel, 2002:2-6

[25] T. J. Carlin. Implementation of an RDMA verbs driver for GridFTP[M]. NEW HAMPSHIRE. UNIVERSITY OF NEW HAMPSHIRE, 2012:5-10

[26] Mellanox.ConnectX Family Programmer’s Reference Manual(PRM). rev 1.3[M].California. Mellanox, 2013:1-500

[27] 方沛.基于 InfiniBand 的分布式文件系统的设计与实现[D].南京,南京大学, 2013:10-56

[28] F. Mietke, R. Rex, R. Baumgartl, et al. Analysis of the memory registration process in the mellanox InfiniBand software stack[C].European Conference on Parallel Processing. Springer Berlin Heidelberg, 2006: 124-133

[29] M. J. Carnevale, C. S. Graham, D. F. Moertl, et al. Method and apparatus for implementing InfiniBand receive function: U.S. Patent 7,225,364[P]. 2007-5-29:1-3

[30] D. F. Craddock, T. A. Gregg, Judd I D, et al. InfiniBand work and completion queue management via head and tail circular buffers with indirect work queue entries: U.S. Patent 6,789,143[P]. 2004-9-7:2-7

[31] R. L. Graham, S. Poole S, P. Shamis, et al. ConnectX-2 InfiniBand management queues: First investigation of the new support for network offloaded collective operations[C] Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. IEEE Computer Society, 2010: 53-62

[32] B. D. Johnsen, Moxnes D G, Hodoba P. System and method for supporting automatic disabling of degraded links in an infiniband network: U.S. Patent Application 13/488,250[P]. 2012-6-4:5

[33] A. Stevens. Introduction to amba 4 ace[J]. ARM whitepaper (June 2011), 2011:2-50

[34] S. S. Math, R. B. Manjula, S. S. Manvi, et al. Data transactions on system-on-chip bus using AXI4 protocol[C] Recent Advancements in Electrical, Electronics and Control Engineering (ICONRAEeCE), 2011 International Conference on. IEEE, 2011: 423-427

[35] 贾志国,赵青苹.InfiniBand: 一种新型的高速互连网络[J]. 计算机工程与应用, 2003, 39(9): 172-175

[36] A. Vishnu, M. Krishnan. Efficient on-demand connection management mechanisms with PGAS models over InfiniBand[C].Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE ACM International Conference on. IEEE, 2010: 175-184

[37] M. Nakamura, K. Nakashima. Apparatus and method for performing infiniband communication between user programs in different apparatuses: U.S. Patent Application 14/682,385[P]. 2015-4-9:4-20

[38] B. McCarty. Learning Red Hat Enterprise Linux and Fedora[M].California. " O'Reilly Media, Inc.", 2004

[39] T. J. Carlin. Implementation of an RDMA verbs driver for GridFTP[M]. New Hampshire. UNIVERSITY OF NEW HAMPSHIRE, 2012:5-31

[40] Virtex X. FPGA VC707 Evaluation Kit[J]. Xinlinx, 2016:1-20

[41] S. Kilts. Advanced FPGA design: architecture, implementation, and optimization[M]. Boston. John Wiley & Sons, 2007:2-3

[42] 吕欣欣,刘淑芬.FPGA 通用验证平台建立方法研究[J].微电子学与计算机, 2010 (5): 46-49

[43] 李晓维,吕涛,李华伟,等. 数字集成电路设计验证:量化评估,激励生成,形式化验证[M].北京. 科学出版社, 2010:23-156

个人简历及攻读硕士期间的研究成果

个人简历：

岳杰，男，汉族，中共党员，四川巴中人，1992 年 8 月 2 日生。

2010 年 9 月-2014 年 6 月，毕业于电子科技大学通信与信息工程学院通信工程专业，获得工学学士学位；

2014 年 9 月-2017 年 6 月，就读于电子科技大学通信与信息工程学院通信与信息系统专业，攻读硕士学位。

参与的科研项目：

2015 年 3 月~2015 年 8 月，5G 高频项目

2015 年 6 月~2016 年 9 月，InfiniBand 接口研究与 FPGA 实现

研究生阶段个人获奖：

[1] 2014 年，研究生三等奖学金

[2] 2015 年，研究生二等奖学金

[3] 2016 年，研究生三等奖学金