

基于 SoC 的 NCSF 总线系统的研究与设计术

0 引言

数控系统现场总线以其高可靠性、抗干扰性的优点适用于数控系统控制通信，随着现场总线在工业控制领域的应用，现场总线已经成为下一代数控系统各设备单元之间数据交换的主要媒体。高档数控系统的特点是高速、高精，而高速高精的数控加工要求数控各设备单元之间的数据交换媒体能完成数据的高速实时传输，因此数控系统现场总线系统是决定下一代高档数控产业发展的一个重要因素。

传统的总线系统(具体系统结构关系如图 1 所示)实现为数控系统平台通信系统的一部分，主要完成上层应用软件如 PLC 模块，运动控制模块等与数控系统总线所连的伺服，I/O 等从站设备之间的数据通信。总线系统通过 NC(Numerical Control, 数控)主机的通信接口(PCI、USB 等)将通信数据发送到主站板卡，由主站板卡执行 PC 机并行数据—总线串行数据的发送/接收处理工作。主站板卡为简单的硬件转发电路，数据帧的封装、解封，差错控制，时钟同步等工作完全由总线系统执行，需要占用大量的 CPU 时间和 NC 主机内存，从而增加了 NC 主机的负载。

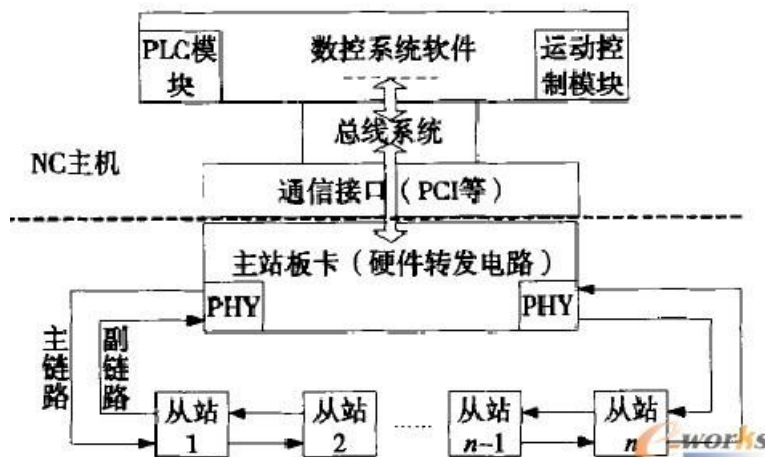


图 1 传统总线系统结构关系

中国科学院沈阳计算技术研究所数控总线实验室参照国家标准 GB/T 18759.3-2009《开放式数控系统第 3 部分：总线接口与通信协议》研发的数控同步现场总线 NCSF(Numerical Control Synchronous FieldBus)在数控现场总线实时通信方面具有优越的性能表现。

针对上述问题本文依据数控现场总线 NCSF 提出一种基于 SoC 的总线系统实现方法，设计选用 Xilinx 公司的 XC3S500E FPGA 芯片，具体设计中，

采用 MicroBlaze 软核 CPU 实现链路层及应用层协议处理。本文将在后续章节对总线系统的架构设计和实现方法进行具体阐述并对统性能进行分析。

1 背景介绍

1.1 NCSF 简介

为满足开放式数控系统总线通信的实时性、同步性、可靠性及可扩展要求，NCSF 总线参考 ISO/OSI 通信模型，由物理层、数据链路层、应用层、用户层行规组成，如图 2 所示。

图 2 数控现场总线 NCSF 网络模型



物理层负责协调总线在物理媒体中传送比特流所需的各种功能，定义总线接插件和传输媒体的机械和电气规约，以及为发生传输所必须完成的过程和功能；数据链路层为应用层提供周期、实时、无差错的数据链路；应用层负责维护站点间的安全、可靠的数据传输通路，并为用户层行规的命令与应答提供传输服务；用户层行规为装置特征、功能特性和行为的规范。

NCSF 基于 ISO/IEC 8802.3 物理层规范设计，利用网络实现高速、高可靠实时数据和基于以太网的非实时数据的传输。通信由主设备发起，主要支持单个主设备和多个从设备间的数据通信、站号的自动分配、网络拓扑结构检测、广播通信、基于广播机制的集总帧通信服务、点对点通信服务、总线延时测量和时间戳服务。在确定的网络拓扑结构下，从站设备对集总帧实时转发保证了通信的实时性，而周期性控制为主站行为。

1.2 SoC 及硬件简介

SoC 称为系统级芯片，也有称片上系统，意指它是一个产品，是一个有专用目标的集成电路，其中包含完整系统并有嵌入软件的全部内容。

FPGA(Filed Programmable Gate Array)是用户现场可编程门阵列集成电路的简称，它使用方便，改变逻辑功能易于实现，能够满足现代通信发展的需求。

MicroBlaze 嵌入式软核是一个被 Xilinx 公司优化过的可嵌入在 FPGA 中的 RISC 处理器软核，运行速度高达 150MHz，采用哈佛体系结构，具有独立的 32 位指令总线和 32 位数据总线，非常适合设计针对网络、电信、数据通信和消费市场的复杂嵌入式系统。

2 基于 SoC 的 NCSF 总线系统的架构设计

基于 SoC 的 NCSF 总线系统保留 NCSF 总线协议栈的用户层行规在 NC 主机中实现，以用户层行规接口的形式为伺服驱动、主轴驱动、I/O 等从站提供通用的系统应用接口(API)。协议栈的应用层和数据链路层实现在 FPGA 内嵌的 MicroBlaze 处理器中，配合 FPGA 内部的高精度时钟、CRC 处理、PHY 控制等硬逻辑模块，实现 NCSF 总线通信控制功能。FPGA 开发板使用 PCI 卡与用户层行规接口程序通信，数据链路层封装的数据最后经由以太网 PHY 芯片发送。具体层次结构如图 3 所示。

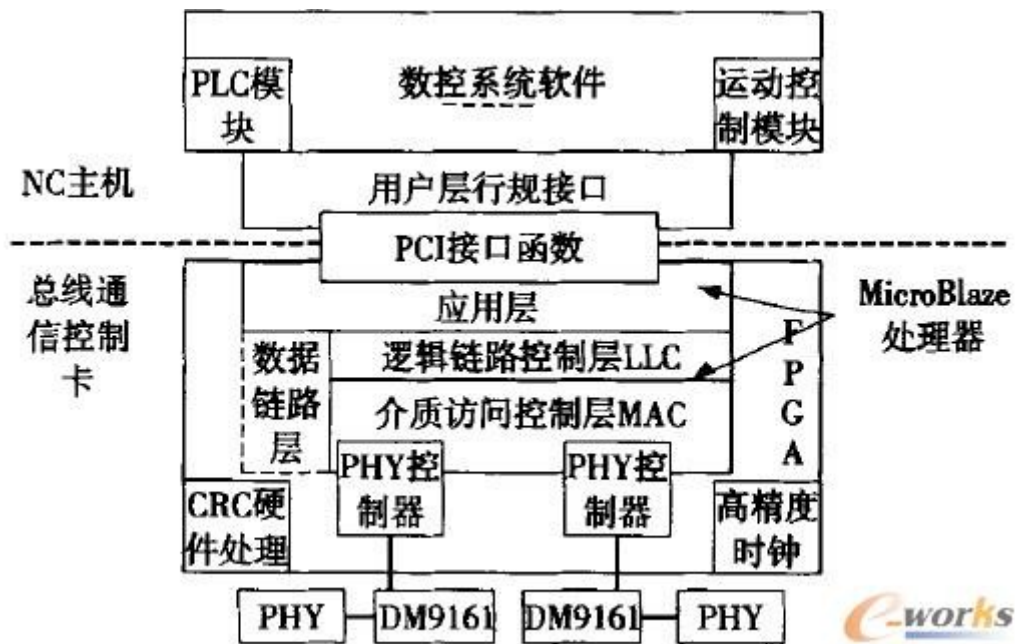


图 3 基于 SoC 的总线系统

2.1 NC 主机用户层行规接口的设计

用户层行规通过定义命令与应答的数据结构，确保装置间的互操作，支持面向应用的实现。根据开放式数控系统中装置的参数与行为特性，用户层行规包括管理、传感器控制、驱动控制与 I/O 控制等部分。用户层行规接口

程序主要负责通信数据的接收与发送。在发送阶段，程序首先读取运动控制、PLC 等模块的命令，然后将其封装成相应的命令数据结构，最后通过 NC 主机的 PCI 将数据结构发送到总线通信控制卡上；接收阶段，程序首先读取 PCI 上由总线通信控制卡发来的数据，然后将其解析为应答数据结构并传送给相应的功能模块。具体过程如图 4 所示。

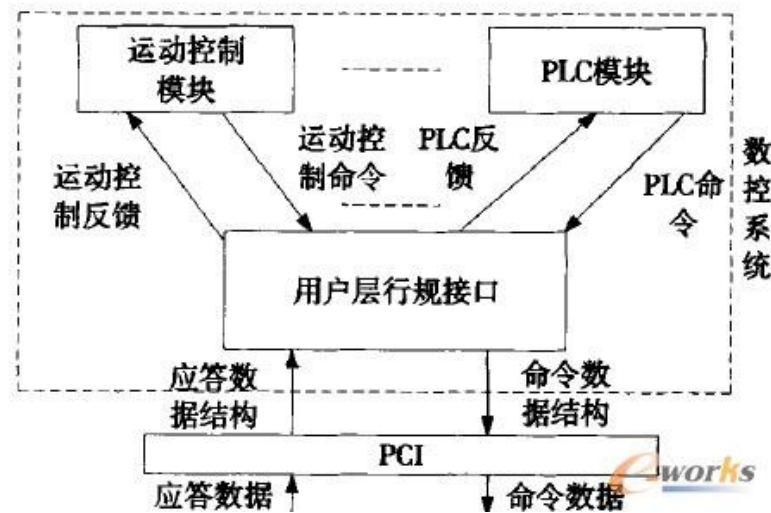


图 4 用户层行规层次结构图

2.2 FPGA 上总线系统的设计

协议栈的应用层和数据链路层实现在 FPGA 内嵌的 MicroBlaze 处理器中，配合 FPGA 内部硬件逻辑资源编程实现高层协议和底层逻辑的解析处理。应用层为用户层行规提供传输服务及安全支持，并实现用户层行规与数据链路层之间数据交互。首先应用层程序读取 PCI 中的命令数据结构并将其封装成 APDU(应用协议数据单元)转发到数据链路层，然后数据链路层程序读取来自应用层的数据并解析封装成数据链路帧转发到 PHY 控制器，最后数据帧由 PHY 以比特流的形式发送到 NCSF 总线上；当以太网控制器接收到来自 NCSF 总线的数据时，执行逆解析操作，最终由应用层程序将应答数据结构的发送到 PCI 上，至此完成一次完整的数据交互过程。具体过程如图 5 所示。

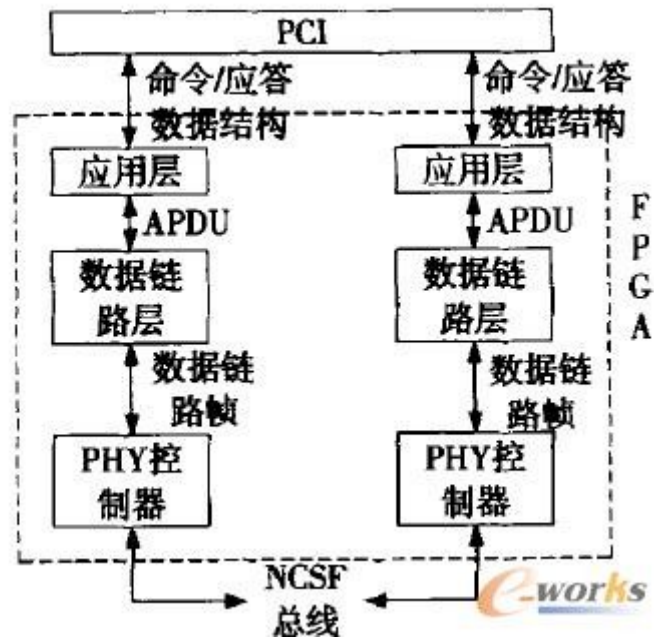


图 5 FPGA 上总线系统层次结构 3 基于 SoC 的 NCSF 总线系统程序设计

基于 SoC 的 NCSF 总线系统将 NCSF 协议栈分两个部分实现，即：用户层行规以用户层行规接口的形式实现在 NC 主机上，应用层及数据链路层实现在 FPGA 内嵌 MicroBlaze 处理器上，相应的程序设计也分两个部分进行。

3.1 用户层行规接口程序设计

用户层行规接口程序以外部时钟中断处理程序的形式调用执行，即当外部时钟计数器到期发送时钟脉冲信号至 NC 主机时，主机调用用户层行规接口程序响应中断。

接口程序与数控系统软件共享一个命令数据结构数组和一个应答数据结构数组。命令数据结构数组中存储的元素为数控系统事先封装好的需要发送的命令数据结构，应答数据结构数组中存储的元素为行规接口程序接收到的应答数据结构。

命令数据结构规划如下，具体实现可以根据实际应用进行调整。


```
char In_Group_Id; //组内命令号
char Parameter[ PARAMETER_LENGTH ];
//可变长命令参数
} Command_Structure;
```

应答数据结构规划如下,具体实现可以根据实际应用进行调整。

```
//应答数据结构
#define DATA_LENGTH 255 //可变长度
typedef struct {
char Command_Id; //命令号
char Reply_status; //报警或故障号
char Reply[ 2 ]; //应答状态
char Data[ DATA_LENGTH ];
//可变长应答数据
} Reply_Structure;
```



命令数据结构数组和应答数据结构数组的大小应根据实际应用中从站的数量进行动态配置,保证发往和来自每个从站的数据都有各自的缓冲空间。

如图 6 所示。其中命令函数是用户层行规为数控系统软件提供的接口,数控系统软件通过调用命令函数封装命令数据结构并写数据到命令数据结构数组。每个函数根据自己的功能将数据写人数组不同的位置,具体内存地址为基地址加偏移地址。行规接口程序主要工作包括:读取命令数据结构数组中的数据并映射到 PIC 的输出缓存中;读取 PCI 输入缓存中的数据并映射到应答数据结构数组中。由于 NCSF 总线支持集总帧(一个大的数据帧,里面包含所有从站的数据,适用于主从环形总线拓扑结构)和非集总帧报文,因此一次函数执行将读写整个数组,即一个通讯周期内所需要发送的周期性数据和非周期性数据;所有的安全性检查、出错重传、时钟同步、PCI 内存与从站地址的映射等控制功能完全由 FPGA 部分实现。

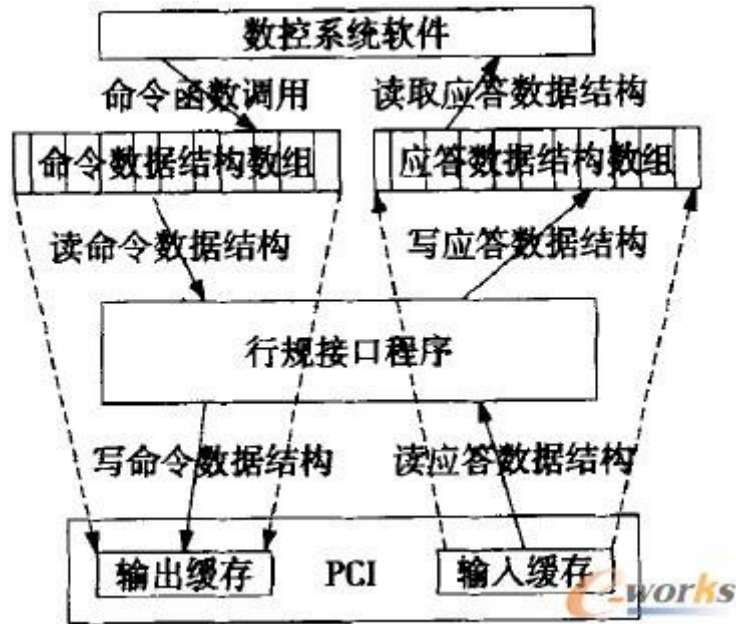


图 6 行规接口程序数据流向程序间的数据流向

3.2 FPGA 上总线系统程序设计

FPGA 上实现部分主要完成站点初始化、从站信息搜集、拓扑判断、数据收发等工作。其中站点初始化、从站信息搜集、拓扑判断工作通常只在 FPGA 上电时执行，除非出现通信故障需要重新初始化；数据收发以及时钟同步等过程则要以中断的形式周期执行。当 MicroBlaze 处理器接收到时钟计数器的脉冲信号时，将调用应用层程序处理中断即开始进行周期通信，主要工作包括数据的收发、状态机维护、安全性检查、出错重传、地址映射、时钟同步等。

3.2.1 应用层处理程序设计

应用层程序可处于不同的运行状态，如初始态、就绪态、异步连接态、同步连接态、连接释放态、故障态、结束态等，程序根据所处状态提供不同的传输服务，应用层服务需要调用数据链路层提供的基本服务命令接口实现。

当处理器接收到时钟脉冲信号开始进行中断处理时应用层程序处于就绪状态，之后程序会参照从站的反馈信息进入不同状态。该程序的主要工作包括：读取 PCI 接口中命令数据结构数组的映射数据，写从站的应答信息到 PCI 的应答数据结构数组的映射内存中。应用层与数据链路层的信息交换通过传输 APDU 进行。APDU 由用户数据负载和控制信息组成，用户数据负载存储用户层行规数据，控制信息用来描述 APDU 属性。APDU 控制信息数据结构设计如下，具体实现可以根据实际应用进行调整。

```
typedef struct {
    char APID; // APDU 标识号
    char O_Id; // 源标识
    char D_Id; // 目的标识
    char Length; // 长度
    char AS_CR; // 同步/应答标识
    char R_WDOG; // 同步/应答计数器
} AL_Structure;
```



应用层处理程序首先读取 PCI 缓冲区映射的命令数据结构数组并进行扫描，分析命令数据结构中的命令分组号和组内命令确定 APDU 控制信息，然后将命令数据结构数组封装成 APDU 数组，这一过程可以通过维护命令数据结构到 APDU 控制信息的映射表进行简化。下一步调用数据链路层基本服务命令接口发送 APDU 数组中的周期数据，接着检查发送状态进行差错控制、出错重传或发送非周期数据。最后时钟同步。每个通信周期都划分为若干个阶段执行，上述各操作必须在各自时间段内完成，其中出错重发和非周期数据发送共用同一个时间段(需要进行动态决策)。应用层处理程序具体执行过程如图 7 所示，具体实现可以根据实际提供的服务进行调整。

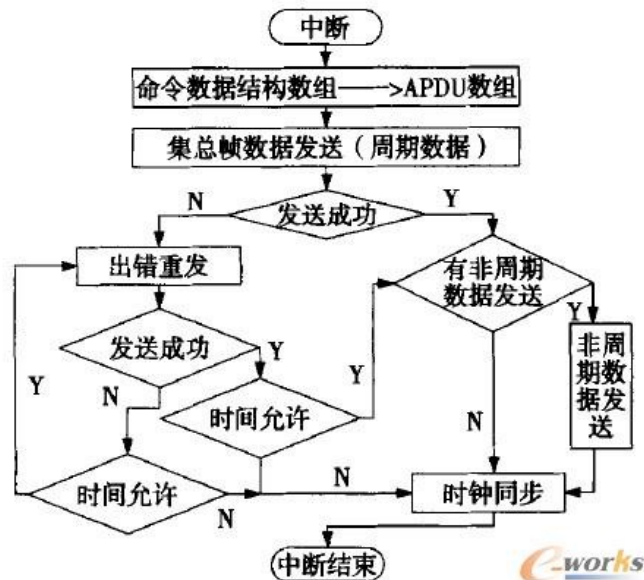


图 7 应用层处理程序执行流程

3.2.2 数据链路层程序设计

NCSF 总线的数据链路层划分成两个子层：LLC(Logic Line Control, 逻辑链路控制子层)和 MAC(Media Access Control, 介质访问控制子层)。向上层提供服务是 LLC 子层的主要功能, 它负责识别数据链路层的上层协议, 然后对它们进行解析; 控制对传输介质的访问是 MAC 子层的主要功能, 主要包括数据帧的封装与卸装, 链路管理, 帧寻址与识别, 帧差错控制等。

(1) LLC 子层程序设计

LLC 子层通过向应用层提供基本服务命令接口提供服务, 包括通信服务命令接口和管理服务命令接口。应用层调用通信服务命令接口收发 APDU; 调用管理服务命令接口进行差错控制、数据安全、时钟管理以及通信管理。

基本服务命令接口对应用层是透明的, LLC 子层需要对应用层调用的接口函数和传递的 APDU 进行解析, 这一过程可以通过维护对象字典来实现。对象字典为 LLC 子层服务和数据管理提供对应的 MAC 子层解释和定义, 应包括服务对象字典和协议对象字典。LLC 子层程序通过查询对象字典, 完成应用层数据通信任务到 MAC 子层数据通信任务间的转换。

LLC 子层还需要维护地址映射表进行上层逻辑地址到 MAC 子层实地址之间的转换(实地址可由 APDU 控制信息唯一确定), 映射表在 FPGA 上电初始化时确定。

表 1 为主要通信服务命令接口函数的定义, 具体参数可根据具体实现进行调整。

表 1 通信服务命令接口设计

接口函数名称	输入参数	返回参数
周期/非周期读	站点标识, 数据长度等	站点标识, 状态, 数据长度, 数据等
周期/非周期写	站点标识, 数据长度, 数据等	站点标识, 状态等

表 2 为主要管理服务命令接口函数的定义, 实际数量与应用中提供的服务数量相关。

表 2 管理服务命令接口设计

接口函数名称	功能描述
差错控制	识别接收到的帧是否正确
数据安全	提供通信安全及可靠性保证
时钟管理	进行延时测量和时钟同步
通信管理	对周期和非周期通信进行管理

(2) MAC 子层程序设计

MAC 子层主要完成 LLC 子层通信服务的具体实现，以及实际通信数据帧的封装。

LLC 子层各服务命令接口以及上层数据的收发工作最终都要通过收发链路层数据帧实现。NCSF 总线支持集总帧和非集总帧报文。非集总帧由头部、指令区、数据区和校验区组成，通信数据区直接由从站数据信息封装得来，组成相对简单。集总帧主要包括集总帧头部和数据部，头部为数据帧控制信息，数据部为各从站数据的集合。下面是集总帧头部和从站数据结构定义，具体结构可根据具实现调整。

```
//集总帧头部数据结构
typedef struct{
char ETHAREA[6]; //跨以太网设备控制字
char G_Length[2]; //帧长度(Byte)
char G_CTAL1[4]; //不同帧,内容不同
char G_Mytype; //帧类型主识别号
char G_stype; //帧类型辅助识别号
char G_Sno; //本帧包含站数
char G_CTRL2; //根据不同帧,内容不同
} Head;
//从站数据结构
#define DATA_LENGTH 255 //可变长度
typedef struct{
char U_Length; //数据块长度
char U_SID; //用户物理 ID
char U_Ctrl; //控制字
char U_WDG; //看门狗
char U_data[DATA_LENGTH]; //可变长数据区
char LCRC[2]; //用户 CRC 可选
} Slave_Data;
```



MAC 子层通过查询对象字典将应用层命令调用和 APDU 映射为 MAC 子层服务, 然后接口函数封装相应从站数据, 最终各从站数据被组装成集总帧发送。集总帧沿着环路传输, 当经过一个从站时, 该从站将对数据帧进行扫描, 查找、读取本站数据, 然后把发反馈数据写入集总帧, 并继续转发经过处理的集总帧。

为实现数据帧的收发和管理控制工作, MAC 子层需要维护数据收发缓冲区, 长度寄存器, 状态寄存器, 控制状态寄存器, 中断清除寄存器和计时器寄存器。MAC 子层收发程序通过设置控制寄存器, 长度寄存器和中断清除寄存器收发数据, 通过检查状态寄存器判断收发状态进行差错控制 and 安全管理。FPGA 上总线系统的数据流向见图 8, 涉及操作请参考应用层处理程序设计和数据链路层程序设计部分。需要注意的是, MAC 子层接收到的反馈信息需要通过查询对象字典和地址映射表以周期/非周期数据的形式传递到应用层, 然后由应用层进一步处理转发。

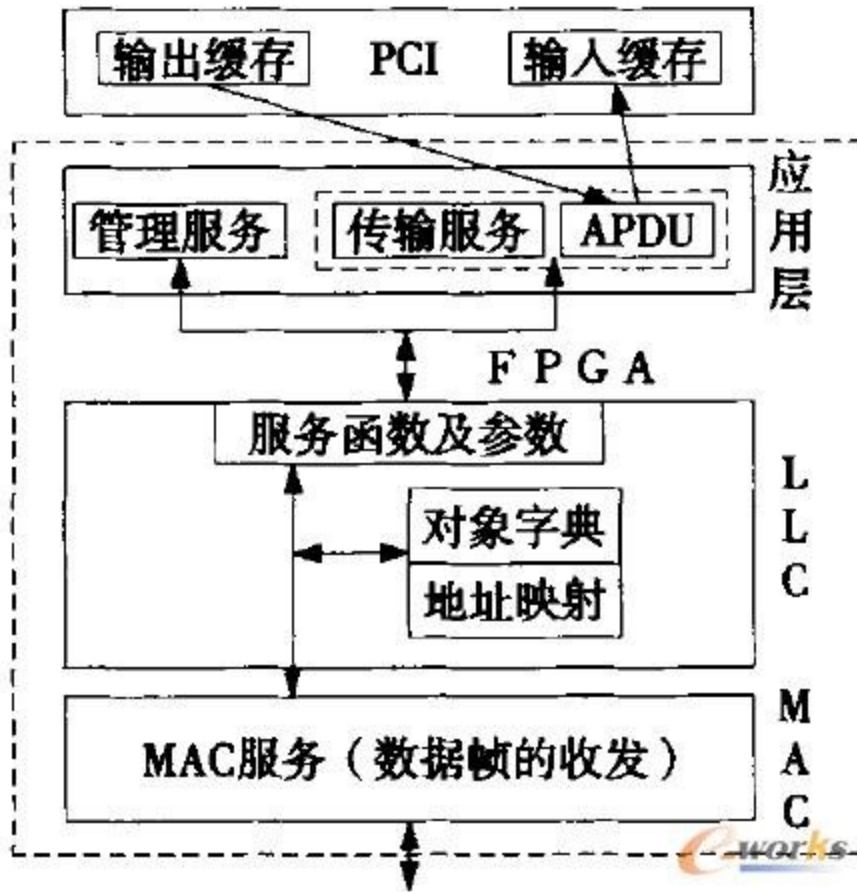


图 8 FPGA 总线系统数据流同关系

4 性能分析

基于 SoC 的 NCSF 总线系统使用 FPGA 软核 MicroBlaze 为现场总线 NCSF 协议栈的实现提供快速的专用处理器，由于该处理器具有运行速度快、占用资源少、可配置性强等优点，且只用于通信控制，可以满足数控现场总线对实时性的要求。本设计采用 Xilinx XC3S500E 芯片，表 3 中给出了总线系统综合时芯片逻辑资源的使用情况，数据表明系统资源占用率相对较低，可实现在 FPGA 上。

表 3 FPGA 资源使用情况

	逻辑使用数	可使用逻辑数	使用率
触发器	3105	9312	33%
4 输入 LUT	2229	9312	23%
Slice 占用	2247	4656	48%
逻辑相关 Slice	2247	2247	100%
非逻辑相关 Slice	0	2247	0%
4 输入 LUT	2304	9312	24%
IOB	90	158	56%
16 位 RAMB	5	20	25%
BUFGMUX	4	24	16%

表 4 给出了新系统性能测试结果，数据表明新设计完全可以满足数控系统对现场总线数据的实时、高速、可靠传输的要求。

表 4 基于 SoC 的 NCSF 总线系统性能指标

项目	指标	项目	指标
站间最大传输距离	100m	同步抖动	小于 0.6 μ s
传输速率	100Mbps	子站转发延时	小于 1.5 μ s
最大节点数量	128	通信周期	31.25 μ s

同时，NC 主机上仅实现 NCSF 总线协议栈用户层行规的内容，比完整的协议栈实现占用更少的 CPU 时间，降低了 NC 主机的负载，提高了 NC 主机的响应速度。

FPGA 实现部分配合 FPGA 内部的高精度时钟、CRC 处理、PHY 控制等硬逻辑模块实现 NCSF 总线通信控制，增加了总线系统的独立性、可扩展性，提高了总线的通信性能、稳定性和可靠性，使得总线系统的升级改造可以独立于数控系统从而降低了数控系统的开发维护成本。

5 结束语

针对数控系统软件对强实时性，高可靠性总线系统的应用需求，本文在沈阳计算技术研究所研发的 NCSF 总线的基础上，提出了一种基于 SoC 的总线系统的实现方法。测试结果表明，新设计可提供速率为 100Mbps，同步抖动小于 0.61 μ s，通信周期为 31.25 μ s 的通信服务，能够满足高档数控机床的

加工需求。同时该方法增加了总线系统的独立性、可扩展性，可在一定程度上减少数控系统响应时间提高总线系统的实时性。