文章编号:1671-4598(2019)03-0164-04 DOI:10.16526/j.cnki.11—4762/tp.2019.03.033 中图分类号:V446 文献标识码:A

基于 FPGA 的星内高速路由实现方案

王翠莲[1],李 寅[2],周 东[1],吴 伟[1]
(1. 北京空间飞行器总体设计部,北京 100094;2. 航天东方红卫星有限公司,北京 100094)

摘要:针对星内各载荷设备数据交互速率快、接口复杂的现状,设计了一种基于现场可编程门阵列(FPGA)的高速路由实现方案,解决了星内路由设计中矩阵交换、路由查找、同步动态随机访问存储器(SDRAM)仲裁访问控制、可靠性设计等多个关键问题;该路由方案采用模块化设计,具有便于集成和扩展的特点,可用于控制局域网络(CAN)总线、通用异步收发传输器(UART)、低电压差分信号(LVDS)等多种标准接口的载荷设备构建通信网络;同时,文章给出了该方案在工程中的具体实施和试验验证情况,可为航天器星内和星间路由设计提供参考。

关键词:路由查找;高速;矩阵交换;SDRAM

Implementation Scheme of Intra-satellite High-speed Routing Based on FPGA

Wang Cuilian[1], Li Yin[2], Zhou Dong[1], Wu Wei[1]
(1. Beijing Institute of Spacecraft System Engineering, Beijing 100094, China;
2. DFH Satellite Co. Ltd, Beijing 100094, China)

Abstract: Aimed at the situation of high data rate and complex interface of payload equipment in the satellite, a high-speed routing scheme based on FPGA is designed in this paper. Key problems such as matrix switching, routing lookup, SDRAM arbitration access, reliability design of intra-satellite are solved. The scheme is modular designed with the characteristic of easy integration and expansion, which can be used to construct communication networks for various standard interfaces such as CAN, UART, LVDS. The concrete implementation and experimental verification of the scheme is given at the same time, which provide reference for the design on intra-satellite and inter-satellite routing.

Keywords: route; high-speed; matrix switching; SDRAM

0 引言

随着卫星通信技术的发展,空间探测器载荷设备种类的增多,实时生成的探测数据数量越来越大,传输速率越来越快,数据传输协议也日趋复杂。因此需要设计星内高速路由转发器,完成各星内载荷设备数据的实时转发。在传统的航天器设计中,一般由星载数据管理软件实现各载荷数据的协议处理、数据转换、路由转发等功能。在数据交互速率快,传输协议复杂的情况下,软件实现复杂度大幅提升;此外,星载数据管理软件可用资源有限,在处理能力接近极限时,可靠性显著下降。文献[1]中提到的星内路由框架采用分层设计,将协议处理、数据转换和路由转发等功能分开处理。数据路由功能由底层的独立模块完成,无需考虑数据帧协议、类型等信息,实现数据帧的透明转发。

星内载荷设备交互数据内容包括科学观测与工程参数、遥控命令、遥测数据、软件重配置数据等,峰值数据吞吐量不小于 1 Gbps。目前星上常用的数字信号处理芯片包括数字信号处理器(DSP)和 FPGA 等。静态随机存取型 FPGA(Static Random Access Memory Based Field Programmable Gate Array,SRAM-FPGA)具有逻辑资源丰富,配置方式灵活,可动态重构等优势,在航天器设计中应用广泛,如接口设计、智能信息处理,大容量存储等。单片 FPGA 的处理时钟大于 100 MHz,采用并行化处理后,实时处理能力可达 Gbps。在高速路由转发器设计中,由于多路载荷设备数据峰值速率不同,在数据帧转发过程中需要很大的中间缓存。同步动态随机访问存储器(Synchronous Dynamic Random Access Memory,SDRAM)具有容量大、读写速率高等优势[2],目前 SDRAM 的时钟速率可达 133 Mbps,单片容量达到 128 MB 以上,可用于高速数据帧缓存。

基于以上分析,本文提出了一种基于 FPGA 的高速路由实现方案。该方案采用矩阵路由技术,完成多种接口形式的载荷设备数据帧接收、路由、转发等功能。

1 系统结构

基于 FPGA 的星内高速路由实现方案采用模块化设计思想,结构框图如图 1 所示。该设计由载荷设备接收模块、载荷设备发送模块、路由及分路模块、合路模块、SDRAM 读写控制模块组成。载荷设备接收模块完成不同接口的数据帧接收及缓存;载荷设备发送模块将数据帧转换为不同的硬件接口形式发送;路由及分路模块、合路模块实现数据帧的矩阵交换;SDRAM 读写控制模块负责多通道 SDRAM 读写访问仲裁以及 SDRAM 芯片的控制操作。

收稿日期:2018-08-27; **修回日期**:2018-09-26。
基金项目:国家自然科学基金(61472260)。
作者简介:王翠莲(1987-),女,河北,硕士,主要从事卫星数管/综合电子设计方向的研究。

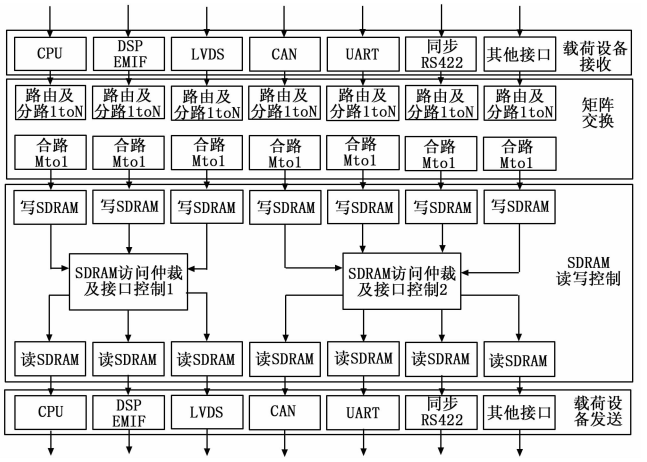

图 1　星内高速路由实现方案框图

2　系统模块设计

2.1　载荷设备接收

各载荷数据与 FPGA 存在多种接口形式，包括 LVDS 接口，同步 RS422 接口、UART 接口、CAN 总线接口、外部存储器接口（EMIF）等。各接口采用相同的数据帧协议，接收模块可以根据数据帧的头、尾指示信号（如门控信号、起始标志信号等）接收完整的数据帧。载荷设备接收的功能框图如图 2 所示

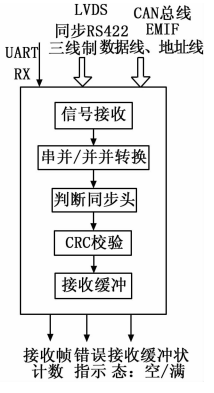

图 2　载荷设备接收模块框图

为保证数据接收的可靠性，在信号接收时加入数字滤波功能，滤除硬件通路上的毛刺。接收到的单比特数据或并行总线数据经过串并转换或并并转换后，生成按字节排序的数据帧。同步头正确且 CRC 校验无误的数据帧存入接收缓冲。接收缓冲采用 FPGA 内自带的 BRAM 实现，每个接收缓冲能够缓存 2 帧数据帧，采用乒乓操作的方式读写。当缓冲内的数据帧数大于 0 时，缓冲非空信号有效；当缓冲内的数据帧数等于 2 时，缓冲满信号有效，非空信号和满信号作为输出信号指示当前缓冲状态。

2.2　载荷设备发送

载荷设备发送端的实现框图如图 3 所示。当某一载荷设备发送缓冲满信号无效时，该模块从 SDRAM 中读取数据帧存入该设备发送缓冲。当发送缓冲非空信号有效时，表示当前缓冲内至少有 1 帧数据帧，该模块按照预先设定好的发送间隔将数据帧发出。在发送时将数据帧转换成符合载荷设备硬件接口要求的格式。

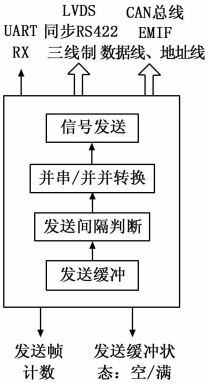

图 3　载荷设备发送模块框图

2.3　路由及分路

路由及分路模块置于载荷设备接收接口模块后端，对接收缓冲中的数据帧进行路由及分路。当载荷设备 a 接收缓冲非空信号有效时，路由及分路模块从接收缓冲中读取数据帧并获取数据帧的目的 IP 信息字段。目的 IP 与本地保存的路由表进行查找匹配。若匹配成功，则输出该路由表项对应的载荷设备编号 b，否则丢弃该帧。路由成功后的数据帧对应唯一的设备编号 b，将该数据帧存入 b 对应的分路缓冲。分路缓冲的个数 N 的选择取决于载荷设备 a 接收的数据帧所有可能的路由数量。分路缓冲的设计与接收缓冲及发送缓冲的设计一致，生成缓冲非空及满标志，指示当前缓冲状态。

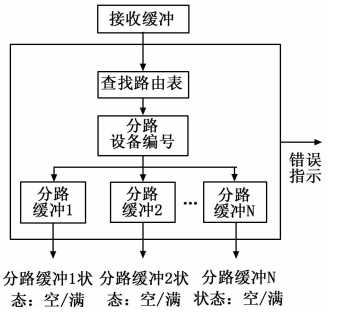

图 4　路由及分路模块框图

2.4　合路

合路模块置于路由及分路模块的后端，功能框图如图 5 所示。M 个路由至同一载荷设备的数据帧在该模块进行合路（一般情况下 $M=N$，即任意两个载荷设备间都可能存在数据交互），可根据需求将数据帧合成 1 路或多路。例如合路至某载荷设备的数据帧需要按优先级输出，可以将合路后的数据帧按优先级存入不同的合路缓冲。如果无特殊需求，每个合路模块仅需设置 1 个合路缓冲。

2.5　SDRAM 读写控制

合路后输出至某一载荷设备的数据速率之和可能远大于该设备的最大输出速率，因此需要设置一个大的缓冲来

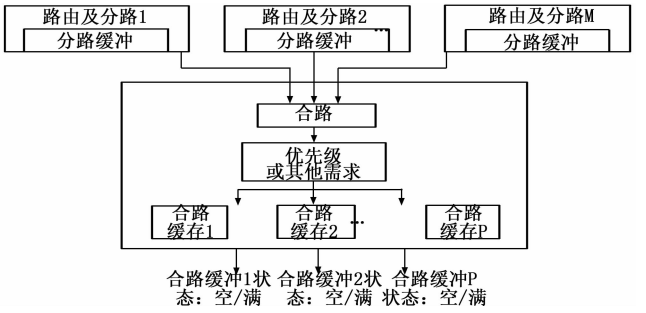

图 5 合路模块框图

缓存所有待发送的数据帧。FPGA 内部自带的 BRAM 存储容量及存储速率都不满足要求，需要外挂 SDRAM 来实现数据的高速随机存储及访问。40 位 SDRAM 在工作时钟为 100 MHz 的情况下，理论上可达到的有效读写速率约为 3.5 Gbps。SDRAM 读写模块由 SDRAM 写、SDRAM 读、SDRAM 访问仲裁、SDRAM 接口控制等部分组成。SDRAM 读写控制模块的组成框图及各子模块之间的数据交互关系如图 6 所示。

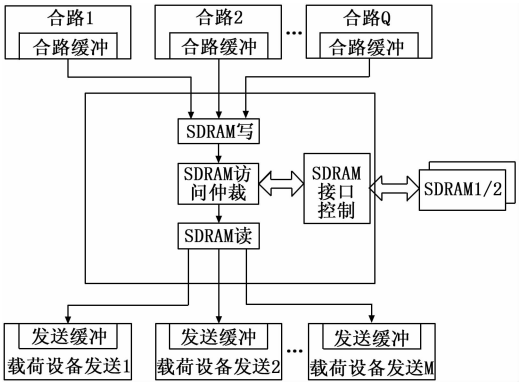

图 6 SDRAM 读写控制模块框图

SDRAM 的存储空间实行分区管理，每个载荷设备对应 SDRAM 的 1 个分区。当合路缓冲的非空信号有效时，SDRAM 写模块从合路缓冲内读取数据帧并转换为 SDRAM 突发包（Burst）的格式，存入 SDRAM 的指定分区。当某载荷设备的发送缓冲未满时，SDRAM 读模块从对应分区读取 Burst 并转换为数据帧后存入发送缓冲。SDRAM 读、写子模块需要管理 SDRAM 的读、写地址指针。SDRAM 访问仲裁模块解决多个读、写通道同时访问 SDRAM 的冲突问题。SDRAM 接口控制模块实现 SDRAM 的 7 种常用命令：空操作（NOP）、模式寄存器配置（Mode Register Set）、激活操作（Active）、突发读（Burst Read）、突发写（Burst Write）、刷新（Refresh）和预充电（Precharge）。

3 关键技术

3.1 路由查找

为解决多个载荷设备的速率不匹配问题，路由转发采用矩阵式设计，每一个载荷设备接收端口均设置一个路由表。常用的路由算法包括三态内容寻址存储器（Ternary Content Addressable Memory，TCAM）、Trie 查找和哈希查找[3]。TCAM 查找算法支持最大长度的路由查找，查找速度快，但在路由表项频繁变动的情况下，计算复杂，成本高。Trie 算法查找数据位宽为 1 bit，采用多分支结构或压缩算法[4]可提高路由表的查找速率，但仍不能满足高速路由转发的需求。哈希查找采用 Hash 函数建立映射表，实现复杂度高[5-6]。航天器设备受功耗的限制，高复杂度的路由查找算法不适用于星载路由器的设计。

本文根据 FPGA 的处理特点，设计了一种基于 BRAM 高速路由查找方法，将路由转发表保存在 FPGA 的内部缓存 BRAM 中。BRAM 在进行初始化时可加载. coe 文件作为初始值；同时，BRAM 可设置为包含读、写功能的双端口 RAM。因此，本文将静态路由表进行格式转换后作为 BRAM 的初始值保存；在航天器应用过程中，动态路由表由路由表管理软件按需写入，实现了动态路由和静态路由自主切换。为了防止单个 BRAM 读写冲突，提高设计的可靠性，路由及分路模块将路由表保存在两片 BRAM 中。两片 BRAM 的初始值均为静态路由表，路由及分路模块默认查找 BRAM2 中保存的路由表。动态路由表更新及查找过程采用乒乓处理，具体实现过程如下：（1）路由管理软件第一次注入动态路由表时，BRAM1 写使能信号有效，动态路由表写入 BRAM1；当 BRAM1 中的动态路由表更新完毕后，路由及分路模块切换到 BRAM1 中进行查找。（2）第二次注入路由表时，动态路由表写入 BRAM2，更新完毕后切换到 BRAM2 中查找。

BRAM 的读写位宽可灵活设计，每个处理时钟可从 BRAM 读出多比特数据，实现路由表的并行查找。表 1 为 BRAM 输出位宽不同的情况下的资源消耗和路由表查找耗时比较，设单个路由表项为 128 bit。

表 1 路由查找资源消耗及耗时比较

BRAM 位宽/bit	资源消耗	单个路由表项查找耗时/ * clk
8	1 BRAM/53 Slice	16
16	1 BRAM/105 Slice	8
32	1 BRAM/217 Slice	4
64	2 BRAM/450 Slice	2
128	4 BRAM/897 Slice	1

由表 1 可知，设计中选择的 BRAM 位宽越大，查找单个路由表项的耗时越短，但相应的资源消耗也越大。因此在实际应用中，应综合考虑实现复杂度和路由查找速率，折中选择 BRAM 位宽。本设计中 BRAM 位宽选择 32 bit。

3.2 SDRAM 仲裁访问机制

常见的仲裁方式包括两种：固定优先级和轮询优先级[7]。在固定优先级设计中，每个载荷设备设置固定的优先级，当多个载荷设备同时访问 SDRAM 时，优先响应高优先级的访问请求。当各设备的访问速率之和超出了 SDRAM 的最高访问速率时，低优先级的载荷设备数据可能

会丢失。在常规的应用场景设计中，一般遥控命令优先级最高，遥测信息次之，探测数据优先级最低。随着航天器任务的日益复杂，在不同的任务阶段，各载荷设备优先级可能发生变化，固定优先级设计不能满足该种应用需求。轮询优先级中设计一个计数器，当计数器达到特定值且设备访问请求信号有效时，响应该设备的访问请求。采用轮询的仲裁机制，对每个载荷设备来说，竞争得到 SDRAM 访问控制权的几率是一样的。轮询优先级的缺点是当载荷设备的访问速率超出了 SDRAM 能力时，访问请求会随机丢失且不可控。本文根据 FPGA 的设计特点和实际应用需求，设计了一种优先级可配置的多通道 SDRAM 控制器，根据不同通道访问 SDRAM 的应用场景和性能需求，通过路由表管理软件设置 SDRAM 访问仲裁方式以及各载荷设备的访问优先级，图 7 为设置帧格式示意图。使用该方法可提高 SDRAM 访问的灵活性，当访问速率之和超过 SDRAM 的能力限制时，可设置特定顺序的固定优先级确保某一路或几路数据帧的可靠交互，适当丢弃不重要的数据帧；反之可设置轮询优先级实现各载荷设备数据帧的平等交互。

访问仲裁方式0x00：固定优先级0xFF：轮询优先级	载荷设备数目 0~255	载荷设备0 优先级标志 (0x00最高，0xFF最低)	载荷设备1 优先级标志 (0x00最高，0xFF最低)	…	载荷设备N 优先级标志 (0x00最高，0xFF最低)

图 7　SDRAM 仲裁访问方式设置帧格式

3.3　可靠性设计

（1）SDRAM 数据 EDAC（Error Detection And Correction）存储[8]。在 SDRAM 中缓存了大量的数据帧，受空间环境单粒子效应的影响，存储数据可能发生错误，因此需要对 SDRAM 中保存的数据进行容错设计。EDAC 对存储数据进行编码，生成校验位与原始数据一起存入 SDRAM。在读出存储数据时，对数据进行译码，EDAC 可纠正 1 bit 错误或检测 2 bit 错误。本设计中选用的 SDRAM 位宽为 40 bit，其中 32 bit 用于存储数据，8 bit 存储校验位，增加约 19％的冗余存储空间。设计中将纠错及检错信息作为遥测上报，当由于空间单粒子效应出现错误累积时，用户可根据需求对 FPGA 进行复位或断电操作。

（2）关键寄存器三模冗余（Trip Module Redundancy，TMR）设计。在星载 FPGA 运行时，由于空间单粒子效应，可能会造成某一个单元被打翻，但连续两个功能相同的单元同时打翻的概率很低。因此，TMR 采用三套相同的逻辑存储或处理数据，在应用时进行三取二举手表决。1 个逻辑电路发生故障时，不会影响系统的正常工作，TMR 可以显著降低单粒子翻转的影响，提高设计的可靠性。同时，TMR 会增加 FPGA 的资源开销，降低系统的最高处理时钟频率。基于上述分析，本设计根据实际应用需求，对路由及分路模块寄存器，SDRAM 读、写地址指针等刷新频率低且作用关键的寄存器进行了 TMR 设计。

4　系统应用

衡量一种路由实现方案性能的主要技术指标是路由转发速率。为评估方案的转发性能，对各模块的流量情况进行统计，统计结果见表 2，其中 P 为载荷设备接收数目，Q 为载荷设备发送数目。FPGA 及 SDRAM 的配置情况如下：FPGA 处理时钟 50 MHz；载荷设备接收及发送模块按字节处理；FPGA 外挂 2 片 SDRAM；SDRAM 工作时钟 50 MHz，单片数据位宽 40 bit，采用（32，40）的 EDAC 纠错设计。

表 2　各模块流量统计

模块名称	速率统计	并行处理位宽/bit
载荷设备接收	P * 400Mbps	8
路由及分路	P * 1600Mbps	32
合路	Q * 400Mbps	8
SDRAM 读写控制	约 2 * 1.4Gbps	40
载荷设备发送	Q * 400Mbps	8
路由转发速率	约 2.8Gbps	/

基于本方案设计了星内高速路由转发器，搭载的载荷设备包括：三线制 LVDS 设备、四线制 LVDS 设备、UART 设备。选用 Xilinx 公司的 Virtex－4 系列 FPGA 芯片作为硬件平台，开发环境选用 Xilinx 集成开发环境 ISE14.4，FPGA 处理时钟为 50 MHz。该高速路由转发器实现了多个载荷设备的数据实时交互。为充分验证设计的功能和性能，专检设备全面模拟了星上的实际工作过程：（1）由专检模拟所有载荷设备，按照最大速率生成数据；（2）专检模拟载荷设备接收路由至该设备的数据帧并进行数据校验。经过多次、长时间的全速率数据试验验证，数据转发正确，无丢帧现象，设备工作稳定正常。

5　结束语

本文介绍了一种基于 FPGA 的星内高速路由实现方案，并在某航天器型号中得到实施，取得了有益效果。本方案采用了模块化设计思想，实现了星内高速路由系统快速集成，具备良好的可扩展性：（1）当载荷设备数目或接口形式改变，或整星修改数据帧协议时，仅需修改载荷设备接收、发送模块即可，不影响其他模块的设计状态。（2）当载荷接口的数据交互峰值速率增加时，可通过增大 FPGA 内部并行处理的路数或提高 FPGA 的时钟频率来解决。

本文提出的路由方案要求星内各载荷设备采用相同的数据帧格式，后续研究的重点将集中于通用的航天器路由器设计，支持不同协议的数据转发。同时进一步提升设计方案的健壮性和鲁棒性，为航天器星内和星间路由设计提供参考。

参考文献：

[1] 张亚航，袁　琚，于俊慧，等. 一种基于星内路由的航天器数管软件框架设计 [J]. 航天器工程，2015，24（6）：70－74.

[2] 苏海冰，吴钦章. 用 SDRAM 在高速数据采集和存储系统中实现海量缓存 [J]. 光学精密工程，2002，10（50）：462－465.

（下转第 172 页）