

# 基于 FPGA 的 No.7 信令 FISU 过滤

罗明阳, 廖红云, 苏凌旭

(重庆金美通信有限责任公司, 重庆 400030)

**摘要:**针对 CPU 处理填充信号单元(FISU)存在的缺陷,分析 No.7 信令系统的初始定位和 FISU 处理过程,提出一种基于 FPGA 实现 FISU 过滤的设计方案,给出实现的原理框图、FPGA 与 CPU 之间的通信机制和信号处理流程,使用 ISE 和 ModelSim 软件进行仿真。仿真结果表明该方案是正确、可行和有效的。

**关键词:**No.7 信令系统; 初始定位过程; 填充信号单元过滤; 现场可编程门阵列

## No.7 Signaling FISU Filter Based on FPGA

LUO Ming-yang, LIAO Hong-yun, SU Ling-xu

(Chongqing Jinmei Communication Co., Ltd., Chongqing 400030)

**【Abstract】**In allusion to the disadvantage of processing Fill-in Signal Unit(FISU) using CPU, on the basis of analyzing the process of initial aligning and FISU processing in No.7 signaling system, a new design scheme adopting Field-Programmable Gate Array(FPGA) to achieve FISU filter is put forward. The principle figures, communication mechanism between FPGA and CPU, and signal processing flow charts are carried out. ISE and ModelSim software are used to achieve simulation. Simulation result proves that the scheme is correct, feasible and effective.

**【Key words】**No.7 signaling system; process of initial aligning; Fill-in Signal Unit(FISU) filter; Field-Programmable Gate Array(FPGA)

### 1 概述

No.7 信令系统是国内外电话网采用的主要信令方式,属于共路信令,通过 E1 群路的 TS16 时隙采用高级数据链路控制器(High-level Data Link Controller, HDLC)协议传输<sup>[1]</sup>。在进入服务后的大部分时间里, No.7 信令系统的填充信号单元(Fill-in Signal Unit, FISU)只是完成链路填充功能<sup>[2]</sup>,但其传输数据量却非常大。

传统通信产品的 No.7 信令处理全部由 CPU 完成,若按照每分钟建立 100 次呼叫进行计算(属于业务通信繁忙状态),占用了 CPU 95% 以上的时间去收发处理正常情况下没有任何实用意义的 FISU。当用户呼叫建立减少或 E1 链路数量增加需要处理多路 No.7 信令时, CPU 资源消耗所付出的代价更加明显。因此,采用 CPU 处理 FISU 的方法存在较大的弊端,对 CPU 的处理时间开销过高,不利于提高系统处理性能。本文提出一种基于现场可编程门阵列(Field-Programmable Gate Array, FPGA)<sup>[3]</sup>实现 FISU 过滤处理的方法。

### 2 FISU 过滤的方案

#### 2.1 FPGA 原理框图

图 1 是 FISU 过滤的 FPGA 内部原理,该方案采用“CPU + FPGA + E1 接口芯片”的硬件结构实现对 No.7 信令系统的处理。其中, CPU 主要完成上层消息信号单元(Message Signal Unit, MSU)[2]和链路层链路状态信号单元(Link Status Signal Unit, LSSU)[2]的处理, FPGA 完成填充功能的 FISU 的自动发送和过滤, E1 接口芯片完成帧同步和 HDB3 编解码等处理。本文重点讨论 FPGA 内部 FISU 的自动发送和过滤的实现。图 1 中 FPGA 内部每个模块的主要功能如下:

(1)发送控制:完成 CPU 的 LSSU 和 MSU 的直接发送和 FISU 的自动发送,进行前向指示比特/前向序号(Forward Indicator Bit/Forward Sequence Number, FIB/FSN)和后向指示

比特/后向序号(Backward Indicator Bit/Backward Sequence Number, BIB/BSN)[2]的自动更新。

(2)接收控制:完成信令识别,将 LSSU 和 MSU 直接送往 CPU,对 FIB/FSN 和 BIB/BSN 进行识别,对具有相同 FIB/FSN 和 BIB/BSN 的 FISU 直接过滤。

(3)FIFO:完成 CPU 到发送和接收控制模块之间的信令缓冲。

(4)控制寄存器:完成 CPU 发送信令的状态控制。

(5)状态寄存器:完成 CPU 接收信令的状态汇报。

(6)HDLC 收发处理:完成 HDLC 的插零、删零、CRC 计算、并/串转换等处理。

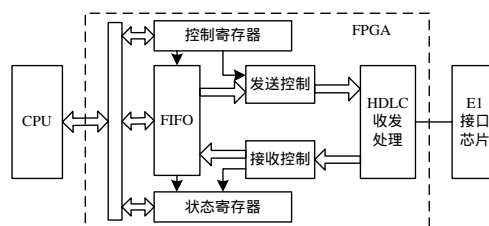


图 1 基于 FPGA 实现的 FISU 处理原理框图

#### 2.2 CPU 与 FPGA 之间的通信机制

当出现以下情况之一时, CPU 需要将信令写到 FPGA 发送链路:

- (1)FISU 的 FIB/FSN 或 BIB/BSN 发生变化。
- (2)发送 LSSU 和 MSU。

当出现以下情况之一时, FPGA 需要将接收到的信令直接送往 CPU:

**作者简介:**罗明阳(1976 -),男,高级工程师、硕士,主研方向:嵌入式系统;廖红云,高级工程师;苏凌旭,工程师

**收稿日期:**2009-09-15 **E-mail:** lmyjinmei@163.com

- (1) FISU 的 FIB/FSN 或 BIB/BSN 发生变化。
- (2) 接收到 LSSU。
- (3) 接收到 MSU。
- (4) 初始定位完成后的第 1 个 FISU。

由于 MSU 完全由 CPU 处理,因此 MSU 的重复缓冲区在内存开销,需要重发时 CPU 只需要将对应的完整 MSU 消息写入发送 FIFO 接口, FPGA 具有 FIB/FSN 或 BIB/BSN 的记忆功能,保证后续 FISU 与上次最新 MSU 的 FIB/FSN 或 BIB/BSN 保持一致。

### 2.3 CPU 与 FPGA 之间的数据传输格式

CPU 与 FPGA 之间的数据传输采用互控方式的 FIFO 实现,数据格式采用标准的 No.7 信令系统链路数据格式。

按照 No.7 信令系统规定,信令长度 64 Byte,因此, FIFO 深度设为 64。其中,Byte3 中的最高位(D7)表示接收信令的错误指示 ERR,当 ERR = 1 时表示该信令帧的 CRC 校验错误;低 6 位(D5-D0)表示后续的信令长度(字节数),只有当 ERR = 0 时,该长度指示才有意义。

## 3 FISU 过滤的实现

### 3.1 发送方向的 FISU 处理

图 2 是发送方向的信令处理原理框图,当 CPU 没有 MSU 或 LSSU 发送时,由 FPGA 自动完成 FISU 的发送。图中每个标志位的状态变化如下:

- (1) Cpu\_Sig\_Flag: 由 CPU 置 1,从双口 RAM 读取数据写入 HDLC 发送逻辑后,由 FPGA 清 0。
- (2) Dpam\_Store\_Flag: 由 CPU 置 1,从双口 RAM 读取前 2 个数据保存到 FSN/BSN 寄存器后,由 FPGA 清 0。
- (3) Hdlc\_Rdy\_Flag: HDLC 发送模块发送完后置 1, FPGA 写数据到 HDLC 的 FIFO 完成后清 0。
- (4) Hdlc\_Eop\_Flag: FPGA 写数据到 HDLC 的 FIFO 完成后置 1, HDLC 发送模块发送完后由 HDLC 模块清 0。

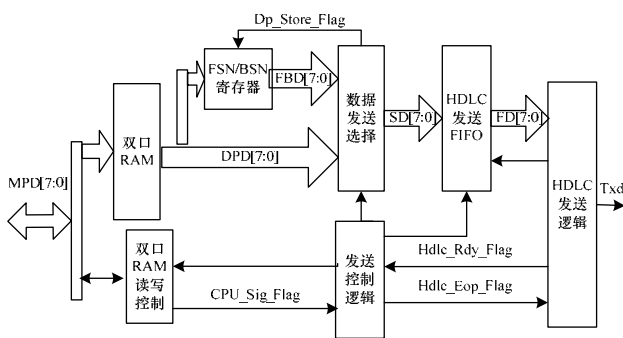


图 2 发送方向的信令处理原理框图

图 3 是发送方向的信号流程及状态转换示意图,图中方框内的文字内容说明了当前的数据操作方式,箭头方向表示触发状态变换的条件。图中的信号流程和状态转换如下:

- (1) CPU 读到 Cpu\_Sig\_Flag = 0 后,向双口 RAM 写入信令,然后对 Cpu\_Sig\_Flag 和 Dpam\_Store\_Flag 置 1。
- (2) FPGA 采样到 Dpam\_Store\_Flag = 0 时,从双口 RAM 读取前 2 个数据保存到 FSN/BSN 寄存器后,对 Dpam\_Store\_Flag 清 0。
- (3) FPGA 采样到 Hdlc\_Rdy\_Flag = 1, Cpu\_Sig\_Flag = 1 以及 Dpam\_Store\_Flag = 0 时,从双口 RAM 读取数据发送到 HDLC

的 FIFO。

(4) FPGA 采样到 Hdlc\_Rdy\_Flag = 1, Cpu\_Sig\_Flag = 0 以及 Dpam\_Store\_Flag = 0 时,从 FSN/BSN 寄存器读取数据发送到 HDLC 的 FIFO。

(5) FPGA 完成从双口 RAM 读取数据发送到 HDLC 的 FIFO 后,对 Cpu\_Sig\_Flag 和 Hdlc\_Rdy\_Flag 清 0,对 Hdlc\_Eop\_Flag 置 1。

(6) FPGA 完成从 FSN/BSN 寄存器读取数据发送到 HDLC 的 FIFO 后,对 Hdlc\_Rdy\_Flag 清 0,对 Hdlc\_Eop\_Flag 置 1。

(7) HDLC 采样到 Hdlc\_Eop\_Flag = 1 时,将 FIFO 中的数据全部发送到 E1 链路。

(8) 重复执行步骤(1)~步骤(7)。

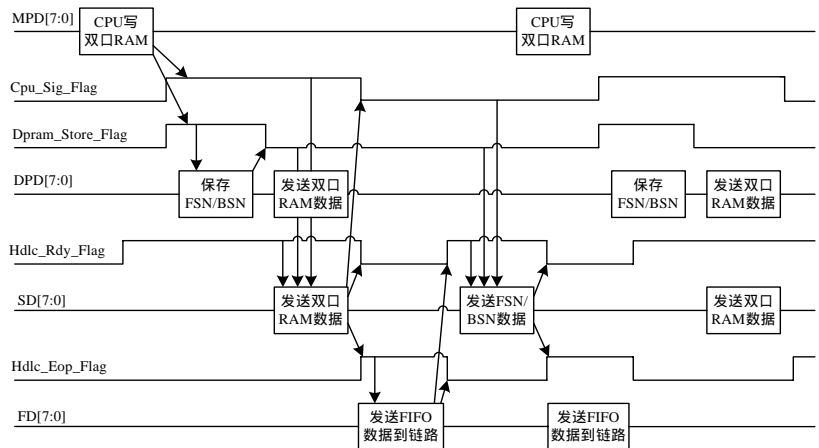


图 3 发送方向的信号流程及状态转换示意图

### 3.2 接收方向的 FISU 处理

图 4 是接收方向的信令处理原理框图,当收到 FISU 的 FIB/FSN 和 BIB/BSN 的值不变时,由 FPGA 直接过滤,不再送往 CPU。

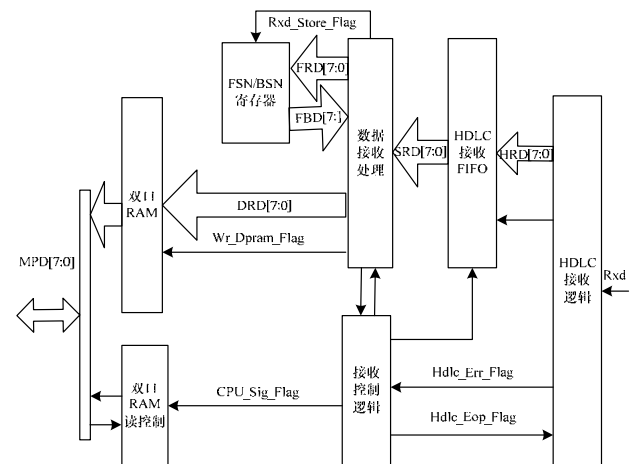


图 4 接收方向的信令处理原理框图

图 4 中每个标志位的状态变化说明如下:

- (1) Hdlc\_Eop\_Flag: HDLC 接收逻辑从链路接收最后一个字节后置 1,数据接收处理完成 LI 判断、FSN/BSN 的比较和出错处理后清 0。
- (2) Hdlc\_Err\_Flag: HDLC 接收逻辑检测到 CRC 校验错误时置 1,数据接收处理完成 LI 判断、FSN/BSN 的比较和出错处理后清 0。
- (3) Dpam\_Store\_Flag: 数据接收处理进行 FSN/BSN 的比较发现不相等时置 1,当存完当前的 FSN/BSN 值后清 0。

(4)Wr\_Dpam\_Flag :数据接收处理进行 FSN/BSN 的比较发现不相等、LI 值不等于 0 或 CRC 校验出错时置 1,在完成双口 RAM 的写数据后清 0。

(5)Cpu\_Sig\_Flag :完成双口 RAM 的写数据后置 1,CPU 从双口 RAM 读完数据后清 0。

图 5 是接收方向的信号流程及状态转换示意图,图中方框内的文字内容说明了当前的数据操作方式,箭头方向表示触发状态变换的条件。

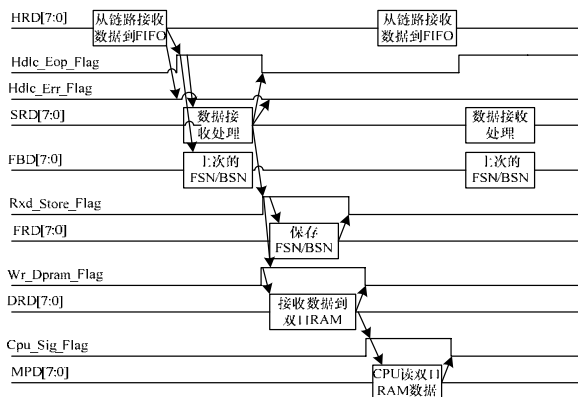


图 5 接收方向的信号流程及状态转换示意图

图 5 中信号流程和状态转换说明如下：

(1)HDLC 接收逻辑从链路连续接收数据并依次写入接收 FIFO,收完最后一个字节数据时,将 Hdlc\_Eop\_Flag 置 1,若检测到 CRC 出错,则同时将 Hdlc\_Err\_Flag 置 1。

(2)数据接收处理模块需要进行以下处理:1)从 FIFO 取出第 3 字节判断 LI 的值,若 LI 不等于 0,将 Wr\_Drpam\_Flag 置 1,否则判断条件 2;2)从 FIFO 取出第 1、第 2 字节,与 FSN/BSN 寄存器的值进行比较,若不相等,将 Wr\_Drpam\_Flag 和 Rxd\_Store\_Flag 置 1,否则判断条件 3;3)若 Hdlc\_Err\_Flag = 1,将 Wr\_Drpam\_Flag 置 1。

(3)当 Rxd\_Store\_Flag = 1 时,将 FIFO 的前 2 个字节保存到 FSN/BSN 寄存器,完成后对 Rxd\_Store\_Flag 清 0。当 Wr\_Drpam\_Flag = 1 时,需要对 Hdlc\_Err\_Flag 进行判断:1)若 Hdlc\_Err\_Flag = 1,直接将双口 RAM 的第 3 字节的最高位置 1,

并将 Cpu\_Sig\_Flag 置 1;2)若 Hdlc\_Err\_Flag = 0,将 FIFO 数据拷贝写入双口 RAM,并将 Cpu\_Sig\_Flag 置 1。

(4)CPU 读到 Cpu\_Sig\_Flag = 0 时,先读出第 3 字节数据进行判断:1)若最高位为 1,则不需要读出其他数据,直接将 Cpu\_Sig\_Flag 清 0;2)若最高位为 0,在双口 RAM 数据全部读出后将 Cpu\_Sig\_Flag 清 0。

(6)重复执行步骤(1)~到步骤(5)。

### 3.3 仿真分析和资源占用情况

本设计使用的 FPGA 芯片为 Xilinx 公司的 XCV600E-6HQ240C<sup>[4]</sup>,设计语言为 VHDL,编译综合环境为 ISE 8.2i,仿真工具为 ModelSim SE 6.2b。

仿真结果显示,发送方向的 FPGA 将前次保存的 FIB/FSN 和 BIB/BSN 作为头部自动发送 FISU,极大地节省了 CPU 发送 FISU 的时间开销。接收方向的 FPGA 将收到具有相同 FIB/FSN 和 BIB/BSN 的 FISU 直接过滤,极大地节省了 CPU 接收 FISU 的时间开销。

整个设计共使用了 1 690 个触发器、1 068 个逻辑单元和 16 个双口 RAM。

## 4 结束语

本文提出一种基于 FPGA 实现 FISU 过滤的设计和实现方案,可以自动发送和过滤接收到的 FISU,极大地节省了 CPU 对 FISU 处理的时间开销。该方案已应用于综合网关设备中,并且已产品化,经过和多个不同设备之间的互联测试,系统运行稳定可靠,证明该方案是切实可行的。本文只讨论了单路系统 FISU 的处理情况,若要处理多路,只需要并行添加多个相同模块即可。

### 参考文献

- [1] 陈建亚. 现代交换原理[M]. 北京: 人民邮电出版社, 2006.
- [2] 杨晋儒, 吴立贞. No.7 信令系统技术手册[M]. 北京: 人民邮电出版社, 1999.
- [3] 周盛雨, 孙辉先, 陈晓敏, 等. 实现 FPGA 回读功能的可重构系统设计[J]. 计算机工程, 2007, 33(12): 270-272, 274.
- [4] Xilinx Corp. Vertex-E FPGA Family Data Sheet[Z]. (2004-06-06). <http://www.xilinx.com>.

编辑 张正兴

(上接第 235 页)

(3)获取测量数据:构造设备命令,命令子 C1 表示获取数据。设备连续发送 24 h 的测量数据,由于数据较长,因此在测试结果中分屏显示。

(4)控制调制解调器,关闭连接:使用 AT 指令+++从数据状态切换到控制状态,并进行 ATH 的 Hang Up 操作。操作完毕后,调制解调器返回 OK。

将由同一接口的一个请求转换为 2 个设备命令组成的序列,并通过统一接口建立连接,完成了与设备通信的过程。在测试结果中,生成的命令被设备认可,结果格式也符合设备所规定的协议,从而验证了适配过程的正确性。

## 6 结束语

本文提出了一种基于适配的异构设备统一访问中间件。该中间件采用分层的架构设计思想,自下而上划分为物理连接适配、设备通信协议适配和设备统一访问接口,逐层屏蔽设备的异构性,实现远程控制和监测设备。在下一步的研究中,首先要在此基础上进一步实现设备资源服务化,将设备

封装为具有统一属性、状态、生命周期管理和访问控制接口的服务资源,其次是完善安全机制,引入分级安全体系以保证设备分布式控制管理中的安全性。

### 参考文献

- [1] 袁理想, 汪文勇, 罗光春. 设备网格技术的研究[J]. 计算机应用, 2005, 25(12): 2914-2915.
- [2] Spencer J B, Finholt T A, Foster I, et al. NEESgrid: A Distributed Collaboratory for Advanced Earthquake Engineering Experiment and Simulation[C]//Proc. of the 13th World Conference on Earthquake Engineering. Vancouver, Canada: [s. n.], 2004.
- [3] Donald M, Randall B. The Xport Collaboratory for High-brilliance X-ray Crystallography[EB/OL]. (2003-04-24). <http://www.cs.indiana.edu/ngi/sc2000/index.html>.
- [4] Ellisman M H, Lu Dai, Lamont S, et al. An Award Winning Methodology for Cyberinfrastructure Driven Collaborative Laboratories[EB/OL]. (2004-11-23). <http://telescience.ucsd.edu/about.shtml>.

编辑 张正兴