

基于 FPGA 的 E1 信号失帧检测研究

蔡绍伟,潘爱先,贺雪飞

(青岛理工大学 自动化工程学院,山东 青岛 266520)

摘要: 针对 E1 信号传输过程中出现的失帧问题进行研究,提出以 FPGA 为控制核心的检测方案,当信号出现失帧状况能及时进行检测并作出相应措施,以保障 E1 信号传输的稳定、流畅。

关键词: 失帧检测;E1;FPGA

中图分类号: TP399

文献标识码: A

文章编号: 1674-7720(2012)21-0062-03

Research on the loss frame detection of E1 signal based on FPGA

Cai Shaowei, Pan Aixian, He Xuefei

(College of Automation, Qingdao Technological University, Qingdao 266520, China)

Abstract: The article researches on the lost frame problem that occurred during the E1 signal transmission, and puts forward a detection scheme which applies FPGA as the control core. As soon as finding the loss frame situation, the system can make a timely detection and corresponding measures, in order to protect the E1 signal transmission stability and smooth.

Key words: loss frame detection; E1; FPGA

伴随着计算机技术和通信技术的飞速发展,E1 通信得到了极大的提升,应用范围日益广泛^[1],如国家安全电路、重大庆典、重大体育比赛的传播,银行、交易所等 DDN 数据传送等。E1 信号在传输过程中会出现失帧等状况,造成信号传输品质下降,严重时会导致信号缺失。因此如何迅速、精确检测出 E1 信号的失帧状况是 E1 传输中的研究重点。

近年来,FPGA 发展迅猛,在无线通信、光纤通信等通信领域,凭借其强大的运算处理能力和低成本,受到开发者的青睐。FPGA 的内部逻辑功能是通过向内部静态存储器单元加载配置数据来实现,其配置文件决定了逻辑单元的逻辑功能以及模块间或与 I/O 间的连接,而 FPGA 结构允许多次编程并享有快速有效地对新设计进行优化的灵活性^[2],所以,选用 FPGA 作为检测控制核心。本文结合 FPGA,针对 E1 信号传输过程中的失帧问题,设计了失帧检测的硬件电路及检测过程的判据流程,使 E1 信号的失帧检测更加快速、精确。

1 E1 概述

E1 是 30 路脉码调制 PCM 的简称,速率是 2.048 Mb/s。其帧结构如图 1 所示。

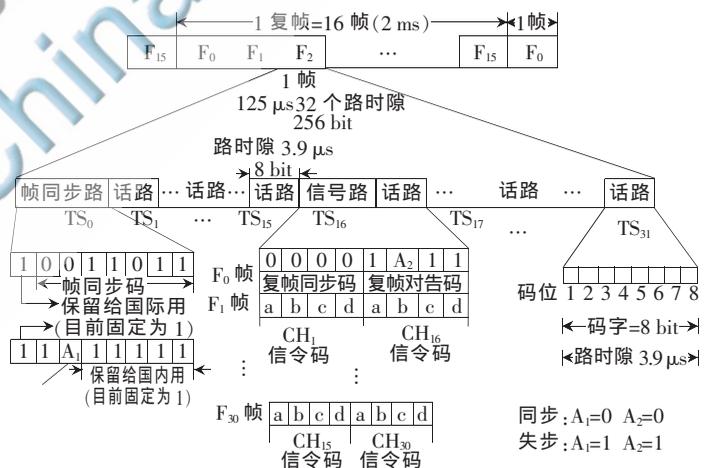


图 1 E1 帧结构图

图 1 中,在 E1 信道,8 bit 组成一个时隙,由 32 个时隙组成了一个帧,16 个帧组成一个复帧。在一帧中,TS₀ 主要用于传输帧定位信号、CRC-4 循环冗余校验及告警指示,TS₁₆ 主要用于传送随路信令、复帧定位信号,TS₁-TS₁₅ 和 TS₁₇-TS₃₁ 等 30 个时隙用于传输语音或数据等信息。

2 失帧检测原理

失帧检测的关键在于同步码组“0011011”的检测。

技术与方法 Technique and Method

同步码组只存在于偶帧的 TS_0 时隙中,所以检测时要将两组 E1 信号看成一组复帧。由于 E1 串行数据流中也会出现和同步码相同的内容,所以在设计中一般会采用计数器来进行帧同步码的定位以消除干扰。

E1 的帧周期为 $125 \mu\text{s}$,两帧即为 $250 \mu\text{s}$,规定在捕捉到第一个同步序列开始,若在之后的一段时间内每隔 $250 \mu\text{s}$,可连续三次(计数器计数)捕捉到同步序列时,则认为系统处于帧同步状态^[3]。同理,若在一段时间内如果连续三次都没有捕捉到帧同步序列,则可认为系统帧同步丢失,所以,帧失步倒换时间为 $3 \times 250 \mu\text{s} = 750 \mu\text{s}$ 。其检测原理框图如图 2 所示。

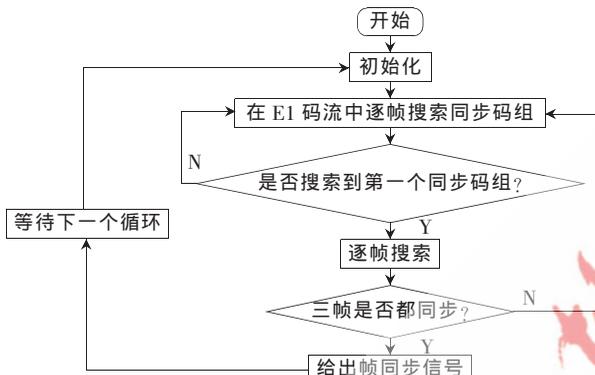


图2 帧同步信号检测流程图

3 硬件电路设计

系统设计基于 FPGA 的帧同步原理图如图 3 所示。

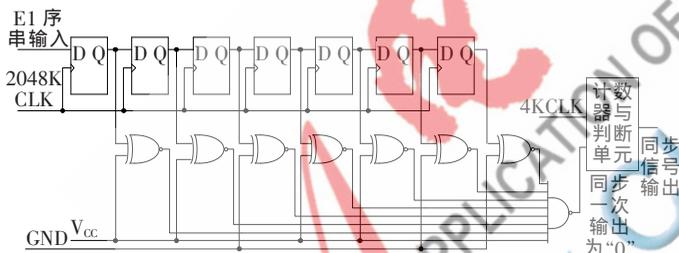


图3 失帧检测原理电路

图3的失帧检测电路由7个D触发器、7个异或非门和一个8输入与非门构成,该电路可以检测出E1信号序中串行输入数据流中包含的特殊码字“0011011”,其中利用地线(GND)和电源线(V_{CC})可将相关运算阵列的一个输入自右向左连接成“0011011”,与同步码字对应的另一个输入端接输入序列移位寄存器的7个输出的对应位进行异或非(同或)运算,对应位匹配时结果为“1”。7个异或非门的运算结果进入求和网络后,只有当7位对应位全都匹配时,捕捉同步信号才有效(有效状态为“0”),此时表明找到了一次同步序列码^[4]。在经过计数与判断单元,对每次捕捉的同步状态进行计数,若连续3次捕捉到同步码,则说明E1码流没有出现失帧;反之,则认为失帧。

4 失帧检测判据流程

帧同步系统的流程图如图4所示。

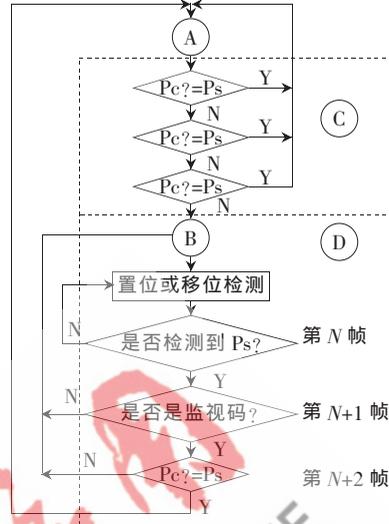


图4 失帧检测判据流程图

图4中A为同步状态信号,表示收发两端的工作状态同步;B表示宣告失步,帧同步系统进入捕捉状态;C所在的虚框表示前方保护计数流程,其作用是防止假失步;D所在的虚方框表示后方保护计数的流程,其作用是防止假同步; P_s 为帧同步码组的检出标志,只有一位脉宽。当 $P_s=0$ 时说明由信道而来的数字码流包括 $P_s=0,1$ bit在内的前7 bit为“0011011”码组^[5]; P_c 为接收端产生的比较标志,在同步状态时,由接收端定时电路在偶帧 TS_0 的D8位出现一次50%占空比的正脉冲 P_c 。

当帧同步处于流程图状态时,表明在预定的时刻已经连续检出了帧同步码组的标志 P_s ,即 $P_c=P_s$ 。如果起初还没有建立E1信号起收发之间的同步,或者由于其他原因,当同步系统连续三次在预定的时刻^[6](该时刻为 TS_0 偶 $\cdot D8$ 时刻,不以发端为准,只有同步时刻才是发端偶帧的 TS_0 偶 $\cdot D8$)没有检出同步码标志,即 $P_c \neq P_s$,则宣告失步(符号B)。同步系统由前方保护计数状态C进入到捕捉状态(应注意,在前方保护计数的这一过程中,E1信号仍处在同步工作的状态,只有连续三次不出现 $P_c=P_s$ 的情况,才宣告失步,进入捕捉状态)。进入捕捉状态后,帧同步系统将开始在接收到的E1数字码流中搜索同步码组。若检测出“0011011”码组,帧同步系统将启动定时电路并同时进入后方保护计数D^[7]。在捕捉状态,若出现 $P_s=0$ 的情况,则认为捕捉到了同步码组,并认为 $P_s=0$ 的时刻是偶帧 TS_0 的D8位,启动定时系统。隔 $125 \mu\text{s}$ 便是假设的 $N+1$ 帧的 TS_0 时隙,在 TS_0 时隙检查 $b_2 \neq 1$,若 $b_2=1$,则说明有监视码,上帧的同步码可能为真,继续检测 $N+2$ 帧;反之, $b_2=0$,表明上帧同步码组为伪同步码组,返回B,重新置位捕捉。若在 $N+2$ 帧有 $P_c=P_s$ 出现,表明假设的 N 帧捕捉的帧同步码组符合周期出现的规律,为真同步码组。帧同步系统进入同步状态A,开始正常工作;若在 $N+2$ 帧没有 $P_c=P_s$ 出现,则认为同步码组的出现不符合规律,帧同步系

技术与方法 Technique and Method

统重新进入捕捉状态 B。

该研究方案的运用能够及时检测出 E1 信号传输过程中出现的失帧状况,并对其作出精确判断以供后续处理。对于 E1 信号的收、发两端之间同步状态的建立和保护有着一定的实际应用价值。

参考文献

- [1] 桑林,郝建军,等.数字通信[M].北京:北京邮电大学出版社,2002.
- [2] KILTS S.高级 FPGA 设计[M].孟宪元,译.北京:机械工业出版社,2009.
- [3] 李刚.数字信号处理器的原理及其开发应用[M].天津:天津大学出版社,2001.
- [4] NAKANISHI Y, MATSUSHITA Y, ULSUMI K, et al.

Optical Transmi-ssion Network System NEC Technology[M]. 1997.

- [5] ARNOLD M. Verilog digital computer design: algorithms tohardware[M]. NJ:Prentice Hall,1998.
- [6] 牟文波,刘橙,等.数字通信中 2M 接口电路帧结构及常见故障分析[J].黑龙江科技信息,2010,(34).
- [7] MYNBAEV D K, et al. Fiber -Optic communicatios Technology[M]. 2000.

(收稿日期:2012-07-06)

作者简介:

蔡绍伟,男,1987 年生,在读研究生,主要研究方向:控制理论与控制工程。

