

多通道ARINC429总线的设计与实现

杨 高, 彭良福*

西南民族大学电子信息学院, 四川 成都

收稿日期: 2022年12月11日; 录用日期: 2023年1月14日; 发布日期: 2023年1月19日

摘 要

为了解决传统ARINC429总线电路设计中采用进口协议芯片带来的占用面积大、不便灵活扩展等问题, 提出一种基于国产FPGA的ARINC429总线协议的设计方案, 并结合国产驱动芯片, 设计了多通道ARINC429的硬件原理图。利用硬件描述语言VerilogHDL进行了发送部分和接收部分的总线协议的设计, 并且对其进行了仿真和验证。除FPGA外, 外围电路只需要搭配国产总线驱动器, 简化了PCB电路的设计, 节省了板卡面积, 可灵活扩展ARINC429总线通道, 有效地提升了数据通信模块的处理能力和集成度。

关键词

ARINC429总线, 国产化, 多通道

Design and Implementation of Multi-Channel ARINC429 Bus

Gao Yang, Liangfu Peng*

School of Electronic Information, Southwest Minzu University, Chengdu Sichuan

Received: Dec. 11th, 2022; accepted: Jan. 14th, 2023; published: Jan. 19th, 2023

Abstract

In order to solve the problems of large occupied areas and inconvenient and flexible expansion caused by the use of imported protocol chips in traditional ARINC429 bus circuit design, a design scheme of ARINC429 bus protocol based on domestic FPGA is proposed, and combined with domestic driver chips, a multi-Hardware schematic of channel ARINC429. Using hardware description language Verilog HDL, the design of the bus protocol of the sending part and the receiving part is carried out, and it is simulated and verified. In addition to FPGA, peripheral circuits only need to

*通讯作者。

be equipped with domestic bus drivers, which simplifies the design of PCB circuits, saves board area, flexibly expands ARINC429 bus channels, and effectively improves the processing capability and integration of data communication modules.

Keywords

ARINC429 Bus, Localization, Multi-Channel

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

ARINC429 是一种标准的高速差分信号通信总线, 在民用和军用航空领域被广泛应用[1]。例如波音 727、737、747、757 和 767 飞机, 空客 A310/A320、A330/A340 飞机等。目前 ARINC429 总线接口一般都采用专用进口芯片[2], 配合 ARINC429 数据收发芯片(如 HI-3282)和 ARINC429 数据发送驱动芯片(如 HI-3182), 形成 2 个数据接收通道和 1 个数据发送通道[3]。这些专用进口芯片价格昂贵, 且通道有限, 使用非常不灵活。根据 ARINC429 总线协议, 结合目前存在的问题, 我们在 FPGA 中通过设计模块化、可配置的 ARINC429 总线协议, 灵活扩展 ARINC429 通信通道, 降低研发成本, 并在 FPGA 外围配合 ARINC429 驱动芯片, 可广泛应用于各种使用 ARINC429 总线的航空电子、航天电子、汽车电子以及工业控制等场合[4]。

2. 协议设计

2.1. ARINC429 协议简介

ARINC429 总线传输的是双极性归零码信号[5], 数据信号是一对差分信号格式[6], 由专用驱动芯片驱动到总线进行电平转换, 转换后的 TTL 电平才能与 FPGA 进行接口[7], 对应转换后的 TTL 电平如图 1 所示的 OUT_A、OUT_B, 其中连续传输的字与字之间至少有 4 个位的时间间隔, 以便于发送下一组数据。

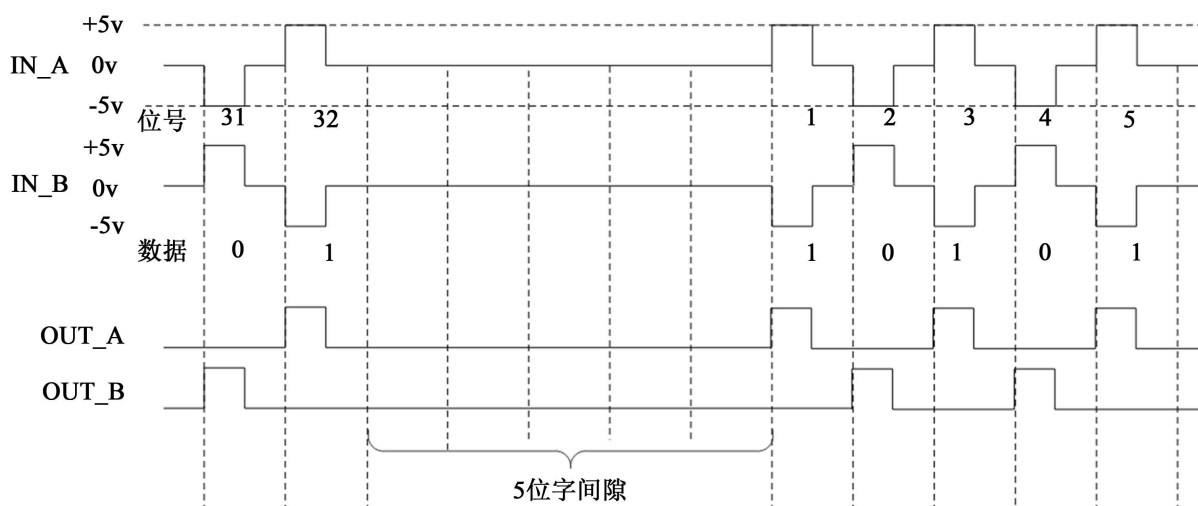


Figure 1. ARINC429 level conversion diagram

图 1. ARINC429 电平转换示意图

本文中基于 FPGA 的 ARINC429 协议设计包含发送部分与接收部分, 两者均可通过对逻辑可重用模块寄存器的配置, 实现发送与接收协议的独立工作, 完成高速、低速两档的位速率设置、奇偶校验、中断控制等功能的配置。

2.2. ARINC429 发送协议逻辑设计

发送部分主要是将总线上的信号转换成 TTL 电平差分串行码的转换[8]。根据 ARINC429 协议, 每一位数据前半周期为 TTL 的具体值, 后半周期同时输出为 0 [9]。其中, 发送部分由总线接口、异步 FIFO 设计、奇偶校验、寄存器控制、并串转换等模块组成。发送协议的 Verilog 设计经过 Vivado 综合后的 RTL 模块原理图如图 2 所示。

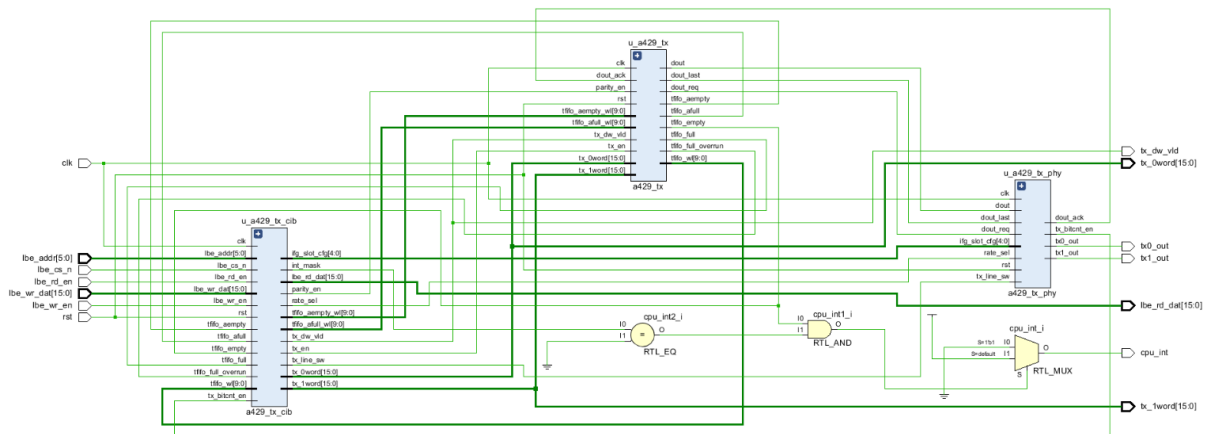


Figure 2. Schematic diagram of ARINC429 sending protocol RTL module

图 2. ARINC429 发送协议 RTL 模块原理图

发送部分的工作过程: 通过 LBE 总线将 tx0_word[15:0]和 tx1_word[15:0]锁存至寄存器 fifo_wdata_latch[31:0]并形成 FIFO 的输入数据信号 txfifo_in[31:0], 在写时钟 txfifo_wr_clk 下, 控制写使能 txfifo_wr_en 有效, 将数据写入 FIFO 中缓存, 直到满信号 txfifo_full 信号告警, 便不再写入数据。通过在读时钟 txfifo_rd_clk 下, 控制读使能 txfifo_rd_en 有效、发送使能信号 tx_en 有效, 输出 FIFO 中缓存的单 bit 数据 dout, 直到空信号 txfifo_empty 告警, 便不再继续输出数据。默认设置端口物理交换信号 tx_line_sw 为 1'b1, 当 dout 输出为 1, tx0_out 和 tx1_out 的前半周期为数据位, 分别输出 1 和 0, 后半周期同时输出为 0; dout 输出为 0 时, tx0_out 和 tx1_out 的前半周期分别输出 0 和 1, 后半周期同时输出为 0。每次 tx0_out、tx1_out 输出之后, 会有一个 dout_ack 输出响应有效信号。每个 arinc429 字由 32 位构成, 格式如下表 1 所示。在 32 位数据传输过程中, 遵循以下传输协议: 先发送第 1 位, 然后依次发送至 32 位, 先传标号位, 后传数据。当传输数据时, 先传输最低位, 但标号位应先传最高位, 后传最低位。

Table 1. ARINC429 word format composition

表 1. ARINC429 字格式构成

位号	32	31	30	29	11	10	9	8	1
功能	奇偶位	符号状态位	数据位	SD					标号位

Arinc429 协议的发送部分工作过程中, 发送状态机是其中的核心, 发送协议的状态转移图如图 3 所示。通过 FIFO 的控制读使能 txfifo_rd_en 有效信号和发送使能信号 tx_en 有效信号产生一个数据有效信

号 data_vld, 当 data_vld 为 1 的时候, 从 START 状态进入奇偶校验状态 PARITY, 奇偶校验状态会将数据的前 31 位进行按位异或操作, 通过判断出“1”的个数, 奇偶校验状态选择奇偶校验电路生成的校验位(第 32 位)进行输出。完成奇偶校验之后进入数据发送状态 TX_DATA, 数据发送状态下, 32 位的发送数据按照发送要求, 被串行顺序发出, 当 dout 输出为 1, tx0_out 和 tx1_out 的数据位(即前半周期)分别输出 1 和 0; dout 输出为 0 时, tx0_out 和 tx1_out 的数据位分别输出 0 和 1。通过 dout_ack 输出响应有效信号, 判断当前 32 位数据是否发送完毕, 发送完毕进入发送完成状态 FINISH, 发送完成状态通过等待 5 位中电平 0V 的时间(即 5 bit 时间), 此时 tx0_out 和 tx1_out 均输出 0, 之后状态机再回到开始状态 START。

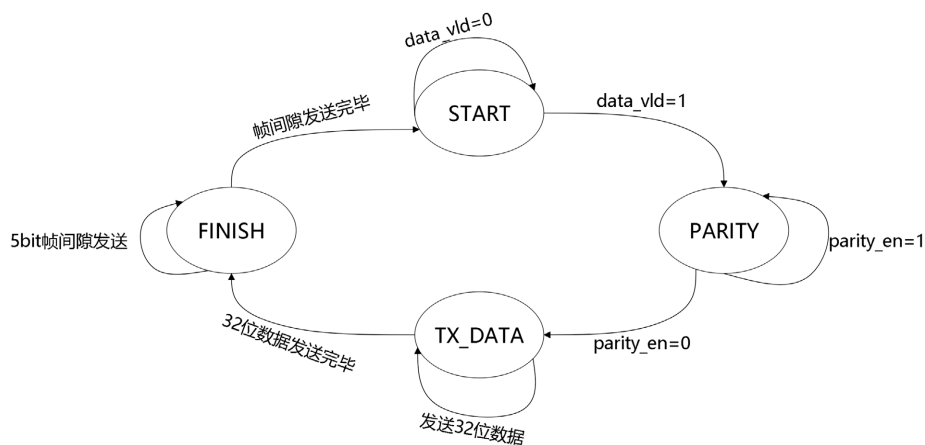


Figure 3. Send state machine state transition diagram

图 3. 发送状态机状态转换图

其中 rate_sel 为发送率, 为“1”时选择 12.5Kbps, 为“0”时选择 100Kbps; parity_en 为奇偶检验使能, 为“1”时表示奇校验使能, 为“0”时表示不使能; tx_line_sw 为端口物理交换信号, 为“1”时表示选择 tx0_out、tx1_out 分别为高、低电平, 为“0”时表示选择 tx0_out、tx1_out 分别为低、高电平; ifg_slot_cfg 为帧间隙, 发送完一帧之后, 默认保持 5 位中电平 0V 的时间。ARINC429 发送端逻辑设计的重要部分信号如表 2 所示。

Table 2. ARINC429 transmitter signal definition

表 2. ARINC429 发送端信号定义

符号	I/O	功能
tx_en	input	tx 方向模块使能。1 表示使能, 0 表示不使能
rate_sel	input	tx 方向位速率选择。1 表示 12.5Kbps; 0 表示 100Kbps
parity_en	input	tx 方向校验使能。1 表示使能; 0 表示不使能
tx_line_sw	input	tx 方向端口物理交换信号。默认值为 1
ifg_slot_cfg	input	tx 最小帧间隙, 表示多少个比特时间, 默认为 0x5
counter_tx_bitcnt	output	tx 发送比特统计
int_mask	input	tx 方向 FIFO 空中断屏蔽。1 表示屏蔽中断; 0 表示不屏蔽中断
tx0_word	input	tx 方向写入 16 位 bit 数据帧发送数据
tx1_word	input	tx 方向写入 16 位 bit 数据帧发送数据
tx0_out	output	tx 方向串行发送 TTL 电平差分码, 表示 TTL “1”

Continued

tx1_out	output	tx 方向串行发送 TTL 电平差分码, 表示 TTL “0”
tx_word	output	tx 方向 32 位并行发送数据
tx_dw_vld	output	tx 方向数据发送有效信号

2.3. ARINC429 接收协议逻辑设计

接收部分输入端接入两路串行数据 rx0_in、rx1_in (即 TTL “1”、TTL “0”), 经过源/目的地识别码校验、符号状态码识别、奇偶校验等过程, 转换成 32 位的并行数据字, 以供上位机读取。其中, 接收部分由总线接口、异步 FIFO 设计、奇偶校验、寄存器控制、串并转换等模块组成。

接收部分主要完成数据接收、数据校验、数据写入。

数据接收工作过程: 通过 LBE 总线控制接收使能信号 rx_en, 启动数据接收, 且从 5 位字间隙判断到数据起始位, 此时寄存器根据设置的位速率, 开始数据接收, 其中接收端采样输入端 rx0_in、rx1_in 信号的前半周期数据, 按照 rx0_in 的前半周期为 1、rx1_in 的前半周期为 0, 接收端输出为 1、rx0_in 的前半周期为 0、rx1_in 的前半周期为 1, 接收端输出为 0 的要求, 依次进行串行接收 32 位数据, 接收完成后完成串并转换, 得到 32 位并行数据 rx_data[31:0]。将得到的 32 位并行数据重新组合, 即调整 rx_data[31:0] 的低八位标号位数据, 以此正确还原接收的数据, 具体调整为 rx_fifo_wdata<={rx_data[31:8],rx_data[0],rx_data[1],rx_data[2],rx_data[3],rx_data[4],rx_data[5],rx_data[6],rx_data[7]}。

数据校验工作过程: LBE 总线控制奇偶校验使能信号 parity_en, 第 32 位的数据, 会根据发送端的低 31 位数据, 判别出相应的奇偶校验位值 rx_p, 并与发送端的奇偶检验位 tx_data_p 进行比对, 产生一个接收部分的奇偶校验比对信号 p_bit_ind。p_bit_ind 为 “1”, 表示比对结果一致, 反之不一致; LBE 总线控制源/目的识别码过滤使能信号 sdi_filter_vector, 将接收的 ARINC429 数据 rx_data[9:8]与发送端对应的 2 位数据进行比对, 产生一个 SDI 比对信号 sdi_match, sdi_match 为 “1”, 表示接收的 2 位数据与发送端的一致, 反之不一致; LBE 总线控制标号位过滤使能信号 label_filter_en, 将接收的 ARINC429 标号位数据与发送端对应的 8 位数据进行比对, 产生一个 label 比对信号 label_match。label_match 为 “1”, 表示接收的 8 位数据与发送端的一致, 反之不一致。

数据写入工作过程: 数据校验模块产生的三个比对信号均为 “1” 且数据没有写满的情况下, 通过设置 Label 过滤操作偏移地址 label_filter_offset 的最大偏移量, 计算出正确的接收地址, 完成 32 位并行数据的写入。其中 ARINC429 接收端逻辑设计的重要部分信号如表 3 所示。

Table 3. ARINC429 receiver signal definition

表 3. ARINC429 接收端信号定义

符号	I/O	功能
rx_en	input	rx 方向模块使能。1 表示使能, 0 表示不使能
label_filter_en	input	label 过滤使能。1 表示使能; 0 表示不使能
parity_en	input	rx 方向校验使能。1 表示使能; 0 表示不使能
sdi_filter_en	input	SDI 过滤使能。1 表示使能; 0 表示不使能
signal_err	output	rx 方向接收信号异常告警, 通常是外部物理芯片异常
counter_rx_bitcnt	output	rx 接收比特统计
counter_frame_drop_ind	output	rx 方向接收过滤 SDI 或者 Label 丢帧统计

Continued

int_mask	input	rx_fifo 非空中断屏蔽寄存器。1 表示屏蔽; 0 表示不屏蔽
rx_ren	input	rx 方向数据帧获取使能, 先写 1, 再写 0
rx0_in	input	rx 方向串行发送 TTL 电平差分码, 表示 TTL “1”
rx1_in	input	rx 方向串行发送 TTL 电平差分码, 表示 TTL “0”
cur_alm_rx_word	output	rx 方向接收的并行数据帧
sdi_filter_vector	input	SDI 过滤向量, 1 表示过滤; 0 表示不过滤。
label_filter_offset	input	Label 过滤 RAM 操作偏移地址
label_filter_vi	input	Label 过滤向量, 1 表示过滤; 0 表示不过滤
rx_dw_vld	output	rx 方向数据发送有效信号

3. 硬件电路设计

3.1. 改进方案

传统的 ARINC429 电路是通过进口的协议芯片(如 HS-3282)加驱动芯片(如 HS-3182)实现[10], 如图 4 所示。HI-3282 实现数据接收和发送[11], HI-3182 实现总线发送驱动功能[12]。局限在于 HI-3282 协议芯片只有 1 路发送通道, 2 路接收通道, 使得通道通信数量有限, 且扩展不方便。改进的通信电路实现方案如图 5 所示。通过在 FPGA 设计 ARINC 协议, 以此替代进口的协议芯片[13], 节省板卡面积、克服协议芯片单通道通信, 特别是当需要多个 ARINC429 通道通信时, 可灵活扩展发送通道数量。SM8444 接收器芯片可同时接入 4 通道通信, FPGA 型号采用国产芯片复旦微 FMQL45T900, 驱动芯片采用国产国徽芯片(SM8444 与 SM0429), 从而解决了 ARINC429 的国产化问题。

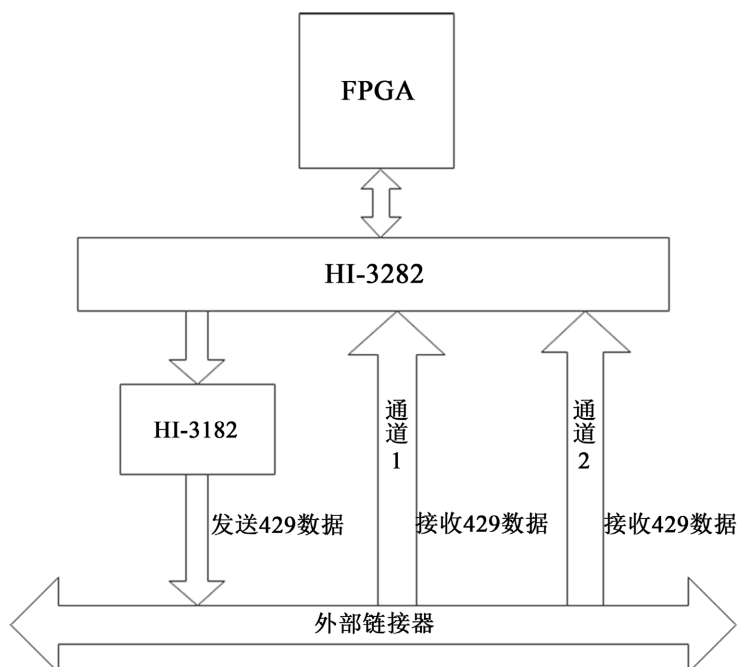


Figure 4. Traditional ARINC429 communication circuit scheme

图 4. 传统型 ARINC429 通信电路方案

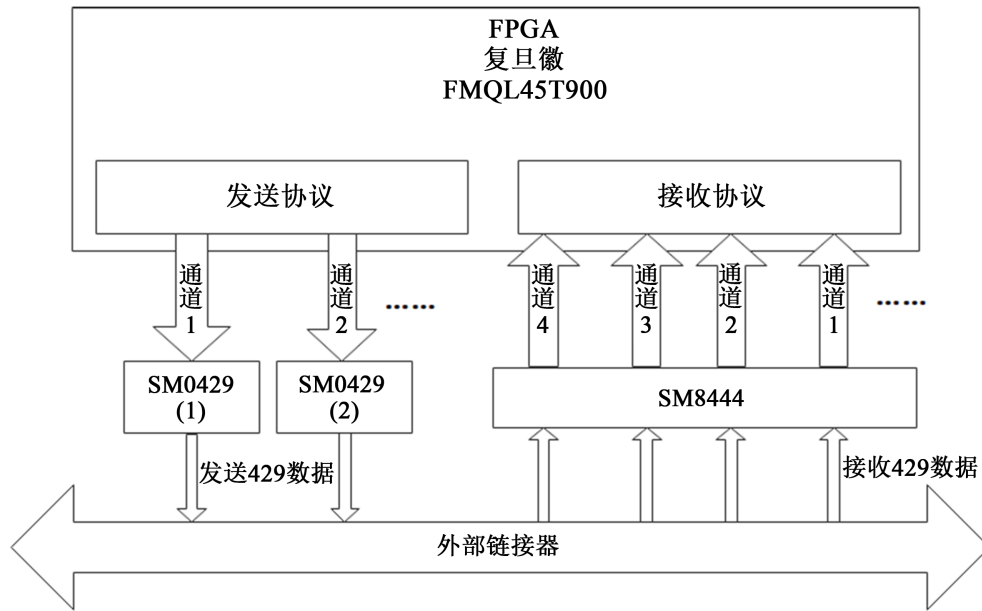


Figure 5. Multi-channel localized improved ARINC429 communication circuit scheme
 图 5. 多通道国产化改进型 ARINC429 通信电路方案

3.2. 硬件电路设计

ARINC429 数据发送：将控制系统 FPGA 输入的串行数据分别从 DATA(A)和 DATA(B)引脚输入，芯片通过 SYNC 和 CLOCK 信号使能，输出电压幅度可通过 VREF 进行调节。典型情况下，VREF = V1 = 5 V。因此，单端输出+5 V、-5 V 和 0 V 电压，差分输出电压为 10 V。ARINC429 总线发送器 SM0429 与 FPGA 硬件电路设计如图 6 所示。

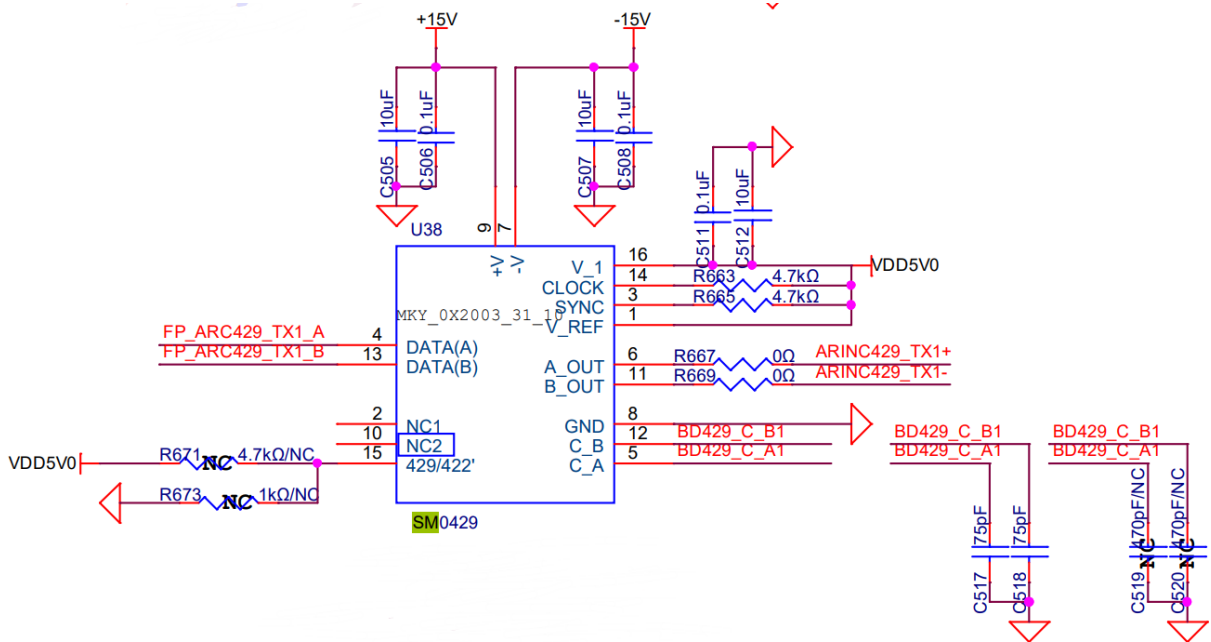


Figure 6. ARINC429 bus transmitter and FPGA hardware design diagram
 图 6. ARINC429 总线发送器与 FPGA 硬件设计图

ARINC429 数据接收: SM8444 是一款四通道 ARINC429 接收器, 接收来自 ARINC429 总线上的信号, 并转换成 CMOS/TTL 信号输出。通过线缆与 ARINC429 总线发送器相连。ARINC429 总线发送器负责发送信号[14], 通过线缆传输到 SM8444 接收端, 经过电路处理转换成 CMOS/TTL 信号输出并传输到 FPGA [15]。ARINC429 总线接收器 SM08444 与 FPGA 硬件电路设计如图 7 所示。

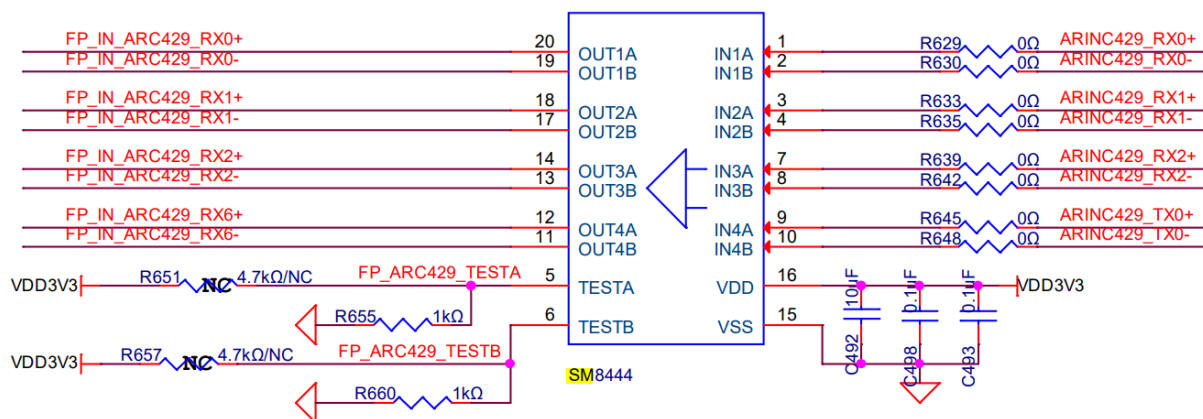


Figure 7. ARINC429 bus receiver and FPGA hardware design diagram
图 7. ARINC429 总线接收器与 FPGA 硬件设计图

4. 仿真验证

本设计是基于 Vivado 和 questasim 进行仿真验证, 通过编写 testbench, 设置 rate_sel 寄存器为 1'b0, 选择位速率为高速 100Kbps; 设置 parity_en 寄存器为 1'b1, 选择奇校验使能; 设置 tx_en 寄存器为 1'b1, 开启发送使能; 设置 ifg_slot_cfg[4:0]寄存器为默认的 5'd5, 选择最小帧间隙为 5bit 等待时间。各个寄存器设置, 如发送部分数据第一次写入仿真波形图 8 矩形标注的 Msgs 一栏中的值, 与 testbench 代码各个寄存器设置的一致。要发送的 32 位 ARINC429 数据分为 2 路各 16 位, 高 16 位 tx1_word、低 16 位 tx0_word 分别由 LBE 总线协议, 通过 lbe 总线数据写入信号 lbe_wr_dat, 写入第一次数据 16'h1234, 16'h5678; 第二次写入数据 16'h90ab, 16'hcdef。发送部分第一次写入数据仿真波形和第二次写入数据仿真波形分别如图 8 及图 9 所示。从图 8 可以看到第一次向 tx1_word 信号输入了 16'h1234, tx0_word 输入了 16'h5678; 从图 9 可以看到第二次向 tx1_word 信号输入了 16'h90ab, tx0_word 输入了 16'hcdef。

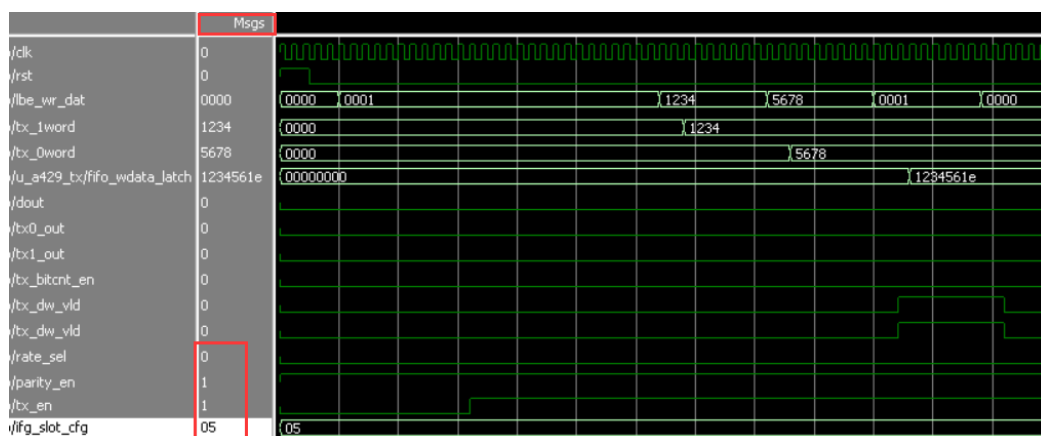


Figure 8. ARINC429 sends part of the first write data simulation waveform
图 8. ARINC429 发送部分第一次写入数据仿真波形

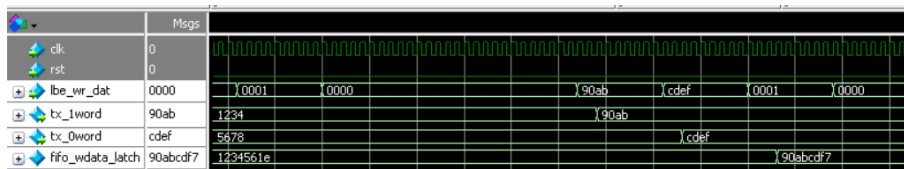


Figure 9. ARINC429 sends part of the second write data simulation waveform
 图 9. ARINC429 发送部分第二次写入数据仿真波形

在向 tx1_word 和 tx0_word 写入数据之后, 写入的数据需要存入 FIFO 中, 等待开启发送使能有效信号 tx_en 之后, 发送存入的 32 位数据。而需要注意, 发送数据需要遵循以下传输协议: 先发送第 1 位, 然后依次发送至 32 位, 先传标号位, 后传数据。当传输数据时, 先传输最低位, 但标号位应先传最高位, 后传最低位。所以存入 FIFO 的 32 位数据 fifo_wdata_latch[31:0]需要按照传输的发送协议, 对 8 位标号位数据 tx_0word[7:0]进行调整, 即 $fifo_wdata_latch \leftarrow \{tx_1word[15:0], tx_0word[15:8], tx_0word[0], tx_0word[1], tx_0word[2], tx_0word[3], tx_0word[4], tx_0word[5], tx_0word[6], tx_0word[7]\}$, 这样才能保证按照发送协议进行正确传输 32 位数据。从图 8 可以看到第一次存入 FIFO 的 32 位数据 fifo_wdata_latch[31:0]为 32'h1234561e, 转为二进制格式为 32'b0001_0010_0011_0100_0101_0110_0001_1110, 将 tx_1word[15:0]的 16'h1234 转为二进制格式为 16'b0001_0010_0011_0100, tx_0word[15:0]的 16'h5678 转为二进制格式数据为 16'b0101_0110_0111_1000。将 fifo_wdata_latch 与 tx_1word 和 tx_0word 的二进制格式数据对比分析, 可以看到, fifo_wdata_latch[31:0]数据的低八位为 8'b0001_1110, 这 8 位是标号位数据, 是将 tx_0word[15:0]低八位的数据 8'b0111_1000 进行高低位转换得到, 这样既满足了先传标号位的要求, 又满足了标号位先传最高位, 后传最低位的要求。

发送第一次和第二次写入的数据如图 10 和图 11 所示, 从图 10 可以看到单 bit 数据输出信号 dout 的波形图, 从波形的电平起始可以分析出, 在系统时钟 clk 下, 依次是 0111_1000_0110_1010_0010_0011_0100_1000, 共 32 位数据。正是按照 fifo_wdata_latch[31:0]数据 32'h1234561e 的二进制格式: 32'b0001_0010_0011_0100_0101_0110_0001_1110, 按照先传输低位要求, 依次串行输出 fifo_wdata_latch 的 32 位数据, 而两路 TTL 电平差分输出信号 tx0_out 和 tx1_out 则是严格按照单 bit 数据输出信号 dout 进行转换输出, dout 串行输出的第 1 位为 0, 此时在图 10 标注的数字 1 后面, tx0_out 输出前半周期为 0, 后半周期为 0, tx1_out 输出前半周期为 1, 后半周期为 0。dout 串行输出的第 2 位为 1, 此时在图 10 标注的数字 2 后面, tx0_out 输出前半周期为 1, 后半周期为 0, tx1_out 输出前半周期为 0, 后半周期为 0。两路 TTL 电平差分输出信号按照前半周期输出数据, 后半周期输出同时为 0 要求, 依次串行输出。至 dout 串行输出的 32 位为 0, 此时在图 10 标注的数字 32 后面, tx0_out 输出前半周期为 0, 后半周期为 0, tx1_out 输出前半周期为 1, 后半周期为 0, 同时此刻发送数据完成有效信号 tx_dw_vld 产生一个高电平, 代表 32 位数据发送完成。至此输入的 32 位数据发送完毕, 两路 TTL 电平差分输出信号 tx0_out 和 tx1_out 输出与预期波形一致, 数据发送成功。在数据发送完毕之后, tx0_out 和 tx1_out 均输出为 0, 即通过保持 5 位中电平 0V 的时间(即 5 bit 时间), 保持时间如图 10 标记的两个紫色箭头。之后两路 TTL 电平差分输出信号 tx0_out 和 tx1_out 再发送第二次写入的数据 fifo_wdata_latch[31:0]为 32'h90abcd7, 第二次数据发送仿真波形如图 11 所示, 可以看到 tx0_out 和 tx1_out 输出与预期波形一致, 数据发送成功。

在 Vivado 平台, 将设计好的发送协议与接收协议搭建回环测试并编写 testbench, 将 ARINC429 总线发送的串行数据接入接收协议的输入端 rx0_in、rx1_in, 如上设置位速率为高速 100Kbps、奇校验使能、接收使能、最小帧间隙 5 bit。在接收使能信号 rx_en 开启之后, 接收端采样输入端 rx0_in、rx1_in 的前半周期数据, 按照 rx0_in 的前半周期为 1、rx1_in 的前半周期为 0, 接收端输出为 1、rx0_in 的前半周期为 0、rx1_in

的前半周期为 1, 接收端输出为 0 的要求, 依次串行接收并经过串并转换后, 将得到的数据重新组合, 即调整 rx_data[31:0] 的低 8 位, 以此正确还原接收的数据, 具体调整为 rx_fifo_wdata<={rx_data[31:8],rx_data[0],rx_data[1],rx_data[2],rx_data[3],rx_data[4],rx_data[5],rx_data[6],rx_data[7]}, 第一次接收处理后得到 32 位并行数据 rx_fifo_wdata 为 32'h12345678, 分别为高 16 位 cur_alm_rx_1word 数据 16'h1234, 低 16 位 cur_alm_rx_0word 数据 16'h5678。第一次接收部分的仿真波形如图 12 所示(部分截图), 与第一次发送的数据一致, 数据接收成功。在接收完第一次发送的数据之后, rx0_in、rx1_in 均保持为 0, 即通过保持 5 位中电平 0 V 的时间(即 5 bit 时间), 保持时间如图 12 标记的两个紫色箭头。之后两路 TTL 电平差输入信号 rx0_in、rx1_in 再接收第二次发送的数据。第二次数据发送仿真波形如图 13 所示, 可以看到第二次接收处理后得到 32 位并行数据 rx_fifo_wdata 为 32'h90abcdef, 分别为高 16 位 cur_alm_rx_1word 数据 16'h90ab, 低 16 位 cur_alm_rx_0word 数据 16'hcdef, 与第二次发送的数据一致, 数据接收成功。

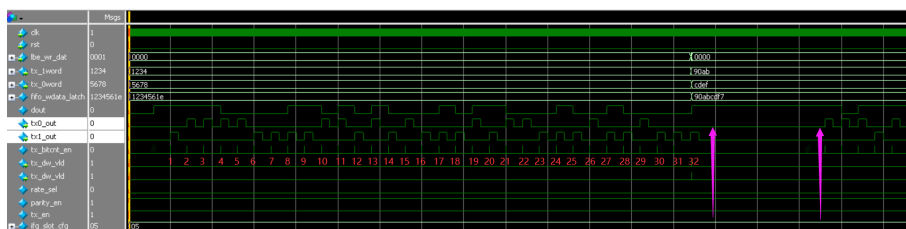


Figure 10. ARINC429 first data transmission simulation waveform

图 10. ARINC429 第一次数据发送仿真波形

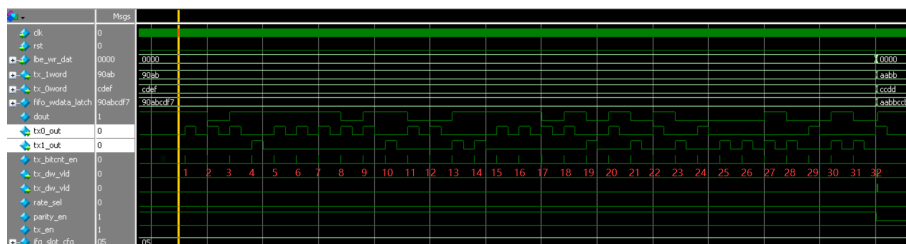


Figure 11. ARINC429 second data transmission simulation waveform

图 11. ARINC429 第二次数据发送仿真波形

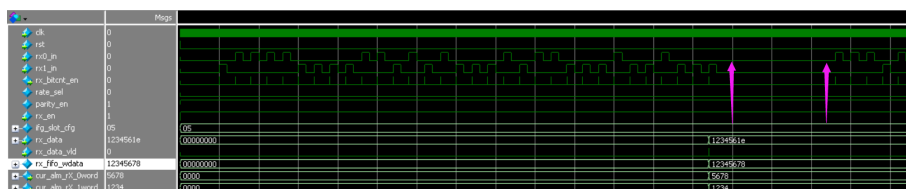


Figure 12. ARINC429 first data receiving simulation waveform

图 12. ARINC429 第一次数据接收仿真波形

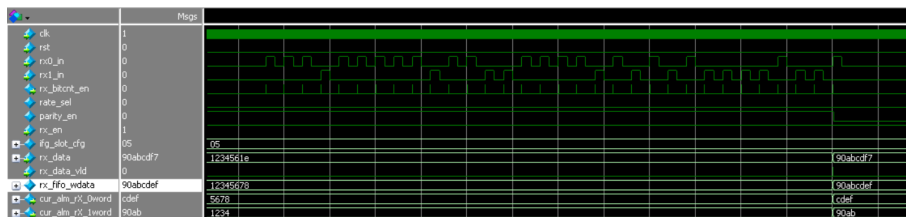


Figure 13. ARINC429 second data receiving simulation waveform

图 13. ARINC429 第二次数据接收仿真波形

5. 结论

本文 ARINC429 总线协议的设计基于 Verilog HDL 语言, FPGA 型号使用了复旦微国产的 FMQL45T900、ARINC 驱动器采用国徽国产芯片 SM8444 与 SM0429。通过仿真验证, 结果表明 ARINC429 总线协议能够实现多通道数据的收发, 设计符合 ARINC429 总线数据传输要求, 该电路可应用于各种使用 ARINC429 总线的场合。

综上所述, 基于国产 FPGA 设计的多通道 ARINC429 总线有如下特点:

- 1) 无需依赖进口的 ARINC429 协议芯片和驱动芯片, 可同时实现 4 路发送、4 路接收的多通道通信, 总线数据支持高速和低速多种工作方式, 真正实现广泛应用的目的。
- 2) 不依赖 Xilinx 自身 IP 核 FIFO, 基于 Verilog HDL 语言自写 FIFO 缓存器, 充分利用时钟同步打拍方法, 提高了可靠性, 有效解决了数据间干扰和亚稳态问题[16], 完全可以保证传输过程中数据字之间为 5 位字间隔。
- 3) 基于 FPGA 的 ARINC429 协议的设计, 极大地方便了硬件工程师设计 PCB 电路, 节省了板卡面积, 方便通信通道的扩展, 有效地提升了数据通信模块的处理能力和集成度, 降低了研发成本, 提高了研发效率。

基金项目

西南民族大学中央高校基本科研业务费专项资金项目+2021NYYXS82。

参考文献

- [1] 张新, 赵峰, 黄栋杉. 基于 FPGA 的多通道 ARINC429 通信芯片的设计[J]. 航空电子技术, 2008, 39(1): 46-52.
- [2] 陈莹. 航空相机性能检测系统中综合测试系统的设计与实现[D]: [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2009.
- [3] 曾锋, 冯宇. 基于 FPGA 的 ARINC429 协议设计[J]. 现代电子技术, 2015, 38(18): 92-94.
- [4] 焦艳, 刘勇智. 基于 FPGA 多通道通用总线数据传输系统的设计[J]. 空军工程大学学报(自然科学版), 2006, 7(3): 41-43.
- [5] 李娥. 基于 MATLAB 的数据通信码型的编码实现[J]. 信息技术与信息化, 2021(11): 199-201.
- [6] 侯利民, 苏淑靖. 基于低压差分信号(LVDS)总线的数字信号源设计与实现[J]. 科学技术与工程, 2013, 13(32): 9714-9718.
- [7] 郑玉. USB-ARINC429 总线通信接口的设计[D]: [硕士学位论文]. 成都: 电子科技大学, 2009.
- [8] 孙戈东. TTL 电平信号与 ARINC429 信号的转换[J]. 通讯世界, 2015(2): 61.
- [9] 蒋学东, 刘勇. 用 FPGA 实现 ARINC429 接口和总线数据接收[J]. 现代电子技术, 2010, 33(6): 116-118.
- [10] 宋尚升, 龙永佳. 基于 J750 的 ARINC429 总线协议芯片测试技术研究[J]. 电子制作, 2022, 30(4): 26-29.
- [11] 吕小凤, 张闯. 基于 HI3282 的 FPGA 与 ARINC429 总线通讯卡设计[J]. 无线电工程, 2016, 46(6): 69-71+75.
- [12] 刘鹏. RTLinux 下 429 总线和 1553B 总线设备驱动程序设计与实现[D]: [硕士学位论文]. 北京: 北京邮电大学, 2007.
- [13] 周婷婷. 航空通信设备 ARINC429 总线多协议处理设计与实现[J]. 中国新通信, 2017, 19(6): 20-22.
- [14] 孙毅刚, 梅傲雪, 崔海青. ARINC429 总线发送任务数据防冲突算法研究[J]. 微电子学与计算机, 2019, 36(4): 93-98.
- [15] 李柬, 谢建庭, 宋光伟, 等. 一种 TTL 电平转换为差分信号电平的输出电路[P]. 中国专利, 202268867U. 2012-06-06.
- [16] 王琦, 张黎烁. 基于 FPGA 的 ARINC429 总线 IP 核的设计与实现[J]. 河南师范大学学报(自然科学版), 2014, 42(4): 149-153.