

DOI:10.3969/j.issn.1003-5060.2025.01.008

# 基于FPGA的TTCAN总线控制器的设计与验证

曹燕, 许高斌, 周东升, 马渊明, 冯建国

(合肥工业大学 微电子学院, 安徽 合肥 230601)

**摘要:**为了提高控制器局域网(controller area network, CAN)通信的实时性,文章基于时间触发CAN(time-triggered CAN, TTCAN)协议,采用verilog硬件描述语言设计实现TTCAN总线控制器。该控制器可在经典CAN模式与TTCAN模式之间进行切换,同时兼容可变速率CAN(CAN with flexible data rate, CAN FD)协议下的高速通信。实验结果证明,相较于经典CAN总线控制器,该TTCAN总线控制器能够减少在高总线负载率下周期性报文的发送延时,在全波特率范围内且总线负载率大于等于60%的情况下,能够减少75%以上的CAN报文发送延时。

**关键词:**控制器局域网(CAN);时间触发;可变速率控制器局域网(CAN FD);总线负载率;周期性报文

中图分类号:TN791

文献标志码:A

文章编号:1003-5060(2025)01-0050-09

## Design and verification of TTCAN bus controller based on FPGA

CAO Yan, XU Gaobin, ZHOU Dongsheng, MA Yuanming, FENG Jianguo

(School of Microelectronics, Hefei University of Technology, Hefei 230601, China)

**Abstract:** To improve the real-time communication performance of the controller area network(CAN), this paper designed a time-triggered CAN(TTCAN) bus controller based on the TTCAN protocol using the verilog hardware description language. The controller can switch between classical CAN mode and TTCAN mode and is compatible with high-speed communication under the CAN with flexible data rate(CAN FD) protocol. Experimental results show that the TTCAN bus controller can reduce the transmission delay of periodic messages under high bus load rates compared with the classical CAN bus controller. Within the full baud rate range, the TTCAN bus controller can reduce the delay of CAN message transmission by more than 75% when the bus load rate is greater than or equal to 60%.

**Key words:** controller area network(CAN); time trigger; controller area network with flexible data rate(CAN FD); bus load rate; periodic message

## 0 引言

控制器局域网(controller area network, CAN)总线的非破坏性仲裁机制能保证报文发送的优先级要求,但是在众多真实工业场景如多传感器信息处理系统<sup>[1]</sup>中,基于事件触发机制的CAN节点之间需要进行快速高效的实时数据交

换。随着总线负载率的增大,当报文的碰撞概率达到一定程度时,由于非破坏性仲裁机制的存在,系统中一部分报文的收发会产生延时,甚至完全被阻塞。为了解决上述问题,博世公司在标准CAN的基础上开发了时间触发CAN(time-triggered CAN, TTCAN)协议<sup>[2]</sup>, TTCAN总线具有实时性高、可预测性好、抗干扰性强的特点<sup>[3]</sup>,适

收稿日期:2023-04-24;修回日期:2023-05-05

基金项目:国家重点研发计划资助项目(2022YFB3205903);安徽省发改委研发创新资助项目(JZ2021AFKJ0050)

作者简介:曹燕(1999—),女,安徽宣城人,合肥工业大学硕士生;

许高斌(1970—),男,安徽合肥人,博士,合肥工业大学教授,博士生导师,通信作者, E-mail: gbxu@hfut.edu.cn.

用于各种安全性要求很高的场景,如汽车电子系统、核电站报警系统和建筑电气监控系统等<sup>[4]</sup>。文献[5]将 TTCAN 总线设计用作时间触发型网络的低成本子节点,实现全网络的时钟同步与时间触发。

另一方面,随着总线节点数目不断增多,具有更高位速率和更大数据容量的可变速率 CAN (CAN with flexible data rate, CAN FD)总线将完全取代经典 CAN 总线,成为未来主流的分布式网络总线,但由于更新成本的问题,未来很长一段时间内将是 CAN 和 CAN FD 共存的网络状态<sup>[6]</sup>。因此,CAN 和 CAN FD 的兼容性是当前主要研究方向。文献[7]通过中心网关的方式实现 CAN 和 CAN FD 总线的兼容通信,但是这种方法需要额外的网关部署成本。

本文在 CAN 总线控制器的基础上,提出一种兼具高可靠性和高实时性的 TTCAN 总线控制器。该控制器实现了经典 CAN 模式与 TTCAN 模式的切换,基于时间触发的 TTCAN 模式降低了高总线负载率下周期性报文的发送延时;同时针对基于事件触发的 CAN 模式,本文对其报文的发送缓存空间进行优化,通过对缓存消息槽地址进行冒泡排序,降低高优先级报文的发送延时;最后通过兼容 CAN FD 总线协议提升总线传输速率和吞吐量,仿真得到的最高净传输速率可达 5 Mb/s。延时实验结果证明,相较于经典 CAN 模式,TTCAN 模式与 CAN FD 的结合能够实现更加稳定的低延时水平。

### 1 TTCAN 总线协议

TTCAN 总线协议是一种 CAN 数据链路层之上的高层扩展协议,可基于经典 CAN 物理层协议实现<sup>[8]</sup>。它采用基于时间触发的静态表调度方法,研究重点是时间同步和系统矩阵的相关问题<sup>[9]</sup>。TTCAN 总线协议的基本工作机制是:构建时间调度表,规定各时间窗口内允许发送的消息类型;由网络中特定的计时主机发送参考消息,所有节点都根据参考消息完成时间同步,节点根据系统矩阵在规定的窗口发送消息<sup>[10]</sup>。本文 TTCAN 总线控制器在经典 CAN 总线控制器逻辑的基础上增加部分设计,从而实现 CAN 模式与 TTCAN 模式的切换。

#### 1.1 经典 CAN 总线通信模型

CAN 总线通信模型如图 1 所示。CAN 总线上存在多个通信节点,每个通信节点包括总线控

制器和 CAN 收发器,并通过上位机进行控制。总线控制器向上通过总线接口与上位机进行通信,向下通过接收信号 RX 和发送信号 TX 与 CAN 收发器进行通信,CAN 收发器将逻辑电平信号 RX 和 TX 转化成物理电平信号 CAN\_L 和 CAN\_H 后,数据才能够在 CAN 总线上传输<sup>[11]</sup>。

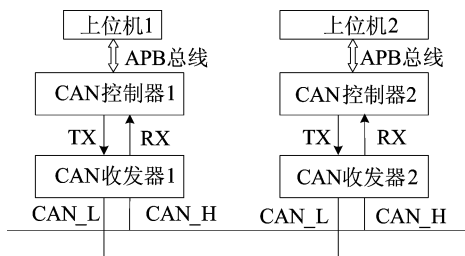


图 1 CAN 总线通信模型

#### 1.2 系统矩阵

TTCAN 是一种基于时间触发的以系统矩阵为核心的总线协议。系统矩阵如图 2 所示。



图 2 系统矩阵

1 个系统矩阵完整的传输周期称为系统矩阵周期(matrix period,MP),1 个 MP 包含多个基本周期(basic period,BP)<sup>[12]</sup>,单个 BP 由每行第 1 列的参考消息开始。每个 BP 包括 3 种时间窗口:专用时间窗口用来传输指定的周期性消息,如图 2 中的消息 A 和消息 B 等;仲裁时间窗口用于事件的争用;保留时间窗口暂时不用于传输任何消息,为以后协议的扩展预留。这些时间窗口可以为特定周期性或非周期性消息的传输提供精确定时限制。

### 2 控制器硬件设计

TTCAN 总线控制器整体架构如图 3 所示,主要包括报文接收、报文发送、位定时、验收滤波、错误管理、循环冗余校验码(cyclic redundancy check,CRC)<sup>[13]</sup>生成、时间同步和接口管理等子模块。

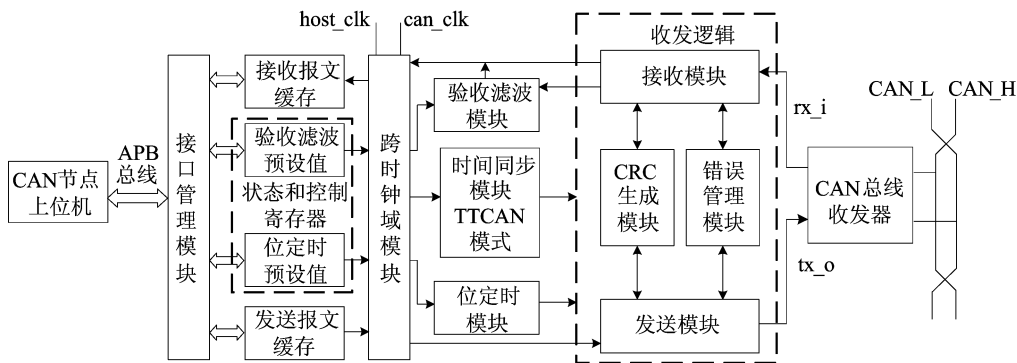


图 3 TTCAN 控制器整体架构

### 2.1 位定时模块

总线控制器的输入输出流的位宽为 1 位,因此每个数据位的定时准确性会直接影响数据采样的准确程度。位定时模块将每个数据位的时间分割为 3 个时间节,位定时状态机如图 4 所示。

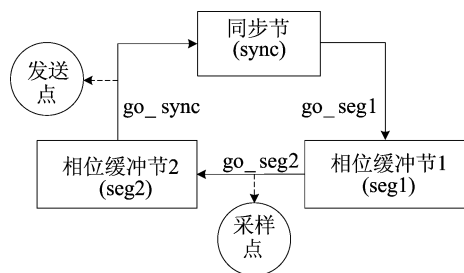


图 4 位定时状态机

从图 4 可以看出:第 1 节是同步节 sync,其作用是检测报文开始的下降沿信号 go-seg1;第 2 节是相位缓冲节 1,即 seg-1;第 3 节是相位缓冲节 2,即 seg2。相位缓冲节 1 的结束同时也是相位缓冲节 2 的开始,此时间点处于数据位的中

间区域,因此将此状态转换信号 go-seg2 设置为采样点;相位缓冲节 2 的结束同时也是整个数据位的结束,将此状态转换信号 go-sync 设置为发送点。位定时状态机的 3 个状态转换条件是由计数器是否达到对应时间节的预设值所决定的,整体减小计数器预设值能够缩减数据位时间,从而实现 CAN FD 报文更高的数据段位速率。

### 2.2 报文接收模块

报文接收模块根据报文帧格式分段、按位解析数据流,能够支持 CAN 报文中的标准帧、扩展帧、遥控帧和 CAN FD 报文中的标准帧、扩展帧的通信,同时能够兼容标准 CAN FD 和非标准 CAN FD 协议。

接收报文状态机如图 5 所示。其中,虚线箭头路径是为兼容 CAN FD 通信而增设的状态。若报文的 ID 信息通过节点的验收滤波,则报文接收模块将有效报文信息存入相应的缓存,以供上位机取用;若报文的 ID 信息没有通过节点的验收滤波,则该报文被丢弃。

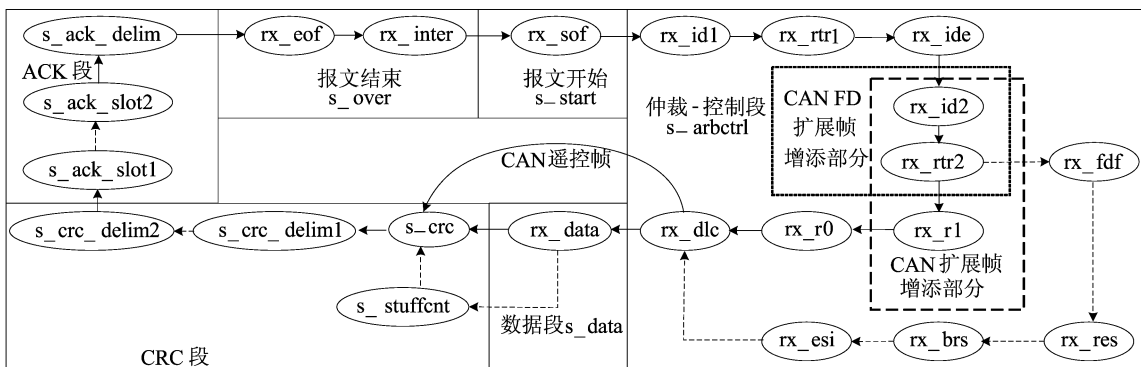


图 5 接收报文状态机

### 2.3 报文发送模块

在 TTCAN 模式下,不再依据 CAN 协议的非破坏仲裁机制来保证优先级高的报文优先获得

总线的控制权,而是根据系统矩阵的调度表来决定在某个触发时间发送某个确定的报文;但是仲裁时间窗口中仍然存在总线竞争,这有利于不需

要严格响应时间要求的软实时消息的传输。因此在保证硬实时消息及时传输的同时适当延长仲裁时间窗口的传输时间有利于整个系统性能的提高<sup>[14]</sup>。

报文发送模块按位、分段组合报文的过程与接收报文模块分段、按位解析报文的步骤相同,但路线方向相反。

### 2.4 错误管理模块

CAN 总线协议在出现接收或者发送错误时没有纠错机制,只有针对出错报文的重发机制,因此本文针对各种错误情况设计了错误检测和处理机制。

错误管理模块的工作流程如图 6 所示。

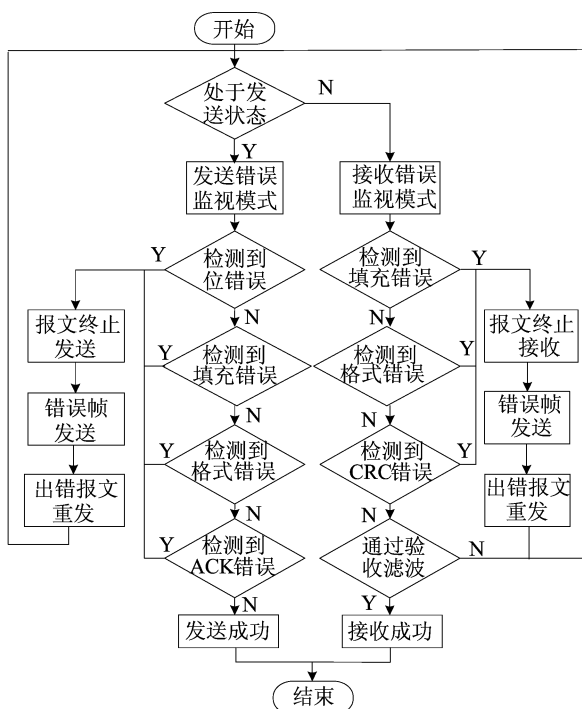


图 6 错误管理模块工作流程

CAN 总线在传输过程中会出现以下 5 种常见的错误,即格式错误、位错误、填充错误、CRC 错误和确认字符(acknowledge character, ACK)错误。只有通过所有类型错误检测的报文才能够完成收发过程,因此错误检测和处理贯穿整个总线控制器工作中。

### 2.5 时间同步模块

通过预设宏定义文件的参数,上位机可以选择是否实例化时间同步模块,并间接控制 TTCAN 模式的使能开关 TTEN。当 TTCAN 模式被禁止时,总线控制器以标准的事件触发模式进行通信;当 TTCAN 模式被打开时,总线控制器以时间触发模式进行通信。时间同步流程如

图 7 所示。

时间同步模块实现了用于记录本地时间的 16 位计数寄存器 counter\_level1,它在每个 CAN 位时间进行累加。每当在总线上有报文接收或发送时,模块将 counter\_level1 的值缓存在寄存器 Sync Mark 中,若经过 ID 信息匹配判定该报文是正确的参考消息,则将寄存器 Sync Mark 的值写入寄存器 Ref Mark 中,该寄存器值代表的是参考消息出现的本地时间。

随着报文收发的进行,寄存器 counter\_level1 与 Ref Mark 之间的差值便是 Cycle Time 的值,时间同步模块通过比较 Cycle Time 与系统矩阵中的窗口时标是否相等来判断是否进行后续时间窗口的报文接收与发送。

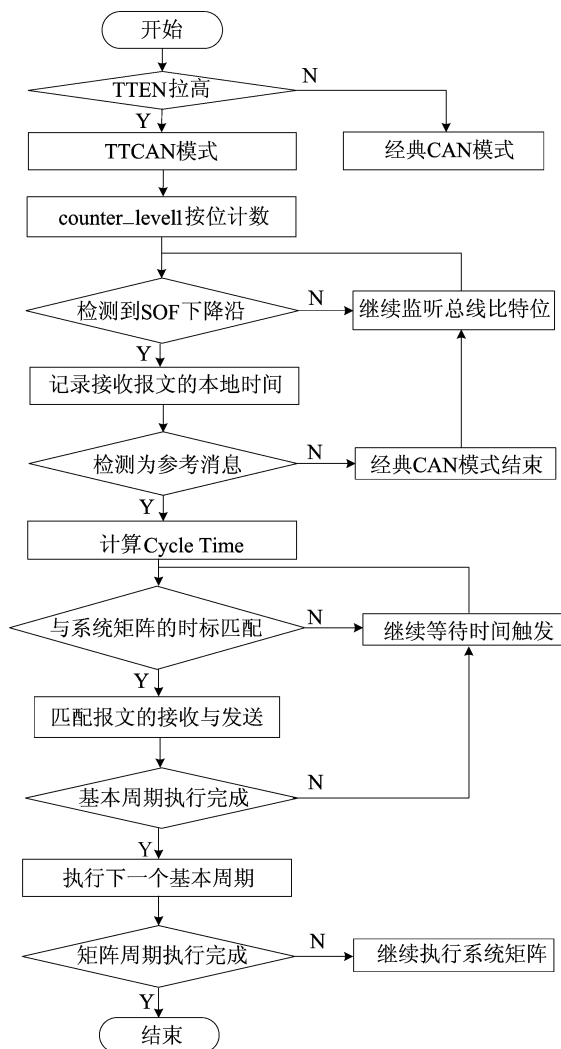


图 7 时间同步流程

### 2.6 发送报文缓存

为了实现节点报文发送顺序的自我优化,本文设计发送报文缓存的重排序功能。采用冒泡排

序算法的报文重排序过程如图 8 所示。

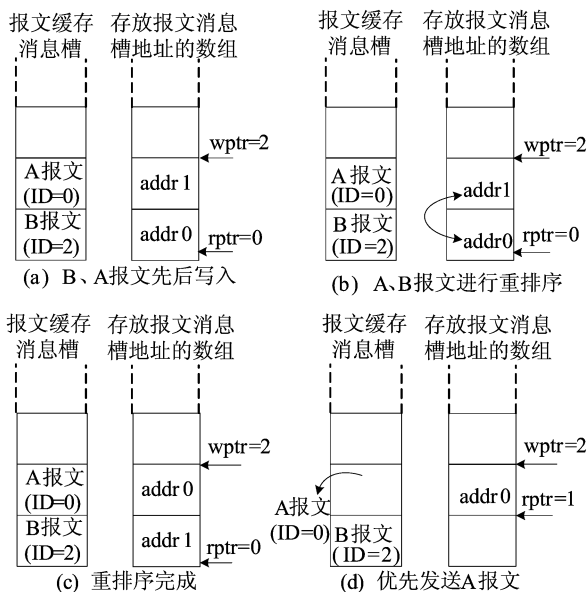


图 8 重排序功能示意图

首先定义一个环形的类先进先出 (first in first out, FIFO) 数组, 用于存放报文的缓存地址; 当发送报文写入缓存时, 数组写指针加 1; 当缓存中报文发送成功时, 数组读指针加 1; 当缓存中有 2 个及以上的报文时, 按照报文优先级从高到低对其地址进行冒泡排序。

重排序功能保证数组读指针始终指向优先级最高的报文的缓存地址, 从而防止高优先级报文被低优先级报文阻塞, 降低节点高优先级报文的发送延时。

### 3 功能验证和性能评估

#### 3.1 功能仿真验证

根据 CAN 及 TTCAN 总线的特性, 需要实例化 2 个及以上的 CAN 节点才能完成正常的收发, 因此本文设计了三节点的 CAN 验证平

台, 对 TTCAN 模式下参考消息的接收与发送、5 种时间触发类型的实现、报文接收与发送等情况分别进行测试。

本文设计的三节点 CAN 验证平台框架如图 9 所示。

三节点验证平台通过实例化 3 个 CAN 总线控制器模块来生成 3 个通信节点, 每个节点内置独立的时钟发生器用以贴合实际通信情况。主进程控制每个节点的从进程, 从进程扮演各个节点上位机的角色。CAN 总线模拟器模拟了实际的 CAN 总线工作状态, 例如模拟了总线上的传输延时; CAN 收发器模型模拟了实际 CAN 总线的收发过程; 总线错误模拟器通过向 CAN 总线注入错误数据位来模拟 CAN 总线上潜在的传输错误。

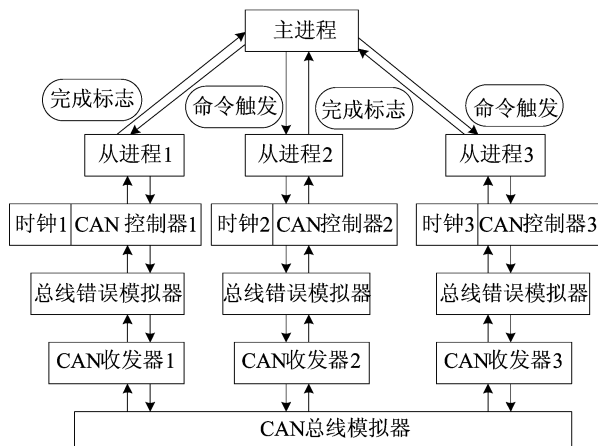


图 9 验证平台框架

TTCAN 模式下报文收发的仿真波形如图 10 所示。

图 10 中: can\_next\_bit 代表位时间; rxd[1 : 3]和 txd[1 : 3]分别代表节点 1、2、3 的接收信号和发送信号。

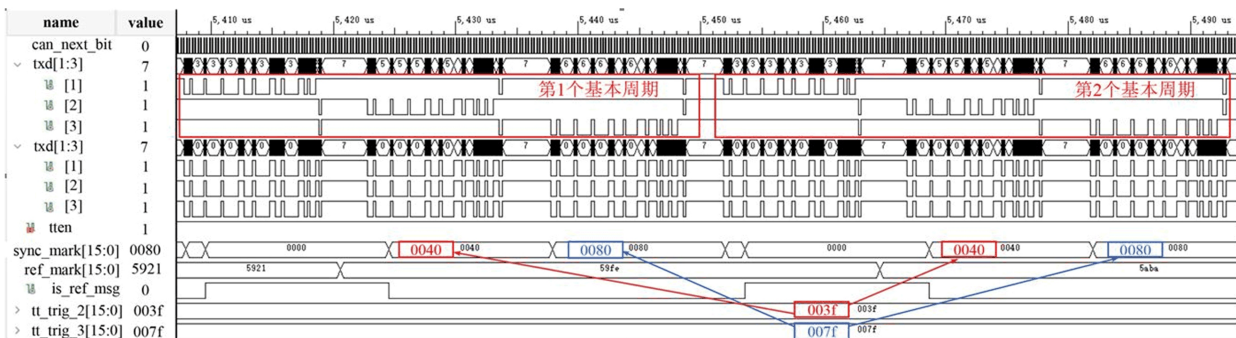


图 10 TTCAN 模式仿真波形图

节点 1 作为时间主机每隔 40  $\mu\text{s}$  发送 1 次参考消息,即系统矩阵的基本周期为 40  $\mu\text{s}$ ,参考消息判定成功后 is\_ref\_msg 信号拉高,直至参考消息接收完成。主进程设置节点 2、节点 3 在系统矩阵中的触发时标 tt\_trig\_2、tt\_trig\_3 分别为 0x003f、0x007f。图 10 中包含 2 个基本周期的波形,从波形中可以读出节点 2 和节点 3 始终在每个基本周期的 0x0040 和 0x0080 时刻成功发送报文,说明节点 2 和节点 3 成功地以时间触发的方

式按照系统矩阵调度表进行了周期性报文的收发。

### 3.2 性能评估实验

为了对比 TTCAN 模式和 CAN 模式的报文发送延时,本文在 CAN 总线上共设置 8 个 CAN 节点,采取改变每个节点报文发送周期的方式来间接改变 CAN 总线上的负载率。

波特率为 50 kb/s 和 500 kb/s 时各个优先级报文的周期配置分别见表 1 所列。

表 1 50 kb/s 和 500 kb/s 下各个优先级 CAN 报文的周期配置

负载率/%	8 节点(ID=[1,2,3,4,5,6,7,8])报文周期/ms	
	50 kb/s	500 kb/s
10	[100,200,200,200,200,200,200,200]	[10,20,20,20,20,20,20,20]
20	[50,50,100,100,100,100,200,200]	[5,5,10,10,10,10,20,20]
30	[20,50,50,50,100,200,200,200]	[2,5,5,5,10,20,20,20]
40	[20,20,50,50,50,100,200,200]	[2,2,5,5,5,10,20,20]
50	[20,20,20,50,50,50,100,200]	[2,2,2,5,5,5,10,20]
60	[10,20,20,50,50,100,100,100]	[1,2,2,5,5,10,10,10]
70	[10,10,20,50,50,100,100,200]	[1,1,2,5,5,10,10,20]
80	[10,10,20,20,50,50,100,100]	[1,1,2,2,5,5,10,10]
90	[10,10,10,20,50,50,100,200]	[1,1,1,2,5,5,10,20]
99	[10,10,10,20,20,50,50,100]	[1,1,1,2,2,5,5,20]

报文数据长度为 8 个字节,在不同负载率下测算不同优先级报文在 2.5 s 固定时间内的发送平均延时。本文选取的延时测量起点为节点发出报文发送请求的时刻,测量终点为报文开始发送的时刻。

CAN 模式和 TTCAN 模式下 CAN 报文发送延时性能评估实验的结果如图 11 所示。其中:左侧纵坐标轴变量为单位时间内报文数,它随着总线负载率的增大而增大;右侧纵坐标轴变量为报文发送平均延时; $D_1$  代表 TTCAN 模式下的发送平均延时,所选取的测算对象是 8 个节点中最低优先级的报文; $D_2$  代表 CAN 模式下的发送平均延时。

波特率为 50 kb/s 时不同负载率下的发送延时性能如图 11a 所示。从图 11a 可以看出:在总线负载率低于 60% 的情况下, $D_1$  与  $D_2$  近似相等;在总线负载率高于 60% 的情况下, $D_1$  基本保持较低的延时水平,且几乎不随着总线负载率的增加而变化, $D_2$  随着总线负载率的增大而显著增大,甚至在 99% 负载率的情况下因发送延时过大而导致最低优先级报文始终无法发出。可知,相较于经典 CAN 总线控制器,该 TTCAN 总线控制器在波特率为 50 kb/s、60%~99% 负载率下可以减少 90.58%~97.96% 的延时。

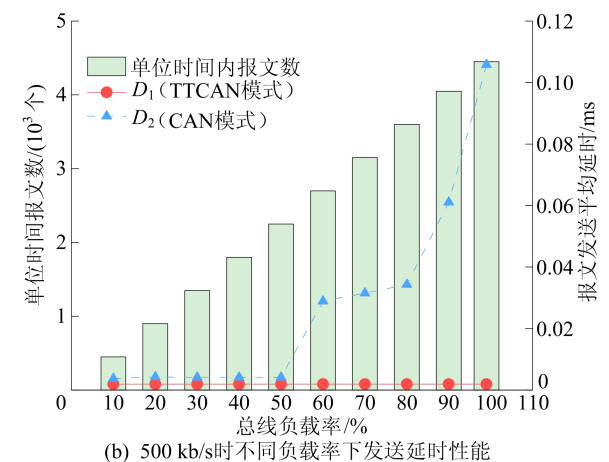
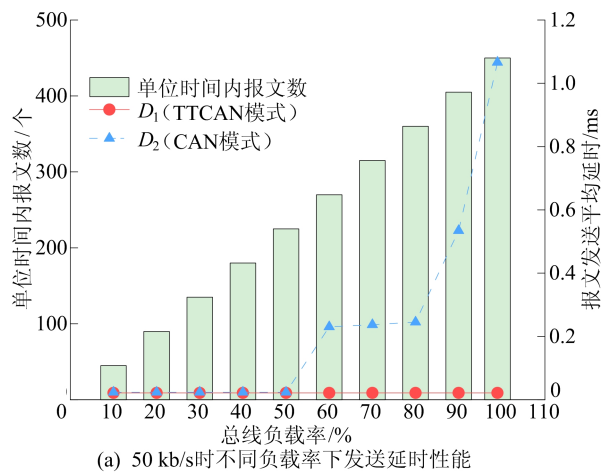


图 11 不同负载率下 CAN 报文发送延时实验结果

波特率为 500 kb/s 时不同负载率下的发送延时性能如图 11b 所示。从图 11b 可知,相较于经典 CAN 总线控制器,该 TTCAN 总线控制器在波特率为 500 kb/s、60%~99% 负载率下可以减少 93.27%~98.16% 的延时,同时高波特率下的发送延时整体比低波特率下的发送延时更低。

综上所述,TTCAN 模式下的报文发送延时比 CAN 模式更加稳定可靠,不会随总线负载率的变化而产生剧烈波动,且在高负载率下 TTCAN 模式能够保持较低的延时水平。

为了对比 CAN FD 报文在 CAN 模式和 TTCAN 模式下的报文发送延时,本文在总线上设置 8 个 CAN FD 节点,改变总线负载率和测算报文发送平均延时的方法均与 CAN 节点实验相同,不同之处是将波特率设置为 2 组。第 1 组 CAN FD 报文仲裁段波特率为 50 kb/s,数据段波特率为 4 Mb/s;第 2 组 CAN FD 报文仲裁段波特率为 500 kb/s,数据段波特率为 4 Mb/s。2 组波特率下各个优先级报文的周期配置见表 2 所列,报文数据长度为 64 个字节。

表 2 2 组波特率下各个优先级 CAN FD 报文的周期配置

负载率/%	8 节点(ID={1,2,3,4,5,6,7,8})报文周期/ms	
	(50 kb/s,4 Mb/s)	(500 kb/s,4 Mb/s)
10	[10,50,100,200,200,200,200,200]	[10,10,20,20,20,20,20,20]
20	[10,10,50,50,100,100,100,100]	[2,10,10,10,10,20,20,20]
30	[10,10,20,20,20,20,100,100]	[1,10,10,10,10,20,20,20]
40	[10,10,10,20,20,20,20,20]	[1,2,5,10,10,20,20,20]
50	[10,10,10,10,10,10,20,20]	[1,2,2,5,5,10,20,20]
60	[5,5,5,10,20,20,50,50]	[1,1,2,5,5,10,20,20]
70	[5,5,5,10,10,10,20,20]	[1,1,2,2,5,5,10,10]
80	[5,5,5,5,10,10,10,50]	[1,1,1,2,5,5,10,10]
90	[5,5,5,5,5,10,10,20]	[1,1,1,1,2,20,20,20]
99	[5,5,5,5,5,5,10,10]	[1,1,1,1,2,2,10,20]

波特率组合为(50 kb/s,4 Mb/s)时不同负载率下发送延时性能如图 12a 所示。由图 12a 可知,相较于经典 CAN 总线控制器,该 TTCAN 总线控制器在波特率组合为(50 kb/s,4 Mb/s)、负载率为 60%~99% 时可减少 CAN FD 报文 57.40%~94.97% 的发送延时。

波特率组合为(500 kb/s,4 Mb/s)时不同负载率下发送延时性能如图 12b 所示。由图 12b 可知,相较于经典 CAN 总线控制器,该 TTCAN 总线控制器在波特率组合为(50 kb/s,4 Mb/s)、负载率为 60%~99% 时可以减少 CAN FD 报文 78.63%~97.96% 的延时,高波特率下的发送延时整体比低波特率下的发送延时低。

对比图 11 与图 12 可以看出,在相同负载率下总线波特率越高,报文发送延时越低。该结论从一方面证明了 CAN FD 总线能够因其更高的净传输速率获得更低的发送延时;另外 CAN FD 报文中加长的数据段能够减少报文数量,进而降低总线负载率。

综上所述,相较于经典 CAN 总线,与时间触发机制相结合的 CAN FD 总线在高负载率下的优势更加明显。

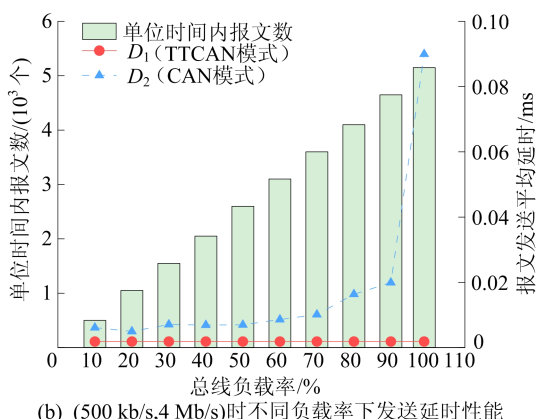
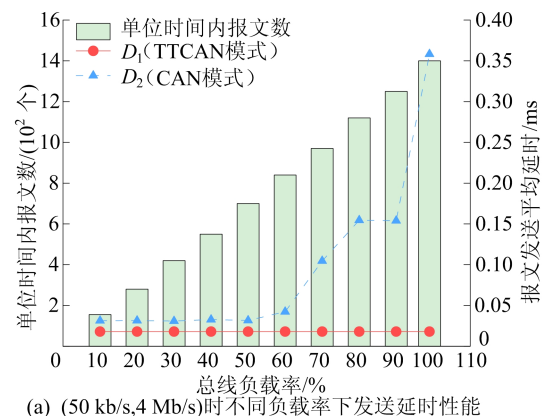


图 12 不同负载率下 CAN FD 报文发送延时实验结果

基于上述单个波特率的延时实验结果,本文在全波特率范围内进行了 60%~99%的高总线负载率下的延时实验。

设 TTCAN 模式比 CAN 模式的报文发送延时降低百分比为  $r$ ,计算公式为:

$$r = \frac{D_{CAN} - D_{TTCAN}}{D_{CAN}} \times 100\% \quad (1)$$

其中: $D_{CAN}$ 为 CAN 模式下的报文发送平均延时; $D_{TTCAN}$ 为 TTCAN 模式下的报文发送平均延时。

60%~99%负载率下  $r$  与波特率的关系如图 13 所示。

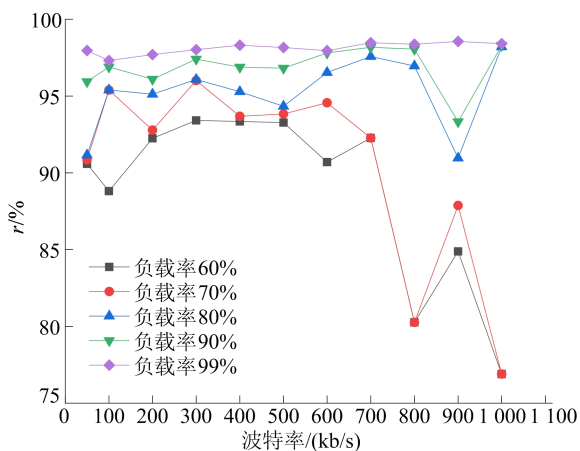


图 13 全波特率范围内 TTCAN 模式的发送延时降低百分比

图 13 实验结果表明,总线负载率越高,采用 TTCAN 模式降低报文发送延时的效果越好,且在总线负载率大于等于 60%的情况下能够减少 75%以上的 CAN 报文发送延时。

### 3.3 硬件验证

TTCAN 总线控制器仿真验证通过后,利用

Vivado 软件对工程进行综合优化和编译实现,所用现场可编程门阵列(field programmable gate array,FPGA)型号为 xc7a200tffg484-1,其综合实验报告显示消耗了 2 509 个查找表逻辑单元和 1 776 个寄存器。

工程资源使用的具体情况见表 3 所列。

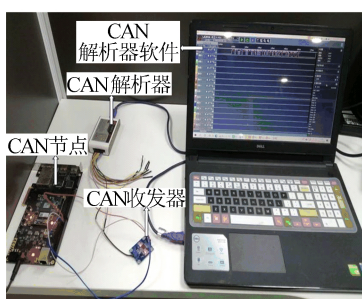
表 3 工程资源使用情况

资源类型	使用数目	可用数目	利用率/%
查找表	2 509	133 800	1.88
寄存器	1 776	267 600	0.66
块存储	1	365	0.27
I/O	75	285	26.32
全局时钟缓冲	4	32	12.50

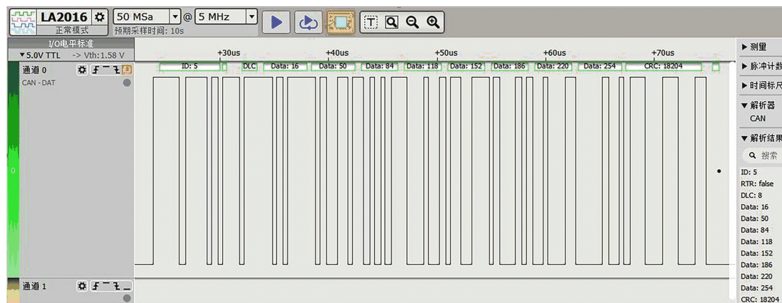
表 3 中为增加 TTCAN 功能而产生的查找表和寄存器开销分别只占总体工程的 2.66%和 6.06%,工程的工作频率最高可达 80 MHz。

实验的硬件环境如图 14a 所示。成功烧录工程的开发板可以作为 CAN 节点,通过 CAN 收发器连接到总线上,最后通过计算机上的 CAN 总线解析器软件成功解析报文。CAN 收发器芯片的型号为 TJA1050,本文使用的 CAN 总线解析器软件为 Kingst Logic Analyzer LA2016。

解析某标准帧报文的的结果如图 14b 所示。其中:左侧为 CAN 解析器在总线上捕获到的 1 个完整报文波形;右侧为 CAN 解析器解析出的报文信息,包括报文 ID、是否为遥控帧、数据长度及内容等。图 12b 报文解析结果直观地验证了 CAN 报文收发功能的正确性。



(a) 实验硬件环境



(b) CAN报文解析成功结果

图 14 真实硬件应用

## 4 结 论

本文设计了一种 TTCAN 总线控制器,支持

经典 CAN 模式与 TTCAN 模式的切换使用,同时兼容 CAN FD 总线下的高速通信,且其应用于 CAN 总线和 CAN FD 总线时最高通信速度分别

为 1 Mb/s 和 5 Mb/s。实验结果证明,相较于经典 CAN 总线控制器,本文 TTCAN 总线控制器能够减少在高总线负载率下周期性报文的发送延时,总线负载率大于等于 60%的情况下能够减少 75%以上的 CAN 报文发送延时。

本文的设计扩展了 TTCAN 协议下的 CAN FD 总线通信,在未来车载电子及多传感系统等硬实时应用中能发挥出更大的优势。

### [参 考 文 献]

- [1] 刘涛,葛雅静,谢宝荣,等.多传感器信息处理与显示系统的设计与实现[J].传感器与微系统,2020,39(1):71-74.
- [2] 周震,杨正林.时间触发 CAN-TTCAN[J].汽车电器,2005(5):3-7.
- [3] 俞植馨,李苏.基于冗余 TTCAN 的核电站报警窗改进设计[J].自动化仪表,2020,41(3):102-106.
- [4] 吴格,易茂祥,王春华,等.建筑电气监控系统中智能网关的设计[J].合肥工业大学学报(自然科学版),2016,39(8):1075-1079.
- [5] 曹云飞,王宁,夏天乾.基于 TTCAN 多级网络时钟同步与加载方案研究[J].中小企业管理与科技(下旬刊),2021,639(2):145-147.
- [6] 陈秋苑.CAN FD 与 CAN 网络共存的解决方案[J].中国仪器仪表,2020(11):94-98.
- [7] PARK T,LYU J R,SHIN K G.Optimal priority assignment for multiple CAN/CAN-FD buses with a central gateway [C]//2020 IEEE Real-Time Systems Symposium. [S. l.]:IEEE,2020:153-165.
- [8] GHODSOLLAHEE I,SEDAGHAT I,POURMOGHADAM M R.Improving the security and reliability of embedded networks with TTMAC-CAN[C]//2020 10th International Conference on Computer and Knowledge Engineering. [S. l.]:IEEE,2020:488-494.
- [9] KEATING D,MCINNES A,HAYES M.Model checking a TTCAN implementation [C]//IEEE Fourth International Conference on Software Testing,Verification & Validation. [S. l.]:IEEE,2011:387-396.
- [10] 李锦.基于 TTCAN 协议的监控系统设计与开发[D].武汉:华中科技大学,2017.
- [11] 李志刚.基于嵌入式的船舶机舱监测系统设计与开发[D].上海:上海海洋大学,2019.
- [12] QIAO X,WANG K F,SUN Y,et al.A Genetic Algorithms based optimization for TTCAN[C]//IEEE International Conference on Vehicular Electronics and Safety. [S. l.]:IEEE,2007:1-7.
- [13] 李世伟,马家庆.基于 FPGA 的 CRC 编解码算法研究[J].微处理机,2022,43(1):18-21.
- [14] 叶彦斐,刘之境,刘帅,等.分布式系统 TTCAN 调度优化算法研究[J].计算机测量与控制,2022,30(3):173-178.
- (责任编辑 胡亚敏)
- 
- (上接第 18 页)
- [7] SUNDERMEYER M,MARTON Z C,DURNER M,et al.Implicit 3D orientation learning for 6D object detection from RGB images [ED/OL].(2019-02-04)[2019-07-17].https://doi.org/10.48550/arXiv.1902.01275.
- [8] XIANG Y,SCHMIDT T,NAYANAN V,et al.PoseCNN:a convolutional neural network for 6D object pose estimation in cluttered scenes [ED/OL].(2017-11-01)[2018-05-26].http://export.arxiv.org/abs/1711.00199.
- [9] CHEN H,SUN K,TIAN Z,et al.Blendmask:top-down meets bottom-up for instance segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. [S. l.]:IEEE,2020:8753-8581.
- [10] MELLODO N,AIGER D,MITRA N J.Super4PCS:fast global pointcloud registration via smart indexing[J].Computer Graphics Forum,2015,33(5):205-215.
- [11] LI Y,QI H,DAI J,et al.Fully convolutional instance-aware semantic segmentation [C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. [S. l.]:IEEE,2017:2359-2367.
- [12] BOLYA D,ZHOU C,XIAO F,et al.Yolact:real-time instance segmentation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. [S. l.]:IEEE,2019:9157-9166.
- [13] HE K,GKIOXARI G,PIOTR D,et al.Mask R-CNN [C]//IEEE Transactions on Pattern Analysis & Machine Intelligence. [S. l.]:IEEE,2017:2961-2969.
- [14] 孙瑞,罗印升,宋伟.一种改进 SAC-IA 和 ICP 的快速高效点云配准算法 [J].工业控制计算机,2023,36(7):64-66,160.
- [15] 陈仲生,李潮林,左旺,等.双重下采样增强的点云改进配准算法研究 [J].汽车工程,2023,45(4):572-578.
- (责任编辑 胡亚敏)