

1.2 CAN同步机制-CSDN博客

成就一亿技术人!

原创

竹影行风 已于 2024-07-18 14:11:15 修改

于 2021-07-07 10:40:09 首次发布

版权声明：本文为博主原创文章，遵循CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

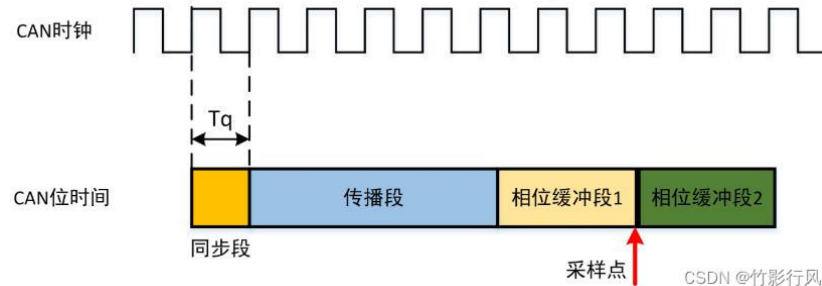
CAN是异步通讯，它跟uart一样，接收方和发送方都需要约定好相同的波特率才能正常通讯，波特率的倒数就是每一bit在传输时所占用的时间片长度。不管是uart还是CAN，一个bit的时间片一般都是由n个总线时钟组成，但是CAN比uart复杂的一点就在于它对时间片的n个总线时钟赋予了额外的含义，并以此为基础引入了一种特殊的同步机制，有力的提高了CAN的可靠性。

在解释CAN同步机制之前，需要先了解CAN大致的通讯过程：

CAN空闲状态为隐性电平（逻辑1），当某个节点要发起通讯时，总是先发送数据帧或遥控帧的1位显性电平（逻辑0）SOF，表示开始通讯，CAN退出空闲状态，后续该节点根据实际数据发送不同的电平，最后发送7位的隐性电平EOF，表示退出通讯。当CAN上出现连续11位隐性电平时，CAN再次进入空闲状态。

一、SS、PTS、PBS1、PBS2

如下图所示，CAN时钟的最小周期称为最小时间单元Time Quantum (Tq)，同时把一bit分为了4段，同步段（SS）、传播时间段（PTS）、相位缓冲段1（PBS1）和相位缓冲段2（PBS2）。每个段都占有数量不同的Tq，一个完整的bit，由8~25个Tq组成，除了SS段是固定的1Tq之外，其他的段占有Tq的数量都是可调的，而每一位的采样点在PBS1段和PBS2段之间，故可以通过控制PTS、PBS1、PBS2的长度，对采样点的位置进行偏移。



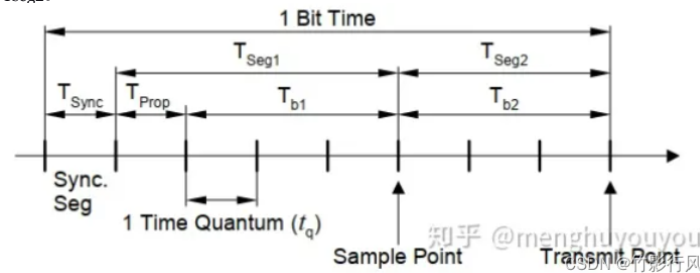
SS段，SYNC SEG，同步段，1Tq。隐性电平向显性电平的跳变沿必须发生在SS段，如果节点接收到的该跳变沿包含在SS范围内，则表示节点与总线的时序是同步的，如果不在SS范围内，则该节点会进行同步处理。

PTS段，PROP SEG，传播段，1~8Tq。用于补偿网络的物理延时时间，是总线上输入比较器和输出驱动器延时总和的两倍。

PBS1段，PHASE SEG1，相位缓冲段，1~8Tq。用来补偿总线跳变沿在位时序中的误差，它的长度可以在重新同步时加长，当重新同步结束，会恢复预设值。

PBS2段，PHASE SEG2，相位缓冲段，2~8Tq。也是用来补偿总线跳变沿在位时序中的误差，它的长度可以在重新同步时缩短，当重新同步结束，会恢复预设值。

在实际的应用中，为了方便，一般会把PTS和PBS1段合并起来使用，并且名字也会发生改变。如下图所示，同步段一般称为Tsync，PTS和PBS1合并，称为Tseg1，PBS2称为Tseg2。



在实际应用中，一般比较关注4个参数：Sample point、BTL cycles、TSEG1、TSEG2。

Sample point：以上图为例，一个位共8Tq，采样点在第5个Tq之后，故5/8=62.5%，故Sample point为62.5%。

BTL cycles：一个位所占用的总Tq，上图中为8Tq。

TSEG1：PTS和PBS1之和，上图中为4Tq。

TSEG2：PBS2，上图中为3Tq。

二、同步机制

首先要明确的是，执行时序同步动作的都是接收节点，最终目的都是要跟发送节点的时序同步起来，所以发送节点是不需要同步的。其次，发送节点在SS段如果因为位的改变，会产生电平跳变沿，只有在同步的情况下，接收节点接收到的跳变沿才能也处于接收节点时序的SS段。

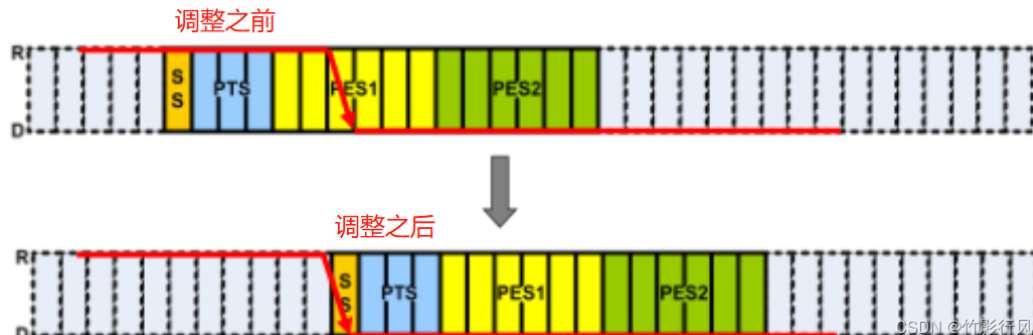
但是在传输过程中，肯定会有各种问题导致发送和接收节点的时序不同步，故引入了同步机制：硬同步、重新同步。

1、硬同步

当总线空闲时，所有的节点均处于接收状态，按照各自的时序来监听总线的信息，在同一时刻，很有可能节点A处于PTS段，节点B处于PBS1段，节点C处于PBS2段。

此时节点A要准备发起通讯，等它的时序到了SS段，它往总线上发送SOF跳变沿，此时，有可能节点B处于PTS，节点C处于PBS1。显然它们的时序没有跟节点A同步。故节点B和C要强行调整它们的SS段，令其包含接收到的SOF的跳变沿。

如下图所示，接收节点在PBS1阶段接收到了SOF的跳变沿，它只能强行调整SS段，变成调整之后的时序。



硬同步只发生在SOF位，它只能保证SOF的同步，无法保证后续位的同步。

2、重新同步

(1) SS段超前隐性电平到显性电平的跳变沿, PBS1暂时延长

第5位，发送和接收节点的时序又同步了，PBS1的延长只在重新同步的那一位生效，该位结束后，PBS1会恢复预设值。

