1.2 CAN同步机制-CSDN博客

成就一亿技术人!



竹影行风 三 已于 2024-07-18 14:11:15 修改

于 2021-07-07 10:40:09 首次发布

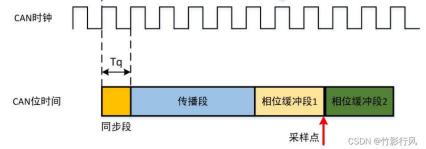
版权声明:本文为博主原创文章,遵循<u>CC 4.0 BY-SA</u>版权协议,转载请附上原文出处链接和本声明。

CAN是异步通讯,它跟<u>uart</u>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:

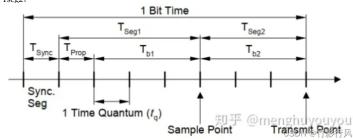
CAN空闲状态为险性电平(逻辑),而安允了那代和人共间通讯位注: CAN空闲状态为险性电平(逻辑),当集个节点要发起通讯时,总是先发送数据帧或遥控帧的1位显性电平(逻辑o)SOF,表示开始通讯,CAN退出空闲状态,后续该节点根据实际数据发送不同的电平,最后发送7位的隐性电平EOF,表示退出通讯。当CAN上出现连续11位隐性电平时,CAN再次进入空闲状态。

-, 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, 如果节点接受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~STQ。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,PROP SEG、传播段,1~STQ。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。
PBS2段,PHASE SEG1,相位缓冲段,1~STQ。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



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。

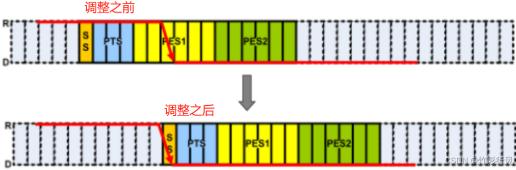
二、同步机制

首先要明确是的,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

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。

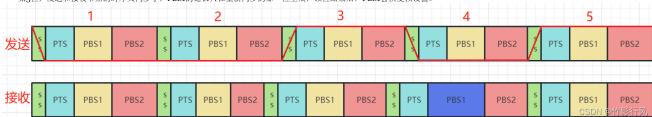
2、重新同步

当硬同步结束后,在后续的数据传输过程中,必须保证每一个由隐性电平到显性电平的跳变沿也必须落在SS段内。否则,产生重新同步。重新同步分为两种情况:

(1) SS段超前隐性电平到显性电平的跳变沿, PBS1暂时延长

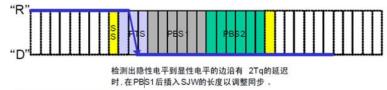
如下图所示,以接收节点的位数来分析: 第1位,硬同步结束后,发送和接收节点的时序已经同步,跳变沿能落在接收节点的SS段内。 第2位,接收节点因为某些干扰,导致该位的时长缩短了。 第3位,发送节点的跳变沿没有落在接收节点的SS段内,而是落在了PTS段,但是这是一个由显性电平到隐性电平的跳变沿,接收节点不会进行同步动作。 第4位,发送节点产生一个电平跳变,同样的接收节点发现该跳变还是落在了PTS段,但这次是由隐性电平到显性电平的跳变,接收节点会进行同步动作,它要延长PBS1,以便 同步时序

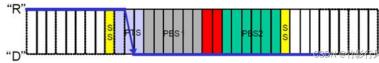
第5位,发送和接收节点的时序又同步了,PBS1的延长只在重新同步的那一位生效,该位结束后,PBS1会恢复预设值。



下图是对第4位时间同步的详细解析,接收节点接收到隐性电平到显性电平跳变沿时,已经处于PTS阶段、SS段超前跳变沿2个Tq,故为了后面的SS段能同步,在本次位<u>时间片</u>必须延长PBS1相同长度的Tq,即图中红色的2个Tq所示。

隐性电平到显性电平的边沿出现在PTS和PBS1之间时 (SJW = 2)



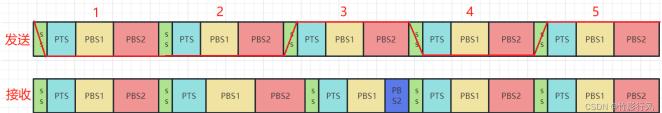


(2) SS段滞后隐性电平到显性电平的跳变沿, PBS2 暂时缩短

如下图所示,以接收节点的位数来分析: 第1位,硬同步结束后,发送和接收节点的时序已经同步,跳变沿能落在接收节点的SS段内。 第2位,接收节点因为某些干扰,导致该位的时长增大了。但此时发送节点已经发送了第3位的跳变沿,它落在了PBS2段内,但这是一个由显性电平到隐性电平的跳变沿,接收 节点不会进行同步动作。

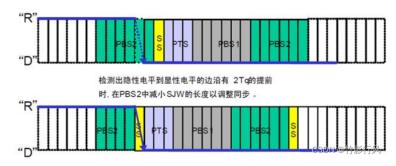
第3位,同样的,接收节点的PBS2接收到了发送节点发送的第4位的跳变沿,这是一个由隐性电平到显性电平的跳变沿,接收节点会进行同步动作,它要缩短PBS2,以便同步时

第4位,发送和接收节点的时序又同步了,PBS2的缩短只在重新同步的那一位生效,该位结束后,PBS2会恢复预设值。



下图是对第3位时间同步的详细解析,接收节点接收到隐性电平到显性电平跳变沿时,已经处于PBS2阶段,SS段落后跳变沿2个Tq,故为了后面的SS段能同步,在本次位时间片中,必须缩短PBS2相同长度的Tq,即图中的2个Tq。

隐性电平到显性电平的边沿出现在PBS2中时(SJW = 2)



(3) 总结

CAN同步只会出现在发送节点产生由隐性电平到显性电平跳变的时刻,该跳变沿必须在SS段内,如果不在则必须进行同步,唯一有区别的是,如果是SOF跳变,则进行硬同

步,其他的,则进行重新同步。
如果干扰太强,可能会导致一次重新同步没有完全同步回来,那也只能等待下一次的隐性电平到显性电平的跳变沿,再次进行重新同步,再次强调,PBS1的延长和PBS2的缩短 只在执行重新同步的位生效,一旦该位结束,不管时序有没有同步回来,PBS1和PBS2均会恢复预设值。

PBS1和PBS2只是提供缓冲而已,不能无限的缓冲。故有一个同步跳转宽度(SJW),它表示PBS1和PBS2重同步时允许跳转的最大宽度,即SJW是PBS1延长或PBS2缩短的上