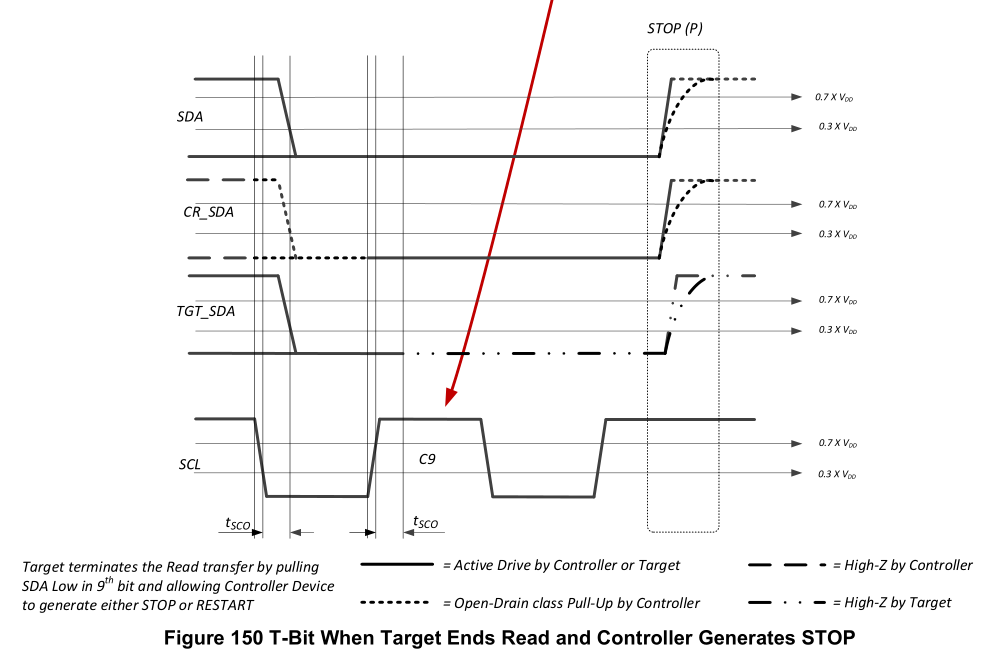
I3C与I2C的一个区别就在于slave具有终止master读数据传输的能力，具体的操作流程细节可参考5.1.2.3.4，T-bit的驱动方式永远为PP

1. slave主导T-bit值为“0”，终止读数据传输后，master产生STOP或START

操作流程为：

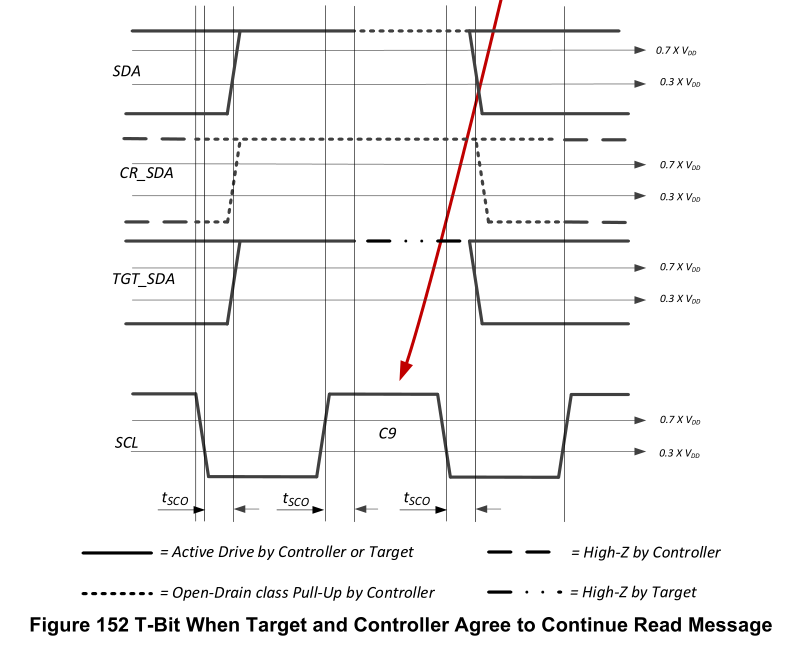
以C8 SCL的下降沿为参考，slave驱动SDA为低，以C9 SCL的上升沿为参考，slave释放SDA；与此同时，master也参考C9 SCL上升沿，驱动SDA为低，所以会有一个overlap，这个操作过程有点类似5.1.2.3.1 (从地址ACK到SDR主机写数据的传输)，当然需要满足图中的时序。随后的SDA总线控制权都在master，产生restart或stop。

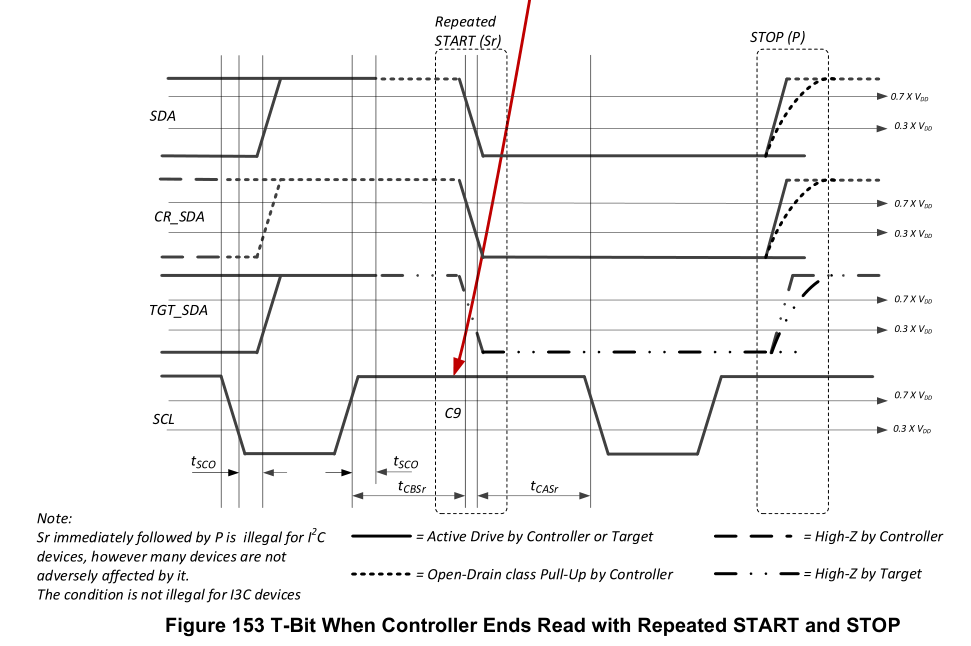


1. slave主导T-bit值为“1”以继续信息传输(当然也允许master终止传输)

以C8 SCL的下降沿为参考，slave驱动SDA为高，以C9 SCL的上升沿为参考，slave驱动SDA为高阻。那么在SCL 高电平期间slave失去控制权，分两种情况讨论：

* 1. master继续数据传输，只需要在SCL高电平期间维持SDA原本状态即可，Firgue 152
  2. master希望终止数据传输，以C9 SCL的上升沿为参考，master驱动SDA为低电平，那么这样就产生了Restart，master在C9的下降沿之后继续控制总线，发起新的传输或产生STOP。需要注意的是为满足时序要求，master可能需要延长SCL高电平的时间。（在混合总线中，则需要延长低电平时长，确保不会被i2c的滤波器过滤）





3、可以发现T-bit虽然算作双线通信的第9位（与I2C相比），但和i2c相比差异还是很大的。

3.1读数据的时候i2c slave不能终止通信，完全依靠master给出的ack/nack判断传输是否结束

3.2波形代表的含义完全不一样，图一波形表示的是slave终止读传输，而i2c协议表示的是master发送了ack，数据继续传输。图2表示的是数据继续传输，而在i2c协议来看，这样的波形表示master发送了nack，终止了数据传输。图3表示的是master终止读数据传输，T-bit相当于Start或Restart。

3.3若要考虑兼容i2c，在设计的时候还需要延长SCL低电平的时间以确保不会被i2c的滤波器给过滤

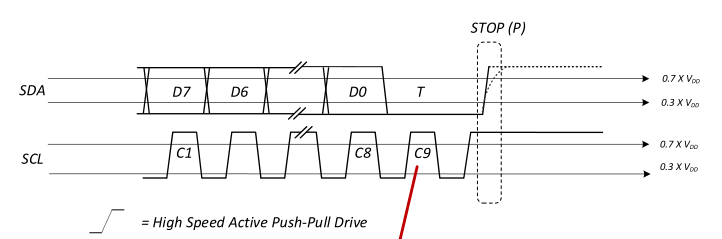


图1，slave终止读传输

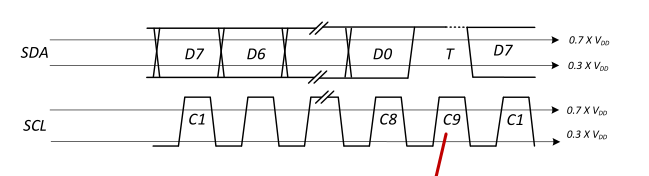
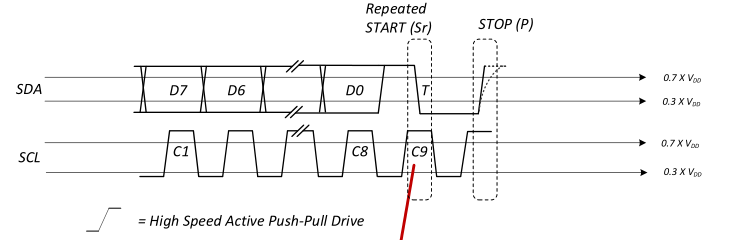


图2，slave和master继续传输



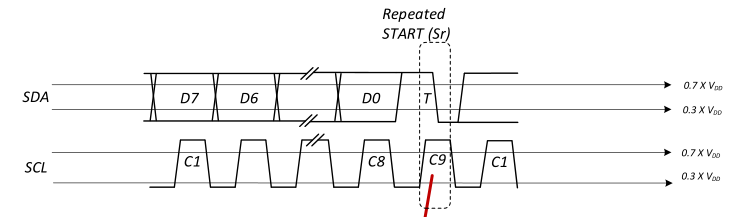


图3，master终止读传输,T-bit其实变成了START