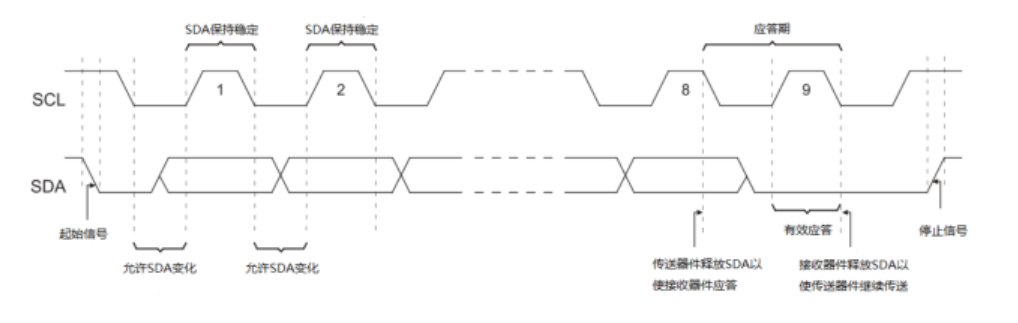
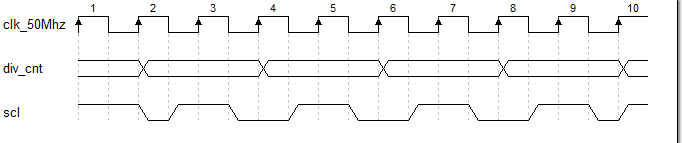
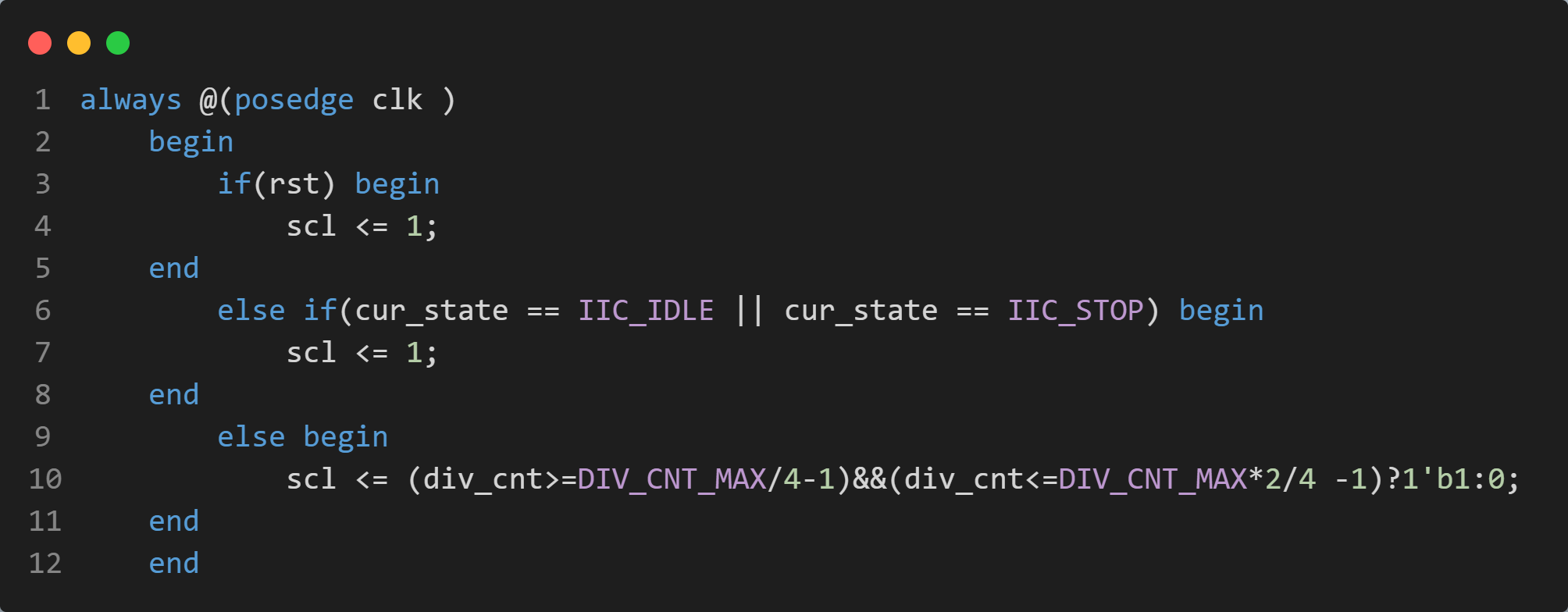
1. IIC时钟产生

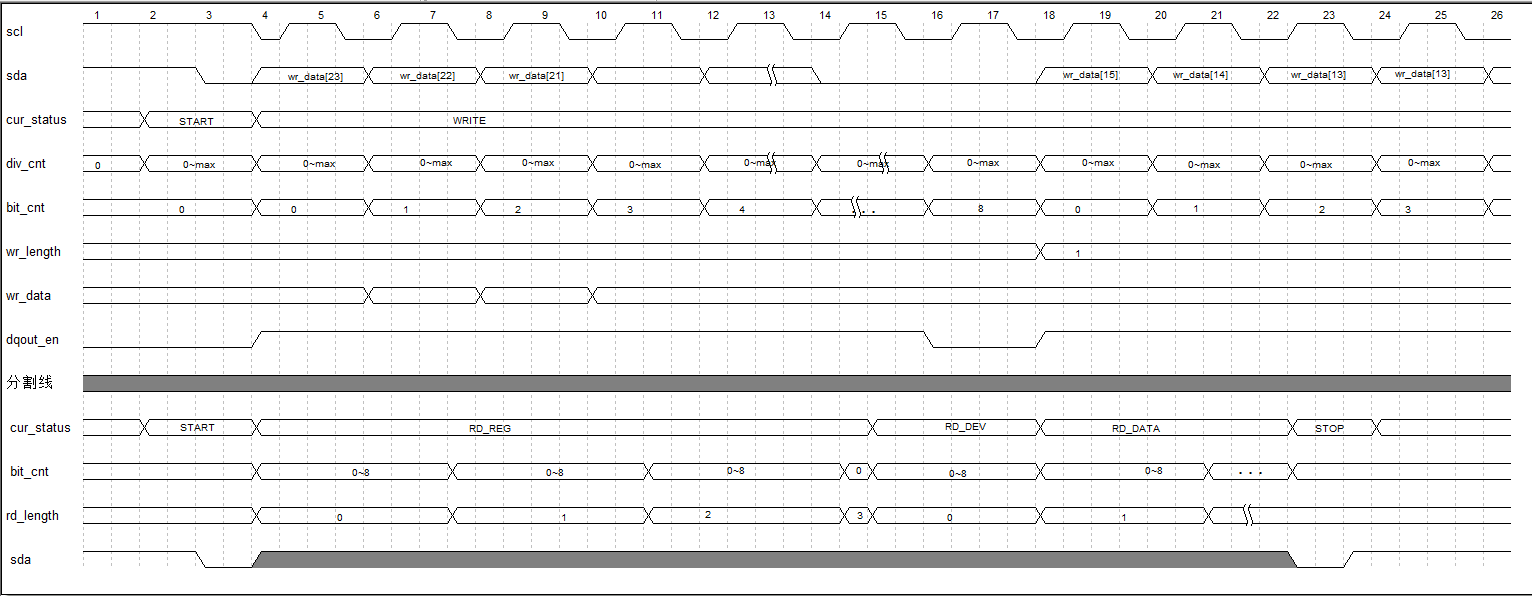
Iic的时钟由50mhz系统时钟分频而来，分频计数器的最大值max为（系统时钟频率/IIC时钟频率-1），在0~1/4max拉低scl，在1/4max~3/4max拉高scl，在3/4max~max拉低scl，这样设计的原因时IIC的数据需要在scl为高时保持稳定，在scl为低时改变



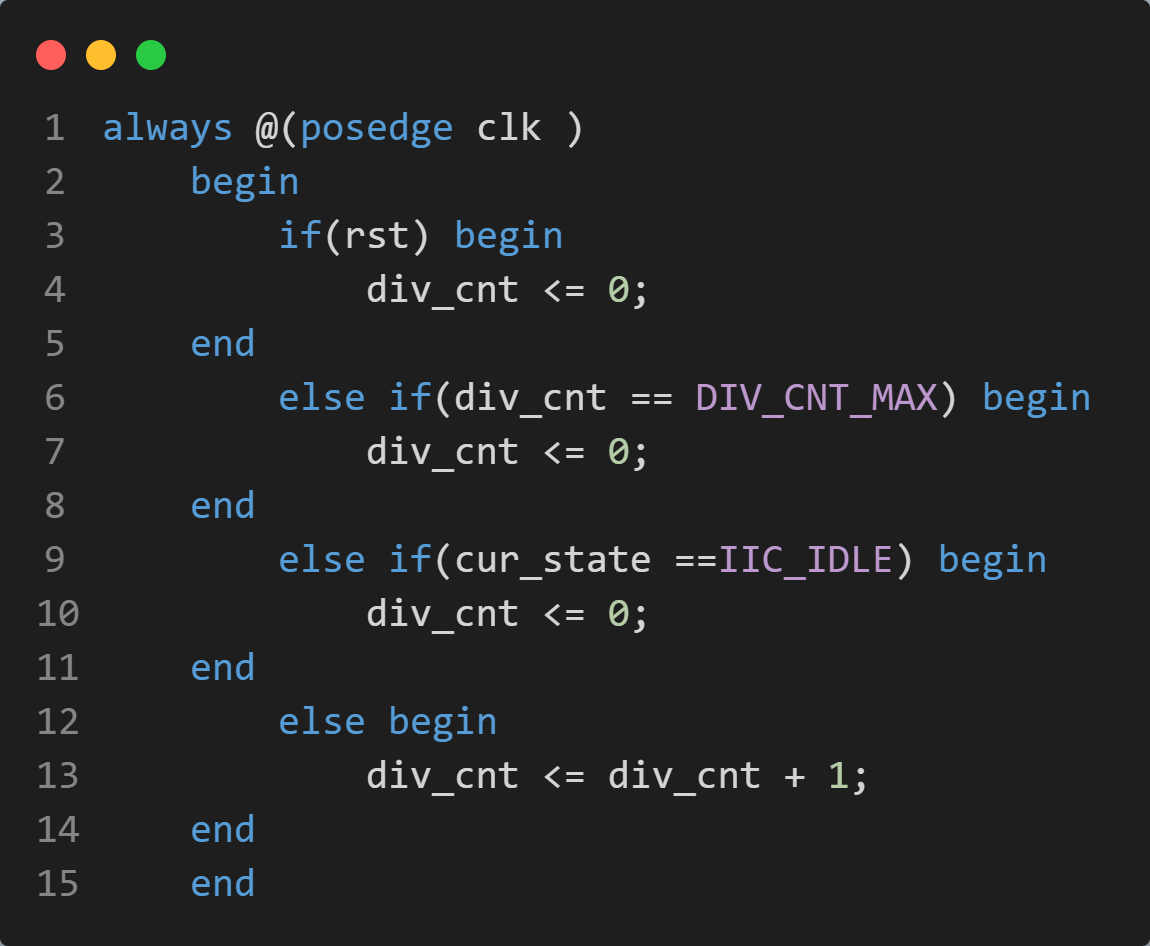
这样设计可以方便通过判断分频计数器的值来改变数据



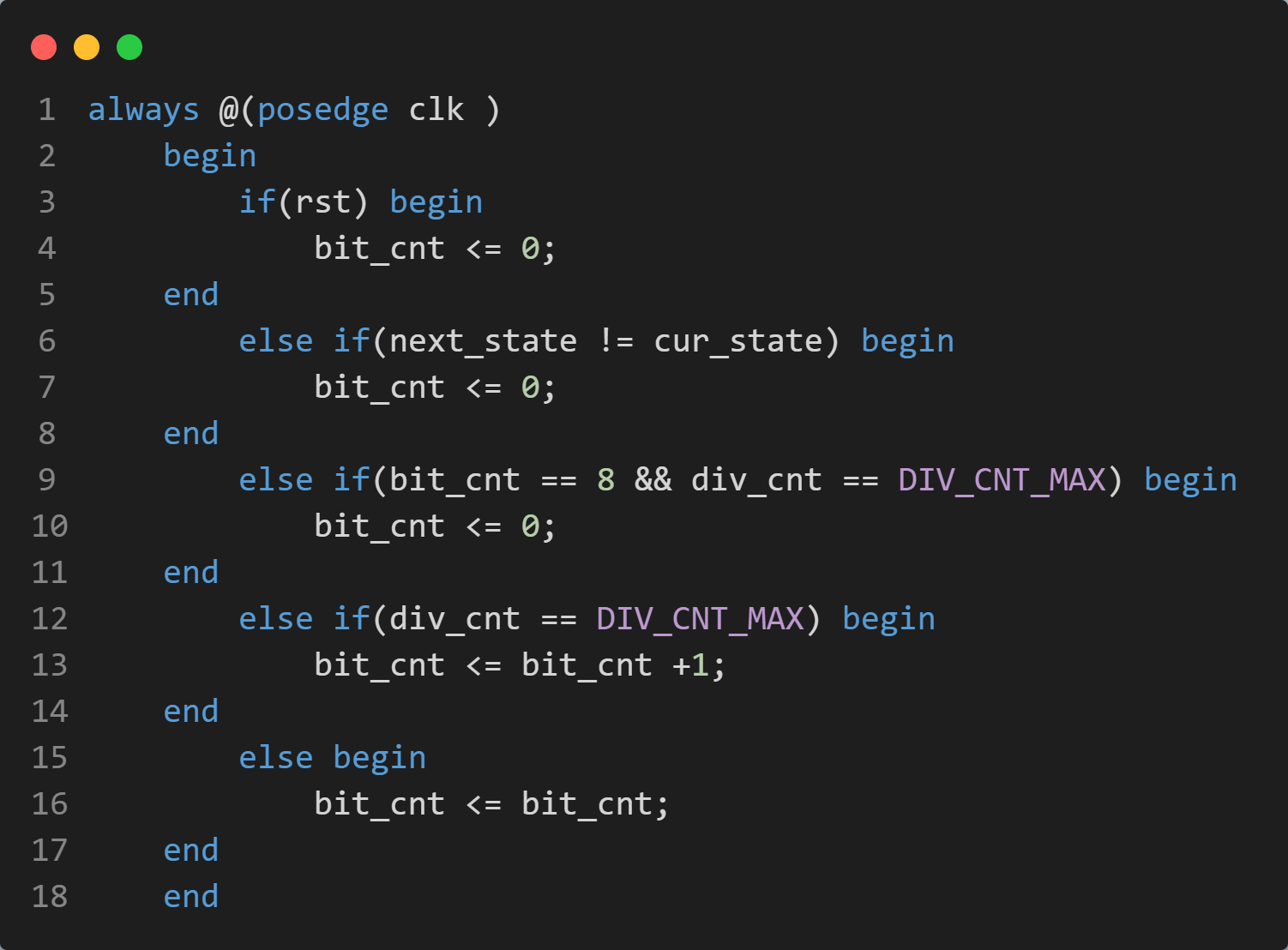




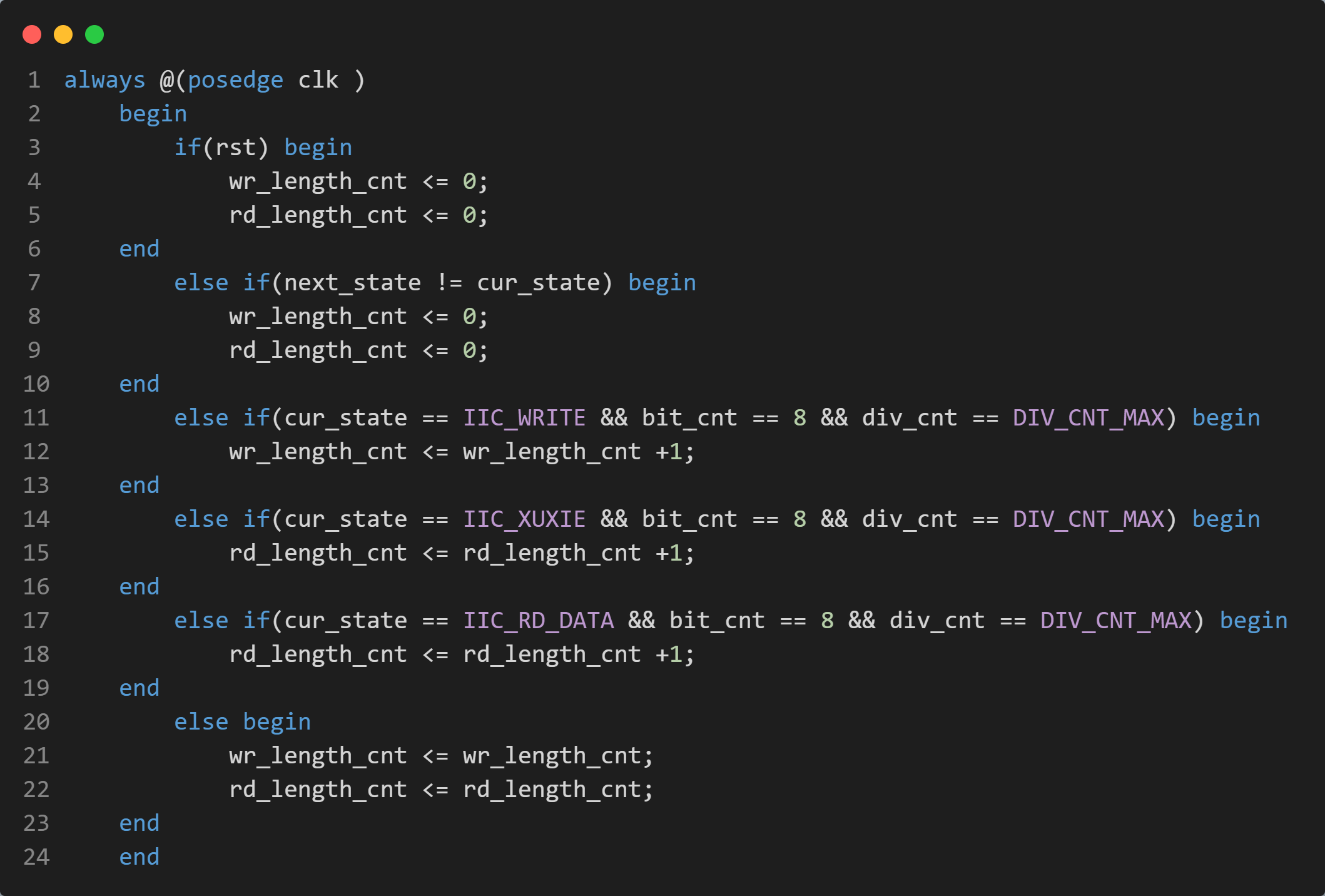
Div\_cnt:分频计数器，模块时钟频率/iic时钟频率可得最大值，div从0到max是一个iic时钟周期，idle状态不需要计数



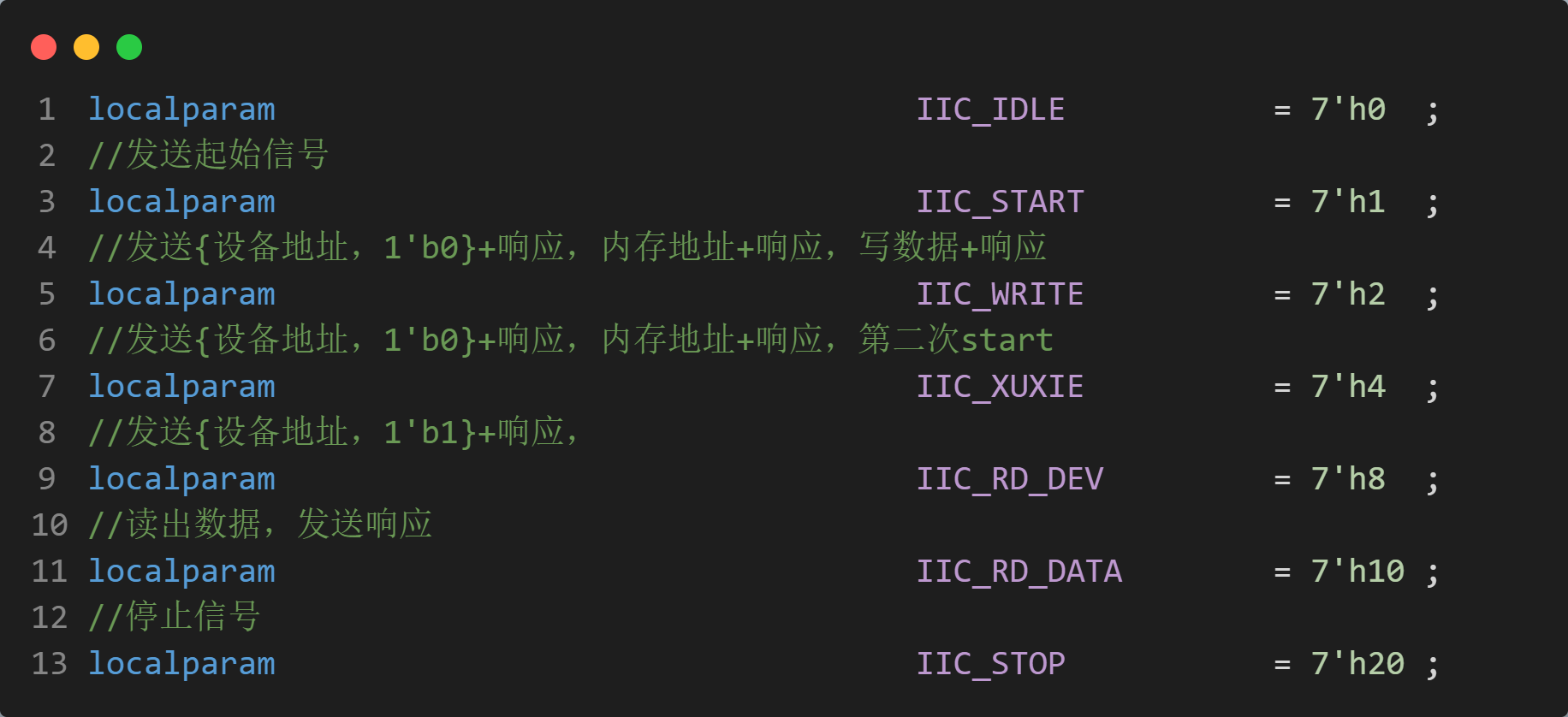
Bit\_cnt：iic协议一次发送一bit数据，bit计数器来指示发送多少bit数据，因为iic协议需要每发送8位数据还要附带一位响应，所以bit\_cnt设计成9位，同时切换状态归零



读写计数器：这两个计数器的目的是完成对写入数据和读出数据数量的计数，在读/写模式下，每发送/接收8位数据和1位响应，计数器值加1

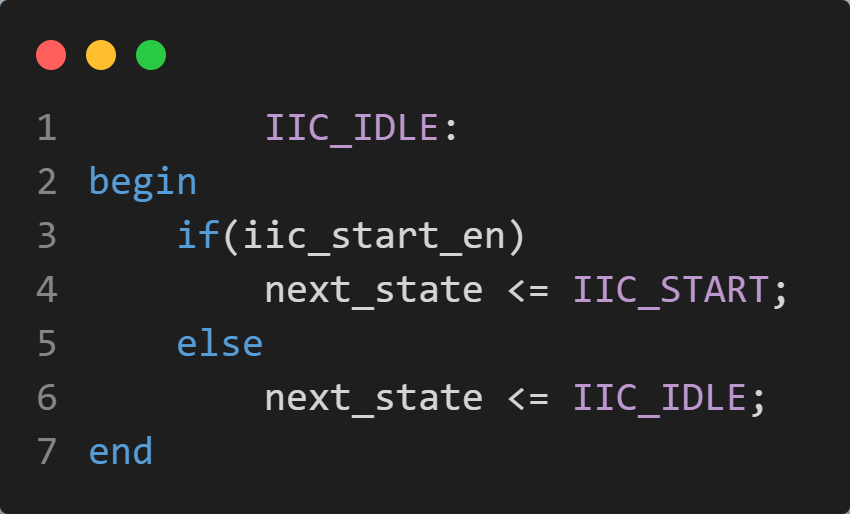


状态机设计：从iic的时序可以看出来，大体可以分成几个状态，首先是开始位和停止位，然后是发送8位数据和接收1位响应，最后是接收8位数据发送1位响应，在连续写时序中，不论是发送设备地址还是发送数据，都是发送8位数据，接收一位响应，整个写可以放到一个状态IIC\_WRITE中，在连续随机地址读之前，需要进行虚写，虚写是先发送设备地址和字地址，然后在发送一个起始位，不同于IIC\_WRITE,因此重新定义状态IIC\_XUXIE，然后发送设备地址，定义为IIC\_DEV,最后接收数据，定义为IIC\_DATA

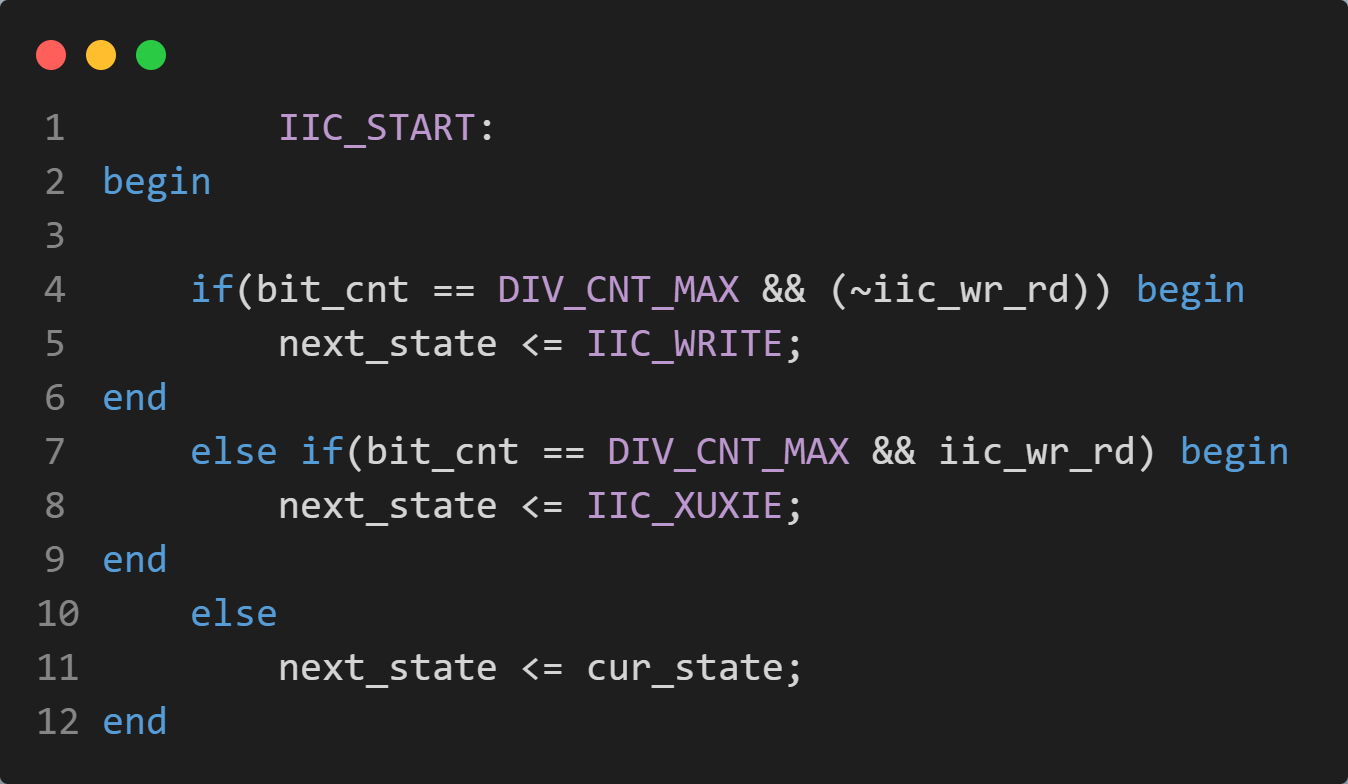


状态机跳转：

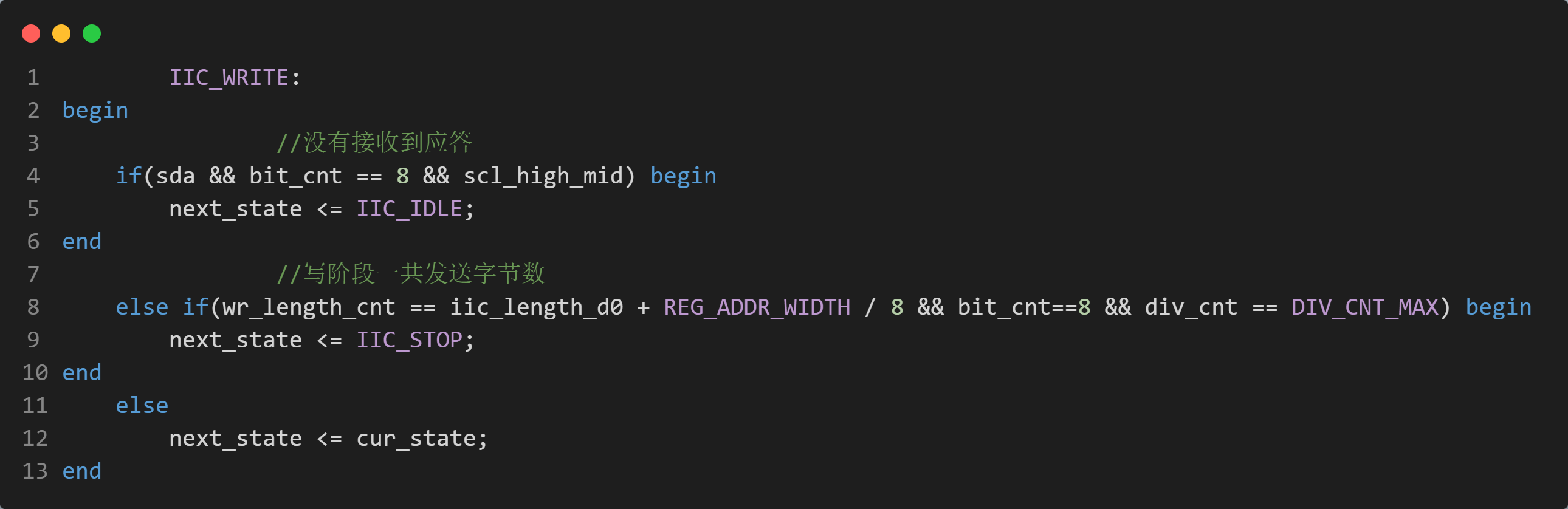
iic的模块设置一个输入的start信号，当start信号为高，则跳出idle状态，进入start



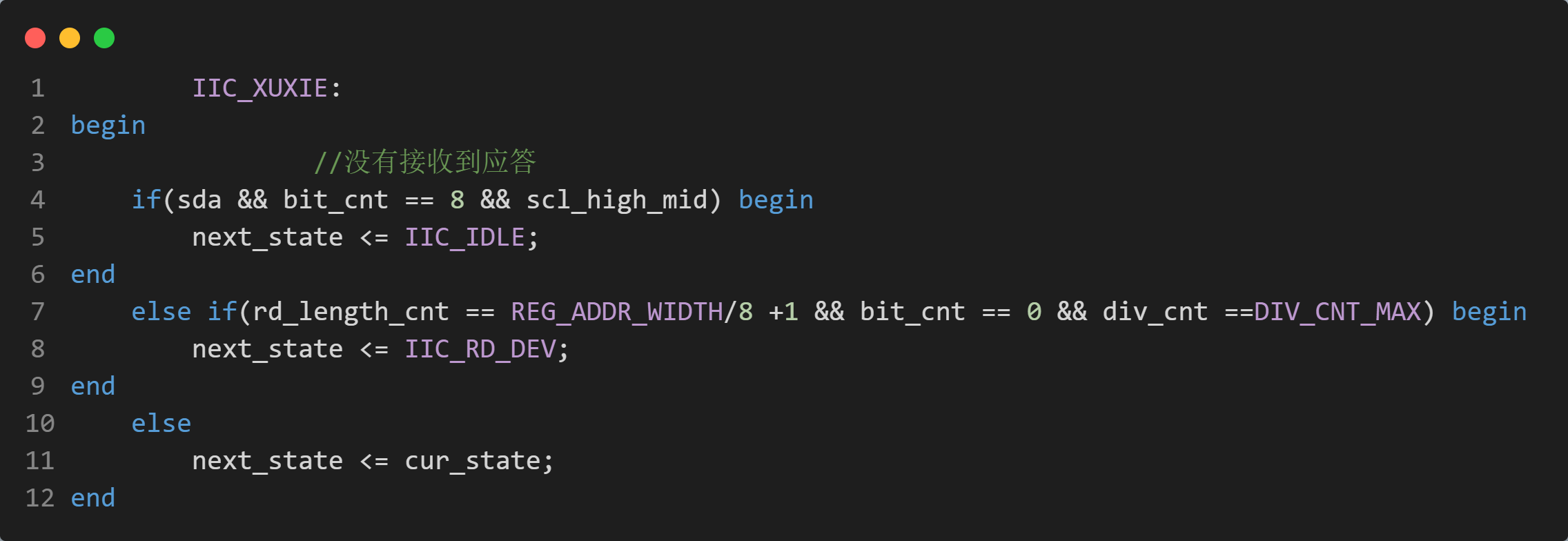
Start状态是发送一个起始位，即scl为高，sda下降沿。发送完成起始位，根据情况可以分为写数据和读数据两种，由输入信号iic\_wr\_rd的值来确定。



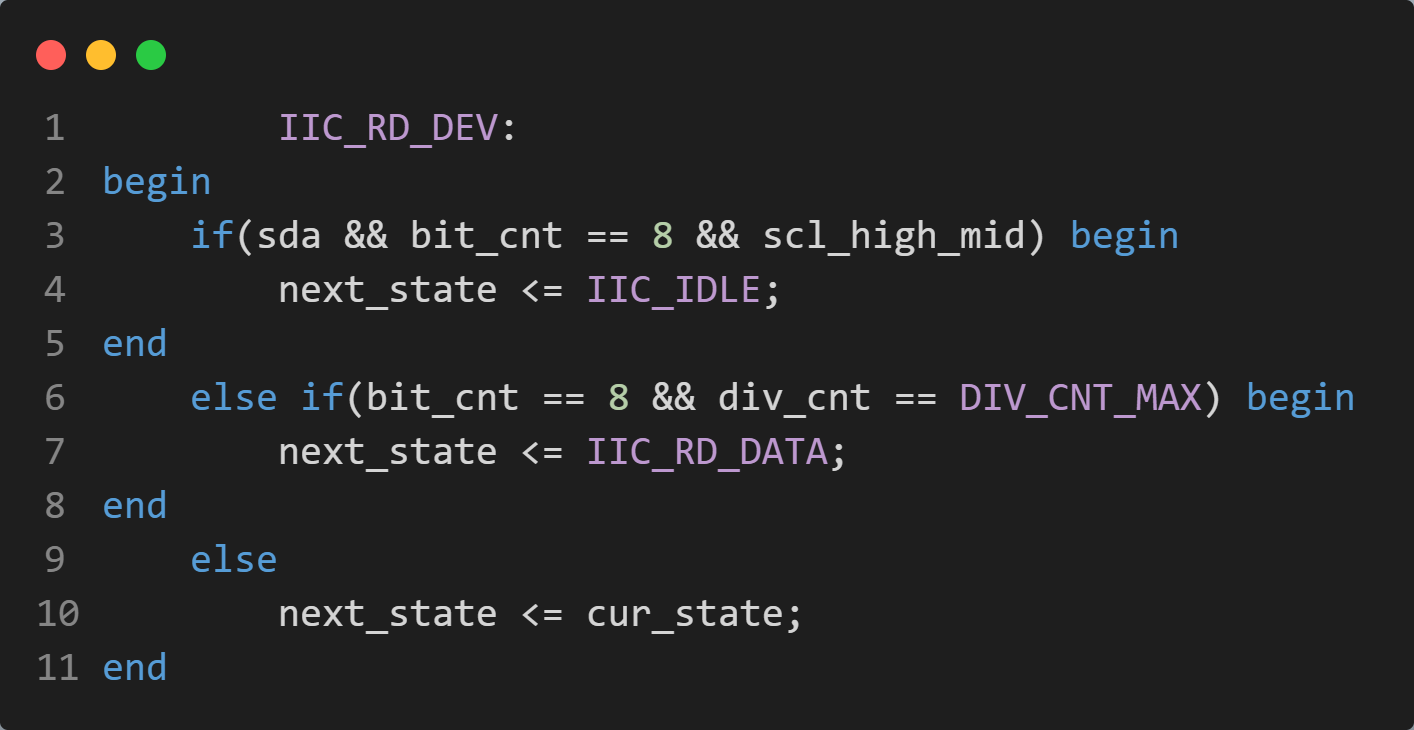
当iic\_wr\_rd为0时，进入IIC\_WRITE状态，这个状态需要发送8为设备地址＋16或者8位内存地址，因此我们定义wr\_data位宽为24，同时输入一个iic\_length来指示要写入的数据字节数，当wr\_length\_cnt计数==设定的字节数+1字节设备地址+内存地址位宽/8的时候，所有的数据发送完毕，跳转stop状态。



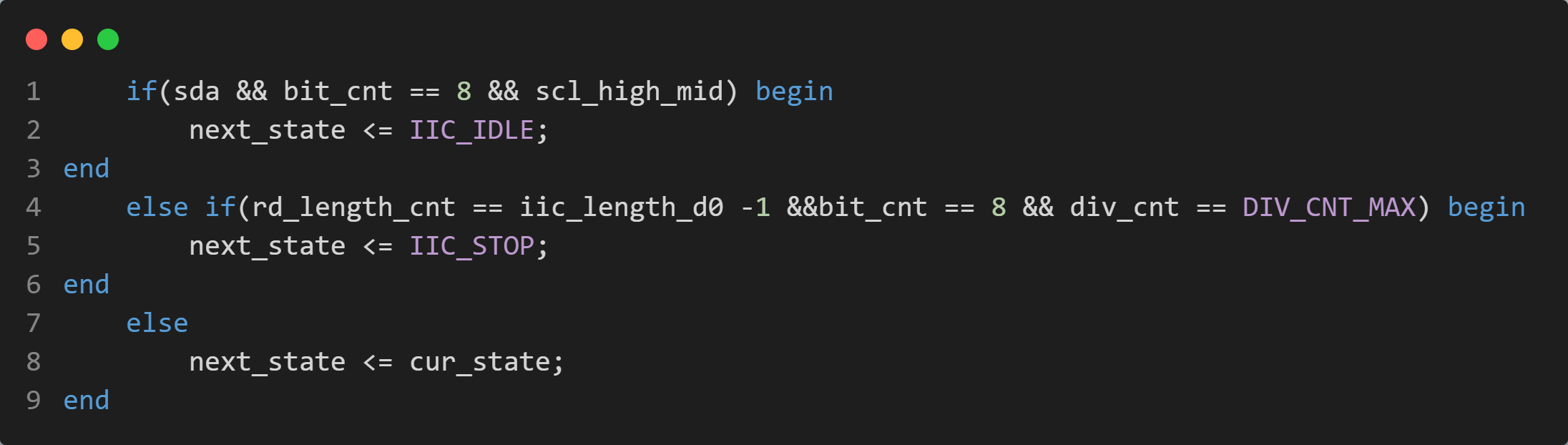
当iic\_wr\_rd为1时，进入IIC\_XUXIE状态，这个阶段需要发送设备地址和字地址，把设备中的地址指针指到我们要读数据的地方，同时还要在最后阶段在发送一次起始信号，发送完成标志为rd\_length\_cnt==1位设备地址＋内存地址位宽除8，并且bit==0，div\_Cnt==max,此时发送完成第二个起始位，进入发送设备地址阶段



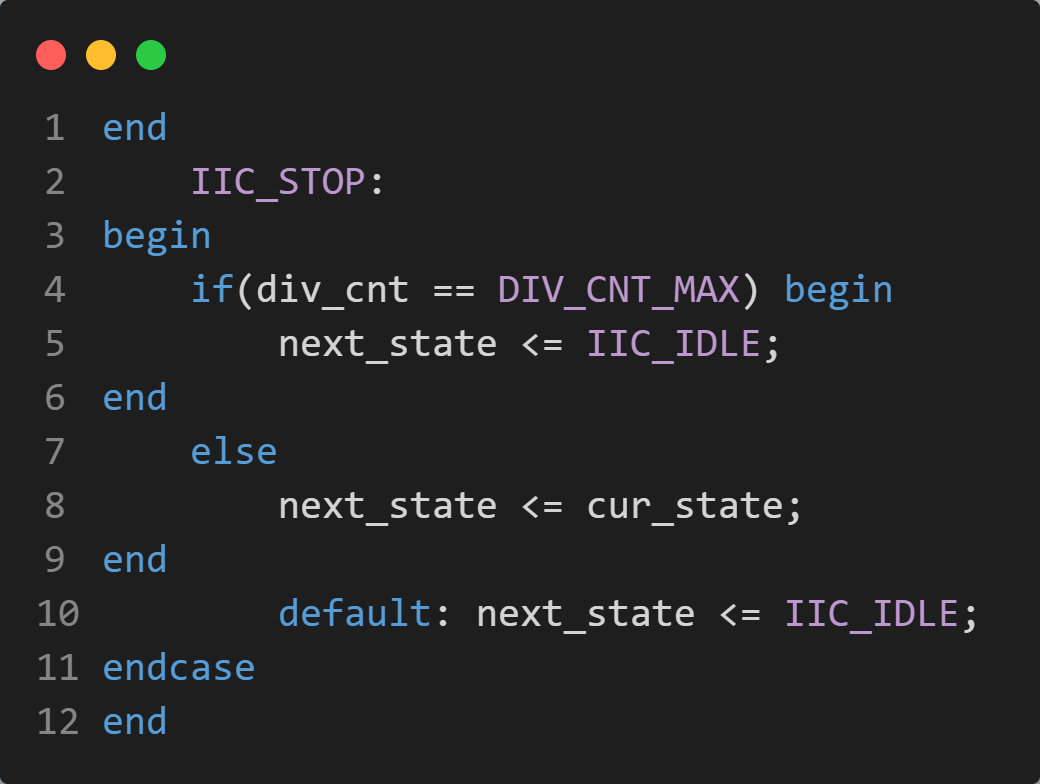
进入IIC\_RD\_DEV,这个阶段发送发送设备地址＋1’b1，表示进行读操作



IIC\_RD\_DATA阶段，当读完设定的字节数跳到stop



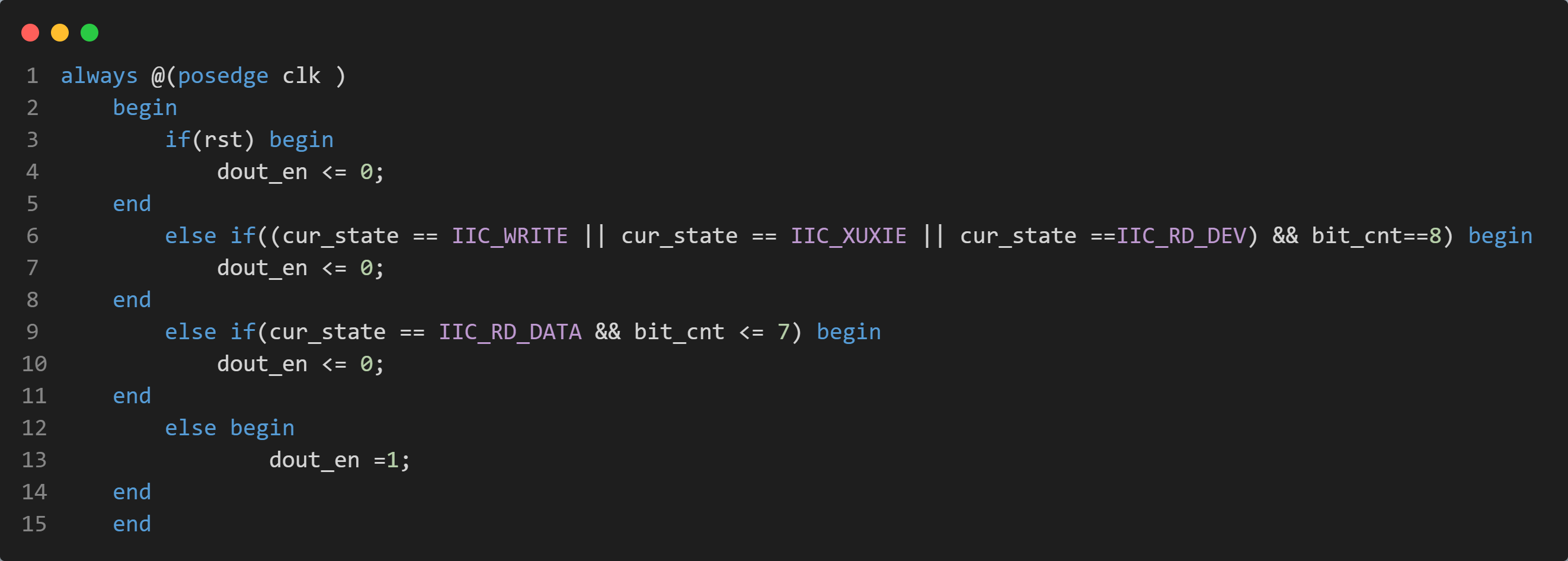
Stop阶段，发送一个iic周期的停止位



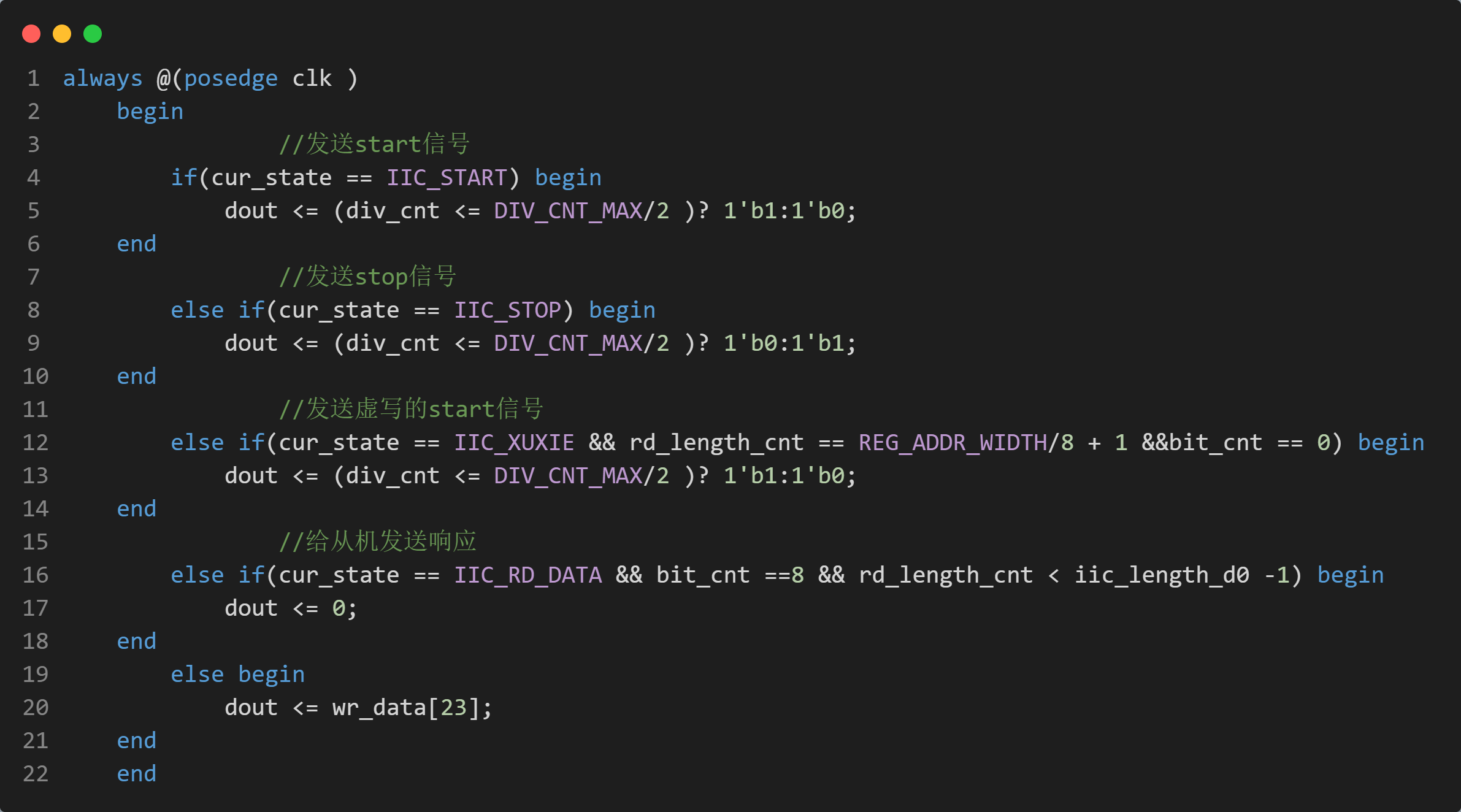
Sda控制：sda为三态门，标准写法如下：当dout\_en为高的时候，sda为输出模式，输出dout的值，当dout为低，处于高阻态，可以接收从机发来的数据



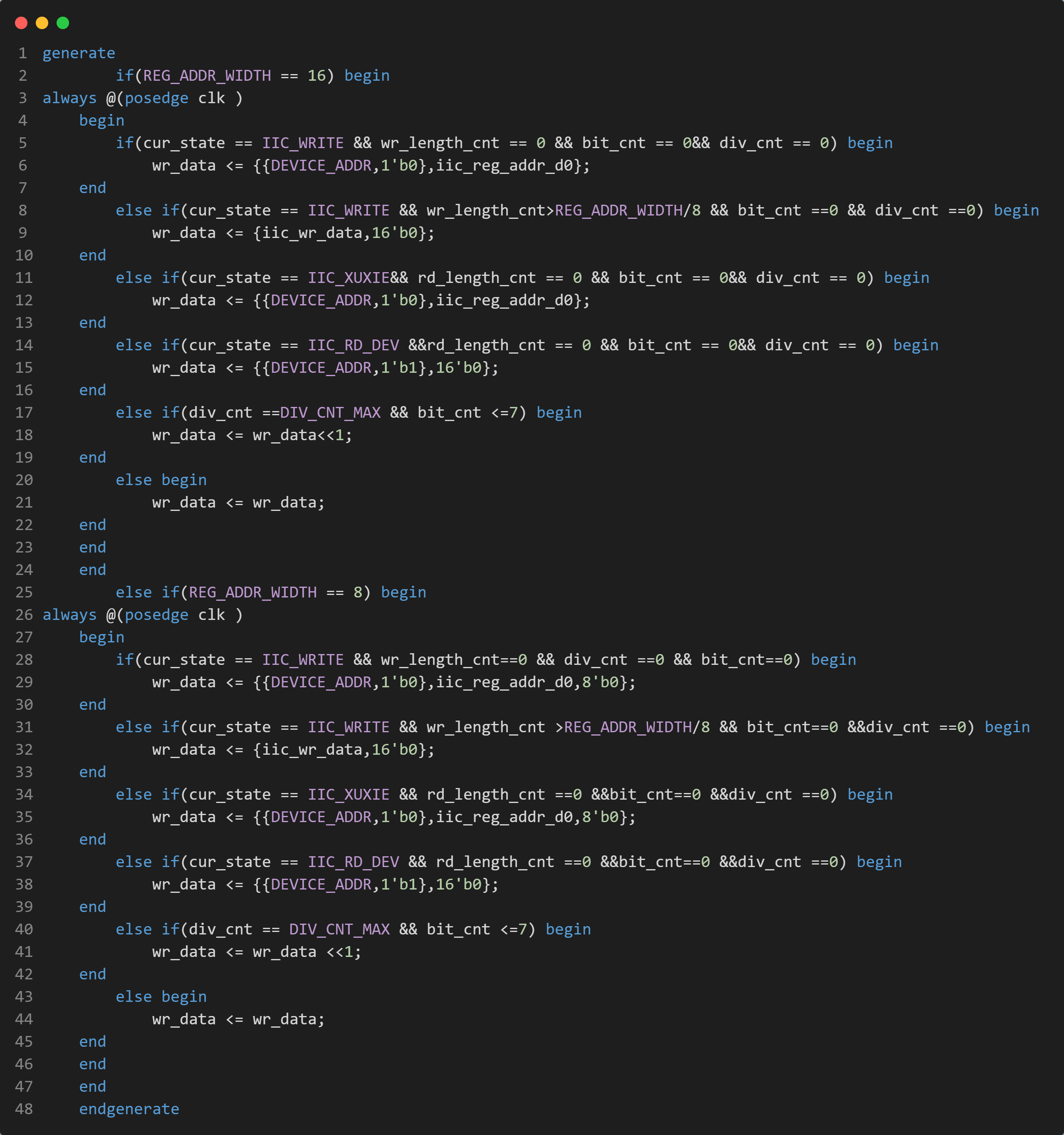
Dout\_en:通过分析时序图可知，在iic的写状态，每写8bit数据，即bit\_cnt==8时，dout\_en拉低，接收从机的响应，虚写和发送读命令也同理，当处于读状态，bit\_cnt<=7时，dout\_en拉低，接收数据，当bit\_cnt>=8,拉高，主机发送1’b1响应



Dout：dout在strat状态，发送起始位，在前半周期输出高，后半周期输出低。在stop状态，发送停止位，前半周期输出低，后半周期输出高。在xuxie状态，当rd\_length\_cnt==1字节设备地址＋内存地址位宽/8，bit\_cnt ==0，这时候发送第二次起始位。在RD\_DATA状态，每当bit\_cnt==8，并且没到读数据最后一个，主机发送响应1‘b0;其他情况dout都是wr\_data的最高位



Wr\_data：



Rd\_data:

