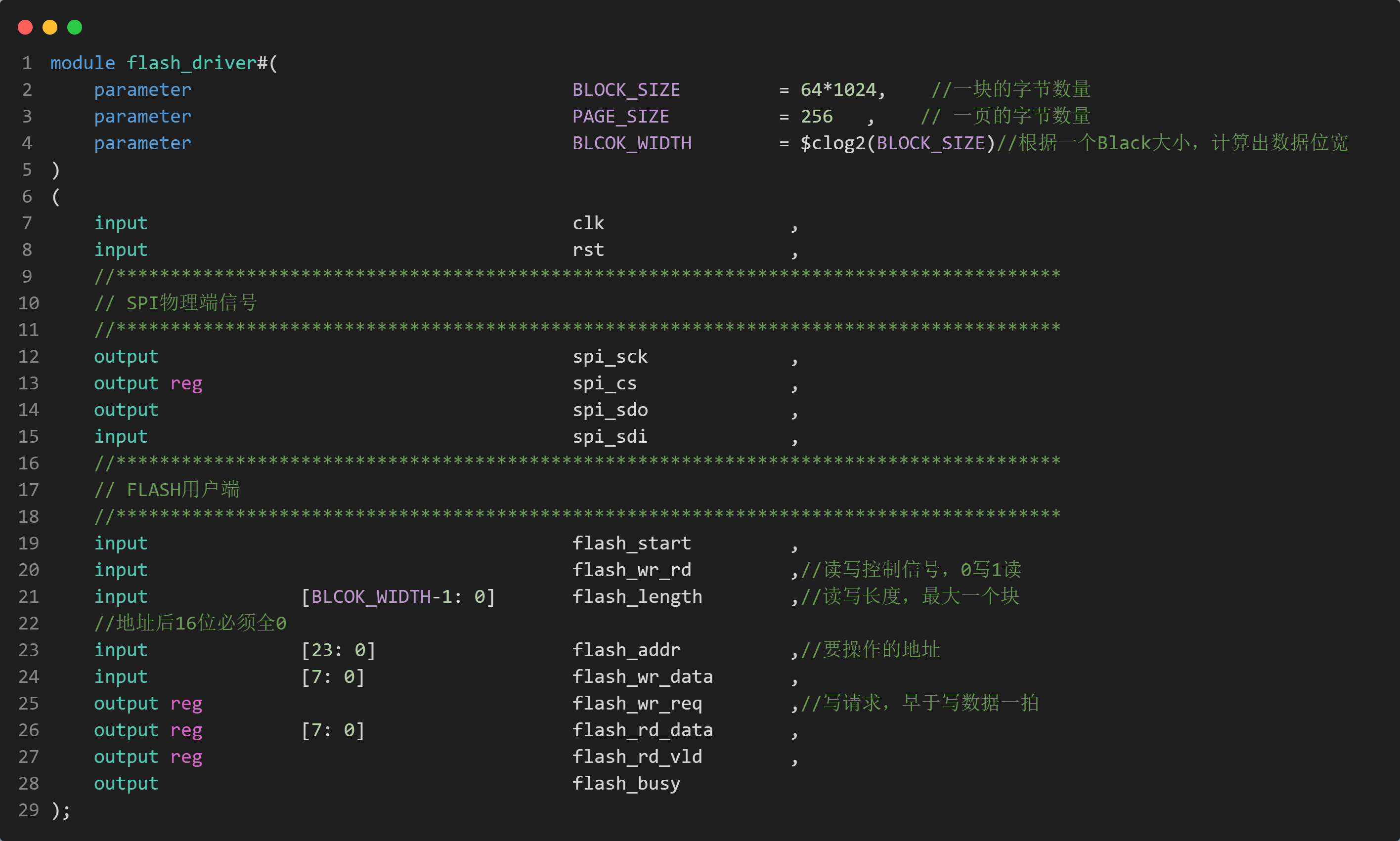
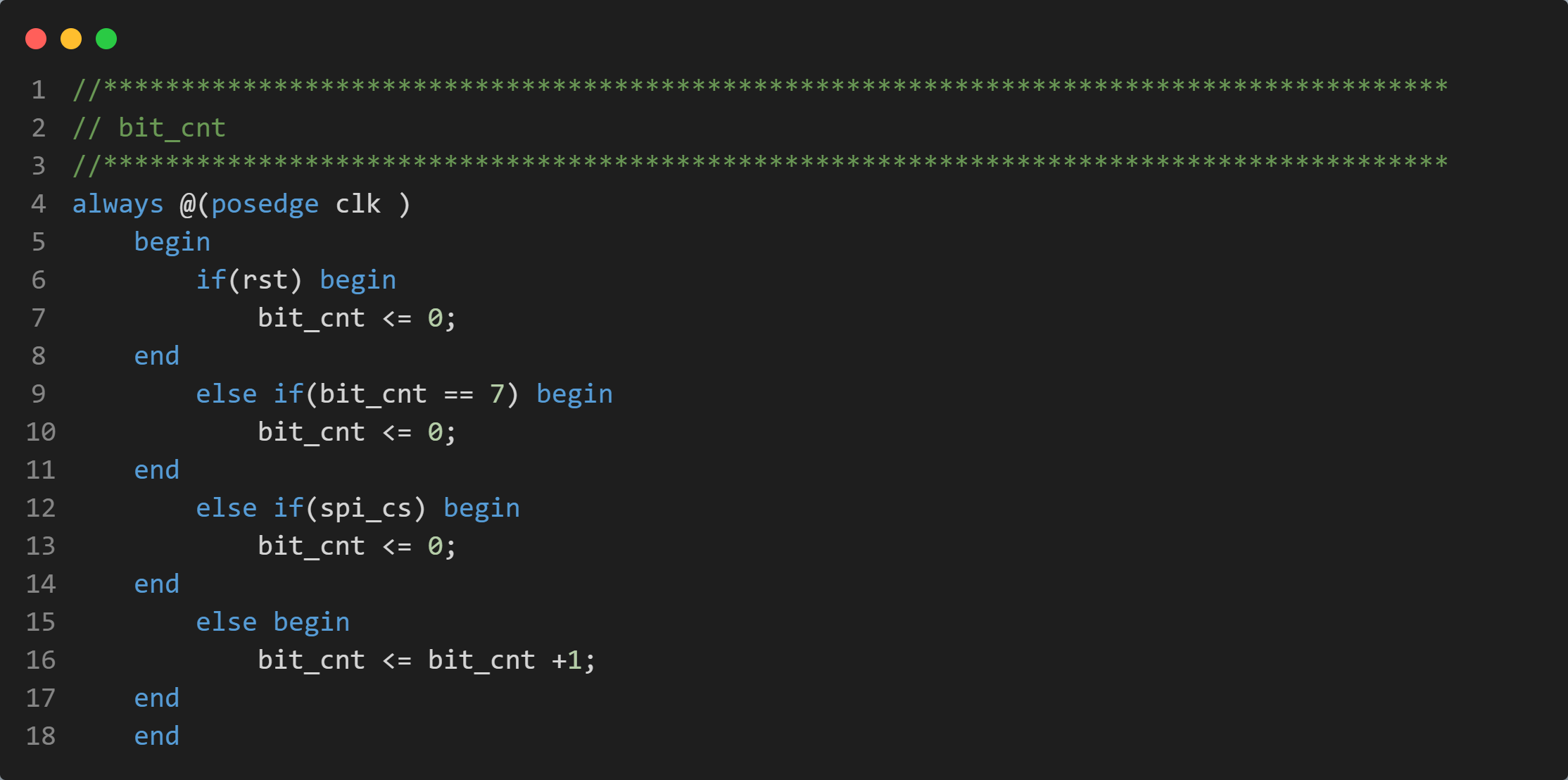
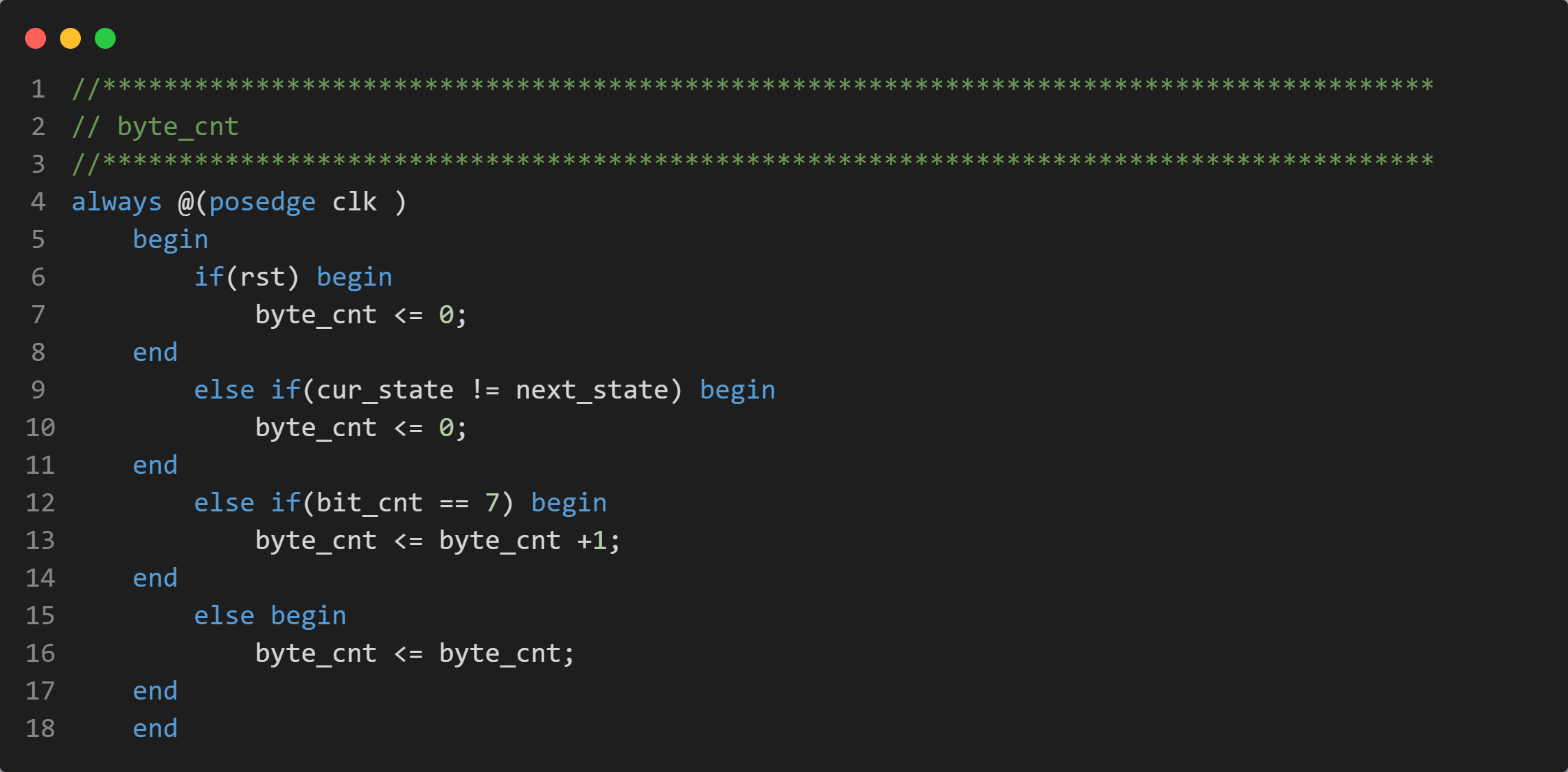
1. 信号定义：



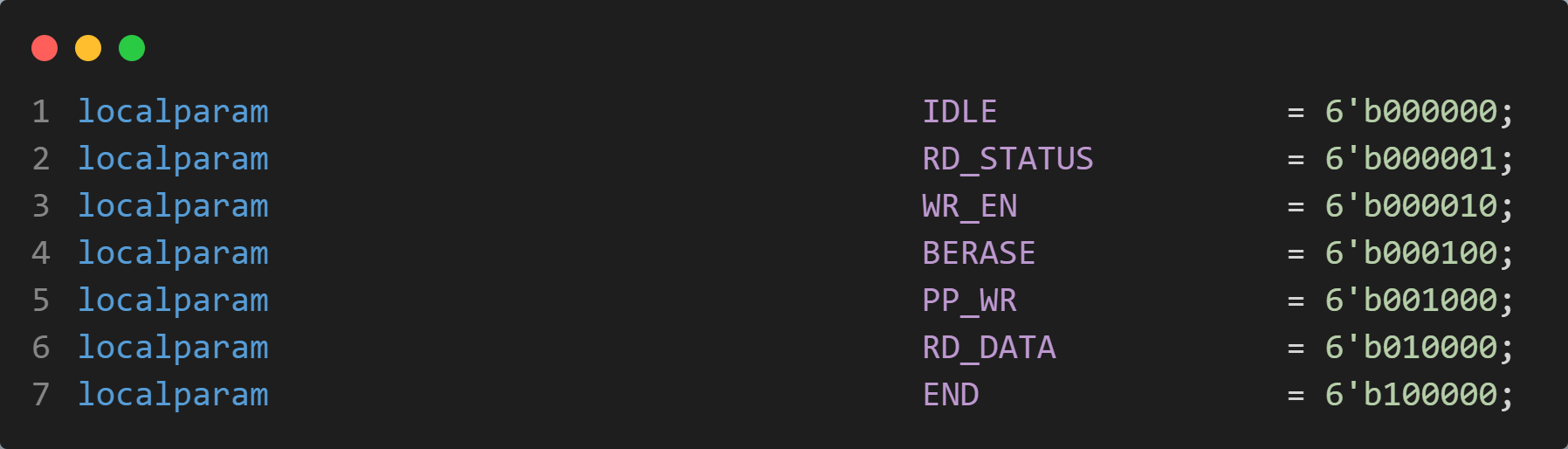
一次最多写一块，每次地址从新一块开始

1. 计数器

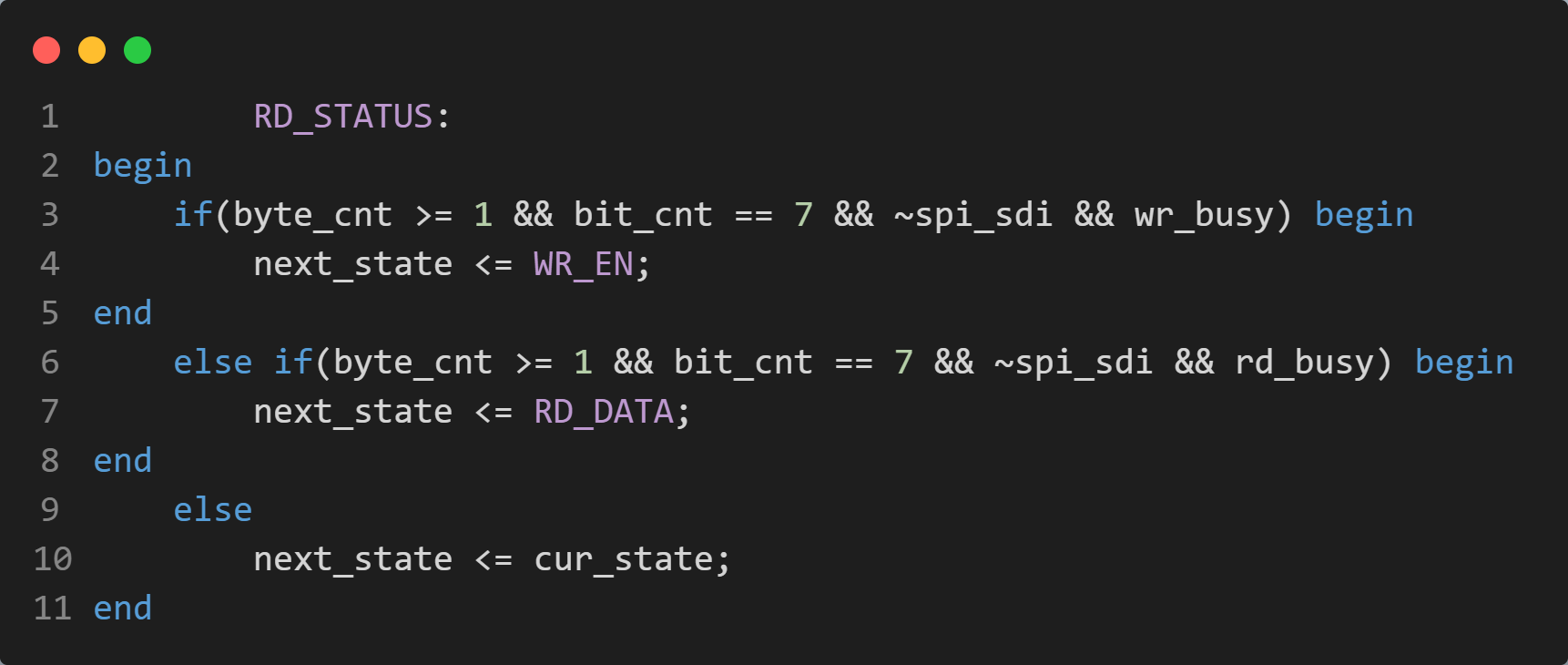


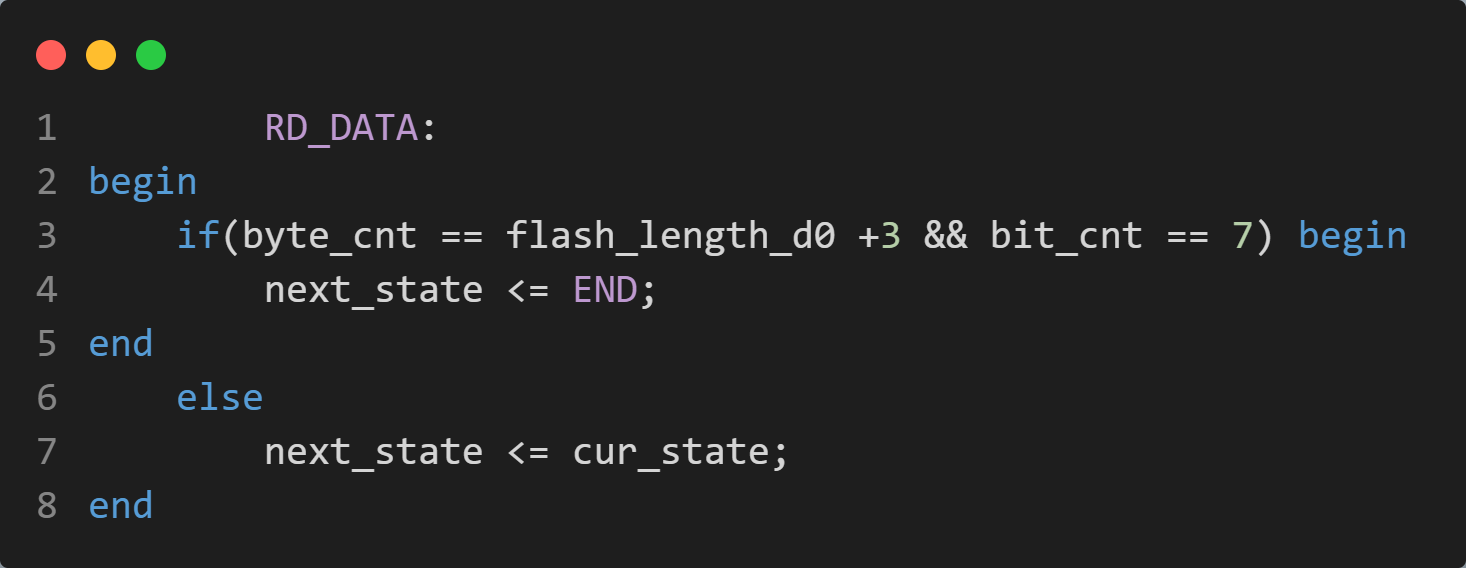


1. 状态机

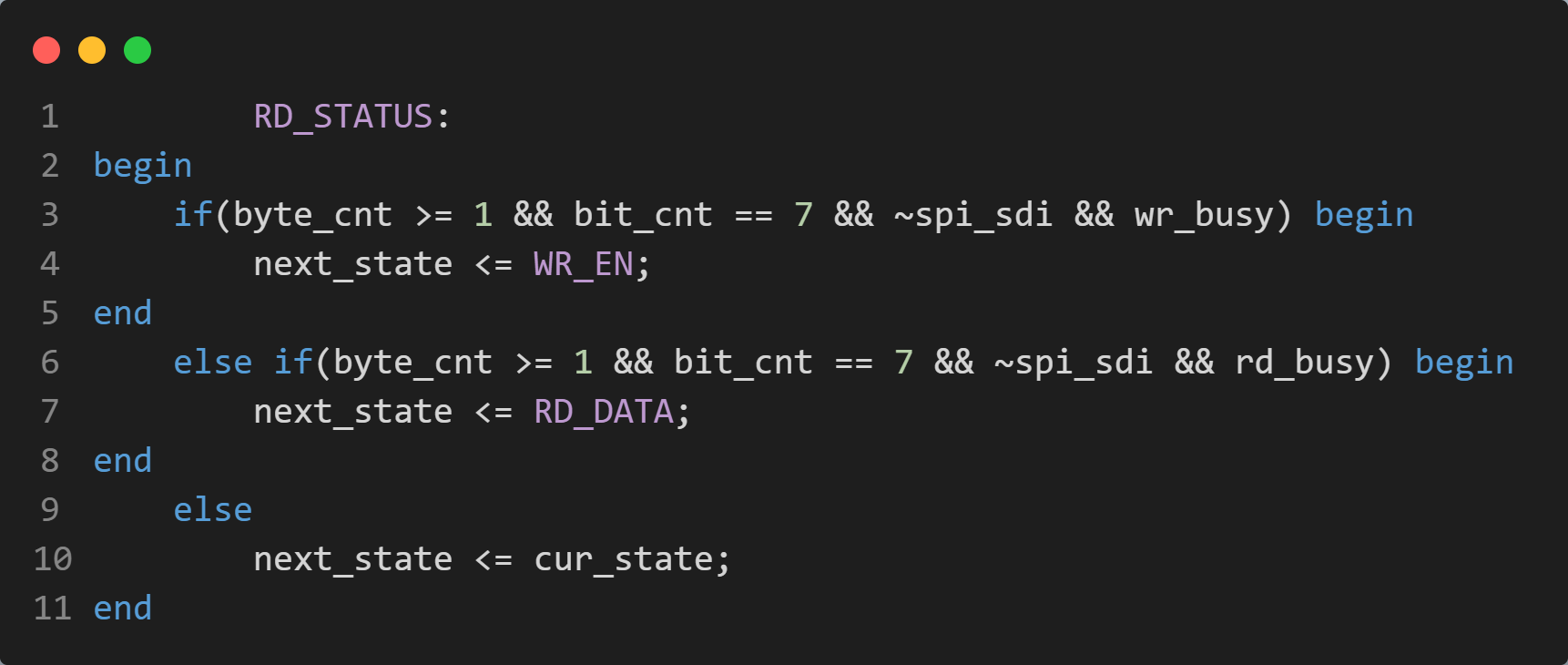


1. Spi\_flash的操作特点是当状态寄存器的busy位为高的时候，不对命令做相应，因此在读写之前先进入读状态寄存器，busy为低再继续。
2. 读的过程：先看读状态寄存器，然后进入读状态，当byte\_cnt == 1个字节读命令+三位地址＋flash\_length -1且bit计数器计满，进入end状态，同样的，进入读数据状态后，每过7bit拉高读有效标准

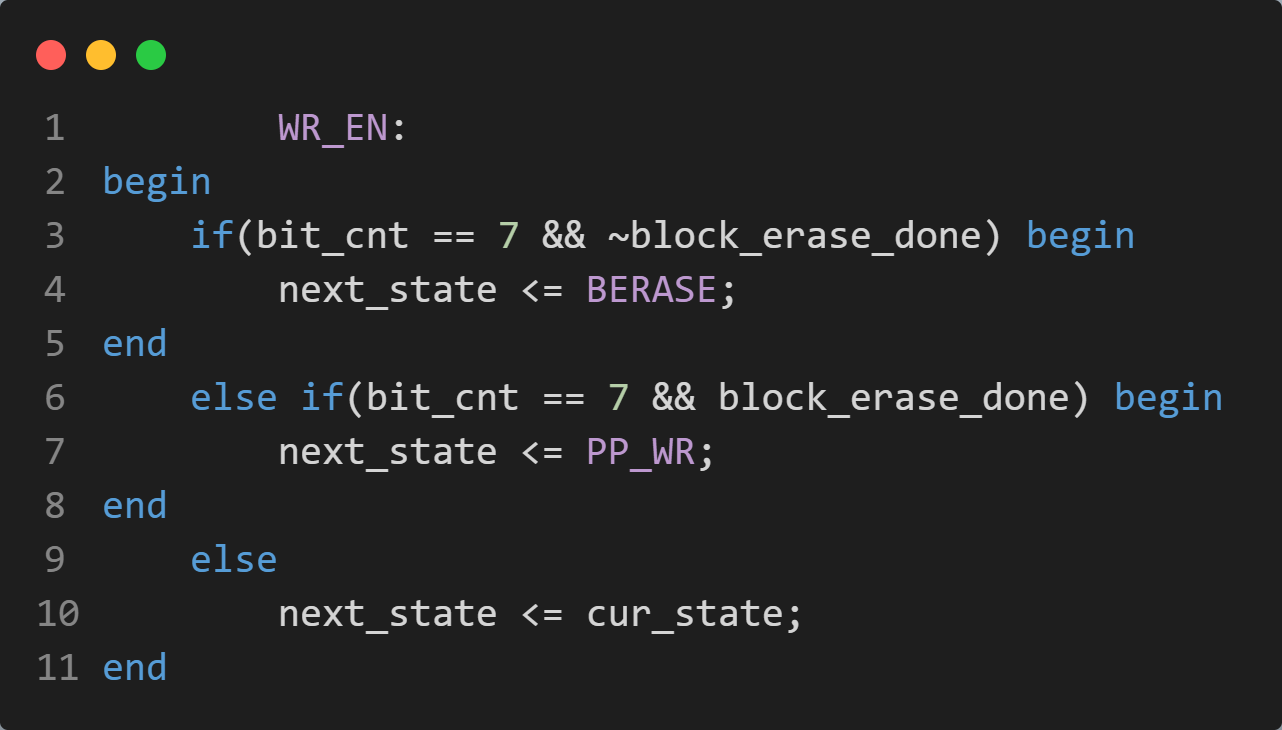


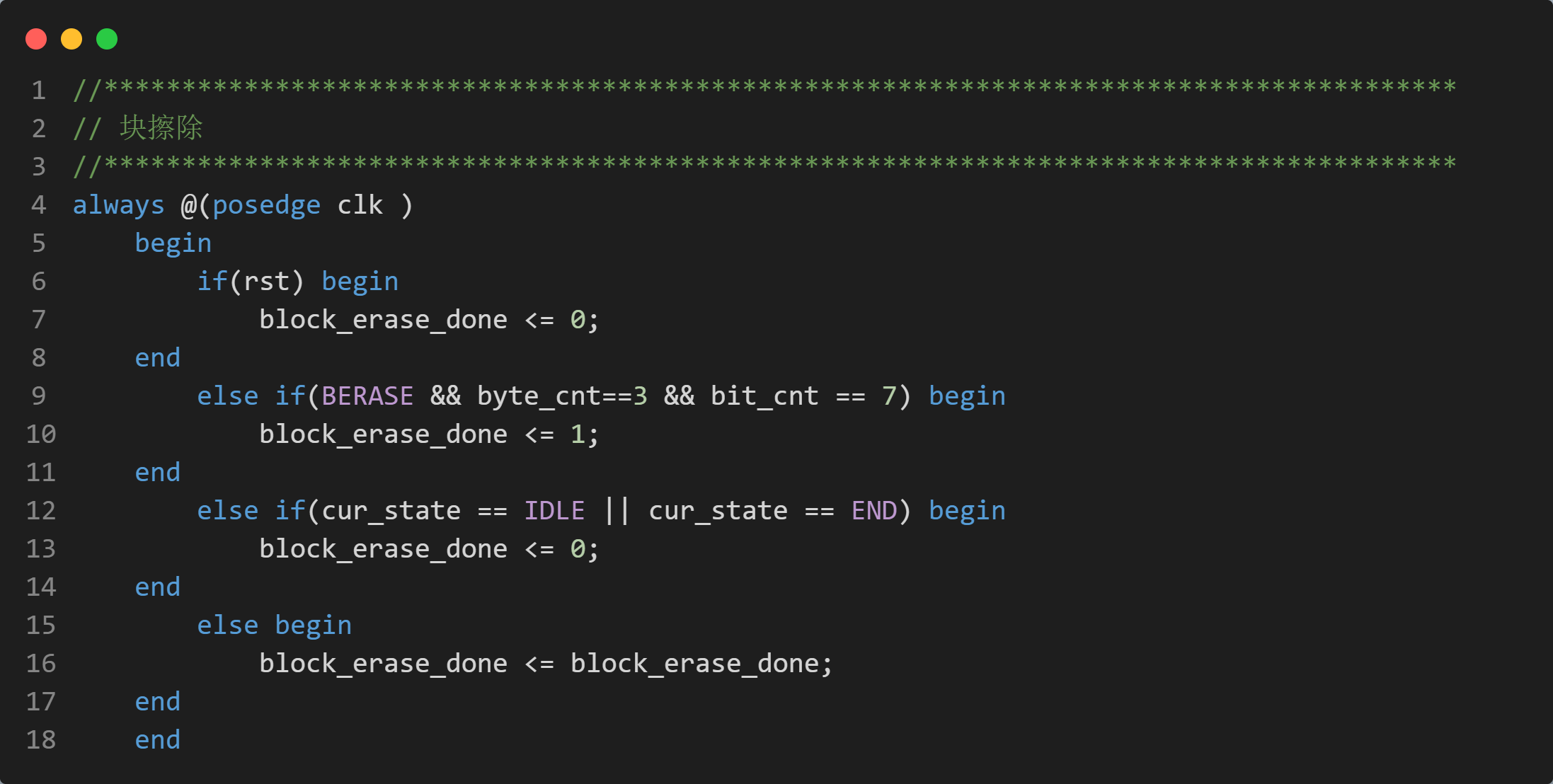


1. 写过程：写数据之前要进行擦除，擦除空间的大小就是你可以一次最多写的大小，我们选择块擦除，首先进入读状态寄存器，等收到busy位为7，即可进入 写使能，

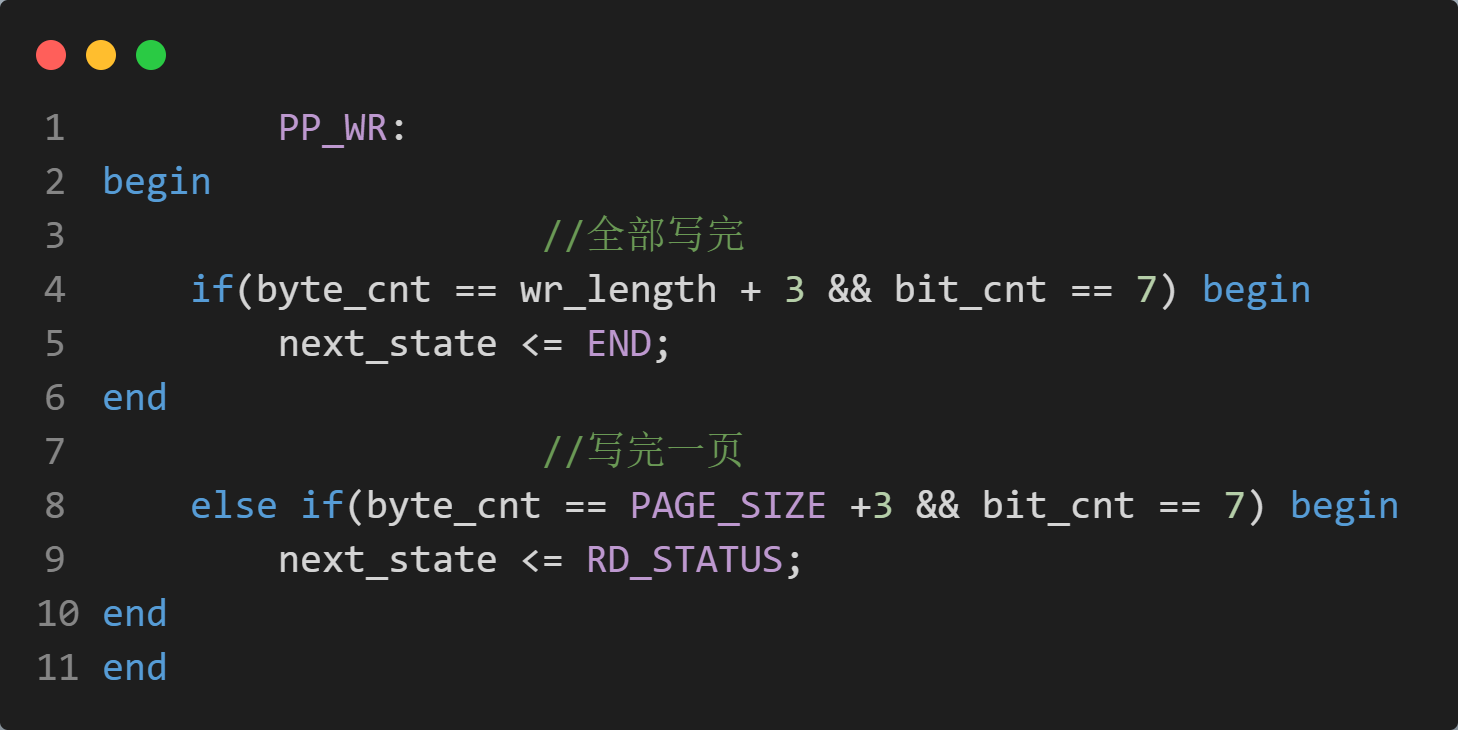


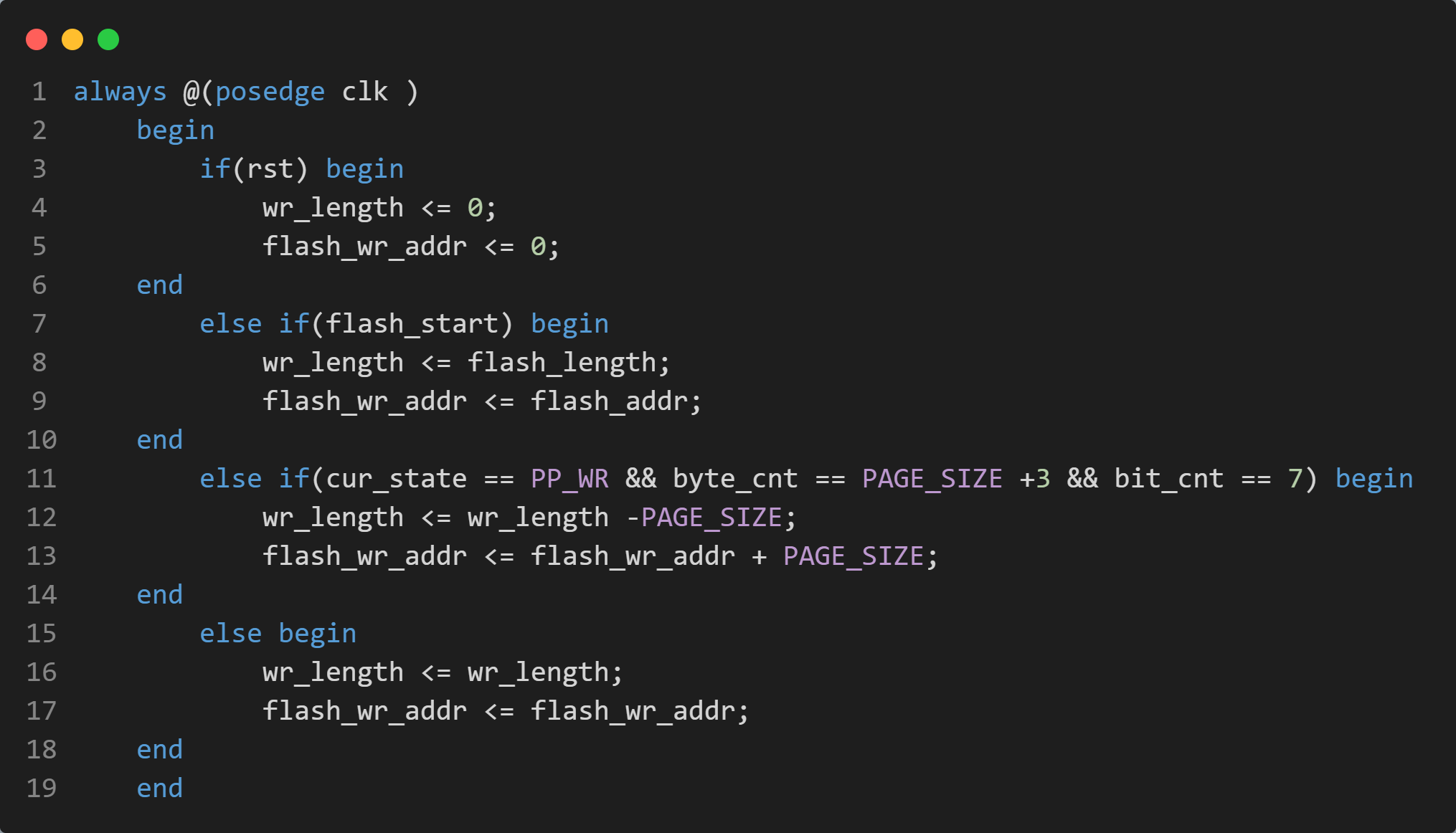
1. 发送完成写使能命令后判断是否进行了擦除，如果没有，先进行擦除然后进入页写状态，因为spi\_flash一次最多写入一个页大小的数据，



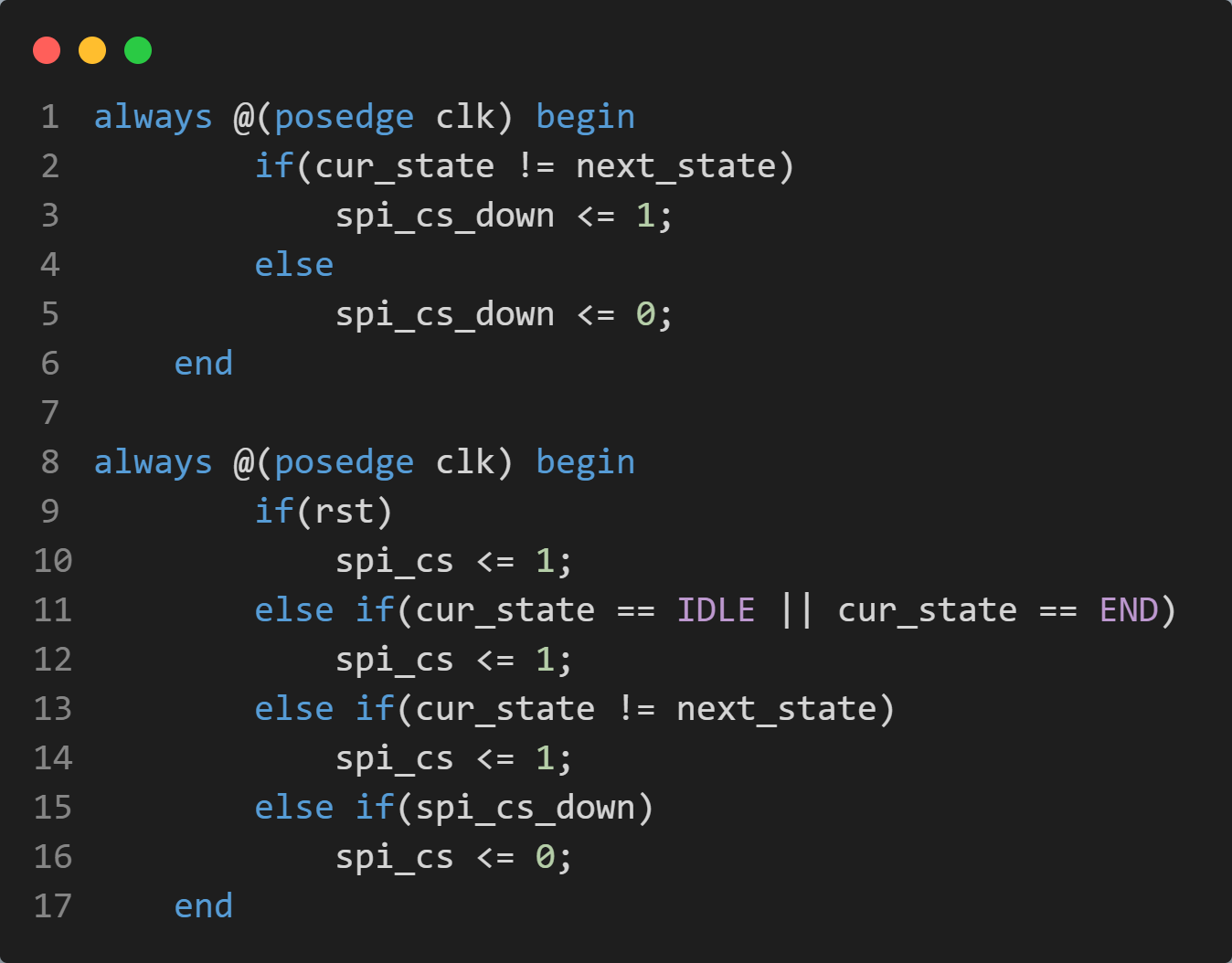


1. 因此byte\_cnt == 页大小+3就跳出页写，进入读状态寄存器，等待数据搬运完毕在重复，同时我们定义一个wr\_length计数器，start信号来的时候存上要写的数据长度，每当byte\_cnt == 页大小+3，此时已经写了页大小的数据，wr\_length – 页大小，这样就可以写完所有的数据。同理定义一个wr\_addr，一开始为用户输入的首地址，每当byte\_cnt == 页大小+3，地址加页大小。





1. Cs:



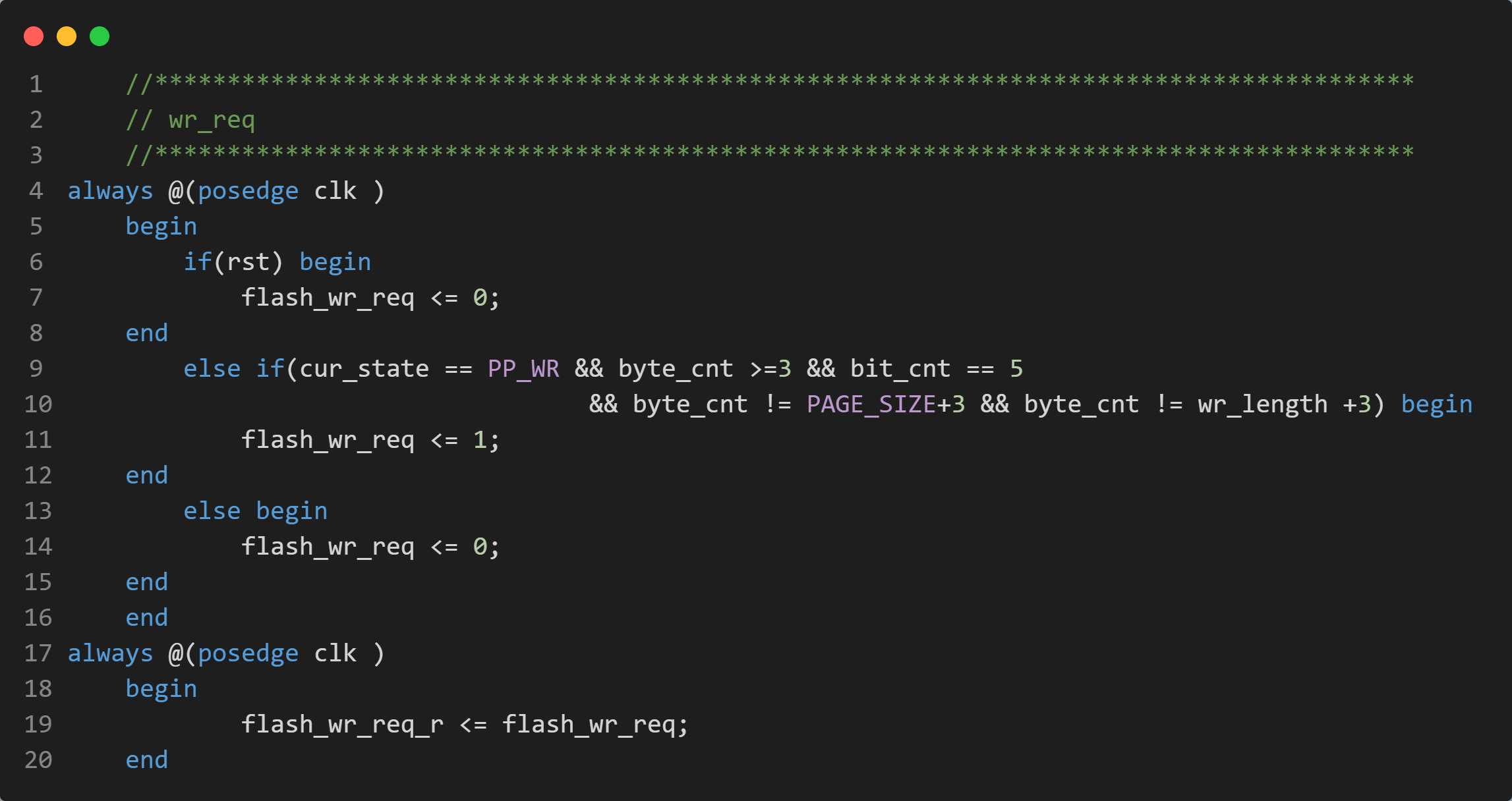
1. 输出



1. wr\_Data:



1. wr\_req：每当进入页写状态并且发送完毕命令和地址后每过5个bit拉高req，这时候wr\_data <= {flash\_wr\_data,24'b0};



1. rd\_data:当进入读数据状态，把sdi发送的数据进行拼接（先发送高位），每过7个bit，发送读数据有效标志

