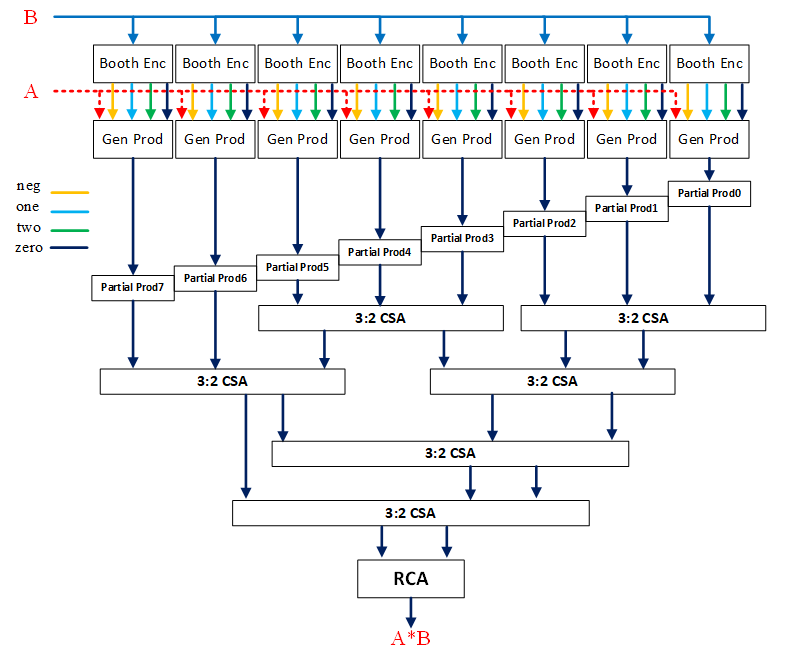
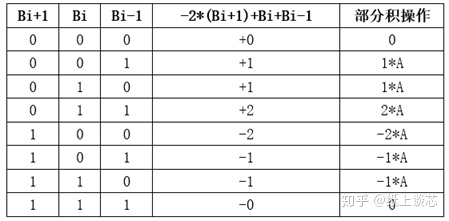
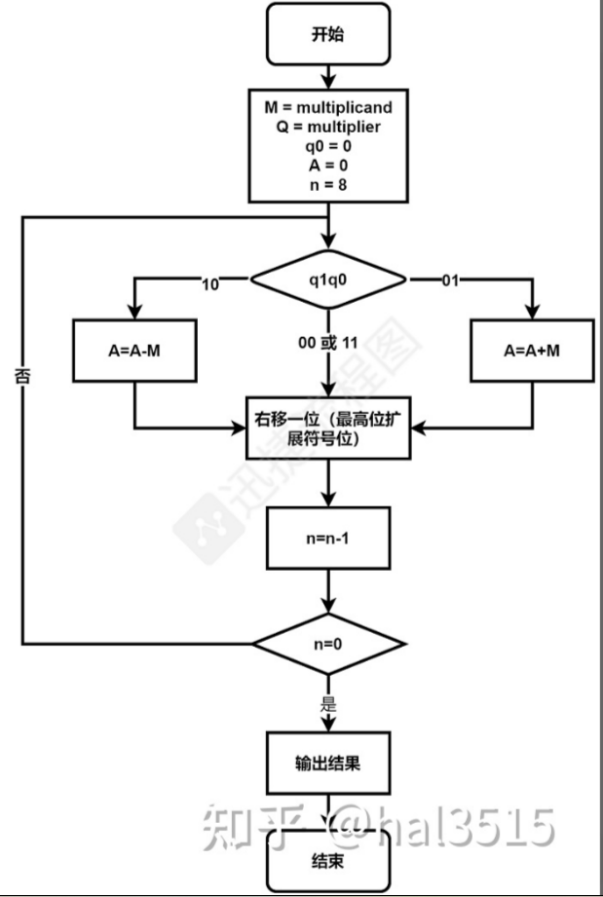
**数字逻辑与数字系统**

**第四次作业**

## 学习基于Booth编码器的乘法器，并画出模块图。

基于Booth编码器的乘法器通过减少部分积数目和加法次数实现了较快的计算速度。假设有一个8位乘数，它的二进制值为0111\_1110，它的6个1将产生6个非零部分积。我们在原值末尾加辅助位0，变为0111\_1110\_0，然后利用低位减去高位将这个二进制值改为1000\_00-10，从而我们只需要计算两个部分积的和。这就是Booth编码。基4 Booth编码表如图，我们将乘数B中的每三位转化为Booth编码后计算部分积。一种电路图如图。

另外，亦可利用Booth算法以移位的方式计算乘法，过程如下：以4位二进制数为例，n=4。设两乘数分别为a和b，则任取其中一乘数a使A=a，B=b0000。q=0。重复n次以下操作：若Aq的末位为00或01，向右移位；若Aq的末位数为10，A=A-B（加上-B的补码）后A向右移位；若Aq的末位数为01，A=A+B后移位。如此即得到A与B的积。计算流程图如图所示。

参考文献：

[1] 【HDL系列】乘法器(6)——Radix-4 Booth乘法器. <https://zhuanlan.zhihu.com/p/143802580>

[2] FPGA的算法解析2：乘法器（基础乘法器+Booth乘法器+LUT乘法器）. <https://zhuanlan.zhihu.com/p/433924897>

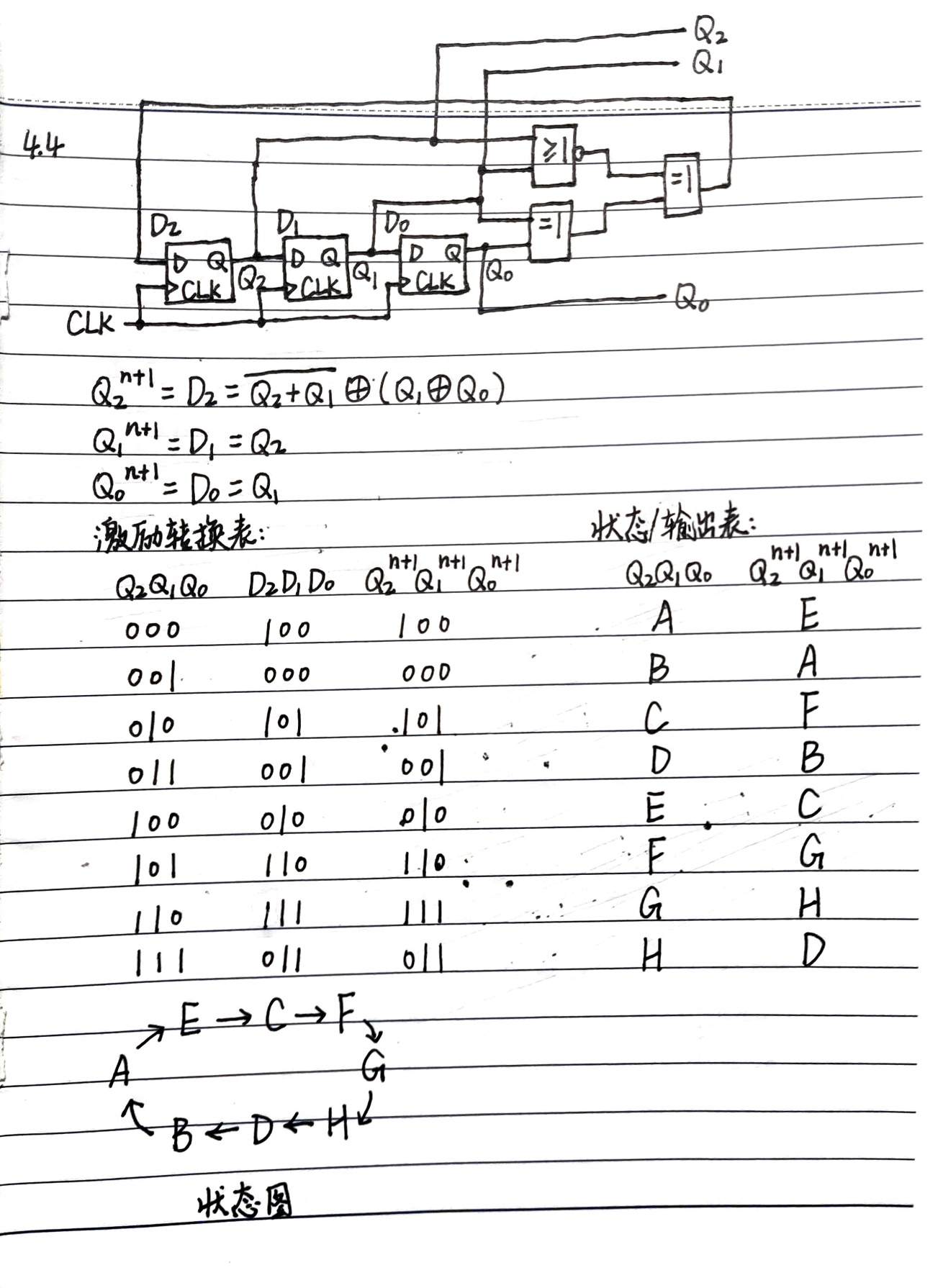
[3] 【龙芯班笔记】基于booth二位乘的八位乘法器. <https://zhuanlan.zhihu.com/p/424668223>

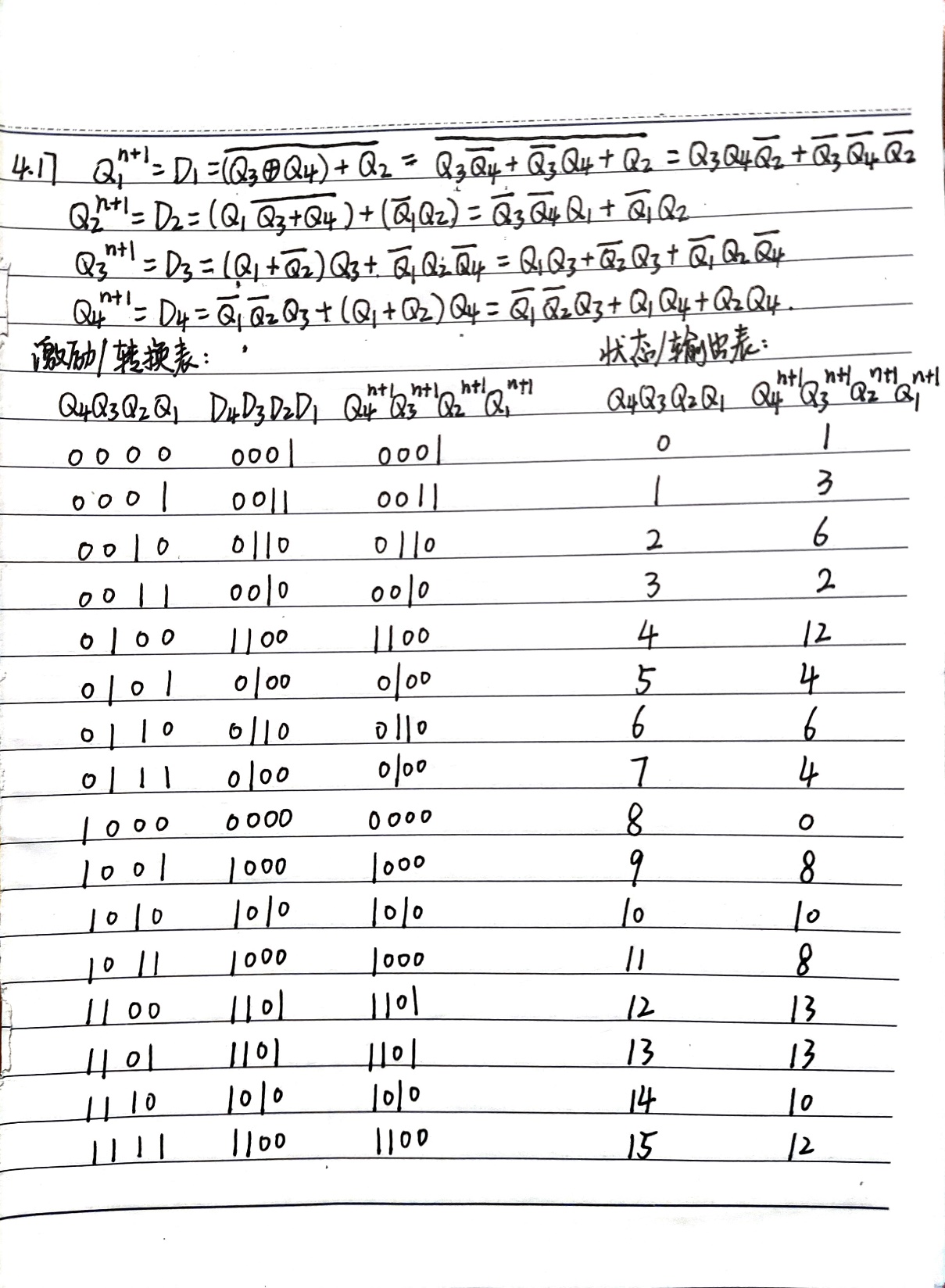
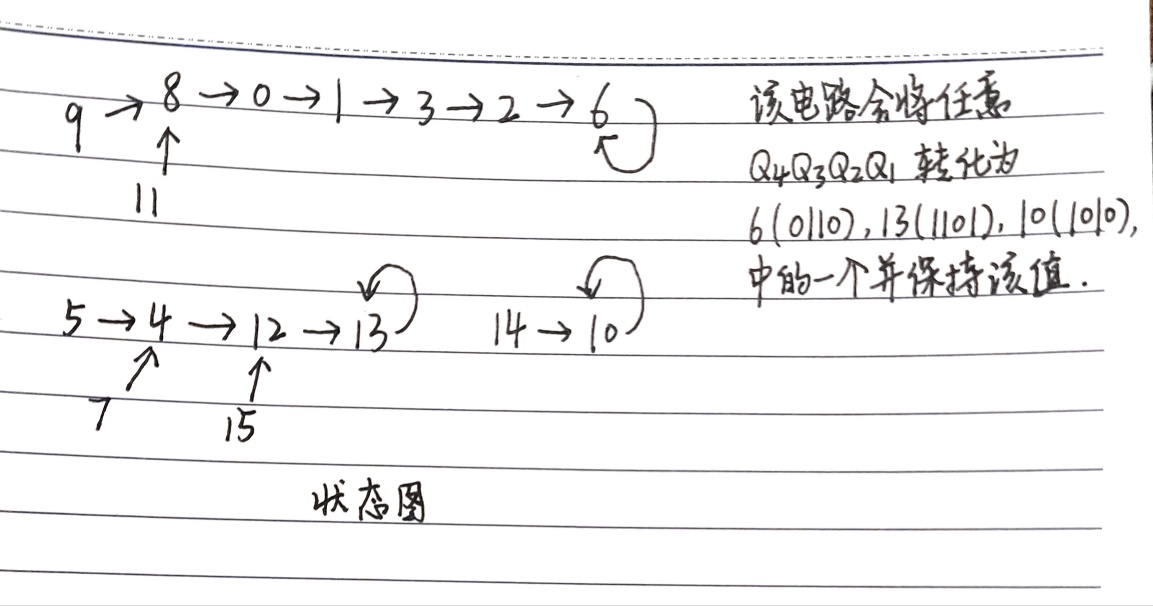
[4] booth编码乘法器. <https://blog.csdn.net/weixin_46076353/article/details/118194302>

[5] Booth's multiplication algorithm-Wikipedia. <https://en.wikipedia.org/wiki/Booth%27s_multiplication_algorithm>

## 课后习题

**4.4 分析习图4.4所示同步电路。写出激励方程、激励/转换表及状态/输出表。设对应的状态名分别为。**

****

**4.17 试分析习图4.11所示电路，分别写出激励函数、激励转换表和状态表（图），说明此电路的逻辑功能。**