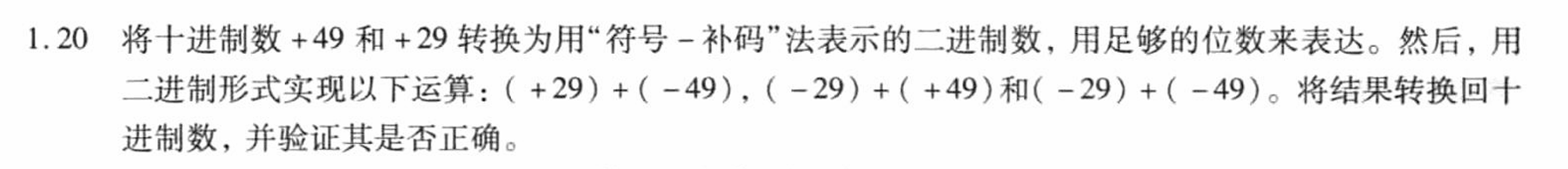
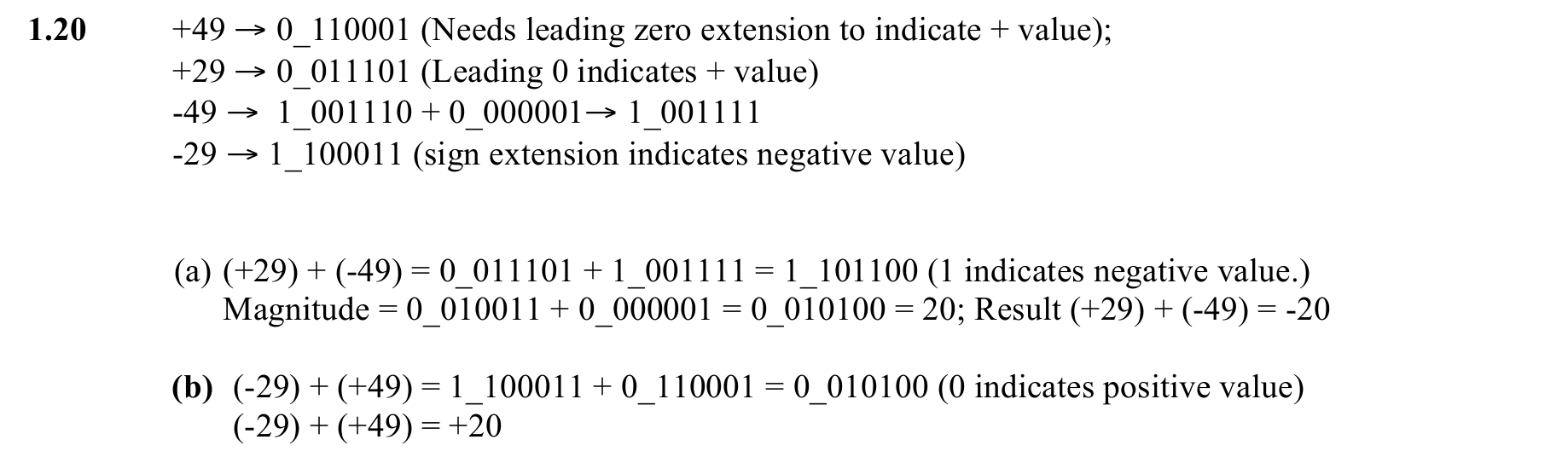
Chapter1

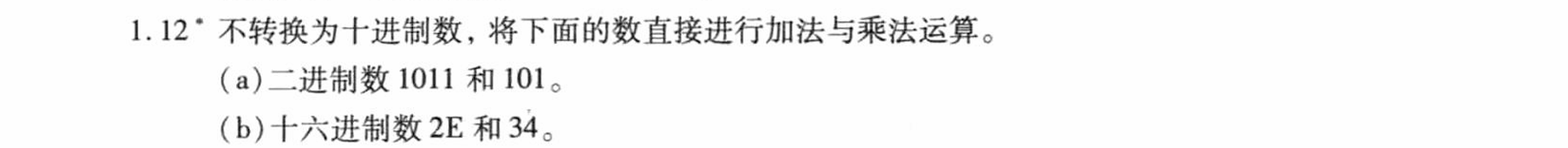


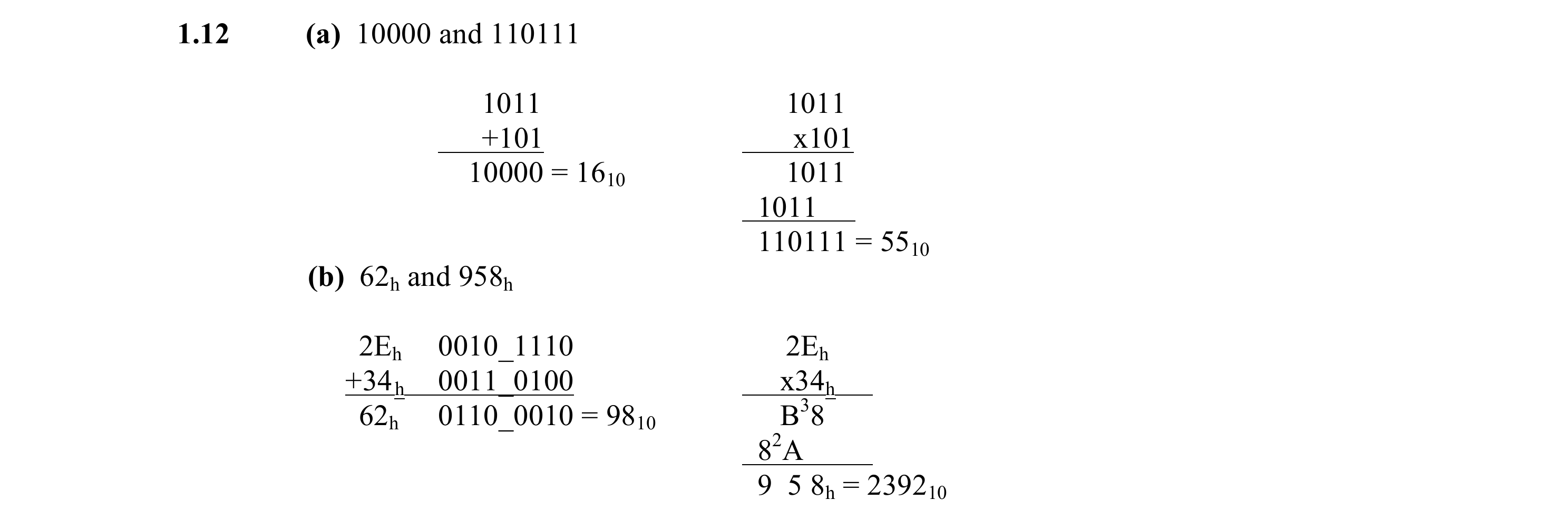
十进制： 0代表正数；9代表负数

二进制： 0代表正数；1代表负数

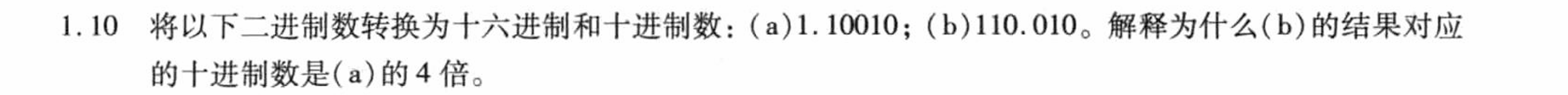


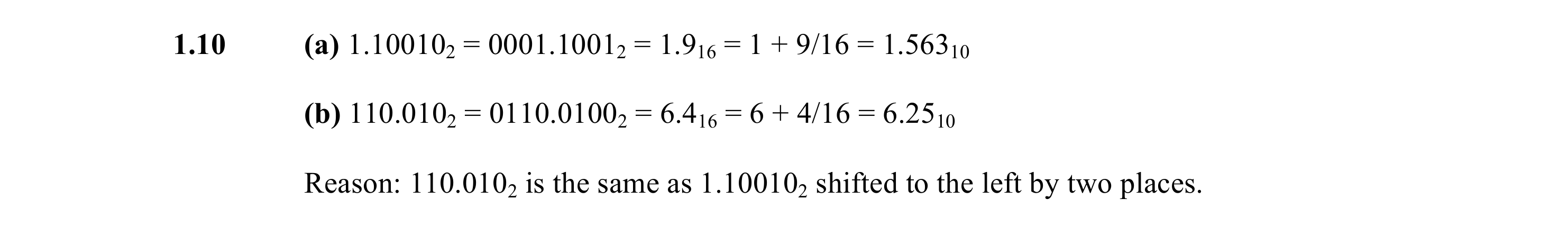




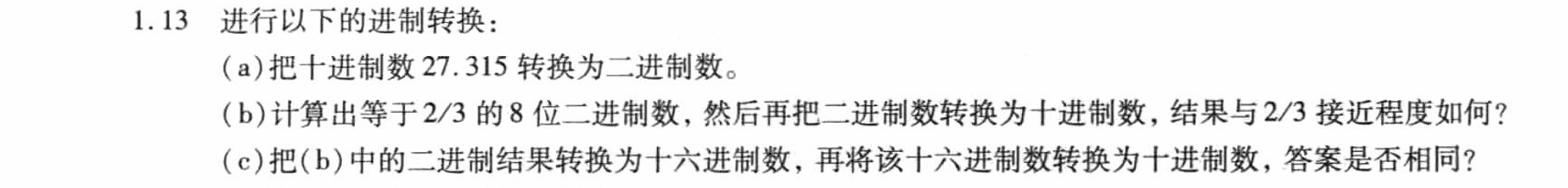


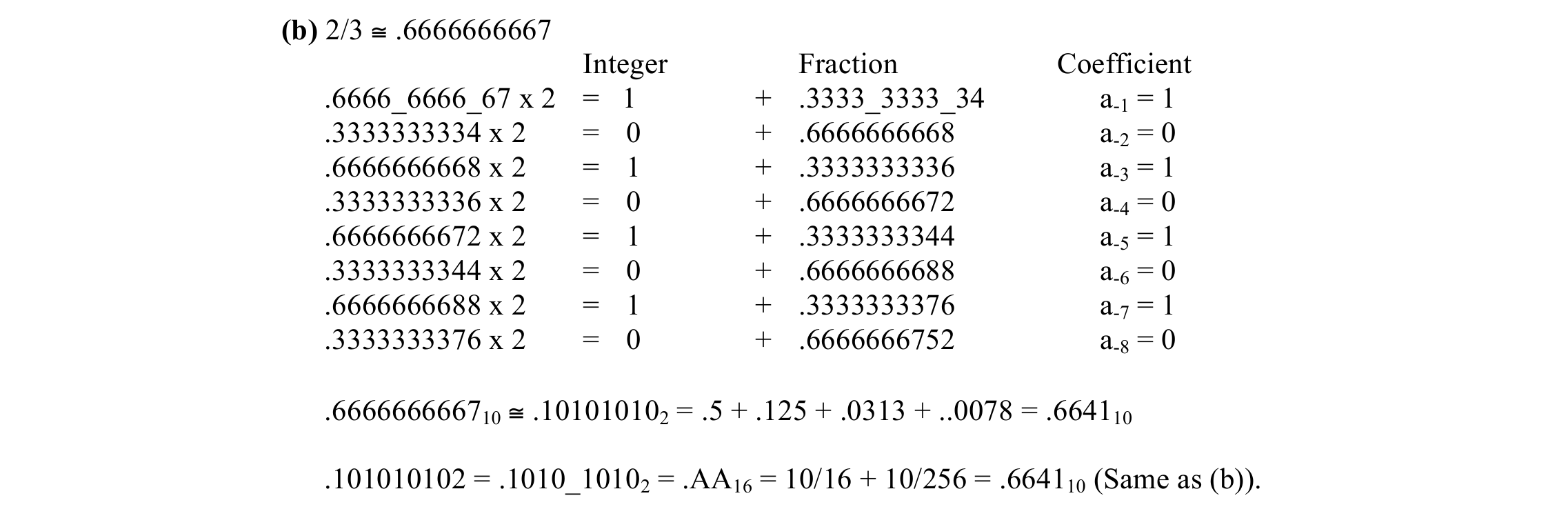
E(14)\*4=56 56/16=3……8



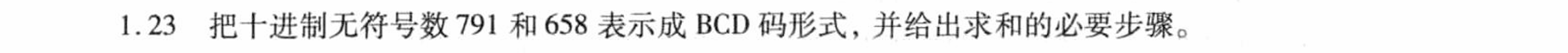


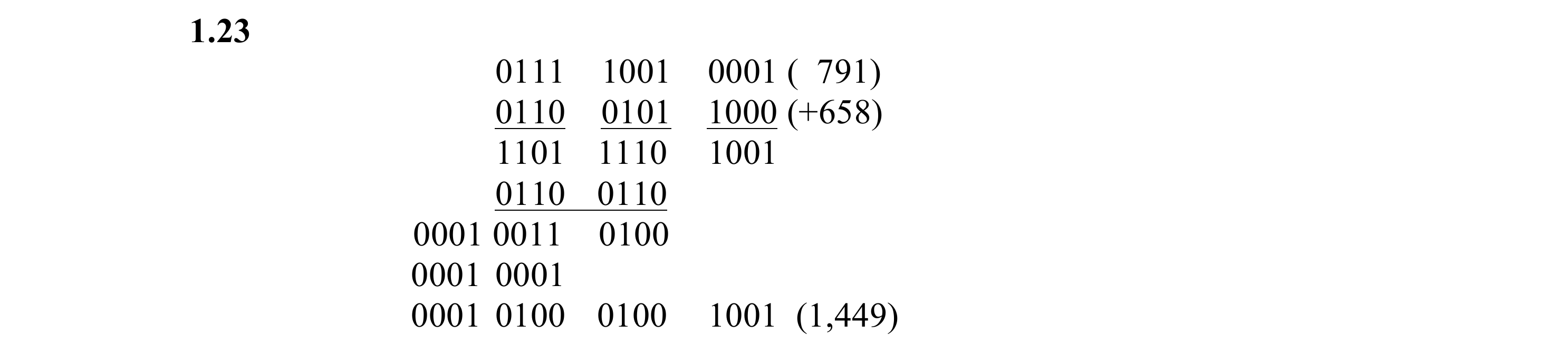
从小数点向左右两端四个一组，从左往右将二进制转换成十六进制，



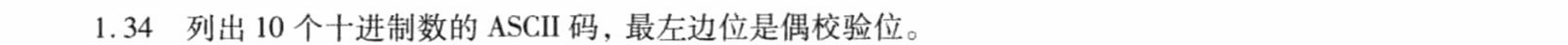


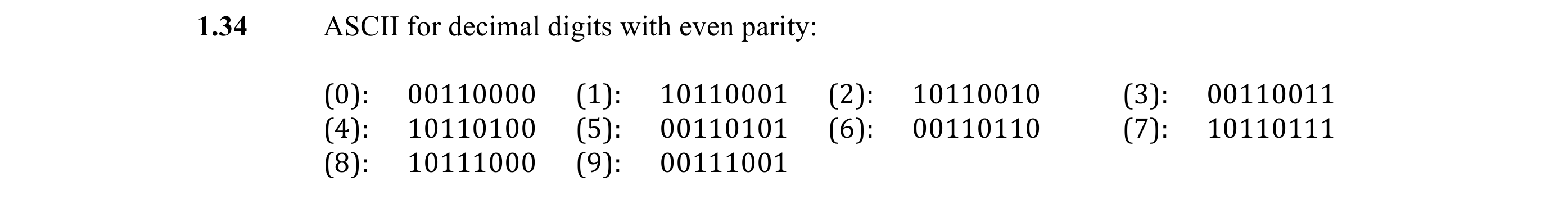
关注小数的二进制转换是连乘





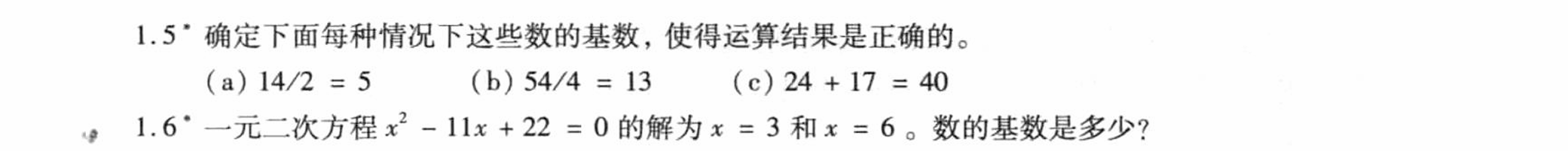
进位的时候向前四位进1

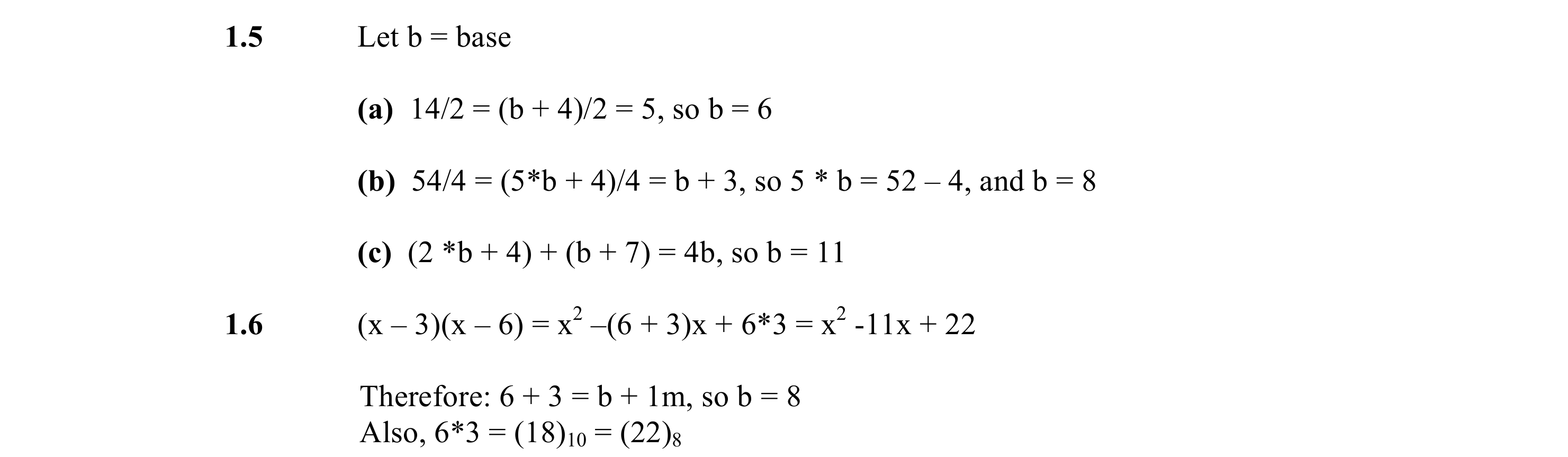




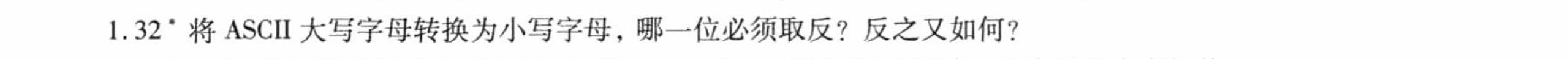
ASCII 码的偶校验位（Even Parity Bit）是 在 7 位 ASCII 码的基础上增加的第 8 位，

它的作用是：让整个 8 位数据中 “1” 的个数为偶数。





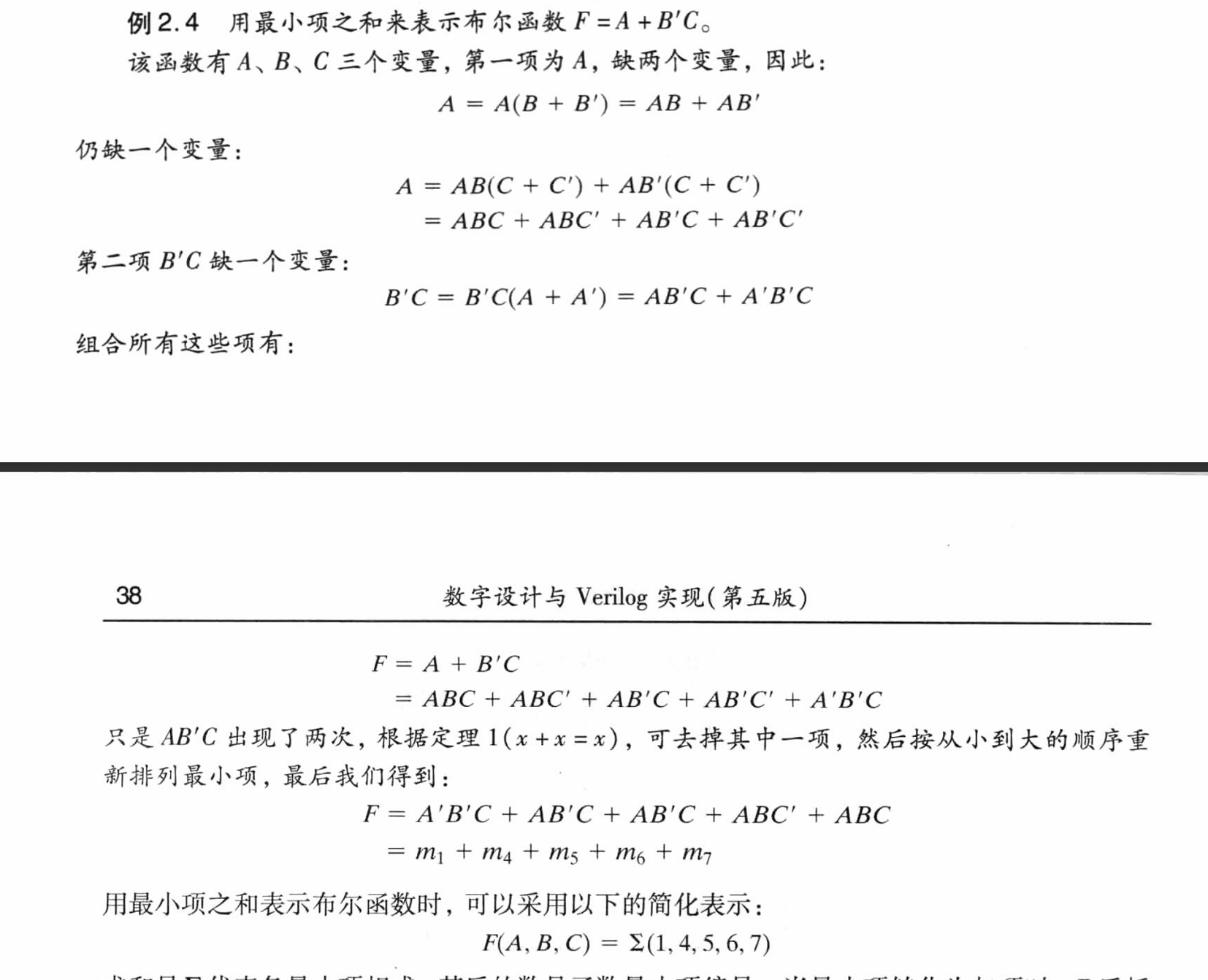
基数就是指是几进制的



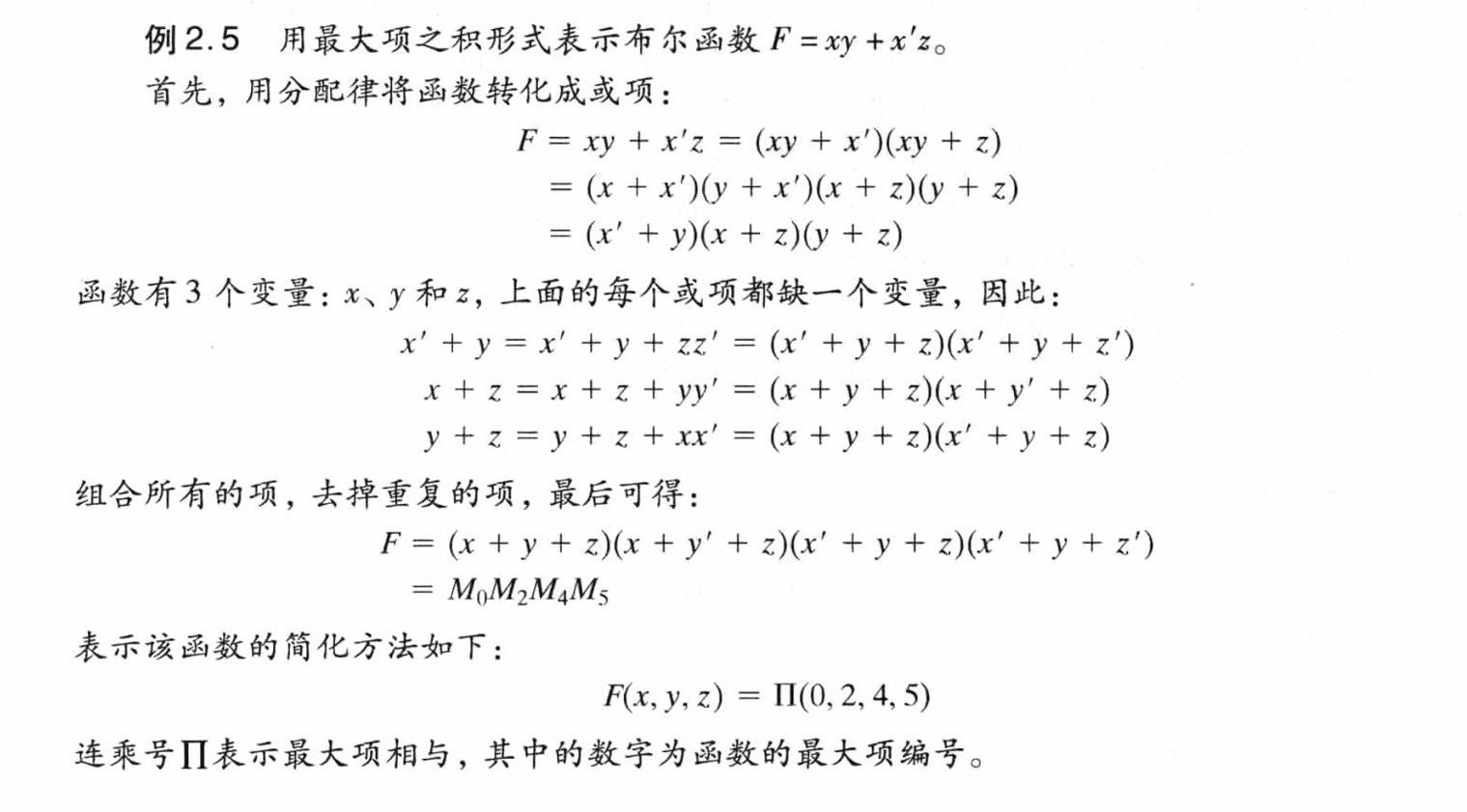


记住就行

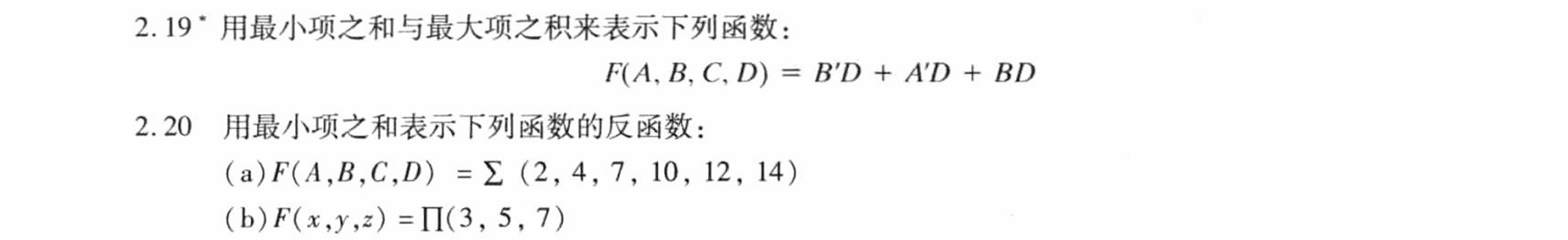
Chapter2

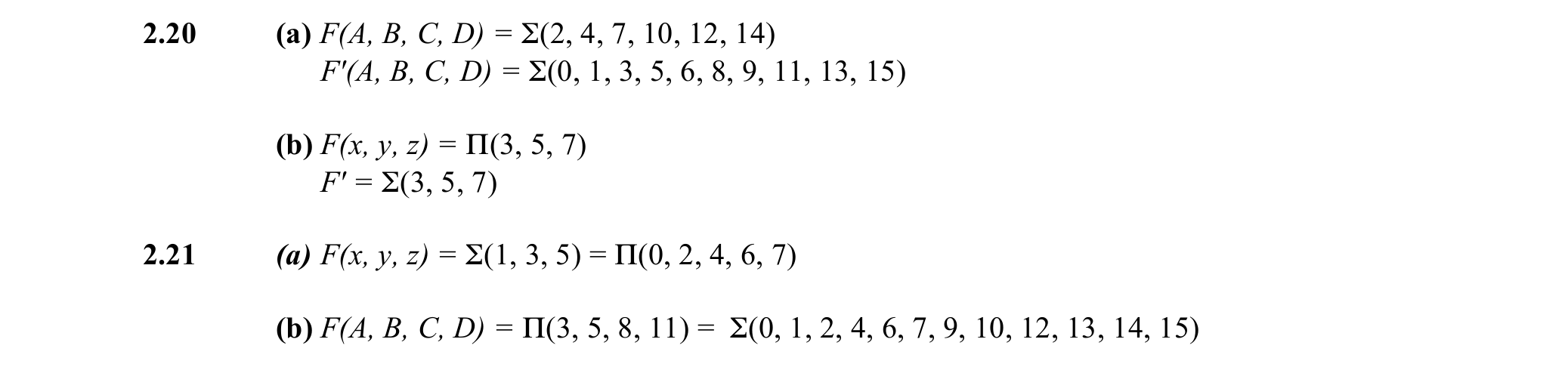


先用卡诺图求解答案，再进行化简

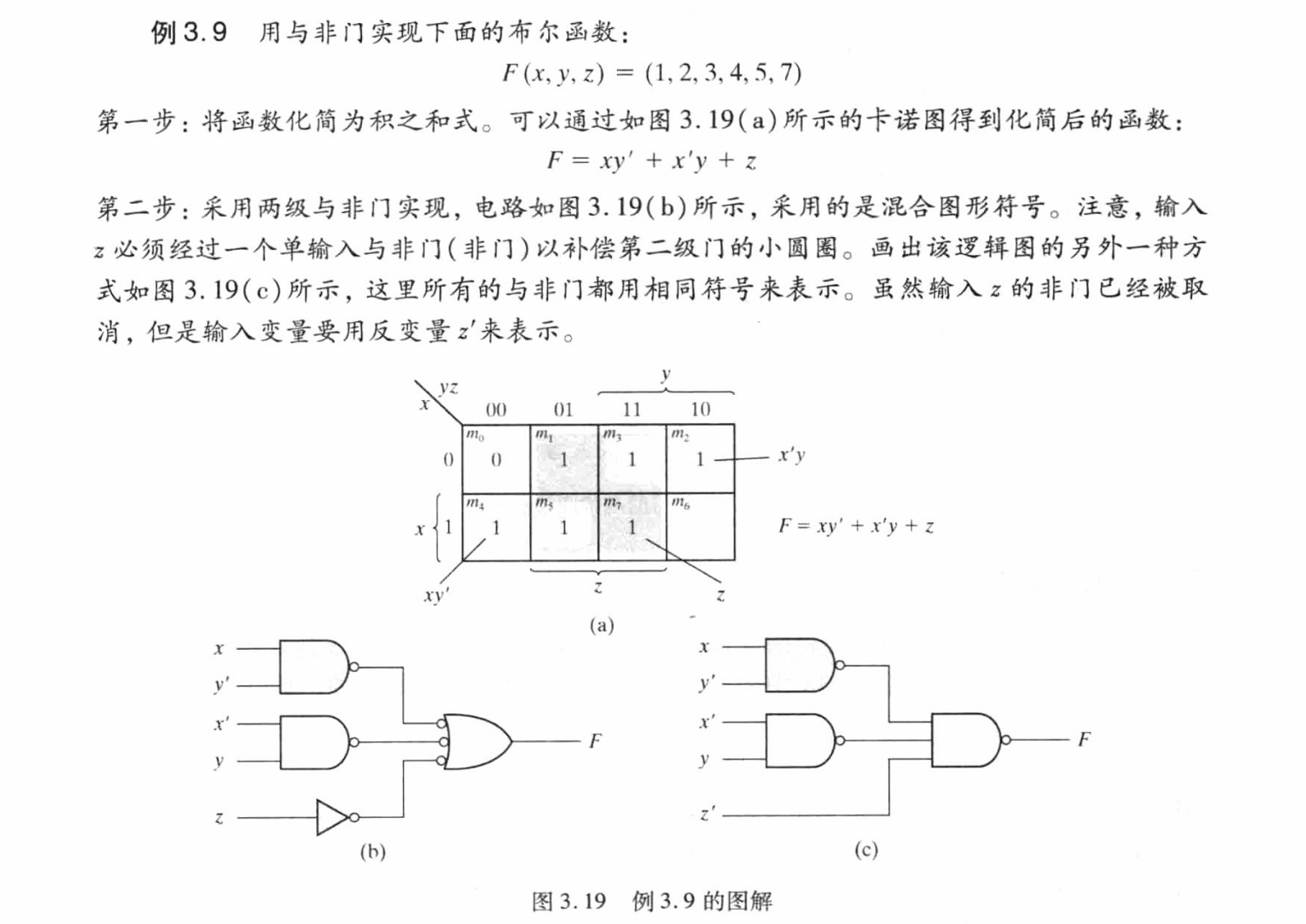


注意用最大项之积表示与POS表示的区别，最大项之积表示只需要写出数字



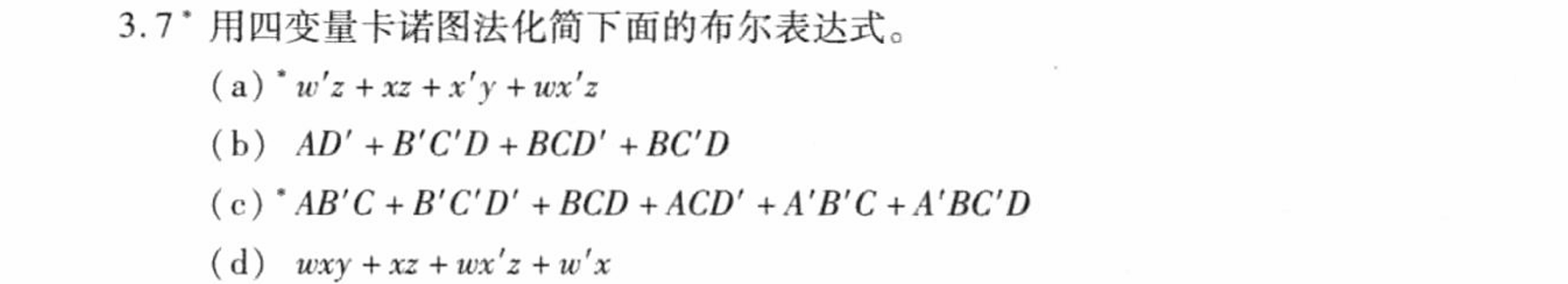


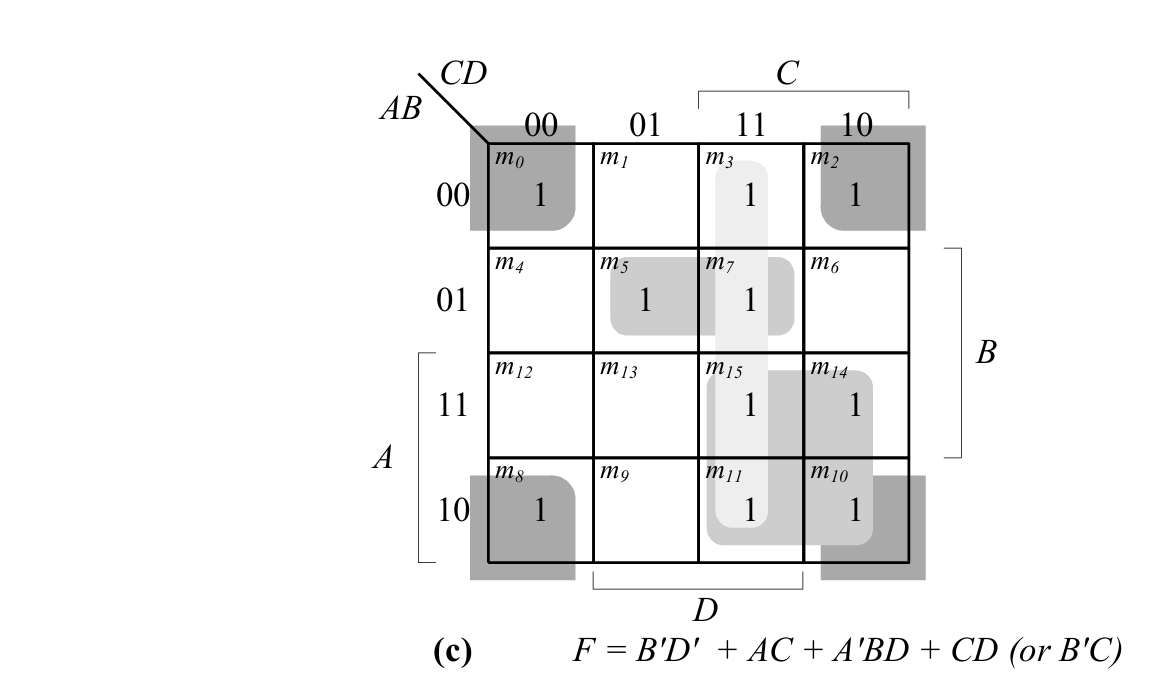
Chapter3



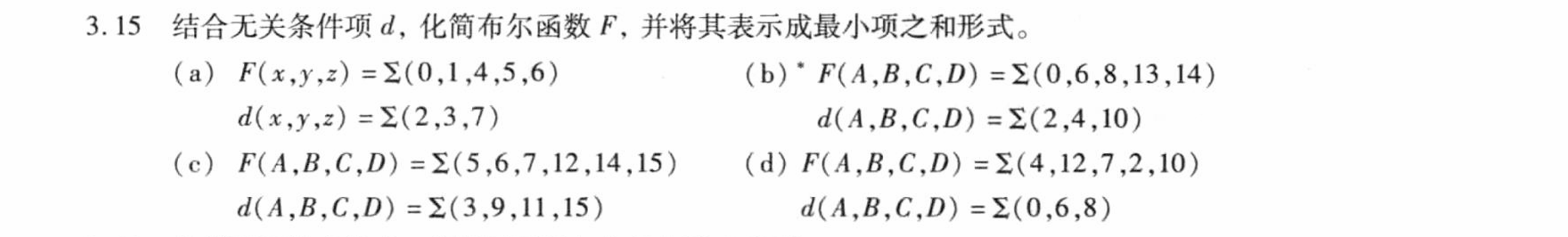
二级逻辑门变换都要从最基础的与-或门开始

NOT-OR=AND-NOT

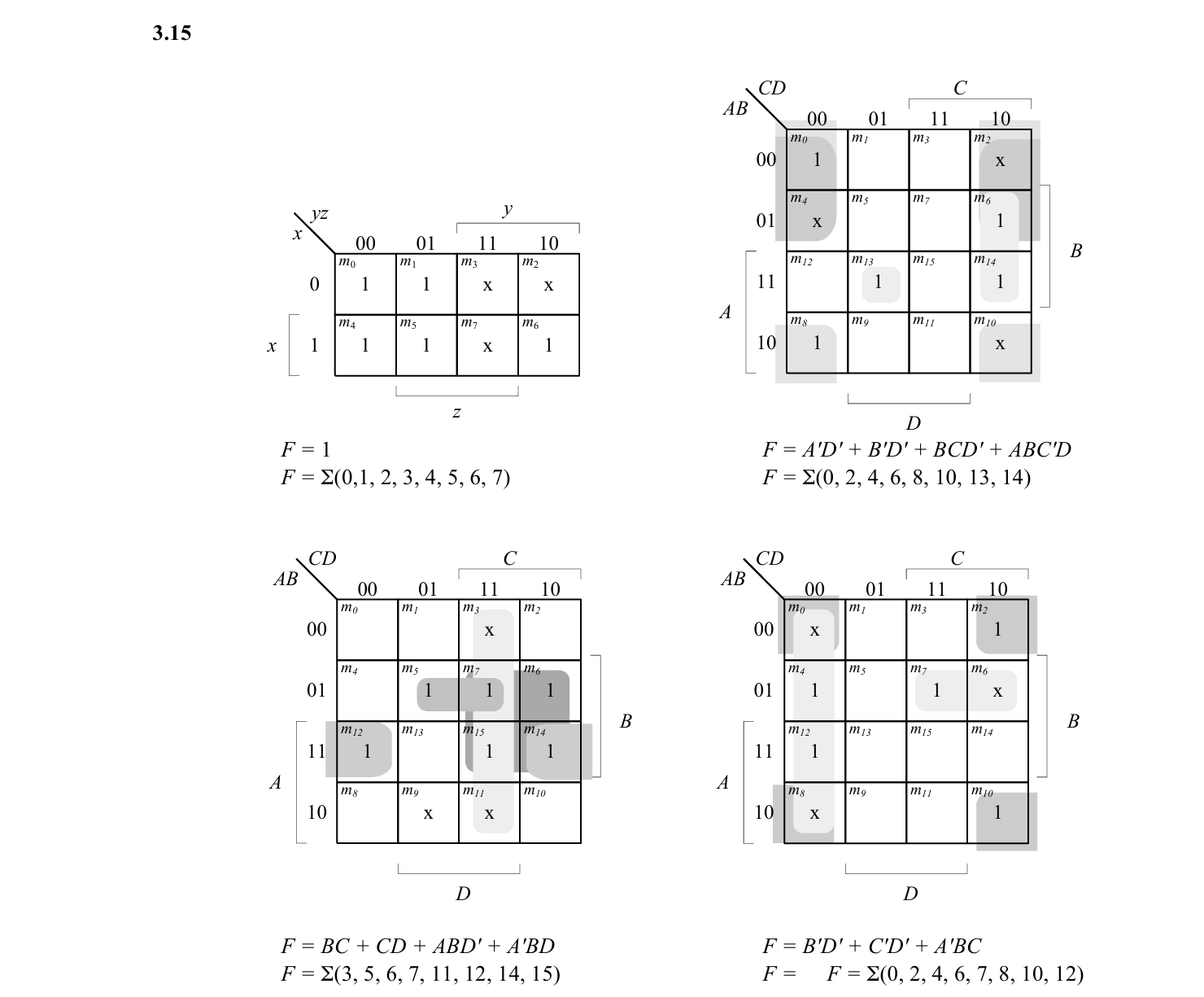




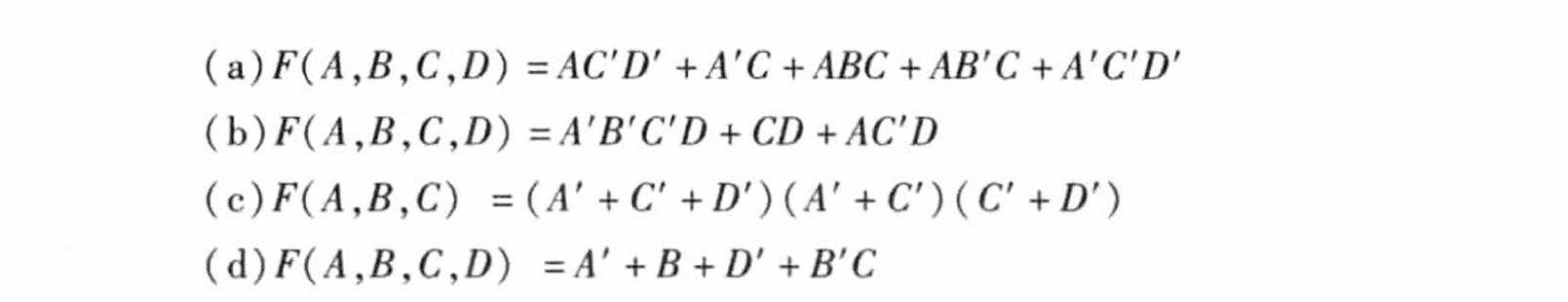
这里的essential只有B’D’, AC, A’BD

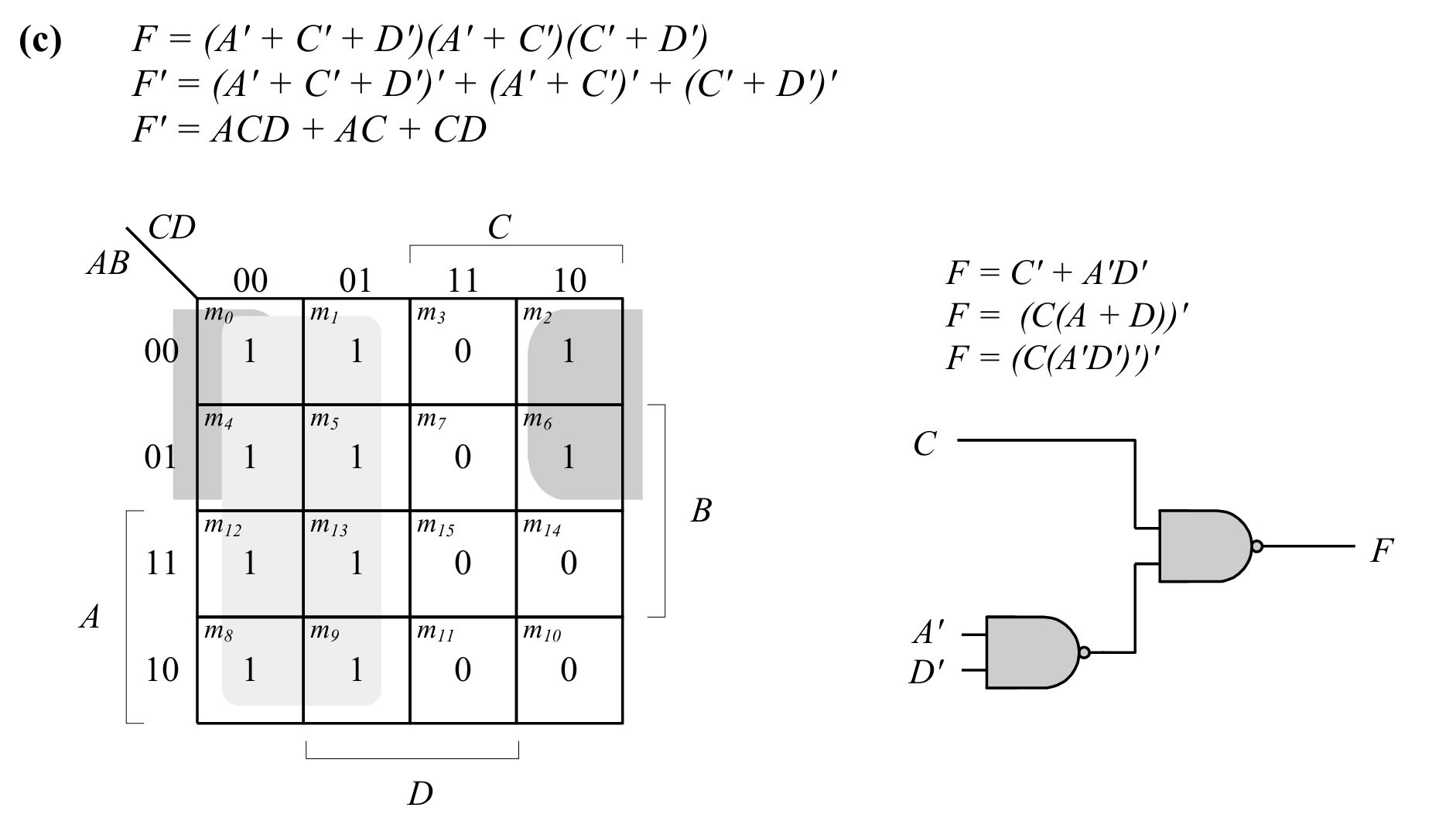


先优化函数F，看哪些d是有需要的，再列出最小项之和（数字表示）



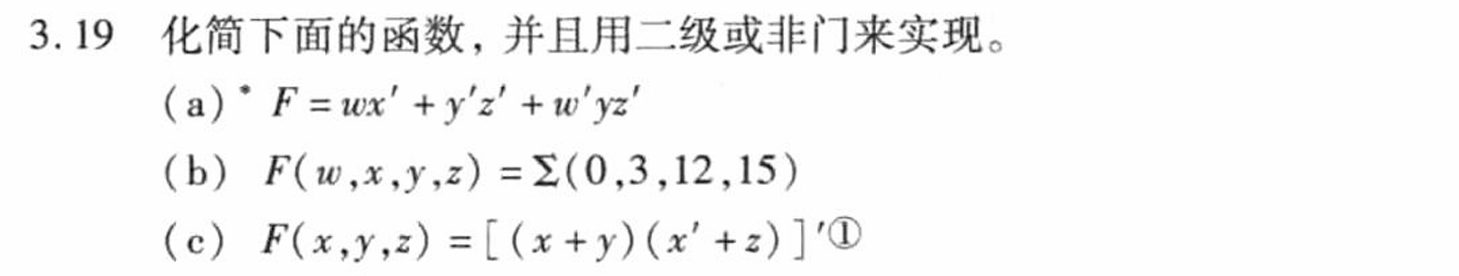


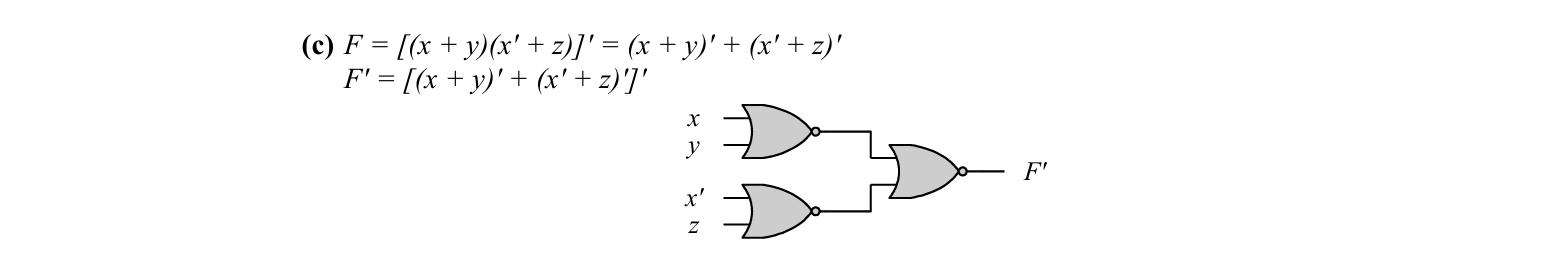




关键步骤在于先列出函数F的SOP形式

与非门表示：先用卡诺图，再画出逻辑门，一级与二级之间二次取反就行

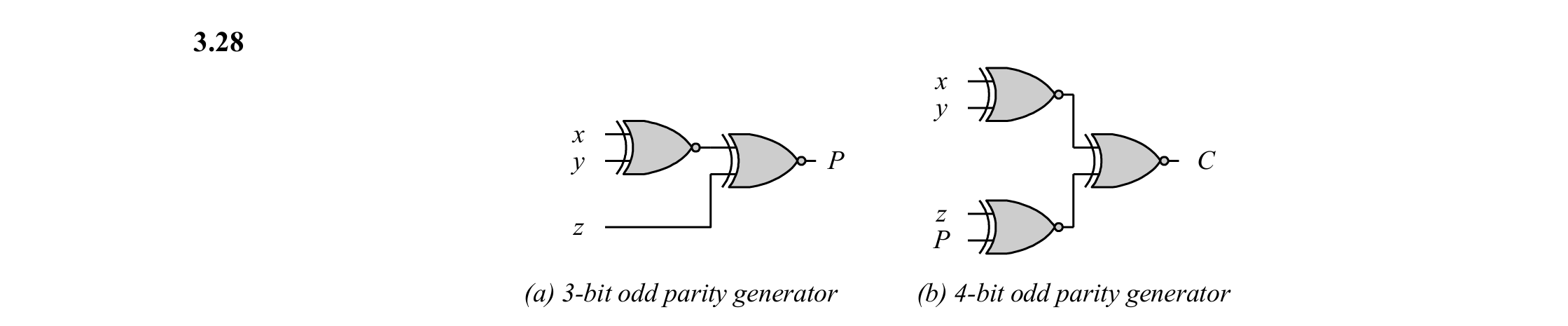




或非门表示的话需要将函数先取反

所有的二级逻辑门电路变换都从最初级的二级逻辑开始





异或函数

两个P是同一个函数

答案有误

左边需要全部改成异或门，最后取反

右边改成全部异或门