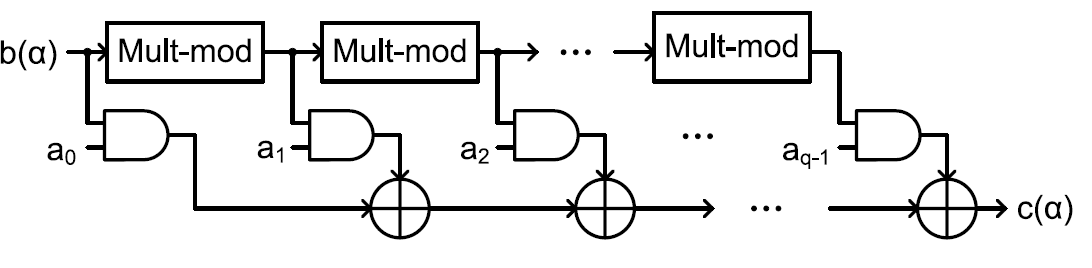
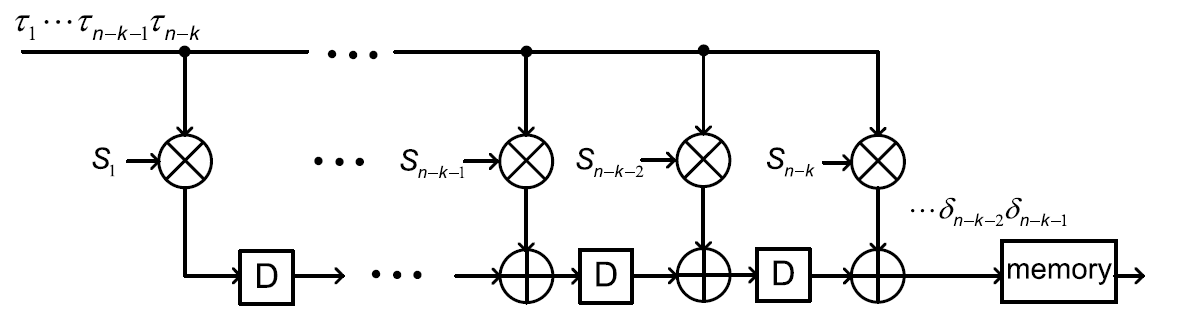
**有限域算法实现时的考虑**

1. Power representation有益于乘法、幂次、求逆的实现。
2. Power representation必须查表转换成polynomial representation才能做加法（做完可能还得再查表转回来）。当有限域的q较大时，这个转换表会很大（2^q个索引，每个索引中、即值域有q个比特）。
3. 转换polynomial representation后，有限域的加法即按位异或运算。
4. 多项式乘法的典型实现方法（c(α) = a(α) · b(α)，用standard basis）：



另附上多项式乘法的一个典型例子（此例子只取读书不小于n-k的项，不影响此乘法结构）：

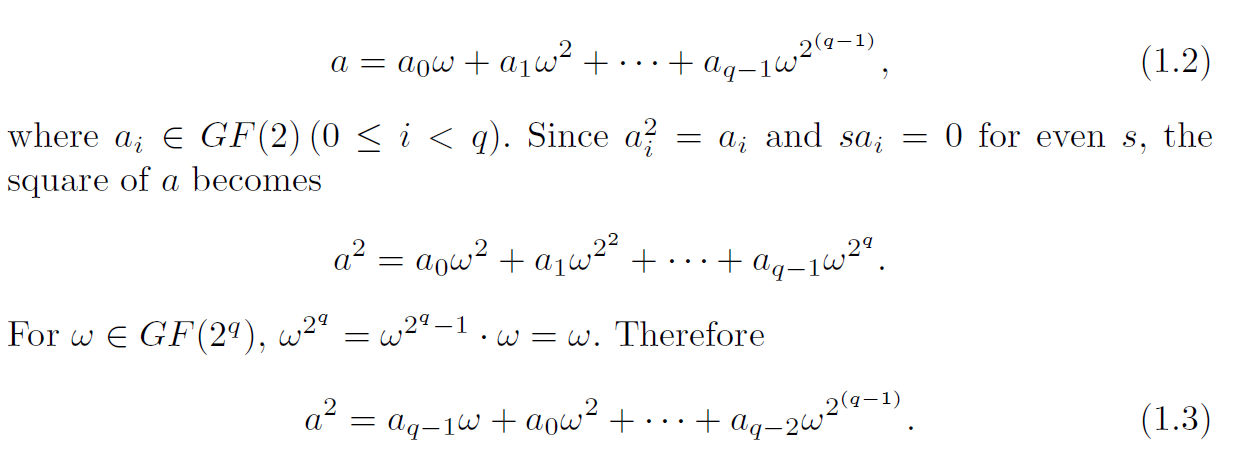




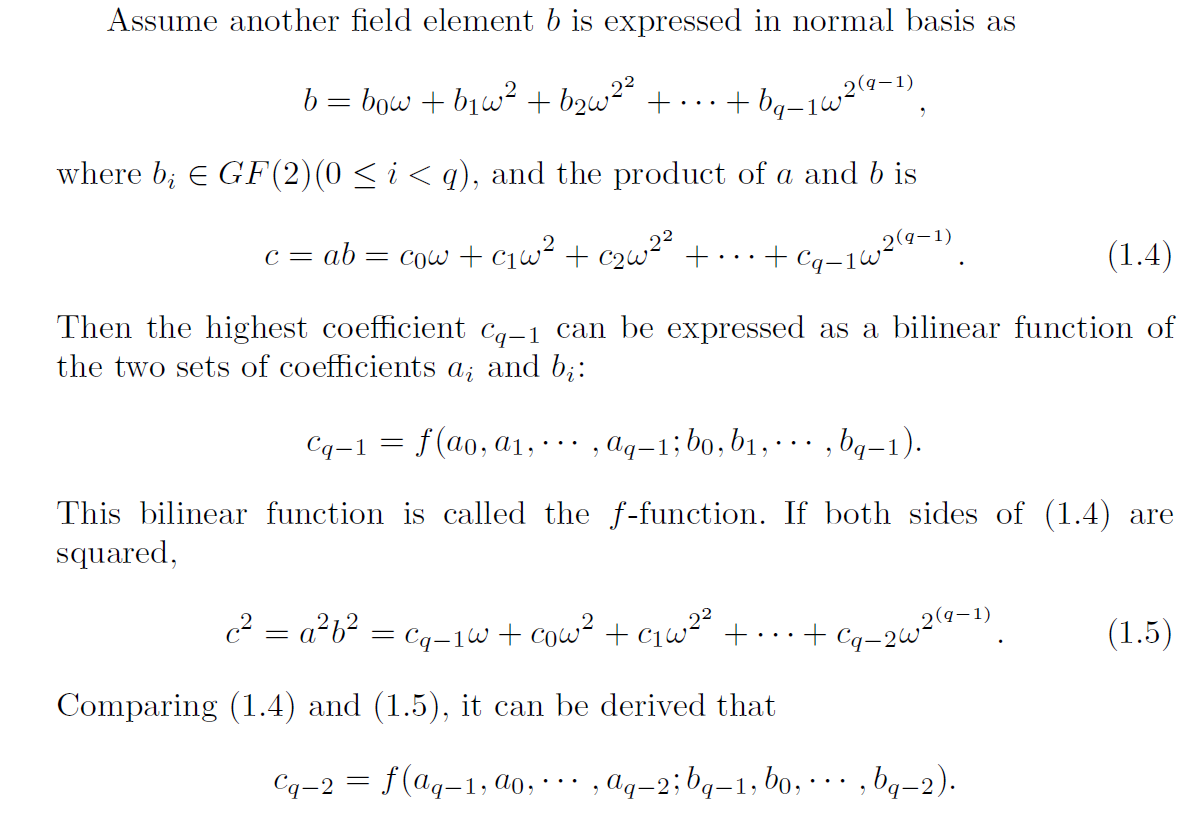
1. Normal basis形式为：



其平方的计算方法为移位运算：

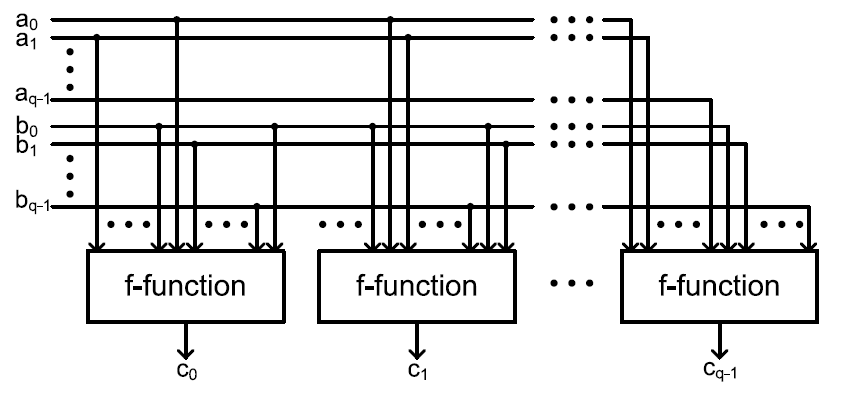


其乘法为



上述f函数为双射函数（一对一、可逆）。

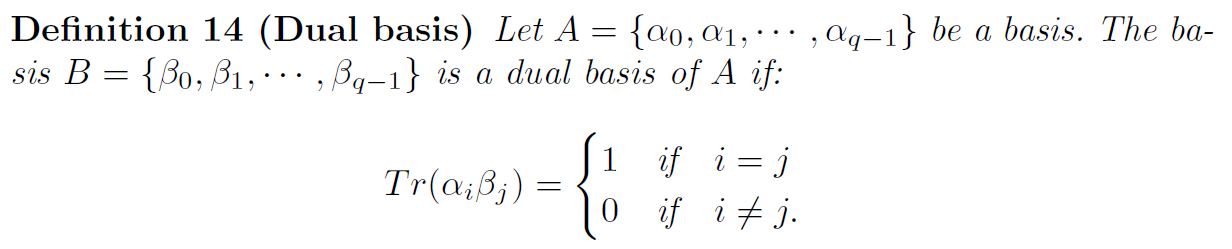
其硬件实现的结构如下，本质上就是f函数的输入不同



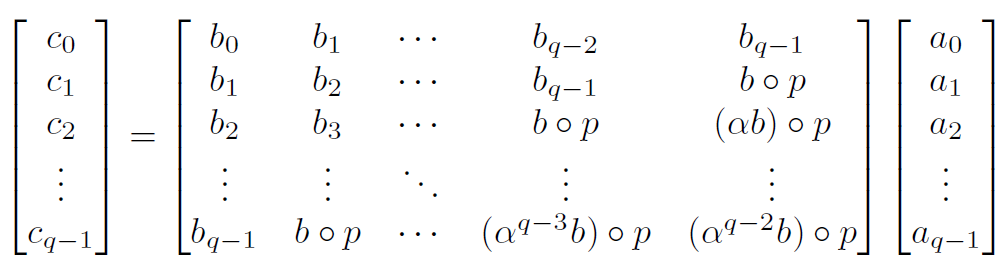
Normal basis一般比standard basis费事，只是算平方比较方便。

简化的normal basis多项式乘法，需要由q^2个与门以及q^2-1个异或门组成，与standdard basis时一样。简化的方法参考M. A. Hasan, M. A. Wang and V. K. Bhargava, “A modified Massey-Omura parallel multiplier for a class of finite fields,” IEEE Trans. On Computers, vol. 42. no. 10, pp. 1278-1280, Oct. 1993。

1. Dual basis的定义如下：



Dual basis用到了trace，其乘法形式如下（⭕代表内积），它有什么好处暂不清楚：



1. 求逆可以通过查表实现，但是这个表的大小是2^q × q，当q不小时，这个规模很可观。
2. 强行求逆的方法（注意，此处α不一定是本原元，也有可能是其幂次）：
3. 由q - 1个平方、q - 2个乘法组成：



1. 由4个乘法器、3个四次幂计算、1个平方组成，q≠8时也可以凑出类似的形式：



1. 这种形式看起来就像完全是凑出来的：

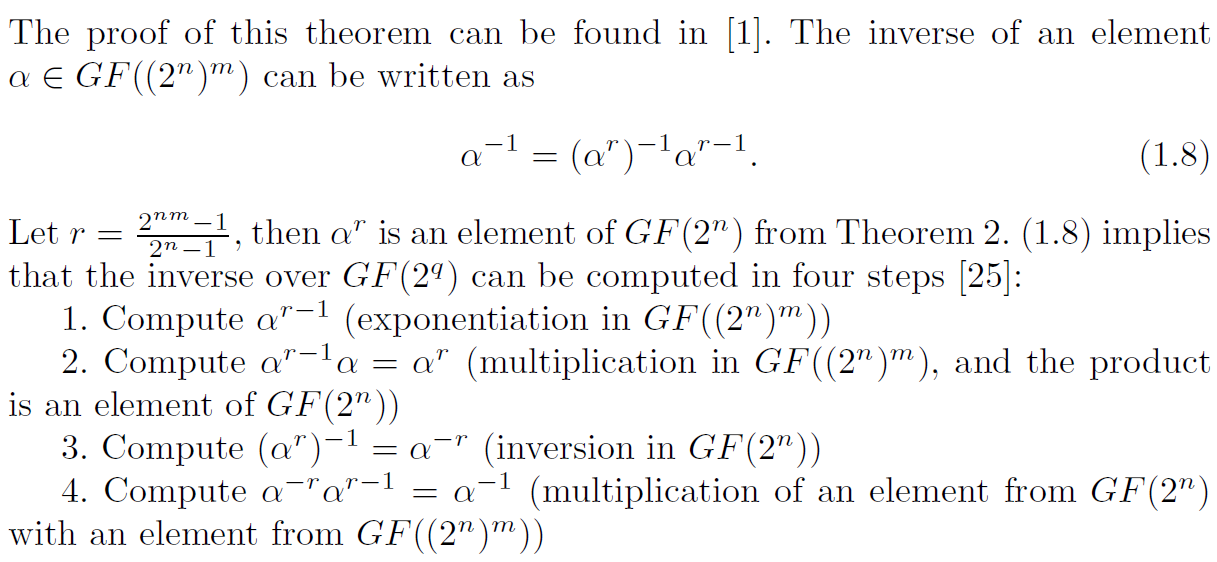


1. 求逆运算还可以在子域做。

有以下定理：



那么，计算α的逆时，可以依据以下步骤：



这时，求逆运算变成了原域和子域上的乘法、以及子域上的求逆，运算得以简化。