分类号：TN492　　学校代码：10363
密　级：公开　　学　　号：2220340120

安徽工程大学
Anhui Polytechnic University

硕士学位论文

题目　基于 GF((2²)⁴)域的 AES S 盒电路优化设计方法研究

论 文 作 者　闫涵
指 导 教 师　张肖强
学科（专业）　控制科学与工程（081100）
研 究 方 向　安全芯片优化设计

论文提交日期：2025 年 05 月 10 日

分类号：TN492　　单位代码：10363
密　级：公开　　学　　号：2220340120

题　　目 基于 $GF((2^2)^4)$域的 AES S 盒电路优化设计方法研究

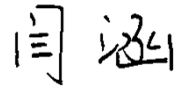
题　　目 RESEARCH ON THE OPTIMIZATION DESIGN METHOD OF AES S-BOX CIRCUIT BASED ON THE $GF((2^2)^4)$ DOMAIN

学 生 姓 名：闫涵
指 导 教 师：张肖强（副教授）
专　　业：控制科学与工程（081100）
研 究 方 向：信息安全芯片

论文答辩日期：2025 年 05 月 10 日

二、安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名： 

日期：2025 年 5 月 10 日

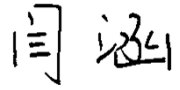
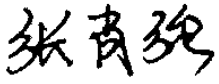
三、安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 □，在 ____ 年解密后适用本版权书。

本学位论文属于

不保密 ☑。

学位论文作者签名：  指导教师签名： 

日期： 2025 年 5 月 10 日　　日期：2025 年 5 月 10 日

基于 $GF((2^2)^4)$域的 AES S 盒电路优化设计方法研究

摘 要

随着现代通信技术和信息技术的快速发展，密码技术有效保障了信息的安全传输与存储。高级加密标准（Advanced Encryption Standard, AES），凭借其较强的抗攻击性以及运算效率，成为了目前应用最广泛的加密标准。本文针对资源受限的应用领域，研究基于复合域 $GF((2^2)^4)$的 AES 密码电路优化设计方法，重点解决了 $GF((2^2)^4)$域乘法逆结构、其运算单元之间公共项消除以及构建最短关键路径等关键问题，并在此基础上利用 Synopsys IC 综合工具对所构建的 AES S 盒电路进行综合评估，主要工作内容如下：

（1）针对目前复合域乘法逆结构单一，电路实现面积和延时较大的问题，根据复合域运算法则，在 $GF(2^8)$域乘法器运算以及不可约多项式的基础上，将 $GF(2^8)$域乘法逆结构域映射到复合域 $GF((2^2)^4)$中获得其乘法逆结构通式；基于复合域 $GF((2^2)^4)$研究其可约与不可约多项式，计算复合域乘法逆通式的最优多项式系数，构建具有最优面积的复合域乘法逆结构。

（2）为消除复合域 $GF((2^2)^4)$乘法逆逻辑单元中的冗余逻辑，基于最短二叉树结构法（Delayed-Driven Binary Tree, DDBT）和延时敏感公共项消除法（Delay Aware CSE, DACSE）对复合域乘法逆结构进行优化，将 $GF(2^2)$域中的常数项$[x, x^2]$作为公共项进行消除时所构建的乘法逆具有最短关键路径和最优资源消耗。

（3）为构建面积和延时最优的 S 盒结构，首先基于复合域 $GF((2^2)^4)$乘法器结构和复合域运算法则，推导计算 $GF(2^8)$不可约多项式的根；其次，在此基础上推导其 8 种映射矩阵和逆映射矩阵架构，利用常数矩阵合并法将逆映射矩阵与仿射矩阵合并；最后，利用公共项消除法消除冗余加法器，其优化率最高达到了 44.9%，最终获得复杂度最低为 25 XOR 的常数矩阵。

（4）采用 Synopsys™ 集成电路 IC 综合工具 Design Compiler，分别基于 SMIC 180nm 和 TSMC 28nm 工艺库对所搭建的 AES S 盒结构优化前后的面积以及延时进

行评估，面积优化率分别达到 61.6%和 62.23%，延时优化率分别达到 37.09%和 17.46%。

关键词：AES，复合域运算，面积优化，公共项消除，S 盒

RESEARCH ON THE OPTIMIZATION DESIGN METHOD OF AES S-BOX CIRCUIT BASED ON THE $GF((2^2)^4)$ DOMAIN

ABSTRACT

With the rapid development of modern communication technology and information technology, cryptography effectively guarantees the secure transmission and storage of information. Advanced Encryption Standard (AES) has become the most widely used encryption standard by virtue of its strong attack resistance and computational efficiency. In this paper, we study the optimal design of AES cryptographic circuits based on the composite domain $GF((2^2)^4)$ for resource-constrained applications, focusing on key issues such as the multiplicative inverse structure of the $GF((2^2)^4)$ domain, the elimination of the common subexpressions between its operation units, the constructed AES S-box circuits are synthesized and evaluated using Synopsys IC synthesis tool the main work is as follows:

(1) Aiming at the current problem of single composite domain multiplicative inverse structure with large area and delay of circuit realization, according to the composite domain algorithm, based on the $GF(2^8)$ domain multiplier operation as well as the incommensurable polynomials, the multiplicative inverse structure domain of the Galois domain $GF(2^8)$ is mapped to the composite domain $GF((2^2)^4)$ to obtain its multiplicative inverse structure general; based on the composite domain $GF((2^2)^4)$, we study Its approximable and non-approximable polynomials are studied based on the composite domain $GF((2^2)^4)$, and the optimal polynomial coefficients of the multiplicative inverse general formula of the composite domain are computed to construct the multiplicative inverse structure of the composite domain with optimal area.

(2) In order to eliminate the redundant logic in the multiplicative inverse logic unit

of the composite domain $GF((2^2)^4)$, the multiplicative inverse structure of the composite domain was optimized based on the Delayed-Driven Binary Tree (DDBT) and the Delay Aware CSE (DACSE) method, and the constant term [x, x^2] in the $GF(2^2)$ domain was optimized. The inverse of the multiplication constructed when eliminated as a common item has the shortest critical path and the optimal resource consumption.

(3) In order to construct the S-box structure with optimal area and delay, firstly, based on the composite domain $GF((2^2)^4)$ multiplier structure and the composite domain algorithm, we derive and compute the roots of the irreducible polynomials of the Galois domain $GF(2^8)$; secondly, based on this, we derive its 8 mapping matrix and inverse mapping matrix architectures, and utilize the constant matrix merging method to merge the inverse mapping matrix with the affine matrix; and lastly, we utilize the common term elimination method to eliminate redundant adders, which has an optimization rate of up to 44.9%, and finally obtains constant matrices with a minimum complexity of 25XOR.

(4)Using Synopsys^{TM} integrated circuit IC synthesis tool Design Compiler, we evaluated the area and delay of the constructed AES S-box structure before and after optimization based on the SMIC 180nm and TSMC 28nm technology libraries. The area optimization rates reached 61.6% and 62.23%, and the delay optimization rates reached 37.09% and 17.46%, respectively.

Keywords: AES, complex domain operations, area optimization, common term elimination, S-box

目 录

插图清单

插表清单

第 1 章 绪论

1.1 研究背景及意义

随着现代通信技术和信息技术的快速发展，信息系统逐渐成为人类社会的基础设施，支撑着人类生活各个方面。特别是随着移动支付、数字版权管理、多媒体通信、智能家居、共享出行等互联网应用的广泛普及，信息的安全保护问题显得更加重要[1]。与此同时，由于信息在存储、传输和处理过程中往往是在开放的通信系统中进行的，易受到窃取、篡改、伪造等安全威胁，使得信息安全问题变得日益严峻。

近些年来相关法律法规的实施，进一步强调了信息安全的关键地位。例如国内 2021 年实施的《数据安全法》和《个人信息保护法》、欧盟所颁布的《通用数据保护条例》（GDPR）等，无一不彰显着各国对信息安全的重视。2025 年，全球范围内的数据泄露事件仍然频繁发生，信息安全已不再仅仅是企业内部的问题，更是关系到国家安全、社会稳定的重要课题。

信息安全问题不仅涉及到金融、军事、政治等国家安全领域，也涉及到个人隐私和私有财产等领域，因此，信息安全是维护国家安全和社会稳定的焦点。信息安全的主要任务是保证信息的保密性、认证性、完整性、不可否认性和可用性。密码保护是实现信息安全的重要手段，而密码保护的核心是密码算法。目前常用的密码算法分为两类：非对称密码（公钥密码）算法和对称密码（私钥密码）算法。

非对称密码算法在加密过程和解密过程中所使用的密钥不相同，且由加密密钥推出解密密钥在计算上不可行[2]。非对称加密算法通常采用两个密钥，分别为公开密钥与私有密钥[3]，因此非对称密钥算法也称为公开密钥算法，主要用于数字签名、身份认证和对称密码的密钥协商等领域。非对称密码算法克服了对称密码算法中密钥传输管理困难、无法提供不可否认功能的缺陷，但由于其计算复杂，加密和解密速度低的缺点，在实际应用中应用范围低，常与对称密码算法结合使用，即在数据传输和存储时采用对称密码算法，而在密钥管理和分发时采用非对称密钥算法。常用的非对称密钥算法有 RSA 算法[4-5]、ElGamal 算法[6-7]、Diffie-Hellman 算法[8-9]、

椭圆曲线密码学（ECC）[10-11]等[12-13]。

对称密码算法要求加/解密双方的密钥必须相同，且是完全保密的。对称密码体制的优点在于加/解密速度快，易于硬件实现，因此主要应用于数据存储和数据传输等领域。缺点是密钥的传输和管理难度大，可实现性低。典型的对称密码算法有数据加密标准（Data Encryption Standard, DES）[14-15]、3DES[16-17]、高级加密标准（Advanced Encryption Standard, AES）[18-19]、Camellia[20-21]、SM3[22-23]、SM4[24-25]等。高级加密标准 AES 是美国国家标准与技术研究院（National Institute of Standards and Technology, NIST）于 2001 年发布了新一代分组密码算法标准[26]，用来取代安全系数较低的 DES 密码算法。AES 算法具有计算简便、加密效率高、安全性强的特点，成为了目前国际上使用最广泛的分组密码算法[27]。表 1-1 对常用加密算法进行了比较。

表 1-1 加密算法比较

算法类型	加密算法名称	运算速度	安全性	资源消耗
对称加密	DES	较快	低	中
	3DES	慢	中	高
	AES	快	高	低
非对称加密	RSA	较快	高	高

密码算法有硬件和软件两种实现方式。软件实现加密技术具有灵活性高、可扩展性强和可移植性好等优点，但是其加密运行速度较低，并且由于软件运行环境的开放性，密码算法和加密信息容易遭到篡改和窃取。与软件加密技术相比，硬件加密技术具有更高的物理安全性和加密速度，应用前景更广阔。近年来随着物联网技术的不断发展，AES 硬件电路在无线传感网[28-29]和射频识别技术（Radio Frequency Identification, RFID）[30-31]等资源有限的应用领域中得到广泛的应用。然而在这种低成本、低功耗、资源受限的硬件平台上如何实现 AES 密码算法，给电路设计带来新的挑战。因此，针对资源受限的应用领域，研究 AES 密码电路优化设计方法，提高加密电路实现效率，具有重要理论意义和应用价值。

1.2 高级加密标准 AES 国内外研究现状

AES 算法作为目前主流的对称算法，有着较高的安全系数和良好的应用空间，

其密钥长度可为 128、192 和 256 位，处理数据位宽为 128 位[32]。相比 DES 和 3DES 加密算法，AES 算法硬件实现所需资源较少并且安全系数更高，所以广泛应用在加密数据处理场景中。AES 算法基于替换-置换（SP）结构设计，每轮包括字节替换（SubBytes）、行移位（ShiftRows）、列混合（MixColumns）和密钥加（AndroundKey）四个步骤[33]。其中字节替换（SubBytes）操作使用 S 盒，是该算法主要的非线性替换步骤，在软件实现中采用查表完成，快速便捷。但是对于硬件实现来说，S 盒查表法有一个缺点：表的每个副本需要 256 字节的存储，并且需要单独的寻址表和获取结果的电路[34]。S 盒是 AES 算法中唯一的非线性部件[35]，也是硬件实现中最为复杂度最高的部分，其占用资源相对较多。AES 轮变换中除了 S 盒运算之外，还包括行移位、列混合和密钥加等运算。这四个运算中，行移位运算不需要任何逻辑资源，密钥加运算只需要一级 XOR 逻辑，两者没有进一步优化的空间。因此，AES 轮变换优化设计的研究主要集中在 S 盒运算和列混合运算的优化设计上。

复合域 S 盒电路的实现关键在于复合域乘法逆电路的实现。基于复合域运算的乘法逆电路不仅具有面积小的优点，还可以通过流水线技术提高电路运行速度[36-37]，或通过路径平衡技术降低电路功耗，因此基于复合域运算的乘法逆电路实现是目前分组密码电路优化设计方面的一个研究热点。2000 年 Rijmen[38]首次将复合域的概念引入 AES 算法，将 S 盒中 $GF(2^8)$域乘法逆结构映射到复合域中运算，这大大降低了 AES S 盒的电路面积。2005 年 Canright[39]提出了将 $GF(2^8)$域乘法逆结构映射至复合域 $GF((2^4)^2)$来实现 S 盒结构，通过两级子域 $GF(2^4)$和 $GF((2^4)^2)$的嵌套运算，减少了乘法逆结构的资源占用；Zhang Xinmiao[40]进一步将子域 $GF(2^4)$映射到复合域 $GF((2^2)^2)$，即将 $GF(2^8)$域乘法逆结构映射至复合域 $GF(((2^2)^2)^2)$，构建了三级子域：$GF(2^2)$、$GF((2^2)^2)$和 $GF(((2^2)^2)^2)$，进一步优化了乘法逆结构的复杂度。此后，关于 AES S 盒的研究都是基于复合域 $GF((2^4)^2)$和复合域 $GF((2^2)^2)$。例如，D. Kamel[41]、Q LuoD[42]、Ashmawy[43]、戴强[44]等人基于复合域 $GF((2^4)^2)$，分别优化了 $GF(2^8)$域乘法逆运算和量子电路的复杂度；M M Wong[45]、JW Kim[46]、X Zhang[47]、S. Morioka[48]等人基于复合域 $GF(((2^2)^2)^2)$，优化算法的复杂度。根据复合域的同构映射法，$GF(2^8)$域有如图 1-1 所示的 3 种映射结构，但是将 $GF(2^8)$域映射至复合域

GF((2²)⁴)的研究至今仍是一片空白，本文将研究基于复合域 *GF*((2²)⁴)的乘法逆结构，弥补这一空白。

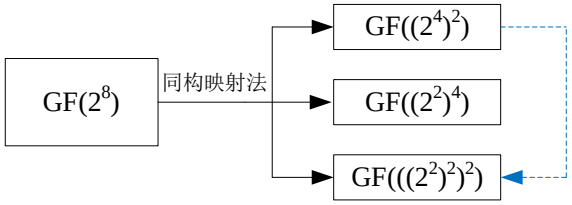


图 1-1 *GF*(2⁸)域同构映射结构

复合域 S 盒电路的实现关键在于复合域乘法逆电路的实现，其硬件复杂度与复合域中每一级子域的基和不可约多项式相关。文献[49-50]讨论了基于多项式基的复合域乘法逆结构特点，文献[51-52]则讨论了正规基下的复合域乘法逆结构特点，相比正规基乘法逆结构，多项式基乘法逆结构具有更短的关键路径，而正规基乘法逆具有更小的电路面积。文献[53]讨论了纵向混合基（即在不同的子域采用不同的基）的复合域乘法逆结构特点，基于纵向混合基的乘法逆结构具有更好的面积-延时折衷效果。文献[54]则讨论了横向混合基（即运算单元的输入输出采用不同的基）的复合域乘法逆结构特点，即横向混合基具有更短的关键路径。现有的关于复合域乘法逆的研究工作主要集中在标准基和正规基的复合域结构上，本文采用具有更短的关键路径的多项式基搭建复合域 *GF*((2²)⁴)的乘法逆结构。

具有相同组合逻辑的运算单元间可以通过提取相同逻辑部件的方式来减少电路冗余逻辑，从而减少电路实现面积。文献[57-58]使用公共项消除法（Common Subexpressions Elimination, CSE）来提取 AES 算法中的公共项表达式，优化算法计算复杂度，减少密码电路的面积占用。CSE 算法可能会带来增加算法最短路径的可能性，文献[59]证明，“延时驱动二叉树”（Delay-Drive-Binary-Tree, DDBT）结构具有最短的关键路径。因此 2015 年 Zhang Xiaoqiang[60]提出了延时敏感公共项消除法（Delay Aware CSE, DACSE）[61]，将 CSE 算法和 DDBT[62]相结合，在不减少最短路径的情况下，最大限度的减少电路面积。此外，对于 AES S 盒中的常数矩阵即映射矩阵、逆映射矩阵盒仿射矩阵等常数矩阵，文献[63-64]采用将多个常数矩阵合并为一个矩阵，从而降低算法的运算复杂度，即常数矩阵合并法。本文采用 DACSE

算法对基于复合域 $GF((2^2)^4)$的乘法逆结构进行优化，并采用常数矩阵合并法对 AES S 盒中的常数矩阵进行合并。

1.3 本文主要研究内容

本文研究了基于复合域 $GF((2^2)^4)$的 AES S 盒，并对其进行优化，首先推导了基于复合域 $GF((2^2)^4)$的乘法逆结构；其次，使用 DACSE 算法对复合域乘法逆结构进行优化，减少其逻辑运算复杂度；然后，搭建了基于复合域 $GF((2^2)^4)$的 AES S 盒结构，并使用常数矩阵合并法和 CSE 算法对常数矩阵进行优化；最后采用 Synopsys™ 集成电路 IC 综合工具 Design Complier（DC）对 AES S 盒结构进行电路综合评估。本文的研究内容如下：

（1）搭建基于复合域 $GF((2^2)^4)$的乘法逆结构。根据复合域运算法则以及同构映射法，将 $GF(2^8)$域乘法逆结构映射至复合域 $GF((2^2)^4)$，推导乘法逆结构通式。为了获得逻辑资源占用更优的乘法逆结构，将复合域 $GF((2^2)^4)$的乘法器结构与 $GF(2^8)$域不可约多项式相结合，研究分析基于复合域 $GF((2^2)^4)$的可约多项式、不可约多项式以及本原多项式，从而搭建系数最优下的复合域 $GF((2^2)^4)$乘法逆结构。

（2）优化基于复合域 $GF((2^2)^4)$的乘法逆结构。由于 $GF((2^2)^4)$域乘法逆结构存在较多的重复逻辑运算，本文采用 DACSE 和 DDBT 算法对乘法逆结构进行优化，在不增加最短路径的同时，最大限度获取面积最优的乘法逆结构。

（3）搭建并优化基于复合域 $GF((2^2)^4)$的 AES S 盒。基于复合域 $GF((2^2)^4)$系数最优的乘法逆结构，推导计算映射矩阵与逆映射矩阵，通过常数矩阵合并法将多个常数矩阵合并为一个，再采用 CSE 算法减少常数矩阵间的冗余逻辑资源。最终，基于结构最优的复合域 $GF((2^2)^4)$乘法逆结构和常数矩阵，构建 AES S 盒。

（4）AES S 盒电路综合评估。采用 Synopsys™ 集成电路 IC 综合工具 DC，分别基于 SMIC 180nm 和 TSMC 28nm 工艺库对所搭建的 AES S 盒结构的面积以及延时进行评估。

1.4 论文章节安排

本文共分为 6 个章节，各章主要内容如下，论文研究内容及结构安排如图 1-2 所示。

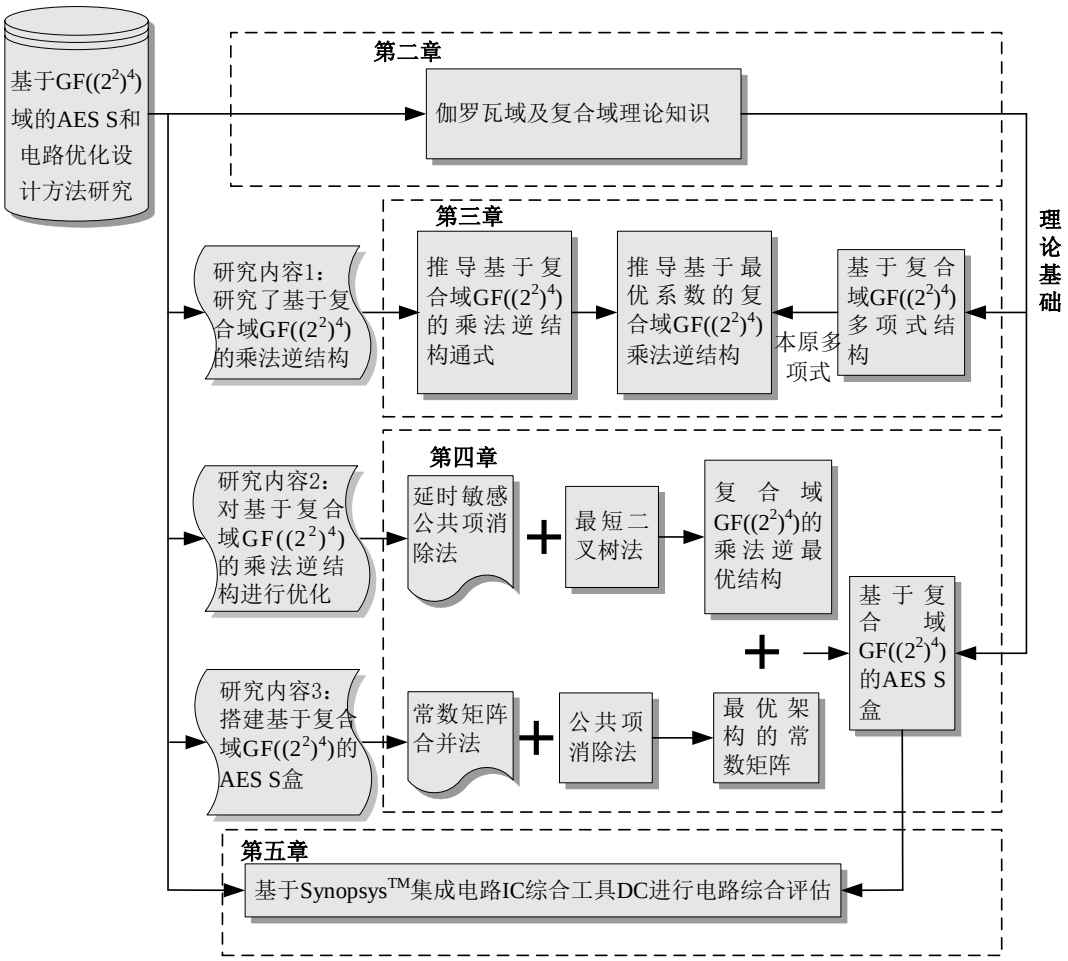


图 1-2 研究内容及结构安排

第一章节是绪论。首先对本章的研究内容进行背景介绍，论述信息安全的重要性，分析目前主流的密码算法；其次分析目前关于基于复合域的乘法逆结构和针对 AES S 盒优化方法的研究现状；最后介绍本文的研究内容和章节安排。

第二章节是理论基础及实验平台。首先对本研究内容的理论基础进行介绍；然后针对于伽罗瓦域的构成和运算法则进行阐述，为本文的算法推导奠定了理论基础。

第三章节是构建基于复合域 $GF((2^2)^4)$乘法逆结构。首先在搭建了基于复合域 $GF((2^2)^4)$乘法逆结构通式和乘法器结构；其次研究分析复合域 $GF((2^2)^4)$的可约多项

式、不可约多项式与本原多项式的不同，利用穷搜索法对复合域 $GF((2^2)^4)$中的多项式进行分类；最后，在不同本原多项式的基础上，查找计算复合域 $GF((2^2)^4)$乘法逆结构复杂度，复杂度最低的一组乘法逆结构，称作最优系数下的复合域 $GF((2^2)^4)$乘法逆结构。

第四章节是构建基于复合域 $GF((2^2)^4)$的 AES S 盒及其优化。首先基于 DDBT 和 DACSE 算法对最优系数下的复合域乘法逆结构进行优化；其次，介绍 AES S 盒的搭建流程；然后，推导计算 AES S 盒的常数矩阵，即映射矩阵及逆映射矩阵，基于常数矩阵合并法和 CSE 算法进行优化。最后，搭建了结构最优的 AES S 盒。

第五章节是 AES S 盒实现与电路综合评估。首先对 AES S 盒的实现流程进行了描述，并推算了元素从 $GF(2^8)$域到复合域 $GF((2^2)^4)$的映射关系；然后，采用 Synopsys™ 集成电路 IC 综合工具 DC，分别基于 SMIC 180nm 和 TSMC 28nm 工艺库对复合域乘法逆结构优化后与直接实现复合域 $GF((2^2)^4)$ S 盒结构的面积和延时综合评估对比。

第六章是全文总结与未来展望。

第 2 章 理论基础

本章首先介绍了伽罗瓦域的理论基础，阐述了伽罗瓦域、本原多项式、本原元素、迹函数等理论的定义；其次，详细的介绍了伽罗瓦域的三种构成方法，即本原元素表示法、多项式表示法和向量基表示法；最后，介绍了伽罗瓦域加法运算法则和乘法运算法则的实现。

2.1 理论基础

本论文的研究主要集中于伽罗瓦域（Galois Field,*GF*）在乘法逆结构中的应用。伽罗瓦域广泛应用于密码学中的各种算法，尤其是在对称加密、密钥生成、伪随机数生成以及数字签名等方面。在硬件实现密码电路时，伽罗瓦域的代数结构为高效运算提供了理论基础，使得这些密码学算法能够在有限的资源和严格的延时要求下得到高效实现。

2.1.1 域的定义

假设 G 是一个集合，在 G 上定义了两个称作加法$\oplus$和乘法$\otimes$的两个运算，并且该运算满足以下性质，则称$(G,\oplus,\otimes)$构成一个域（Field）[65]。

（1）封闭性：对于任意$a,b\in G$，有$\begin{cases}a\oplus b\in G\\ a\otimes b\in G\end{cases}$；

（2）交换性：对于任意$a,b\in G$，有$\begin{cases}a\oplus b=b\oplus a\\ a\otimes b=b\otimes a\end{cases}$；

（3）结合律：对于任意$a,b,c\in G$，有$\begin{cases}(a\oplus b)\oplus c=a\oplus(b\oplus c)\\ (a\otimes b)\otimes c=a\otimes(b\otimes c)\end{cases}$；

（4）分配律：对于任意$a,b,c\in G$，有$a\otimes(b\oplus c)=(a\otimes b)\oplus(a\otimes c)$；

（5）单位元存在：对任意的$a\in G$，有$\begin{cases}a\oplus 0=a\\ a\otimes 1=a\end{cases}$，则称 0 为加法单位元，1 为加法单位元；

（6）加法逆元存在：对于任意元素 $a \in G$，都有加法逆元 a^{-1}，使得 $a \oplus a^{-1} = 0$；

（7）乘法逆元存在：对于任意非零元素 $a \in G$，都有乘法逆元 a^{-1}，使得 $a \otimes a^{-1} = 1$。

对于任意非零元素 $a,b \in G$，则必有 $ab \neq 0$，若 $ab = 0$，则称 G 中零因子，此时集合 G 不构成域[66]。

2.1.2 伽罗瓦域的定义

给定一个域G，其元素个数为m，当且仅当 $m = p^n$（其中p为素数，n为正整数）时，这个域可以称作有限域（Finite Field）或伽罗瓦域（Galois Field），记为$GF(p^n)$，在本文中，有限域和伽罗瓦域可以互换使用。其中，p^n称为阶数，即此伽罗瓦域中元素的个数为p^n，素数p叫做域的特征。例如，12个元素组成的集合并不能构成一个伽罗瓦域，因为12不能表示为一个素数的幂；而8个元素组成的集合可以构成一个伽罗瓦域，即 $8 = 2^3$，通常记作$GF(2^3)$[39-40,66]。

假设 $GF(p^n)$域中任一元素 a，而 $k = p^n - 1$，则 k 是使得 a 满足 $a^k = 1$的最小正整数，称 a^k 是 $GF(p^n)$单位元。例如，在 $GF(2^3)$域中，$k = 2^3 - 1 = 7$，则对于 $GF(2^3)$域中任一元素 a，有 $a^7 = 1$。

2.1.3 伽罗瓦域上的多项式

在 $GF(p^n)$域中，多项式 $f(x)$可以表示为：

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_1 x + a_0 \tag{2-1}$$

其中，系数 $\{a_n, a_{n-1}, a_{n-2}, \cdots, a_1, a_0\} \in GF(p^n)$。

多项式 $f(x)$ 以是否可以分解为多个多项式的乘积为标准，分为可约多项式和不可约多项式。如果 $GF(p^n)$域上的一个多项式 $f_1(x)$可以被分解为 $GF(p^n)$域中除 $f_1(x)$本身以外的多项式乘积，则称多项式 $f_1(x)$为 $GF(p^n)$域的可约多项式，形如：$f_1(x) = f_2(x) \times f_3(x)$，$f_2(x)$ 和 $f_3(x)$是 $GF(p^n)$域的多项式；不可分解为除 $f_1(x)$本身以外的多项式乘积，则称多项式 $f_1(x)$为 $GF(p^n)$域的不可约多项式。

例如，在 $GF(2^2)$域中，多项式 $f_1(x) = x^2 + 1$，而 $x^2 + 1 = x^2 + x + x + 1 = (x+1)^2$，因此多项式 $f_1(x)$是可约多项式。

$GF(p^n)$域中的 n 次不可约多项式的个数为：

$$N_p(n)=\frac{1}{n}\sum_{d|n}\mu(d)\,p^{\frac{n}{d}} \tag{2-2}$$

其中，$\mu(n)$ 为莫比乌斯函数，$d|n$ 表示 d 是 n 的因子。

例如，对于 $GF(2^4)$而言，取 d 为 n=4 的因子，即 1，2，4；

（1） 当 d=1 时，$\mu(1)=1$，$2^{\frac{4}{1}}=2^4=16$，则 $\mu(1)\cdot 2^4=16$；

（2） 当 d=2 时，$\mu(2)=-1$，$2^{\frac{4}{2}}=2^2=4$，则 $\mu(2)\cdot 2^2=-4$；

（3） 当 d=4 时，$\mu(4)=0$，$2^{\frac{4}{4}}=2=2$，则 $\mu(4)\cdot 2=0$；

则 $N(4)=\frac{1}{4}(16-4+0)=3$。

当 p=2 时，$GF(2)$域有 1 个不可约多项式，$GF(2^2)$域有 1 个不可约多项式，$GF(2^4)$域有 3 个不可约多项式，$GF(2^8)$域有 30 个不可约多项式。表 2-1 给出了伽罗瓦域常见的不可约多项式。

表 2-1 常见的不可约多项式

伽罗瓦域	不可约多项式
$GF(2)$	$x+1$
$GF(2^2)$	x^2+x+1
$GF(2^4)$	x^4+x+1
	x^4+x^3+1
	$x^4+x^3+x^2+x+1$
$GF(2^8)$	$x^8+x^4+x^3+x^2+1$
	$x^8+x^6+x^5+x^2+1$
	$x^8+x^7+x^2+x+1$
	$x^8+x^4+x^3+x^2+x+1$
	$x^8+x^6+x^5+x+1$
	$x^8+x^7+x^5+x^3+1$

2.1.4 本原元素

若 $GF(p^n)$域中有一非零元素 a 能够生成域中其余非零元素，即：$a^0\neq a^1\neq a^2\neq\cdots\neq a^{p^n-2}$，则称此非零元素 a 为本原元素。对于本原元素而言 $a^{p^n-1}=1$，$a^{p^n}=a$，每个伽罗瓦域都有本原元素[39-40,66]。

例如，在$GF(2^2)$域中，其域中元素有$\{0,1,\Omega,\psi\}$，不可约多项式为$p(x)$的表达式为$p(x)=x^2+x+1$，且$x^2+x+1=0$，则$1=\Omega^0$，$\Omega=\Omega$，$\psi=\Omega^2$，即Ω可以表示$GF(2^2)$域中所有非零元素，称Ω为$GF(2^2)$域的本原元素。

2.1.5 本原多项式

本原多项式是一种特殊的不可约多项式，本原多项式的根是本原元素，或者说如果一个多项式的根是该有限域中的本原元素，那么这个多项式就被称为本原多项式。本原多项式一定是不可约多项式，但是不可约多项式不一定是本原多项式。

判断一个多项式是否为本原多项式有两种方法。

方法一：若$GF(p^n)$域上的一个多项式$f(x)$可以整除x^s-1（$s=p^n-1$），则$f(x)$称为$GF(p^n)$域的本原多项式。例如，$GF(2^4)$域有三个不可约多项式，其中$\{x^4+x^3+1,x^4+x+1\}$可以整除$x^{15}-1$，因此$\{x^4+x^3+1,x^4+x+1\}$是本原多项式，而$x^4+x^3+x^2+x+1$不能整除$x^{15}-1$，因此不是本原多项式。

方法二：判断$GF(p^n)$域上不可约多项式$f(x)$的根a的阶数是否为p^n-1，如果是，则这个多项式为本原多项式，否则不是本原多项式。例如，在$GF(2^4)$域中，有不可约多项式$f_1(x)=x^4+x+1$，且$f_1(x^{15})=0$。

根据方法二可以得到一个结论，$GF(p^n)$域上的n次不可约多项式的根是$GF(p^n)$域的单位元，即a^k（$k=p^n-1$）。表 2-2 给出了伽罗瓦域常见的本原多项式。

表 2-2 常见的本原多项式

伽罗瓦域	本原多项式
$GF(2)$	$x+1$
$GF(2^2)$	x^2+x+1
$GF(2^4)$	x^4+x+1
	x^4+x^3+1
	$x^4+x^3+x^2+x+1$
$GF(2^8)$	$x^8+x^4+x^3+x^2+1$
	$x^8+x^6+x^5+x+1$
	$x^8+x^7+x^5+x^3+1$

2.1.6 复合域的定义

复合域可以理解为由一个较为简单的伽罗瓦域的嵌套使用构造为一个较为复杂

的伽罗瓦域。对于 $GF(p^q)$域而言，其中 $q=mn$，则伽罗瓦域可以同构映射于复合域 $GF((p^n)^m)$和复合域 $GF((p^m)^n)$。以复合域 $GF((p^n)^m)$为例，假设其不可约多项式表示为 $f(x)=a_m x^m+a_{m-1}x^{m-1}+\cdots+a_1x+a_0$，系数$\{a_m,a_{m-1},\ldots,a_1,a_0\}$属于 $GF(p^n)$域；复合域元素表示为$c_0x^0+c_1x^1+\cdots+c_{n-1}x^{m-1}$，系数$\{c_0,c_1,\ldots,c_{n-1}\}$属于 $GF(p^n)$域。因此 $GF(2^4)$域可以同构映射于复合域 $GF((2^2)^2)$，$GF(2^8)$域可以同构映射于复合域 $GF((2^2)^4)$，$GF((2^4)^2)$和 $GF(((2^2)^2)^2)$[39-40]。

例如，对于 $GF(5^6)$域，6=2×3，所以 $GF(5^6)$域可以同构映射于复合域 $GF((5^2)^3)$和 $GF((5^3)^2)$，以复合域 $GF((5^2)^3)$为例，其元素表示为$c_2x^2+c_1x+c_0x$，$\{c_2,c_1,c_0\}\in GF(5^2)$域，选择复合域的不可约多项式 $f(x)=x^3+x^2+x+1$，则复合域 $GF((5^2)^3)$的两个元素$\{a,b\}$的运算可表示为$a\otimes b=(a_2x^2+a_1x+a_0x)\times(a_2x^2+a_1x+a_0x)mod(x^3+x^2+x+1)$。

2.1.7 共轭

对于 $GF(p^n)$域，具有相同最小多项式的元素称为共轭元素（conjugate elements），他们属于一个共轭类。若其中一个元素是 a，则关于 a 的共轭类为$a,a^p,a^{p^2},\cdots,a^{p^{n-1}}$，即 a^p 的幂次数。

若 a 的共轭类是 $GF(p^n)$中不可约多项式 $p(x)$的所有根，则 $p(x)$可表示为：

$$p(x)=(x-a)(x-a^p)(x-a^{p^2})\cdots(x-a^{p^{n-1}}) \tag{2-3}$$

表 2-3 给出了伽罗瓦域常见的共轭类及其对应的不可约多项式。

表 2-3 常见的共轭类

共轭类	不可约多项式
$\{1\}$	$x+1$
$\{a, a^2, a^4, a^8\}$	x^4+x+1
$\{a^3, a^6, a^9, a^{12}\}$	$x^4+x^3+x^2+x+1$
$\{a^5, a^{10}\}$	x^2+x+1
$\{a^7, a^{11}, a^{13}, a^{14}\}$	x^4+x^3+1

2.1.8 迹函数

$GF(p^n)$的迹函数可以表示为：

$$Tr\left(\alpha\right)=\sum_{i=0}^{n-1}\alpha^{p^i} \tag{2-4}$$

其中，$\alpha\in GF\left(p^n\right)$，$Tr\left(\alpha\right)\in GF\left(p\right)$。

迹函数具有以下性质：

（1）$Tr\left(\alpha+\beta\right)=Tr\left(\alpha\right)+Tr\left(\beta\right)$；

（2）$Tr\left(c\alpha\right)=cTr\left(\alpha\right)$；

因此，$GF(p^n)$域的迹函数可以认为是 $GF(p^n)$域映射到子域 $GF(p)$的线性变换，扩展域 $GF(p^n)$可以通过使用 $GF(p)$上的 n 次不可约多项式 $p(x)$来构造。扩展域的加法运算和乘法运算分别定义为，多项式的加法运算和多项式的乘法运算结果模 $p(x)$。在这种情况下，$p(x)$被称为 $GF(p)$域的多项式[40]。

以 $GF(2^4)$为例，从 $GF(2^4)$到 $GF(2)$的线性变化，其形式为：

$$Tr_{GF\left(2^n\right)/GF(2)}\left(x\right)=x+x^2+x^{2^2}+x^{2^3} \tag{2-5}$$

加法运算是 $GF(2)$域的模 2 运算，则 $Tr_{GF\left(2^n\right)/GF(2)}\left(x\right)\in GF\left(2\right)$，结果为 0 或者 1。

例如，$GF(2)$域的扩展域 $GF(2^4)$，可以通过 $GF(2)$域上的一个 4 次不可约多项式 $p\left(x\right)=x^4+x+1$构造，扩展域 $GF(2^4)$中的元素则可以表示为$\{a_0+a_1b+a_2b^2+a_3b^3\}$，其中$\left\{a_0,a_1,a_2,a_3\right\}\in GF\left(2\right)$域，$GF(2^4)$域中的元素表示是 2.2.2 节的多项式基表示法。

2.2 伽罗瓦域的构成

伽罗瓦域是指具有有限个元素的代数结构，其元素具有三种不同的构成方式，第一种是基于本原元素的幂构成伽罗瓦域中非零元素，第二种是由多项式表示伽罗瓦域元素，第三种是基于向量基构造伽罗瓦域元素[39-40]。

2.2.1 本原元素表示

根据本原元素的定义，伽罗瓦域的所有非零元素可以由本原元素的幂表示。假设 $GF(p^n)$域的本原元素为 a，则 $GF(p^n)$域中的所有元素可用集合$\left\{0,1,a^2,a^3\cdots a^{p^n-2}\right\}$表示。

例如，在 $GF(2^2)$域中，本原元素是Ω，则Ω可以构成 $GF(2^2)$域中其余非零元

素，即$1=\Omega^0$，$\Omega=\Omega$，$\psi=\Omega^2$，则 $GF(2^2)$域的元素集合可以表示为$\{0,\Omega^0,\Omega^1,\Omega^2\}$。基于本原元素构成的伽罗瓦域，可以极大地简化伽罗瓦域的计算复杂度。

2.2.2 多项式表示

$GF(p^n)$域的每个元素可以被认为是次数小于 n 的多项式，其系数属于 $GF(p)$域。加法运算是其系数逐位相加，得到的系数和对 p 取余；乘法运算是计算多项式的乘积，其结果对某个不可约多项式 $p(x)$取余，其前导系数等于 1。

例如，$GF(2^4)$域的元素$(1011)_2$，其多项式表示为x^3+x+1。$GF(2^2)$域的元素$(11)_2$，其多项式表示为$x+1$，元素$(10)_2$的多项式表示为 x，$(11)_2$和$(10)_2$的加法运算可表示为$(11)_2\oplus(10)_2=(x+1)(x)=(2mod2)*x+(1mod2)=1$；$GF(2^2)$域的不可约多项式$p(x)=x^2+x+1$，其乘法运算可表示为：$(11)_2\otimes(10)_2=(x+1)\times(x)mod\left(x^2+x+1\right)=1$。

2.2.3 向量基表示

多项式基、正规基和对偶基是目前伽罗瓦域常用的三种基，上述三种基在伽罗瓦域的向量空间结构中具有不同的性质。

（1）多项式基（Polynomial Basis）：对于 $GF(p^n)$域而言，多项式基通常定义为$\{1,b,b^2,\ldots,b^{n-1}\}$，伽罗瓦域中任意元素都可以由多项式基唯一的表示：$a_0+a_1b+a_2b^2+\cdots+a_{n-1}b^{n-1}$。例如，在 $GF(2^2)$域中，取多项式基$\{1,b\}$，伽罗瓦域中元素集合可表达为$\{0,1,b,1+b\}$。

（2）正规基（Normal Basis）：对于 $GF(p^n)$域而言，正规基通常由 n 个属于 $GF(p^n)$域且线性无关的元素组成，形如：$\{w,w^2,w^{2^2},\ldots,w^{2^{n-1}}\}$，正规基元素间的关系是共轭，伽罗瓦域中每个元素都可以由正规基的线性组合表达。例如，在 $GF(2^2)$域中，取正规基$\{w,w^2\}$，域中元素集合可表达为$\{0,w+w^2,w,w^2\}$。

（3）对偶基（Dual Basis）：对偶基是通过与原基的对偶关系定义的另一种基。若$\{\alpha_0,\alpha_1,\alpha_2,\ldots,\alpha_{n-1}\}$是 $GF(p^n)$域的一个基，设其对偶基为$\{\beta_0,\beta_1,\beta_2,\ldots,\beta_{n-1}\}$，则满足：

$$Tr(\alpha_i\beta_j)=\begin{cases}1 & if\ i=j\\ 0 & if\ i\neq j\end{cases} \tag{2-6}$$

根据文献[67-68]可以证明，对于任何伽罗瓦域中的基都存在唯一的对偶基。

例如，在 $GF(2^2)$域中，选择一组多项式基$\{1,a\}$，假设其对偶基为$\{b_0,b_1\}$，联立方程：

$$\begin{cases} Tr(1\cdot b_0)=1 \\ Tr(a\cdot b_0)=0 \\ Tr(1\cdot b_1)=0 \\ Tr(a\cdot b_1)=1 \end{cases} \tag{2-7}$$

计算得到多项式基$\{1,a\}$的对偶基为$\{a,a+1\}$。在原基下 $GF(2^2)$域元素表示为$\{0,1,a,1+a\}$，在对偶基表示下为$\{0,a+a+1,a,a+1\}$，如图 2-1 是基于多项式基和对偶基的 $GF(2^2)$域元素真值表。

⊕	1	a
0	0	0
1	1	0
a	0	1
a^2	1	1

(*a*) 原基真值表

⊕	a	a^2
0	0	0
1	1	1
a	1	0
a^2	0	1

(*b*) 对偶基真值表

图 2-1 对偶基下的 $GF(2^2)$域真值表

2.3 伽罗瓦域的运算法则

2.3.1 加法运算

伽罗瓦域中的加法运算可以看作模运算。对于 $GF(p^n)$域，加法运算可以看作模 p 运算。对于 $GF(p^n)$域中任意两个元素$\{a,b\}$，其加法运算规则如下：

$$a\oplus b\equiv(a+b)\bmod p \tag{2-8}$$

对于特征值为 2 的 $GF(2^n)$域，即 p=2，其加法运算可以看作按位异或运算或者模 2 运算。

例如，$GF(2^4)$域的两个元素$a=(11011111)_2$，$b=(10100011)_2$，元素的加法运算为$a\oplus b=(11011111)_2\oplus(10100011)_2$。对于 $GF(2^4)$域的多项式运算等价于逐项模 2 加法，假设多项式$m(x)=x^4+x+1$，$n(x)=x^4+x^3+1$，其多项式的加法运算为

$m(x)\oplus n(x)=\left(x^4+x^4\right)+x^3+x+1+1=x^3+x$。多项式 $m(x)$的元素表示为$(10011)_2$，多项式 $n(x)$的元素表示为$(11001)_2$，则其元素形式的加法运算为 $m(x)\oplus n(x)=(10011)_2\oplus(11001)_2=(01010)_2$，加法运算的多项式表示为$x^3+x$。

因此，在 $GF(2^2)$域中，多项式$x^2+1=x^2+x+x+1=(x+1)^2$，所以多项式x^2+1为可约多项式。

2.3.2 乘法运算

$GF(p^n)$域的多项式乘法运算可以看作对两个多项式（次数小于 n，系数属于 $GF(p)$）的乘积结果模不可约多项式 $p(x)$的取余运算，形如：

$$c(x)=a(x)\cdot b(x)\bmod p(x) \tag{2-9}$$

其中，$a(x),b(x)\in GF\left(p^n\right)$。

当 $GF(p^n)$域使用本原元素 a（阶数为 p^n-1）作为生成元时，其余非零元素可以表示为 a^k，则任意两个元素的乘法运算可以看作对两个元素乘积结果的阶数模 p^n-1，形如：

$$a^i\cdot a^j=a^{(i+j)\bmod\left(p^n-1\right)} \tag{2-10}$$

其中，$a^i,a^j\in GF\left(p^n\right)$。

例如，使用本原元素的构成 $GF(2^2)$域元素$\{0,1,x,x^2\}$，$GF(2^2)$域唯一的不可约多项式 $p(x)=x^2+x+1$，则 $GF(2^2)$域的多项式运算为：

$(x+1)\otimes(x)=(x+1)\times(x)\,mod\left(x^2+x+1\right)=\left(x^2+x\right)\oplus\left(x^2+x+1\right)=1$；

$\left(x^2+1\right)\otimes\left(x^2\right)=\left(x^2+1\right)\times\left(x^2\right)mod\left(x^2+x+1\right)=\left(x^2+x\right)\oplus\left(x^2+x+1\right)=1$；

对于元素的乘法运算：$1\otimes x=x^{1mod3}=x$，$x\otimes x^2=x^{3mod3}=1$。图 2-2 给出了 $GF(2^2)$域的元素运算结果，其中图 2-2(a)为加法运算，图 2-2(b)为乘法运算。

⊕	0	1	x	x^2
0	0	1	x	x^2
1	1	0	x^2	x
x	x	x^2	0	1
x^2	x^2	x	1	0

(*a*) 加法运算

⊗	0	1	x	x^2
0	0	0	0	0
1	0	1	x	x^2
x	0	x	x^2	1
x^2	0	x^2	1	x

(*b*) 乘法运算

图 2-2 $GF(2^2)$域运算法则

2.4 本章小节

本章系统地阐述了伽罗瓦域和复合域的理论基础，涵括了本文所涉及到的伽罗瓦域的基本性质、伽罗瓦域的构成以及伽罗瓦域计算法则。首先，详细地介绍了伽罗瓦域中的定义，如伽罗瓦域中的可约多项式、不可约多项式、本原多项式、本原元素以及复合域等；其次，介绍了伽罗域的三种构成方法，并举例示意其构成步骤；最后，针对伽罗瓦域中的乘法运算和加法运算进行了分析。本章的理论基础介绍为全文基于复合域的 AES S 盒的设计与优化提供了数学基础的支撑。

第 3 章 $GF((2^2)^4)$乘法逆结构

本章针对 AES 在密码电路中占用资源消耗过高的问题，对其轮变换结构中唯一的非线性部件 S 盒进行优化，研究了基于复合域 $GF((2^2)^4)$的乘法逆结构。首先，通过将出初等代数运算与有限域运算法则相结合的方式，推导复合域乘法逆结构通式，其次，分析复合域多项式特性，通过穷搜索法求出所有可约多项式、不可约多项式及本原多项式；最后对比分析不同系数下的复合域乘法逆结构复杂度，对复合域乘法逆结构进行优化，从而满足减少电路资源消耗的目的。

3.1 $GF((2^2)^4)$乘法逆结构通式

根据定理 2.1.1 知一个伽罗瓦域中的任意一非零元素 a 都有相对应的乘法逆元 a^{-1}，使得 $a\times a^{-1}=1$。设 $GF(p^n)$域一元素 g，存在 $g\times g^{-1}=1$，伽罗瓦域乘法运算是对乘积模不可约多项式 $q(x)$，即 $g\times g^{-1}=g\times g^{-1}\ mod\ q(x)$。

在 $GF(2^8)$域中，AES 算法使用 8 位字节的特定伽罗瓦域，位是多项式的系数。在 $GF(2^8)$域中常基于不可约多项式 $q(x)=x^8+x^4+x^3+x+1$（二进制表示为 $q(x)=100011011$；这是 $GF(2^8)$域上 8 次“最小的”不可约多项式）进行分析。直接计算一个八次多项式的逆（模为一个八次多项式）是不容易的。但是，正如 D. Canright[39]所指出的，计算一个四级多项式的逆（模一个四阶多项式）是相对容易的，因此将伽罗瓦域乘法逆结构映射到复合域中计算，将大大降低算法复杂度。

在复合域 $GF((2^2)^4)$中求解乘法逆元，依赖于其子域 $GF(2^2)$域的运算法则。具体推理流程如图 3-1 所示。

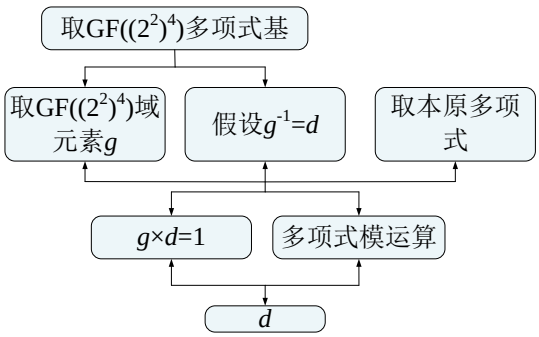


图 3-1 复合域 $GF((2^2)^4)$乘法逆结构求解流程

（1）首先根据 2.2.3 节多项式基的定理，取复合域 $GF((2^2)^4)$的多项式基$\{y^3, y^2, y^2, y\}$；

（2）在此多项式基的基础上构造复合域 $GF((2^2)^4)$元素 g，以及 g 的逆 d，（$\{g, d\} \in GF((2^2)^4)$）假设 g 和 d 表达式如下：

$$\begin{cases} g = r_3 y^3 + r_2 y^2 + r_1 y + r_0 \\ d = g^{-1} \\ d = \delta_3 y^3 + \delta_2 y^2 + \delta_1 y + \delta_0 \end{cases} \tag{3-1}$$

（3）取复合域 $GF((2^2)^4)$的不可约多项式 $q_1(y)$，假设 $q_1(y)$的表达式如下：

$$q_1(y) = y^4 + \beta_3 y^3 + \beta_2 y^2 + \beta_1 y + \beta_0 \tag{3-2}$$

（4）根据 2.3.2 节的伽罗瓦域乘法运算法则知伽罗瓦域的乘法运算是对乘积模不可约多项式运算，即：

$$g \times d = \left(r_3 y^3 + r_2 y^2 + r_1 y + r_0\right) \times \left(\delta_3 y^3 + \delta_2 y^2 + \delta_1 y + \delta_0\right) \bmod q_1(y) \tag{3-3}$$

联立公式(3-1)(3-2)(3-3)得到下式：

$$\begin{aligned} g \times d &= (r_3 y^3 + r_2 y^2 + r_1 y + r_0)\left(\delta_3 y^3 + \delta_2 y^2 + \delta_1 y + \delta_0\right) \bmod q_1(y) \\ &= [(r_3\delta_3) y^6 + (r_3\delta_2 + r_2\delta_3) y^5 + (r_3\delta_1 + r_2\delta_2 + r_1\delta_3) y^4 \\ &+ (r_3\delta_0 + r_2\delta_1 + r_1\delta_2 + r_0\delta_3) y^3 + (r_2\delta_0 + r_1\delta_1 + r_0\delta_2) y^2 \\ &+ (r_1\delta_0 + r_0\delta_1) y + r_0\delta_0] \\ &\bmod \left(y^4 + \beta_3 y^3 + \beta_2 y^2 + \beta_1 y + \beta_0\right) \\ &= (r_3\delta_3) y^6 + (r_3\delta_2 + r_2\delta_3) y^5 \\ &+ (r_3\delta_0 + r_2\delta_1 + r_1\delta_2 + r_0\delta_3 + (r_3\delta_1 + r_2\delta_2 + r_1\delta_3)\beta_3) y^3 \\ &+ \left(r_2\delta_0 + r_1\delta_1 + r_0\delta_2 + \beta_2 (r_3\delta_1 + r_2\delta_2 + r_1\delta_3)\right) y^2 \\ &+ \left(r_1\delta_0 + r_0\delta_1 + \beta_1 (r_3\delta_1 + r_2\delta_2 + r_1\delta_3)\right) \\ &y + r_0\delta_0 + \beta_0 (r_3\delta_1 + r_2\delta_2 + r_1\delta_3) \\ &= 1 \end{aligned} \tag{3-4}$$

（5）基于不可约多项式 $q_1(y)$，用多项式基$\{y^3, y^2, y^2, y\}$代表(3-4)式中的未知参数$\{y^6, y^5, y^4\}$，式(3-5)列出了其公式表示，其真值表如表 3-1 所示：

$$\begin{cases} y^4 = \beta_3 y^3 + \beta_2 y^2 + \beta_1 y + \beta_0 \\ y^5 = \left(\beta_3^2 + \beta_2\right) y^3 + \left(\beta_3\beta_2 + \beta_1\right) y^2 + \left(\beta_3\beta_1 + \beta_0\right) y + \beta_3\beta_0 \\ y^6 = \left(\beta_3^3 + \beta_1\right) y^3 + \left(\beta_3^2\beta_2 + \beta_2^2 + \beta_3\beta_1 + \beta_0\right) y^2 + \left(\left(\beta_3^2 + \beta_2\right)\beta_1 + \beta_3\beta_0\right) y \\ \qquad + \left(\beta_3^2 + \beta_2\right)\beta_0 \end{cases} \tag{3-5}$$

表 3-1 复合域 $GF((2^2)^4)$元素真值表

	y^3	y^2	y	1
1	0	0	0	1
y	0	0	1	0
y^2	0	1	0	0
y^3	1	0	0	0
y^4	β_3	β_2	β_1	β_0
y^5	$\beta_3^2 + \beta_2$	$\beta_3\beta_2 + \beta_1$	$\beta_3\beta_1 + \beta_0$	$\beta_3\beta_0$
y^6	$\beta_3^3 + \beta_1$	$\beta_3^2\beta_2 + \beta_2^2 + \beta_3\beta_1 + \beta_0$	$\left(\beta_3^2 + \beta_2\right)\beta_1 + \beta_3\beta_0$	$\left(\beta_3^2 + \beta_2\right)\beta_0$

（6）将公式(3-5)带入到公式(3-4)得到基于多项式基$\{y^3, y^2, y^2, y\}$的复合域乘法逆运算结果：

$$\begin{aligned} g \times d = & \begin{pmatrix} \left(\beta_3^3 + \beta_1\right) r_3\delta_3 + \left(\beta_3^2 + \beta_2\right)\left(r_3\delta_2 + r_2\delta_3\right) + r_3\delta_0 + r_2\delta_1 + r_1\delta_2 + r_0\delta_3 \\ + \beta_3\left(r_3\delta_1 + r_2\delta_2 + r_1\delta_3\right) \end{pmatrix} y^3 \\ & \begin{pmatrix} \left(\beta_3^2\beta_2 + \beta_2^2 + \beta_3\beta_1 + \beta_0\right) r_3\delta_3 + \left(\beta_3\beta_2 + \beta_1\right)\left(r_3\delta_2 + r_2\delta_3\right) + r_2\delta_0 \\ + r_1\delta_1 + r_0\delta_2 + \beta_2\left(r_3\delta_1 + r_2\delta_2 + r_1\delta_3\right) \end{pmatrix} y^2 \\ & + \begin{pmatrix} \left(\left(\beta_3^2 + \beta_2\right)\beta_1 + \beta_3\beta_0\right) r_3\delta_3 + \left(\beta_3\beta_1 + \beta_0\right)\left(r_3\delta_2 + r_2\delta_3\right) + r_1\delta_0 + r_0\delta_1 \\ + \beta_1\left(r_3\delta_1 + r_2\delta_2 + r_1\delta_3\right) \end{pmatrix} y \\ & + \left(\left(\left(\beta_3^2 + \beta_2\right)\beta_0\right) r_3\delta_3 + \beta_3\beta_0\left(r_3\delta_2 + r_2\delta_3\right) + r_0\delta_0 + \beta_0\left(r_3\delta_1 + r_2\delta_2 + r_1\delta_3\right)\right) \\ & = 1 \end{aligned} \tag{3-6}$$

（7）因为$g \times d = 1$，则式(3-6)中的非常数项的系数为 0，常数项的系数为 1，求解方程，从而得到如式(3-7)的复合域 $GF((2^2)^4)$的乘法逆结构通式：

$$\begin{cases} \delta_3 = Y_3 \times inv^{-1} \\ \delta_2 = Y_2 \times inv^{-1} \\ \delta_1 = Y_1 \times inv^{-1} \\ \delta_0 = Y_0 \times inv^{-1} \end{cases} \tag{3-7}$$

其参数$\{Y_3, Y_2, Y_1, Y_0, inv\}$见公式(3-8)所示，其中乘法运算是基于 $GF(2^2)$域乘法

器结构的运算，加法运算是异或运算。

$$
\begin{cases}
Y_3 = \beta_3^2 r_0 r_2 r_3 + \beta_3^2 r_1^2 r_3 + \beta_3\beta_2 r_0 r_3^2 + \beta_3\beta_2 r_1 r_2 r_3 + \beta_3\beta_1 r_2^2 r_3 + \beta_3 r_0 r_1 r_3 + \beta_3 r_0 r_2^2 + \beta_3 r_1^2 r_2 \\
\quad + \beta_3\beta_0 r_2 r_3^2 + \beta_2^2 r_1 r_3^2 + \beta_2\beta_1 r_2 r_3^2 + \beta_2 r_1 r_2^2 + \beta_2\beta_0 r_3^3 + \beta_1^2 r_3^3 + \beta_1 r_1 r_2 r_3 + \beta_1 r_2^3 + r_0^2 r_3 \\
\quad + r_1^3 + \beta_0 r_1 r_3^2 + \beta_0 r_2^2 r_3 \\
Y_2 = \beta_3^3 r_0 r_2 r_3 + \beta_3^3 r_1^2 r_3 + \beta_3^2\beta_2 r_0 r_3^2 + \beta_3^2\beta_2 r_1 r_2 r_3 + \beta_3^2\beta_1 r_2^2 r_3 + \beta_3^2 r_0 r_2^2 + \beta_3^2 r_1^2 r_2 \\
\quad + \beta_3^2\beta_0 r_2 r_3^2 + \beta_3\beta_2^2 r_1 r_3^2 + \beta_3\beta_2\beta_1 r_2 r_3^2 + \beta_3\beta_2 r_0 r_2 r_3 + \beta_3\beta_2 r_1 r_2^2 + \beta_3\beta_2\beta_0 r_3^3 + \beta_3\beta_1^2 r_3^3 \\
\quad + \beta_3\beta_1 r_1 r_2 r_3 + \beta_3\beta_1 r_2^3 + \beta_3 r_0 r_1 r_2 + \beta_3 r_1^3 + \beta_2^2 r_0 r_3^2 + \beta_2 r_0 r_2^2 + \beta_2\beta_0 r_2 r_3^2 + \beta_1 r_0 r_2 r_3 \\
\quad + \beta_1\beta_0 r_3^3 + r_0^2 r_2 + r_0 r_1^2 + \beta_0 r_0 r_3^2 + \beta_0 r_2^3 \\
Y_1 = \beta_3^3 r_0 r_1 r_3 + \beta_3^2\beta_2 r_1^2 r_3 + \beta_3^2\beta_1 r_0 r_3^2 + \beta_3^2\beta_0 r_1 r_3^2 + \beta_3^2\beta_0 r_2^2 r_3 + \beta_3^2 r_0 r_1 r_2 + \beta_3\beta_2^2 r_1 r_2 r_3 \\
\quad + \beta_3\beta_2\beta_1 r_2^2 r_3 + \beta_3\beta_2 r_0 r_1 r_3 + \beta_3\beta_2 r_1^2 r_2 + \beta_3\beta_1\beta_0 r_3^3 + \beta_3\beta_0 r_0 r_3^2 + \beta_3\beta_0 r_1 r_2 r_3 + \beta_3\beta_0 r_2^3 \\
\quad + \beta_3 r_0 r_1^2 + \beta_2^3 r_1 r_3^2 + \beta_2^2\beta_1 r_2 r_3^2 + \beta_2^2\beta_0 r_3^3 + \beta_2^2 r_1 r_2^2 + \beta_2\beta_1^2 r_3^3 + \beta_2\beta_1 r_0 r_3^2 + \beta_2\beta_1 r_1 r_2 r_3 \\
\quad + \beta_2\beta_1 r_2^3 + \beta_2\beta_0 r_2^2 r_3 + \beta_2 r_1^3 + \beta_1 r_0 r_1 r_3 + \beta_1 r_0 r_2^2 + \beta_0^2 r_3^3 + \beta_0 r_1^2 r_3 + \beta_0 r_1 r_2^2 + r_0^2 r \\
Y_0 = \beta_3^3 r_0^2 r_3 + \beta_3^2\beta_2 r_0 r_1 r_3 + \beta_3^2\beta_1 r_1^2 r_3 + \beta_3^2\beta_0 r_1 r_2 r_3 + \beta_3^2 r_0^2 r_2 + \beta_3\beta_2^2 r_0 r_2 r_3 + \beta_3\beta_2\beta_1 r_0 r_3^2 \\
\quad + \beta_3\beta_2\beta_1 r_1 r_2 r_3 + \beta_3\beta_2\beta_0 r_2^2 r_3 + \beta_3\beta_2 r_0^2 r_3 + \beta_3\beta_2 r_0 r_1 r_2 + \beta_3\beta_1^2 r_2^2 r_3 + \beta_3\beta_1\beta_0 r_2 r_3^2 \\
\quad + \beta_3\beta_1 r_0 r_1 r_3 + \beta_3\beta_1 r_1^2 r_2 + \beta_3\beta_0^2 r_3^3 + \beta_3\beta_0 r_0 r_2 r_3 + \beta_3\beta_0 r_1 r_2^2 + \beta_3 r_0^2 r_1 + \beta_1^3 r_0 r_3^2 \\
\quad + \beta_2^2\beta_1 r_1 r_3^2 + \beta_2^2\beta_0 r_2 r_3^2 + \beta_2^2 r_0 r_2^2 + \beta_2\beta_1^2 r_2 r_3^2 + \beta_2\beta_1 r_0 r_2 r_3 + \beta_2\beta_1 r_1 r_2^2 + \beta_2\beta_0 r_2^3 \\
\quad + \beta_2 r_0 r_1^2 + \beta_1^3 r_3^3 + \beta_1^2 r_0 r_3^2 + \beta_1^2 r_1 r_2 r_3 + \beta_1^2 r_2^3 + \beta_1\beta_0 r_2^2 r_3 + \beta_1 r_0^2 r_3 + \beta_1 r_0 r_1 r_2 + \beta_1 r_1^3 \\
\quad + \beta_0^2 r_2 r_3^2 + \beta_0 r_0 r_2^2 + \beta_0 r_1^2 r_2 + r_0^3 \\
inv = \beta_3^3 r_0^3 r_3 + \beta_3^2\beta_2 r_0^2 r_1 r_3 + \beta_3^2\beta_1 r_0 r_1^2 r_3 + \beta_3^2\beta_0 r_0^2 r_3^2 + \beta_3^2\beta_0 r_0 r_1 r_2 r_3 + \beta_3^2\beta_0 r_1^3 r_3 + \beta_3^2 r_0^3 r_2 \\
\quad + \beta_3\beta_2^2 r_0^2 r_2 r_3 + \beta_3\beta_2\beta_1 r_0^2 r_3^2 + \beta_3\beta_2\beta_1 r_0 r_1 r_2 r_3 + \beta_3\beta_2\beta_0 r_0 r_1 r_3^2 + \beta_3\beta_2\beta_0 r_1^2 r_2 r_3 \\
\quad + \beta_3\beta_2 r_0^3 r_3 + \beta_3\beta_2 r_0^2 r_1 r_2 + \beta_3\beta_1^2 r_0 r_2^2 r_3 + \beta_3\beta_1\beta_0 r_0 r_2 r_3^2 + \beta_3\beta_1\beta_0 r_1 r_2^2 r_3 + \beta_3\beta_1 r_0^2 r_1 r_3 \\
\quad + \beta_3\beta_1 r_0 r_1^2 r_2 + \beta_3\beta_0^2 r_0 r_3^3 + \beta_3\beta_0^2 r_1 r_2 r_3^2 + \beta_3\beta_0^2 r_2^3 r_3 + \beta_3\beta_0 r_0^2 r_2 r_3 + \beta_3\beta_0 r_0 r_1^2 r_3 \\
\quad + \beta_3\beta_0 r_0 r_1 r_2^2 + \beta_3\beta_0 r_1^3 r_2 + \beta_3 r_0^3 r_1 + \beta_2^3 r_0^2 r_3^2 + \beta_2^2\beta_1 r_0 r_1 r_3^2 + \beta_2^2\beta_0 r_1^2 r_3^2 + \beta_2^2 r_0^2 r_2^2 \\
\quad + \beta_2\beta_1^2 r_0 r_2 r_3^2 + \beta_2\beta_1\beta_0 r_0 r_3^3 + \beta_2\beta_1\beta_0 r_1 r_2 r_3^2 + \beta_2\beta_1 r_0^2 r_2 r_3 + \beta_2\beta_1 r_0 r_1 r_2^2 + \beta_2\beta_0^2 r_2^2 r_3^2 \\
\quad + \beta_2\beta_0 r_0^2 r_3^2 + \beta_2\beta_0 r_1^2 r_2^2 + \beta_2 r_0^2 r_1^2 + \beta_1^3 r_0 r_3^3 + \beta_1^2\beta_0 r_1 r_3^3 + \beta_1^2 r_0^2 r_3^2 + \beta_1^2 r_0 r_1 r_2 r_3 \\
\quad + \beta_1^2 r_0 r_2^3 + \beta_1\beta_0^2 r_2 r_3^3 + \beta_1\beta_0 r_0 r_1 r_3^2 + \beta_1\beta_0 r_0 r_2^2 r_3 + \beta_1\beta_0 r_1^2 r_2 r_3 + \beta_1\beta_0 r_1 r_2^3 + \beta_1 r_0^3 r_3 \\
\quad + \beta_1 r_0^2 r_1 r_2 + \beta_1 r_0 r_1^3 + \beta_0^3 r_3 + \beta_0^2 r_2 + \beta_0 r_1 + r_0
\end{cases}
\tag{3-8}
$$

式(3-8)中的所有系数皆属于 $GF(2^2)$域。

3.2 $GF((2^2)^4)$乘法器结构

计算复合域 $GF((2^2)^4)$的乘法器结构，需要首先明确复合域的元素表示。根据同

构映射法将 $GF((2^8)$域中的 8bit 元素表示为复合域 $GF((2^2)^4)$中二元组的形式，即 $GF((2^8)$域中一个元素有 8 位，即 $x=(x_7x_6x_5x_4x_3x_2x_1x_0)_2$，从高位到低位两位一组，依次记作$\{a_3,a_2,a_1,a_0\}$，则复合域 $GF((2^2)^4)$元素 a 表示为$\left(a_3a_2a_1a_0\right)$。其元素映射图解如图 3-2 所示。

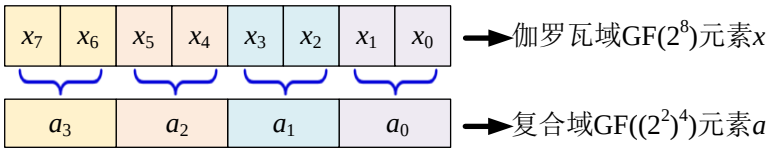


图 3-2 $GF((2^8)$域中元素映射到复合域 $GF((2^2)^4)$

复合域 $GF((2^2)^4)$的元素运算是基于 $GF(2^2)$域的运算。$GF(2^2)$域在多项式基$\{1,y\}$下，任意两个元素$\{a,b\}$分别表示为$\{a=a_1y+a_0,b=b_1y+b_0\}$，根据 $GF(2^2)$域的乘法运算公式得到$a\times b=(a_0b_1+a_1b_0+a_1b_1)y+(a_0b_0+a_1b_1)$。

复合域 $GF((2^2)^4)$的乘法器运算步骤如下所示：

（1）取复合域 $GF((2^2)^4)$不可约多项式 $f\left(y\right)=y^4+\beta_3y^3+\beta_2y^2+\beta_1y+\beta_0$；

（2）取复合域 $GF((2^2)^4)$的多项式基：$\{y^3,y^2,y^2,y\}$；

（3）取复合域 $GF((2^2)^4))$三个元素$\{a,b,c\}$，其多项式基表示如下：

$$\begin{cases}a=a_3y^3+a_2y^2+a_1y+a_0\\b=b_3y^3+b_2y^2+b_1y+b_0\\c=c_3y^3+c_2y^2+c_1y+c_0\end{cases}\tag{3-9}$$

假设 $c=a\times b$，则：

$$\begin{aligned}a\times b=&a_3b_3y^6+a_3b_2y^5+a_3b_1y^4+a_3b_0y^3\\&+a_2b_3y^5+a_2b_2y^4+a_2b_1y^3+a_2b_0y^2\\&+a_1b_3y^4+a_1b_2y^3+a_1b_1y^2+a_1b_0y\\&+a_0b_3y^3+a_0b_2y^2+a_0b_1y^1+a_0b_0\\=&c_3y^3+c_2y^2+c_1y+c_0\end{aligned}\tag{3-10}$$

（4）利用如表 3-1 所示的真值表得到$\{y^6,y^5,y^4\}$的多项式基表示；

（5）将第四步的结果带入(3-10)中的复合域乘法器得到：

$$\begin{cases} c_3=\left(\beta_3^3+\beta_1\right)a_3b_3+\left(\beta_3^2+\beta_2\right)\left(a_3b_2+a_2b_3\right)+\beta_3\left(a_3b_1+a_2b_2+a_1b_3\right) \\ \quad +a_3b_0+a_2b_1+a_1b_2+a_0b_3 \\ c_2=\left(\beta_3^2\beta_2+\beta_2^2+\beta_3\beta_1+\beta_0\right)a_3b_3+\left(\beta_3\beta_2+\beta_1\right)\left(a_3b_2+a_2b_3\right) \\ \quad +\beta_2\left(a_3b_1+a_2b_2+a_1b_3\right)+a_2b_0+a_1b_1+a_0b_2 \\ c_1=\left(\left(\beta_3^2+\beta_2\right)\beta_1+\beta_3\beta_0\right)a_3b_3+\left(\beta_3\beta_1+\beta_0\right)\left(a_3b_2+a_2b_3\right) \\ \quad +\beta_1\left(a_3b_1+a_2b_2+a_1b_3\right)+a_1b_0+a_0b_1 \\ c_0=\left(\left(\beta_3^2+\beta_2\right)\beta_0\right)a_3b_3+\beta_3\beta_0\left(a_3b_2+a_2b_3\right)+\beta_0\left(a_3b_1+a_2b_2+a_1b_3\right) \\ \quad +a_0b_0 \end{cases} \tag{3-11}$$

其中形如$\{a_3b_3,a_3b_2,a_3b_1,a_3b_0,\ldots\ldots,a_0b_1,a_3b_3\}$等运算是基于 $GF(2^2)$域的乘法运算。

当 $a=b$ 时，$a\times b=a^2$ 运算可由如式(3-11)乘法器计算得到：

$$a^2=\begin{cases} a_3^2=\left(\beta_3^3+\beta_1\right)a_3+\beta_3a_2 \\ a_2^2=\left(\beta_3^2\beta_2+\beta_2^2+\beta_3\beta_1+\beta_0\right)a_3+\beta_3a_2+a_1 \\ a_1^2=\left(\left(\beta_3^2+\beta_2\right)\beta_1+\beta_3\beta_0\right)a_3+\beta_1a_2 \\ a_0^2=\left(\left(\beta_3^2+\beta_2\right)\beta_0\right)a_3+\beta_0a_2+a_0 \end{cases} \tag{3-12}$$

3.3 复合域多项式

本文将 $GF(2^8)$域同构映射至复合域 $GF((2^2)^4)$域，并以此为研究对象，假设复合域 $GF((2^2)^4)$的多项式 $f\left(y\right)=y^4+\beta_3y^3+\beta_2y^2+\beta_1y+\beta_0$，其中$\{\beta_3,\beta_2,\beta_1,\beta_0\}\in GF\left(2^2\right)$，$GF(2^2)$中元素有$\left\{0,1,x,x^2\right\}$。

根据 2.1.3 章节知伽罗瓦域 $GF(p^n)$域的多项式可以分为可约多项式与不可约多项式，图 3-3 给出了伽罗瓦域中多项式的分类。

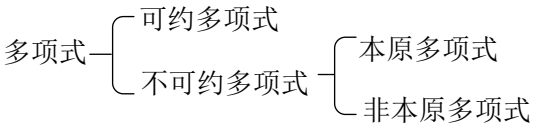


图 3-3 伽罗瓦域多项式分类

可约多项式具有以下特点：

（1）由于可约多项式可以分解为多个多项式的乘积，则可约多项式所构造的伽罗瓦域不是一个单一的代数结构，简单来讲，就是基于可约多项式的伽罗瓦域，其域中任意元素进行加法或者乘法运算时会生成不属于该伽罗瓦域的元素，会破坏了伽罗瓦域的封闭性；

（2）基于可约多项式所构造的伽罗瓦域，会导致部分元素无法在伽罗瓦域中找到相对应的唯一乘法逆元，违背了伽罗瓦域的乘法逆元存在性。

例如，基于可约多项式 $f(x)=x^2+x$ 所构造的 $GF(2^2)$域，$x\times(x+1)=0$，存在两个非零元素的乘积为 0，即此伽罗瓦域存在零因子，根据 2.1.1 节中域的定义所述，域中不存在零因子，零因子是伽罗瓦域的元素之外的元素，与之相悖，因此违背了伽罗瓦域的封闭性；$GF(2^2)$域中任意一非零元素为 $a=a_1x+a_0$，设其逆为 $a^{-1}=A_1x+A_0$，则 $a\otimes a^{-1}=1$，而元素 x^2 在此可约多项式下就不存在乘法逆元，即违背了 2.1.1 域的定义中的乘法逆元存在性。因此基于可约多项式无法构建伽罗瓦域结构，图 3-4 给出了此示例的具体计算过程。

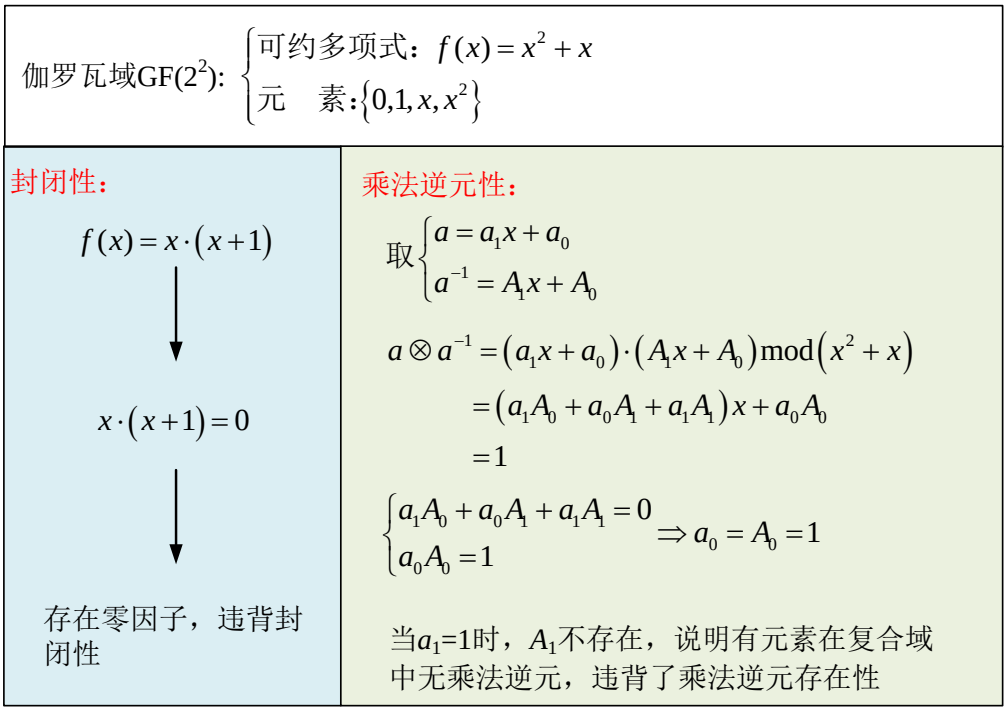


图 3-4 基于可约多项式的伽罗瓦域

不可约多项式是构建伽罗瓦域的基础，根据定义 2.1.3 知不可约多项式无法分解为多个多项式的乘积，因此基于不可约多项式的伽罗瓦域不会生成零因子，确保了

此伽罗瓦域的封闭性。

不可约多项式又可以分为非本原多项式和本原多项式，本原多项式的根为本原元素，根据定义 2.1.4 知，本原元素的幂可以生成伽罗瓦域中其余非零元素。也就是说，在存在本原元素的伽罗瓦域中，每个非零元素都存在它的乘法逆元，且不会生成伽罗瓦域以外的元素，保证了伽罗瓦域构成的封闭性和乘法逆元存在性，所以本原多项式的根能够构建一个完整的伽罗瓦域。

基于非本原多项式的不可约多项式所构建的伽罗瓦域，虽然可以保证所伽罗瓦域的封闭性，由于它的根是非本原元素，无法生成域中所有非零元素，因此无法保证伽罗瓦域的乘法逆元存在性。例如，基于非本原多项式的不可约多项式 $f(x)=x^3+x^2+1$ 构造 $GF(2^3)$域，对于 $GF(2^3)$域中任意一非零元素 $a=a_2x^2+a_1x+a_0$，设其逆为 $a^{-1}=A_2x^2+A_1x+A_0$，则 $a\otimes a^{-1}=1$，而元素 $(001)_2$ 在此不可约多项式下不存在乘法逆元，违背了 2.1.1 节定义中的乘法逆元存在性，图 3-5 给出了此理论的验证过程。

伽罗瓦域GF(2^3): $\begin{cases}\text{非本原不可约多项式：}f(x)=x^3+x^2+1\\ \text{元　素：}\{0,1,x,x^2,x^3,x^4,x^5,x^6,x^7\}\end{cases}$

乘法逆元性：

取 $\begin{cases}a=a_2x^2+a_1x+a_0\\ a^{-1}=A_2x^2+A_1x+A_0\end{cases}$

由不可约多项式$f(x)$知，$x^4=x^3+x$

$$a\otimes a^{-1}=(a_2x^2+a_1x+a_0)\cdot(A_2x^2+A_1x+A_0)\bmod(x^3+x^2+1)$$

$$=(a_2A_2)x^4+(a_2A_1+a_1A_2)x^3+(a_2A_0+a_0A_2+a_1A_1)x^2+(a_1A_0+a_0A_1)x+a_0A_0 \bmod (x^3+x^2+1)$$

$$=(a_2A_2+a_2A_0+a_1A_1+a_0A_2+a_2A_1+a_1A_2)x^2+(a_2A_2+a_1A_0+a_0A_1)x+(a_1A_0+a_0A_1+a_0A_0)$$

$$=1$$

$$\begin{cases}a_2A_2+a_2A_0+a_1A_1+a_0A_2+a_2A_1+a_1A_2=0\\ a_2A_2+a_1A_0+a_0A_1=0\\ a_1A_0+a_0A_1+a_0A_0=1\end{cases}$$

当a_2=0，a_1=0，a_0=1时，A_1不存在，说明有元素在复合域中无乘法逆元，违背了乘法逆元存在性

图 3-5 基于非本原不可约多项式的伽罗瓦域

因此构造复合域 $GF((2^2)^4)$的前提，就是查找其本原多项式。对于复合域 $GF((2^2)^4)$的多项式 $f(x)$，具有 $4^4=256$ 种结构。在这 256 种结构中查找本原多项式的步骤如图 3-6 所示，首先在这 256 种结构中列出所有多项式可分解形式，利用穷搜索法逆推找出所有可约多项式，除去可约多项式外其余的为不可约多项式；其次在不可约多项式中利用穷搜索法求出本原多项式。在求取复合域的本原多项式中容易有一个误区为：易将 p 看作 2^2，利用 2.1.5 节定义求 $GF(p^4)$域的本原多项式，但是根据 2.1.2 节定理知在有限域中 p 应当为素数，而 2^2 并不满足这一条件，因此不能将 $f(x)$整除 $x^{255}-1$ 推导复合域的本原多项式。

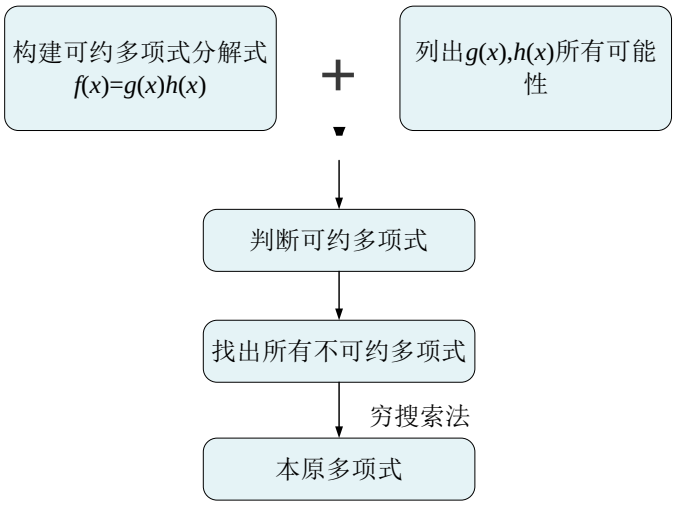


图 3-6 本原多项式计算步骤

3.3.1 可约多项式

对于复合域 $GF((2^2)^4)$的多项式 $f(y)=y^4+\beta_3y^3+\beta_2y^2+\beta_1y+\beta_0$，$(\{\beta_3,\beta_2,\beta_1,\beta_0\}\in GF(2^2))$，可约多项式可写作 $f(x)=g(x)h(x)$，按照子项 $g(x)$和 $h(x)$的最高次幂可以将 $f(x)$分为如式 3-13 所示的三种情况，其中 $\{m,n,p,q\}\in GF(2^2)$：

$$y^4+\beta_3y^3+\beta_2y^2+\beta_1y+\beta_0=\begin{cases}(y+m)(y+n)(y+p)(y+q)\\(y^2+my+n)(y^2+py+q)\\(y^3+my^2+ny+p)(y+q)\end{cases}\tag{3-13}$$

根据穷搜索法可以得到以下结果：

（1）分解子项 $g(x)$、$h(x)$的最高次幂为 1 时，$f(x)=(y+m)(y+n)(y+p)(y+q)$。当系

数{m, n, p, q}四个元素互相相等时，即 $m=n=p=q$，$f(x)=(y+m)^4$，存在 4 个可约多项式；当系数{m, n, p, q}中有三个元素相等时，$f(x)=(y+m)^3(y+n)$，存在 12 个可约多项式；当系数{m, n, p, q}中元素两两相等时，$f(x)=(y+m)^2(y+n)^2$，存在 6 个可约多项式；当系数{m, n, p, q}中元素有两个相等时，$f(x)= (y+m)^2(y+n)(y+p)$，存在 12 个可约多项式；当系数{m, n, p, q}互不相等时，$f(x)=(y+m)(y+n)(y+p)(y+q)$，存在 1 个可约多项式。最终，在分解子项的最高次幂为 1 的情况下，可以得到 35 个可约多项式。其分类见表 3-2 所示。

表 3-2 分解子项的最高次幂为 1 时的可约多项式个数

系数相等个数	形式	可约多项式个数	重复可约多项式个数	不重复可约多项式个数
4	$f(x)=(y+m)^4$	4	0	4
3	$f(x)=(y+m)^3(y+n)$	12	0	12
2	$f(x)=(y+m)^2(y+n)^2$	6	0	6
	$f(x)= (y+m)^2(y+n)(y+p)$	12	0	12
1	$f(x)=(y+m)(y+n)(y+p)(y+q)$	1	0	1
总计	—	35	0	35

（2）分解子项的最高次幂为 2 时，$f(x)=(y^2+my+n)(y^2+py+q)$。当系数{$m$, n, p, q}中不存在零元素时，$f(x)=(y^2+my+n)(y^2+py+q)$，存在 45 个可约多项式，其中有 6 个与已计算可约多项式重复，即存在 39 个不重复可约多项式。

当系数{m, n, p, q}中存在一个零元素 m 或者 p 时，$f(x)=(y^2+n)(y^2+py+q)$或者 $f(x)=(y^2+my+n)(y^2+q)$，存在 27 个可约多项式，其中有 9 个与已计算可约多项式重复，即存在 18 个不重复可约多项式；当系数{m, n, p, q}中存在一个零元素 n 或者 q 时，$f(x)=(y^2+my)(y^2+py+q)$或者 $f(x)=(y^2+my+n)(y^2+py)$，存在 27 个可约多项式，其中有 9 个与已计算可约多项式重复，即存在 18 个不重复可约多项式。总结得到，当系数{m, n, p, q}中存在一个零元素，共存在 36 个不重复可约多项式。

当系数{m, n, p, q}中存在两个零元素{m, n}或者{p, q}时，$f(x)=y^2(y^2+py+q)$或者 $f(x)=y^2(y^2+my+n)$，存在 9 个可约多项式，其中有 3 个与已计算可约多项式重复，即存在 6 个不重复可约多项式；当系数{m, n, p, q}中存在两个零元素{m, q}或者{n, p}时，$f(x)=y(y^2+n)(y+p)$或者 $f(x)=y(y+m)(y^2+q)$，存在 9 个可约多项式，其中有 9 个与已计算可约多项式重复，即存在 0 个不重复可约多项式；当系数{m, n, p, q}中存在

两个零元素$\{m, p\}$时，$f(x)=(y^2+n)(y^2+q)$，存在 9 个可约多项式，其中有 9 个与已计算可约多项式重复，即存在 0 个不重复可约多项式；当系数$\{m, n, p, q\}$中存在两个零元素$\{n, q\}$时，$f(x)=y^2(y+m)(y+p)$，存在 9 个可约多项式，其中有 9 个与已计算可约多项式重复，即存在 0 个不重复可约多项式。总结得到，当系数$\{m, n, p, q\}$中存在两个零元素，共存在 6 个不重复可约多项式。

当系数$\{m, n, p, q\}$中存在三个零元素$\{n, p, q\}$或者$\{m, n, q\}$时，$f(x)=y^3(y+m)$或者$f(x)=y^3(y+p)$，存在 3 个可约多项式，其中有 3 个与已计算可约多项式重复，即存在 0 个不重复可约多项式；当系数$\{m, n, p, q\}$中存在三个零元素$\{m, p, q\}$或者$\{m, n, p\}$时，$f(x)=y^2(y^2+n)$或者$f(x)=y^2(y^2+q)$，存在 3 个可约多项式，其中有 3 个与已计算可约多项式重复，即存在 0 个不重复可约多项式。总结得到，当系数$\{m, n, p, q\}$中存在三个零元素，共存在 0 个不重复可约多项式。

当系数$\{m, n, p, q\}$中存在四个零元素时，$f(x)=y^4$，与已计算可约多项式重复。

综上所述，当分解子项的最高次幂为 2 时，共存在 81 个不重复可约多项式。其分类可见表 3-3。

表 3-3 分解子项的最高次幂为 2 时的可约多项式个数

系数中 0 的个数	系数间关系	形式	可约多项式个数	重复可约多项式个数	不重复可约多项式个数
0		$f(x)=(y^2+my+n)(y^2+py+q)$	45	6	39
1	$m=0$ or $p=0$	$f(x)=(y^2+n)(y^2+py+q)$ $f(x)=(y^2+my+n)(y^2+q)$	27	9	18
	$n=0$ or $q=0$	$f(x)=(y^2+my)(y^2+py+q)$ $f(x)=(y^2+my+n)(y^2+py)$	27	9	18
2	$m=n=0$ or $p=q=0$	$f(x)=y^2(y^2+py+q)$ $f(x)=y^2(y^2+my+n)$	9	3	6
	$m=p=0$ or $n=q=0$	$f(x)=y(y^2+n)(y+p)$ $f(x)=y(y+m)(y^2+q)$	9	9	0
	$m=p=0$	$f(x)=(y^2+n)(y^2+q)$	9	9	0
	$n=q=0$	$f(x)=y^2(y+m)(y+p)$	9	9	0
3	$n=p=q=0$ or $m=n=q$	$f(x)=y^3(y+m)$ $f(x)=y^3(y+p)$	3	3	0
	$m=p=q$ or $m=n=p$	$f(x)=y^2(y^2+n)$ $f(x)=y^2(y^2+q)$	3	3	0
4	$m=n=p=q=0$	$f(x)=y^4$	1	1	0
总计	—	—	142	61	81

（3）分解子项的最高次幂为 3 时，$f(x)=(y^3+my^2+ny+p)(y+q)$。当系数$\{m, n, p, q\}$中不存在零元素时，$f(x)=(y^3+my^2+ny+p)(y+q)$，存在 81 个可约多项式，其中有 42 个与已计算可约多项式重复，即存在 39 个不重复可约多项式。

当系数$\{m, n, p, q\}$中存在一个零元素 m 时，$f(x)=(y^3+ny+p)(y+q)$，存在 27 个可约多项式，其中有 17 个与已计算可约多项式重复，即存在 10 个不重复可约多项式；当系数$\{m, n, p, q\}$中存在一个零元素 n 时，$f(x)=(y^3+my^2+p)(y+q)$，存在 27 个可约多项式，其中有 19 个与已计算可约多项式重复，即存在 8 个不重复可约多项式；系数$\{m, n, p, q\}$中存在一个零元素 p 时，$f(x)=y(y^2+my+n)(y+q)$，存在 27 个可约多项式，其中有 27 个与已计算可约多项式重复，即存在 0 个不重复可约多项式；系数$\{m, n, p, q\}$中存在一个零元素 q 时，$f(x)=y(y^3+my^2+ny+p)$，存在 27 个可约多项式，其中有 15 个与已计算可约多项式重复，即存在 12 个不重复可约多项式。总结得到，当系数$\{m, n, p, q\}$中存在一个零元素，共存在 30 个不重复可约多项式。

当系数$\{m, n, p, q\}$中存在两个零元素$\{m,n\}$时，$f(x)=(y^3+p)(y+q)$，存在 9 个可约多项式，其中有 4 个与已计算可约多项式重复，即存在 5 个不重复可约多项式；当系数$\{m, n, p, q\}$中存在两个零元素$\{m,p\}$时，$f(x)=y(y^2+n)(y+q)$，存在 9 个可约多项式，其中有 9 个与已计算可约多项式重复，即存在 0 个不重复可约多项式；当系数$\{m, n, p, q\}$中存在两个零元素$\{m, q\}$时，$f(x)=y(y^3+ny+p)$，存在 9 个可约多项式，其中有 6 个与已计算可约多项式重复，即存在 3 个不重复可约多项式；当系数$\{m, n, p, q\}$中存在两个零元素$\{n, p\}$时，$f(x)=y^2(y+m)(y+p)$，存在 9 个可约多项式，其中有 9 个与已计算可约多项式重复，即存在 0 个不重复可约多项式；当系数$\{m, n, p, q\}$中存在两个零元素$\{n, q\}$时，$f(x)=y(y^3+my^2+p)$，存在 9 个可约多项式，其中有 6 个与已计算可约多项式重复，即存在 3 个不重复可约多项式；当系数$\{m, n, p, q\}$中存在两个零元素$\{p,q\}$时，$f(x)=y^2(y^2+my+n)$，存在 9 个可约多项式，其中有 9 个与已计算可约多项式重复，即存在 0 个不重复可约多项式。总结得到，当系数$\{m, n, p, q\}$中存在两个零元素，共存在 11 个不重复可约多项式。

当系数$\{m, n, p, q\}$中存在三个零元素$\{n,p,q\}$时，$f(x)=y^3(y+m)$，存在 3 个可约多项式，其中有 3 个与已计算可约多项式重复，即存在 0 个不重复可约多项式；当系

数{m, n, p, q}中存在三个零元素{m, p, q}时，$f(x)=y^2(y^2+n)$，存在 3 个可约多项式，其中有 3 个与已计算可约多项式重复，即存在 0 个不重复可约多项式；当系数{m, n, p, q}中存在三个零元素{m, n, q}时，$f(x)=y(y^3+p)$，存在 3 个可约多项式，其中有 1 个与已计算可约多项式重复，即存在 2 个不重复可约多项式；当系数{m, n, p, q}中存在三个零元素{m, n, p}时，$f(x)=y^3(y+q)$，存在 3 个可约多项式，其中有 3 个与已计算可约多项式重复，即存在 0 个不重复可约多项式。总结得到，当系数{m, n, p, q}中存在三个零元素，共存在 2 个不重复可约多项式。

当系数{m, n, p, q}中存在四个零元素时，$f(x)=y^4$，与已计算可约多项式重复。

综上所述，当分解子项的最高次幂为 3 时，共存在 82 个不重复可约多项式。其分类如表 3-4 所示。

表 3-4 分解子项的最高次幂为 3 时的可约多项式个数

系数中 0 的个数	系数间关系	形式	可约多项式个数	重复可约多项式个数	不重复可约多项式个数
0		$f(x)=(y^3+my^2+ny+p)(y+q)$	81	42	39
1	$m=0$	$f(x)=(y^3+ny+p)(y+q)$	27	17	10
	$n=0$	$f(x)=(y^3+my^2+p)(y+q)$	27	19	8
	$p=0$	$f(x)=y(y^2+my+n)(y+q)$	27	27	0
	$q=0$	$f(x)=y(y^3+my^2+ny+p)$	27	15	12
2	$m=n$	$f(x)=(y^3+p)(y+q)$	9	4	5
	$m=p$	$f(x)=y(y^2+n)(y+q)$	9	9	0
	$m=q$	$f(x)=y(y^3+ny+p)$	9	6	3
	$n=p$	$f(x)=y^2(y+m)(y+p)$	9	9	0
	$n=q$	$f(x)=y(y^3+my^2+p)$	9	6	3
	$p=q$	$f(x)=y^2(y^2+my+n)$	9	9	0
3	$n=p=q$	$f(x)=y^3(y+m)$	3	3	0
	$m=p=q$	$f(x)=y^2(y^2+n)$	3	3	0
	$m=n=q$	$f(x)=y(y^3+p)$	3	1	2
	$m=n=p$	$f(x)=y^3(y+q)$	3	3	0
4	$m=n=p=q$	$f(x)=y^4$	1	1	0
总计			256	174	82

综上所述，复合域 $GF((2^2)^4)$共存在 198 种可约多项式。具体的可约多项式结构见附表 1。

3.3.2 不可约多项式

复合域 $GF((2^2)^4)$共存在 256 个多项式结构中，其中可约多项式有 198 个，则不可约多项式个数为 58 个，具体的不可约多项式结构见附表 2。

对 $GF((2^8)$域的不可约多项式进行求根运算来判断该不可约多项式是否为本原多项式，其运算关系建立在复合域 $GF((2^2)^4)$的基础上，运算法则建立在伽罗瓦域的运算法则中。不同的复合域 $GF((2^2)^4)$不可约多项式，将会导致 $GF(2^8)$域不可约多项式根的不同。若此不可约多项式为本原多项式，则 $GF(2^8)$域具有 8 个根；否则，为非本原多项式。

3.3.3 本原多项式

在复合域 $GF((2^2)^4)$的 58 个不可约多项式中，根据穷搜索法逐一在复合域 $GF((2^2)^4)$不可约多项式的基础上对 $GF(2^8)$域不可约多项式 $P(x)=x^8+x^4+x^3+x+1$ 进行求根运算，求根运算的图解如图 3-7 所示。

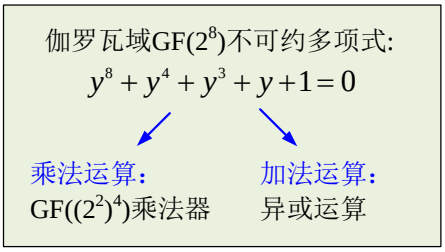


图 3-7 $GF(2^8)$域求根运算图解

其加法运算可以看作异或运算，乘法运算是复合域 $GF((2^2)^4)$乘法器运算，从而计算出 $GF(2^8)$域不可约多项式的根。

表 3-5 基于复合域 $GF((2^2)^4)$的本原多项式

本原多项式	$GF((2^8)$域的根
$y^4+y^2+x^2y+1$	86, 93, 94, 98, *a*3, *a*6, *ae*, *b*2
$y^4+xy^2+x^2y+1$	87, 89, 8*f*, 95, *a*1, *a*6, *ac*, *bf*
$y^4+y^3+x^2y+1$	*fc*, *d*4, *c*3, *b*7, *a*3, *ac*, 58, 39
$y^4+y^3+y^2+x^2y+x^2$	*e*7, *da*, *ce*, *be*, *a*8, 82, *cb*, 2*c*
$y^4+y^3+xy^2+x^2$	38, 75, 81, 94, *aa*, *ce*, *d*8, *e*4
$y^4+y^3+xy^2+y+1$	2*d*, 73, 9*c*, *a*3, *b*4, *c*0, *eb*, *fe*

通过计算最终得到 6 个不可约多项式满足条件，即复合域 $GF((2^2)^4)$共有 6 个本

原多项式，表 3-5 给出了 6 个本原多项式的表达式以及 $GF((2^8)$域不可约多项式分别在此基础上的根。

3.4 最优系数下的乘法逆结构

如式(3-7)(3-8)，基于复合域 $GF((2^2)^4)$的乘法逆结构逻辑资源复杂，占用大量的加法器和乘法器，影响电路的运行速率。因此降低乘法逆结构在片上系统（System on Chip, Soc）的逻辑资源消耗，并提高其运行速率，是一个关键问题。乘法逆结构的逻辑资源占用和本原多项式系数的选择息息相关，不同的本原多项式系数将会导致其加法器和乘法器的数量的差异，因此本节采用最优系数法对基于复合域 $GF((2^2)^4)$的乘法逆结构进行初步化简，以获取相对最优的乘法逆结构。

最优系数法的核心思想是通过选择不可约多项式 $q_1(y)$的系数，简化乘法逆结构中的冗余加法器和乘法器，在保证乘法逆运算正确性的前提下，将已知系数代替未知系数，从而有效优化伽罗瓦域乘法逆运算的复杂度。

由于复合域 $GF((2^2)^4)$是基于其子域 $GF(2^2)$域的运算，利用 $GF(2^2)$域的运算特性，将高次运算转换为低次运算，优化复合域乘法逆结构，降低逻辑资源消耗。如表 3-6 给出了 $GF(2^2)$域运算规则，$\{0, 1, x, x^2\}$为 $GF(2^2)$域的元素。

表 3-6 $GF(2^2)$域运算规则

加法运算规则	乘法器运算规则	
$1+x=x^2$	$x^2=1$	$x^6=1$
$1+x^2=x$	$x^4=x$	
$x+x^2=1$	$x^5=x^2$	

如 3.3 节而言，只有 $q_1(y)$为本原多项式时，所搭建的复合域结构才能满足伽罗域特性，根据 3.3.3 节所述，复合域 $GF((2^2)^4)$共有 6 个本原多项式，分别将这 6 个本原多项式带入复合域乘法逆结构中，并结合 $GF(2^2)$域运算法则进行化简。

表 3-7 基于本原多项式的复合域 $GF((2^2)^4)$乘法逆结构复杂度

本原多项式	乘法器数量	加法器数量
$y^4+y^2+x^2y+1$	55	66
$y^4+xy^2+x^2y+1$	65	66
$y^4+y^3+x^2y+1$	66	67
$y^4+y^3+y^2+x^2y+x^2$	66	67
$y^4+y^3+xy^2+x^2$	66	66
$y^4+y^3+xy^2+y+1$	66	67

表 3-7 给出了复合域 $GF((2^2)^4)$乘法逆结构复杂度，分析得到当$q_1(y)=y^4+y^2+x^2y+1$时，复合域 $GF((2^2)^4)$乘法逆结构中的乘法器和加法器最少，即其逻辑资源占用最低，乘法逆结构最优。则将式(3-14)带入到式(3-8)(3-9)中。

$$\begin{cases}\beta_3=0\\ \beta_2=1\\ \beta_1=x^2\\ \beta_0=1\end{cases} \tag{3-14}$$

最终得到复合域 $GF((2^2)^4)$乘法逆结构中的参数$\{Y_3,\ Y_2,\ Y_1,\ Y_0,\ inv\}$如式(3-15)所示：

$$\begin{cases}
Y_3=r_0^2r_3+r_1^3+r_1r_2^2+r_1r_2r_3x^2+r_2^3x^2+r_2^2r_3+r_2r_3^2x^2+r_3^3x^2\\
Y_2=r_0^2r_2+r_0r_1^2+r_0r_2^2+r_0r_2r_3x^2+r_2^3+r_2r_3^2+r_3^3x^2\\
Y_1=r_0^2r_1+r_0r_1r_3x^2+r_0r_2^2x^2+r_0r_3^2x^2+r_1^3+r_1^2r_3+r_1r_2r_3x^2+r_1r_3^2+r_2^3x^2+r_2^2r_3\\
\quad +r_2r_3^2x^2+r_3^3x\\
Y_0=r_0^3+r_0^2r_3x^2+r_0r_1^2+r_0r_1r_2x^2+r_0r_2r_3x^2+r_0r_3^2x^2+r_1^3x^2+r_1^2r_2+r_1r_2^2x^2\\
\quad +r_1r_2r_3x+r_1r_3^2x^2+r_2^3x^2+r_2^2r_3x^2+r_2r_3^2x+r_3^3\\
inv=r_0+r_0^3r_3x^2+r_0^2r_1^2+r_0^2r_1r_2x^2+r_0^2r_2^2+r_0^2r_2r_3x^2+r_0^2r_3^2x+r_0r_1^3x^2+r_0r_1r_2^2x^2\\
\quad +r_0r_1r_2r_3x+r_0r_2^3x+r_0r_2^2r_3x^2+r_0r_2r_3^2x+r_0r_3^3x+r_1+r_1^2r_2^2+r_1^2r_2r_3x^2+r_1^2r_3^2\\
\quad +r_1r_2^3x^2+r_1r_2r_3^2x^2+r_1r_3^3x+r_2+r_2^2r_3^2+r_2r_3^3x^2+r_3
\end{cases} \tag{3-15}$$

3.5 本章小节

本章主要搭建了基于复合域 $GF((2^2)^4)$的乘法器结构和乘法逆结构。首先，在伽罗瓦域以及复合域的运算法则上搭建了复合域 $GF((2^2)^4)$的乘法逆结构通式和乘法器

结构；其次，分析阐述了可约多项式、不可约多项式以及本原多项式在搭建伽罗瓦域中的重要性；然后，分析并计算了基于复合域 $GF((2^2)^4)$的 167 个可约多项式结构、89 个不可约多项式以及 6 个本原多项式；最后在本原多项式的基础上，利用最优系数法，对复合域 $GF((2^2)^4)$的乘法逆结构进行优化设计。

第 4 章 基于 $GF((2^2)^4)$的 AES S 盒及其优化

本章基于复合域 $GF((2^2)^4)$乘法逆结构搭建了 AES S 盒部件并对其进行优化。首先，采用 DACSE 与 DDBT 算法对复合域 $GF((2^2)^4)$的乘法逆结构进行优化，消除冗余逻辑资源；其次基于最优系数乘法逆结构研究了 8 种映射矩阵及逆映射矩阵，并采用常数矩阵合并法与 CSE 算法对映射矩阵及逆映射矩阵进行优化；最后，在逻辑资源最优的常数矩阵和复合域乘法逆结构上搭建的 AES S 盒结构。

4.1 乘法逆架构优化

本节的优化对象是式(3-15)中的乘法逆结构参数$\{Y_3, Y_2, Y_1, Y_0, inv\}$，目的在于减少乘法逆结构的运算复杂度。本节采用 DACSE 算法针对乘法逆结构进行了两种优化方案：

（1）将 $GF(2^2)$域元素$[x, x^2]$作为常数项进行优化；

（2）将 $GF(2^2)$域元素$[x, x^2]$作为公共项进行优化。

算法 4.1：DACSE 算法优化流程	
	输入算法表达式
step1	将算法表达式中的每一个子项按照一定的元素排序列为一个矩阵，并设置最短路径约束；
step 2	for (i=1;i<n;++i)；
	{
step 3	选取其组合出现频率最大的两列，如：第五列和第七列；向原矩阵中加入 n+1 列，此时 n+1 列代表原两列的积，如： r_2r_3；
step 4	计算 n+1 列代表式的最短路径；
step 5	判断第四步的最短路径是否超过最短路径约束值，若是超过则执行 step6，反之，执行 step7；
step 6	撤销本次公共项消除
step 7	将原操作式中的组合列在原矩阵中设置为 0，并由第 n+1 列替换生成新的矩阵；
	}
step 8	由新矩阵列出优化后的算法逆表达式。

DACSE 算法是一种为了解决 CSE 算法所引起的密码算法延时增大的可能性，

将 CSE 与 DDBT 两种算法相结合的算法。其延优化目的是在消除公共项减少冗余逻辑资源占用的同时，减少密码算法的延时或者最短路径大小。通过设置延时约束，判断选取的公共项延时是否小于延时约束值，从而达到在消除冗余逻辑项，减少电路面积资源消耗的同时，降低算法延时，提高电路运行速率的目的。其算法优化流程如算法 2.1 所示，本文设置的最短路径约束为 2。

DACSE 算法的关键一步在于“提取算法公共项时加入了最短路径约束”，避免以电路性能为代价获得更优的资源消耗。

4.1.1 将[x,x^2]作为常数项优化

基于复合域 $GF((2^2)^4)$的乘法逆结构含有大量的乘法器和加法器，共有 138 个子项，直接查找它们之间的公共项计算量过于庞大。为了简化公式的计算量，将每个子项都作为一个行向量，那么乘法逆结构的 138 个子项可以写作一个 8×138 的矩阵，其中每列所代表的元素排序为：$\{r_0\ \ r_0^2\ \ r_1\ \ r_1^2\ \ r_2\ \ r_2^2\ \ r_3\ \ r_3^2\}$。

采用 DACSE 算法优化后，得到了 51×138 列矩阵，每一列代表着提取多的公共因子，即提取了 43 组公共项，所提取的公共项如附录 3 所示，此时乘法逆结构参数 $\{Y_3, Y_2, Y_1, Y_0, inv\}$式(4-1)所示：

$$\begin{cases} Y_3 = s_5 + s_6 + s_7 + s_{14} + s_1 * x^2 + s_2 * x^2 + s_3 * x^2 + s_8 * x^2 \\ Y_2 = s_1 + s_2 + s_9 + s_{10} + s_{23} + s_{12} * x^2 + s_3 * x^2 \\ Y_1 = s_5 + s_7 + s_{13} + s_{15} + s_{32} + s_1 * x^2 + s_2 * x^2 + s_8 * x^2 + s_{10} * x^2 + s_{11} * x^2 + s_{40} * x^2 \\ \quad + s_3 * x \\ Y_0 = s_3 + s_9 + s_{16} + s_{30} + s_1 * x^2 + s_5 * x^2 + s_6 * x^2 + s_7 * x^2 + s_{11} * x^2 + s_{12} * x^2 \\ \quad + s_{14} * x^2 + s_{15} * x^2 + s_{36} * x^2 + s_2 * x + s_8 * x \\ inv = b_0 + b_2 + b_4 + b_6 + s_{22} + s_{24} + s_{31} + s_{33} + s_{38} + s_{20} * x^2 + s_{21} * x^2 + s_{26} * x^2 \\ \quad + s_{27} * x^2 + s_{34} * x^2 + s_{35} * x^2 + s_{37} * x^2 + s_{39} * x^2 + s_{41} * x^2 + s_{17} * x + s_{18} * x \\ \quad + s_{19} * x + s_{25} * x + s_{29} * x + s_{42} * x \end{cases} \tag{4-1}$$

再利用 CSE 提取式(4-1)中的公共子项，并以新的变量代替公共项，新变量与公共子项间的关系如表 4-1 所示。

表 4-1 提取常数项后新变量与公共子项的关系

$c_0 = s_2 + s_8$	$c_1 = s_1 + s_{11}$	$c_2 = c_0 * x$
$c_3 = s_1 + s_3 + c_0$	$c_4 = c_3 * x^2$	—
$c_5 = s_{12} + s_3$	$c_6 = c_5 * x^2$	—
$c_7 = c_0 + c_1 + s_{10} + s_{40}$	$c_8 = c_7 * x^2$	$c_9 = s_3 * x$
$c_{10} = c_1 + s_5 + s_6 + s_7 + s_{12} + s_{14} + s_{15} + s_{36}$	$c_{11} = c_{10} * x^2$	—
$c_{12} = s_{20} + s_{21} + s_{26} + s_{27} + s_{28} + s_{34} + s_{35} + s_{37} + s_{39} + s_{41}$	$c_{13} = c_{12} * x^2$	—
$c_{14} = s_{17} + s_{18} + s_{19} + s_{25} + s_{29} + s_{42}$	$c_{15} = c_{14} * x$	$c_{16} = s_5 + s_7$

引入新变量后，基于复合域 $GF((2^2)^4)$的乘法逆结构参数$\{Y_3, Y_2, Y_1, Y_0, inv\}$如式(4-2)表示：

$$\begin{cases} Y_3 = s_6 + s_{14} + c_4 + c_{16} \\ Y_2 = s_1 + s_2 + s_9 + s_{10} + s_{23} + c_6 \\ Y_1 = s_{13} + s_{15} + s_{32} + c_8 + c_9 + c_{16} \\ Y_0 = s_3 + s_9 + s_{16} + s_{30} + c_{11} + c_2 \\ inv = b_0 + b_2 + b_4 + b_6 + s_{22} + s_{24} + s_{31} + s_{33} + s_{38} + c_{13} + c_{15} \end{cases} \tag{4-2}$$

4.1.2 将$[x,x^2]$作为公共项优化

将 $GF(2^2)$域元素$[x, x^2]$作为公共项进行优化时，此时的乘法逆结构所构建的矩阵的维度是 10×138，其中每列所代表的元素排序为：$\{x \ \ x^2 \ \ r_0 \ \ r_0^2 \ \ r_1 \ \ r_1^2 \ \ r_2 \ \ r_2^2 \ \ r_3 \ \ r_3^2\}$。采用 DACSE 算法对其进行优化后得到维度为 73×138 的矩阵，即提取公共项 62 组。优化后的基于复合域 $GF((2^2)^4)$的乘法逆结构如式(4-3)所示，所提取的公共项如附录 4 所示。

$$\begin{cases} Y_3 = s_5 + s_7 + s_9 + s_{10} + s_{12} + s_{14} + s_{18} + s_{19} \\ Y_2 = s_{14} + s_{15} + s_{16} + s_{29} + s_{34} + s_{45} + s_{46} \\ Y_1 = s_5 + s_7 + s_{12} + s_{13} + s_{17} + s_{18} + s_{19} + s_{20} + s_{32} + s_{42} + s_{48} + s_{59} \\ Y_0 = s_3 + s_{12} + s_{15} + s_{16} + s_{20} + s_{23} + s_{24} + s_{25} + s_{26} + s_{27} + s_{28} + s_{30} + s_{40} \\ \qquad + s_{52} + s_{60} \\ inv = b_2 + b_4 + b_6 + b_8 + s_{31} + s_{33} + s_{35} + s_{37} + s_{38} + s_{39} + s_{41} + s_{43} + s_{44} \\ \qquad + s_{47} + s_{49} + s_{50} + s_{51} + s_{53} + s_{54} + s_{55} + s_{56} + s_{57} + s_{58} + s_{61} + s_{62} \end{cases} \tag{4-3}$$

4.1.3 优化结果对比

采用 DDBT 算法构建上述两种复合域 $GF((2^2)^4)$乘法逆结构的最短路径，即迭代

选择最小延时变量来构建一个逻辑网络，调整节点的延时分布，从而实现优化目的的。通过对变量延时大小排序、延时路径选择、插入新延时变量等关键步骤，降低密码算法的延时大小和最短路径，其构建流程如下所示。

算法 4.2：DDBT 算法优化流程	
	输入算法表达式
step1	将所有输入变量的最短路径构建为一个集合 C 并从小到大进行排列：$\{C_0, C_1, C_2, \ldots, C_n\}$；
step 2	for (i=1;i<n;++i)；
	{
step 3	在输入变量集合中选择最短路径最小的两个变量如：C_0, C_1；
step 4	设定：$C_{new}=S_0 \oplus S_1$，计算新变量 C_{new} 的最短路径，并取代集合 C 中的$\{C_0, C_1\}$，此时新集合可表示为：$\{C_{new}, C_2, \ldots, C_n\}$；
	}

结果表明，当$[x, x^2]$作为常数项进行优化时，其最短路径为 15 个乘法器，17 个加法器。如式(4-4)所示，其中 T_{AND}，T_{XOR} 表示最短路径乘法器和加法器：

$$T_1 = 15T_{AND} + 17T_{XOR} \tag{4-4}$$

当$[x, x^2]$作为公共项进行优化时，其最短路径为 15 个乘法器，16 个加法器。如式(4-5)所示：

$$T_2 = 15T_{AND} + 16T_{XOR} \tag{4-5}$$

对比可得，当$[x, x^2]$作为公共项进行优化时，具备更有的延时。

4.2 S 盒架构

基于 $GF(2^8)$域的 S 盒结构如图 4-1 所示，其运算流程如下：输入的 8bit 数据 x，首先经过一个基于 $GF(2^8)$域的乘法逆结构，得到 8bit 的乘法逆元 x^{-1}；然后再对该乘法逆元 x^{-1} 进行仿射运算，仿射运算包括了 x^{-1} 和仿射矩阵之间的乘法运算，和乘法运算结果与一个 8bit 的常数列向量的加法运算；最终输出结果 y 的字节是 8bit。

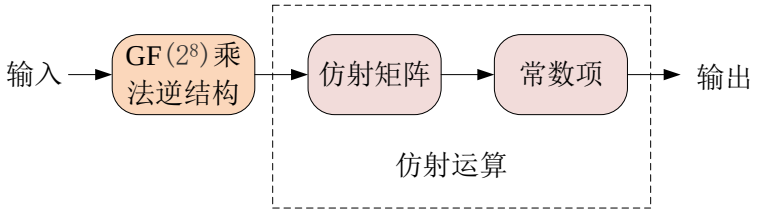


图 4-1 基于 $GF(2^8)$域的 S 盒架构

S 盒结构包括了两个关键部件：“乘法逆结构”和“仿射运算”，仿射运算包括了一个仿射矩阵 M 和一个常数项 C。仿射矩阵 M 是一个常数矩阵，其表达式如式(4-6)所示：

$$M=\begin{bmatrix}1&0&0&0&1&1&1&1\\1&1&0&0&0&1&1&1\\1&1&1&0&0&0&1&1\\1&1&1&1&0&0&0&1\\1&1&1&1&1&0&0&0\\0&1&1&1&1&1&0&0\\0&0&1&1&1&1&1&0\\0&0&0&1&1&1&1&1\end{bmatrix} \tag{4-6}$$

仿射运算中常数列向量 C 的矩阵表达如式(4-7)所示：

$$C=(0,1,1,0,0,0,1,1)^T \tag{4-7}$$

则上所述 S 盒的运算表达式如下：

$$y=Mx^{-1}\oplus C \tag{4-8}$$

如第三章节所述，基于 $GF(2^8)$域所构建的乘法逆结构计算量大，计算复杂，本文采用同构映射法将 $GF(2^8)$域乘法逆结构映射到复合域 $GF((2^2)^4)$乘法逆结构，降低了算法运算的复杂性，提高了运算速率。其映射过程如图 4-2 所示，输入的 8bit 数据首先通过一个映射矩阵，映射到复合域 $GF((2^2)^4)$进行乘法逆运算，得到了输入数据 x 的乘法逆元 x^{-1}，这一过程中，输入数据由一个 8bit 字节转化为了 4 个 2bit 字节，这大幅度的降低了乘法逆运算的复杂性，提升了算法的运算速率；然后再通过一个逆映射矩阵将乘法逆元 x^{-1} 映射回 $GF(2^8)$域。

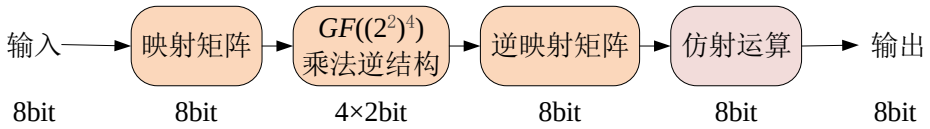


图 4-2 基于复合 $GF((2^2)^4)$的 S 盒架构

基于复合域 $GF((2^2)^4)$的 S 盒结构可由式(4-9)所表示：

$$y = M(\delta^{-1}(\delta X^T)^{-1}) + C \tag{4-9}$$

其中 δ 为映射矩阵，δ^{-1} 为逆映射矩阵，X 是输入的 8bit 数据 x 从低位到高位排列的一个 1×8 的向量。假设输入数据 $x = (x_7x_6x_5x_4x_3x_2x_1x_0)_2$，则 X 为：

$$X = \begin{bmatrix} x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \end{bmatrix} \tag{4-10}$$

4.3 常数矩阵

基于复合域 $GF((2^2)^4)$的 S 盒结构中的常数矩阵包括了三个维度为 8×8 的二进制矩阵：映射矩阵、逆映射矩阵和仿射矩阵。仿射矩阵是加密过程中对乘法逆元进行的一个线性变换，其目的是增加 AES 算法的非线性性和混淆性，从而提高加密算法的安全性。

对于式(4-9)所示的复合域 S 盒表达式，本节利用常数矩阵合并法，通过将共享同一个输入端的若干矩阵合并成一个规模更大的常数矩阵，即将多个常数矩阵合并为一个矩阵，从而减少逻辑资源消耗和缩短电路的关键路径，获得更优的电路性能。图 4-3 给出了本文所完成的三个常数合并矩阵。

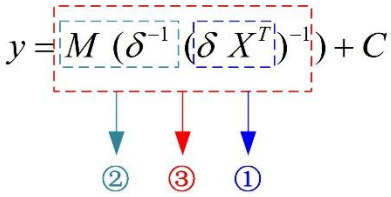


图 4-3 基于复合域的 S 盒的常数矩阵合并

（1）映射矩阵 δ 与输入向量 X 间的合并：映射矩阵 δ 的维度为 8×8，输入向量 X 的维度为 8×1，两者合并得到一个维度为 8×1 的矩阵 H_1，如式(4-11)所示：

$$H_1 = \delta X^T \tag{4-11}$$

（2）仿射矩阵 M 与逆映射矩阵 δ^{-1} 间的合并：仿射矩阵 M 的维度为 8×8，逆映射矩阵 δ^{-1} 的维度为 8×8，两者合并得到一个维度为 8×8 的矩阵 H_2，如式(4-12)所示：

$$H_2 = M\delta^{-1} \tag{4-12}$$

（3）合并 H_1 的逆与合并矩阵 H_2 间的合并：合并 H_1^{-1} 的维度为 8×1，合并矩阵 H_2 的维度为 8×8，两者合并得到一个维度为 8×1 的矩阵 H_3，如式(4-13)所示：

$$H_3 = H_2 H_1^{-1} \tag{4-13}$$

4.3.1 映射矩阵

映射矩阵是反映 $GF(2^8)$域元素 x 映射到复合域 $GF((2^2)^4)$映射关系的矩阵，假设 $GF(2^8)$域元素 x 映射到复合域 $GF((2^2)^4)$中的元素是 a，其映射关系如式(4-14)所示，其中 A 是元素 a 的向量表示：

$$\delta X^T = A^T \tag{4-14}$$

映射矩阵的结构依赖于伽罗瓦域和复合域的不可约多项式的选择，其推导流程如图 4-4 所示。

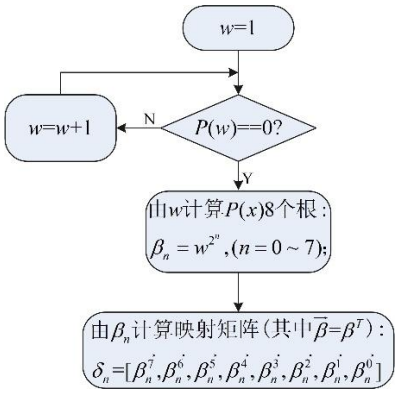


图 4-4 映射矩阵推导流程

在复合域 $GF((2^2)^4)$的运算基础上，查找 $GF(2^8)$域的不可与多项式 $P(x) = x^8 + x^4 + x^3 + x + 1$的八个根，对其中一个根 β 进行幂运算，β 的幂所构成的向量矩阵

则组成了映射矩阵。根据 2.1.7 节共轭的定理知，不可约多项式的根是一组共轭根，则根据公式(4-15)可以由一个根找出该不可约多项式的其余 7 个根：

$$\beta_n = w^{2^n}, (n = 0 \sim 7); \tag{4-15}$$

同理，若w不是不可与多项式$P(x)$=0的一个根，则其同一共轭类中的其余7个元素也不是$P(x)$=0的根。因此映射矩阵推导流程图可以改进为图4-5，只需计算$P(x)$的一个根就可以计算出所有根，这大大的增加了查找根的运算速率。

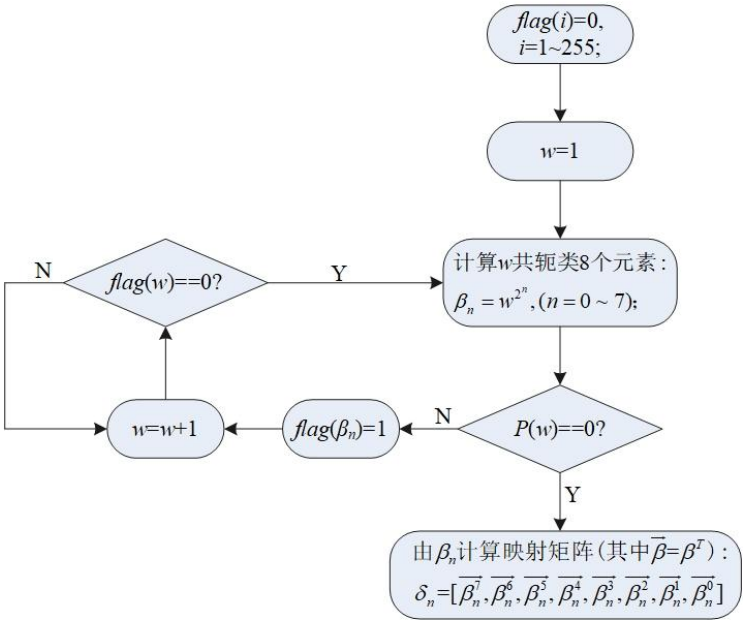


图 4-5 改进映射矩阵推导流程

在 3.4 节所查找的基于本原多项式$q(y) = y^4 + y^2 + x^2 y + 1$时，复合域 $GF((2^2)^4)$具有结构最优的乘法逆结构。在此本原多项式下，$P(x)$=0 的八个根表示为：

$$\begin{cases} \beta_0 = (86)_{16} = (10000110)_2 \\ \beta_1 = (98)_{16} = (10011000)_2 \\ \beta_2 = (a6)_{16} = (10100110)_2 \\ \beta_3 = (a3)_{16} = (10100011)_2 \\ \beta_4 = (b2)_{16} = (10110010)_2 \\ \beta_5 = (ae)_{16} = (10101110)_2 \\ \beta_6 = (93)_{16} = (10010011)_2 \\ \beta_7 = (94)_{16} = (10010100)_2 \end{cases} \tag{4-16}$$

据上述映射矩阵推导流程，对这八个根分别进行幂运算，搭建映射矩阵结构，对着八个根所进行的幂运算结果的 16 进制表示如表 4-2 所示。

表 4-2 不可约多项式的根的幂运算的 16 进制表示

	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7
β_n^0	$(01)_{16}$	$(01)_{16}$	$(01)_{16}$	$(01)_{16}$	$(01)_{16}$	$(01)_{16}$	$(01)_{16}$	$(01)_{16}$
β_n^1	$(86)_{16}$	$(98)_{16}$	$(a6)_{16}$	$(a3)_{16}$	$(b2)_{16}$	$(ae)_{16}$	$(93)_{16}$	$(94)_{16}$
β_n^2	$(98)_{16}$	$(a6)_{16}$	$(a3)_{16}$	$(b2)_{16}$	$(ae)_{16}$	$(93)_{16}$	$(94)_{16}$	$(86)_{16}$
β_n^3	$(82)_{16}$	$(88)_{16}$	$(b2)_{16}$	$(9f)_{16}$	$(b4)_{16}$	$(bd)_{16}$	$(8c)_{16}$	$(ab)_{16}$
β_n^4	$(a6)_{16}$	$(a3)_{16}$	$(c3)_{16}$	$(ae)_{16}$	$(93)_{16}$	$(94)_{16}$	$(86)_{16}$	$(98)_{16}$
β_n^5	$(c4)_{16}$	$(56)_{16}$	$(9f)_{16}$	$(44)_{16}$	$(dd)_{16}$	$(7a)_{16}$	$(d8)_{16}$	$(6b)_{16}$
β_n^6	$(88)_{16}$	$(bb)_{16}$	$(86)_{16}$	$(b4)_{16}$	$(bd)_{16}$	$(8c)_{16}$	$(ab)_{16}$	$(82)_{16}$
β_n^7	$(7a)_{16}$	$(d8)_{16}$	$(6b)_{16}$	$(c4)_{16}$	$(56)_{16}$	$(c3)_{16}$	$(44)_{16}$	$(dd)_{16}$

通过将表 4-2 所示的 16 进制幂运算结果转化为二进制表示，从低位到高位，以 $[\beta_n^0\ \beta_n^1\ \beta_n^2\ \beta_n^3\ \beta_n^4\ \beta_n^5\ \beta_n^6\ \beta_n^7]^T$ 形式构建了 8 个映射矩阵。

4.3.2 逆映射矩阵

在 AES S 盒结构中逆映射矩阵是反映复合域 $GF((2^2)^4)$的乘法逆结构映射至 $GF(2^8)$域的映射关系的矩阵，其映射关系由式(4-17)表示：

$$\delta^{-1}\left(A^T\right)^{-1}=\left(X^T\right)^{-1} \tag{4-17}$$

逆映射矩阵的推导与映射矩阵息息相关，是映射矩阵的逆。对逆映射矩阵的推导有以下两种方法：

（1）基于伽罗瓦域运算法则，对映射矩阵逐行逐列进行求逆运算；

（2）对映射矩阵 δ 求逆，并对其逆进行(module 2)运算。

逆映射矩阵与映射矩阵的关系如式(4-18)：

$$\delta*\delta^{-1}=E \tag{4-18}$$

对表 4-2 所构建的 8 个映射矩阵分别求逆映射矩阵，假设其逆映射矩阵 $\delta_n^{-1}=\left[\alpha_{n,0},\alpha_{n,1},\alpha_{n,2},\alpha_{n,3},\alpha_{n,4},\alpha_{n,5},\alpha_{n,6},\alpha_{n,7}\right]$，则这 8 个逆映射矩阵行向量的 16 进制表示如表 4-3 所示。

表 4-3 逆映射矩阵行向量的 16 进制表示

	δ_0^{-1}	δ_1^{-1}	δ_2^{-1}	δ_3^{-1}	δ_4^{-1}	δ_5^{-1}	δ_6^{-1}	δ_7^{-1}
$\alpha_{n,0}$	$(80)_{16}$	$(e0)_{16}$	$(be)_{16}$	$(ea)_{16}$	$(9e)_{16}$	$(c4)_{16}$	$(a3)_{16}$	$(cf)_{16}$
$\alpha_{n,1}$	$(26)_{16}$	$(1f)_{16}$	$(27)_{16}$	$(1c)_{16}$	$(26)_{16}$	$(1f)_{16}$	$(27)_{16}$	$(1c)_{16}$
$\alpha_{n,2}$	$(5b)_{16}$	$(64)_{16}$	$(6d)_{16}$	$(4c)_{16}$	$(61)_{16}$	$(5d)_{16}$	$(55)_{16}$	$(77)_{16}$
$\alpha_{n,3}$	$(62)_{16}$	$(5c)_{16}$	$(56)_{16}$	$(76)_{16}$	$(58)_{16}$	$(65)_{16}$	$(6e)_{16}$	$(4d)_{16}$
$\alpha_{n,4}$	$(57)_{16}$	$(75)_{16}$	$(59)_{16}$	$(66)_{16}$	$(6f)_{16}$	$(4e)_{16}$	$(63)_{16}$	$(5f)_{16}$
$\alpha_{n,5}$	$(51)_{16}$	$(44)_{16}$	$(43)_{16}$	$(71)_{16}$	$(6a)_{16}$	$(7e)_{16}$	$(7a)_{16}$	$(49)_{16}$
$\alpha_{n,6}$	$(18)_{16}$	$(15)_{16}$	$(07)_{16}$	$(32)_{16}$	$(1b)_{16}$	$(14)_{16}$	$(04)_{16}$	$(33)_{16}$
$\alpha_{n,7}$	$(53)_{16}$	$(46)_{16}$	$(41)_{16}$	$(73)_{16}$	$(68)_{16}$	$(7c)_{16}$	$(78)_{16}$	$(4b)_{16}$

4.3.3 常数矩阵优化结果

对 4.3.2 节所推导的 8 个逆映射矩阵与仿射矩阵 M 进行合并，假设合并矩阵为 $H_2=\left[h_0^{'},h_1^{'},h_2^{'},h_3^{'},h_4^{'},h_5^{'},h_6^{'},h_7^{'}\right]$，则根据式(4-12)在 8 个不同的逆映射矩阵基础上所得到的合并矩阵 H_2 的行向量 16 进制表示如表 4-4 所示。

表 4-4 合并矩阵 H_2 行向量的 16 进制表示

	$M\delta_0^{-1}$	$M\delta_1^{-1}$	$M\delta_2^{-1}$	$M\delta_3^{-1}$	$M\delta_4^{-1}$	$M\delta_5^{-1}$	$M\delta_6^{-1}$	$M\delta_7^{-1}$
$h_0^{'}$	$(cd)_{16}$	$(82)_{16}$	$(e2)_{16}$	$(bc)_{16}$	$(e8)_{16}$	$(9c)_{16}$	$(c6)_{16}$	$(a1)_{16}$
$h_1^{'}$	$(bc)_{16}$	$(e8)_{16}$	$(9c)_{16}$	$(c6)_{16}$	$(a1)_{16}$	$(cd)_{16}$	$(82)_{16}$	$(e2)_{16}$
$h_2^{'}$	$(b6)_{16}$	$(c8)_{16}$	$(b2)_{16}$	$(f6)_{16}$	$(aa)_{16}$	$(ee)_{16}$	$(ad)_{16}$	$(dc)_{16}$
$h_3^{'}$	$(cc)_{16}$	$(81)_{16}$	$(e3)_{16}$	$(bf)_{16}$	$(e9)_{16}$	$(9f)_{16}$	$(c7)_{16}$	$(a2)_{16}$
$h_4^{'}$	$(c8)_{16}$	$(b2)_{16}$	$(fb)_{16}$	$(aa)_{16}$	$(ee)_{16}$	$(ad)_{16}$	$(dc)_{16}$	$(b6)_{16}$
$h_5^{'}$	$(19)_{16}$	$(16)_{16}$	$(06)_{16}$	$(31)_{16}$	$(1a)_{16}$	$(17)_{16}$	$(05)_{16}$	$(30)_{16}$
$h_6^{'}$	$(27)_{16}$	$(1c)_{16}$	$(26)_{16}$	$(1f)_{16}$	$(27)_{16}$	$(1c)_{16}$	$(26)_{16}$	$(1f)_{16}$
$h_7^{'}$	$(2f)_{16}$	$(3e)_{16}$	$(0a)_{16}$	$(20)_{16}$	$(2e)_{16}$	$(3d)_{16}$	$(0b)_{16}$	$(23)_{16}$

首先，基于常数矩阵合并法对映射矩阵进行合并得到合并矩阵 H_1，如式(4-11)，并基于复合域运算法则求其逆 H_1^{-1}；其次，基于常数矩阵合并法将合并矩阵 H_2 与 H_1^{-1} 进行合并得到合并矩阵 H_3，如式(4-13)；最后，采用 CSE 消除合并矩阵 H_3 行向量间的公共子项。对合并矩阵 H_1, H_3 进行公共项消除前后的复杂度对比如表 4-5 所示。

表 4-5 常数矩阵合并 H_1, H_3 优化前后复杂度对比

	合并矩阵	β_0	β_1	β_2	β_3	β_4	β_5	β_6	β_7
优化前加法器个数	H_1	15	20	26	21	26	23	17	20
	H_3	26	18	23	28	25	30	30	22
	总计	**41**	**38**	**49**	**49**	**51**	**53**	**37**	**42**
优化后加法器个数	H_1	12	13	15	14	17	15	13	12
	H_3	16	12	12	16	16	16	14	14
	总计	**28**	**25**	**27**	**30**	**33**	**31**	**27**	**26**
—	优化率	31.7%	34.2%	44.9%	38.8%	35.3%	41.5%	27.0%	38.1%

由此可知在根 β_2 的基础上搭建的常数矩阵优化率最高，达到了44.9%，在根 β_1 的基础上搭建的常数矩阵优化后的复杂度最低。本文所搭建的S盒结构以最大限度的减少电路面积占用为优先考虑，因此本文选择在根 β_1 的基础上搭建的常数矩阵。

4.4 本章小节

本章使用了 CSE、DDBT、DACSE 以及常数矩阵合并法对 AES S 盒中的常数矩阵和乘法逆部件进行优化，针对常数矩阵部件的优化率最高达到 44.9%；针对乘法逆部件的采取的不同优化方法，结构最优的乘法逆结构提取了 62 组公共项，得到的最短延时为 $15T_{AND}+16T_{XOR}$。最终，本文选用资源消耗最低的一组常数矩阵和复合域 $GF((2^2)^4)$乘法逆结构搭建 AES S 盒。

第 5 章 AES S 盒实现与电路综合评估

本章针对第四章节所搭建的AES S盒结构进行实现流程阐述以及电路综合评估结果分析。首先，详细阐述了 AES S 盒在复合域上的元素转换、常数矩阵实现细则以及逻辑电路实现；然后，采用 Synopsys™ 集成电路 IC 综合工具 DC，分别基于 SMIC 180nm 和 TSMC 28nm 工艺库对所搭建的 AES S 盒结构的面积以及延时进行综合分析，评估其在不同工艺库下面积和时序优化率，以验证优化策略在设计目标下的优化率。

5.1 AES S 盒实现

5.1.1 元素映射关系

在 AES 中 $GF(2^8)$域元素 g 是 8 字节元素。基于多项式基$\{A^7, A^6, A^5, A^4, A^3, A^2, A^1, A^0\}$ $GF(2^8)$域元素 g 可表示为：$g = g_7A^7 + g_6A^6 + g_5A^5 + g_4A^4 + g_3A^3 + g_2A^2 + g_1A^1 + g_0A^0$，其中 $g_i \in GF(2)$。

复合域 $GF((2^2)^4)$元素 d 在多项式基$\{Y^3, Y^2, Y^1, Y^0\}$的基础上可表示为：$d = \lambda_3Y^3 + \lambda_2Y^2 + \lambda_1Y^1 + \lambda_0Y^0$，其中$\{\lambda_3, \lambda_2, \lambda_1, \lambda_0\} \in GF(2^2)$，因此系数可表示为 $\lambda = \eta_1Z + \eta_0$，$\{\eta_1, \eta_0\} \in GF(2)$。对于元素 g 从 $GF(2^8)$域映射到复合域 $GF((2^2)^4)$，使用复合域新字节$\{b_7, b_6, b_5, b_4, b_3, b_2, b_1, b_0\}$表示 $GF(2^8)$域元素 g，其关系见式(5-1)：

$$\begin{aligned} & g_7A^7 + g_6A^6 + g_5A^5 + g_4A^4 + g_3A^3 + g_2A^2 + \ g_1A^1 + g_0A^0 \\ & = (b_7Z + b_6)Y^3 + (b_5Z + b_4)Y^2 + (b_3Z + b_2)Y^1 + (b_1Z + b_0)Y^0 \\ & = b_7ZY^3 + b_6Y^3 + b_5ZY^2 + b_4Y^2 + b_3ZY^1 + b_2Y^1 + b_1ZY^0 + b_0Y^0 \end{aligned} \tag{5-1}$$

而上式中的$\{ZY^3, Y^3, ZY^2, Y^2, ZY^1, Y^1, ZY^0, Y^0\}$可用映射矩阵的列表示。根据 4.3 节的常数矩阵复杂度对比结果，本文选用复杂度最低的映射矩阵 δ_1，元素映射关系如式(5-2)所示：

$$\begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \end{bmatrix} \tag{5-2}$$

逆映射关系如式(5-3)所示：

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \end{bmatrix} \tag{5-3}$$

5.1.2 常数矩阵合并细节

按照4.3节所给出的三种常数矩阵合并方法，针对复杂度最低的一组常数矩阵，对映射矩阵δ_1、逆映射矩阵δ_1^{-1}以及仿射矩阵M所进行的常数矩阵合并如下。

（1）映射矩阵与输入向量X间的合并所得到的合并矩阵H_1如式(5-4)：

$$H_1 = \delta_1 X^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \end{bmatrix} \times \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} x_0 + x_4 + x_6 \\ x_2 + x_4 + x_5 + x_6 \\ x_2 + x_5 \\ x_1 + x_3 + x_6 + x_7 \\ x_1 + x_5 + x_6 + x_7 \\ x_2 + x_4 + x_6 \\ x_5 + x_7 \\ x_1 + x_2 + x_3 + x_4 + x_6 + x_7 \end{bmatrix} \tag{5-4}$$

采用 CSE 算法对合并矩阵 H_1 的行向量进行优化，共提取了 3 个公共子项，减少了 7 个加法器的逻辑资源占用，所提取的公共子项如式(5-5)所示，优化后的矩阵

$H_1^{'}$如式(5-6)所示：

$$\begin{cases} \alpha_0 = x_4 + x_6 \\ \alpha_1 = x_1 + x_6 + x_7 \\ \alpha_3 = x_2 + x_5 \end{cases} \tag{5-5}$$

$$H_1^{'} = \begin{bmatrix} \alpha_0 + x_0 \\ \alpha_2 + x_0 \\ \alpha_2 \\ \alpha_1 + x_3 \\ \alpha_1 + x_5 \\ \alpha_0 + x_2 \\ x_5 + x_7 \\ \alpha_1 + x_2 + x_3 + x_4 \end{bmatrix} \tag{5-6}$$

（2）逆映射矩阵δ_1^{-1}与仿射矩阵 M 合并所得到的合并矩阵 H_2 结构如式(5-7)所示：

$$H_2 = M\delta_1^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{5-7}$$

（3）假设合并矩阵$H_1^{'}$的逆为：

$$H_1^{'-1} = A^T = \left[a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7\right]^T \tag{5-8}$$

则将优化后的合并矩阵$H_1^{'}$的逆与 H_2 相结合得到的合并矩阵 H_3 的表达式如式（5-9）所示：

$$H_3 = H_2 H_1^{'-1} = H_2 A^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = \begin{bmatrix} a_0 + a_6 \\ a_0 + a_1 + a_2 + a_4 \\ a_0 + a_1 + a_4 \\ a_0 + a_7 \\ a_0 + a_2 + a_3 + a_6 \\ a_3 + a_5 + a_6 \\ a_3 + a_4 + a_5 \\ a_2 + a_3 + a_4 + a_5 + a_6 \end{bmatrix} \quad (5\text{-}9)$$

采用 CSE 算法对合并矩阵 H_3 的行向量进行优化，共提取了 4 个公共子项，减少了 6 个加法器的逻辑资源占用，所提取的公共子项如式(5-10)所示，优化后的矩阵 $H_3^{'}$ 如式(5-11)所示：

$$\begin{cases} \partial_0 = a_3 + a_5 \\ \partial_1 = a_0 + a_1 + a_4 \\ \partial_2 = a_0 + a_6 \\ \partial_3 = \partial_0 + a_4 \end{cases} \quad (5\text{-}10)$$

$$H_3^{'} = \begin{bmatrix} \partial_2 \\ \partial_1 + a_2 \\ \partial_1 \\ a_0 + a_7 \\ \partial_2 + a_2 + a_3 \\ \partial_0 + a_6 \\ \partial_0 + a_4 \\ \partial_3 + a_2 + a_6 \end{bmatrix} \quad (5\text{-}11)$$

5.1.3 S 盒实现细节

将加密数据所构成的字符串，按照 8 个字节一组划分作为 S 盒输入数据，记作 x，以第一组数据为例，将输入数据 x 从低位到高位搭建一个 8×1 列向量，记作 X^T，针对输入数据的 S 盒具体实现见图 5-1。

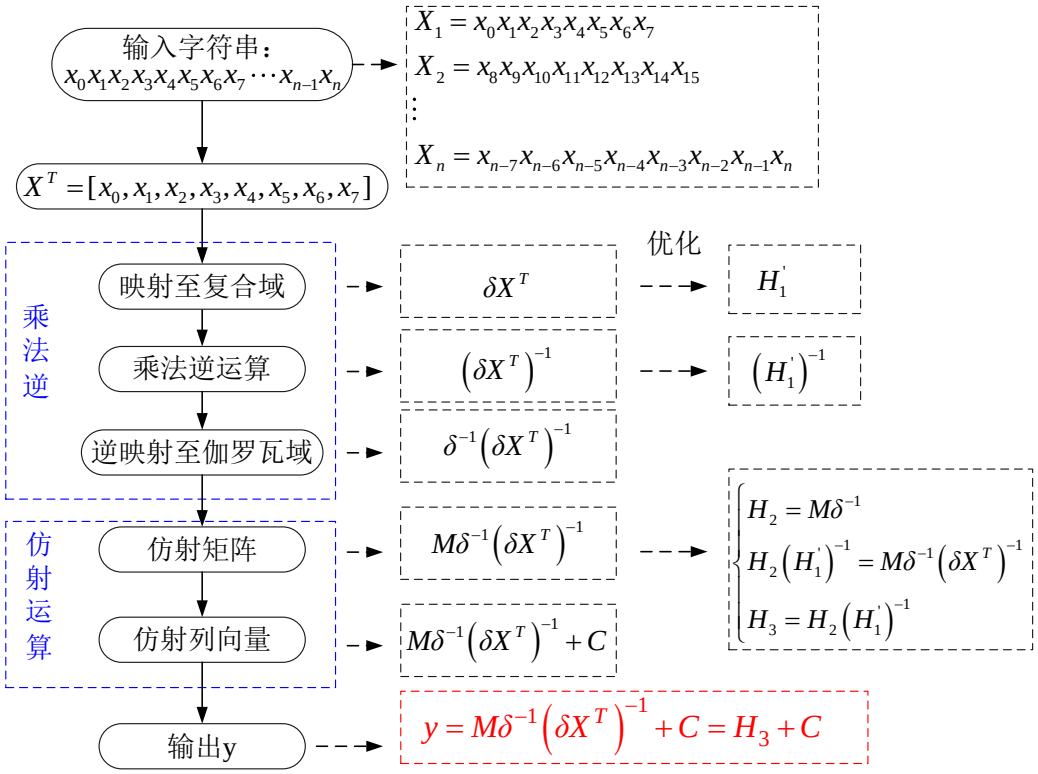


图 5-1 S 盒的具体实现示意图

第一步：将输入数据映射到复合域 $GF((2^2)^4)$域。为了计算基于复合域 $GF((2^2)^4)$ 的乘法逆结构，需要将输入向量 X^T 映射转化为复合域元素向量，即与映射矩阵相乘 δX^T，这一步骤在常数矩阵合并中进行了合并与优化，即 H_1'，公式(5-6)给出了其矩阵；

第二步：对复合域元素求乘法逆元：对 H_1' 向量进行求逆操作，即根据公式(4-15)(4-16)推导其乘法逆结构，记作 $A^T = (H_1')^{-1} = (\delta_1 X^T)^{-1}$；

第三步：将复合域乘法逆结构 A^T 与逆映射矩阵相乘，并映射至 $GF(2^8)$域，记作 $\delta_1^{-1}A^T = \delta_1^{-1}(H_1')^{-1} = \delta_1^{-1}(\delta_1 X^T)^{-1}$；

第四步：对乘法逆结构进行仿射运算。首先是仿射矩阵与乘法逆结构相乘，即 $M\delta_1^{-1}A^T$，在 5.1.2 节中逆映射矩阵 δ_1^{-1} 与仿射矩阵 M 已完成合并，记作 H_2；根据本文所研究的常数矩阵合并，得 $H_3 = H_2A^T = M\delta_1^{-1}(\delta_1 X^T)^{-1}$，采用 CSE 算法优化后得到的矩阵为 H_3'，公式(5-11)给出了其矩阵实现；

其次是，H_3' 与仿射运算的常数项的加法运算，即异或一个列向量，公式表示

为 $y = M(\delta^{-1}(\delta X^T)^{-1}) + C = H_3^{'} + C$，其具体实现见公式(5-12)。

$$y = H_3^{'} + C = \begin{bmatrix} \partial_2 \\ \partial_1 + a_2 \\ \partial_1 \\ a_0 + a_7 \\ \partial_2 + a_2 + a_3 \\ \partial_0 + a_6 \\ \partial_0 + a_4 \\ \partial_3 + a_2 + a_6 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \partial_2 \\ \partial_1 + a_2 + 1 \\ \partial_1 + 1 \\ a_0 + a_7 \\ \partial_2 + a_2 + a_3 \\ \partial_0 + a_6 \\ \partial_0 + a_4 + 1 \\ \partial_3 + a_2 + a_6 + 1 \end{bmatrix} \tag{5-12}$$

5.2 Synopsys 集成电路综合评估

Synopsys DC 是 Synopsys 公司推出的一款将 RTL（Register Transfer Level）设计转换为门级网表（Gate-level Netlist）的工具。DC 通过强大的逻辑优化能力、全面的设计约束支持以及多种工艺兼容性，成为现代集成电路设计中不可或缺的工具。Design Compiler 的核心功能是逻辑综合和优化，将 Verilog 或 VHDL 等硬件描述语言编写的 RTL 代码转化为优化的门级网表，此综合过程可以看作先将 Verilog 或 VHDL 描述转化为库元件组成的逻辑电路[69]；其次，对逻辑电路进行冗余逻辑消除以减少资源消耗，重组逻辑电路以获得最短关键路径，进行功耗优化降低静态功耗和动态功耗，在此逻辑优化过程中 DC 工具会根据用户所设定的延时、面积等参数对门级网表进行进一步的优化；最后，DC 将逻辑优化完成后的门级网表映射到目标工艺库的具体标准单元中。

Design Compiler 的综合流程如图 5-2 所示，其具体综合步骤如下：

（1）建立 RTL 工程：编写 RTL 源代码，并对其进行前仿真后用于综合；

（2）定义库(Specify Libraries)：设定好所用到的综合库；

（3）读取设计(Read Design)：导入 RTL 代码，并进行读取和分析；

（4）定义设计环境(Define Design Environment)：设置设计的工作环境，如设计连线负载模型、设置驱动单元的特性、设置设计的负载等；

（5）设置设计约束(Set Design Constraints)：正确且合理的约束设置是获得准确综合结果的关键，主要包括时钟约束和 I/O 接口的相关约束设置，例如定义时钟频

率、输入输出延时及时序要求；

（6）优化设计(Optimize the Design)：使用综合工具（如 compile）对设计进行优化，将高层描述转化为门级网表，同时满足性能和面积等约束；

（7）分析与解决设计问题(Analyze and Resolve the Design Problem)：根据报告结果对设计进行的分析，并调试；

（8）设计数据的存储：综合设计完成后，将生成的设计数据（如网表文件、约束文件和时序报告）进行保存，以供后续的布局布线使用。

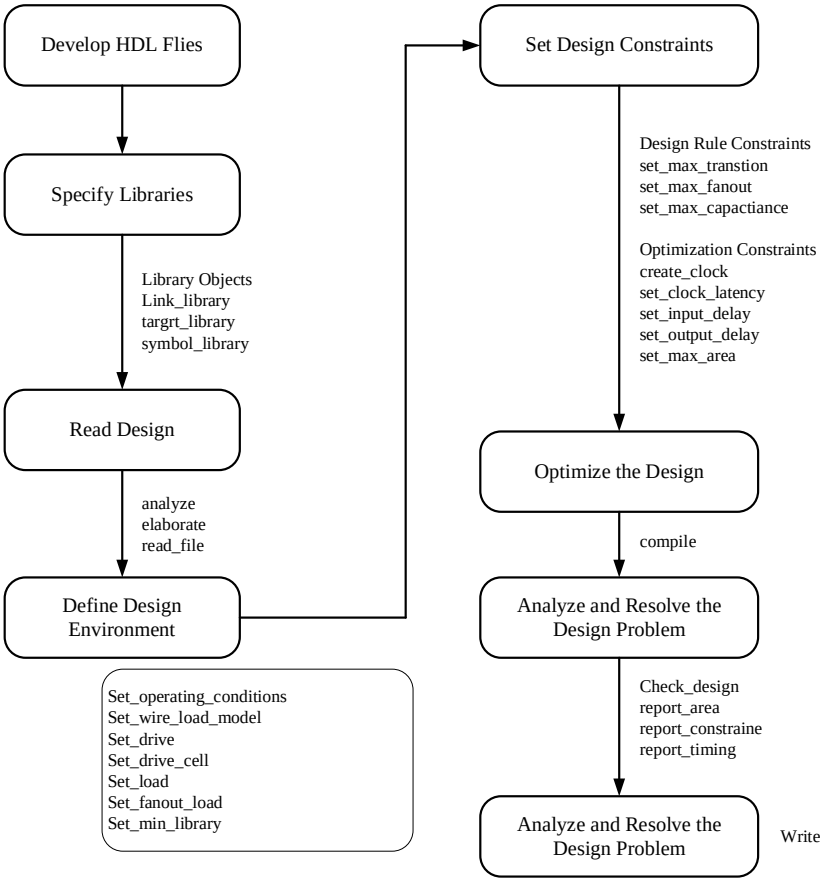


图 5-2 Design Compiler 的综合流程

针对本文所搭建的基于复合域 $GF((2^2)^4)$的 AES S 盒结构，基于 SMIC 180nm 和 TSMC 28nm 工艺库对所搭建的 AES S 盒结构的电路面积以及延时进行综合评估，对比分析在不同的延时约束下，复合域乘法逆结构优化后与直接实现复合域 $GF((2^2)^4)$ S 盒结构的面积和延时优化率。

5.2.1 基于 SMIC 180nm 工艺库最小延时约束下 AES S 盒优化评估

基于 SMIC 180nm 工艺库最小延时约束，对直接实现基于复合域 $GF((2^2)^4)$的 AES S 盒的电路面积评估如图 5-3(*a*)所示；延时评估如图 5-3(*b*)所示。

Hierarchical cell	Global cell area: Absolute Total	Global cell area: Percent Total	Local cell area: Combinational	Local cell area: Noncombinational	Local cell area: Black boxes	Design
subBytes_GF224_pm1	37508.3398	100.0	36.5904	0.0000	0.0000	subBytes_GF224_pm1
affine_inv	661.9536	1.8	661.9536	0.0000	0.0000	AffineDeltaInver_A1
map_inv	35771.9414	95.4	6429.9268	0.0000	0.0000	GF224_inverse_p
map_multi	1037.8369	2.8	1037.8369	0.0000	0.0000	mappMultiplier_M1
Total		37508.4805	0.0000	0.0000		

(*a*) 面积评估

max_delay	0.00	0.00
output external delay	0.00	0.00
data required time		0.00
data required time		0.00
data arrival time		-4.59
slack (VIOLATED)		-4.59

(*b*) 延时评估

图 5-3 最小延时下直接实现 AES S 盒的电路综合评估结果

由上图知，在最小延时约束下，直接实现基于复合域 $GF((2^2)^4)$的 AES S 盒占用的电路面积为 37508.3398μm²，其中合并矩阵 $H_1^{'}$ 占用电路面积为 1037.8369μm²，合并矩阵 $H_3^{'}$ 占用电路面积为 661.9536μm²，乘法逆结构占用电路面积为 35772.1055μm²；电路延时为 4.59ns，即数据到达时间超过了时序约束 4.59ns。

在此约束条件下，复合域 $GF((2^2)^4)$乘法逆结构优化后的 AES S 盒占用电路面积为 14423.2734μm²，其中合并矩阵 $H_1^{'}$ 占用电路面积为 1057.7953μm²，合并矩阵 $H_3^{'}$ 占用电路面积为 665.2798μm²，乘法逆结构占用电路面积为 12663.5928μm²，如图 5-4(*a*)所示；其电路延时为 4.17ns，如图 5-4(*b*)所示。

Hierarchical cell	Global cell area: Absolute Total	Global cell area: Percent Total	Local cell area: Combinational	Local cell area: Noncombinational	Local cell area: Black boxes	Design
subBytes_GF224_S0M1	14423.2637	100.0	36.5904	0.0000	0.0000	subBytes_GF224_S0M1
affine_inv	665.2798	4.6	665.2798	0.0000	0.0000	AffineDeltaInver_A1
map_inv	12663.5928	87.8	3818.7090	0.0000	0.0000	GF224_inverse_S0
map_multi	1057.7953	7.3	1057.7953	0.0000	0.0000	mappMultiplier_M1
Total		14423.2734	0.0000	0.0000		

(*a*) 面积评估

max delay	0.00	0.00
output external delay	0.00	0.00
data required time		0.00
data required time		0.00
data arrival time		-4.17
slack (VIOLATED)		4.17

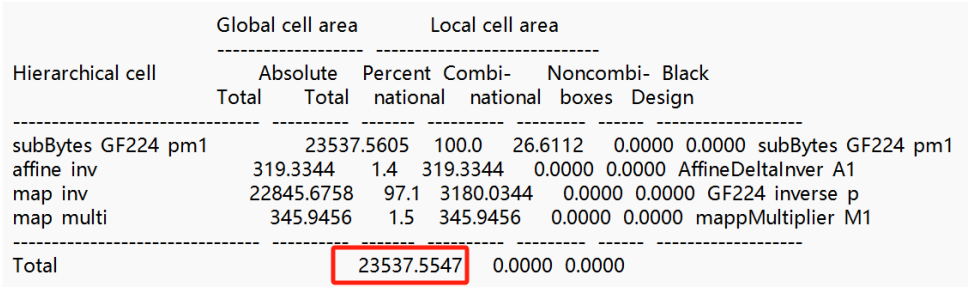
(*b*) 延时评估

图 5-4 最小延时下乘法逆结构优化后的 AES S 盒电路综合评估结果

对比得到，复合域 $GF((2^2)^4)$乘法逆结构优化后的 AES S 盒结构在逻辑综合过程中实现了较大的面积优化，相较于直接实现 S 盒的电路面积优化了 61.6%，其中复合域 $GF((2^2)^4)$乘法逆结构优化率达到了 64.6%，极大地降低了硬件资源占用。复合域 $GF((2^2)^4)$乘法逆结构优化后的 AES S 盒的电路延时较直接实现 S 盒减少了 0.42ns，优化率达到了 9.15%，表明本文的优化减少了 AES S 盒的关键路径。

5.2.2 基于 SMIC 180nm 工艺库最小面积约束下 AES S 盒优化评估

基于 SMIC 180nm 工艺库最小面积约束下，直接实现基于复合域 $GF((2^2)^4)$的 AES S 盒结构的综合评估结果如图 5-5 所示。直接实现基于复合域 $GF((2^2)^4)$的 AES S 盒结构占用电路面积为 23537.5547μm²，其中常数矩阵占用的电路面积为 665.28μm²，乘法逆结构占用的电路面积为 22845.6758μm²，如图 5-5(*a*)所示；如图 5-5(*b*)所示，直接实现基于复合域的 AES S 盒的电路延时为 8.17ns。



Hierarchical cell	Global cell area: Absolute Total	Global cell area: Percent Total	Local cell area: Combi-national	Local cell area: Noncombi-national	Local cell area: Black boxes	Local cell area: Design
subBytes GF224 pm1	23537.5605	100.0	26.6112	0.0000	0.0000	subBytes GF224 pm1
affine inv	319.3344	1.4	319.3344	0.0000	0.0000	AffineDeltaInver A1
map inv	22845.6758	97.1	3180.0344	0.0000	0.0000	GF224 inverse p
map multi	345.9456	1.5	345.9456	0.0000	0.0000	mappMultiplier M1
Total		23537.5547	0.0000	0.0000		

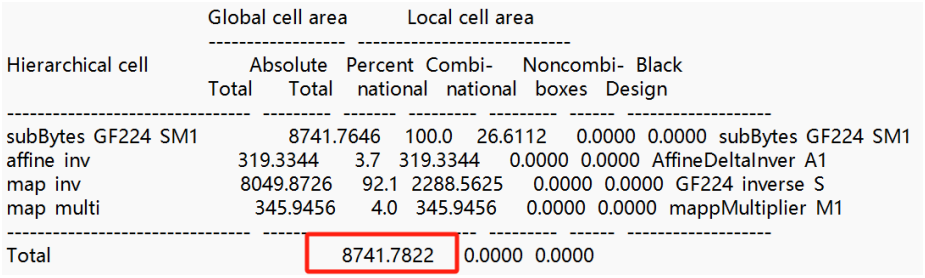
(*a*) 面积评估

data required time	20.00
data arrival time	-11.83
slack (MET)	8.17

(*b*) 延时评估

图 5-5 最小面积下直接实现 AES S 盒的电路综合评估结果

在此约束下，复合域 $GF((2^2)^4)$乘法逆结构优化后的 AES S 盒占用电路面积为 8741.7822μm²，其中常数矩阵占用的面积为 665.28μm²，乘法逆结构占用的电路面积为 8049.8726μm²，如图 5-6(*a*)所示；如图 5-6(*b*)所示，优化后的 AES S 盒的电路延时为 11.20ns。



Hierarchical cell	Global cell area: Absolute Total	Global cell area: Percent Total	Local cell area: Combi-national	Local cell area: Noncombi-national	Local cell area: Black boxes	Local cell area: Design
subBytes GF224 SM1	8741.7646	100.0	26.6112	0.0000	0.0000	subBytes GF224 SM1
affine inv	319.3344	3.7	319.3344	0.0000	0.0000	AffineDeltaInver A1
map inv	8049.8726	92.1	2288.5625	0.0000	0.0000	GF224 inverse S
map multi	345.9456	4.0	345.9456	0.0000	0.0000	mappMultiplier M1
Total		8741.7822	0.0000	0.0000		

(*a*) 面积评估

max delay	20.00	20.00
output external delay	0.00	20.00
data required time		20.00
data required time		20.00
data arrival time		-8.80
slack (MET)		11.20

(*b*) 延时评估

图 5-6 最小面积下乘法逆结构优化后的 AES S 盒电路综合评估结果

对图 5-5 和图 5-6 对比得到，基于复合域 $GF((2^2)^4)$所搭建的优化后的 AES S 盒结构的逻辑综合结果，相较于直接实现 S 盒的电路面积优化率达到了 62.86%，复合

域乘法逆结构优化率达到了 64.76%，即乘法逆结构优化效果显著。复合域 $GF((2^2)^4)$ 乘法逆结构优化后的 AES S 盒结构的电路延时较直接实现 S 盒减少了 3.03ns，优化率达到了 37.09%，表明本文的优化方法对关键路径的优化效果显著。

5.2.3 基于 TSMC 28nm 工艺库电路综合评估结果

基于 TSMC 28nm 工艺库，直接实现基于复合域 $GF((2^2)^4)$的 AES S 盒结构的电路综合评估结果如图 5-7。如图 5-7(*a*)所示，直接实现基于复合域的 AES S 盒占用电路面积为 848.862013μm²；如图 5-7(*b*)所示，其电路延时为 1.89ns。

```
Combinational area:          848.862013
Buf/Inv area:                2.016000
Noncombinational area:       0.000000
Macro/Black Box area:        0.000000
Net Interconnect area:       undefined  (Wire load has zero net area)

Total cell area:             848.862013
Total area:                  undefined
```

(*a*) 面积评估

```
max delay                                    0.00      0.00
output external delay                        0.00      0.00
data required time                                     0.00
-------------------------------------------------------------------------
data required time                                     0.00
data arrival time                                     -1.89
-------------------------------------------------------------------------
slack (VIOLATED)                                      -1.89
```

(*b*) 延时评估

图 5-7 TSMC 28nm 工艺库下直接实现 AES S 盒的电路综合评估结果

如图 5-8(*a*)所示，基于 TSMC 28nm 工艺库，复合域 $GF((2^2)^4)$乘法逆结构优化后的 AES S 盒占用电路面积为 312.102005μm²，相较于直接实现 S 盒的电路面积优化率达到了 62.23%；如图 5-8(*b*)所示，电路延时为 1.56ns，相较于直接实现 S 盒的电路延时优化率达到了 17.46%。

```
Combinational area:          312.102005
Buf/Inv area:                2.016000
Noncombinational area:       0.000000
Macro/Black Box area:        0.000000
Net Interconnect area:       undefined  (Wire load has zero net area)

Total cell area:             312.102005
Total area:                  undefined
1
```

(*a*) 面积评估

```
max delay                                    0.00       0.00
output external delay                        0.00       0.00
data required time                                      0.00
---------------------------------------------------------------
data required time                                      0.00
data arrival time                                      -1.56
---------------------------------------------------------------
slack (VIOLATED)                                       -1.56
```

(*b*) 延时评估

图 5-8 TSMC 28nm 工艺库下乘法逆结构优化后的电路综合评估结果

5.2.4 综合评估结果

本节采用 Synopsys™ 集成电路 IC 综合工具 DC，基于 SMIC 180nm 和 TSMC 28nm 工艺库设计不同约束下，分别对复合域 $GF((2^2)^4)$乘法逆结构优化后与直接实现复合域 $GF((2^2)^4)$ S 盒结构进行综合评估，并对其面积以及延时等关键性能指标进行对比分析。其综合评估结果汇总如表 5-1 所示。

表 5-1 Synopsys 综合评估结果

工艺库	约束	Area(μm²)			Delay(ns)		
		优化前	优化后	优化率	优化前	优化后	优化率
180nm	最小延时	37508.3398	14423.2637	61.6%	4.59	4.17	9.15%
	最小面积	23537.5605	8741.7646	62.86%	8.17	11.2	37.09%
28nm	—	848.862013	312.102005	62.23%	1.89	1.56	17.46%

根据表 5-1 可看出，本文所使用的优化方法，在对 AES S 盒的面积和延时有显著效果，在 SMIC 180nm 工艺库中其面积优化率最高可达到 62.86%，同时关键路径延时也有所降低，最高可达到了 37.09%；在 TSMC 28nm 工艺库中，其面积优化率达到了 62.23%，延时优化率达到了 17.46%。

与其他文献的对比如表 5-2 所示。

表 5-2 相关文献对比

论文	工艺库	Area(μm²)	Delay(ns)	Area-Delay(μm²× ns)
[41]	130nm	—	3.30	—
[41]	65nm	—	7.5	—
[44]	65nm	343.08	2.24	768.50
[44]	65nm	354.60	2.18	773.03
[45]	180nm	2900.6208	26	75416.1408
[47]	180nm	3659.0400	23	84157.9200
[48]	250nm	—	5.0	—
[55]	180nm	2940.5376	23	67632.3648
本文	180nm	14423.2637	4.17	60145.0096
本文	180nm	8741.7646	11.2	97907.7635
本文	28nm	312.102005	1.56	486.879

根据表 5-2 知，在 SMIC 180nm 工艺库中，本文所设计的基于复合域 $GF((2^2)^4)$ 的 S 盒结构 Area-Delay 积在最下延时约束下比论文[45]减少了 20.25%，较论文[46]减少了 28.53%，较论文[55]减少了 11.07%，具有更优的面积-延时性能。研究表明，在不同的工艺库中，本文的研究方法具备较好的优化性能。

5.3 本章小节

本章首先详细地描述了基于复合域 $GF((2^2)^4)$的 AES S 盒的实现细节，包括了伽罗瓦域元素与复合域元素间的转换、常数矩阵合并细节以及 AES S 盒实现流程。其次，采用 Synopsys™ 集成电路 IC 综合工具 DC，分别基于 SMIC 180nm 和 TSMC 28nm 工艺库对复合域 $GF((2^2)^4)$乘法逆结构优化前后的 AES S 盒结构进行综合评估分析。基于 SMIC 180nm 工艺库，其面积优化率最高可达到了 62.86%，关键路径延时优化率最高可达到了 37.09%；基于 TSMC 28nm 工艺库，其面积优化率达到了 62.23%，延时优化率达到了 17.46%，在对 AES S 盒的面积优化中有显著效果的同时，关键路径延时也有所降低，提高了整体电路的时序性能。

第 6 章 总结与展望

6.1 全文总结

本文针对资源受限的应用领域，为了降低 AES 算法中的唯一非线性部件 S 盒的计算复杂度，研究了将 $GF(2^8)$域乘法逆结构映射至复合域 $GF((2^2)^4)$，并对其进行优化，以获取逻辑复杂度最低的乘法逆结构；在乘法逆结构最优的基础上搭建的 AES S 盒，并对其乘数矩阵进行优化设计，获得最优架构的 AES S 盒。最后利用 Synopsys DC 设计工具对构建 AES S 盒进行电路性能综合评估。本文主要研究内容总结如下：

（1）根据同构映射法和复合域 $GF((2^2)^4)$多项式的基础上，研究复合域 $GF((2^2)^4)$的元素表达式，并结合线性代数和复合域运算法则，从而推导复合域 $GF((2^2)^4)$乘法逆结构通式。通过研究分析基于复合域 $GF((2^2)^4)$的可约多项式、不可约多项式以及本原多项式，利用穷搜索法，最终获得在不同本原多项式下逻辑资源最优的复合域 $GF((2^4)^2)$乘法逆结构，即系数最优的复合域 $GF((2^2)^4)$乘法逆结构。

（2）采用 DACSE、DDBT 和 CSE 算法对基于复合域 $GF((2^2)^4)$乘法逆结构进行优化，按照是否将 $GF(2^2)$域元素$[x,\ x^2]$作为公共项进行优化两种方案对复合域 $GF((2^2)^4)$乘法逆结构优化，最终获得最短路径为 15 个乘法器，16 个加法器的最优延时乘法逆结构。

（3）基于复合域 $GF((2^2)^4)$乘法逆结构搭建 AES S 盒，采用常数矩阵合并法对 S 盒中的常数矩阵进行优化合并，对合并后的矩阵采用 CSE 算法进一步减少其逻辑占用，其常数矩阵的最高优化率了达到 44.9%，本文选用了复杂度最低的常数矩阵，共占用 25 个加法器。

（4）本文采用 Synopsys™ 集成电路 IC 综合工具 DC，基于 SMIC 180nm 和 TSMC 28nm 工艺库对所搭建的基于复合域 $GF((2^2)^4)$的 AES S 盒进行电路综合评估。基于 SMIC 180nm 工艺库综合分析表明，相较于直接实现 AES S 盒，乘法逆结构优化后的 AES S 盒面积优化率最高达到了 62.86%，关键路径优化率最高达到了 37.09%；基于 TSMC 28nm 工艺库综合分析表明，相较于直接实现 AES S 盒，乘法逆结构优

化后的AES S盒面积优化率最高达到了62.23%，关键路径优化率最高达到了17.46%，表明本文所使用的优化方法对AES S盒的电路性能进行了有效提升。

6.2 未来展望

为了加强密码算法的安全性以及降低其电路资源消耗，在此我提出了以下展望：

（1）基于$GF(2^8)$域不同的不可约多项式或者不同的基，如：正规基、对偶基等，以获取电路资源最优的AES S盒结构。将结构最优的复合域$GF((2^2)^4)$运算应用于同态加密，利用多项式基、正规基、对偶基等表示伽罗瓦域中元素，针对于伽罗瓦域运算中的加法器和乘法器则可以同态加密中的加法同态运算和乘法同态运算来实现，针对于伽罗瓦域中加法器的设计，可利用逻辑门电路对其不可约多项式系数进行模运算。通过将伽罗瓦域运算与同态加密相融合的方式，可以在保证加密性的同时提高计算效率。

（2）为了平衡密码算法的安全性和密码电路的资源消耗，运行速度等性能之间的关系，可以将同态加密、混沌加密和基于复合域$GF((2^2)^4)$的AES算法相结合，例如，利用同态加密的第三方计算能力构建AES动态S盒，以平衡AES算法安全性不足和同态加密资源消耗多大的特点；利用混沌加密的复杂性，先构建一个较为简单的混沌加密系统对明文数据进行加密，再将此密文数据作为AES加密算法的明文数据进行AES加密运算，这在保证了AES加密系统的低资源消耗和高运行速度的同时，也利用混沌加密的高安全性提高了整个系统的安全性。

（3）随着信息技术发展，侧信道攻击未来有新趋势。研究表明基于功耗攻击和错误攻击的联合旁路攻击对安全芯片具有更高的威胁。为了抵御旁路攻击和错误注入攻击，需要引入一些防御措施，而这些防御措施会大大增加密码电路成本，如何采用较少的代价实现非入侵物理攻击的防御也是目前安全芯片一个关键问题。

参考文献

[1] 赵良驹, 汤宇锋, 龚征. 白盒密码实现的侧信道分析技术综述 [J]. 网络空间安全科学学报, 2024, 2(06): 57-73.

[2] Diffie W, Hellman ME. New directions in cryptogram-phy [M]. Democratizing Cryptography: The Work of Whit-field Diffie and Martin Hellman. ACM, 2022: 365-390.

[3] 程实, 袁毅, 崔峰. 计算机安全与加密算法分析 [J]. 网络安全技术与应用, 2024,(06): 35-37.

[4] Huang H, Han ZG. Computational ghost imaging encryption using RSA algorithm and discrete wavelet transform [J]. RESULTS IN PHYSICS, 2024, 56: 107282.

[5] Shivaramakrishna D, Nagarathna M. A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating AES-OTP and RSA with adaptive key management and Time-Limited access control [J]. ALEXANDRIA ENGINEERING JOURNAL, 2023, 84: 275-284.

[6] Kim GC, Ji HA, Jong YB, etc. Possibility of decryption speed-up by parallel processing in CCA secure hashed ElGamal [J]. PLOS ONE, 2023, 18(11): e0294840.

[7] 王东, 李春平, 肖亚光. 基于 Paillier 和 ElGamal 算法的双层同态密码系统 [J]. 科技创新与应用, 2022, 12(18): 22-25.

[8] Cao XF, Dang LJ, Fan K, etc. A Dynamic and Efficient Self-Certified Authenticated Group Key Agreement Protocol for VANET [J]. IEEE INTERNET OF THINGS JOURNAL, 2024, 11(17): 29146-29156.

[9] Cao XF, Kou WD, Du XN. A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges [J]. INFORMATION SCIENCES, 2010, 180(15): 2895-2903.

[10] Ullah S, Zheng JB, Din N, etc. Elliptic Curve Cryptography; Applications, challenges, recent advances, and future trends: A comprehensive survey [J].

COMPUTER SCIENCE REVIEW, 2023, 47.

[11]薛来军. 基于改进的 ECC 的网络信息安全加密方法 [J]. 太原师范学院学报(自然科学版), 2022, 21(02): 48-50+89.

[12]高永琳. 基于区块链的安全技术研究 [D]. 北京: 华北电力大学, 2019.

[13]姚重阳. 基于 AES 算法的加密 I~2C 控制器设计与验证 [D]. 西安: 西安电子科技大学, 2022.

[14]吴静莉. 基于改进 DES 算法的网络链路安全通信方法 [J]. 无线互联科技, 2024, 21(18): 47-49.

[15]张洁, 朱丽娟. DES 加密算法分析与实现 [J]. 软件导刊, 2007, (03): 95-97.

[16]Tezcan C. Key lengths revisited: GPU-based brute force cryptanalysis of DES, 3DES, and PRESENT [J]. JOURNAL OF SYSTEMS ARCHITECTURE, 2022, 124.

[17]Hsiao FH. Applying 3DES to Chaotic Synchronization Cryptosystems [J]. IEEE ACCESS, 2022, 10: 1036-1050.

[18]Liu GZ, Li W, Fan XK, etc. An Image Encryption Algorithm Based on Discrete-Time Alternating Quantum Walk and Advanced Encryption Standard [J]. ENTROPY, 2024, 24(5): 608.

[19]Lee WK, Seo HJ, Seo SC, etc. Efficient Implementation of AES-CTR and AES-ECB on GPUs With Applications for High-Speed FrodoKEM and Exhaustive Key Search [J]. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II-EXPRESS BRIEFS, 2022, 69(6): 2962-2966.

[20]李湘锋, 赵有健, 全成斌. 对称密钥加密算法在 IPsec 协议中的应用 [J]. 电子测量与仪器学报, 2014, 28(01): 75-83.

[21]李超, 沈静. Camellia 的差分和线性迭代特征 [J]. 电子学报, 2005, (08): 1345-1348.

[22]Li JL, Mo YN, Su T, etc. Hardware Design of High-speedHybrid Eneryption System Based on National Secret AlgorithmsSM2, SM3, SM4 [J]. Computer Application Research, 2022. 39(9): 2818-2825, 2831.

[23]王晓燕, 杨先文. 基于 FPGA 的 SM3 算法优化设计与实现 [J]. 计算机工程, 2012, 38(06): 244-246.

[24]Bai KP, Wu CK. A secure white-box SM4 implementation [J]. SECURITY AND COMMUNICATION NETWORKS, 2016, 9(10): 996-1006.

[25]Wang CF, Ding Q. SM4 key scheme algorithm based on chaotic system [J]. ACTA PHYSICA SINICA, 2017, 66(2): 020504.

[26]National Institute of Standards and Technology. Advanced Encryption Standard (AES) [R]. Gaithersburg, MD 20899 - 8900: U.S. Department of Commerce, 2001.

[27]张敬源. AES 算法在数据安全加密中的应用 [J]. 现代工业经济和信息化, 2023, 13(3): 75-76.

[28]Hammad I, El-Sankary K, El-Masry E. High-Speed AES Encryptor With Efficient Merging Techniques [J]. IEEE Embedded Systems Letters, 2010, 2(3): 67–71.

[29]Mentens N, Batina L, Preneel B, etc. A Systematic Evaluation of Compact Hardware Implementations for the Rijndael S-Box [C]. Processing of Cryptographers Track at the RSA Conference (CT-RSA’05), 2006: 323–333.

[30]Abbasi I, Afzal M. A Compact S-Box Design for SMS4 Block Cipher [C]. International Conference on Information Technology Convergence and Service (ITCS 2011), IT Convergence and Services, LNEE 108, Springer Netherlands, 2011: 641-658.

[31]Martínez-Herrera AF, Mex-Perera JC, Nolazco-Flores JA. Some Representations of the S-Box of Camellia in $GF(((2^2)^2)^2)$ [C]. 11th International Conference on Cryptology and Network Security, Springer LNCS 7712, 2012: 296-309.

[32]李炽阳, 雷倩倩, 杨延飞. 全通用 AES 加密算法的 FPGA 实现 [J]. 计算机工程与应用, 2020, 56(10): 5-10.

[33]王亚涛. AES 加解密算法及其安全性分析 [J]. 网络安全技术与应用, 2022, (09): 33-35.

[34]李艳俊, 张伟国, 葛耀东, 等. 类 AES 算法 S 盒的优化实现 [J]. 密码学报, 2023,

10(03): 531-538.

[35]付婷婷. 基于动态 S 盒的 AES 加密算法设计 [J]. 计量与测试技术, 2025, 51(01): 115-118.

[36]Wong MM, Wong MLD, Nandi AK, etc. Construction of Optimum Composite Field Architecture for Compact High-Throughput AES S-boxes [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2012, 20(6): 1151-1155.

[37]Fan CP, Hwang JK. Implementations of High Throughput Sequential and Fully Pipelined AES. Processors on FPGA [C]. Proceedings of 2007 International Symposium on Intelligent Signal Processing and Communication Systems, 2007: 353-356.

[38]Rijmen, Vincent. Efficient Implementation of the Rijndael S-box [EB/OL]. https: //api. semanticscholar. org/CorpusID: 16891004, 2000.

[39]Canright, D. A Very Compact S-Box for AES [J]. Cryptographic Hardware and Embedded Systems – CHES 2005, 2005: 441-455.

[40]Zhang XM. High-speed VLSI Architectures for Error-correcting Codes and Cryptosystems [D]. Minnesota: University of Minnesota System, 2005.

[41]Kamel D, Standaert FX, Flandre D. Scaling trends of the AES S-box lower power consumption in 130 and 65 nm CMOS Technology Nodes [C]. Proceedings of the International Symposium on Circuits and Systems 2009 (ISCAS 2009). 2009: 1385-1388.

[42]Luo QB, Li XY, Yang GW, etc. Quantum reversible circuits for $GF(2^8)$ multiplication based on composite field arithmetic operations [J]. Quantum Information Processing, 2023, 22(1).

[43]Ashmawy D, Reyhani-Masoleh A. A Faster Hardware Implementation of the AES S-box [C]. 2021 IEEE 28th Symposium on Computer Arithmetic (ARITH). Lyngby, Denmark, 2021: 123-130.

[44]戴强, 戴紫彬, 李伟. 基于增强型延时感知 CSE 算法的 AES S 盒电路优化设计

[J]. 电子学报, 2019, 47(01): 129-136.

[45] Wong MM, Wong MLD, Nandi AK, etc. Composite field$GF(((2^2)^2)^2)$advanced encryption standard (AES) S-boxwith algebraic normal form representation in the subfield inver-sion [J]. Circuits, Devices & Systems, lET, 2011, 5(6): 471-476.

[46] Kim JW, Kang MS. Design of Advanced Multiplicative Inverse Operation Circuit for AES Encryption [J]. The Journal of the Institute of Webcasting, Internet and Telecommunication, 2020, 20: 1-6.

[47] Zhang XM, Parhi KK. High-speed VLSI architectures for theAES algorithm [J]. lEEE Transactions on Very Large Scale In-tegration(VLSI) Systems, 2004, 12(9): 957-967.

[48] Morioka S, Satoh A. A 10-Gbps Full-AES Crypto Design With a Twisted BDD S-Box Architecture [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2004, 12(7): 686-691.

[49] Mentens N, Batina L, Preneel B, etc. A Systematic Evaluation of Compact Hardware Implementations for the Rijndael S-Box [C]. Processing of Cryptographers Track at the RSA Conference (CT-RSA'05), 2006: 323-333.

[50] Zhang XM, Parhi KK. On the optimum constructions of composite field for the AES algorithm [J]. IEEE Transaction on Circuits System II, 2006, 53(10): 1153-1157.

[51] Canright D. A Very Compact S-Box for AES [C]. Workshop on Cryptographic Hardware and Embedded Systems (CHES 2005). Springer-Verlag, 2006: 441-455.

[52] Abbasi I, Afzal M. A Compact S-Box Design for SMS4 Block Cipher [C]. International Conference on Information Technology Convergence and Service (ITCS 2011), IT Convergence and Services, LNEE 108. Springer Netherlands, 2011: 641-658.

[53] Martínez-Herrera F, Mex-Perera JC, Nolazco-Flores JA. Some Representations of the S-Box of Camellia in $GF(((2^2)^2)^2)$ [C]. 11th International Conference on Cryptology and Network Security, Springer LNCS 7712, 2012: 296-309.

[54] Patel K. Performance analysis of AES, DES and Blowfish cryptographic algorithms on small andlarge data files [J]. International Journal of Information Technology, 2019, 11(4): 813-819.

[55] 曾纯, 吴宁, 张肖强, 等. 基于多因子 CSE 算法的 AES S-盒电路优化设计 [J]. 电子学报, 2014, 42(006): 1238-1243.

[56] Gebeyehu N, Demilew S. Enhanced Security of Advanced Encryption Standard (ES-AES) Algorithm [J]. American Journal of Computer Science and Technology, 2022, 5 (2): 41.

[57] Singh A, Prasad A, Talwar Y. Compact and Secure S-Box Implementations of AES —A Review [C]. A. K. Somani, R. S. Shekhawat, A. Mundra, S. Srivastava, V. K. Verma. Smart Systems and IoT: Innovations in Computing (Smart Innovation, Systems and Technologies, Vol. 141). 2020.

[58] Zhang XQ, Wu N, Zhou F, etc. Low-delay parallel Chien search architecture for RS decoder [J]. IEICE Electronics Express, 2016. 1-6.

[59] PetraN, Caro DD, Strollo AGM. A Novel Architecture for Galois Fields $GF(2^m)$ Multipliers Based on Mastrovito Scheme [J]. IEEE Transactions on Computers, 2007, 56(1): 1470-1483.

[60] Zhang XQ. An optimized delay-aware common subexpression elimination algorithm for hardware implementation of binary-field linear transform [J]. IEICE Electron. Express, 2014, 11: 20140934.

[61] Zhang XQ, Wu N, Zhou F, etc. An optimized delay-aware common subexpression elimination algorithm for hardware implementation of binary-field linear transform [J]. IEICE Electronics Express, 2014, 1-8.

[62] PETRA N, CARO DD, STROLLO AGM. A Novel Architecture for Galois Fields *GF*(2m) Multipliers Based on Mastrovito Scheme [J]. IEEE Transactions on Computers, 2007, 56(1): 1470-1483.

[63] Chen N, Yan ZY. Compact designs of mixcolumns and subbytes using a novel

common subexpression elimination algorithm [C]. IEEE International Symposium on Circuits and Systems (ISCAS 2008), 2008: 1584-1587.

[64] Chen N, Yan ZY. High-Performance Designs of AES Transformations [C]. IEEE International Symposium on Circuits and Systems (ISCAS 2009), 2009: 2906-2909.

[65] 焦占亚, 曾永莹, 刘海峰. 一种基于伽罗瓦域的密码系统 [J]. 计算机工程与应用, 2005, (30): 146-148.

[66] 杨波. 现代密码学(第 4 版) [M]. 北京: 清华大学出版社, 2017 年: 83-95.

[67] Wang JH, Chang HW, Chiou CW, etc. Low-complexity design of bit-parallel dual-basis multiplier over $GF(2^m)$ [J]. IET Information Security, 2012, 6(4): 324-328.

[68] Hua YY, Lin JM, Che WC, etc. Low space-complexity digit-serial dual basis systolic multiplier over Galois field $GF(2^m)$ using Hankel matrix and Karatsuba algorithm [J]. IET Information Security, 2013, 7(2): 75-86.

[69] 秦菁. 基于 Synopsys 的 8051 单片机 IP 核的设计 [D]. 保定: 华北电力大学, 2014.

攻读硕士期间取得的学术成果

发表论文：

[1] Zheng XX, Zhang XQ, Yang F, Xu MY, Yan H. A Low Delay AES Round Transform Architecture Using Constant Matrix Multiplications Merging Technologies[C]. 2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA), 2022, 1364-1371. （**本人五作**，导师二作）

[2] Zhang XQ, Peng ZW, Yan H, Zheng XX, Xu MY. A Low-Delay Circuit Structure Construction Method for AES Key Expansion Units［C］. 2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA), 2024, 1 - 4. （**本人三作**，导师一作）

[3] 闫涵, 张肖强, 彭志伟, 等. 面向探测卫星通信的高性能低开销 AES 加解密电路研究［J］. 重庆工商大学学报 (自然科学版), 2024：1-9. （**本人一作**，导师二作）

[4] Zheng XX, Yan H, Peng ZW, Zhang XQ. Low-Delay AES Key Expansion Units Based on DDBT Structure［J］. ELECTRONICS, 2025, 14(2)：384. （**本人二作**，老师四作）

参加项目：

[1] 纵向项目：安徽省车载显示集成系统工程研究中心学生类开放课题：“基于对偶基的 $GF((2^2)^4)$域 AES S 盒电路优化设计方法研究”，项目周期：2023. 12-2025. 11，项目编号：VDIS2023D02，**项目主持人**。

致谢

岁月缱绻，葳蕤生香。当我敲下毕业论文的最后一个字符，这段承载着梦想与拼搏的研究生时光，已悄然临近终点。在这段充满挑战与收获的的求学时光里，我能稳步前行，离不开众多良师益友、家人的无私帮助与坚定支持。此刻，心中满是感激，愿借这寥寥数语，向每一位曾给予我关怀、支持与帮助的人，倾吐我最真挚的谢意。

我要将最诚挚的感恩，敬献给我的导师张肖强副教授。您是我学术海洋中的引航者，亦是我人生道路上的启迪者。“云山苍苍，江水泱泱，先生之风，山高水长。”在学术的浩瀚星空中，您宛如一颗明亮的北斗星，为我指引前行的方向。还记得初次踏入密码学领域时，我满心迷茫，是您用一次一次细致入微的讲解，让我逐渐敲开了知识的大门。每一次面对我的疑问，您总是耐心倾听，随后给出精准且富有建设性的建议；每一次面对公式与程序的错误，您总是一遍一遍的实验，不厌其烦帮助我查找错误。山高水长有时尽，唯我师恩日月长。在您的悉心指导下，我从一个对学术懵懵懂懂的新人，逐渐成长为能够独立思考、开展研究的学子。您严谨的治学态度，已深深烙印在我心间，成为我未来学术道路上的行事准则。

家，是我永远的避风港。我要感谢我的父母。在我因研究进展不顺而焦虑时，你们温柔的话语总能抚平我内心的波澜；经济上的全力支持，让我毫无后顾之忧地投身学业。在漫长的求学岁月里，给予我陪伴与理解，每当我疲惫不堪，是你们的拥抱与鼓励，给予我重新出发的力量。

我的同学们，感谢你们为我的硕士生涯增添不一样的色彩。如今回忆起来满是温馨。感谢你们在我迷茫时的陪伴，在我成功时的欢呼，是你们让我的校园生活充满色彩。

在此，我再次向所有关心、支持与帮助过我的人表达最诚挚的谢意。这份感恩将化作我前行的动力，激励我在未来的日子里，不断探索，努力奋进，以更好的成绩回报大家的厚爱。

附录 1 可约多项式

假设复合域 $GF((2^2)^4)$的多项式为 $f(y)=y^4+\beta_3y^3+\beta_2y^2+\beta_1y+\beta_0$，其中 $\{\beta_3,\beta_2,\beta_1,\beta_0\}\in GF(2^2)$，$GF(2^2)$域的元素有：$\{0,1,x,x^2\}$，下表列出了基于假设复合域 $GF((2^2)^4)$的所有可约多项。

序号	系数				$f(y)$
	β_3	β_2	β_1	β_0	
1	0	0	0	0	y^4
2	0	0	0	1	y^4+1
3	0	0	0	x	y^4+x
4	0	0	0	x^2	y^4+x^2
5	0	0	1	0	y^4+y
6	0	0	1	1	y^4+y+1
7	0	0	1	x	y^4+y+x
8	0	0	1	x^2	y^4+y+x^2
9	0	0	x	0	y^4+xy
10	0	0	x	1	y^4+xy+1
11	0	0	x	x	y^4+xy+x
12	0	0	x	x^2	y^4+xy+x^2
13	0	0	x^2	0	y^4+x^2y
14	0	0	x^2	1	y^4+x^2y+1
15	0	0	x^2	x	y^4+x^2y+x
16	0	0	x^2	x^2	$y^4+x^2y+x^2$
17	0	1	0	0	y^4+y^2
18	0	1	0	1	y^4+y^2+1
19	0	1	0	x	y^4+y^2+x
20	0	1	0	x^2	$y^4+y^2+x^2$
21	0	1	1	0	y^4+y^2+y
22	0	1	1	1	y^4+y^2+y+1
23	0	1	1	x	y^4+y^2+y+x
24	0	1	1	x^2	$y^4+y^2+y+x^2$
25	0	1	x	0	y^4+y^2+xy
26	0	1	x	x	y^4+y^2+xy+x
27	0	1	x^2	0	$y^4+y^2+x^2y$
28	0	1	x^2	x^2	$y^4+y^2+x^2y+x^2$
29	0	x	0	0	y^4+xy^2
30	0	x	0	1	y^4+xy^2+1
31	0	x	0	x	y^4+xy^2+x
32	0	x	0	x^2	$y^4+xy^2+x^2$
33	0	x	1	0	y^4+xy^2+y
34	0	x	1	1	y^4+xy^2+y+1
35	0	x	1	x	y^4+xy^2+y+x
36	0	x	1	x^2	$y^4+xy^2+y+x^2$
37	0	x	x	0	y^4+xy^2+xy
38	0	x	x	1	y^4+xy^2+xy+1
39	0	x	x^2	0	$y^4+xy^2+x^2y$
40	0	x	x^2	x	$y^4+xy^2+x^2y+x$
41	0	x^2	0	0	$y^4+x^2y^2$
42	0	x^2	0	1	$y^4+x^2y^2+1$
43	0	x^2	0	x	$y^4+x^2y^2+x$
44	0	x^2	0	x^2	$y^4+x^2y^2+x^2$
45	0	x^2	1	0	$y^4+x^2y^2+y$
46	0	x^2	1	1	$y^4+x^2y^2+y+1$
47	0	x^2	1	x	$y^4+x^2y^2+y+x$
48	0	x^2	1	x^2	$y^4+x^2y^2+y+x^2$
49	0	x^2	x	0	$y^4+x^2y^2+xy$
50	0	x^2	x	x^2	$y^4+x^2y^2+xy+x^2$
51	0	x^2	x^2	0	$y^4+x^2y^2+x^2y$
52	0	x^2	x^2	1	$y^4+x^2y^2+x^2y+1$
53	1	0	0	0	y^4+y^3
54	1	0	0	1	y^4+y^3+1
55	1	0	0	x	y^4+y^3+x
56	1	0	0	x^2	$y^4+y^3+x^2$
57	1	0	1	0	y^4+y^3+y
58	1	0	1	1	y^4+y^3+y+1

序号	系数				$f(y)$
	β_3	β_2	β_1	β_0	
59	1	0	x	0	y^4+y^3+xy
60	1	0	x	x^2	$y^4+y^3+xy+x^2$
61	1	0	x^2	0	$y^4+y^3+x^2y$
62	1	0	x^2	x	$y^4+y^3+x^2y+x$
63	1	0	x^2	x^2	$y^4+y^3+x^2y+x^2$
64	1	1	0	0	$y^4+y^3+y^2$
65	1	1	0	1	$y^4+y^3+y^2+1$
66	1	1	1	0	$y^4+y^3+y^2+y$
67	1	1	1	1	$y^4+y^3+y^2+y+1$
68	1	1	1	x^2	$y^4+y^3+y^2+y+x^2$
69	1	1	x	0	$y^4+y^3+y^2+xy$
70	1	1	x	1	$y^4+y^3+y^2+xy+1$
71	1	1	x	x^2	$y^4+y^3+y^2+xy+x^2$
72	1	1	x^2	0	$y^4+y^3+y^2+x^2y$
73	1	1	x^2	1	$y^4+y^3+y^2+x^2y+1$
74	1	1	x^2	x	$y^4+y^3+y^2+x^2y+x$
75	1	1	x^2	x^2	$y^4+y^3+y^2+x^2y+x^2$
76	1	x	0	0	$y^4+y^3+xy^2$
77	1	x	0	1	$y^4+y^3+xy^2+1$
78	1	x	0	x	$y^4+y^3+xy^2+x$
79	1	x	1	0	$y^4+y^3+xy^2+y$
80	1	x	1	x^2	$y^4+y^3+xy^2+y+x^2$
81	1	x	x	0	$y^4+y^3+xy^2+xy$
82	1	x	x	1	$y^4+y^3+xy^2+xy+1$
83	1	x	x	x	$y^4+y^3+xy^2+xy+x$
84	1	x	x^2	0	$y^4+y^3+xy^2+x^2y$
85	1	x	x^2	1	$y^4+y^3+xy^2+x^2y+1$
86	1	x	x^2	x	$y^4+y^3+xy^2+x^2y+x$
87	1	x	x^2	x^2	$y^4+y^3+xy^2+x^2y+x^2$
88	1	x^2	0	0	$y^4+y^3+x^2y^2$
89	1	x^2	0	1	$y^4+y^3+x^2y^2+1$
90	1	x^2	0	x^2	$y^4+y^3+x^2y^2+x^2$
91	1	x^2	1	0	$y^4+y^3+x^2y^2+y$
92	1	x^2	1	x	$y^4+y^3+x^2y^2+y+x$
93	1	x^2	1	x^2	$y^4+y^3+x^2y^2+y+x^2$
94	1	x^2	x	0	$y^4+y^3+x^2y^2+xy$
95	1	x^2	x	1	$y^4+y^3+x^2y^2+xy+1$
96	1	x^2	x	x	$y^4+y^3+x^2y^2+xy+x$

序号	系数				$f(y)$
	β_3	β_2	β_1	β_0	
97	1	x^2	x	x^2	$y^4+y^3+x^2y^2+xy+x^2$
98	1	x^2	x^2	0	$y^4+y^3+x^2y^2+x^2y$
99	1	x^2	x^2	1	$y^4+y^3+x^2y^2+x^2y+1$
100	x	0	0	0	y^4+xy^3
101	x	0	0	1	y^4+xy^3+1
102	x	0	0	x	y^4+xy^3+x
103	x	0	0	x^2	$y^4+xy^3+x^2$
104	x	0	1	0	y^4+xy^3+y
105	x	0	1	1	y^4+xy^3+y+1
106	x	0	1	x	y^4+xy^3+y+x
107	x	0	x	0	y^4+xy^3+xy
108	x	0	x	1	y^4+xy^3+xy+1
109	x	0	x	x	y^4+xy^3+xy+x
110	x	0	x	x^2	$y^4+xy^3+xy+x^2$
111	x	0	x^2	0	$y^4+xy^3+x^2y$
112	x	0	x^2	1	$y^4+xy^3+x^2y+1$
113	x	0	x^2	x^2	$y^4+xy^3+x^2y+x^2$
114	x	1	0	0	$y^4+xy^3+y^2$
115	x	1	0	x	$y^4+xy^3+y^2+x$
116	x	1	0	x^2	$y^4+xy^3+y^2+x^2$
117	x	1	1	0	$y^4+xy^3+y^2+y$
118	x	1	1	1	$y^4+xy^3+y^2+y+1$
119	x	1	1	x^2	$y^4+xy^3+y^2+y+x^2$
120	x	1	x	0	$y^4+xy^3+y^2+xy$
121	x	1	x	x	$y^4+xy^3+y^2+xy+x$
122	x	1	x^2	0	$y^4+xy^3+y^2+x^2y$
123	x	1	x^2	1	$y^4+xy^3+y^2+x^2y+1$
124	x	1	x^2	x	$y^4+xy^3+y^2+x^2y+x$
125	x	1	x^2	x^2	$y^4+xy^3+y^2+x^2y+x^2$
126	x	x	0	0	$y^4+xy^3+xy^2$
127	x	x	0	1	$y^4+xy^3+xy^2+1$
128	x	x	0	x	$y^4+xy^3+xy^2+x$
129	x	x	1	0	$y^4+xy^3+xy^2+y$
130	x	x	1	1	$y^4+xy^3+xy^2+y+1$
131	x	x	1	x^2	$y^4+xy^3+xy^2+y+x^2$
132	x	x	x	0	$y^4+xy^3+xy^2+xy$
133	x	x	x	1	$y^4+xy^3+xy^2+xy+1$
134	x	x	x	x	$y^4+xy^3+xy^2+xy+x$

序号	系数				$f(y)$
	β_3	β_2	β_1	β_0	
135	x	x	x	x^2	$y^4+xy^3+xy^2+xy+x^2$
136	x	x	x^2	0	$y^4+xy^3+xy^2+x^2y$
137	x	x	x^2	x	$y^4+xy^3+xy^2+x^2y+x$
138	x	x^2	0	0	$y^4+xy^3+x^2y^2$
139	x	x^2	0	x	$y^4+xy^3+x^2y^2+x$
140	x	x^2	0	x^2	$y^4+xy^3+x^2y^2+x^2$
141	x	x^2	1	0	$y^4+xy^3+x^2y^2+y$
142	x	x^2	1	1	$y^4+xy^3+x^2y^2+y+1$
143	x	x^2	1	x	$y^4+xy^3+x^2y^2+y+x$
144	x	x^2	1	x^2	$y^4+xy^3+x^2y^2+y+x^2$
145	x	x^2	x	0	$y^4+xy^3+x^2y^2+xy$
146	x	x^2	x	1	$y^4+xy^3+x^2y^2+xy+1$
147	x	x^2	x	x	$y^4+xy^3+x^2y^2+xy+x$
148	x	x^2	x^2	0	$y^4+xy^3+x^2y^2+x^2y$
149	x	x^2	x^2	x	$y^4+xy^3+x^2y^2+x^2y+x$
150	x	x^2	x^2	x^2	$y^4+xy^3+x^2y^2+x^2y+x^2$
151	x^2	0	0	0	$y^4+x^2y^3$
152	x^2	0	0	1	$y^4+x^2y^3+1$
153	x^2	0	0	x	$y^4+x^2y^3+x$
154	x^2	0	0	x^2	$y^4+x^2y^3+x^2$
155	x^2	0	1	0	$y^4+x^2y^3+y$
156	x^2	0	1	x^2	$y^4+x^2y^3+y+x$
157	x^2	0	x	0	$y^4+x^2y^3+xy$
158	x^2	0	x	1	$y^4+x^2y^3+xy+1$
159	x^2	0	x	x	$y^4+x^2y^3+xy+x$
160	x^2	0	x^2	0	$y^4+x^2y^3+x^2y$
161	x^2	0	x^2	1	$y^4+x^2y^3+x^2y+1$
162	x^2	0	x^2	x	$y^4+x^2y^3+x^2y+x$
163	x^2	1	0	0	$y^4+x^2y^3+y^2$
164	x^2	1	0	x	$y^4+x^2y^3+y^2+x$
165	x^2	1	0	x^2	$y^4+x^2y^3+y^2+x^2$
166	x^2	1	1	0	$y^4+x^2y^3+y^2+y$

序号	系数				$f(y)$
	β_3	β_2	β_1	β_0	
167	x^2	1	1	1	$y^4+x^2y^3+y^2+y+1$
168	x^2	1	1	x	$y^4+x^2y^3+y^2+y+x$
169	x^2	1	x	0	$y^4+x^2y^3+y^2+xy$
170	x^2	1	x	1	$y^4+x^2y^3+y^2+xy+1$
171	x^2	1	x	x	$y^4+x^2y^3+y^2+xy+x$
172	x^2	1	x	x^2	$y^4+x^2y^3+y^2+xy+x^2$
173	x^2	1	x^2	0	$y^4+x^2y^3+y^2+x^2y$
174	x^2	1	x^2	x^2	$y^4+x^2y^3+y^2+x^2y+x^2$
175	x^2	x	0	0	$y^4+x^2y^3+xy^2$
176	x^2	x	0	x^2	$y^4+x^2y^3+xy^2+x^2$
177	x^2	x	1	0	$y^4+x^2y^3+xy^2+y$
178	x^2	x	1	1	$y^4+x^2y^3+xy^2+y+1$
179	x^2	x	1	x	$y^4+x^2y^3+xy^2+y+x$
180	x^2	x	1	x^2	$y^4+x^2y^3+xy^2+y+x^2$
181	x^2	x	x	0	$y^4+x^2y^3+xy^2+xy$
182	x^2	x	x	x	$y^4+x^2y^3+xy^2+xy+x$
183	x^2	x	x	x^2	$y^4+x^2y^3+xy^2+xy+x^2$
184	x^2	x	x^2	0	$y^4+x^2y^3+xy^2+x^2y$
185	x^2	x	x^2	1	$y^4+x^2y^3+xy^2+x^2y+1$
186	x^2	x	x^2	x^2	$y^4+x^2y^3+xy^2+x^2y+x^2$
187	x^2	x^2	0	0	$y^4+x^2y^3+x^2y^2$
188	x^2	x^2	0	1	$y^4+x^2y^3+x^2y^2+1$
189	x^2	x^2	0	x^2	$y^4+x^2y^3+x^2y^2+x^2$
190	x^2	x^2	1	0	$y^4+x^2y^3+x^2y^2+y$
191	x^2	x^2	1	1	$y^4+x^2y^3+x^2y^2+y+1$
192	x^2	x^2	1	x	$y^4+x^2y^3+x^2y^2+y+x$
193	x^2	x^2	x	0	$y^4+x^2y^3+x^2y^2+xy$
194	x^2	x^2	x	x^2	$y^4+x^2y^3+x^2y^2+xy+x^2$
195	x^2	x^2	x^2	0	$y^4+x^2y^3+x^2y^2+x^2y$
196	x^2	x^2	x^2	1	$y^4+x^2y^3+x^2y^2+x^2y+1$
197	x^2	x^2	x^2	x	$y^4+x^2y^3+x^2y^2+x^2y+x$
198	x^2	x^2	x^2	x^2	$y^4+x^2y^3+x^2y^2+x^2y+x^2$

附录 2 不可约多项式

下表列出了基于假设复合域 $GF((2^2)^4)$的所有不可约多项。

序号	系数				$f(y)$
	β_3	β_2	β_1	β_0	
1	0	1	x	1	y^4+y^2+xy+1
2	0	1	x	x^2	$y^4+y^2+xy+x^2$
3	0	1	x^2	1	$y^4+y^2+x^2y+1$
4	0	1	x^2	x	$y^4+y^2+x^2y+x$
5	0	x	x	x	y^4+xy^2+xy+x
6	0	x	x	x^2	$y^4+xy^2+xy+x^2$
7	0	x	x^2	1	$y^4+xy^2+x^2y+1$
8	0	x	x^2	x^2	$y^4+xy^2+x^2y+x^2$
9	0	x^2	x	1	$y^4+x^2y^2+xy+1$
10	0	x^2	x	x	$y^4+x^2y^2+xy+x$
11	0	x^2	x^2	x	$y^4+x^2y^2+x^2y+x$
12	0	x^2	x^2	x^2	$y^4+x^2y^2+x^2y+x^2$
13	1	0	1	x	y^4+y^3+y+x
14	1	0	1	x^2	$y^4+y^3+y+x^2$
15	1	0	x	1	y^4+y^3+xy+1
16	1	0	x	x	y^4+y^3+xy+x
17	1	0	x^2	1	$y^4+y^3+x^2y+1$
18	1	1	0	x	$y^4+y^3+y^2+x$
19	1	1	0	x^2	$y^4+y^3+y^2+x^2$
20	1	1	1	x	$y^4+y^3+y^2+y+x$
21	1	1	x	x	$y^4+y^3+y^2+xy+x$
22	1	x	0	x^2	$y^4+y^3+xy^2+x^2$
23	1	x	1	1	$y^4+y^3+xy^2+y+1$
24	1	x	1	x	$y^4+y^3+xy^2+y+x$
25	1	x	x	x^2	$y^4+y^3+xy^2+xy+x^2$
26	1	x^2	0	x	$y^4+y^3+x^2y^2+x$
27	1	x^2	1	1	$y^4+y^3+x^2y^2+y+1$
28	1	x^2	x^2	x	$y^4+y^3+x^2y^2+x^2y+x$
29	1	x^2	x^2	x^2	$y^4+y^3+x^2y^2+x^2y+x^2$

序号	系数				$f(y)$
	β_3	β_2	β_1	β_0	
30	x	0	1	x^2	$y^4+xy^3+y+x^2$
31	x	0	x^2	x	$y^4+xy^3+x^2y+x$
32	x	1	0	1	$y^4+xy^3+y^2+1$
33	x	1	1	x	$y^4+xy^3+y^2+y+x$
34	x	1	x	1	$y^4+xy^3+y^2+xy+1$
35	x	1	x	x^2	$y^4+xy^3+y^2+xy+x^2$
36	x	x	0	x^2	$y^4+xy^3+xy^2+x^2$
37	x	x	1	x	$y^4+xy^3+xy^2+y+x$
38	x	x	x^2	1	$y^4+xy^3+xy^2+x^2y+1$
39	x	x	x^2	x^2	$y^4+xy^3+xy^2+x^2y+x^2$
40	x	x^2	0	1	$y^4+xy^3+x^2y^2+1$
41	x	x^2	x	x^2	$y^4+xy^3+x^2y^2+xy+x^2$
42	x	x^2	x^2	1	$y^4+xy^3+x^2y^2+x^2y+1$
43	x^2	0	1	1	$y^4+x^2y^3+y+1$
44	x^2	0	1	x	$y^4+x^2y^3+y+x$
45	x^2	0	x	x^2	$y^4+x^2y^3+xy+x^2$
46	x^2	0	x^2	x^2	$y^4+x^2y^3+x^2y+x^2$
47	x^2	1	0	1	$y^4+x^2y^3+y^2+1$
48	x^2	1	1	x^2	$y^4+x^2y^3+y^2+y+x^2$
49	x^2	1	x^2	1	$y^4+x^2y^3+y^2+x^2y+1$
50	x^2	1	x^2	x	$y^4+x^2y^3+y^2+x^2y+x$
51	x^2	x	0	1	$y^4+x^2y^3+xy^2+1$
52	x^2	x	0	x	$y^4+x^2y^3+xy^2+x$
53	x^2	x	x	1	$y^4+x^2y^3+xy^2+xy+1$
54	x^2	x	x^2	x	$y^4+x^2y^3+xy^2+x^2y+x$
55	x^2	x^2	0	x	$y^4+x^2y^3+x^2y^2+x$
56	x^2	x^2	1	x^2	$y^4+x^2y^3+x^2y^2+y+x^2$
57	x^2	x^2	x	1	$y^4+x^2y^3+x^2y^2+xy+1$
58	x^2	x^2	x	x	$y^4+x^2y^3+x^2y^2+xy+x$

附录 3 将[x,x^2]作为常数项提取公共项

将[x,x^2]作为常数项进行优化提取公共项时，所生成的 51×138 列矩阵，每列所代表的公共项及其最短路径如下表。

列数	元素	表达式	最短路径	
			XOR	AND
1	b_0	r_0	0	0
2	b_1	r_0^2	1	0
3	b_2	r_1	0	0
4	b_3	r_1^2	1	0
5	b_4	r_2	0	0
6	b_5	r_2^2	1	0
7	b_6	r_3	0	0
8	b_7	r_3^2	1	0
9	s_0	b_4*b_6	4	5
10	s_1	b_4*b_5	4	5
11	s_2	b_4*b_7	4	5
12	s_3	b_6*b_7	4	5
13	s_4	b_0*b_2	3	5
14	s_5	b_5*b_6	4	5
15	s_6	b_1*b_6	4	5
16	s_7	b_2*b_3	4	5
17	s_8	b_2*s_0	7	10
18	s_9	b_0*b_3	4	5
19	s_{10}	b_0*b_5	4	5
20	s_{11}	b_0*b_7	4	5
21	s_{12}	b_0*s_0	7	10
22	s_{13}	b_1*b_2	4	5
23	s_{14}	b_2*b_5	4	5
24	s_{15}	b_2*b_7	4	5
25	s_{16}	b_0*b_1	4	5
26	s_{17}	b_0*s_1	7	10

列数	元素	表达式	最短路径	
			XOR	AND
27	s_{18}	b_0*s_2	7	10
28	s_{19}	b_0*s_3	7	10
29	s_{20}	b_0*s_5	7	10
30	s_{21}	b_0*s_6	7	10
31	s_{22}	b_1*b_3	4	5
32	s_{23}	b_1*b_4	4	5
33	s_{24}	b_1*b_5	4	5
34	s_{25}	b_1*b_7	4	5
35	s_{26}	b_1*s_0	7	10
36	s_{27}	b_2*s_1	7	10
37	s_{28}	b_2*s_2	7	10
38	s_{29}	b_2*s_3	6	10
39	s_{30}	b_3*b_4	4	5
40	s_{31}	b_3*b_5	4	5
41	s_{32}	b_3*b_6	4	5
42	s_{33}	b_3*b_7	4	5
43	s_{34}	b_3*s_0	7	10
44	s_{35}	b_3*s_4	6	10
45	s_{36}	b_4*s_4	6	10
46	s_{37}	b_4*s_{13}	7	10
47	s_{38}	b_5*b_7	4	5
48	s_{39}	b_5*s_4	6	10
49	s_{40}	b_6*s_4	6	10
50	s_{41}	b_7*s_0	7	10
51	s_{42}	s_0*s_4	7	10

附录 4 将[x,x^2]作为公共项提取公共项

将[x,x^2]作为公共项进行优化提取公共项时，所生成的 73×138 列矩阵，每列所代表的公共项及其最短路径如下表。

列数	元素	表达式	最短路径	
			XOR	AND
1	b_0	x	0	0
2	b_1	x^2	0	0
3	b_2	r_0	0	0
4	b_3	r_0^2	1	0
5	b_4	r_1	0	0
6	b_5	r_1^2	1	0
7	b_6	r_2	0	0
8	b_7	r_2^2	1	0
9	b_8	r_3	0	0
10	b_7	r_3^2	1	0
11	s_0	b_1*b_6	1	0
12	s_1	b_1*b_2	1	0
13	s_2	b_8*s_0	4	5
14	s_3	b_8*b_9	4	5
15	s_4	b_0*b_6	2	0
16	s_5	b_4*b_5	4	5
17	s_6	b_4*s_0	4	5
18	s_7	b_7*b_8	4	5
19	s_8	b_2*s_4	5	5
20	s_9	b_3*b_8	4	5
21	s_{10}	b_4*b_7	4	5
22	s_{11}	b_4*b_8	3	5
23	s_{12}	b_7*s_0	4	5
24	s_{13}	b_0*s_3	6	5
25	s_{14}	b_1*s_3	5	5
26	s_{15}	b_2*b_5	4	5
27	s_{16}	b_2*s_2	7	10
28	s_{17}	b_4*b_9	2	0
29	s_{18}	b_4*s_2	6	10
30	s_{19}	b_9*s_0	4	5

列数	元素	表达式	最短路径	
			XOR	AND
31	s_{20}	b_9*s_1	4	5
32	s_{21}	b_0*b_3	2	0
33	s_{22}	b_0*b_9	2	0
34	s_{23}	b_1*s_5	5	5
35	s_{24}	b_1*s_7	5	5
36	s_{25}	b_1*s_9	5	5
37	s_{26}	b_1*s_{10}	5	5
38	s_{27}	b_1*s_{17}	3	0
39	s_{28}	b_2*b_3	4	5
40	s_{29}	b_2*b_7	4	5
41	s_{30}	b_2*s_6	7	10
42	s_{31}	b_2*s_{13}	9	10
43	s_{32}	b_3*b_4	4	5
44	s_{33}	b_3*b_5	4	5
45	s_{34}	b_3*b_6	4	5
46	s_{35}	b_3*b_7	4	5
47	s_{36}	b_3*b_9	4	5
48	s_{37}	b_3*s_2	7	10
49	s_{38}	b_3*s_6	7	10
50	s_{39}	b_4*s_{13}	9	10
51	s_{40}	b_5*b_6	4	5
52	s_{41}	b_5*b_7	4	5
53	s_{42}	b_5*b_8	4	5
54	s_{43}	b_5*b_9	4	5
55	s_{44}	b_5*s_2	7	10
56	s_{45}	b_6*b_7	4	5
57	s_{46}	b_6*b_9	4	5
58	s_{47}	b_7*b_9	4	5
59	s_{48}	b_7*s_1	4	5
60	s_{49}	b_7*s_6	7	10

列数	元素	表达式	最短路径	
			XOR	AND
61	s_{50}	$b_7 * s_8$	8	10
62	s_{51}	$b_9 * s_2$	7	10
63	s_{52}	$b_9 * s_4$	5	5
64	s_{53}	$b_9 * s_6$	7	10
65	s_{54}	$b_9 * s_8$	8	10
66	s_{55}	$s_1 * s_5$	7	10
67	s_{56}	$s_1 * s_7$	7	10
68	s_{57}	$s_1 * s_9$	7	10
69	s_{58}	$s_1 * s_{10}$	7	10
70	s_{59}	$s_1 * s_{11}$	6	10
71	s_{60}	$s_4 * s_{11}$	6	10
72	s_{61}	$s_8 * s_{11}$	8	10
73	s_{62}	$b_9 * s_{21}$	5	5