第4章

エラー訂正や暗号処理で使われる演算回路を極める

――ガロア体を回路で実現するには…

森岡澄夫

近年，CRC(cyclic redundancy code)やリードソロモン符号などのエラー訂正機能はもちろん，AES(advanced encryption standard)などのセキュリティ機能も通信処理用チップに搭載されることが多くなっている．これらには「ガロア体」という数体系が使用されている．ここでは，最低限必要な数学的知識を交えながら，ガロア体演算を回路化するための考えかたを解説する．　（編集部）

ガロア体というのは日ごろ聞きなれないことばであり，読者のみなさんとは何の関係もない特殊なものに思われるかもしれません．しかし，CRC（cyclic redundancy code）のようなエラー検出処理，SDRAMや通信で用いられるエラー訂正処理，AES（advanced encryption standard）のような暗号処理など，身近で使われている数体系なのです．最近ではエラー検出・訂正回路や暗号処理回路を組む機会も増えています．ここでは難しい数学の話は最低限に抑え，ガロア体の演算回路をどう組むかという，従来あまり紹介されてこなかった点にフォーカスして説明します（数学的用語はp.60のコラム「とてもやさしい（？）用語解説」を参照）．

① 公開かぎや共通かぎに使われるガロア体演算

まず初めに，ガロア体とはいったい何で，どこにどう使われているのかを紹介します．

●四則演算を行える数の集合が「ガロア体」

有限個の数の集合で，その数の間に四則演算（＋－×÷）が定義できるものをガロア体（Galois field）と言います．数学上は有限体（finite field）と言うほうが適切ですが，工学系ではガロア体と言うほうがとおりがよいようです（**図1**）．

要素の数がp個のガロア体を$GF(p)$と書きます．ここでpの値は素数か，または素数のべき乗の値でなければなりません．例えば，$GF(2)$，$GF(3)$，$GF(2^2)$，$GF(5)$，$GF(7)$，$GF(2^3)$，$GF(3^2)$などは存在しますが，$GF(6)$や$GF(10)$は存在しません．また，要素数が素数のべき乗p^m（$m \geqq 2$）である場合，その体を$GF(p)$の拡大体と言います（数学的に厳密な説明は稿末の参考文献1）などを参照）．拡大体は普通$GF(2^2)$，$GF(2^3)$などと表記し，$GF(4)$，$GF(8)$とはあまり書きません．$GF(p)$（$m=1$の場合）は素体と言います．

〔図1〕
ガロアが考えたこと
有限体は，ガロア理論の基礎に出てくるものである．歴史的にガロア理論は，5次以上の方程式に解の公式がないことや，コンパスと定規で角の3等分ができないという代数の古典的定理を証明するところから生まれている．それゆえ，有限体の理論は方程式（多項式）と密接にからんでいる．

●いろいろなガロア体とそのアプリケーション

ガロア体の主なアプリケーションは，エラー検出符号，エラー訂正符号，共通かぎ暗号，および公開かぎ暗号です．

どのガロア体がどのアプリケーションに使われているかを**表1**に示します．詳細は後述しますが，ここでは$GF(p)$の要素が$\log_2 p$ビットの値であること（例えば$GF(2^4)$の要素は4ビット，$GF(2^8)$の要素は8ビット）を知っていただければ，どのような場合にどのようなガロア体が使われるかが，なんとなく見えてくるでしょう．

1）エラー検出・訂正に使われるガロア体

CRCなどのエラー検出符号や，SDRAMで用いられるSEC-DED符号（single error correction-double error detection），ハミング符号，リードソロモン符号といったエラー訂正符号では，p^mが2のべき乗のガロア体，すなわち$GF(2)$，$GF(2^2)$，$GF(2^4)$，$GF(2^8)$，$GF(2^{16})$がよく用いられます．これらの符号を使う局面では，ビット・エラーやニブル（4ビット）エラー，バイト・エラーなどを検出・訂正したい場合が多く，そのため入力データをビット，ニブル，バイト，ワード単位で区切って計算を行うことになります．もしワード単位で区切れば$GF(2^{16})$，バイト単位で区切れば$GF(2^8)$が使われる，といったぐあいです．

ここで，例えばリードソロモン符号を使って128ビットの入力データ中のエラー訂正を行いたいとします．このとき，そのデータをバイト単位で区切って8ビット×16とみなしても，ニブル単位で区切って4ビット×32とみなしても別に同じではないかと思われるかもしれません．しかし，同じリードソロモン符号で同じデータ長であっても，区切りかたによって必要な演算の数（計算量）やパリティ長などが変わってきます．こういった点を考慮して，どのガロア体を使うかを選定します．

2）共通かぎ暗号に使われるガロア体

AESやCamellia（参考文献8））など最近の共通かぎ暗号の多くでは，その内部にあるS-Boxという主要な演算処理に$GF(2^8)$の逆元演算（詳細は後述）が使われています．

3）公開かぎ暗号に使われるガロア体

公開かぎ暗号のうち「だ円暗号」と呼ばれるものでは，p^mの値が2^{160}以上と大きな体が使われています．

このようなエラー訂正処理や暗号処理は，身の周りに必ずといっていいほど使われています．ですから，ガロア体というのは特殊なようで，実は日常的にお世話になっているものなのです．本稿では，みなさんが扱う機会があると思われる，上記の**1**）と**2**）を対象とします．以下では特に断らないかぎり，$GF(2^m)$でmが16程度までのものを扱います．

●ガロア体の数の表しかた

ここではガロア体の演算がどのようなものであるかを紹介します．「四則演算を行える数の集まり」と前述しましたが，ガロア体は整数とはいろいろと異なる点があります．

整数では，数を0，1，2，3，…と表記します．一方，ガロア体$GF(2^m)$の数は，特別な記号αを使って0，1，α，α^2，α^3，…，$\alpha^{(2^m-2)}$と表します（**図2**，これを「べき表現」という）．0と1は整数の場合と同じと考えてかまいません[注1]．αは原始元（primitive element）と呼ばれるものです．ただし，$GF(2)$の場合には要素は0，1の2個しかありません．

さて，整数演算では，数0，1，2，…をどのように2進表現するかが回路を組むうえで，問題となりますが，これはガロア体でも同じです．基本的にはガロア体であっても数0，1，α，α^2，…と2進値の間のマッピングは好きなように決めてかまいません．しかし，普通は$GF(2^m)$の値はmビットで表現しますし，限られたマッピングしか用いません．ここで覚えておいていただきたいのは，標準的なマッピングのもとでは，**図2**（a）に示すように以下の三つによって2進値が決定されるということです．

- 体の要素数2^m
- 既約多項式（irreducible polynomial）：符号理論のテキストを見ると一覧表が載っている．ほとんどの場合，原始既約多項式（primitive irreducible polynominal）と呼

〔**表1**〕いろいろなガロア体とそのアプリケーション

エラー検出符号やエラー訂正符号では$GF(2^m)$（m＝1～16）が使われる．$GF(2^m)$はmビットの値を表しており，例えばバイト単位でエラー訂正を行いたいのなら，$GF(2^8)$が使われることになる．また，最近の共通かぎ暗号では$GF(2^8)$が使われている．だ円暗号などの一部の公開かぎ暗号では，$GF(p^m)$で$p^m=2^{160}$以上のものが使われる．

アプリケーション	使われる体
エラー検出符号（CRCなど）	$GF(2)$
弱いエラー訂正符号 （ハミング符号やSEC-DEDなど）	$GF(2)$，$GF(2^m)$ $m=2\sim4$
比較的強いエラー訂正符号 （リードソロモン符号など）	$GF(2^m)$ $m=4\sim16$
共通かぎ暗号（AES，Camellia など，最近提案されているもの）	$GF(2^8)$
だ円暗号（公開かぎ暗号の一種）	$GF(p^m)$ $p^m \geqq 2^{160}$

注1：‘0’はいわゆる零元，‘1’はいわゆる乗法単位元である．また，本稿では乗法逆元を単に逆元と略して記す．