プロジェクト実習1 D03回 RS符号

山本担当分

有理数係数多項式の乗算

- x^2+2x-1 の係数
  - 2次: 1, 1次: 2, 0次(定数項): -1
- 有理数係数の多項式の計算ならなじみがある
  - 有理数:整数 m とゼロでない整数 n で m/n として表せる数

式の乗算: $(x^2+2x-1)(x+3)=$

係数は乗算と加算しかしないので
式どうしの乗算は有理数係数の
多項式になる

乗算について閉じている

	x^2	$+2x$	-1
$\times$		x	$+3$
	$3x^2$	$+6x$	-3
x^3	$+2x^2$	$-x$	
x^3	$+5x^2$	$+5x$	-3

有理数係数多項式の除算

係数は除算,減算を使う
有理数は割り算,
引き算しても有理数

						$\frac{1}{1}x$	$\frac{6}{1}$	
						x	$+6$	商
x^2	$-x$	$+1$	)	x^3	$+5x^2$	$+3x$	-1	
				x^3	$-x^2$	$+x$		除数x 1/1
					$6x^2$	$+2x$	-1	
					$6x^2$	$-6x$	$+6$	除数x 6/1
						$8x$	-7	剰余

除算についても閉じている

例：GF(2^3)

ベクトル表現

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

- 元は 2 次以下の（係数が{0,1}の素体の）多項式全て
  - 0
  - 1
  - x
  - x+1
  - x^2
  - x^2 +1
  - x^2+x
  - x^2+x+1

この 8 個が GF(2^3)の元

3ビットの全パターン（2次の係数,1次の係数,0次の係数）

GF(2^3)の加算

+	0	1	x	x+1	x^2	x^2+1	x^2+x	x^2+x+1
0	0	1	x	x+1	x^2	x^2+1	x^2+x	x^2+x+1
1	1	0	x+1	x	x^2+1	x^2	x^2+x+1	x^2+x
x	x	x+1	0	1	x^2+x	x^2+x+1	x^2	x^2+1
x+1	x+1	x	1	0	x^2+x+1	x^2+x	x^2+1	x^2
x^2	x^2	x^2+1	x^2+x	x^2+x+1	0	1	x	x+1
x^2+1	x^2+1	x^2	x^2+x+1	x^2+x	1	0	x+1	x
x^2+x	x^2+x	x^2+x+1	x^2	x^2+1	x	x+1	0	1
x^2+x+1	x^2+x+1	x^2+x	x^2+1	x^2	x+1	x	1	0

加算はベクトル表現の排他的論理和

GF(2^3)の乗算

×	0	1	x	x+1	x^2	x^2+1	x^2+x	x^2+x+1
0	0	0	0	0	0	0	0	0
1	0	1	x	x+1	x^2	x^2+1	x^2+x	x^2+x+1
x	0	x	x^2	x^2+x	x+1	1	x^2+x+1	x^2+1
x+1	0	x+1	x^2+x	x^2+1	x^2+x+1	x^2	1	x
x^2	0	x^2	x+1	x^2+x+1	x^2+x	x	x^2+1	1
x^2+1	0	x^2+1	1	x^2	x	x^2+x+1	x+1	x^2+x
x^2+x	0	x^2+x	x^2+x+1	1	x^2+1	x+1	x	x^2
x^2+x+1	0	x^2+x+1	x^2+1	x	1	x^2+x	x^2	x+1

（多項式の乗算をして原始多項式 $p(x) = x^3 + x + 1$ で割った余り）

GF(2^3)の乗算（べき表現）

- $p(x) = x^3 + x + 1$ で割った余り（2次式）が元全てである世界
- α を多項式 x とする
- $\alpha^0 = 1, \alpha^1 = x, \alpha^2 = x^2$
- $\alpha^3 = x^3$ を $p(x)$ で割った余り $= x + 1$
- $\alpha^4 = \alpha^3 \times \alpha = x^2 + x$
- $\alpha^5 = \alpha^4 \times \alpha = x^3 + x^2$ を $p(x)$ で割った余り $= x^2 + x + 1$
- $\alpha^6 = \alpha^5 \times \alpha = x^3 + x^2 + x$ を $p(x)$ で割った余り $= x^2 + 1$
- $\alpha^7 = 1$（α^0に戻る）
  - α^0 の逆元は α^0, $1 \leq i \leq 6$の α^i の逆元は α^{7-i}

GF(2^3)の原始限 α

- $\alpha^0 = 1, \alpha^1 = x, \alpha^2 = x^2, \alpha^3 = x + 1$
- $\alpha^4 = x^2 + x, \alpha^5 = x^2 + x + 1, \alpha^6 = x^2 + 1$
- 0 以外の元は α^0 から α^6 で表せる（べき表現）
- α を原始元という
- 多項式 0 はαのべき乗ではできない
  - 多項式 0 の元のみべき表現できない
- 多項式0の元以外はベクトル表現とべき表現に一対一対応

べき→ベクトル

α^0	α^1	α^2	α^3	α^4	α^5	α^6
001	010	100	011	110	111	101
1	x	x2	x+1	x^2+x	x^2+x+1	x^2+1

べき表現によるGF(2^3)の乗算

×	(0)	α^0 (1)	α^1 (x)	α^2 (x^2)	α^3 (x+1)	α^4 (x^2+x)	α^5 (x^2+x+1)	α^6 (x^2+1)
多項式0	多項式0	多項式0	多項式0	多項式0	多項式0	多項式0	多項式0	多項式0
α^0	多項式0	α^0	α^1	α^2	α^3	α^4	α^5	α^6
α^1	多項式0	α^1	α^2	α^3	α^4	α^5	α^6	α^0
α^2	多項式0	α^2	α^3	α^4	α^5	α^6	α^0	α^1
α^3	多項式0	α^3	α^4	α^5	α^6	α^0	α^1	α^2
α^4	多項式0	α^4	α^5	α^6	α^0	α^1	α^2	α^3
α^5	多項式0	α^5	α^6	α^0	α^1	α^2	α^3	α^4
α^6	多項式0	α^6	α^0	α^1	α^2	α^3	α^4	α^5

指数の加算（mod 7）でできるので乗算はべき表現のほうが良い（多項式0は表現できない）

GF(2^3)の演算では

- 加減算はベクトル表現，乗除算はべき表現が向いている
- プログラムを書く場合，ベクトル表現，べき表現の互いの変換方法があればいい
- 配列でできる

べき→ベクトル

α^0	α^1	α^2	α^3	α^4	α^5	α^6
001	010	100	011	110	111	101

ベクトル→べき

000	001	010	011	100	101	110	111
	α^0	α^1	α^3	α^2	α^6	α^4	α^5

元をコンピュータに向いた数値で表現

べき表現の数値表現

本来の元	1	x	x^2	x+1	x^2+x	x^2+x+1	x^2+1
べき表現	α^0	α^1	α^2	α^3	α^4	α^5	α^6
数値表現	0	1	2	3	4	5	6

ベクトル表現の数値表現

本来の元	0	1	x	x+1	x^2	x^2+1	x^2+x	x^2+x+1
ベクトル表現	000	001	010	011	100	101	110	111
数値表現	0	1	2	3	4	5	6	7

数値表現で変換を配列で実装

べき→ベクトル

α^0	α^1	α^2	α^3	α^4	α^5	α^6
001	010	100	011	110	111	101

ベクトル→べき

000	001	010	011	100	101	110	111
	α^0	α^1	α^3	α^2	α^6	α^4	α^5

べき→ベクトル 配列 p2v

p2v[0]	p2v[1]	p2v[2]	p2v[3]	p2v[4]	p2v[5]	p2v[6]
1	2	4	3	6	7	5

ベクトル→べき 配列 v2p

v2p[0]	v2p[1]	v2p[2]	v2p[3]	v2p[4]	v2p[5]	v2p[6]	v2p[7]
	0	1	3	2	6	4	5

コーディング

- ベクトル→べきの変換配列 v2p とべき→ベクトルの変換配列 p2v のどちらが簡単に計算できるか
- 片方ができればそれを使って逆を計算できる
- $\alpha^0, \alpha^1, \ldots, \alpha^6$ の多項式を順に乗算で計算すれば p2v はできる
  - $\alpha^0 = 1, \alpha^1 = x, \alpha^2 = x^2, \alpha^3 = x^3 = x + 1, \alpha^4 = \alpha^3 \times \alpha = x^2 + x$
  - $\alpha^5 = \alpha^4 \times \alpha = x^3 + x^2 = x^2 + x + 1, \alpha^6 = \alpha^5 \times \alpha = x^3 + x^2 + x = x^2 + 1$
- v2p[p2v[0]]=0, v2p[p2v[1]]=1, v2p[p2v[2]]=2, … とすれば v2p を作れる

p2vの作り方

- α^0は1なのでp2v[0]=1（多項式1のベクトル表現の数値表現）
- α^1は$\alpha^0 \times \alpha^1$ つまり今の α^0の式にxを掛ける
  - 字数が一つ上がる $x \rightarrow x^2$
  - ベクトル表現では 010→100
  - x を掛けることはビット演算で左シフトに相当
  - ただし，3ビットに収まらなくなったときは 3 次式になったときである
    - $p(x) = x^3+x+1$ で割ったあまりにする
    - 4ビットでいうと 1011 と XOR
    - シフトする前に最上位ビットが1ならシフトして011とXORすればいい
    - そうでなければシフトするだけでいい

バージョン2，誤り訂正能力MのQRコード

- GF(2^8)を使う
  - $p(x) = x^8+x^4+x^3+x^2+1$ が指定されている
  - $p(x)$ で割った余りは高々7次式→2^8通り，元は 256 個
- この元を係数とする多項式の加算減算乗算除算が必要
- サイズ 256 のべき→ベクトルの変換配列とベクトル→べきの変換配列を作る
- 多項式の係数（GF(2^8)）の加算乗除算のルーチンを作る
  - XORなので減算は加算と同じ
- 多項式の加減乗除算，剰余計算のルーチンを作る

課題（第3回授業）

- プログラムはホームディレクトリのpp1直下　report01.c
- 2個の関数を作る
  - unsigned char add_c_v(unsigned char cf_v1, unsigned char cf_v2)
  - unsigned char mlt_c_p(unsigned char cf_p1, unsigned char cf_p2)
- add_c_v は 2 個のベクトル表現の GF(2^8) の要素を引数として，それらを足した結果をベクトル表現で返す関数
- mlt_c_p は 2 個のべき表現の GF(2^8) の要素を引数として，それらを掛けた結果をべき表現で返す関数
- メインルーチンは自由

配列と多項式

- 各要素が unsigned char である配列
- unsigned char の変数 1 個はGF(2^8)の要素（多項式の係数）
- 配列で多項式を表せる
  - 例：$5x^3 + x$ は 0000 5010
  - i バイトが i-1次式に対応

生成多項式

- 情報が k=28バイト，符号語全体が n=44バイト
- 情報は27次式 $i(x)$，符号語は43次式 $c(x)$
  - 生成多項式が決まっている
  - $g(x) = \prod_{i=0}^{n-k-1}(x - \alpha^i)$
  - 今回の符号では

$$g(x) = x^{16} + \alpha^{120}x^{15} + \alpha^{104}x^{14} + \alpha^{107}x^{13} + \alpha^{109}x^{12} + \alpha^{102}x^{11} + \alpha^{161}x^{10} + \alpha^{76}x^{9} + \alpha^{3}x^{8} + \alpha^{91}x^{7} + \alpha^{191}x^{6} + \alpha^{147}x^{5} + \alpha^{169}x^{4} + \alpha^{182}x^{3} + \alpha^{194}x^{2} + \alpha^{225}x + \alpha^{120}$$

  - 43次式のうち16次式 $g(x)$で割り切れるものを符号語とする

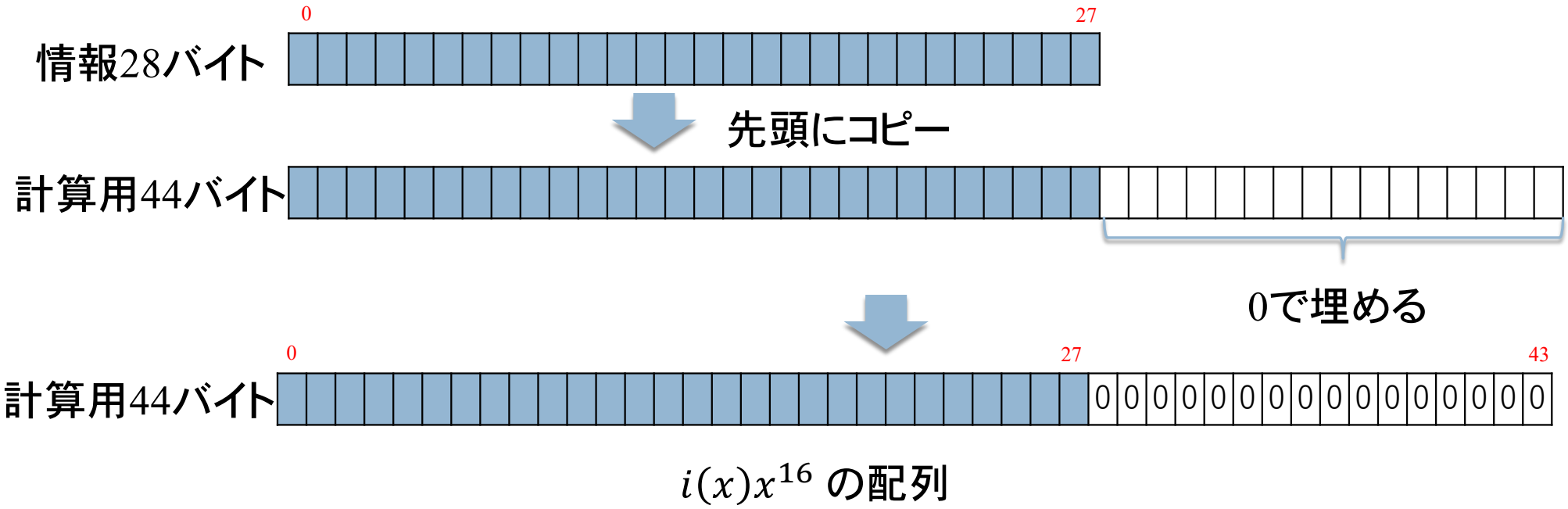
符号化の考え方

- 情報28バイト（27次式）を$g(x)$で割り切れる43次式に変換したい
- 情報28バイトの27次式を情報多項式 $i(x)$とする
- $i(x)x^{n-k} = i(x)x^{16}$ は 43 次式
  - しかし $g(x)$で割りきれるとはかぎらない
  - $g(x)$で割って得た余りの式$r(x)$を最初に足しておけば割り切れる
  - その多項式を符号語とする

符号化計算

- $i(x)x^{n-k}$を $g(x)$ で割った余りの式 $r(x)$を求める
- $c(x) = i(x)x^{n-k} + r(x)$ を符号語多項式とする
- $c(x)$ の係数の配列が符号語

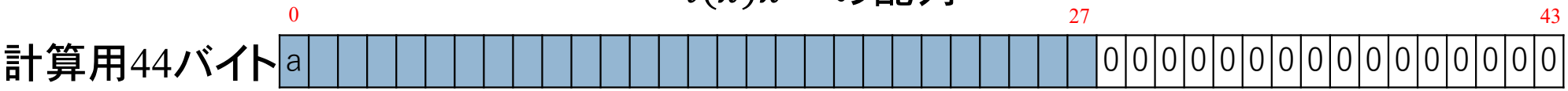
$i(x)x^{16}$ を作る

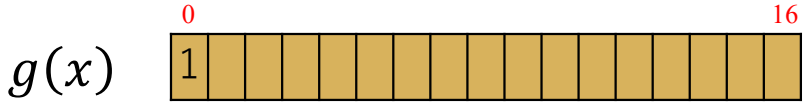


$i(x)x^{16}$ の配列

$g(x)$ で割った余りを作る(1)

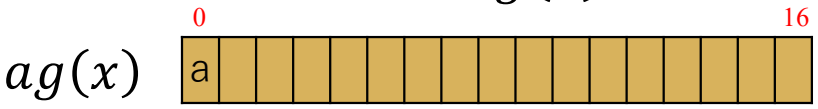
$i(x)x^{16}$ の配列



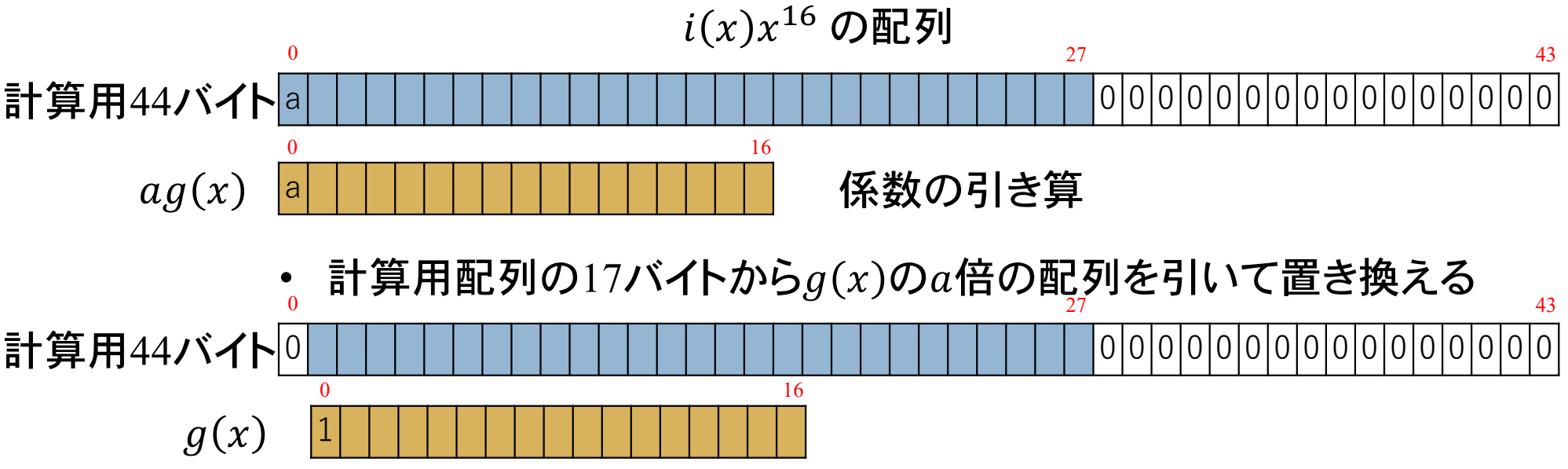


（先頭の係数は必ず1）

- 計算用配列の先頭と $g(x)$ の配列の先頭を合わせる
- $g(x)$の先頭と合っている計算用配列の要素が0のときは何もしない
- aだとすると$g(x)$全体をa倍した配列を作る



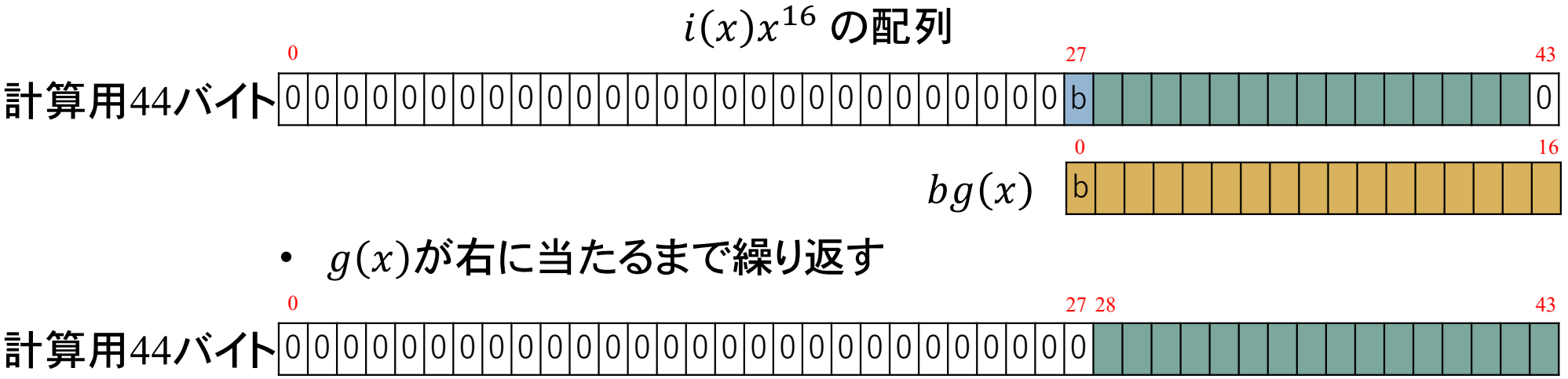
$g(x)$ で割った余りを作る(2)



- 計算用配列の17バイトから$g(x)$のa倍の配列を引いて置き換える

- 最高次の項が消える
- $g(x)$を右に1ずらして同じことを繰り返す。

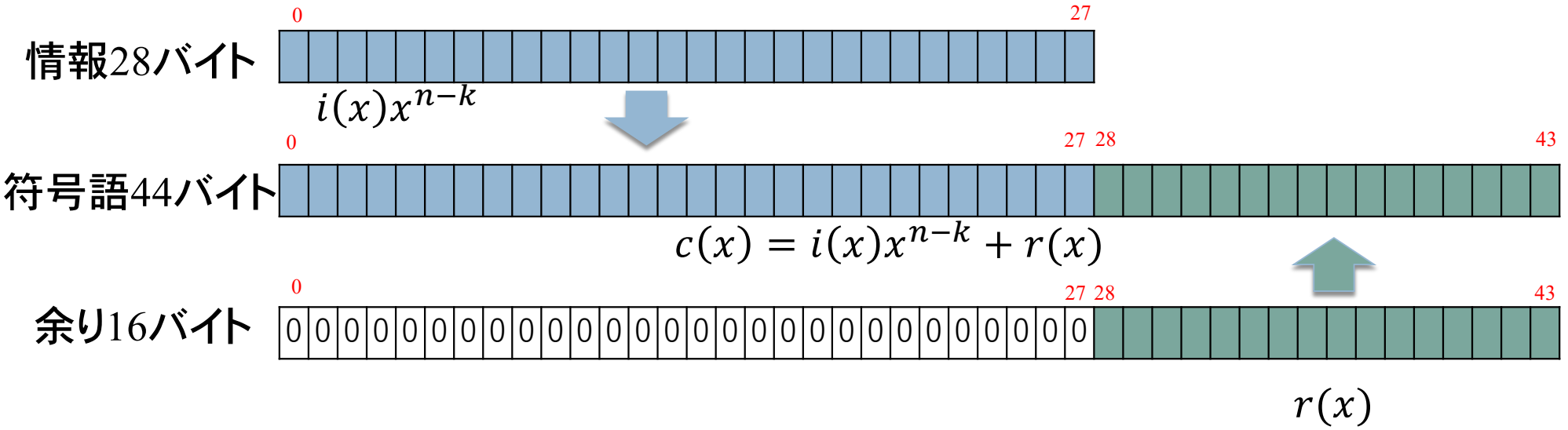
$g(x)$ で割った余りを作る(3)



- $g(x)$が右に当たるまで繰り返す

- 残った16バイト（15次式）が $r(x)$ になる（$g(x)$が16次式だから）

符号語$c(x)$を作る


情報28バイト
0
27
$i(x)x^{n-k}$
符号語44バイト
0
27 28
43
$c(x) = i(x)x^{n-k} + r(x)$
余り16バイト
0
27 28
43
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$r(x)$