

コンピュータ・サイエンス1

第6回

コンピュータでの情報の扱い方(3)

人間科学科コミュニケーション専攻

白銀 純子

第6回の内容

- コンピュータでの情報の扱い方(3)

前回の出席問題の解答

- 設問1: 12個の情報を表現するためには、最低何ビットの0と1が必要か

nビットで 2^n 個の情報を表現できるので...

➤ $2^2 = 4$	➡ 足りない!
➤ $2^3 = 8$	
➤ $2^4 = 16$	➡ 足りる!

解答: 4ビット

前回の出席問題の解答

- 設問2: 「やってみよう!」の10進数 $\leftrightarrow$ 2進数の変換問題の4. と6. の計算結果を報告すること
 - 4. の問題: 10進数の「135」を10桁の2進数に

$$\begin{array}{r} 2 \overline{) 135} \\ 2 \overline{) 67} \cdots \text{余り: } 1 \\ 2 \overline{) 33} \cdots \text{余り: } 1 \\ 2 \overline{) 16} \cdots \text{余り: } 1 \\ 2 \overline{) 8} \cdots \text{余り: } 0 \\ 2 \overline{) 4} \cdots \text{余り: } 0 \\ 2 \overline{) 2} \cdots \text{余り: } 0 \\ 2 \overline{) 1} \cdots \text{余り: } 0 \\ 0 \cdots \text{余り: } 1 \end{array}$$



$$(135)_{10} = (10000111)_2 \xrightarrow{\text{「10桁」の指定がついているので...}} \text{解答: } (0010000111)_2$$

前回の出席問題の解答

- 設問2: 「やってみよう!」の10進数 $\Leftrightarrow$ 2進数の変換問題の4. と6. の計算結果を報告すること
 - 6. の問題: 2進数の「001010101010」を10進数に

2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
×	×	×	×	×	×	×	×	×	×	×	×
0	0	1	0	1	0	1	0	1	0	1	0

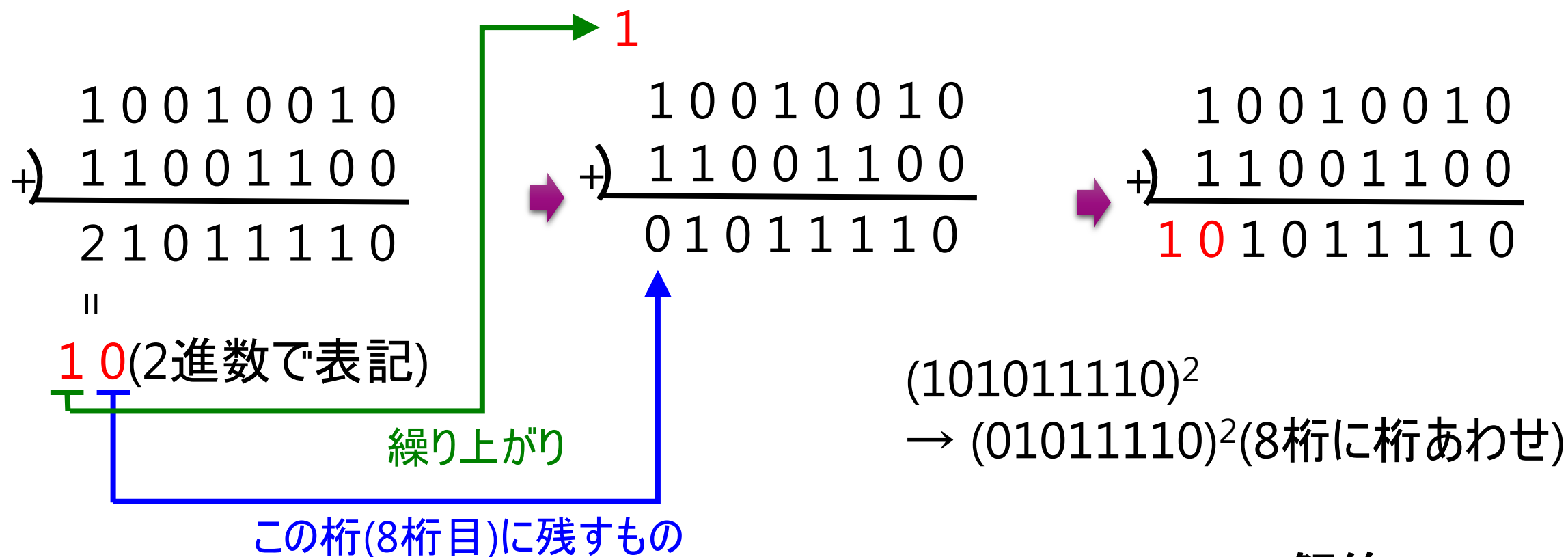


$$\begin{aligned} & 0 + 0 + 2^9 + 0 + 2^7 + 0 + 2^5 + 0 + 2^3 + 0 + 2^1 + 0 \\ &= 512 + 128 + 32 + 8 + 2 \\ &= 682 \end{aligned}$$

解答: $(682)_{10}$

前回の出席問題の解答

- 設問3: 「やってみよう!」の2進数の足し算の3. の計算結果を報告すること
 - 3. の問題: $10010010 + 11001100$ の結果を10進数で(8ビットでオーバーフローも考慮)



解答: $(94)_{10}$

前回の質問の解答

- 前回の復習とあわせて

Question!

前回の復習

2進数[1](p. 4)

- n進数: 数をn個の文字で表す方法
 - 10進数: 数を10個の文字で表す方法(普段使っている数の表現方法)
 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9の10個の文字
 - 2進数: 数を2個の文字で表す方法
 - 0, 1の2個の文字

コンピュータ: 「0」と「1」で全ての情報を表現

➡ 「2進数で情報を表現している」、と言える

2進数での情報の表現

- 数: 10進数 $\Leftrightarrow$ 2進数の計算が可能
- 文字や色: 人間が番号を設定

数

- $(50)_{10} = (110010)_2$
 - $(100)_{10} = (1100100)_2$
- } 計算することが可能

文字

- 半角アルファベットの「N」: 01001110
- 日本語の「ん」: 1010010011110011

色

- 赤: 111111110000000000000000
- 青: 0000000000000000000011111111

} 人間が番号のつけ方の
ルールを作成

10進数を2進数に変換

- 10進数の数は、2進数の表現に直すことができる

2) 10進数の数
2) 商1...余り1
2) 商2...余り2
.
.
.
2) 商n-1...余りn-1
商n...余りn

1. 10進数の数を2で割って商1と余り1を計算する
2. 商1を2で割って商2と余り2を計算する
3. 商2を2で割って商3と余り3を計算する
4.

商が0になるまで繰り返す
※小数の計算はしない

余り1
余り2
.
.
.

時計回りに倒す

余りを余りnから余り1の順に
左から並べたものが2進数

余りn-1
余りn

余り
2
余り
1
.
.
.

10進数を2進数に変換(例)

10進数の13を2進数に変換

$$\begin{array}{r} 2 \overline{) 13} \\ 2 \overline{) 6} \cdots \text{余り: } 1 \\ 2 \overline{) 3} \cdots \text{余り: } 0 \\ 2 \overline{) 1} \cdots \text{余り: } 1 \\ 0 \cdots \text{余り: } 1 \end{array}$$



$$(13)_{10} = (1101)_2$$

10進数の50を2進数に変換

$$\begin{array}{r} 2 \overline{) 50} \\ 2 \overline{) 25} \cdots \text{余り: } 0 \\ 2 \overline{) 12} \cdots \text{余り: } 1 \\ 2 \overline{) 6} \cdots \text{余り: } 0 \\ 2 \overline{) 3} \cdots \text{余り: } 0 \\ 2 \overline{) 1} \cdots \text{余り: } 1 \\ 0 \cdots \text{余り: } 1 \end{array}$$



$$(50)_{10} = (110010)_2$$

※矢印の方向に余りを並べる

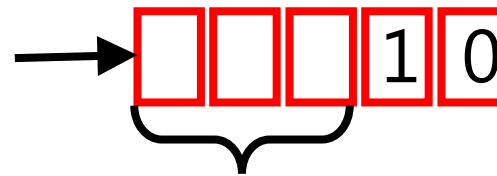
2進数の桁数[1]

- コンピュータでは、2進数の各桁の0と1をそれぞれ箱に入れて扱うイメージ
 - 箱の数は32または64が多い

箱の数が5個のコンピュータだとすると... $(2)_{10} = (10)_2$

各桁を箱に入れると...

(各桁は右詰めで入れる)



箱が余ることもある!



余った箱には「0」を入れる

➤ つまり $(10)_2$ を $(00010)_2$ と表現する

コンピュータの世界で数を2進数で表現するとき:
数の左側に、足りない分だけ0をつけて表現する

※授業や書籍の説明、問題などでは32桁や64桁は書ききれないので、もっと短い桁数で扱う

2進数の桁数[2]

- コンピュータ関係の問題では、2進数の桁数を指定されることが多い
 - Ex. 10進数の50を8桁の2進数で表しなさい。

$(50)_{10} = (110010)_2$

➡ 答え: $(110010)_2$ ❌

もらえても△
(つまり、○はもらえない)

なぜ? → 問題文に「8桁」と指定されているのに、6桁で答えているから

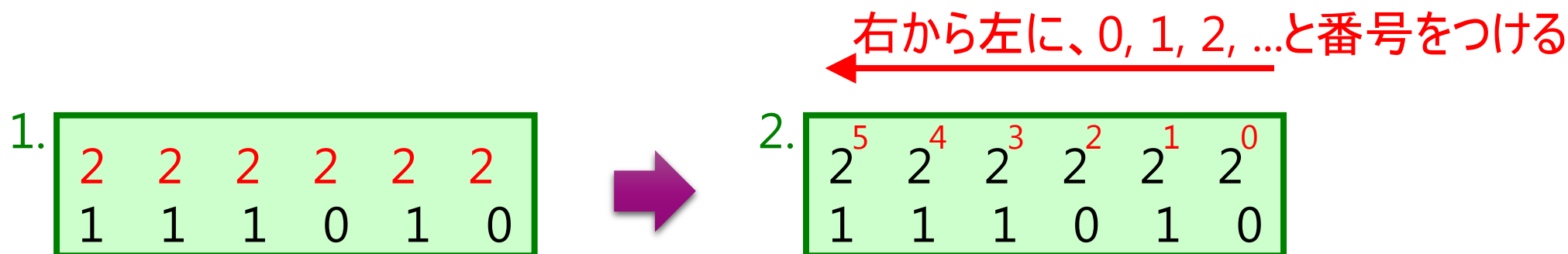
➡ 答え: $(00110010)_2$ ○

桁数を気にしなければならない場面は多いので注意!

※問題文に桁数が書いていなければ気にしなくてOK

2進数を10進数に変換

1. 2進数の各桁の上にそれぞれ「2」を書く
2. 1. で書いた「2」の右肩に、右から0, 1, 2, ...と書いていく
 - $2^0, 2^1, 2^2, \dots$ ができていく



※ 2^n : 2をn回かけ算する

➤ Ex. 2^3 : $2 \times 2 \times 2 = 8$

2進数を10進数に変換

3. 各桁の上の「 2^n 」と、それぞれの桁の数をかけあわせる
4. 2. の結果を足し合わせる

2.

2^5	2^4	2^3	2^2	2^1	2^0
1	1	1	0	1	0



3.

2^5	2^4	2^3	2^2	2^1	2^0
×	×	×	×	×	×
1	1	1	0	1	0

||

2^5 2^4 2^3 0 2^1 0



4.

2^5	2^4	2^3	0	2^1	0
-------	-------	-------	---	-------	---

足し合わせる
||

$2^5 + 2^4 + 2^3 + 0 + 2^1 + 0 = 58$

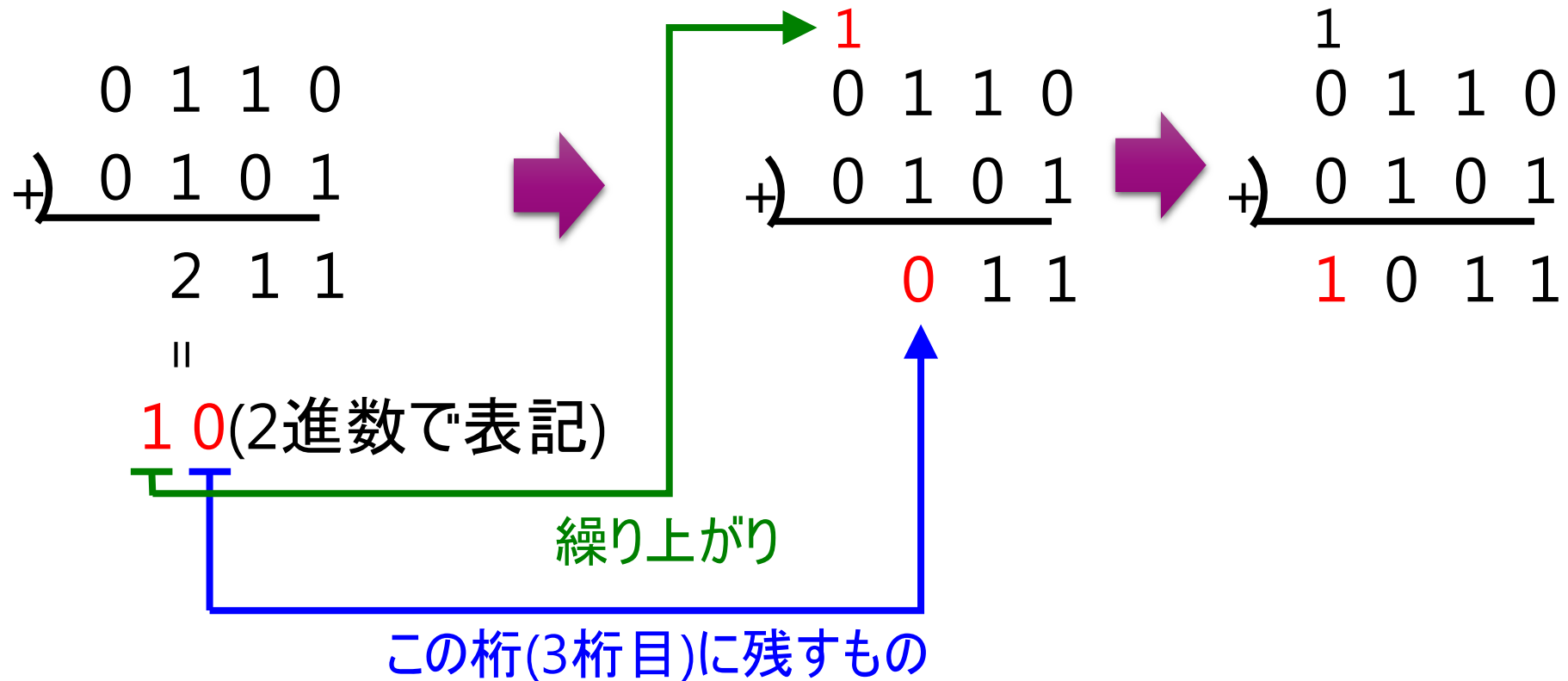
足し算をする方法[1](p. 6)

- 10進数での1桁の足し算
 - たくさん($10 \times 10 = 100$)のパターンが存在
 - $1+1, 1+2, 1+3, \dots 2+1, 2+2, 2+3, \dots \dots 8+6$ (繰り上がり1), $8+7$ (繰り上がり1),
... ..
- 2進数での1桁の足し算
 - 4通り
 - 足した結果が2になると繰り上がり1(2進数では10進数の2を「10」と表すため)
 - $0+0, 0+1, 1+0, 1+1$ (繰り上がり1)

足し算をする方法[2](p. 6)

- 基本的な2進数の足し算の方法は10進数と同じ

0110(10進数で6)と0101(10進数で5)の足し算



➡ 計算結果: 1011(10進数で11)

桁あふれ(オーバフロー)[1]

- コンピュータでは数を表すビット数(2進数の桁数)は固定されている
 - 計算の結果、決まった桁数を超えると...?

Ex. 数を4ビット(4桁)で表す場合:

1110(10進数で14)と0101(10進数で5)の足し算

$$\begin{array}{r} 1110 \\ +) 0101 \\ \hline 10011 \end{array}$$

桁あふれ(オーバーフロー)[2]

Ex. 数を4ビット(4桁)で表す場合

1110(10進数で14)と0101(10進数で5)の足し算

$$\begin{array}{r} 1\ 1\ 1\ 0 \\ +) 0\ 1\ 0\ 1 \\ \hline 1\ 0\ 0\ 1\ 1 \end{array}$$

↑ 5ビット目(5桁目, 決められた桁数を越えてしまった部分)

- ➡ 決められた桁数を越える = 2進数の各桁を入れる箱の数足りなくなる
- ➡ 決められた桁数を越えた部分は無視される(捨てられてしまう)

~~1~~ 0 0 1 1

↑ 無視される(捨てられる)

➡ 計算結果: 0011(10進数で3)

計算結果が決められた桁数を越えること:
桁あふれ(オーバーフロー)

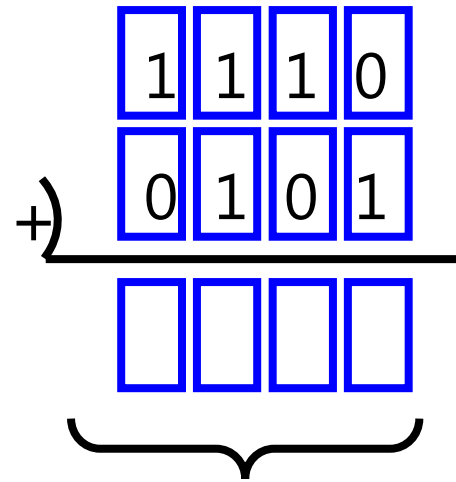
桁あふれ(オーバーフロー)の扱い[1]

- コンピュータでは、2進数の各桁を、1つずつ箱に入れて扱っている、というイメージ
 - 各桁を入れる箱の数に限りがある
 - Ex. 数を4ビットで表す = 数を4桁で表す(2進数の各桁を入れる箱の数が4個)
 - **どのような計算をしたとしても、箱の数は変更されない**
 - 計算結果を入れるための箱は、計算に使う数と同じ個数しか用意されない
 - Ex. 数を4ビットで表すときに、 $(1110 + 0101)_2$ の計算結果も4ビットでしか表現できない(計算結果を入れるための箱は4個しか用意されない)

桁あふれ(オーバーフロー): 本来の計算結果(人間が自分の手で行った計算結果)とコンピュータが行った計算結果(Ex. 電卓などの計算結果)が違ってしまう現象

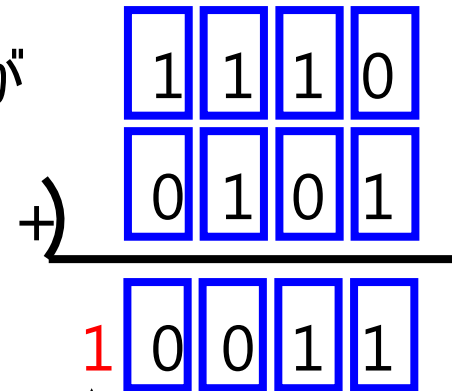
桁あふれ(オーバーフロー)の扱い[2]

- 2進数の各桁を入れる箱は、小さい桁(右の桁)の分から用意される
 - 大きい桁が箱に入れずにあふれてしまい、なくなってしまう
 - 1つの箱に2つ以上の桁を入れることはできない
 - あふれても、コンピュータは余分の箱を用意してくれない(用意できない)



計算結果を入れるために
用意されている箱

小さい桁から結果が
箱に入れられる



計算の結果、5桁目に突入してしまった
but...

箱は4つしか用意されていない



5桁目は無視されるので計算結果は(0011)₂

桁あふれ(オーバーフロー)の扱い[3]

- コンピュータ関係の問題では、2進数の桁数を指定されることが多い
 - Ex. 2進数4桁の数1110(10進数で14)と0101(10進数で5)を足した結果を10進数で答えなさい。

$$(1110+0101)_2 = (10011)_2 = (19)_{10}$$

➡ 答え: $(19)_{10}$ ❌

なぜ? → 問題文に「4桁」と指定されているのに、2進数の計算結果が5桁になっていてそのまま10進数に直しているから

$$(1110+0101)_2 = (10011)_2 \rightarrow (0011)_2 \text{ (桁数あわせ)} = (3)_{10}$$

➡ 答え: $(3)_{10}$ ○

2進数の段階で常に桁数を気にしよう!

2進数で計算(足し算・かけ算)するとき...

■ オーバーフローがどーの考えるのではなく!!

1. 計算結果を2進数で出す

- 10進数で計算結果を出すとオーバーフローの対処ができないため

2. 1. の計算結果を、指定された桁数に合わせる

- 桁数の指定がなければ、このステップは不要
- オーバーフローするか? という問題であれば、このステップで捨てる桁があればオーバーフロー

3. 2. の結果を10進数に直す

Ex. 2進数4桁の数1110(10進数で14)と0101(10進数で5)を足した結果を10進数で答えなさい

1. $1110 + 0101 = 10011$

2. $1\ 0\ 0\ 1\ 1 \xrightarrow{\text{「4桁」という指定がついているので...}} \begin{array}{c} \text{消す} \swarrow \quad \nwarrow \text{残す} \\ \text{1} \boxed{0\ 0\ 1\ 1} \end{array} \longrightarrow 0011$

3. $(0011)_2 = (3)_{10}$

オーバーフローでなく桁数あわせを気にすると計算できる!
➤ オーバーフローは桁数をあわせた計算の結果でしかない

2進数の 2^n 倍と $1/2^n$

2進数 $\times 2^n$

- 「2進数 $\times 2^n$ 」の計算は簡単
 - 2進数の一番右に、 n 個分「0」をつけるだけ
 - 2^n は2進数で表現すると、 $(10)_2$ を n 回掛け算した数だから

Ex:

$$\begin{aligned}(101101)_2 &\times (8)_{10} \\ &= (101101)_2 \times (2^3)_{10} \\ &= (101101)_2 \times (1000)_2 \\ &= \underline{101101000}\end{aligned}$$

もとの2進数の一番右に3個「0」がついているだけ

2進数 $\times 2^n$

- かけ算する2進数を小数で表現したとき、小数以下に「0」が並んでいる
 - 2^n を2進数にかけると、小数以下に並んでいた「0」が出てきて、
もとの数がn個分左にずれる、というイメージ

「左にnビットシフトする」と呼ぶ

Ex:

$$\begin{aligned}(101101)_2 \times (8)_{10} \\ = (101101)_2 \times (2^3)_{10} = (101101)_2 \times (1000)_2\end{aligned}$$

101101.000000000.....
1011010.00000000.....
10110100.0000000.....
101101000.000000.....

101101を左に3ビットシフトした数(小数点が移動しているだけ)

2進数 $\div 2^n$

- 「2進数 $\div 2^n$ 」の計算も簡単
 - 2進数の右からn桁分を小数部分にするだけ
 - 「2進数 $\div 2^n$ 」は2進数で表現すると、2進数を $(10)_2$ でn回割り算した数だから

Ex:

$$\begin{aligned}(111101)_2 &\div (8)_{10} \\ &= (111101)_2 \div (2^3)_{10} \\ &= (111101)_2 \div (1000)_2 \\ &= \underline{111.101}\end{aligned}$$

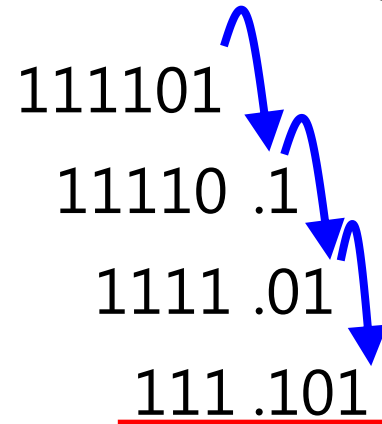
もとの2進数右から3桁分を小数部分にただけ
(小数点が移動しているだけ)

2進数 $\div 2^n$

- 2^n で2進数を割ると、**その2進数がn個分右にずれる**、というイメージ
「右にnビットシフトする」と呼ぶ

Ex:

$$(111101)_2 \div (8)_{10} \\ = (111101)_2 \div (2^3)_{10} = (111101)_2 \div (1000)_2$$



111101
11110 .1
1111 .01
111 .101

111101を右に3ビットシフトした数

シフト算でもオーバーフロー

- オーバーフローが起こるのは...

- 足し算
- かけ算(左にシフトする計算)

かけ算の場合... Ex. 4ビットの数: $(1011)_2 \times (8)_{10}$

$$(1011)_2 \times (8)_{10} = (1011)_2 \times (2^3)_{10}$$

$$= (1011)_2 \times (1000)_2$$

$$= (1011000)_2$$

7ビット(7桁)になってしまった

but... 各桁を入れる箱は、小さい桁から4桁分



大きい桁(左の桁)から3桁分が無視されるので、計算結果は $(1000)_2$

やってみよう![3]

- 10010を左に4ビットシフトした数
- 11001を左に7ビットシフトした数
- 1110101を左に2ビットシフトした数
- 10100000を右に3ビットシフトした数
- 11010100000を右に5ビットシフトした数
- 10101000を右に2ビットシフトした数

※すべて2進数のままで良い

コンピュータでの情報量

バイト[1](p. 8)

- コンピュータでの情報量:

情報を表現する「0」と「1」の数 = **ビット**

コンピュータの世界では、「0」と「1」を8個単位で扱うことが多い

Ex.:

半角英数の文字: 8個の「0」と「1」で構成 (8個 × 1)

全角の文字: 16個の「0」と「1」で構成 (8個 × 2)

画像などの色: 24個の「0」と「1」で構成 (8個 × 3)

8ビットで1つの単位: **バイト(byte)**

バイト[2](p. 8)

- 1バイト(byte) = 8ビット(bit)
 - 半角英数1文字(8ビット): 1バイト
 - 全角1文字(16ビット): 2バイト
 - 画像などの色1つ(24ビット): 3バイト

バイト[3](p. 8)

- 現実世界: 1000で1つの単位
 - 1000: 1K (1000m = 1Km)
- コンピュータの世界では 2^{10} で1つの単位
 - 1Kbyte(キロバイト, KB): 1024byte
 - 1Mbyte(メガバイト, MB): 1024Kbyte
 - 1Gbyte(ギガバイト, GB): 1024Mbyte
 - 1Tbyte(テラバイト, TB): 1024Gbyte

便宜上、1KB = 1000byte, 1MB = 1000KB, etc. とすることもある

8進数と16進数

8進数と16進数[1]

- コンピュータでの情報: 2進数で扱われる

情報量が多くなると桁数が大きくなって、人間には扱いにくい

Ex.

アルファベットの「N」: 01001110 (8桁)

日本語の「ん」: 1010010011110011 (16桁)

赤色: 111111110000000000000000 (24桁)

人間にとっては扱いにくい

(コンピュータの制御を考えると、人間もコンピュータのように考える必要)



8進数&16進数

8進数と16進数[2]

- 8進数: 数を0～7の8つの数字で表現
- 16進数: 数を0～9とA～Fの16個の文字で表現

- A: 10
- B: 11
- C: 12
- D: 13
- E: 14
- F: 15

覚えよう!

※いろいろな試験で、電卓の持ち込みはできないので、
きちんと自分で計算できるようになろう!

8進数と16進数[3]

Ex.

アルファベットの「N」

2進数: 01001110

8進数: 116

16進数: 4E

日本語の「ん」:

2進数: 1010010011110011

8進数: 51163

16進数: 5273

赤色:

2進数: 111111110000000000000000

8進数: 77600000

16進数: FF0000

16進数

- 16進数が特によく使われる
 - コンピュータの世界では0～255の数値で表現されるものが多い

Ex. 色

- 色は赤・緑・青の濃淡を混ぜ合わせて表現する
- 赤・緑・青を0～255の256段階の濃淡を混ぜ合わせる

赤成分: 255, 緑成分: 102, 青成分: 153 →

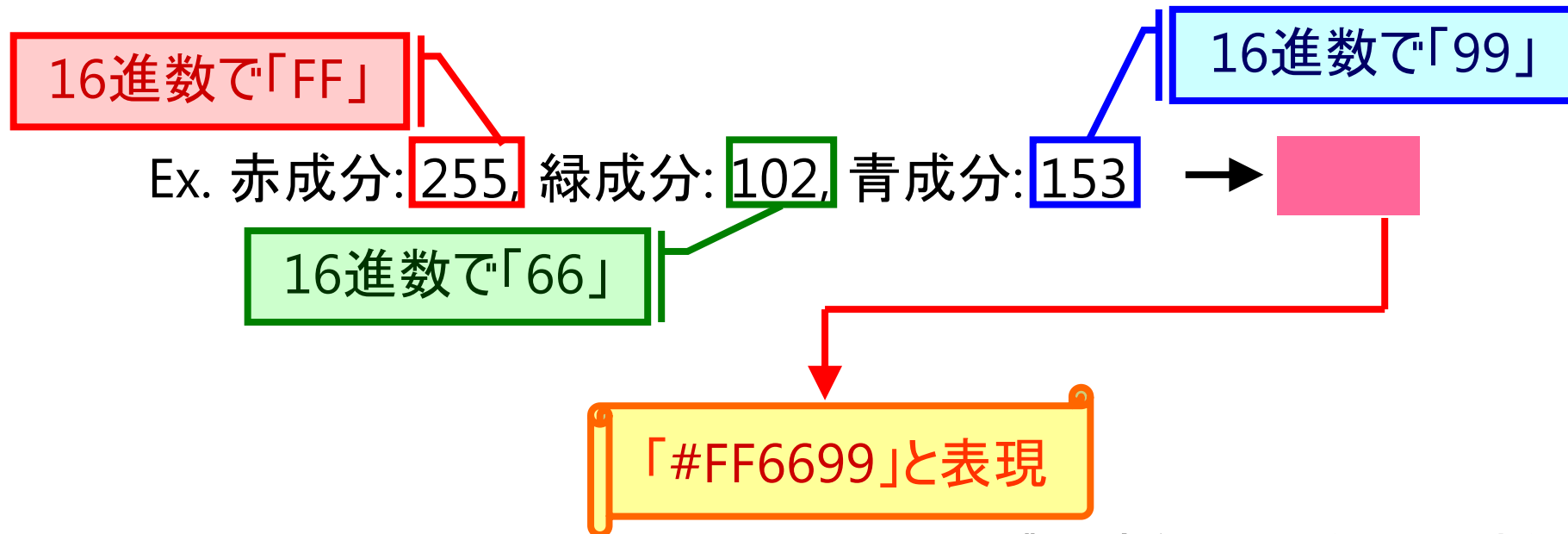


0～255を16進数で表すと0～FFで、ちょうど2桁で表せる

※色のほかに、文字も16進数で表すことが多い
(半角英数: 16進数2桁, 全角: 16進数4桁)

16進数がよく使われる例

- 色の表現: 赤・緑・青の256段階の濃淡で表現
 - それぞれの濃淡の度合いを0～255の数値で表現
 - 濃淡の度合いの数値を16進数で表現
 - 16進数の数値を赤・緑・青の順に並べ、先頭に「#」をつけて色を表現 (色の名前として利用)



※Webページ作成などのときによく使う

やってみよう!

- 授業の資料のページに行って...
 1. 「PECO STEP」のページに行く
 2. 「カラーコード」の入力欄に、16進数の色の名前を入力する
 - 色の名前は適当で良い(1文字目が「#」、2文字目以降が1～9とa～fの文字)
 - 色の名前のアルファベットは大文字でも小文字でも良い
 3. 下側の領域に、その色が表示される
 - R・G・Bの欄に、赤・緑・青の濃淡の10進数
 - 「色の見本」の欄に、2. で入力した色

※必要な色を探すときは、「色見本」などのキーワードで検索すると良い

10進数を16進数に変換

■ 16進数を求める計算方法

1. 10進数の数を16で割って商1と余り1を計算する
2. 商1を16で割って商2と余り2を計算する
3. 商2を16で割って商3と余り3を計算する
4.

$$\begin{array}{r} 16 \overline{) 10\text{進数の数}} \\ 16 \overline{) \quad \quad \quad \text{商1} \cdots \text{余り1}} \\ 16 \overline{) \quad \quad \quad \text{商2} \cdots \text{余り2}} \\ \quad \cdot \\ \quad \cdot \\ 16 \overline{) \quad \quad \quad \cdot} \\ \quad \quad \quad \text{商n} \cdots \text{余りn} \end{array}$$

商が0になるまで繰り返す
小数の計算はしない

余り1
余り2
.
.
.

時計回りに倒す

余りn-1
余りn

余り
2
1

余り
2
1

余りを余りnから余り1の順に左から並べたものが16進数
(ただし10～15の余りは、A～Fに置き換えること)

10進数を16進数に変換(例)

10進数の255を16進数に変換

$$\begin{array}{r} 16 \overline{) 255} \\ 16 \overline{) 15} \cdots \text{余り: } 15 \\ \underline{ 0} \cdots \text{余り: } 15 \end{array}$$



$$(15)_{10} = (F)_{16}$$



$$(255)_{10} = (FF)_{16}$$

10進数の2000を16進数に変換

$$\begin{array}{r} 16 \overline{) 2000} \\ 16 \overline{) 125} \cdots \text{余り: } 0 \\ 16 \overline{) 7} \cdots \text{余り: } 13 \\ \underline{ 0} \cdots \text{余り: } 7 \end{array}$$



$$(13)_{10} = (D)_{16}$$



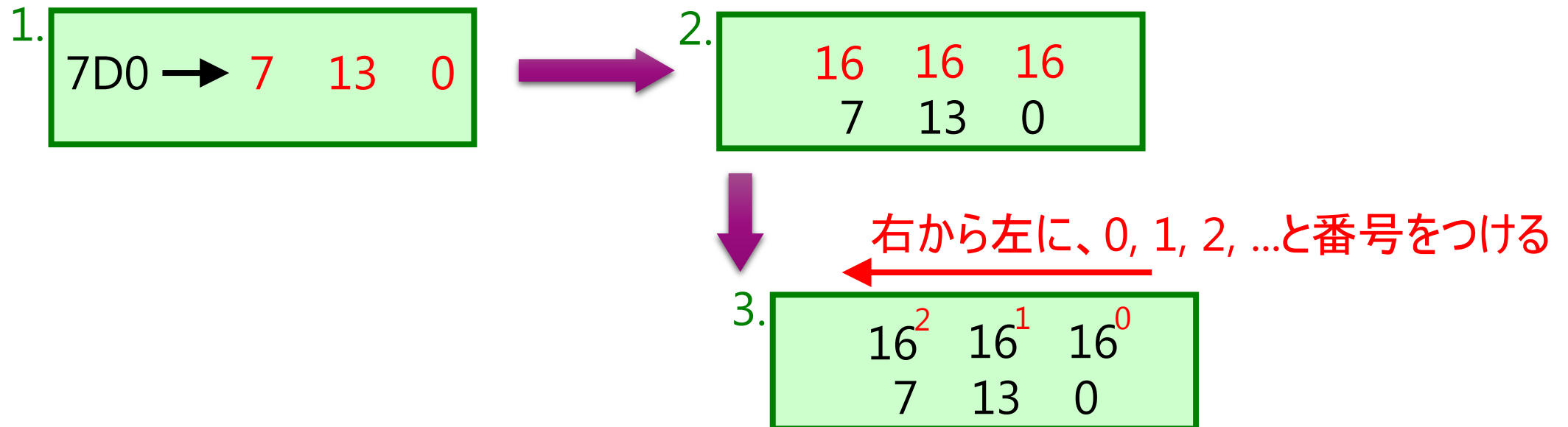
$$(2000)_{10} = (7D0)_{16}$$

→ の方向に余りを並べる

16進数を10進数に変換

■ 16進数→10進数の変換

1. アルファベットを10進数の数になおす
2. 2進数の各桁の上にそれぞれ「16」を書く
3. 1. で書いた「16」の右肩に、右から0, 1, 2, ...と書いていく
4. 16^0 , 16^1 , 16^2 , ...ができていく



16進数を10進数に変換

■ 16進数→10進数の変換

5. 各桁の上の「 16^n 」と、それぞれの桁の数をかけあわせる

6. 2. の結果を足し合わせる

3.

16^2	16^1	16^0
7	13	0



4.

16^2	16^1	16^0
×	×	×
7	13	0
7×16^2	13×16^1	0



5.

7×16^2	13×16^1	0
足し合わせる		
$7 \times 16^2 + 13 \times 16^1 + 0 = 2000$		

やってみよう!

1. 10進数の「240」を16進数に
2. 10進数の「3000」を16進数に
3. 10進数の「50000」を16進数に
4. 16進数の「64」を10進数に
5. 16進数の「FA0」を10進数に
6. 16進数の「4E20」を10進数に
7. 16進数の「A3」を10進数に
(2012年度ITパスポート秋季試験問題)

Question!

8進数

8進数

- 16進数ほどではないが、知っておくべき数の表現方法
 - 情報処理技術者試験などには出ることも

10進数を8進数に変換

■ 8進数を求める計算方法

1. 10進数の数を16で割って商1と余り1を計算する
2. 商1を16で割って商2と余り2を計算する
3. 商2を16で割って商3と余り3を計算する
4.

8) 10進数の数
8) 商1...余り1
8) 商2...余り2
.
.
.
8) 商n...余りn

商が0になるまで繰り返す
小数の計算はしない

➡ 余りを余りnから余り1の順に左から並べたものが8進数

余り1
余り2
.
.
.
余りn-1
余りn

時計回りに倒す

余り
2
余り
1
.
.
.

※10進数→X進数の計算は、割る数がXになるだけで、やり方はどのX進数でも同じ

10進数を8進数に変換(例)

10進数の255を8進数に変換

$$\begin{array}{r} 8 \overline{) 255} \\ 8 \overline{) 31} \cdots \text{余り: } 7 \\ 8 \overline{) 3} \cdots \text{余り: } 7 \\ 0 \cdots \text{余り: } 3 \end{array}$$



$$(255)_{10} = (377)_8$$

10進数の2000を8進数に変換

$$\begin{array}{r} 8 \overline{) 2000} \\ 8 \overline{) 250} \cdots \text{余り: } 0 \\ 8 \overline{) 31} \cdots \text{余り: } 2 \\ 8 \overline{) 3} \cdots \text{余り: } 7 \\ 0 \cdots \text{余り: } 3 \end{array}$$



$$(2000)_{10} = (3720)_8$$

→ の方向に余りを並べる

8進数を10進数に変換

■ 8進数→10進数の変換

1. 8進数の各桁の上にそれぞれ「8」を書く
2. 1. で書いた「8」の右肩に、右から0, 1, 2, ...と書いていく
 - $8^0, 8^1, 8^2, \dots$ ができていく

1.

8	8	8	8
3	7	2	0



2.

右から左に、0, 1, 2, ...と番号をつける

³ 8	² 8	¹ 8	⁰ 8
3	7	2	0

8進数を10進数に変換

■ 8進数→10進数の変換

3. 各桁の上の「 8^n 」と、それぞれの桁の数をかけあわせる
4. 2. の結果を足し合わせる

2.

8^3	8^2	8^1	8^0
3	7	2	0



3.

8^3	8^2	8^1	8^0
×	×	×	×
3	7	2	0
3×8^3	7×8^2	2×8^1	0



4.

3×8^3	7×8^2	2×8^1	0
足し合わせる			
$3 \times 8^3 + 7 \times 8^2 + 2 \times 8^1 + 0 = 2000$			

※X進数→10進数の計算は、累乗する数がXになるだけで、やり方はどのX進数でも同じ

やってみよう!

1. 10進数の「200」を8進数に
2. 10進数の「1000」を8進数に
3. 10進数の「555」を8進数に
4. 8進数の「100」を10進数に
5. 8進数の「2734」を10進数に
6. 8進数の「55」を16進数に
(2009年度ITパスポート秋季試験問題)

負数の表現方法

負の数の表現[1](p. 9)

- コンピュータでの計算は、全て足し算
 - コンピュータでの計算は、電気回路で実行
 - 電気回路: 電気が通る線を組み合わせて、様々な処理をするためのもの(コンピュータを構成する最も基本的な部品)
 - 足し算, 引き算, かけ算, 割り算をするには、それぞれのために専用の回路が必要
 - 足し算専用回路, 引き算専用回路, かけ算専用回路, 割り算専用回路
- 経済的に良くない



足し算専用回路(加算器)を
組み合わせて他の計算をカバー

負の数の表現[2](p. 9)

- 足し算の組み合わせで他の計算も行う
 - 引き算: 「 $a-b$ 」を、「 $a+(-b)$ 」(b を負数と考える)
 - かけ算: 足し算の繰り返しとして計算
 - 割り算: 引き算の繰り返しとして計算

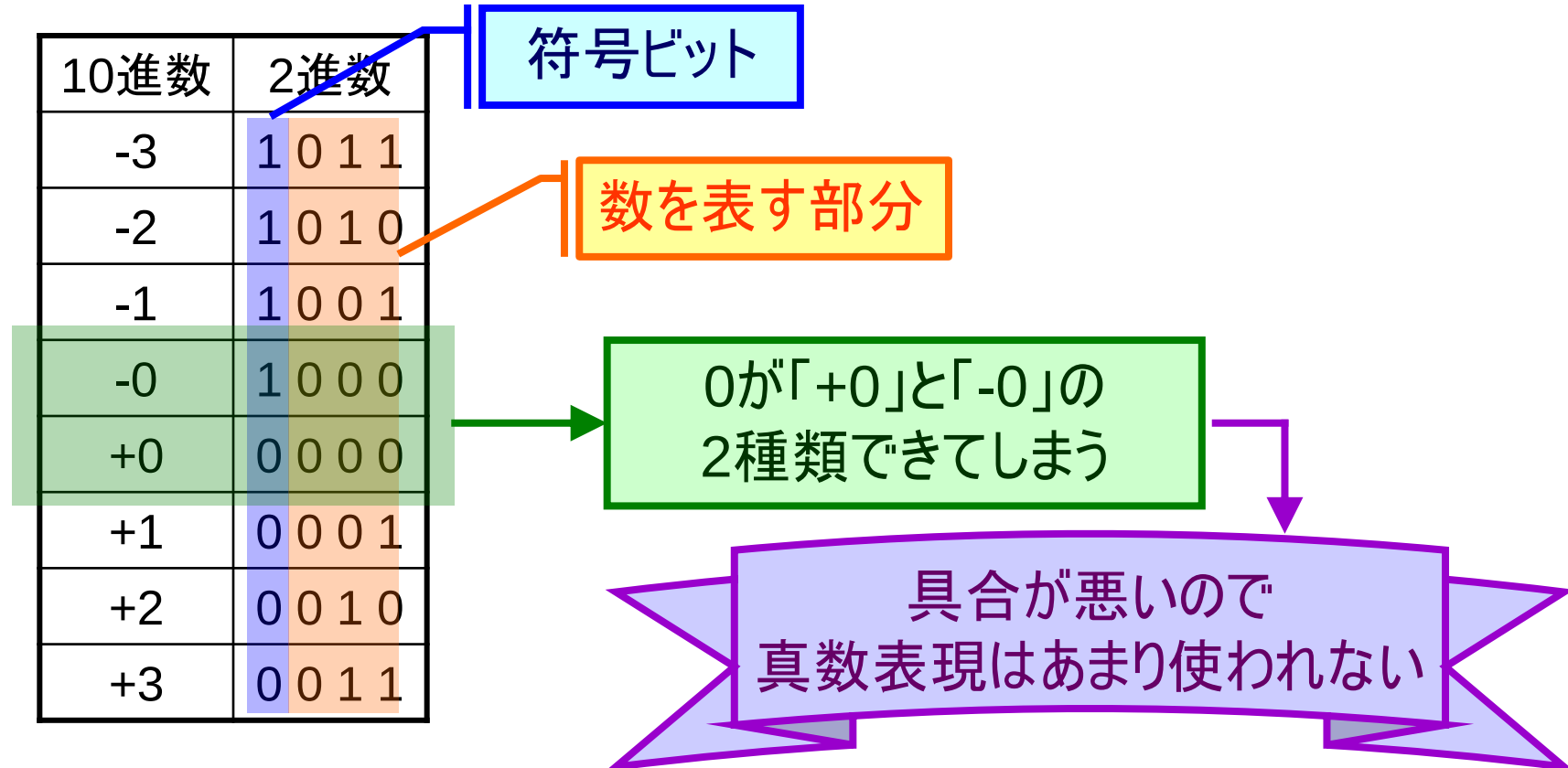
コンピュータでも負数を扱う

- 方法1: 真数表現
- 方法2: 2の補数表現

真数表現(p. 9)

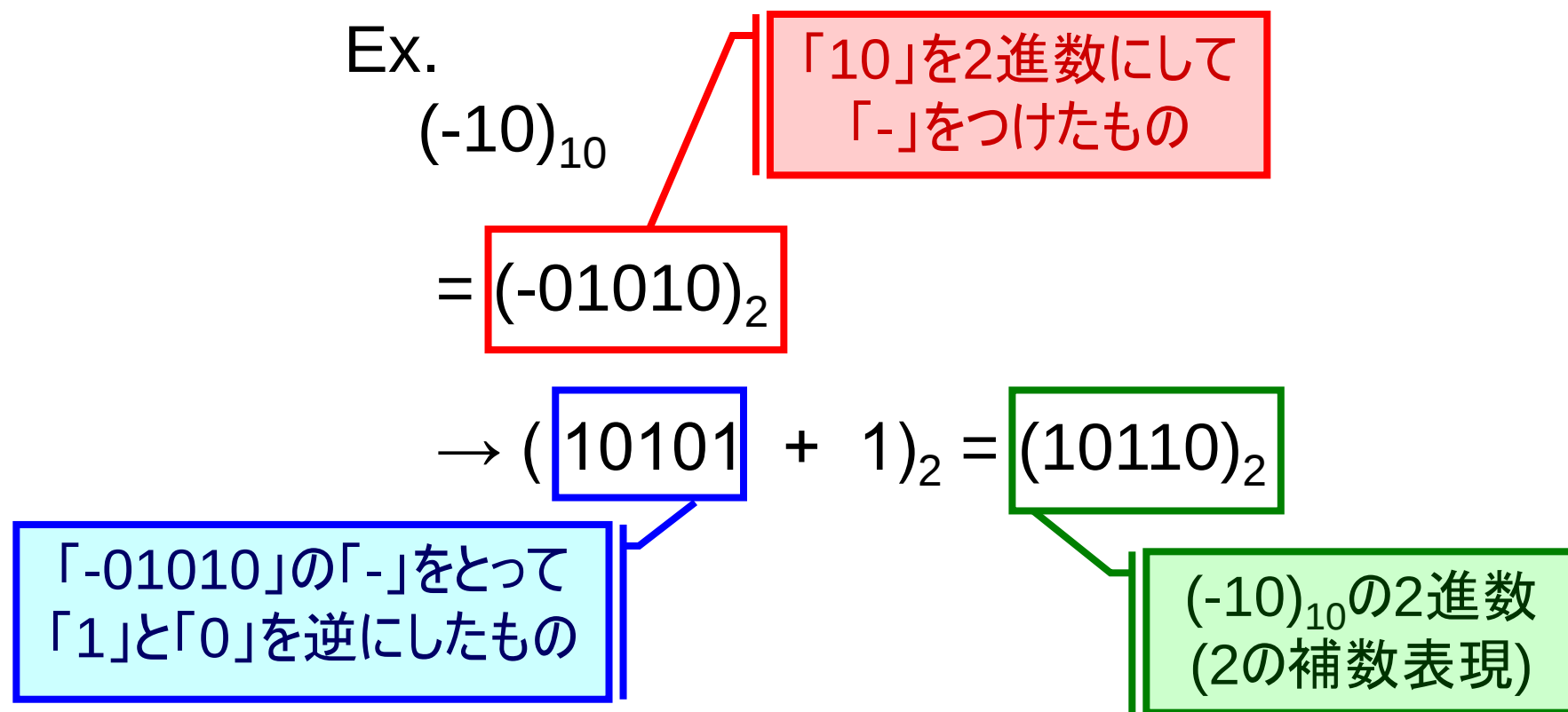
- 数を表す2進数に符号(+ or -)を表す1ビットを付加
 - 数の先頭のビットで符号を表す
 - 0が「+」、1が「-」を表す

「符号ビット」と呼ぶ



2の補数表現[1](p. 9)

- 負の数Nを、正の数N(2進数)の0と1を反転させて1を加えた数で表現する方法
 - 0と正の整数(自然数)は、そのまま表現(この計算はしない)



2の補数表現[2](p. 9)

- 2の補数 = 負の数を2進数で表現したもの(コンピュータの世界では)
- 計算方法(例: -20を10桁の2進数に直す)
 1. 2の補数に直したい10進数のマイナスを取り除く
 - $(-20)_{10} \rightarrow (20)_{10}$
 2. 1. の結果を2進数に直す
 - $(20)_{10} = (0000010100)_2$
 3. 2. の結果の0と1を逆にする(0の桁を1、1の桁を0にする)

0 0 0 0 0 1 0 1 0 0



1 1 1 1 1 0 1 0 1 1

2の補数表現[2](p. 9)

- 2の補数 = 負の数を2進数で表現したもの(コンピュータの世界では)
- 計算方法(例: -20を10桁の2進数に直す)

4. 3. の結果に1を足し算する

$$\begin{array}{r} 1111101011 \\ +) \quad \quad \quad 1 \\ \hline 1111101100 \end{array}$$

-20を2進数に直した結果
(2の補数 = 2進数での負の数の表現)

2進数での負の数の表現では、
「-」の記号はつけない

2の補数表現の利点(p. 10)

- 引き算(符号付きの足し算)をそのまま足し算として処理できる
(自然数と同様に処理できる)

Ex.

$$(10 + 3)_{10} = (01010 + 00011)_2 = (01101)_2$$

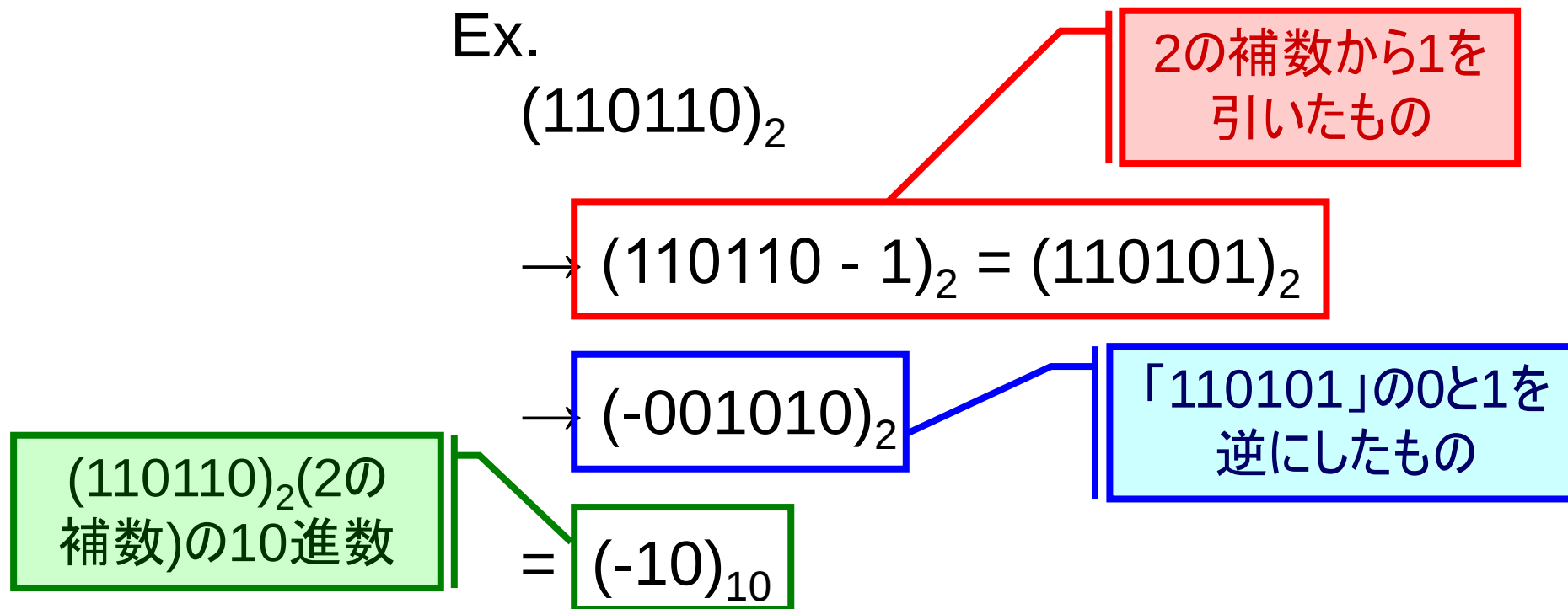
$$(-6 + 3)_{10} = (11010 + 00011)_2 = (11101)_2$$



真数表現: 符号付きの足し算を処理するには、別の回路が必要
(単純に足すことはできない)

2の補数を10進数に変換[1]

- 2の補数から1を引き、0と1を反転させて10進数になおして「-」をつける
 - 負の数を2の補数に変換するときの逆
 - この計算は、負の数だけ



2の補数を10進数に変換[2]

- 計算方法(例: 1111101100を10進数に直す)

1. 2の補数から1を引き算する

$$\begin{array}{r} 1111101100 \\ -) 1 \\ \hline 1111101011 \end{array}$$

2. 1. の結果の0と1を逆にする(0の桁を1、1の桁を0にする)

$$\begin{array}{c} 1111101011 \\ \downarrow \\ 0000010100 \end{array}$$

※2の補数→10進数の方法は、10進数→2の補数の逆

2の補数を10進数に変換[2]

■ 計算方法(例: 1111101100を10進数に直す)

1.2. の結果を10進数に直す

- $(0000010100)_2 = (20)_{10}$

2.3. の結果に-(マイナス)をつける

- $(20)_{10} \rightarrow (-20)_{10}$

1111101100を
10進数に直した数

※2の補数→10進数の方法は、10進数→2の補数の逆

2進数の引き算[1]

- 10進数の引き算だと...

- ある桁の引かれる数が引く数より小さければ、1つ大きな桁から10を借りる
 - 10を借りる: 貸した桁から1を引き、借りた桁に10を足す

10を借りる

$$\begin{array}{r} 100 \\ - 1 \\ \hline \end{array} \rightarrow \begin{array}{r} 090 \\ - 1 \\ \hline \end{array} \rightarrow \begin{array}{r} 099 \\ - 1 \\ \hline \end{array}$$

引き算の答え: 99

2進数の引き算[2]

- 2進数の引き算だと...
 - ある桁の引かれる数が引く数より小さければ、1つ大きな桁から $(10)_2$ (10進数で2)を借りる
 - 2を借りる: 貸した桁から1を引き、借りた桁に2を足す

2(2進数で10)を借りる

$$\begin{array}{r} 1\ 0\ 0 \\ -)0\ 0\ 1 \\ \hline \end{array}$$



2(2進数で10)を借りる

$$\begin{array}{r} 0\ 2\ 0 \\ -)0\ 0\ 1 \\ \hline \end{array}$$



$$\begin{array}{r} 0\ 1\ 2 \\ -)0\ 0\ 1 \\ \hline 0\ 1\ 1 \end{array}$$



引き算の答え: 011

※コンピュータ的には引き算はしないので、人間が2の補数→10進数の計算をするための引き算

やってみよう!

1. -25を2の補数10桁で表現
2. -32を2の補数10桁で表現
3. 2の補数10000を10進数で表現
4. 2の補数1011000を10進数で表現

Question!