

コンピュータ・サイエンス1

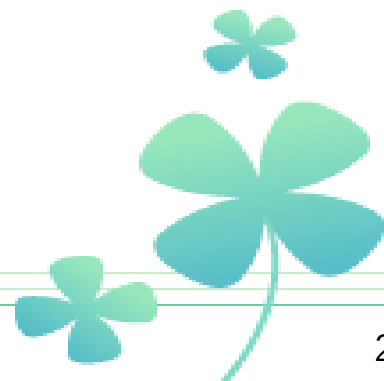
第5回 コンピュータでの情報の扱い方

人間科学科コミュニケーション専攻
白銀 純子



第5回の内容

◆コンピュータでの情報の扱い方



前回の出席問題の解答

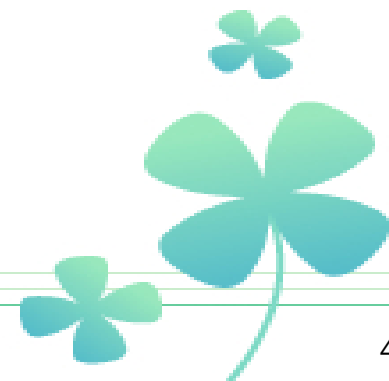
- ◆ 設問1: 「USB」というインターフェースについて、USBが現れたことにより、従来と比べてどのようなことが良くなったかを答えなさい。

解答

従来はキーボード・マウスやプリンタなど、装置ごとに異なるポートに接続していたが、USBの登場により、統一することができた。



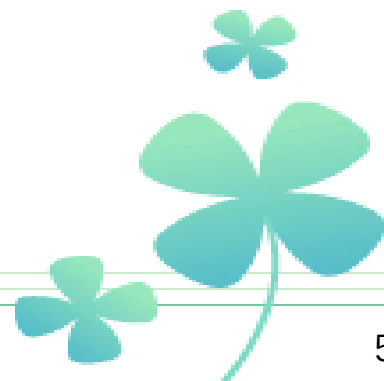
前回の質問の回答



Bluetooth

★電波で様々な装置とコンピュータを接続するためのインタフェース

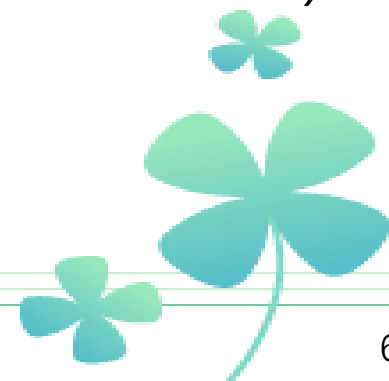
- ★キーボード
- ★マウス
- ★スピーカー
- ★イヤホン
- ★etc.



部品を接続する場所[1]

★ 形が一致する場所に接続するのが基本

- ★ 多くの場合、形が一致する場所であれば、どこに接続してもOK
- ★ マザーボードに直接差し込む部品: CPU, メインメモリ, 拡張カード
 - ★ すべて、マザーボード上のスロットの形が違うので、形が合うスロットに取り付け
 - ★ CPUは専用の取り付け台(ほぼ正方形)
 - ★ メインメモリは、CPU近くのみぞ
 - ★ 拡張カードは、メインメモリよりもCPUから遠いみぞ
- ★ マザーボードとケーブルで接続する部品: HDD/SSD, 光ディスクのドライブ
 - ★ 接続口の形が一致すれば、どのケーブルを使ってもOK(ケーブルも前後などの方向はなし)
 - ★ マザーボード上にも、同じ形のケーブルの接続口があり



部品を接続する場所[2]

★ 電源: メイン電源, 12V電源, 外部記憶装置用の電源

★ メイン電源: 細かい四角が12個ずつ2列に並んでいる電源

- ★ よく見ると、細かい四角は、四角形のものや山形になっているものあり

- ★ マザーボードに電源の差込口あり

- ★ 電源ケーブル側の爪と、マザーボード側の爪を引っ掛ける部分をあわせて差込み

★ 12V電源(CPU用の電源): 細かい四角が2個ずつ2列に並んでいる電源

- ★ よく見ると、細かい四角は、四角形のものや山形になっているものあり

- ★ マザーボードに電源の差込口あり

- ★ 電源ケーブル側の爪と、マザーボード側の爪を引っ掛ける部分をあわせて差込み

★ 外部記憶装置用の電源

- ★ 電源から出ているケーブルの形と、記憶装置がわの差込口の形をよく見て、形の合うものを差込み

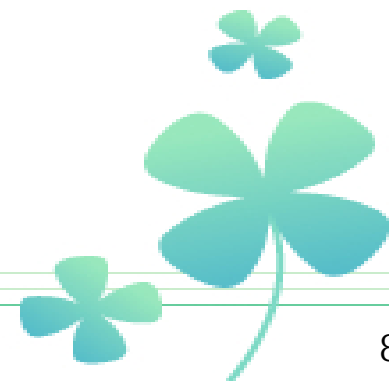


PCが重くなる原因[1]

- ★ CPUが熱くなる→PCが重くなる ×
- ★ CPUの作業量が多くなる→PCが重くなる & CPUが熱くなる ○

★ PCが重くなる原因

- ★ CPUの作業量が多い
- ★ HDDやSSDへのアクセスが多い
- ★ メインメモリが足りない
- ★ ファイルが断片化されている



PCが重くなる原因[2]

★ 具体的には...

★ 多くの常駐アプリケーション

→ 終了する・自動で起動しない設定をする, いらないものを削除する

★ 常駐アプリケーション: PCが起動している間、常に動作し続けているアプリケーション

★ Windowsだと、画面右下にずらっと表示されているアイコンは、多くが常駐アプリケーション



★ ウィルスソフトの定期チェック → スケジュールを変更する

★ ウィルスソフトは、定期的に、コンピュータ内の全ファイルのチェックをする

★ ソフトウェアの自動アップデート(WindowsやMac OS、ウィルスソフトなど)

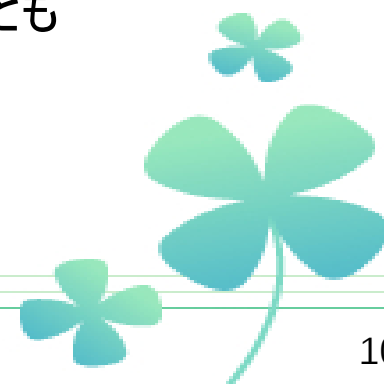
→ スケジュールを変更する



PCが重くなる原因[3]

★ 具体的には...(続き)

- ★ 分割してHDD・SSD内のあちこちに保存されている大きなファイル → デフラグ
 - ★ ファイルを作ったり削除したりを繰り返していると、後から保存するファイルは分割されてHDD・SSD内でばらばらに保存される
 - ★ 分割されて保存されると、そのファイルにアクセスするときに時間がかかる
 - ★ デフラグ(分割されているファイルを1つにまとめる処理): 「スタート」→「Windowsシステムツール」→「コントロールパネル」→「システムとセキュリティ」→「ドライブのデフラグと最適化」
- ★ ウィルスやスパイウェアに感染 → ウィルスソフトのアップデートとPC内チェック
 - ★ スパイウェア: PCでのユーザの行動や個人情報を収集したりするソフトウェア
 - ★ ソフトウェアの使用許諾条件に書かれていることも多いので、意識せずに同意していることも



今後の実習予定

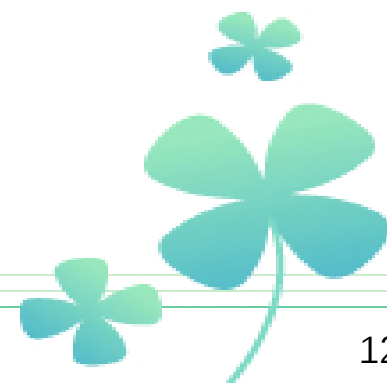
- ★ 7月頃に2～3回を予定

- ★ 情報処理教室での実習





Question!





コンピュータでの情報の扱い方



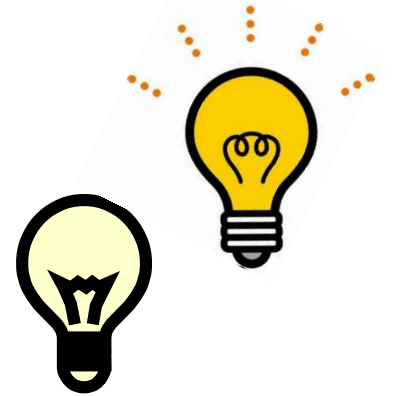
コンピュータの基本構成

★ コンピュータは電気回路で構成

- ★ 電気回路: 電気が通ることによって動作する様々な部品(電気素子)を電気を通す線で結んだもの
- ★ CPUなど、ほとんどの部品は電気回路で構成

★ 様々な情報を電気で表す必要

- ★ 電気が伝えることができるのは、2種類の情報のみ → 1と0で表す
 - ★ 部品内に電気が通っている(電圧が高い) → 1と表す
 - ★ 部品内に電気が通っていない(電圧が低い) → 0と表す



豆知識

電源マーク 

- 0と1を組み合わせて作られたマーク
- 1つのボタンでON・OFFを切り替える場合は組み合わせ
- ONとOFFそれぞれボタンがある場合はONが1、OFFが0のボタン

コンピュータでの情報の扱い方[1](p. 2)

- ◆ コンピュータが扱える情報は「0」と「1」のみ
- ◆ 大量の「0」と「1」を組み合わせて情報を表現
 - ◆ 部品をたくさん用意して、それぞれに電気が通っている・いないの状況を組み合わせて情報を表現
 - ◆ それぞれの物事は、決まった個数の0と1で表現
 - ◆ 半角英数1文字: 8個
 - ◆ 全角1文字: 16個
 - ◆ etc.
- ◆ 様々な情報を「0」と「1」の形(ビット)に変換して記録
 - ◆ 数, 文字, 画像, 音声, etc.は、全てそれぞれの方法で0と1の並びに変換

コンピュータでの情報の扱い方[2](p. 2)

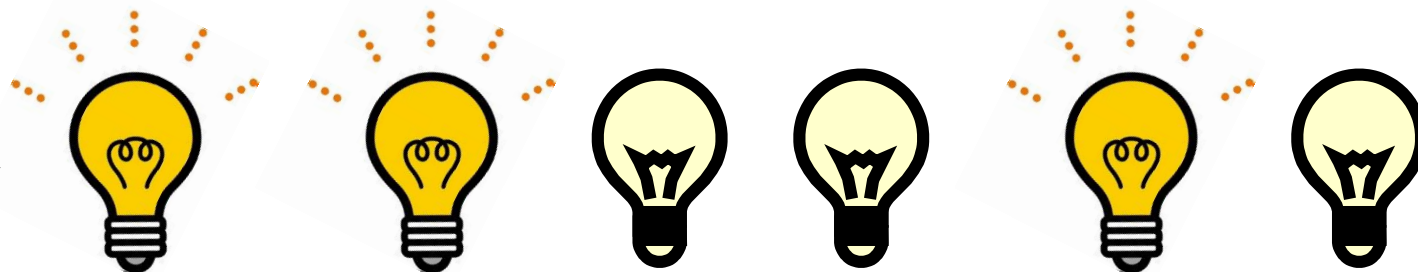
◆ 数値は0と1の並びで表現

- ◆ 数値を表す0と1の個数は、扱い方によっていくつか種類が存在

例えば...

「50」: 110010

「100」: 1100100



◆ 1文字1文字は0と1の並びで表現

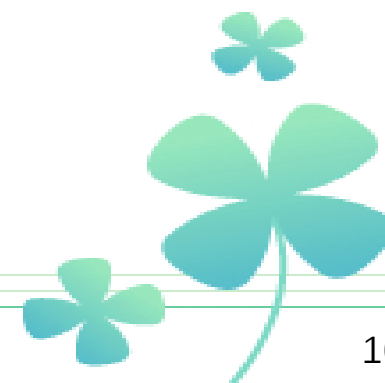
例えば...

アルファベットの「N」: 01001110

8個の0と1

日本語の「ん」: 1010010011110011

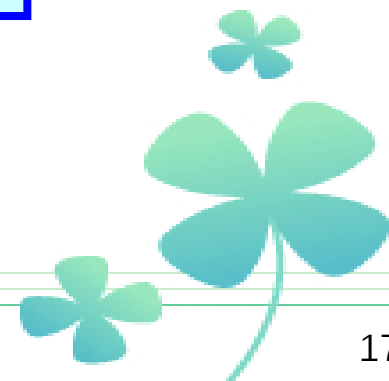
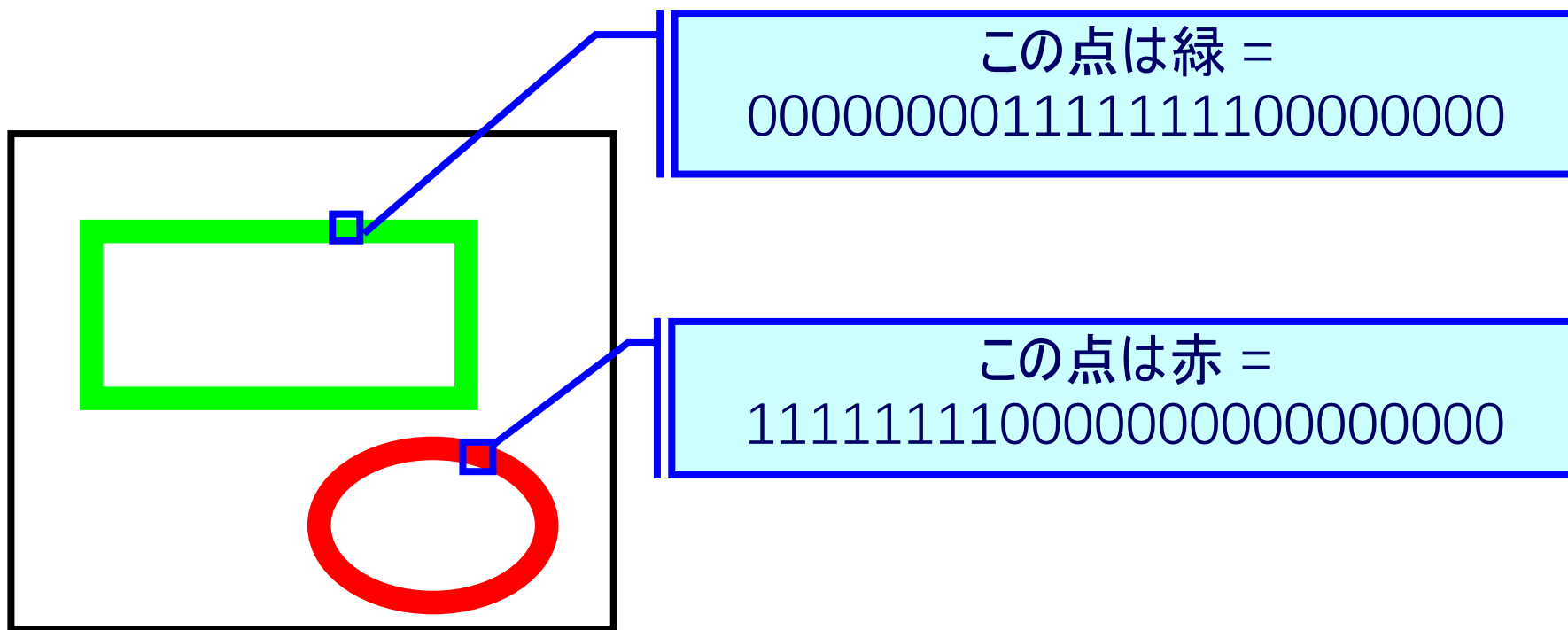
16個の0と1



コンピュータでの情報の扱い方[3](p. 2)

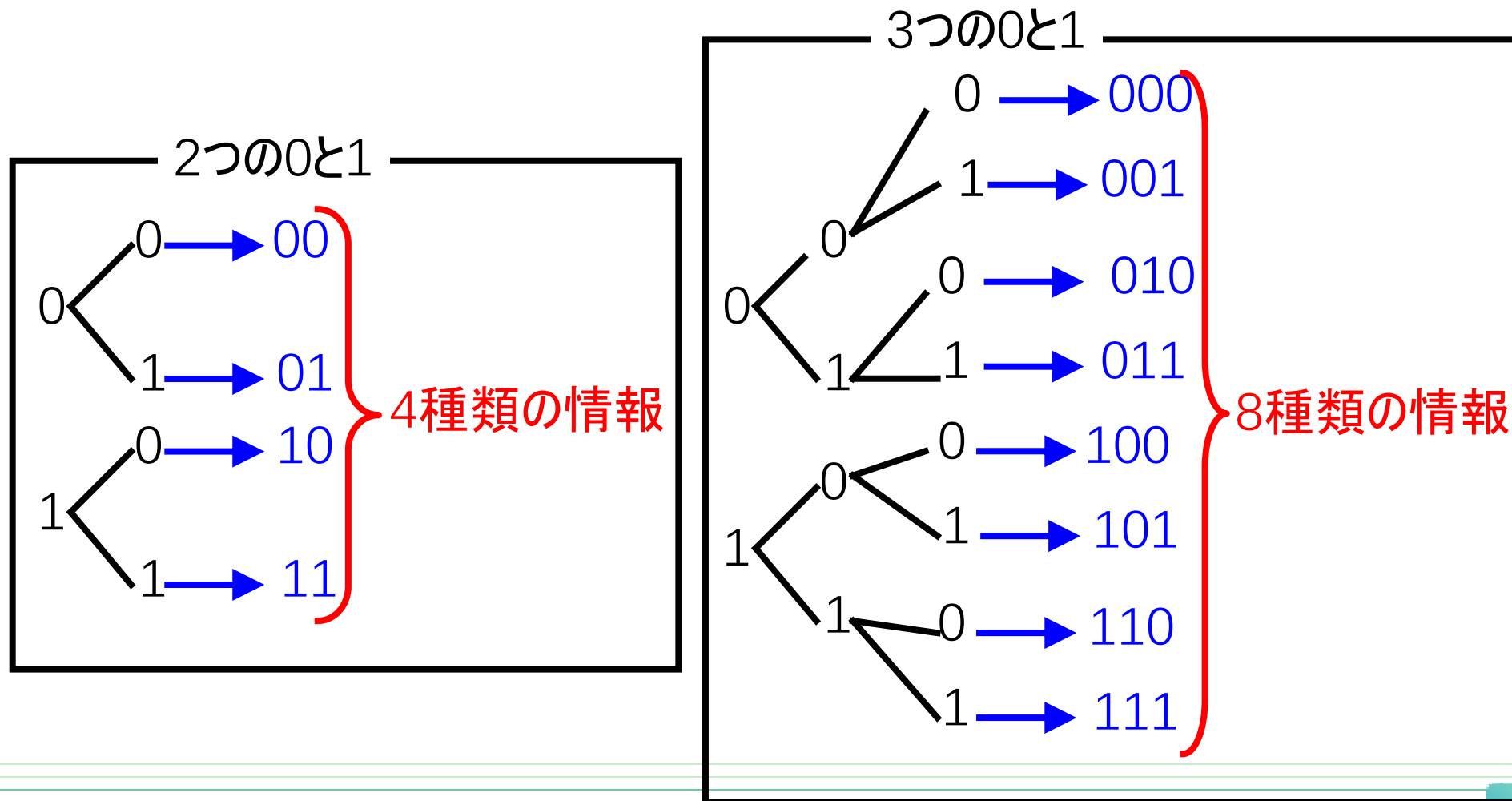
◆ 画像は、コンピュータにとっては点の集まり

◆ 1つ1つの点は何色かで絵を表現



ビット[1](p. 4)

◆ 0と1の組み合わせ



ビット[2](p. 4)

★0と1の個数が1つ増えると、表現できる情報の種類は2倍に

★2個の「1」と「0」→ 4種類の情報

★3個の「1」と「0」→ 8種類の情報

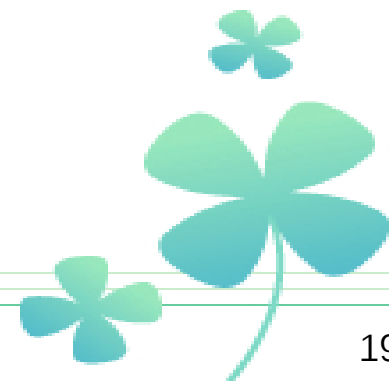
★n個の「1」と「0」→ 2^n 種類の情報

★ 2^n : 2をn回掛け算する

★例: $2^5 = 2 \times 2 \times 2 \times 2 \times 2 = 32$



組み合わせる「0」と「1」の数が多くなれば、
表現できる情報の種類も増える



ビット[3](p. 4)

★ビット: 情報を表現する1つ1つの「0」と「1」

★コンピュータでの情報量の基本単位

★情報を表現する「0」と「1」の個数

★ビット列: 情報を表現する1つ1つの「0」と「1」の並び

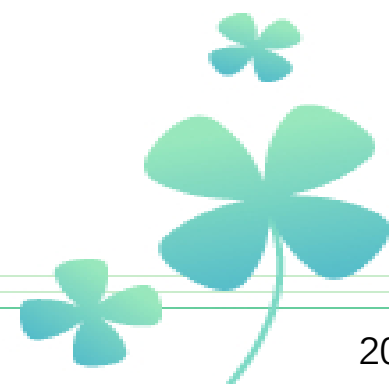
例えば...

「50」: 110010  6 ビット

「100」: 1100100  7 ビット

アルファベットの「N」: 01001110  8 ビット

日本語の「ん」: 1010010011110011  16 ビット



情報をどうやって0と1で表す?[1]

- ◆ 原則: 個々の情報の内容はすべて同じビット数で表す
- ◆ 考え方: 何種類の情報の内容があるか? をもとに、ビット数を決める
 - ◆ nビットとすると、 2^n 個の情報の内容が表せる

Ex1. 1週間の朝食メニュー

洋食・和食・その他・なし、の4種類だった → $2^2=4$ なので、2ビットで表せる

00: 洋食, 01: 和食, 10: その他, 11: なし

※洋食は000、和食は01...のように違うビット数では表さないのが原則

そうすると、4/1～4/7の1週間の朝食メニューを表した文書は、コンピュータ的には...

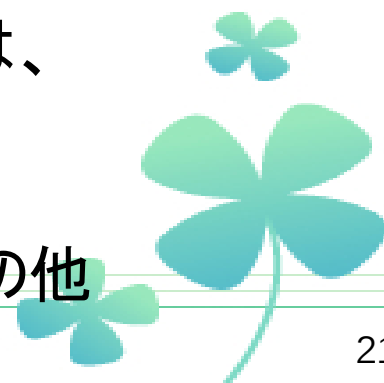
00000100101110



各情報が2ビットなのはわかっているので、各日の朝食が何だったかは、2ビットずつで区切ればわかる



4/1: 洋食, 4/2: 洋食, 4/3: 和食, 4/4: 洋食, 4/5: その他, 4/6: なし, 4/7: その他



情報をどうやって0と1で表す?[2]

Ex1. 天気

晴れ・曇り・小雨・大雨・小雪・大雪、の6種類だった

- $2^2=4$ だと足りない
- $2^3=8$ だと足りる(使っていないビット列の番号があってもOK)

000: 晴れ, 001: 曇り, 010: 小雨, 011: 大雨, 100: 小雪, 101: 大雪

そうすると、ある日の関東(東京・神奈川・埼玉・千葉・群馬・茨城・栃木)の天気を表した文書は、コンピュータ的には... **100011011001001011010**

各情報が3ビットなのはわかっているので、各県のある日の天気が何だったかは、3ビットずつで区切ればわかる

東京: 小雪, 神奈川: 大雨, 埼玉: 大雨, 千葉: 曇り, 群馬: 曇り, 茨城: 大雨, 栃木: 小雨

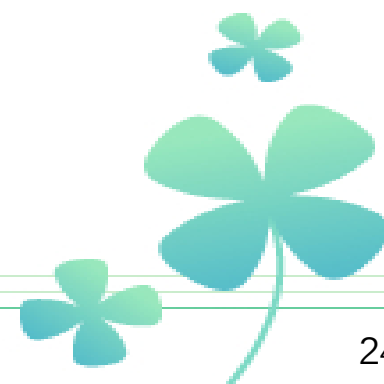
やってみよう!

- ★ 自分の身近なもの(何でもOK)を0と1で表してみよう!
 - ★ 何ビットで表現できるか?
 - ★ それぞれどういうビット列で表すか?
 - ★ ある一連の情報をビット列で表すとどうなるか?
 - ★ 例の1週間の朝食や関東各県の天気のような感じ





Question!



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

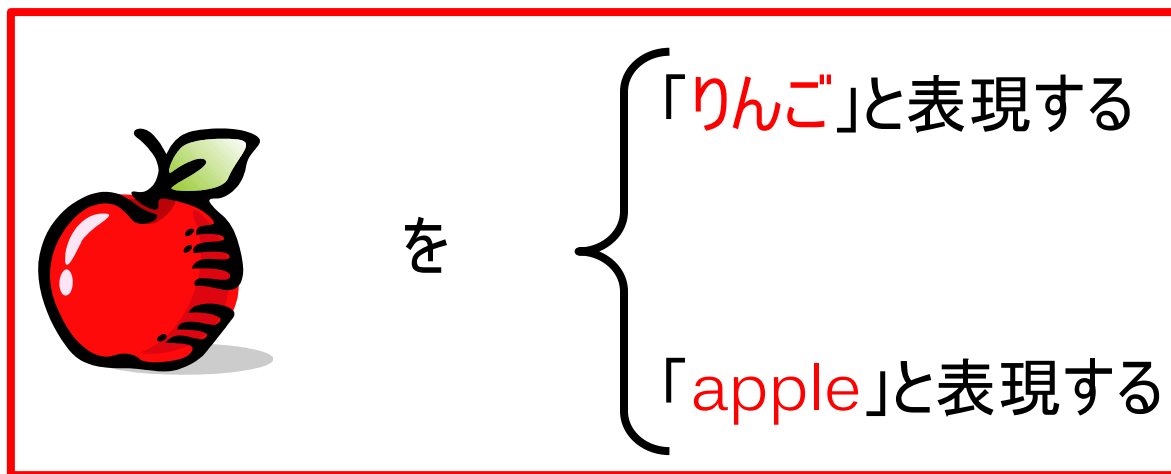
➡ 「2進数で情報を表現している」、と言える

2進数[2](p. 4)

◆ 2進数

◆ 「0」と「1」だけで全ての数を表現

10進数の「50」= 2進数で「110010」



— 表現方法が違うだけ

2進数は、10進数での表現を違う表現にただけ
(数の量などが変わるわけではない)

2進数[3](p. 4)

◆ 2進数

- ◆ 「0」と「1」だけで全ての数を表現
- ◆ 「2」で繰り上がる、という考え方
 - ◆ 10進数: 10で繰り上がる

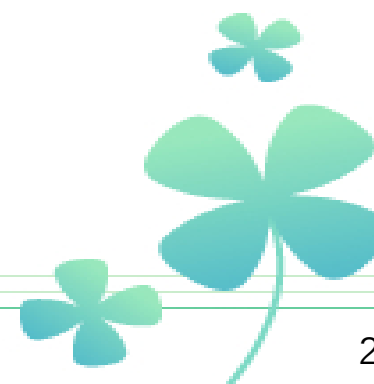
10進数	2進数
0	00
1	01
2	10
3	11



10進数	2進数
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111



繰り上がり



2進数[4](p. 4)

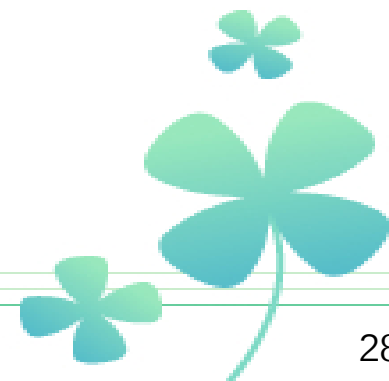
★ n 進数を区別して数を表記する場合: (数) _{n} と表記

★ 10進数: (数)₁₀

★ 2進数: (数)₂

$(100)_{10}$: 10進数の百

$(100)_2$: 2進数の100(10進数で4)



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

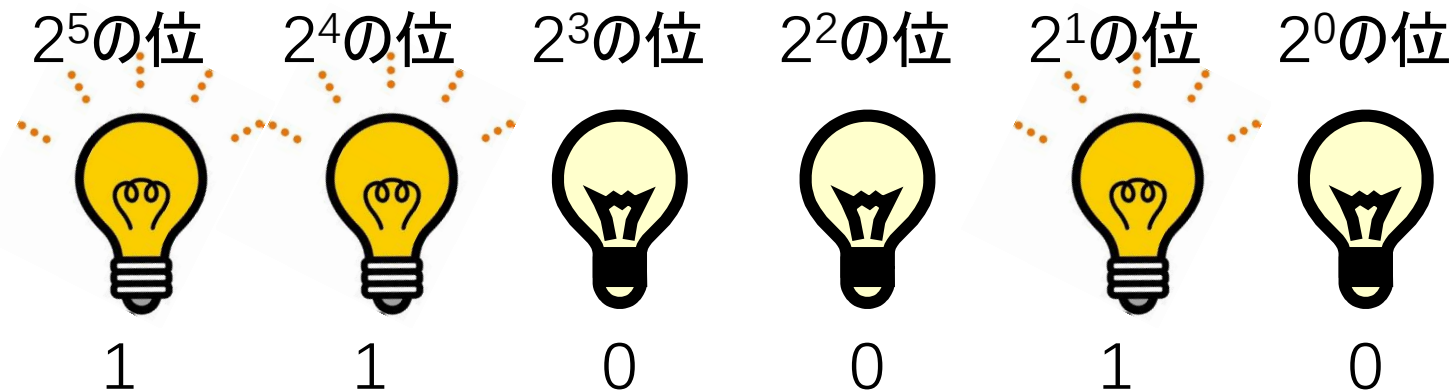
} 人間が番号のつけ方の
ルールを作成



数の考え方

◆ 電気を通す1つ1つの部品を、数の各桁と考える

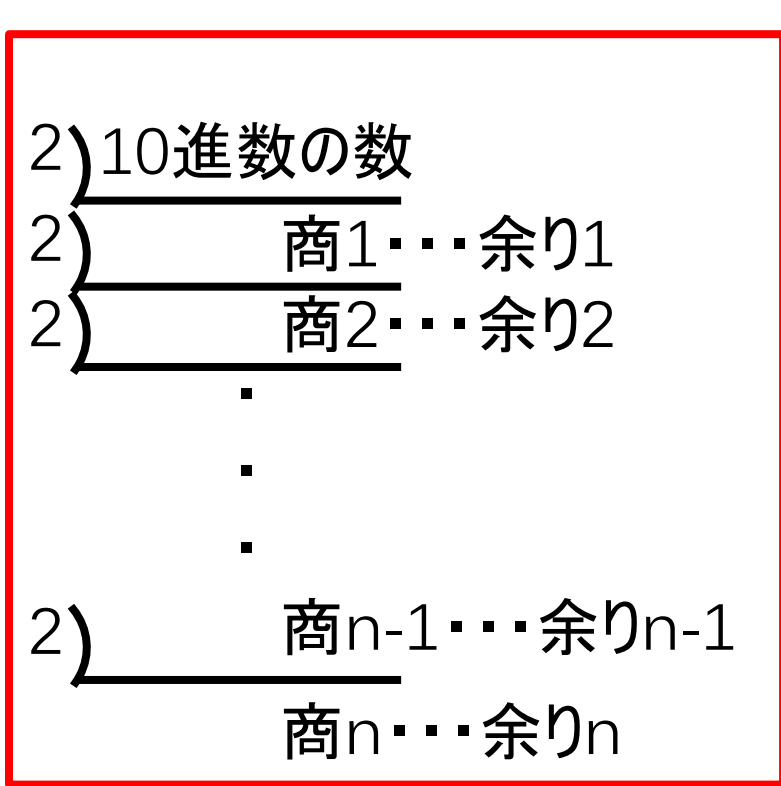
- ◆ ON(電気が通っている)の部品を1、OFF(電気が通っていない)の部品を0とする
- ◆ 各桁は、 2^0 、 2^1 、 2^2 、...、 2^n の位と呼ぶこととする
 - ◆ 1つの桁で表すことができる数は2個だから
 - ◆ ※10進数は、1つの桁で表すことができる数が10個(0～9)なので、 10^n の位と呼ぶ
- ◆ 各桁の数(0または1)と、その桁の位の数(2^n)をかけたものを足し合わせると、1つの数となる



➡ $1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 50$

10進数を2進数に変換

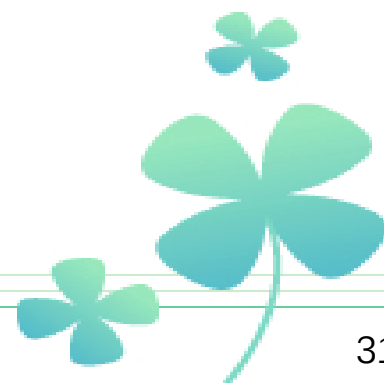
★ 10進数の数は、2進数の表現に直すことができる



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
(余りを の方向に倒す)



10進数を2進数に変換(例)

10進数の13を2進数に変換

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

倒す



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

10進数の50を2進数に変換

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

倒す



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

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

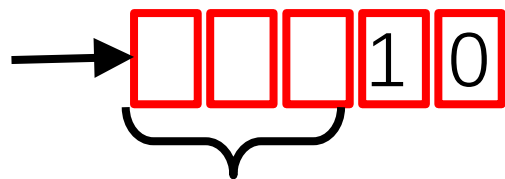
◆ 数はいくつかの部品を使って表す、というように、部品の個数が固定されているため

◆ 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

★ 単純に...

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

1.

2	2	2	2	2	2
1	1	1	0	1	0



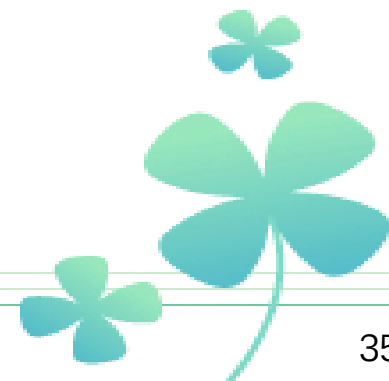
2.

右から左に、0, 1, 2, ...と番号をつける

⁵ 2	⁴ 2	³ 2	² 2	¹ 2	⁰ 2
1	1	1	0	1	0

※ 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

★ 単純に...

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

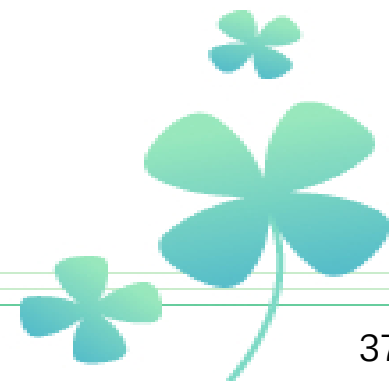
||

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

2進数を10進数に変換[4]

★ $2^0 \sim 2^{10}$ の数は覚えておくと便利

2のべき乗	10進数	2進数
2^0	1	1
2^1	2	10
2^2	4	100
2^3	8	1000
2^4	16	10000
2^5	32	100000
2^6	64	1000000
2^7	128	10000000
2^8	256	100000000
2^9	512	1000000000
2^{10}	1024	10000000000

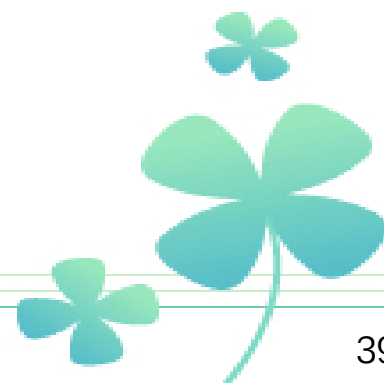


やってみよう! [1]

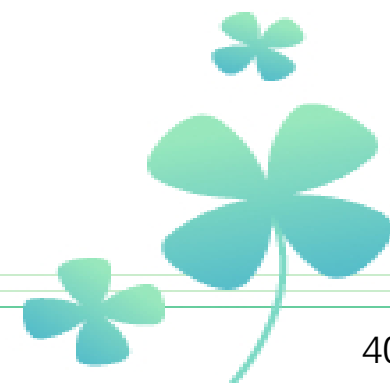
1. 10進数の「25」を2進数に
 2. 10進数の「500」を2進数に
 3. 10進数の「255」を2進数に
 4. 10進数の「135」を10桁の2進数に
 5. 10進数の「200」を12桁の2進数に
 6. 2進数の「001010101010」を10進数に
 7. 2進数の「01111000010」を10進数に
 8. 2進数の「0010000111001」を10進数に
- } ビット数も数えてみよう!

※計算方法は、自分でやりやすい方法があれば、それを使って良い

Question!



2進数での足し算



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

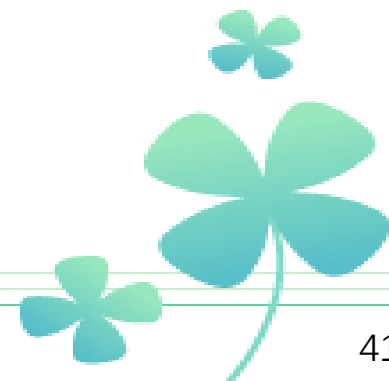
- ◆ $1+1, 1+2, 1+3, \dots 2+1, 2+2, 2+3, \dots \dots 8+6$ (繰り上がり1), $8+7$ (繰り上がり1),
... ..

◆ 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)

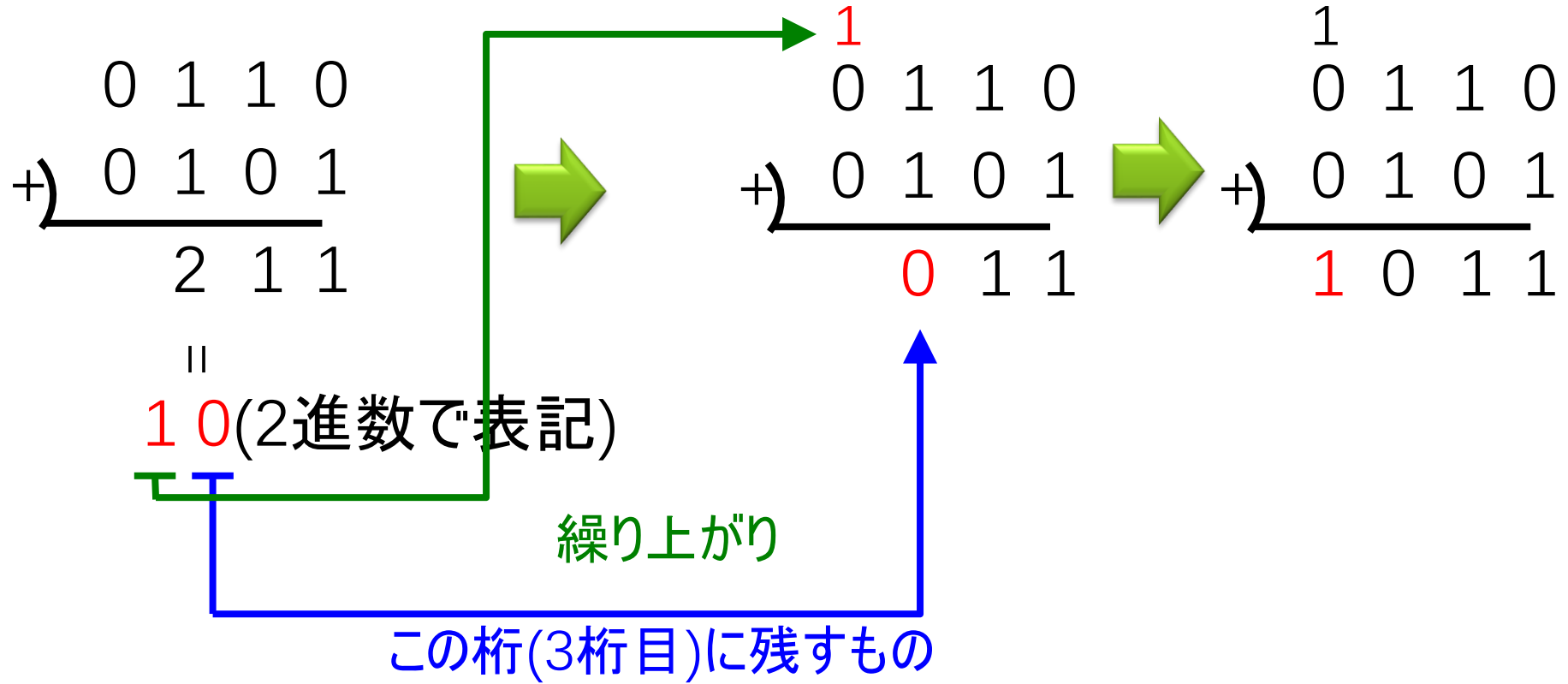
- ◆ $0+0, 0+1, 1+0, 1+1$ (繰り上がり1)



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)

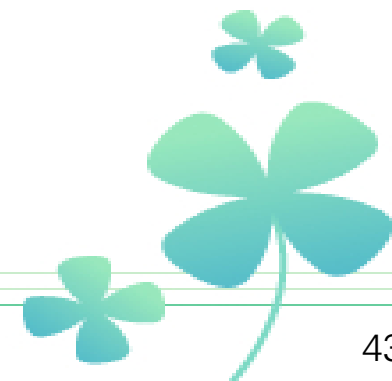
桁あふれ(オーバーフロー)[1]

- ◆ 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

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



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

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

↑ 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)

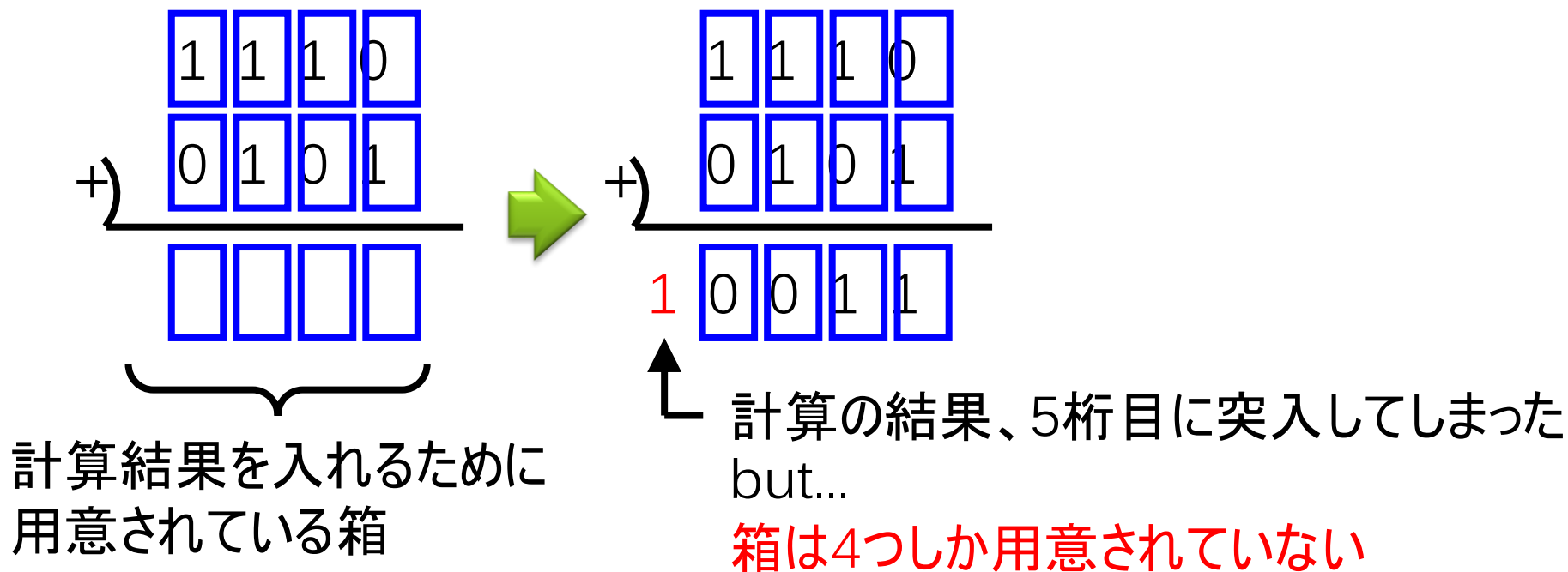
桁あふれ(オーバースhoot)の扱い[1]

- ◆ 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(箱は4個しかない)

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)が違ってしまふ現象

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



5桁目は無視されるので計算結果は $(0011)_2$

やってみよう![2]

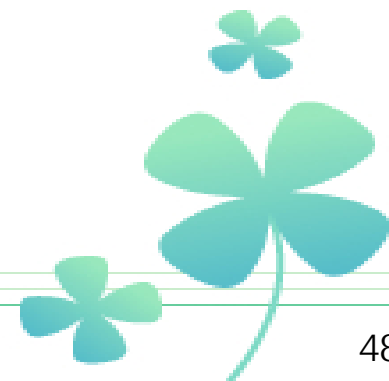
◆ 8ビットの数の足し算をし、結果を10進数で計算すること
(桁あふれも考えて結果を計算すること)

1. $10101010 + 01010101$

2. $11110000 + 01000000$

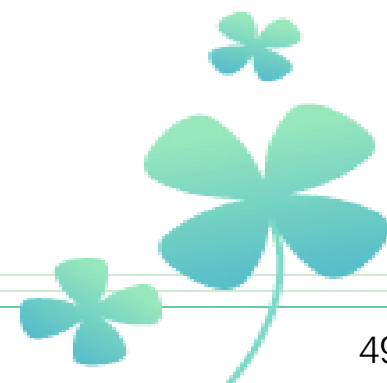
3. $10010010 + 11001100$

◆ 2進数10110を3倍した数を計算すること
(2009年度ITパスポート春期試験問題)

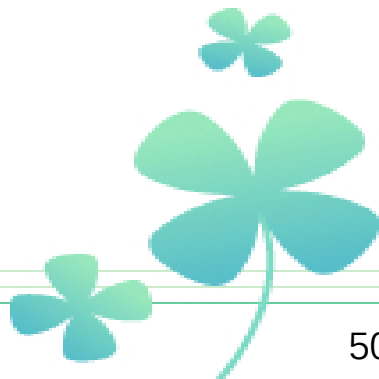




Question!



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

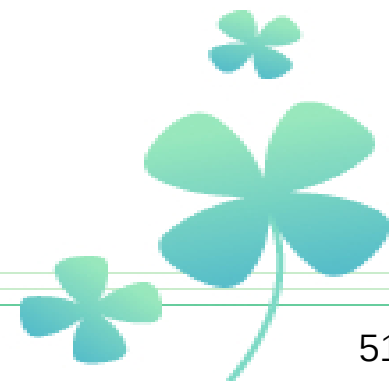
★ 2進数の一番右に、 n 個分「0」をつけるだけ

★ 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:

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

101101000
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:

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

101101.0000000000.....
1011010.0000000000.....
10110100.00000000.....
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 $\div 2^n$

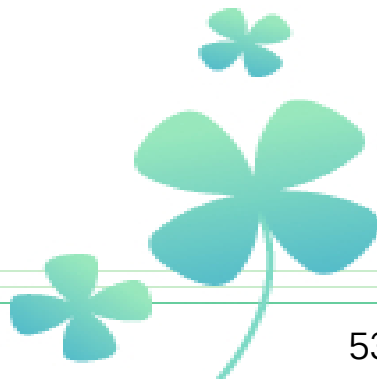
★「2進数 $\div 2^n$ 」の計算も簡単

- ★ 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
(小数点が移動しているだけ)



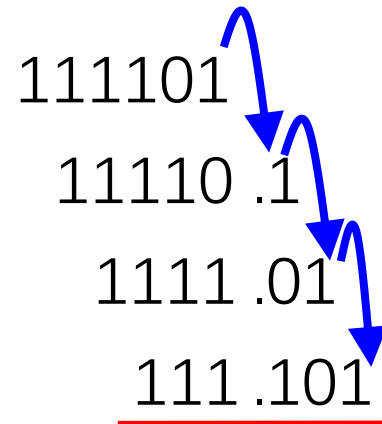
2進数 $\div 2^n$

- ★ 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:

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

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
- ◆ 110101000000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

