演算の仕組みと
コンピュータの限界

教科書P66-67

この時間の目標

- 演算の仕組みとコンピュータの限界

- S　コンピュータの演算の仕組みがよく理解でき、特性を活かして活用しようと思った
- A　コンピュータの演算の仕組みがよく理解できた
- B　コンピュータの演算の仕組みが理解できた
- C　コンピュータの演算の仕組みが理解できなかった

演算の仕組みと
コンピュータの限界

教科書P64-65

コンピュータは足し算だけできる

- 掛け算→何回も足す
- 引き算→繰り上がりを無視する方法(補数)
- 割り算→引き算を繰り返す

- コンピュータの計算
  - 2進法で計算する
  - あらかじめビット数を決める

教科書P66

あらゆる計算は3つの回路で実現できる

- 論理回路
  - 基本論理回路　AND･OR･NOT
  - 回路の組み合わせで計算

教科書P66-67

コンピュータも間違える

- コンピュータの計算
  - あらかじめ定めたビット数で計算
  - 小数などで誤差が出ることも
    - 実用上困らない程度に設定されている

教科書P67

2進法の計算

教科書P64-65

2進法と乗法

- 2倍・4倍・8倍は簡単
  - 0を増やす
  - ビットシフト

- 10倍するには
  - 10回足す
  - 2倍を5回足す
  - 8倍と2倍を足す

	10進法	2進法
	5	101
2倍	10	1010
4倍	20	10100
8倍	40	101000

2進法で負の数を表す

- 何ビットで表すかあらかじめ決める
- 最上位の桁が1の数を負の数とする
  - 4ビットの場合
    - 正の数のみ　　0～15
    - 正負の数　　-8～7

10進法	2進法
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

負の数を使った引き算

- 5-5＝5+(-5)

$$\begin{array}{r} 0101 \\ +1011 \\ \hline 10000 \end{array}$$

桁あふれは
捨てる

10進法	2進法
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

負の数を使った引き算

- 7-5＝7+(-5)

$$\begin{array}{r} 0111 \\ +1011 \\ \hline 10010 \end{array}$$

桁あふれは
捨てる

10進法	2進法
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

2進法の負の数の求め方

- 補数を求めると簡単
  - 反転して1を足す

- 4bitの時　0010　の補数
  - 反転　1101　→1を足す　1110
- 8bitの時　01001110　の補数
  - 反転　10110001　→1を足す　10110010

2進法の計算まとめ

- 最初にビット数を決める
  - 必要なビット数を用意する
- 掛け算
  - ビットシフトと足し算
- 引き算
  - 桁あふれを利用した負の数
  - 補数を利用（反転して1を足す）

コンピュータの限界

Pythonで計算してみよう

- 次の計算を実行
  - 0.2-0.1
  - 0.3-0.2
  - 0.5-0.25
  - 4.3-4.2
  - 0.3-0.24
- 他の値も試してみよう

Excelで計算してみよう

- Excelを起動
  - →[空白のブック]
- 次の計算を実行
  - =0.2-0.1
  - =0.3-0.2
  - =0.5-0.25
  - =4.3-4.2
  - =0.3-0.24

	A	B
1	0.1	
2	0.1	
3	0.25	
4	0.1	
5	0.06	

- 範囲選択して ←.0 .00 ボタンを連打！

= は
[Shift]＋[ほ]

範囲選択は
白十字 ✚ で！

Excelで計算してみよう

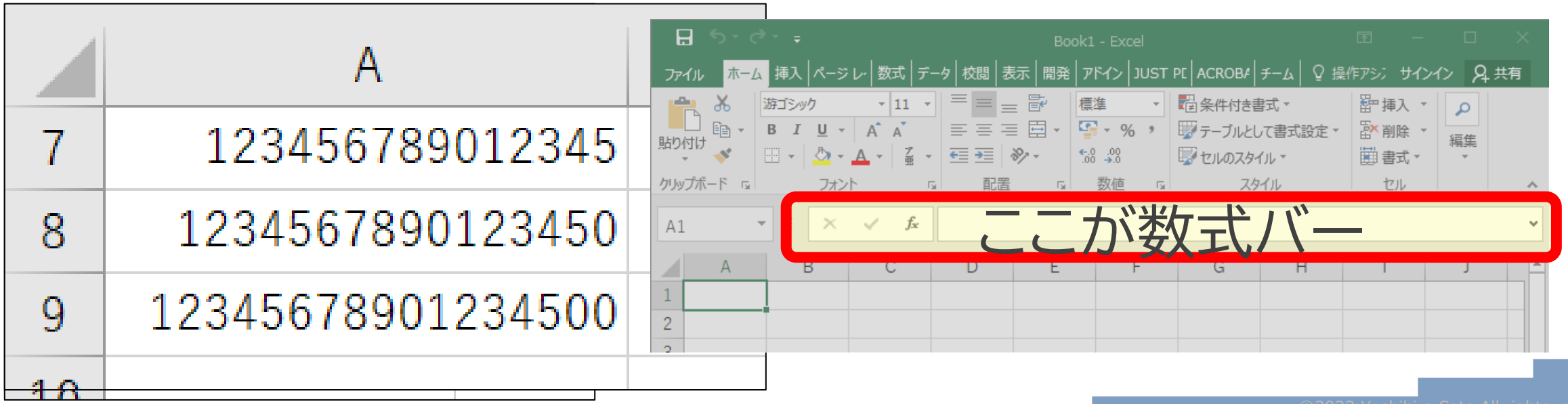
- Excelを起動
  - →[空白のブック]
- 次の計算を実行
  - =0.2-0.1
  - =0.3-0.2
  - =0.5-0.25
  - =4.3-4.2
  - =0.3-0.24

	A
1	0.10000000000000000
2	0.10000000000000000
3	0.25000000000000000
4	0.09999999999999960
5	0.06000000000000000

- 範囲選択して ←.0 .00 ボタンを連打！

Excelに入力してみよう

- 15桁・16桁・17桁・・・の数を入力
  - 123456789012・・・の繰り返しがおすすめ
- 入力したセルをクリックし数式バーを確認する
- 入力した値が切り捨てられていることが確認できる



Pythonで計算してみよう

- 1234567890123 45+1
- 1234567890123 456+1
- 1234567890123 4567+1

論理回路による計算

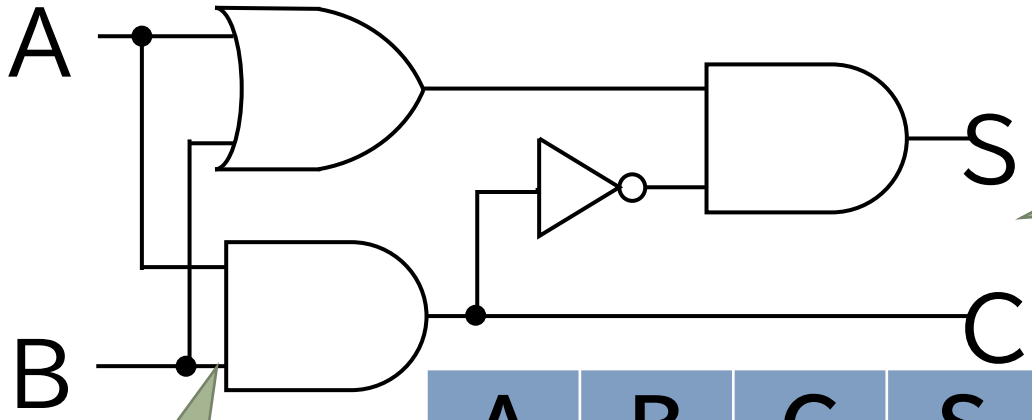
計算する　ということ

- 2進法1桁の計算
  - 0+0=00
  - 0+1=01
  - 1+0=01
  - 1+1=10
- この結果が出る　ということ
- 右の真理値表を満たす機械を作る

入力		出力	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

半加算器

- 2進法1桁どおしの足し算しかできない



この回路で
2進法1桁の
足し算ができる

これを
半加算器
という

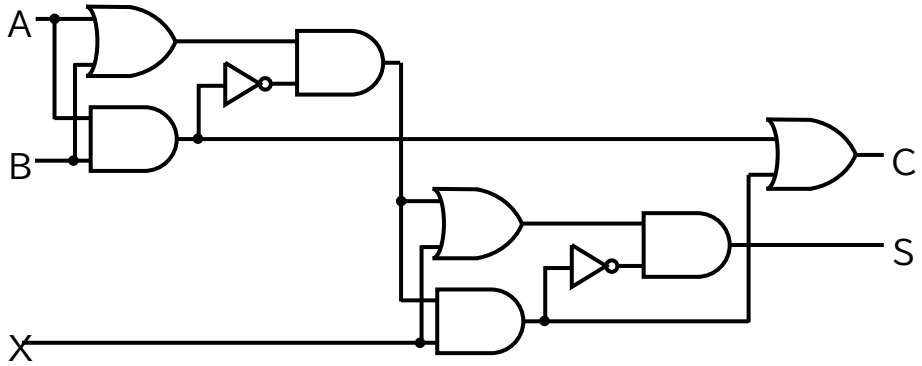
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

2進法1桁の足し算

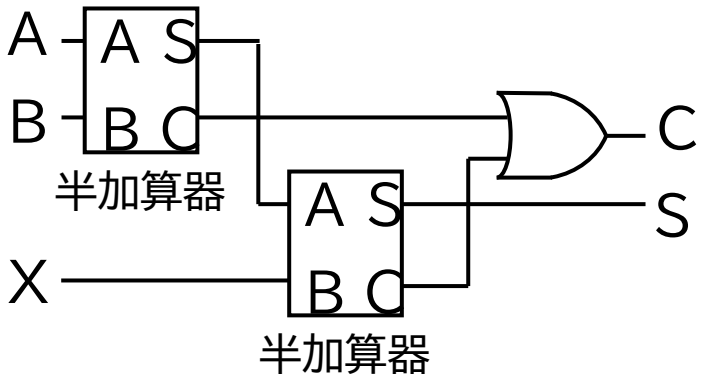
A	B	繰上	答え
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

全加算器

- 全加算器の回路図は以下の通り
  - Xは下の桁からの繰り上がり

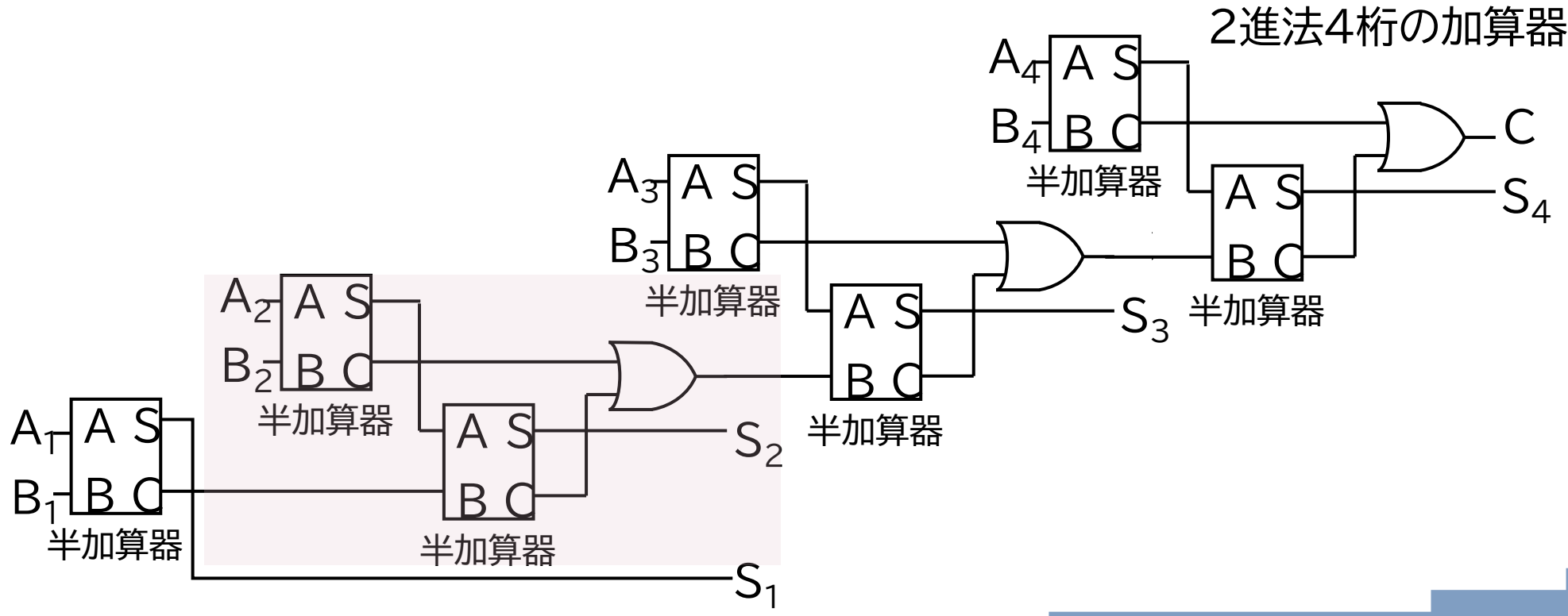


  - 簡略化して右のように記載する



2進法2桁の加算器

- 2進法2桁の加算器の回路図は以下の通り
  - □□ ＋ □□ ＝ □□□
    $A_2\ A_1$　$B_2\ B_1$　$C\ S_2\ S_1$



加算機を作ってみよう

論理回路シミュレータSimcirJS

- 情報Ⅰのページ[論理回路シミュレータ]
- 半加算器を作ってみる
- 全加算機を組み合わせて2桁+2桁