# 京都物理グランプリ2012
# 2ndステージ 物理チャレンジ道場

資料

2013 年 2 月 3 日、24 日

**概要**

*** 2 進数**

*** 論理ゲート**

*** 論理ゲートの組合せ**

*** 基本的な計算機『半加算器』を作ってみよう**

*** 計算機を設計・作成してみよう**

- **(A) 等号判定**
- **(B) 加算**
- **(C) 乗算**
- **(D) 大小判定**
- **(E) 減算**

# 日時

**1 日目（2 月 3 日（日））**13:00〜17:00

**2 日目（2 月 24 日（日））**13:00〜17:00

場所：北部総合教育研究棟 201 号室

# スケジュール

■ **1 日目（2 月 3 日（日））**13:00〜17:00

- 13:00〜14:00
  2 進数
  論理ゲート（AND 回路、OR 回路、NOT 回路）
- 14:00〜14:30
  論理ゲートの組合せ
- 14:30〜15:30
  基本的な計算機『半加算器』を作ってみよう
- 15:30〜16:45
  計算機を設計・作成してみよう

■ **2 日目（2 月 24 日（日））**13:00〜17:00

- 13:00〜14:15
  計算機を設計・作成してみよう（１日目の続き）
- 14:15〜15:45
  発表準備
- 15:45〜16:45
  発表・講評

**必要なもの**

机の上に次のものがあるかを確認してください。

(1) 電源装置　1 台
(2) テスター　5 台
(3) ブレッドボード　9 枚
(4) 先曲がりピンセット　1 人あたり 1 本
(5) ダイオード　BAT41　20 個
(6) 電気回路素子セット（1 つの袋にまとめて入っています）
　（a）電気抵抗　1k Ω 20 本、100 Ω 、10k Ω 、100k Ω 各 10 本ずつ（計 50 本）
　（b）トランジスタ (NPN 型)　2SC1815GR　10 個
　（c）スイッチ　SS-12D00G3　5 個
　（d）LED(発光ダイオード)　5 個
(7) ジャンプワイヤキット　2 セット
(8) ハサミ　1 本[*1]

電気抵抗・ダイオード・トランジスタはこの後、大量に利用します。足りない分は (他の物品でも) チューターに申し出てもらえば、いつでも追加で渡します。

[*1] ハサミは本当は挺や丁と数えます。

# 導入と復習　1 日目 13:00〜14:00

## 2 進数

普通、数を 1 から数えていくと 9 の次は 10 と表記し、桁をずらして「十」の”かたまり”が 1 こと「一」の”かたまり”が 0 こあることを表します。この後も、99 の次は 10 の”かたまり”が 10 こある”かたまり”：百（$= 10^2$）の位を新たに考えて 100、101…と表わし、999 の次は千（$= 10^3$）の位を考えて 1000、1001 と表していきます。このように 10 を基本的な”かたまり”とする表記法を「10 進法」と呼びます。

日常生活では多くの場合は「10 進数」が使われていますが、数字はかならずしも 10 進数で表す必要はありません。たとえば、60 秒で 1 分、60 で 1 時間、時計の文字盤は 12 時でちょうど 1 周で、2 周（24 時間）すると 1 日です。このような場合、基本的な”かたまり”は 60 だったり 24 だったりします。

私たちは今からパソコンなどの中で用いられている論理回路について実験していきますが、そこで用いるのは「10 進数」はなく、「2 進数」です。2 進数では 2 を基本的な”かたまり”と考えるので、使用する数字は「0」と「1」だけです。従って 2 進数の位取りは 1 の位、2 の位、4（$= 2^2$）の位、8（$= 2^3$）の位…となります。2 進数は 2 種類の数字のみを扱うので、機械などでも扱いやすく、現在のコンピューターはほとんどが 2 進数のデジタルコンピューターです。具体的な 2 進数を表 1 に表わしておきます。

表 1　2 進数と 10 進数の対応

| 2 進数 | 10 進数 | 2 進数 | 10 進数 |
|---|---|---|---|
| 0 | 0 | 110 | 6 |
| 1 | 1 | 111 | 7 |
| 10 | 2 | 1000 | 8 |
| 11 | 3 | 1001 | 9 |
| 100 | 4 | 1010 | 10 |
| 101 | 5 | ⋮ | ⋮ |

# 回路図凡例

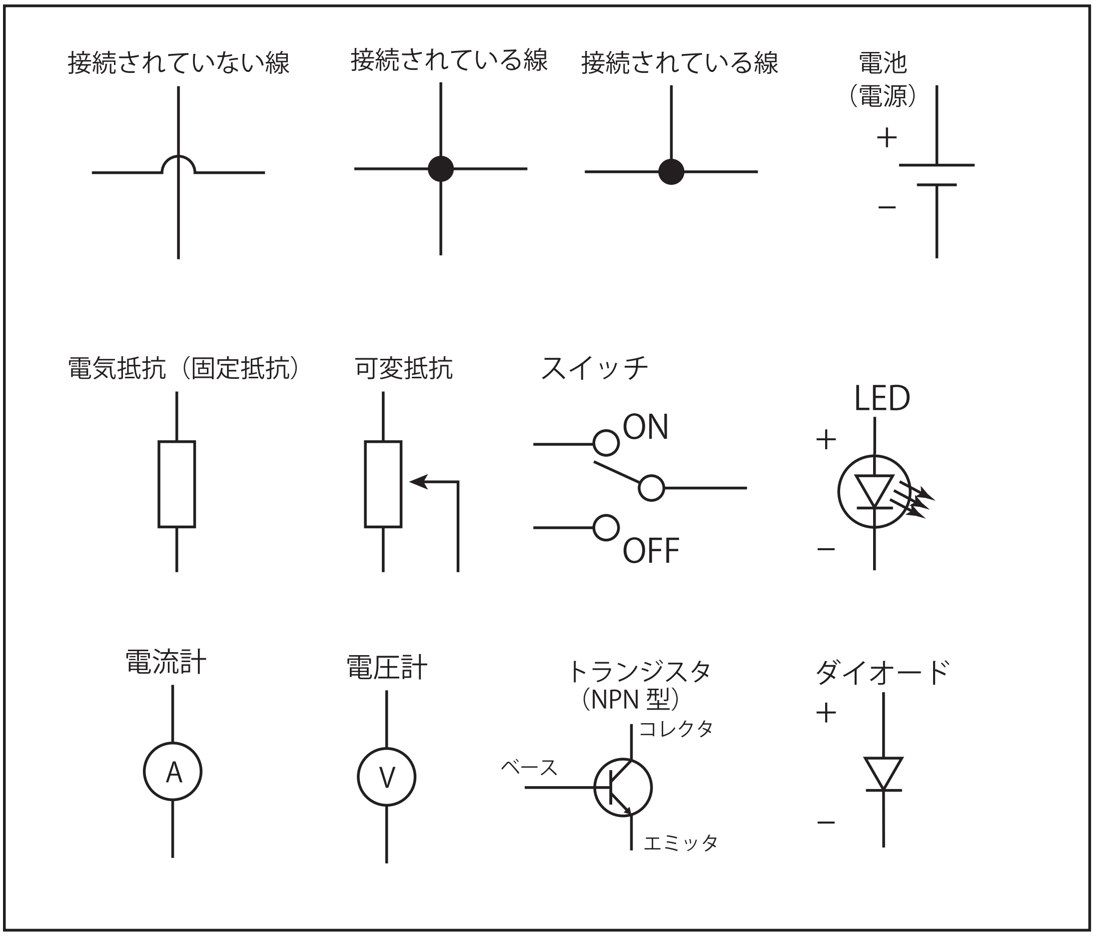


図 1　回路図凡例

# 論理の表現

電子回路で 2 進数を用いるとき、「0」と「1」を、それぞれスイッチの「OFF」と「ON」で表します。具体的には、電源電圧に 5V を使い、電子回路に 0 ~ 1.5V がかかっているときを「0」、3.5 ~ 5V がかかっているときを「1」とします。ここで、「0」と「1」を 0V、5V とせずに幅を持たせているのは、実際の回路では「0」なのにわずかに電圧がかかっていたり、「1」なのに電圧が 5V よりも低かったりすることがあるためです。

## 論理演算

論理演算とは 1 と 0 の 2 つの状態のみを扱う計算のことで、現在のコンピューターも原理的にはこの論理演算によって動いています。全ての論理演算は、これから紹介する AND 回路、OR 回路、NOT 回路の 3 つの回路[*2]をうまく組み合わせて用いることで実現できることが知られています。

| 入力 A | 入力 B | 出力 |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

表 2　AND 回路の真理値表

| 入力 A | 入力 B | 出力 |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

表 3　OR 回路の真理値表

| 入力 | 出力 |
|---|---|
| 0 | 1 |
| 1 | 0 |

表 4　NOT 回路の真理値表

[*2] 実は AND と NOT だけ、OR と NOT だけ、もしくは NAND という 1 つの回路のみで全ての論理演算を表すこともできますが、実用的には AND、OR、NOT の 3 つを組み合わせるのが比較的分かりやすいです。

## よくあるミス

ここから、実際に様々な回路を作っていきます。そこでは次のようなミスがよくあります。うまく回路が動かない時は以下のようなことに注意してみましょう。

(1) 穴のさし間違い
(2) ブレッドボードの穴にちゃんとささっていない。もしくはさしすぎて他の穴の部分と繋がってしまう。
(3) 電源電圧につなぐことやグランドにつなぐことを忘れやすい（特にＮＯＴ回路）
(4) ブレットボードの上（空中）で金属部分が接触してしまう
(5) ダイオードやトランジスタの向きを間違える
(6) ブレッドボード同士の接続忘れ（電源電圧やアース）

## AND 回路

AND 回路を実際に手を動かしてつくってみよう。きちんと出力が正しいかどうかも確かめましょう。この際、テスターで電圧を測ってもよいですし、LED を使って光らせて確かめてみてもよいです。

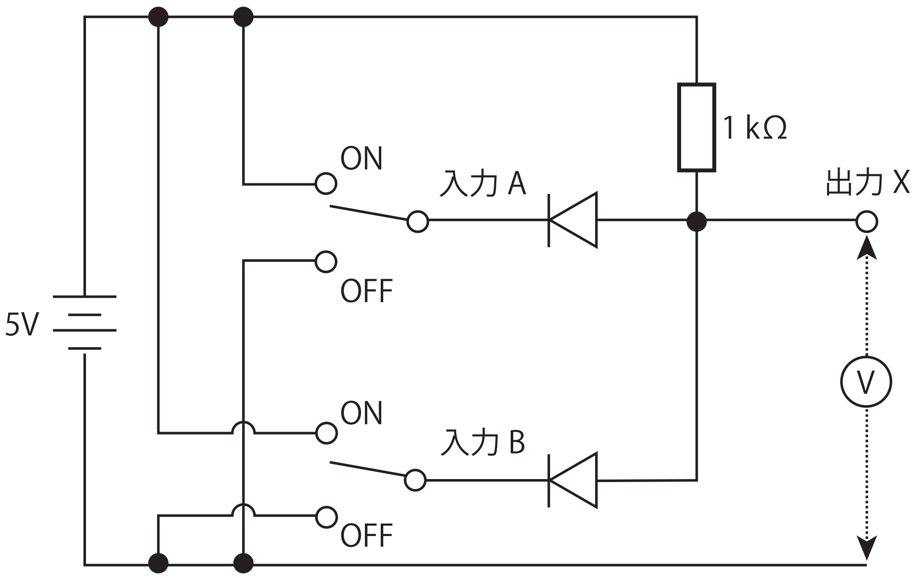


図 2　AND 回路

## NOT 回路

NOT 回路を実際に手を動かしてつくってみよう。

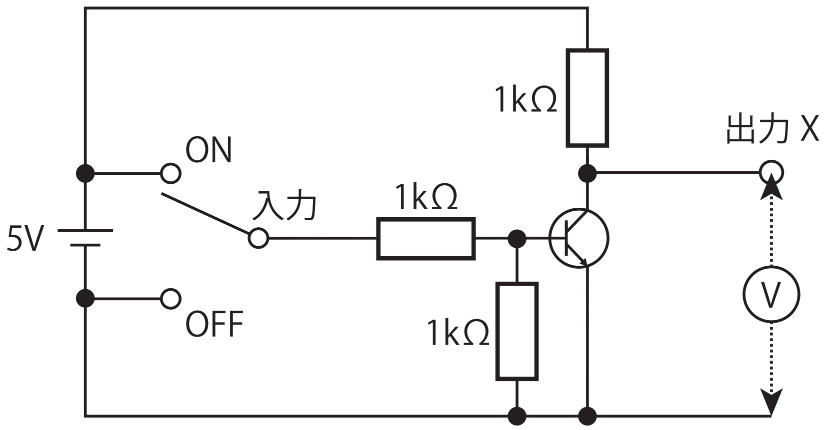


図 3　NOT 回路

## OR 回路

OR 回路を実際に手を動かしてつくってみよう。

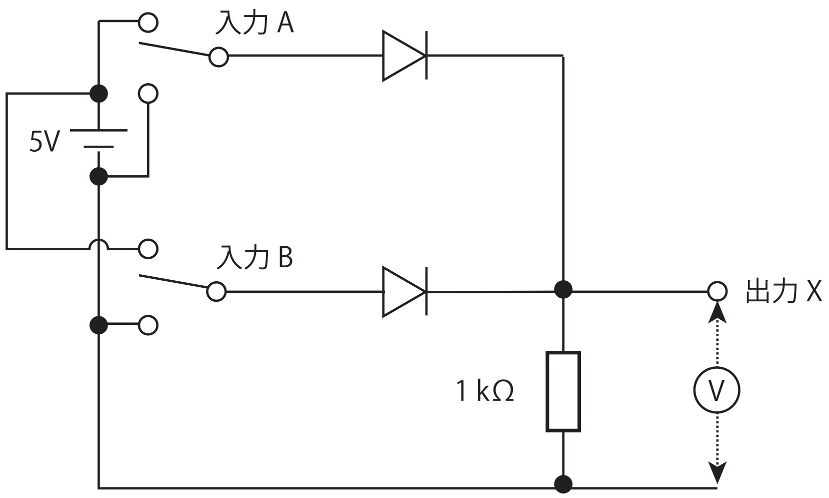


図 4　OR 回路

## 論理記号

ここまで、トラジスタや抵抗などを用いて、図を書いてきました。ここからは繰り返しが多くなるので、それぞれ AND や OR や NOT については、図 5 のように表すことにします。

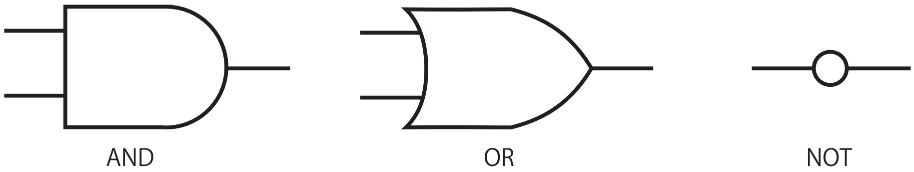


図 5　論理記号。左から、AND、OR、NOT。

# 論理ゲートの組み合わせ　1 日目 14:00〜14:30

論理記号を組み合わせることによって、多様な論理を表現することが可能です。論理学では、NOT と、AND もしくは OR を組み合わせることで、すべての論理が表現できることが知られています。それでは、論理ゲートを組み合わせる練習として、図 6 のような論理記号で表されるような回路*3 をつくってみましょう。

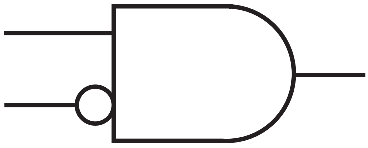

図 6　2 つの入力のうち一方を NOT し、その後 AND をとる回路を論理記号で表したもの

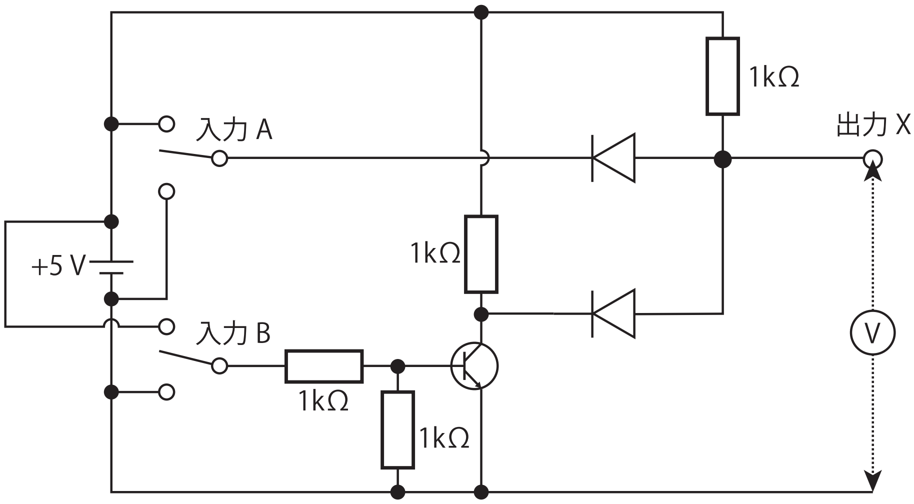


図 7　2 つの入力のうち一方（入力 B）を NOT し、その後 AND をとる回路図

*3 実は次の半加算器のヒントになっている

以下に真理値表を載せていますので、出力が予想通りになるかを確かめましょう。

| 入力 A | 入力 B | not B | 出力 |
|---|---|---|---|
| 0 | 0 | | |
| 0 | 1 | | |
| 1 | 0 | | |
| 1 | 1 | | |

表 5　A かつ not B の真理値表

# 基本的な計算機『半加算器』を作ってみよう
# 1 日目 14:30〜15:30

半加算器（はんかさんき）とは、2 進数で 1 桁の数字 2 つの足し算をする計算機です。論理回路を作る上で基本となるこの半加算器を実際に作ってみましょう。

まずは、半加算器はどのような振る舞いをするかを調べてみましょう。2 進数の 1 桁同士の足し算の場合、全ての場合を書きだしても 4 通り[*4] しかないので簡単に調べられます。

| 入力A | 入力B | 答 |
|---|---|---|
| 0 | 0 | |
| 0 | 1 | |
| 1 | 0 | |
| 1 | 1 | |

表 6　半加算器の計算。**足し算した答を書き込もう**

1 桁同士の足し算でも繰り上がることがあるので、半加算器の出力は同じ桁と繰り上がりの桁の 2 つの出力が必要です。同じ桁と繰り上がりの桁に分けて考えて、次のように真理値表にまとめてみましょう。

| 入力A | 入力B | 繰り上がり | 同じ桁 |
|---|---|---|---|
| 0 | 0 | | |
| 0 | 1 | | |
| 1 | 0 | | |
| 1 | 1 | | |

表 7　半加算器の真理値表　**答を書き込もう**

[*4] 10 進数の場合は 1 桁の足し算でも 100 通りもあります。他の演算でも同じで、10 進数なら九九を（0 の段まで含めて）100 個も覚えないといけませんが、2 進数は九九はたったの 4 つ覚えればよいのです。

真理値表ができたら、次にこの真理値表を実現する回路を考えましょう。

繰り上がり桁の方が簡単です。見覚えのある働きをする回路だと思います。繰り上がり桁ができたら次は同じ桁の出力を考えます。これは少し難しいですが、問題を分割してみましょう。この出力を次のように考えます。

「入力 A が 1　かつ　入力 B が 0」または「入力 A が 0　かつ　入力 B が 1」

このように考えると、同じ桁の出力*5 も論理ゲートを使って表すことができます。

ここで、正しく回路を作っても出力が 3.5V より小さくなってしまう（2.5V ぐらいになる）と思います。そこで、次のような電圧増幅回路を作成します。

## 電圧増幅回路

電気回路中では、抵抗の分圧によって電圧が低くなってしまう現象 (電圧降下) が生じます。例えば、最初の電圧が 5 V であっても、出力時にはその半分やそれよりもさらに小さくなってしまい、最終的には ON と OFF の区別がつかなくなってしまいます。これを回避するために、次の 2 つの方法を組み合わせてみてください。

(1) 出力の電圧が下がってしまったら、NOT 回路を 2 つ組み合わせた増幅回路を通して出力を上げる。
(2) 増幅回路を使った所から下流では使う電気抵抗の抵抗値を大きくする。

*5 このように入力 A と入力 B が異なる時に 1 を、入力 A と B が同じときには 0 を出力するような論理ゲートを exclusive OR(排他的論理和) と言います。

# 計算機を設計・作成してみよう 1 日目 15:30〜16:45/2 日目 13:00〜14:15

内容は、次の 5 つの課題の中からひとつを選択して、進めます。なお、難易度順に並んでいて、等号判定が一番容易で、減算が難しいです。また、割算については、あまりに難易度が高すぎるため選択課題から外します。

(1) 等号判定
(2) 加算
(3) 乗算
(4) 大小判定
(5) 減算

なお、それぞれの内容について、bit 数を減らすことで問題は容易になります。2bit 同士の入力まで、動作する回路は確認されています。2bit と 1bit の組合せのように bit 数に違いがあっても構いません。のちのページにこれらの回路を作る時のヒントや考え方を載せてありますので、参考にしてください。

| 入力 A | 入力 B |
|---|---|
| 1 bit | 1 bit |
| 2 bit | 1 bit |
| 2 bit | 2 bit |

表 8　入力 A と入力 B の組み合わせ

## 電流増幅回路

出力にＬＥＤを使うとき、電圧は足りていても電流が小さいと LED はほとんど光りません。そのようなときに次のようなトランジスタをコレクタ接地（電流増幅回路）で使ってみましょう。

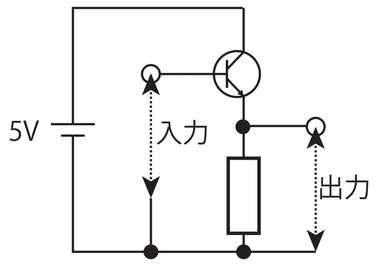


図 8　コレクタ設地回路（電流増幅回路）

# 各計算機作成のヒントや考え方

## 加算回路

### 1bit+1bit

**半加算器**で実現される。

### 2bit+1bit

$A_2A_1 + B_1$ を考える（$A_2$ で A の 2 の位の数字、$A_1$ で A の 1 の位の数字を表す)。1 の位の答は半加算器で $A_1 + B_1$ を計算する。2 の位の答は 1 の位の半加算器の繰り上がりと $A_2$ を半加算器で計算し、その答を 2 の位と 4 の位（繰り上がり）の答えとすればよい。

### 2bit+2bit

$A_2A_1 + B_2B_1$ を考える。1 の位の答は半加算器で $A_1 + B_1$ を計算する。2 の位の答は 1 の位からの繰り上がりと $A_2, B_2$ の 3 つの数字を足す必要がある。その実現のためには半加算器を 2 つ使う。（＝**全加算器**）

## 減算回路

減算は答が負の時の処理がかなり難しい。答が負のときは無視してしまう（エラーとする）なども１つの方法としてあり。実際のコンピューターでも負の数の時は特殊な処理[*6]をしており、この表記方法にならうと少し簡単になる。

### 1bit-1bit

| 入力A | 入力B | 出力（負号） | 出力 (１の位) |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

表 9　1bit-1bit の減算の真理値表

１の位は半加算器の１の位のみの部分と同じ（Exclusive OR）である。

### 2bit-2bit

$A$ の否定を $\overline{A}$ とかく。（つまり、$A=1$ の時、$\overline{A}=0$ である。$A=0$ の時、$\overline{A}=1$）また、$A_2$ で A の 2 の位の数字、$A_1$ で A の 1 の位の数字を表す。この時、

$$B_2B_1+\overline{B_2}\,\overline{B_1}=11_{(2)} \Leftrightarrow -B_2B_1=\overline{B_2}\,\overline{B_1}-11_{(2)}$$

である[*7]ので、

$$A_2A_1-B_2B_1=A_2A_1+\overline{B_2}\,\overline{B_1}-11_{(2)}=A_2A_1+\overline{B_2}\,\overline{B_1}+1_{(2)}-100_{(2)}$$

であることを用いると、減算回路は基本的には加算回路に帰着できる。また、$A_2A_1+\overline{B_2}\,\overline{B_1}+1_{(2)}$ の 4 の位（３桁目）が 0 である時、$A_2A_1-B_2B_1$ は負になるので、この４の位を否定すれば負号として利用できる。（4 の位が 1 ときはマイナス、４の位が０のときはプラスもしくはゼロ）

[*6] 一般的に用いられている方法は 1 番上の桁を符号用に扱い、すべてのビットが 1 のとき、-1 とする方法である。8 ビットの場合、-1 は $11111111_{(2)}$、-2 は $11111110_{(2)}$ として扱われている。
[*7] $11_{(2)}$ の右下のカッコ内の小さな数字は二進数表記であることを表している。

これで減算器として成り立つが、負の数のときは表記が変わっている（脚注参照）。ここまででも十分に良いが、より直感的な出力にするためには次のような操作が必要である。答がプラスになるときは２の位、１の位の数が答の絶対値になるが、答えがマイナスのときは２の位、１の位に出力される数は 4 – (答の絶対値) となるので、以下のように答を書き換える必要がある。

| 出力 (２の位) | 出力 (１の位) | 正しい答の２の位 | 正しい答の１の位 |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

表 10　答が負の時の出力と正しい答の対応

表を見ると、１の位はそのまま出力すればよいが、２の位では「答がマイナスの時（４の位の出力が 1)」かつ「１の位の出力が１の時」には否定したものが答えになる。

# 乗算回路

## 1bit × 1bit

| 入力A | 入力B | 出力 |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

表 11　1bit+1bit の乗算の真理値表

1bit × 1bit の計算はＡＮＤ回路そのものである。

## 2bit × 1bit

2bit × 1bit の乗算も各桁に分解すると、1bit × 1bit の掛け算と同じものを繰り返せばよい。

## 2bit × 2bit

筆算と同様に考える。桁ごとに分けて考えたとき、それぞれの桁の掛け算はＡＮＤ回路でできる。その結果を半加算器を用いて加えればよい。

# 等号判定

## 1bit& 1bit

| 入力A | 入力B | 出力 |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

表 12　1bit & 1bit の等号判定の真理値表

入力が 0 & 0 もしくは 1 & 1 なら良い。

## 2bit& 2bit

全ての桁が等しい必要があるので、各桁で 1bit のときと同じ判定をして、全ての出力が 1 なら良いので、AND でつなげばよい。

# 大小判定

## 1bit & 1bit

| 入力A | 入力B | 出力 |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

表 13　1bit & 1bit の大小判定の真理値表

## 2bit & 2bit

やり方その１：真理値表をすべての場合について書き下せばよい。

やり方その２：2bit の減算回路と同じように考えると (減算回路の項参照)、

$$A_2A_1 - B_2B_1 = A_2A_1 + \overline{B_2}\,\overline{B_1} + 1_{(2)} - 100_{(2)}$$

であるので、$A_2A_1$ と $B_2B_1$ の大小関係は上式の値の正負で決まる。従って、$A_2A_1 + \overline{B_2}\,\overline{B_1} + 1_{(2)}$ の 4 の位が１の時、$A_2A_1 \geq B_2B_1$ であり、$A_2A_1 + \overline{B_2}\,\overline{B_1} + 1_{(2)}$ の４の位が０のとき、$A_2A_1 < B_2B_1$ である。ゆえにこの回路の４の位が大小判定になる。

資料作成：山本拓弥
京都大学大学院理学研究科　修士１年生

**京都物理グランプリ 2012**
**2nd ステージ 物理チャレンジ道場**