

目次

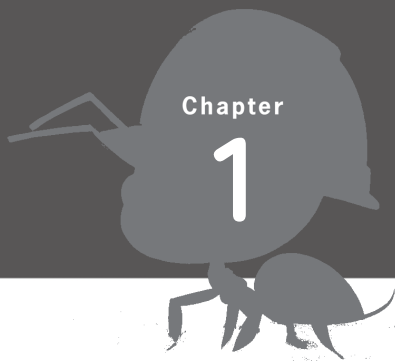
Chapter 1

| | |
|----------------------------|----------|
| コンピュータの重箱の隅 | 3 |
| 1-1 論理回路から加算器は飛躍しすぎ？ | 4 |
| 1-1-1 論理回路の誕生 5 | |
| 1-2 通信機能は5大装置のどこに入る？ | 7 |
| 1-3 デジタルとアナログ | 9 |
| 1-3-1 ルビサンプル 12 | |
| 1-3-2 数式サンプル 14 | |

Chapter 2

| | |
|----------------------------|-----------|
| コンピュータの重箱の隅 | 17 |
| 2-1 論理回路から加算器は飛躍しすぎ？ | 18 |
| 2-1-1 論理回路の誕生 19 | |
| 2-2 通信機能は5大装置のどこに入る？ | 21 |
| 2-3 デジタルとアナログ | 23 |
| 2-3-1 ルビサンプル ?? | |
| 2-3-2 数式サンプル ?? | |

白ページが入る場合はhr要素で改ページ



Chapter

1

コンピュータの 重箱の隅

章のリード文。章のリード文。章のリード文。章のリード文。
章のリード文。章のリード文。章のリード文。章のリード文。

論理回路から加算器は飛躍しすぎ？



過去に「コンピュータの歴史」を**ブール論理 (Boolean Logic)**の誕生あたりからたどろうとして、いろいろと調べてみたのだが、**ジョージ・ブール氏**の考えたものは現在ブール論理として習うものとかかなり違っていた。ブール氏が考えたのは、言葉で推論するどうしてもあまいまさが生じるので、**数式を使って推論すべきだ**というものだった。実際、ブール氏の論文を見ると、足し算や掛け算らしき数式しか書かれていない。

| | |
|---|---|
| <p>CHAPTER VII. OF ELIMINATION 76</p> <p>Substitute these expressions for x and $1-x$ in the fundamental equation</p> $x(1-x) = 0,$ <p>and there results</p> $\frac{f(0)f(1)}{f(0)-f(1)} = 0;$ <p>or, $f(1)f(0) = 0,$ (1)</p> <p>the form required.</p> <p>6. It is seen in this process, that the elimination is really effected between the given equation $f(x) = 0$ and the universally true equation $x(1-x) = 0$, expressing the fundamental law of logical symbols, que logical. There exists, therefore, no need of more than one premise or equation, in order to render possible the elimination of a term, the necessary law of thought virtually supplying the other premise or equation. And though the demonstration of this conclusion may be exhibited in other forms, yet the same element furnished by the mind itself will still be virtually present. Thus we might proceed as follows:</p> <p>Multiply (1) by x, and we have</p> $f(1)x = 0, \quad (2)$ <p>and let us seek by the forms of ordinary algebra to eliminate x from this equation and (1).</p> <p>Now if we have two algebraic equations of the form</p> $ax + b = 0,$ $a'x + b' = 0,$ <p>it is well known that the result of the elimination of x is</p> $ab' - a'b = 0 \quad (3)$ <p>But comparing the above pair of equations with (1) and (2) respectively, we find</p> $a = f(1) - f(0), \quad b = f(0);$ $a' = f(1) \quad b' = 0;$ <p>which, substituted in (4), give</p> $f(1)f(0) = 0,$ <p>as before. In this form of the demonstration, the fundamental equation $x(1-x) = 0$, makes its appearance in the derivation of (3) from (1).</p> <p>7. I shall add yet another form of the demonstration, partaking of a half logical character, and which may set the demonstration of this important theorem in a clearer light.</p> | <p>CHAPTER VII. OF ELIMINATION 77</p> <p>We have as before</p> $f(1)x + f(0)(1-x) = 0.$ <p>Multiply this equation first by x, and secondly by $1-x$, we get</p> $f(1)x = 0 \quad f(0)(1-x) = 0.$ <p>From these we have by solution and development,</p> $f(1) = \frac{0}{1-x} = 0(1-x), \text{ on development,}$ $f(0) = \frac{0}{1-x} = 0x.$ <p>The direct interpretation of these equations is—</p> <p>1st. Whatever individuals are included in the class represented by $f(1)$, are not x's.</p> <p>2nd. Whatever individuals are included in the class represented by $f(0)$, are x's.</p> <p>Whence by common logic, there are no individuals at once in the class $f(1)$ and in the class $f(0)$, i.e. there are no individuals in the class $f(1)f(0)$. Hence,</p> $f(1)f(0) = 0. \quad (4)$ <p>Or it would suffice to multiply together the developed equations, whence the result would immediately follow.</p> <p>8. The theorem (5) furnishes us with the following Rule:</p> <p>TO ELIMINATE ANY SYMBOL FROM A PROPOSED EQUATION.</p> <p>RULE.—The terms of the equation having been brought, by transposition if necessary, to the first side, give to the symbol successively the values 1 and 0, and multiply the resulting equations together.</p> <p>The first part of the Proposition is now proved.</p> <p>9. Consider in the next place the general equation</p> $f(x, y) = 0;$ <p>the first member of which represents any function of x, y, and other symbols. By what has been shown, the result of the elimination of y from this equation will be</p> $f(x, 1)f(x, 0) = 0;$ <p>for such is the form to which we are conducted by successively changing in the given equation y into 1, and y into 0, and multiplying the results together.</p> <p>Again, if in the result obtained we change successively x into 1, and x into 0, and multiply the results together, we have</p> $f(1, 1)f(1, 0)f(0, 1)f(0, 0) = 0; \quad (5)$ |
|---|---|

The Mathematical Analysis of Logic by George Boole

The Mathematical Analysis of Logic by George Boole

<https://www.gutenberg.org/ebooks/36884>にリンクしたい

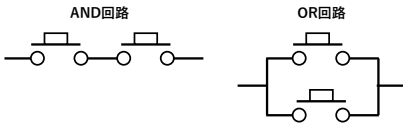
現在使われている論理記号やAND、ORなどの用語は、後世に**数理論理学**として発展して



いく中で作られたもので、最初のとっかかりのアイデアを考えた人として、ブール氏の名前を冠することにしたらしい。

論理回路の誕生

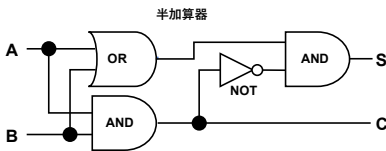
ブール氏が論文を書いたから 100 年ほどして、電信に使われるリレーを使った**論理回路 (Logic Circuit)** が誕生した。**クロード・シャノン氏**が論理回路についての論文を書いたのが 1937 年で、ほぼ同時期に最初のデジタルコンピュータも誕生している。このあたり順番がはっきりしなくてモヤッとするが、たいていの発明・発見は、複数の人が同時期に同じ答えにたどりつくもののようなのだ。



論理回路 (Logic Circuit)

ここで不思議なのは、スイッチを直列に並べて AND 回路、並列に並べて OR 回路が作れるというのは、ブール論理の延長として理解できる。しかし、その先、つまり**論理回路を使って計算する**という発想はどこからでてきたのだろうか？

2 進数 1 桁の計算を行う**半加算器 (Half Adder)** は、AND 回路 2 つと OR 回路 1 つ、NOT 回路 1 つの組み合わせで作れる。



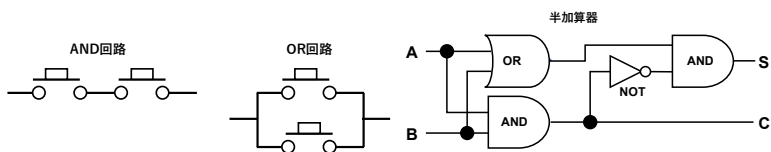
半加算器 (Half Adder)

しかし、これがブール論理の延長で、普通に出てくるとはちょっと考えにくい気がする。ブール論理は「AND (かつ)」「OR (または)」などで真偽を求めるものであって、四則演算するものではないはずだ。

残念ながら、加算器の発明者の資料は見つけられなかった。おそらく、何かを作ろうとし

て論理回路の組み合わせを考えているうちに、「 $0+0=00$ 」「 $0+1=01$ 」「 $1+0=01$ 」「 $1+1=10$ 」という結果を出せば足し算する回路が作れるという答えにパツとたどりついたのだろう。

世の中には頭のいい人はたくさんいるので、このぐらいは複数の人が同時に思いつくものなのかもしれない。



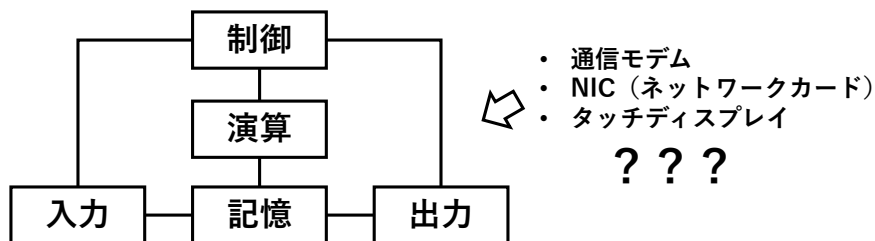
論理回路と半加算器

通信機能は5大装置のどこに入る？



コンピュータサイエンスの初歩として必ず登場するのが、**ノイマン型コンピュータ**（von Neumann architecture）もしくは**コンピュータの5大装置**だ。コンピュータは「制御」「演算」「記憶」「入力」「出力」という5つの装置で構成されるという説で、**フォン・ノイマン氏**がその論文を書いたので、ノイマン型と呼ぶ。ノイマン氏は悪魔のように頭がいい人だったが、5大装置をノイマン氏自身がゼロから考えたわけではなく、**EDVAC**という開発中のコンピュータを見学して、「つまり、こういうことだろ？」と論文をまとめたそうだ。

- **制御：記憶装置からプログラムを読み込み、各装置に指示を出す**
 - CPU
- **演算：四則演算などの計算を行う**
 - CPU
- **記憶：プログラムやデータを記憶する**
 - メモリ
 - HDD
 - SSD
- **入力：外部から情報を取り込む**
 - キーボード
 - マウス
- **出力：外部に情報を書き出す**
 - ディスプレイ
 - スピーカー



von Neumann architecture

今のパソコンやスマートフォンでも、5大装置という構成は変わらないので、今の教科書にもそのまま出てくる。ただ、5大装置の図に、実際のCPUやメモリ、キーボードなどの部品を当てはめていくと、ふと困ったことに気付く。通信モデムやNIC、タッチディスプレイなど、5つのカテゴリにうまく当てはまらない機器が結構出てくるのだ。

そのときはしかたなく、入力と出力を図の右側に配置して、「入出力兼用の機器もある」と説明したのだが、もっとうまいやり方があったのではと今でもときどき思う。

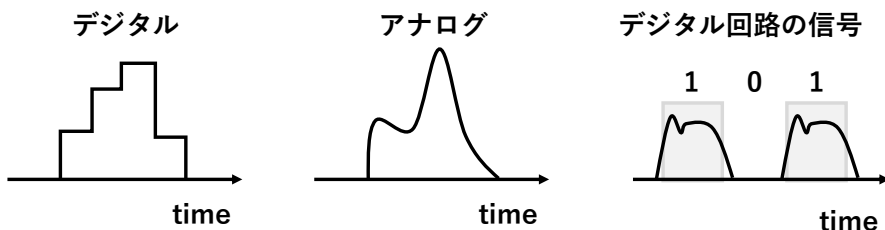
ノイマン氏に聞いてみたいところだが、悪魔のように頭のいい人だったらしいので、「馬鹿め、そのぐらい自分で考えろ」といわれそうだ。

デジタルとアナログ



よくコンピュータは**デジタル**だという。デジタルは「連続した量」を意味する**アナログ**の対義語で、連続していないという意味から日本語では「離散的な値」ともいう。電気や光、音といった自然現象はすべてアナログだ。

ここでよく考えてみると、コンピュータの電子回路も電気が流れているからアナログではないかという答えにたどり着く。実際のところ、コンピュータの電気信号も連続的に変化していて微妙なブレがあり、この範囲にだいたいおさまったら「1」、この範囲だったら「0」という感じに決めて、デジタルと見なしているようだ。



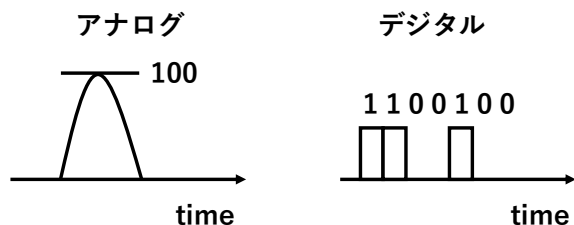
digital analog

自然界にはアナログしか存在せず、コンピュータもアナログ。ということは、純粋なデジタルは、人間の頭の中にしか存在しないということになる。加えていうとプログラム（ソフトウェア）も、人間のデジタル的な考え方を反映したものであるので、デジタルといえる。

デジタルとアナログについて考えていくと面白いことがいろいろあって、昔のビデオテープや音楽テープはすべてアナログで、デジタル化したDVDやCDが登場したのは結構あとのことだ。これはなぜかということ、デジタルのほうがデータ量が増えるからだ。

例えばアナログで「100」を表す場合、単位や媒体は何かはわからないが、100の強さの信号があればいい。デジタルで100を表す場合、2進数だと「01100100」となるので7ビット必要になる。仮に1秒間に1つしか信号を送れないとすると、アナログでは1秒で済むと

ころが、デジタルでは7秒必要という計算になる。つまり、デジタルのほうがデータ量が増えるわけで、これを解決するには電子回路をはるかに高速にするしかない。その発展に時間がかかったわけだ。



digital analog

データ量が増える以外のデジタルの弱点として、連続して変化するアナログを、段階的なデジタルに変換する過程で、変化途中の細かな情報が切り捨てられるという点がある。アナログの信号（レコードとかフィルムとか）だったら、昔は読み取れなかった情報が未来の技術で読み取れる可能性があるが、デジタルの信号（DVDとかBDとか）はいつまで経っても記録時点のままなので、切り捨てられた情報は未来永劫復元できない。最近よく聞くデジタルリマスターも、大元のフィルムから起こすのであって、DVDを原本とすることはない。

などと書きつつも、最近のAI技術では「存在しないものをそれらしく復元する」こともできてしまうので、将来的には何とかなってしまうかもしれないなとも思う。未来を見通すのは難しい。



| ●列見出し | ●列見出し（中央そろえ） | ●列見出し（右そろえ） |
|-------|---------------|-------------|
| セル | セル | セル |
| セル | セル | 90 |
| セル | セルの中に長めの文を入れる | 1000 |
| ●列見出し | ●列見出し（中央そろえ） | ●列見出し（右そろえ） |
| セル | セル | セル |
| セル | セル | 90 |
| セル | セルの中に長めの文を入れる | 1000 |

ルビサンプル

コンピュータサイエンスの初歩^{しよ ぽ}として必ず登^{かなら}場^{どうじょう}するのが、ノイマン型^{がた}コンピュータ (von Neumann architecture) もしくはコンピュータの5大装置^{だいごう ち}だ。

コンピュータサイエンスの初歩^{しよ ぽ}として必ず登^{かなら}場^{どうじょう}するのが、ノイマン型^{がた}コンピュータ (von Neumann architecture) もしくはコンピュータの5大装置^{だいごう ち}だ。コンピュータは「制御」「演算」「記憶」「入力」「出力」という5つの装置で構成されるという説で、フォン・ノイマン氏がその論文を書いたので、ノイマン型と呼ぶ。

```
from pathlib import Path
from bs4 import BeautifulSoup

result = '## 目次{#toc role="doc-toc"}\n'
outpath = Path('tocoutput.md')
outpath.write_text(result, encoding='utf-8')
```

generate_toc.py

```
from pathlib import Path
from bs4 import BeautifulSoup

result = '## 目次{#toc role="doc-toc"}\n'
outpath = Path('tocoutput.md')
outpath.write_text(result, encoding='utf-8')
```

ここまでは本文です。

Column コラム記事タイトル

ここは**囲み記事**にします。囲み記事は<div class="column-article">と</div>で囲んでください。ここは**囲み記事**にします。囲み記事は<div class="column-article">と</div>で囲んでください。ここは**囲み記事**にします。囲み記事は<div class="column-article">と</div>で囲んでください。

数式サンプル

インライン数式は $y = \frac{1}{x^2}$ こんな感じになります。

ディスプレイ数式は

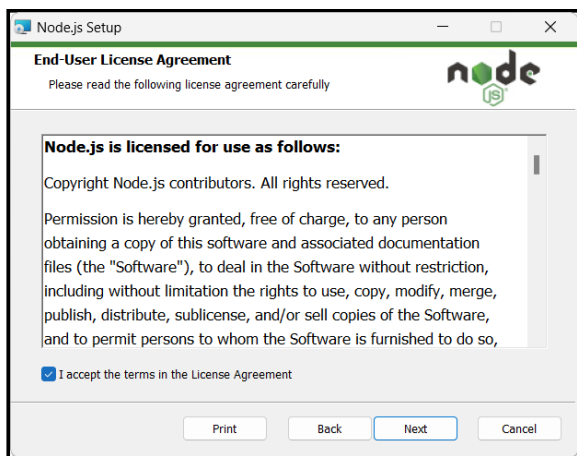
$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right)$$

となります。

$$\begin{aligned} \frac{2}{3} + \frac{1}{5} &= \frac{10}{15} + \frac{3}{15} \\ &= \frac{13}{15} \end{aligned}$$

SVG
20mm
四方

CSS
20mm
四方



RGB そのままの画像



Chapter

2

コンピュータの 重箱の隅

章のリード文。章のリード文。章のリード文。章のリード
文。章のリード文。章のリード文。章のリード文。章のリー
ド文。

論理回路から加算器は飛躍しすぎ？



過去に「コンピュータの歴史」を**ブール論理 (Boolean Logic)**の誕生あたりからたどろうとして、いろいろと調べてみたのだが、**ジョージ・ブール氏**の考えたものは現在ブール論理として習うものとかかなり違っていた。ブール氏が考えたのは、言葉で推論するどうしてもあまいさが生じるので、**数式を使って推論すべきだ**というものだった。実際、ブール氏の論文を見ると、足し算や掛け算らしき数式しか書かれていない。

| | |
|---|---|
| <p>CHAPTER VII. OF ELIMINATION 76</p> <p>Substitute these expressions for x and $1-x$ in the fundamental equation</p> $x(1-x) = 0,$ <p>and there results</p> $\frac{f(0)f(1)}{f(0)-f(1)} = 0;$ <p>or, $f(1)f(0) = 0,$ (1)</p> <p>the form required.</p> <p>6. It is seen in this process, that the elimination is really effected between the given equation $f(x) = 0$ and the universally true equation $x(1-x) = 0$, expressing the fundamental law of logical symbols, que logical. There exists, therefore, no need of more than one premise or equation, in order to render possible the elimination of a term, the necessary law of thought virtually supplying the other premise or equation. And though the demonstration of this conclusion may be exhibited in other forms, yet the same element furnished by the mind itself will still be virtually present. Thus we might proceed as follows:</p> <p>Multiply (1) by x, and we have</p> $f(1)x = 0, \quad (2)$ <p>and let us seek by the forms of ordinary algebra to eliminate x from this equation and (1).</p> <p>Now if we have two algebraic equations of the form</p> $ax + b = 0,$ $a'x + b' = 0;$ <p>it is well known that the result of the elimination of x is</p> $ab' - a'b = 0 \quad (3)$ <p>But comparing the above pair of equations with (1) and (2) respectively, we find</p> $a = f(1) - f(0), \quad b = f(0);$ $a' = f(1) \quad b' = 0;$ <p>which, substituted in (4), give</p> $f(1)f(0) = 0,$ <p>as before. In this form of the demonstration, the fundamental equation $x(1-x) = 0$, makes its appearance in the derivation of (3) from (1).</p> <p>7. I shall add yet another form of the demonstration, partaking of a half logical character, and which may set the demonstration of this important theorem in a clearer light.</p> | <p>CHAPTER VII. OF ELIMINATION 77</p> <p>We have as before</p> $f(1)x + f(0)(1-x) = 0.$ <p>Multiply this equation first by x, and secondly by $1-x$, we get</p> $f(1)x = 0 \quad f(0)(1-x) = 0.$ <p>From these we have by solution and development,</p> $f(1) - \frac{0}{1-x} = \frac{0}{1-x}(1-x), \text{ on development,}$ $f(0) = \frac{0}{1-x} = \frac{0}{1}x.$ <p>The direct interpretation of these equations is—</p> <p>1st. Whatever individuals are included in the class represented by $f(1)$, are not x's.</p> <p>2nd. Whatever individuals are included in the class represented by $f(0)$, are x's.</p> <p>Whence by common logic, there are no individuals at once in the class $f(1)$ and in the class $f(0)$, i.e. there are no individuals in the class $f(1)f(0)$. Hence,</p> $f(1)f(0) = 0. \quad (4)$ <p>Or it would suffice to multiply together the developed equations, whence the result would immediately follow.</p> <p>8. The theorem (5) furnishes us with the following Rule:</p> <p>TO ELIMINATE ANY SYMBOL FROM A PROPOSED EQUATION.</p> <p>RULE.—The terms of the equation having been brought, by transposition if necessary, to the first side, give to the symbol successively the values 1 and 0, and multiply the resulting equations together.</p> <p>The first part of the Proposition is now proved.</p> <p>9. Consider in the next place the general equation</p> $f(x, y) = 0;$ <p>the first member of which represents any function of x, y, and other symbols. By what has been shown, the result of the elimination of y from this equation will be</p> $f(x, 1)f(x, 0) = 0;$ <p>for such is the form to which we are conducted by successively changing in the given equation y into 1, and y into 0, and multiplying the results together.</p> <p>Again, if in the result obtained we change successively x into 1, and x into 0, and multiply the results together, we have</p> $f(1, 1)f(1, 0)f(0, 1)f(0, 0) = 0; \quad (5)$ |
|---|---|

The Mathematical Analysis of Logic by George Boole

The Mathematical Analysis of Logic by George Boole

<https://www.gutenberg.org/ebooks/36884>にリンクしたい

現在使われている論理記号やAND、ORなどの用語は、後世に**数理論理学**として発展して

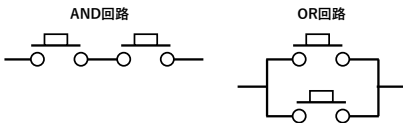


いく中で作られたもので、最初のとっかかりのアイデアを考えた人として、ブール氏の名前を冠することにしたらしい。

論理回路の誕生

2

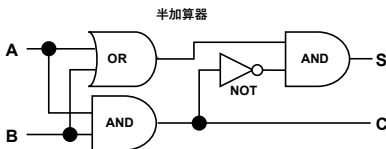
ブール氏が論文を書いたから 100 年ほどして、電信に使われるリレーを使った**論理回路 (Logic Circuit)** が誕生した。**クロード・シャノン氏**が論理回路についての論文を書いたのが 1937 年で、ほぼ同時期に最初のデジタルコンピュータも誕生している。このあたり順番がはっきりしなくてモヤッとするが、たいていの発明・発見は、複数の人が同時期に同じ答えにたどりつくもののようなのだ。



論理回路 (Logic Circuit)

ここで不思議なのは、スイッチを直列に並べて AND 回路、並列に並べて OR 回路が作れるというのは、ブール論理の延長として理解できる。しかし、その先、つまり**論理回路を使って計算する**という発想はどこからでてきたのだろうか？

2 進数 1 桁の計算を行う**半加算器 (Half Adder)** は、AND 回路 2 つと OR 回路 1 つ、NOT 回路 1 つの組み合わせで作れる。



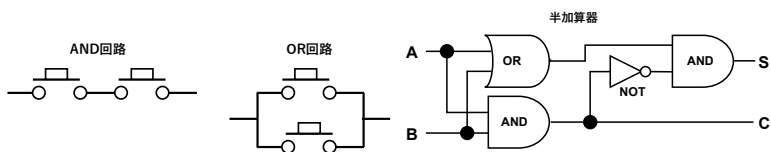
半加算器 (Half Adder)

しかし、これがブール論理の延長で、普通に出てくるとはちょっと考えにくい気がする。ブール論理は「AND (かつ)」「OR (または)」などで真偽を求めるものであって、四則演算するものではないはずだ。

残念ながら、加算器の発明者の資料は見つけられなかった。おそらく、何かを作ろうとし

て論理回路の組み合わせを考えているうちに、「 $0+0=00$ 」「 $0+1=01$ 」「 $1+0=01$ 」「 $1+1=10$ 」という結果を出せば足し算する回路が作れるという答えにパツとたどりついたのだろう。

世の中には頭のいい人はたくさんいるので、このぐらいは複数の人が同時に思いつくものなのかもしれない。



論理回路と半加算器



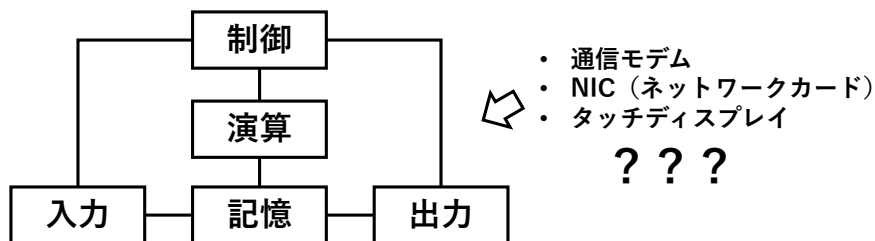
通信機能は5大装置のどこに入る？



2

コンピュータサイエンスの初歩として必ず登場するのが、**ノイマン型コンピュータ**（von Neumann architecture）もしくは**コンピュータの5大装置**だ。コンピュータは「制御」「演算」「記憶」「入力」「出力」という5つの装置で構成されるという説で、**フォン・ノイマン氏**がその論文を書いたので、ノイマン型と呼ぶ。ノイマン氏は悪魔のように頭がよい人だったが、5大装置をノイマン氏自身がゼロから考えたわけではなく、**EDVAC**という開発中のコンピュータを見学して、「つまり、こういうことだろ？」と論文をまとめたそうだ。

- **制御：記憶装置からプログラムを読み込み、各装置に指示を出す**
 - CPU
- **演算：四則演算などの計算を行う**
 - CPU
- **記憶：プログラムやデータを記憶する**
 - メモリ
 - HDD
 - SSD
- **入力：外部から情報を取り込む**
 - キーボード
 - マウス
- **出力：外部に情報を書き出す**
 - ディスプレイ
 - スピーカー



von Neumann architecture

今のパソコンやスマートフォンでも、5大装置という構成は変わらないので、今の教科書にもそのまま出てくる。ただ、5大装置の図に、実際のCPUやメモリ、キーボードなどの部品を当てはめていくと、ふと困ったことに気付く。通信モデムやNIC、タッチディスプレイなど、5つのカテゴリにうまく当てはまらない機器が結構出てくるのだ。

そのときはしかたなく、入力と出力を図の右側に配置して、「入出力兼用の機器もある」と説明したのだが、もっとうまいやり方があったのではと今でもときどき思う。

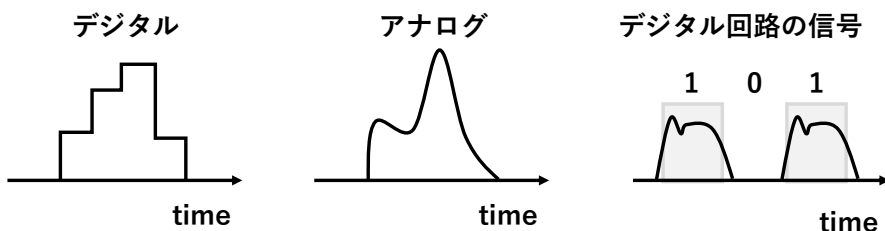
ノイマン氏に聞いてみたいところだが、悪魔のように頭のいい人だったらしいので、「馬鹿め、そのぐらい自分で考えろ」といわれそうだ。



デジタルとアナログ

よくコンピュータは**デジタル**だという。デジタルは「連続した量」を意味する**アナログ**の対義語で、連続していないという意味から日本語では「離散的な値」ともいう。電気や光、音といった自然現象はすべてアナログだ。

ここでよく考えてみると、コンピュータの電子回路も電気が流れているからアナログではないかという答えにたどり着く。実際のところ、コンピュータの電気信号も連続的に変化していて微妙なブレがあり、この範囲にだいたいおさまったら「1」、この範囲だったら「0」という感じに決めて、デジタルと見なしているようだ。



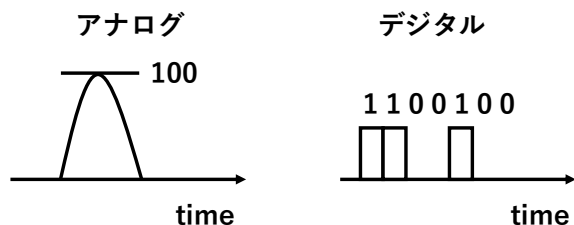
digital analog

自然界にはアナログしか存在せず、コンピュータもアナログ。ということは、純粋なデジタルは、人間の頭の中にしか存在しないということになる。加えていうとプログラム（ソフトウェア）も、人間のデジタル的な考え方を反映したものであるので、デジタルといえる。

デジタルとアナログについて考えていくと面白いことがいろいろあって、昔のビデオテープや音楽テープはすべてアナログで、デジタル化したDVDやCDが登場したのは結構あのことだ。これはなぜかという、デジタルのほうがデータ量が増えるからだ。

例えばアナログで「100」を表す場合、単位や媒体は何かはわからないが、100の強さの信号があればいい。デジタルで100を表す場合、2進数だと「01100100」となるので7ビット必要になる。仮に1秒間に1つしか信号を送れないとすると、アナログでは1秒で済むと

ころが、デジタルでは7秒必要という計算になる。つまり、デジタルのほうがデータ量が増えるわけで、これを解決するには電子回路をはるかに高速にするしかない。その発展に時間がかかったわけだ。



digital analog

データ量が増える以外のデジタルの弱点として、連続して変化するアナログを、段階的なデジタルに変換する過程で、変化途中の細かな情報が切り捨てられるという点がある。アナログの信号（レコードとかフィルムとか）だったら、昔は読み取れなかった情報が未来の技術で読み取れる可能性があるが、デジタルの信号（DVDとかBDとか）はいつまで経っても記録時点のままなので、切り捨てられた情報は未来永劫復元できない。最近よく聞くデジタルリマスターも、大元のフィルムから起こすのであって、DVDを原本とするのではない。

などと書きつつも、最近のAI技術では「存在しないものをそれらしく復元する」こともできてしまうので、将来的には何とかなってしまうかもしれないなとも思う。未来を見通すのは難しい。

フォントテスト

- フォント1～Noto Sans JP (Web)
- フォント2～UDPゴシック(Web)
- フォント3～UD 明朝 (Web)
- フォント4～ZEN丸ゴシック (Web)
- フォント5～ZEN オールド明朝 (Web)
- フォント6～Klee One (Web)
- フォント7～游ゴシック Light (Win 付属)
- フォント8～游明朝 Light (Win 付属)