

■目次 (ほげ が今回の範囲)

- 第1部 0と1だけでどうやって計算するの? ⇒ ★1 p進法による数の表現
- 第2部 コンピュータの気持ち ★2 論理回路と加算器
- 第3部 情報をどのように表現するか ★3 負の数の表現と減算の仕組み
- 第4部 コンピュータシステム

前回は説明したように、コンピュータは論理回路できている。論理回路は入力も出力も2種類の値(例えば0と1)しかとらないが、2進数の加算だけでなく、様々な計算・情報処理を行うことができる。加算の過程を分解して考えてみた経験からもわかるように、複雑なことをこなす論理回路はとても複雑なものとなる。しかし、複雑な論理回路も、どんどん分解していくと単純な「部品」の集まりであることがわかる。その「部品」がどのようなものかを学び、それを使って加算の回路を作ってみよう。

★2 論理回路と加算器 (承前) (☆1)

★2.2 論理演算と論理ゲート

論理回路を分解していくと、以下の表に示すような、「1つまたは2つの0/1の値(表中ではA, Bで表されている)を入力すると1つの0/1(Xで表されている)を出力する仕掛け」の組み合わせできているとみなせることがわかる(☆2)。この表の中の真理値表で表されているような0/1の計算を論理演算といい、論理演算を行う回路を論理ゲート(または論理素子)という。

☆1) 「承前」とは、前の文章から続いていることを意味します。ここでは「前回の★2の続きだよ」ってこと。

☆2) そのことは数学を使ってきちんと示せるのですが、この授業ではやりません。(☆5)も参照してください。

	否定 (NOT)	論理積 (AND)	論理和 (OR)	排他的論理和 (XOR)																																																			
真理値表	<table border="1"> <tr><td>A</td><td>X</td></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table>	A	X	0	1	1	0	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	X	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <tr><td>A</td><td>B</td><td>X</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table>	A	B	X	0	0	0	0	1	1	1	0	1	1	1	0
A	X																																																						
0	1																																																						
1	0																																																						
A	B	X																																																					
0	0	0																																																					
0	1	0																																																					
1	0	0																																																					
1	1	1																																																					
A	B	X																																																					
0	0	0																																																					
0	1	1																																																					
1	0	1																																																					
1	1	1																																																					
A	B	X																																																					
0	0	0																																																					
0	1	1																																																					
1	0	1																																																					
1	1	0																																																					
ゲートの図記号																																																							
論理式	$X = \neg A$ $X = \bar{A}$	$X = A \wedge B$ $X = A \cdot B$	$X = A \vee B$ $X = A + B$	$X = A \veebar B$ $X = A \oplus B$																																																			
ベン図																																																							

注1: 論理式については、この授業では扱いませんが「計算機システム I」, 「離散数学」等できっと出てきます。ちなみに、上段が数理論理学など数学や情報科学の方でよく使われる表記, 下段が情報工学の方でよく使われる表記。
 注2: ベン図による表現についてもこの授業では触れないが、円の内部はAやBが1であることを表し、網掛けはXが1となる所を表す。

表に示した論理演算の説明を以下に記す (☆3)

否定 (NOT) 入力 $0/1$ を反転したものを出力する

論理積 (AND) 「 $A = 1$ かつ $B = 1$ 」のとき $X = 1$, さもなくば $X = 0$

(X は, A も B も 1 のときだけ 1 になる)

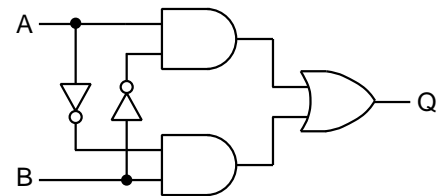
論理和 (OR) 「 $A = 1$ または $B = 1$ 」のとき $X = 1$, さもなくば $X = 0$

(X は, A と B のどちらか一方でも 1 なら 1 になる)

排他的論理和 (XOR) (☆4) X は, A と B のどちらか一方だけが 1 のときだけ 1 になる

ここではこれ以上は紹介しないが, 論理演算/論理ゲートにはこれら 4 種類の他にも様々なものがある. ただし, たくさんある論理演算も, いくつかの基本的な論理演算の組み合わせで作ることができる (☆5).

右の図は, 上記の論理演算に対応した論理ゲートを組み合わせて作った論理回路の例である. この例では, A, B それぞれに $0/1$ が入力されると, その値の組み合わせに応じて出力 Q が $0/1$ の値をとる. 線分の交わる箇所に●が描かれている場合と描かれていない場合があるが, 前者は線がつながっており, 後者はつながっていないことを意味する.



☆3) 真理値表を丸暗記するのはつらすぎるのでやめましょう. それぞれの演算の意味を覚えておけば, すぐに真理値表作れますから.

☆4) 英語で exclusive or と言うことから XOR と表記される.

☆5) 「計算機システム I」では, NOT, AND, OR の 3 種類だけで任意の論理回路を作れることを学ぶでしょう.

Q1. [論理回路から真理値表を作ろう] 上図の論理回路の真理値表を求めなさい (☆6).

☆6) この論理回路は実は...

★ 2.3 論理ゲートの実現

ここまでの話で「たくさんの論理ゲートを組み合わせたらコンピュータ作れそうや」ということがイメージできたとしても, 「ほな論理ゲートは実際にはどうやって作んねん?」という疑問が湧く. 近代および現代のコンピュータでは, 電気信号を利用して, 電流が流れている/いない, または, 電圧が高い/低い, ということで $0/1$ を表現し, それを切り替えるスイッチの働きをする電気回路・電子回路を使って論理ゲートを実現している. その仕組みについても学ぶべき興味深い事柄がたくさんあるのだが, この授業では割愛する. 以下に, いくつかのキーワードをごく簡単な説明とともに上げておく.

- リレー: 電気信号のオンオフを制御する機器. 初期のコンピュータで使われた.
- 真空管: 電球のような管の中に電極が並んでいる. 初期のコンピュータで使われた.
- トランジスタ: 半導体を使ってリレーや真空管と同様の機能を実現したもの.
- 集積回路 (IC, Integrated Circuit): トランジスタなどの半導体素子を小さくして, たくさん集めて一つにしたもの. 例えば, 現代のコンピュータの CPU (☆7) は, 数十億個のトランジスタを集積してできている.

☆7) コンピュータの中核となっている部品. 詳しくは後日解説します.

中野先生が作成された 2013 年度「情報処理の基礎」の第 9 回講義資料 (☆8)

<http://www602.math.ryukoku.ac.jp/~nakano/IPS1/printed/part09.pdf>

に, リレーによる論理ゲートの実現例が説明されています. また, 実用的ではありませんが, ドミノ倒して 2 進数の加算を実現した人たちもいます. 動画もあり.

<http://think-maths.co.uk/downloads/domino-computer-worksheets>

☆8) この資料の PDF 版では, URL をクリックするとブラウザでその先へジャンプするようにしてあります.

★ 2.4 加算を行う論理回路

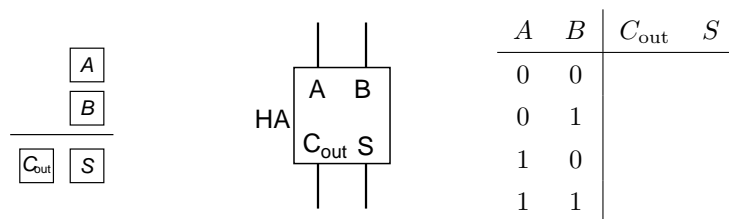
論理ゲートを組み合わせて、2進数で表された2つの数の加算を行う論理回路を作ろう。いきなり全体を作るのは大変なので、次のステップで考える。

1. AND や OR などの組み合わせで「半加算器」という論理回路を作る
2. 半加算器を組み合わせて、1ビットの加算をする論理回路「全加算器」(☆9)を作る
3. 全加算器を組み合わせて、複数ビットの加算をする論理回路を作る

★ 2.4.1 半加算器

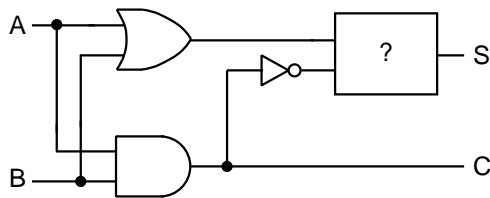
半加算器 (☆10) は、入力として2つの1ビットの数を受け取り、それらの和と上のビットへの桁上げの値を出力する論理回路である。下のビットからの桁上げを無視しているので、1ビットの加算の仕組みとしては半人前である。後述の全加算器の部品として使える。

入力を A, B , 和の出力を S , 桁上げ出力を C_{out} と表すと、半加算器の真理値表は下表のようになる。



Q2. [真理値表から論理回路を作ろう (1)] ★ 2.2 に登場した論理ゲートを組み合わせて、半加算器を作りなさい (つまり、上の真理値表を実現する論理回路の回路図を描きなさい)。

Q3. [真理値表から論理回路を作ろう (2)] 下図は、半加算器の回路図である (☆11)。[?]の部分には、講義資料1ページ目に示した論理ゲートのうちのいずれか一つが当てはまる (☆12)。当てはまるのは何か。



☆9) これは、第1回の「○と●で謎の計算をしよう」で出てきた真理値表、第2回の★2.1のはじめに出てきた回路図と真理値表に対応している。

☆10) 英語で Half Adder というので、HA と略記することがある。

☆11) 授業時に Q2 の回答として説明したものと異なる論理ゲートの組み合わせで作ったもの。

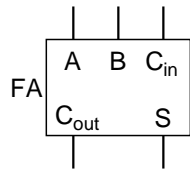
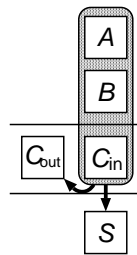
☆12) 途中の論理ゲートの出力の真理値表を書いてみるとわかるが、「?」への2つの入力は、両方同時に0になることがない。したがって、入力が2つとも0の時の出力が0/1のどちらであっても、この回路の動作には支障ない。そのため、講義資料に示した論理ゲートに限らないとしたら、「XOR と逆の 0/1 を出力する論理ゲート」も当てはまることになる。

★ 2.4.2 全加算器

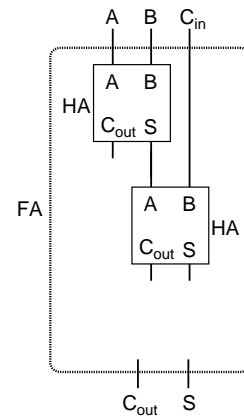
全加算器 (☆13) は、入力として2つの1ビットの数と下のビットからの桁上げの値を受け取り、和と上のビットへの桁上げの値を出力する論理回路である。2つの2進数の加算において、1ビット分の加算を実行することができる。

2つの1ビットの値を A, B 、下位桁からの桁上げ入力を C_{in} とし、和の出力を S 、上位桁への桁上げ出力を C_{out} と表すと、全加算器の真理値表は下表のようになる。

全加算器は、半加算器2つと論理ゲート1つを用いて作ることができる(下図右端、ただしこの図は未完成)。他にも様々な作り方ができるが、この授業では省略する。



A	B	C_{in}	C_{out}	S
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

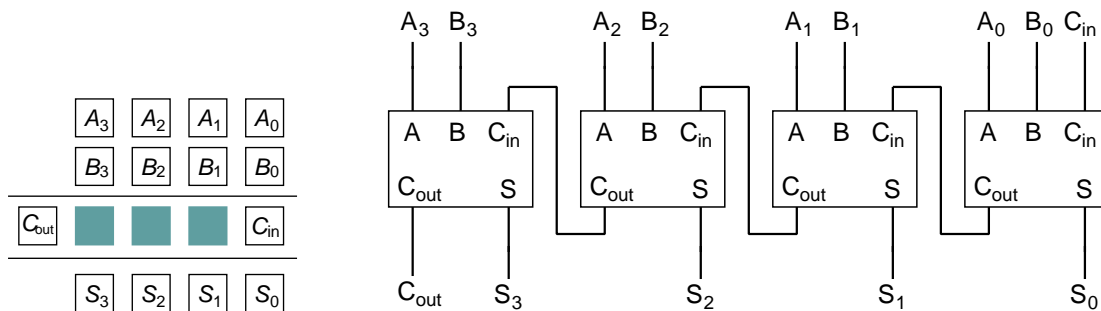


Q4. 上図右端の全加算器の回路図を完成させなさい。

★ 2.4.3 複数ビットの加算を行う論理回路

複数の全加算器をつなぎ合わせると、複数ビットの加算を行う論理回路を作ることができる。下図右は、4ビットの2進数 $A_3A_2A_1A_0$ と $B_3B_2B_1B_0$ の和 $S_3S_2S_1S_0$ を計算する回路である。入力 C_{in} は通常は0とするが、長いビット長の数を4ビット毎に分割し、4ビットの加算を繰り返して加算を行う際に、下の桁からの桁上げを受け取るために用いられる。出力 C_{out} も同様に、上の桁への桁上げを渡すために用いられる (☆14)。

このように、 n 個の全加算器を接続すれば任意の n ビットの加算を行う論理回路ができる (☆15)。以上のことから、NOT や AND, OR といった論理演算を行う論理ゲートを部品として、複数ビットの加算を行う論理回路を作れることがわかる。



☆13) 英語で Full Adder というので、FA と略記することがある。

☆14) 実は他にも使い道があるが、それはいづれまた。

☆15) この回路では、 n が大きいと信号がたくさん論理ゲートを通せなければならないため、加算にかかる時間も長くなってしまう。実際のコンピュータで用いられる加算の回路では、そうならないような工夫をしている。