

# ICUHS 数学ツアー 2019

鈴木寛 (Hiroshi Suzuki)

国際基督教大学 (International Christian University)

August 29-30, 2019

# Day 1

## NAND ゲートの加算器

# NAND ゲートの加算器

[\*]

クラスで考えてみたいこと

最初に、1.3.1 にあることから確認していききたいと思います。

- ①  $p, q, r$  がすべて 0 か 1 とすると、三組  $(p, q, r)$  は

$(0, 0, 0), (0, 0, 1), (0, 1, 0), (0, 1, 1), (1, 0, 0), (1, 0, 1), (1, 1, 0), (1, 1, 1)$

全部で 8 通りあります。入力が、8 種類の  $(p, q, r)$  のときに、出力が、それぞれ、たとえば、

$(0, 1, 1, 0, 1, 0, 0, 1), (0, 0, 0, 1, 0, 1, 1, 1)$

となるような回路は作れますか。( $(0, 0, 0)$  のとき 0,  $(0, 0, 1)$  のとき 1,  $(0, 1, 0)$  のとき、1 などです。) 加算器の問題とは、どのように関係しているのでしょうか。

- ② 1 で出力の 8 個組がなんであっても、それを出力する回路は作れるでしょうか。
- ③ 2 の回路は、NAND だけでつくれるでしょうか。
- ④ 基本的なものをまず作って、それを組み合わせてできないだろうか。

# 復習1: 2進数の演算

答えだけ書いておきます。

## ① 10進数の 0-15 を 2進数で表すと

- $0000_2, 0001_2, 0010_2, 0011_2, 0100_2, 0101_2, 0110_2, 0111_2, 1000_2, 1001_2, 1010_2, 1011_2, 1100_2, 1101_2, 1110_2, 1111_2$ .
- これらを 16 進記法で、 $0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F$  と書く場合もあります。

## ② 10進数を 2進数で表す方法

- 10 進数を  $n$  としたとき、2 で割ってあまりを計算していく方法と、 $2^m \leq n < 2^{m+1}$  を満たす、 $m$  を見つけて順に引いていく方法があります。
- $n = a_m 2^m + a_{m-1} 2^{m-1} + \dots + a_1 2^1 + a_0 2^0$  ( $a_m, a_{m-1}, \dots, a_1, a_0$  は 0 または 1) ですから、上の方法は、 $a_0, a_1, \dots$  の順番に見つけていく方法と、 $a_m, a_{m-1}, \dots$  と見つけていく方法と考えることもできます。
- $n = a_m a_{m-1} \dots a_1 a_0_2$  または、 $n = (a_m a_{m-1} \dots a_1 a_0)_2$  などと書きます。

## ③ 2進数を 10進数で表す方法

上の 2 の説明を逆にたどれば、よいですね。

# 復習1: 2進数の演算 (つづき)

## ④ 2進数の足し算 例: $101_2 + 110_2$

- 10進の5と6ですから、和は11、すなわち、 $1011_2$  となります。
- 10進の筆算のように繰り上がりにも注意すれば、以下のようになります。

$$\begin{array}{r}
 \phantom{+} \phantom{1} \phantom{0} \phantom{1} \\
 \phantom{+} \phantom{1} \phantom{0} \phantom{1} \phantom{0} \\
 + \phantom{1} \phantom{0} \phantom{1} \phantom{0} \\
 \hline
 1 \phantom{0} \phantom{1} \phantom{1} \phantom{0}
 \end{array}$$

## ⑤ 2進数の掛け算 例: $101_2 \times 110_2$

$$\begin{array}{r}
 \phantom{\times} \phantom{1} \phantom{0} \phantom{1} \\
 \phantom{\times} \phantom{1} \phantom{0} \phantom{1} \phantom{0} \\
 \times \phantom{1} \phantom{0} \phantom{1} \phantom{0} \\
 \hline
 \phantom{\times} \phantom{1} \phantom{0} \phantom{1} \phantom{0} \\
 \phantom{\times} 1 \phantom{0} \phantom{1} \phantom{0} \\
 \hline
 1 \phantom{0} \phantom{1} \phantom{1} \phantom{1} \phantom{0}
 \end{array}$$

## ⑥ 2進数で1より小さい小数を表す方法

- 基本的には、10進数の場合と同じです。 $0.5 = 2^{-1} = .1_2$  などとなります。 $n$  を1より小さい小数とすると、 $n = b_1 2^{-1} + b_2 2^{-2} + b_3 2^{-3} + \dots$  ( $b_1, b_2, \dots$  は、0 または 1) となります。ただし、10進数では、小数点以下有限であっても、無限小数になる場合もあります。今回は、使いませんが、考えてみるとよいと思います。

# 授業 I: 論理回路設計 (1)

2 進演算で、加算器を作を考えてみましょう。単純なものから考えます。すなわち、二進一桁を一ビット (bit) と呼びますが、1 ビット + 1 ビットの加算器です。A と B であらわしましょう。

- ①  $0 + 0 = 0$ ,  $0 + 1 = 1$ ,  $1 + 0 = 1$ ,  $1 + 1 = 10_2$  です。最後だけ二桁になっています。添字の 2 は、2 進であることを示したものです。繰り上がりを Carry と言います。ですから、みな二桁だと考えて、
- ②  $0 + 0 = 00_2$ ,  $0 + 1 = 01_2$ ,  $1 + 0 = 01_2$ ,  $1 + 1 = 10_2$  と出力を二桁にしてみましょう。
- ③ 一桁目 ( $2^0$  の位) を  $A \oplus B$  繰り上がりを  $C$  であらわしましょう。すると次のようになっていることがわかります。

A	B	$A \oplus B$	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- ④ 入力が A と B の二つ。出力が、 $A \oplus B$  と C の二つということになります。出力は一つ一つ作っていくことにしましょう。

# 入力が $A$ と $B$ の二つ、出力は、 $A \oplus B$ と $C$

- ⑤ これを、いくつかのゲート（ポート）を組み合わせで作りたいので、それぞれの、ゲートが入力に対して、どのような出力をしているか見てみましょう。

$A$	$B$	$A \oplus B$	$C$	$A \text{ NAND } B$	$A \text{ AND } B$	$A \text{ OR } B$	$\text{NOT } A$	$A \text{ XOR } B$	$A \text{ NOR } B$
0	0	0	0	1	0	0	1	0	1
0	1	1	0	1	0	1	1	1	0
1	0	1	0	1	0	1	0	1	0
1	1	0	1	0	1	1	0	0	0

- ⑥ どんなことがわかりますか。

- $A \oplus B \equiv A \text{ XOR } B$
- $C \equiv A \text{ AND } B$

- ⑦ NAND だけを使うということであれば、これですでに加算器はできているようです。確認してみてください。

"http:

//www.neuroproductions.be/logic-lab/index.php?id=104988":

"Simple Half Adder" へのリンク

# 同値な表示

- ⑧ もう一つ確認しておきたいことがあります。

$A$	$B$	NOT	$(A \text{ AND } B)$	NOT( $A$ )	OR	NOT( $B$ )	$A \text{ NAND } B$
0	0	1	0	1	1	1	1
0	1	1	0	1	1	0	1
1	0	1	0	0	1	1	1
1	1	0	1	0	0	0	0

- ⑨ 入力は何であっても、出力が同じであるとき、同値といい、 $\equiv$  の記号を使って、次のように書きます。
- $\text{NOT}(A \text{ AND } B) \equiv A \text{ NAND } B$
  - $\text{NOT}(A) \text{ AND } \text{NOT}(B) \equiv A \text{ NAND } B$
  - $\text{NOT}(A \text{ AND } B) \equiv \text{NOT}(A) \text{ AND } \text{NOT}(B)$
  - 公式のようなもので、最後のものは、ド・モルガンの法則とも言われ、高校で習った人もいると思います。



# 論理記号による表示

- ⑩ 複雑になると、簡単な記号を使ったほうがわかりやすいこともありますので、一般的に使われる（論理）記号を紹介しておきます。
- NOT( $A$ ):  $\bar{A}$ ,  $\neg A$ ,  $\sim A$  日本の高校では1つ目を使っているので、ここでもそれを使うことにします。
  - $A$  AND  $B$ :  $A \wedge B$
  - $A$  OR  $B$ :  $A \vee B$
  - $A$  XOR  $B$ :  $A \oplus B$ ,  $A \veebar B$
  - $A$  NAND  $B$ : 一般的な記号はありません。  $A \uparrow B$
  - $A$  NOR  $B$ : 一般的な記号はありません。  $A \downarrow B$
- ⑪ 記号をつかうと、次のように書くことができます。
- ⑫  $A \uparrow B \equiv \overline{A \wedge B} \equiv \overline{A} \vee \overline{B}$ .
- ⑬ 二桁以上の場合も一桁ずつ計算しますが、桁上り（Carry）の部分も計算しないといけませんから、入力が  $A, B, C$  で出力が  $X, Y$  のようになっています。  $Y$  を桁上り分としましょう。

# 全加算器 Full Adder

入力は  $A, B, C$  の三つ、出力は、 $A \oplus B \oplus C$  と繰り上がり

- 12  $A = B = C = 1$  であっても、桁上りをふくめて二桁で収まることを確認しておきましょう。するとどうなるでしょうか。

$A$	$B$	$C$	$X$	$Y$	$Z$
0	0	0	0	0	
0	0	1	1	0	
0	1	0	1	0	
0	1	1	0	1	
1	0	0	1	0	
1	0	1	0	1	
1	1	0	0	1	
1	1	1	1	1	

- 13 これを実現する回路を全加算器 (Full Adder) といいます。さきほどの、入力に、桁上りを考えないものを、半加算器 (Half Adder) といいます。
- 14 いろいろなゲートを用いて、全加算器を作り、あとから、上のような公式を使って、NAND 回路にしていく方法も一つです。そのことは、次の時間で説明します。

# NAND だけで表す方法

- 15 上の公式では、NAND を他のもので書き換えてみましたが、逆に、他のゲートを、NAND で置き換えることはできないでしょうか。下の右辺を NAND だけでかけますか。NOT, AND, OR, XOR, NOR は使わないということです。

- $\bar{A} \equiv$
- $A \wedge B \equiv$
- $A \vee B \equiv$
- $A \oplus B \equiv$

- 16 問題をまとめておきます。

- 1 上の表の  $X$  や  $Y$  または、 $Z$  に 0, 1 がどのように並んでいても、いろいろな論理記号を使って、値がちょうどそのようになるものを作れるか。
- 2 他の記号をつかわず、NAND だけで書くことができるか。
- 3 二桁以上になったときに、どのように組み合わせていけばよいか。

## 授業 II: 論理回路設計 (2)

最後に述べた問題の 1 と 2 について考えてみたいと思います。  
具体的には、次の二つの項目について話します。

- ① 表の  $Z$  に 0, 1 がどのように並んでいても、入力の  $A, B, C$  と NOT, AND, OR だけを組み合わせ、 $Z$  が出力になるようにすることができる。  
(Conjunctive Normal Form, CNF)
- ② NOT, AND, OR いずれも、NAND で書くことができる。(Functional Completeness, FC)

注: XOR については、述べていませんが、 $A \oplus B$  の値を、 $Z$  に書いておけば、そのように、NOT, AND, OR だけで書けるのですから、心配いらないことがわかります。

# NAND の完全性

- $\bar{A} \equiv A \uparrow A$
- $A \wedge B \equiv \overline{A \uparrow B}$  : NOT を使っていますが、上を使えば、NOT を使わない形に変形できます。
- $A \vee B \equiv \bar{A} \uparrow \bar{B}$  : 上と同様

Hint:  $Z$  がどんな式であっても、 $\overline{\bar{Z}} = Z$ ,  $A \uparrow B \equiv \overline{A \wedge B} \equiv \bar{A} \vee \bar{B}$

# Conjunctive Normal Form, CNF

$X$	$Y$	$Z$	$X \wedge Y \wedge Z = E_8$	$X \vee Y \vee Z$	$S$	$C$	$E_2$	$E_3$	$E_5$
0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	1	0	1	0	0
0	1	0	0	1	1	0	0	1	0
0	1	1	0	1	0	1	0	0	0
1	0	0	0	1	1	0	0	0	1
1	0	1	0	1	0	1	0	0	0
1	1	0	0	1	0	1	0	0	0
1	1	1	1	1	1	1	0	0	0

- $S \equiv E_2 \vee E_3 \vee E_5 \vee E_8$
- $E_8 \equiv X \wedge Y \wedge Z$
- $E_2 \equiv \bar{X} \wedge \bar{Y} \wedge Z$
- $E_3 \equiv$
- $E_5 \equiv$

注：日本語では、連言標準形というらしいです。他方、Disjunctive Normal Form (DNF) は、選言標準形。

# 回路の設計と課題

- 原理的には、全加算器の回路ができることを示しました。
- かなり複雑ですから、全加算器を NAND で簡単に書く方法を考えてください。
- できれば、3桁 + 3桁の計算までできる加算器を作ってください。
- 設計した加算器と工夫したこと、考え方などを、説明してもらおうと思います。
- NAND ではなく、他の一種類のゲートだけで、書くことはできないのでしょうか。考えてみてください。

## 練習問題

- ①  $A \Rightarrow B$  ( $A \text{ IM } B$  ともかく) ゲートを下の表で定義する。空欄を埋めよ。

$A$	$B$	$A \Rightarrow B$	$\bar{A} \vee B$	$\overline{A \wedge \bar{B}}$	$\bar{B} \Rightarrow \bar{A}$
0	0	1			
0	1	1			
1	0	0			
1	1	1			

このことから、次がわかる。

$$A \Rightarrow B \equiv \bar{A} \vee B \equiv \overline{A \wedge \bar{B}} \equiv \bar{B} \Rightarrow \bar{A}.$$

$\bar{B} \Rightarrow \bar{A}$  は対偶 (contrapositive) と呼ばれる。

- ② 次の二つの式で表される論理回路の出力が同じではない (同値でない) ことを示せ。 $(A \wedge B) \vee C$   $A \wedge (B \vee C)$ 。(回路を作って示しても、表を書いて示しても、他の方法でも構いません。)
- ③  $A \Rightarrow (B \vee C) \equiv A \wedge (\bar{B}) \Rightarrow C$  すなわち、どちらの論理回路も  $(A, B, C)$  のすべての値について同じ出力となること (同値であること) を、三つの方法で示せ。
- ① 表を完成することで確認せよ。
  - ② 実際に、論理回路を作って確認せよ。
  - ③ 練習問題 1 などの公式を使った変形により示せ。



# 練習問題（つづき）

- ④ 論理記号  $\uparrow$  NAND だけを用い（NAND 回路で）次を表わせ。NOT も使ってはいけません。
- ①  $A \wedge B$
  - ②  $A \vee B$
  - ③  $A \oplus B$
  - ④  $A \Rightarrow B$
  - ⑤  $A \downarrow B$
- ⑤  $(A \wedge \bar{B}) \Rightarrow C$  を以下の指示にしたがって変形せよ。
- ①  $\vee$  と NOT（否定）だけを使い、 $\wedge$ 、 $\Rightarrow$  などを用いない。
  - ②  $\uparrow$  だけを使い、 $\wedge$ 、 $\vee$ 、 $\Rightarrow$  NOT（否定）などを用いない。
- ⑥  $(A, B, C)$  の入力が  $(0, 0, 0)$ ,  $(0, 1, 0)$ ,  $(1, 0, 1)$  のときだけ 1 でそれ以外、0 を出力する回路を作成せよ。
- ① AND, OR, NOT ゲートだけを用いよ。
  - ② NAND ゲートだけを用いよ。

# スケジュール: 8月29日(木)

- 9:00 ICU (大学) N 館ラウンジ集合ののち、教室 N-232 に移動
- 9:10～ 先生の紹介
- 9:15～ 講義Ⅰ 論理回路設計 (1)
- 10:15～ 休憩 (N307 に移動)
- 10:30～ コンピュータの説明と個人での確認 (N307)
- 11:00 ～ 講義Ⅱ 論理回路設計 (2) および 質疑応答 (N307)
- 12:00～ 昼食休憩 (大学の学食に移動)
- 13:00 集合 (N307)
- 13:10～ 研究室見学
- 14:10～ グループワーク (4～5 名の 3 グループに分かれて) (N307)
- 15:30～ 発表について・予告 (N307)
- 16:00 終了

# スケジュール: 8月30日(金)

- 9:00 ICU (大学) N 館教室 N-232 に直接に集合
- 9:05～ 講義 III いくつかの点を通る多項式関数 (1) (N232)
- 10:05～ 質疑応答 (N232)
- 10:15～ 休憩
- 10:30～ 講義 IV いくつかの点を通る多項式関数 (2) (N232)
- 11:30～ 質疑応答・発表について (N232)
- 12:00～ 昼食休憩 (大学の学食に移動)
- 13:00 集合 (N232)
- 13:10～ グループワーク (N232, N307)
- 13:40～ グループワークと平行して、研究室見学
- 15:00～ グループの発表、各グループ 15 分 (N232)
- 15:55～ 講評と補足 (N232)
- 16:15 終了