電気工学通論Ⅱ（２）

坂井　修一
東京大学大学院 情報理工学系研究科 電子情報学専攻
東京大学　工学部　電子情報工学科

- はじめに
- 論理演算

はじめに

- 本講義の目的
  - ディジタル回路とコンピュータアーキテクチャの基礎を学習する
- 時間・場所
  - 火曜日 8:30 – 10:15、工学部2号館213
- ホームページ（ダウンロード可能）
  - url: http://www.mtl.t.u-tokyo.ac.jp/~sakai/tsuron2/
- 教科書
  - 坂井修一『論理回路入門』（培風館）
  - 坂井修一 『コンピュータアーキテクチャ』（コロナ社）

講義の概要と予定（１／２）

１．ディジタル回路入門

ディジタルとアナログ、２進数、補数表現、四則演算

２． 論理演算

論理演算とは、３つの基本論理演算、NORとNAND、完備性、
ディジタル回路の表現、ブール代数、標準形

３．組み合わせ回路の構成法

真理値表と組合せ回路、組合せ回路の簡単化

４．組合せ回路の実例

加算器、補数、減算器、ＡＬＵ、デコーダ、セレクタなど

５．フリップフロップ

ＦＦ、ＳＲラッチ、Ｄラッチ、非同期と同期、SR−FF、Ｄ−FF、
マスタスレーブ形とエッジトリガ形, JK−FF, レジスタ

６．基本的な順序回路

７．一般的な順序回路の作り方

講義の概要と予定（２／２）

８．コンピュータアーキテクチャ入門

計算のサイクル、主記憶装置、メモリの構成と分類、レジスタファイル、命令、命令実行の仕組み、実行サイクル、算術論理演算命令、シーケンサ、条件分岐命令

９．命令セットアーキテクチャ

操作とオペランド、命令の表現形式、アセンブリ言語、命令セット、算術論理演算命令、データ移動命令、分岐命令、アドレシング、サブルーチン、RISCとCISC

10．パイプライン

11．キャッシュと仮想記憶

12．入出力と周辺装置

周辺装置、ディスプレイ、二次記憶装置、ハードウェアインタフェース、割り込みとポーリング、アービタ、DMA、例外処理

休講：　X月Y日　　試験：　Z月W日

成績：　出席（小テスト）＋試験　　４年生：追試・レポート無し

2. 論理演算

論理演算

- 論理関数を用いた演算のこと。
  - ・1出力論理関数とは、次のようなもの

    $F: (I_1, I_2, \ldots I_n) \rightarrow O$、 I_j, O : 0または1

    1出力論理関数は、全部で 2^{2^n} 通り存在し得る
- 組合せ回路
  - ・論理演算を実現する回路のこと

論理関数の表現

- 真理値表
- ブール代数
- MIL記号

論理関数の表現

■ 真理値表

X_1	X_2	Y_1	Y_2	Y_3	Y_4
0	0	0	1	0	1
0	1	0	0	1	1
1	0	0	0	0	0
1	1	0	0	0	0

○　正確：　入出力のパターンをすべて列挙

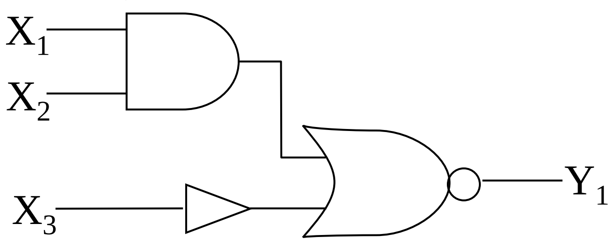
×　量！　意味？

■ ブール代数

・ $Y_1 = X_1 \cdot X_2 + X_3$

○　正確で簡潔。直感に訴えやすい

■ MIL記法



○　４つの記号で図示。理解しやすい
回路図に結びつけやすい

3つの基本関数 － AND, OR, NOT －

■ 真理値表による定義

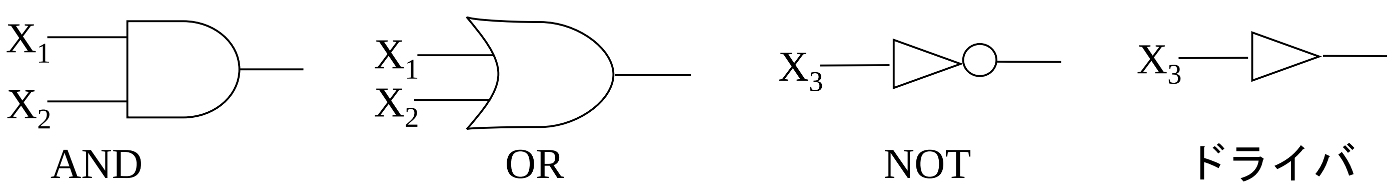
X_1	X_2	AND	OR
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	1

X_1	NOT
0	1
1	0

■ ブール代数： 論理変数と論理演算を記号化して表現する

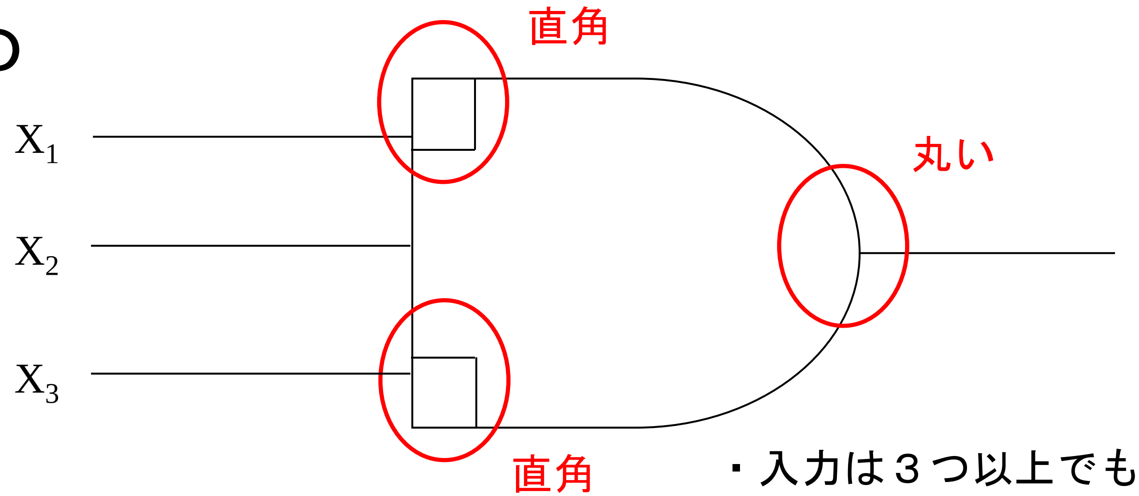
AND: $X_1 \cdot X_2$, OR: X_1+X_2 NOT: $\overline{X_1}$

■ MIL記法： ４つの図式と結線で表現する



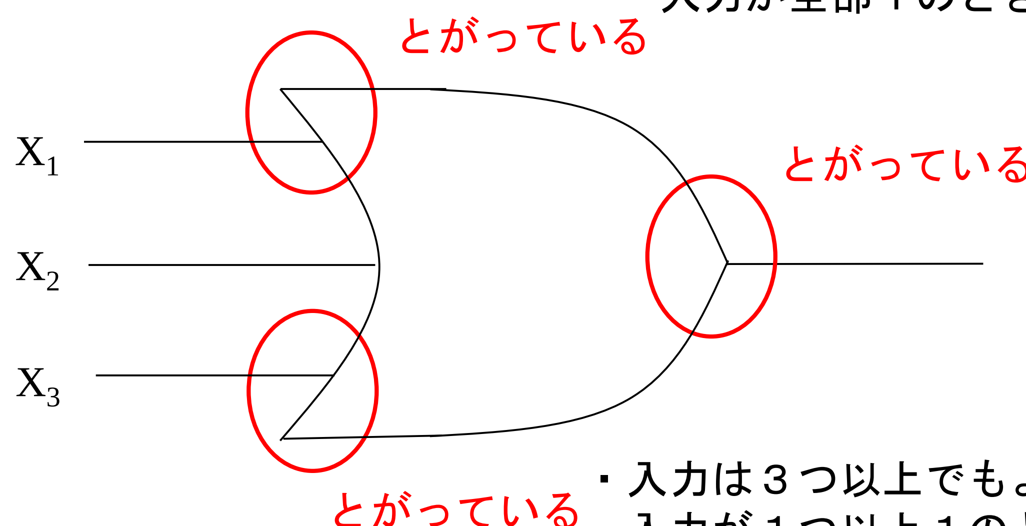
MIL記法の書き方　（米国military）

ＡＮＤ



・入力は３つ以上でもよい
・入力が全部１のときだけ１

OR



・入力は３つ以上でもよい
・入力が１つ以上１のとき１

NAND, NOR

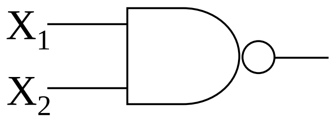
- 真理値表による定義

X_1	X_2	NAND	NOR
0	0	1	1
0	1	1	0
1	0	1	0
1	1	0	0

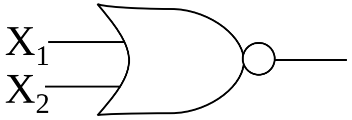
- ブール代数

NAND: $\overline{X_1 \cdot X_2}$, NOR: $\overline{X_1 + X_2}$

- MIL記法



NAND



NOR

XOR, EQUIV

真理値表による定義

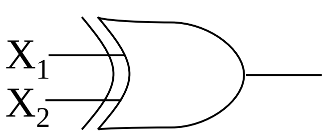
X_1	X_2	XOR	EQUIV
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

XORは入力の１つだけ１のとき１
EQUIVは$X_1 = X_2$のとき１

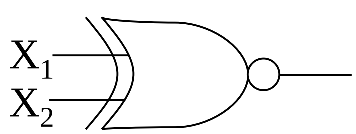
ブール代数

XOR: $X_1 \oplus X_2$,　EQUIV: $\overline{X_1 \oplus X_2}$

MIL記法



XOR



EQUIV

完備性 (Completeness)

- 「M入力N出力のすべての論理関数を、少数の基本論理関数の組合せで表現できる」とき、その基本論理関数の集合を、「完備集合」(Complete Set)という
- 完備集合の例
  - {AND, OR, NOT}
  - {AND, NOT}
  - {OR, NOT}
  - {NAND}
  - {NOR}

完備性の証明手順(1) {AND, OR, NOT}

- 帰納法で証明する

　１．１入力１出力の論理関数(全部で2種類)がすべて{AND, OR, NOT}の組合せで表現できることを証明する(自明)

　２．M入力１出力の関数を{AND, OR, NOT}の組合せで表現できたと仮定して、M＋１入力１出力の関数が表現できることをいう

I_1 I_2 …. I_M	I_{M+1}	Out
F_1	0	1
	0	0
	.	.
	0	1
F_2	1	0
	1	1
	.	.
	1	0

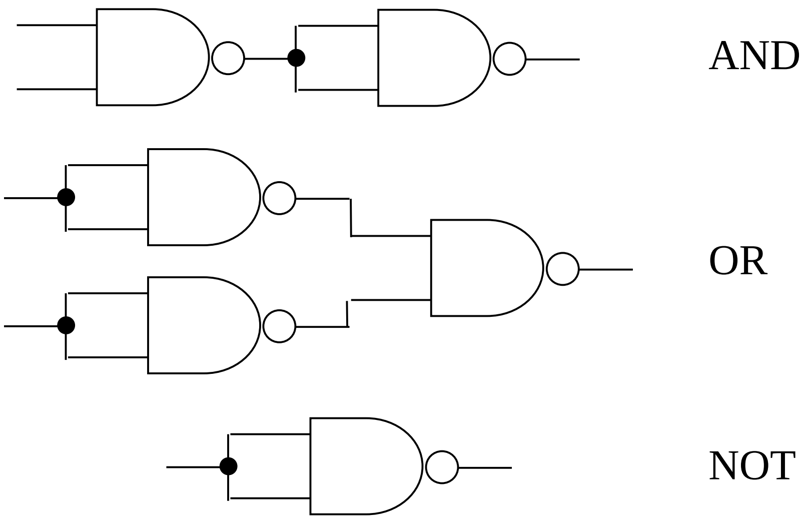
$$\mathbf{F} = \mathbf{F_1} \cdot \overline{\mathbf{I_{M+1}}} + \mathbf{F_2} \cdot \mathbf{I_{M+1}}$$

(1)F_1, F_2はM入力：帰納法の仮定より表現可能
(2)上の式は、F_1, F_2, I_{M+1}と２入力のAND, OR, NOTだけでできている

→　Ｆは{AND, OR, NOT}の組合せで表現可能

完備性の証明手順(2)　一般の場合

- （さきと同様に）帰納法で証明する
- 当該集合の要素で{AND, OR, NOT}が作れることを言う
    - 例：{NAND}



ブール代数

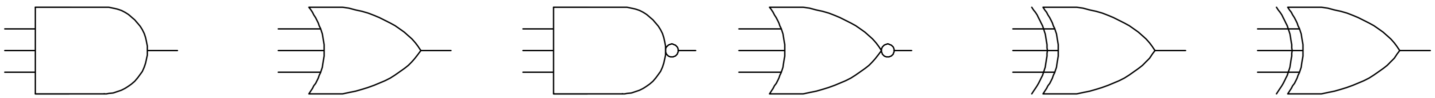
■ 演算規則

- $X + 0 = X$　　$X \cdot 0 = 0$　　$X + 1 = 1$　　$X \cdot 1 = X$
- $X + X = X$　　$X \cdot X = X$　　$X + \overline{X} = 1$　　$X \cdot \overline{X} = 0$
- $X + Y = Y + X$　　$X \cdot Y = Y \cdot X$　（交換法則）
- $X + (Y + Z) = (X + Y) + Z$　　$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$　（結合法則）
- $X \cdot (Y + Z) = X \cdot Y + X \cdot Z$　　$X + (Y \cdot Z) = (X + Y) \cdot (X + Z)$
  （分配法則）
- $\overline{X + Y} = \overline{X} \cdot \overline{Y}$　　$\overline{X \cdot Y} = \overline{X} + \overline{Y}$　（ド・モルガンの定理）

多入力素子

- **多入力素子**
  - ２入力の演算について、交換法則と結合法則が成り立つときに、これを多入力演算（３個以上の入力をもつ演算）に拡張することができる

X_0	X_1	X_2	AND	OR	NAND	NOR	XOR	EQ
0	0	0	0	0	1	1	0	0
0	0	1	0	1	1	0	1	1
0	1	0	0	1	1	0	1	1
0	1	1	0	1	1	0	0	0
1	0	0	0	1	1	0	1	1
1	0	1	0	1	1	0	0	0
1	1	0	0	1	1	0	0	0
1	1	1	1	0	0	0	1	1



用語の定義

- リテラル
  - あるの論理変数Xが与えられたとき、Xまたは$\overline{X}$をリテラル(literal)と呼ぶ
- 積項・和項
  - リテラルの論理積（２変数以上、３変数以上であってもよい）を積項(product term)と呼ぶ。同様に、リテラルの論理和を和項(sum term)と呼ぶ。
- 最小項・最大項
  - N個のリテラルの論理積（ただしリテラルはすべて異なる）からなる積項を最小項(minterm)と呼ぶ。同様に、N個のリテラルの論理和（ただしリテラルはすべて異なる）からなる和項を最大項(maxterm)と呼ぶ。

標準形

- 加法標準形(積和標準形)
  - 論理関数を、最小項の論理和として表現したもの
  - 論理関数は必ず加法標準形で表現される(展開定理)
    - 証明: 完備性の証明の後半を繰り返し用いればよい

$$
\begin{aligned}
& F(I_1, I_2, \ldots I_M) \\
= \; & \overline{I_1} \cdot \overline{I_2} \cdot \ldots \cdot \overline{I_M} \cdot F(0,0,\ldots.0) \\
& + \overline{I_1} \cdot \overline{I_2} \cdot \ldots \cdot I_M \cdot F(0,0,\ldots.1) \\
& \ldots\ldots\ldots\ldots \\
& + I_1 \cdot I_2 \cdot \ldots \cdot I_M \cdot F(1,1,\ldots.1)
\end{aligned}
$$

加法標準形と真理値表

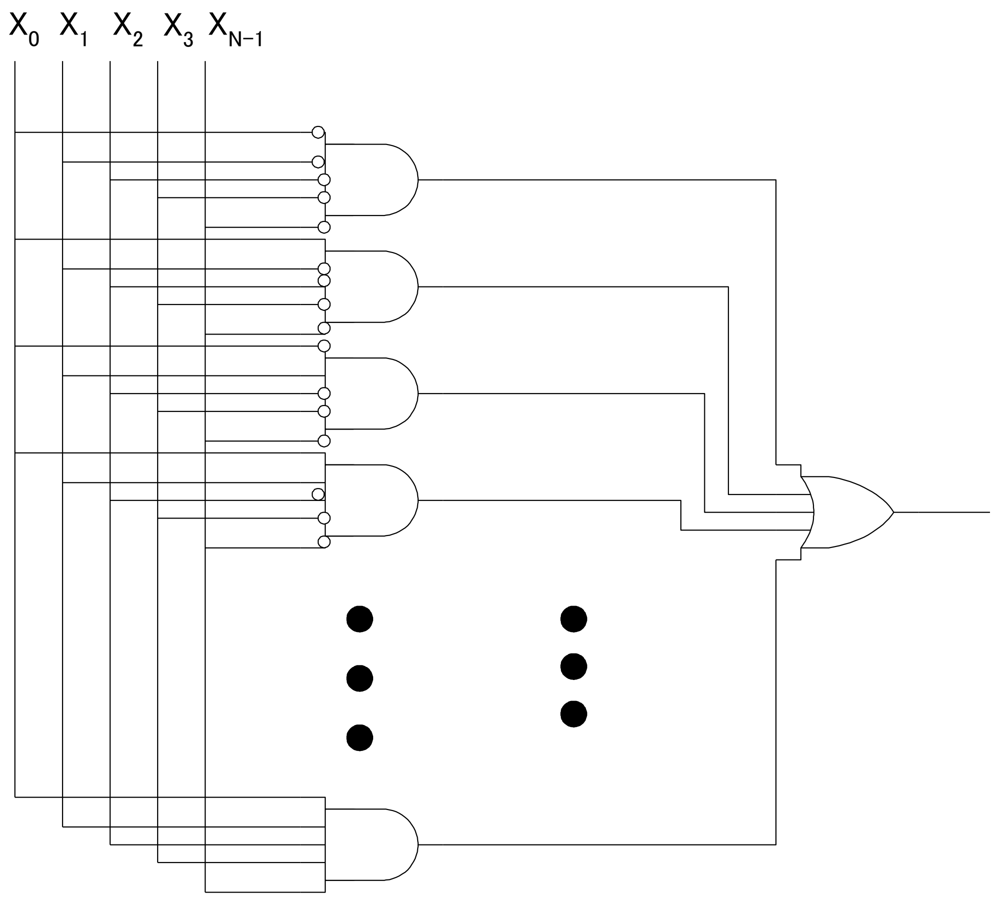
- 加法標準形の最小項
  - 真理値表で、論理関数の値が1になるときのリテラルの組み合わせをあらわしている
- 加法標準形
  - 「論理関数の値が1になるときのリテラルの積（AND）」の論理和（OR）

4入力1出力論理関数の真理値表

X	Y	Z	W	O
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

$$\overline{X}\,\overline{Y}ZW+\overline{X}Y\overline{Z}\,\overline{W}+\overline{X}YZ\overline{W}+X\overline{Y}\,\overline{Z}\,\overline{W}+X\overline{Y}Z\overline{W}+X\overline{Y}ZW+XYZW$$

加法標準形の一般形



注意．加法標準形は、いつも論理関数のもっとも簡単な表現法になるとは限らない

加法標準形の記述法

- 例

4入力1出力論理関数の真理値表

X	Y	Z	W	O	
0	0	0	0	0	(0)
0	0	0	1	0	(1)
0	0	1	0	1	(2)
0	0	1	1	1	(3)
0	1	0	0	1	(4)
0	1	0	1	0	(5)
0	1	1	0	1	(6)
0	1	1	1	0	(7)
1	0	0	0	1	(8)
1	0	0	1	0	(9)
1	0	1	0	1	(10)
1	0	1	1	1	(11)
1	1	0	0	0	(12)
1	1	0	1	0	(13)
1	1	1	0	0	(14)
1	1	1	1	1	(15)

$$O = \Sigma(2, 3, 4, 6, 8, 10, 11, 15)$$

加法標準形の記述法（続）

1. 真理値表を、「入力の昇順」に書き、行番号（0から）を上から順番につける。

   入力の昇順とは、「N個の入力をNビットの２進数とみなして小さいものから順番」、という意味

2. １で出力が1になる行の行番号をΣ（）の括弧内に並べて書く

   各行が該当する最小項を表す

練習問題　＝　宿題

1　ド・モルガンの定理を真理値表を用いて証明せよ

2　次の論理式を加法標準形に直せ。答えはブール代数の式とΣ()の式の両方で示せ。

- $X \oplus Y$
- $X \oplus Y \oplus Z$
- (X＋Y＋Z)・(Y＋Z＋W)

3 論理関数を、最大項の論理積として表現したものを「乗法標準形」と呼ぶ。

- 乗法標準形の求め方を示せ
- 2の最初の2つの式を乗法標準形にせよ