計算機システムの基礎(第8回配布)

第3章　論理回路

（1）集合　　集合、補集合、集合の要素、集合の基本演算

（2）2値論理　と　基本論理回路　　ブール代数、公理、定理、基本論理回路

（3）組み合わせ回路

回路図解析と回路図設計　（論理式から論理回路へ）

真理値表　と　カルノー図、　主加法標準形、

半加算器と全加算器、

簡単化、エンコーダとデコーダ

（4）順序回路

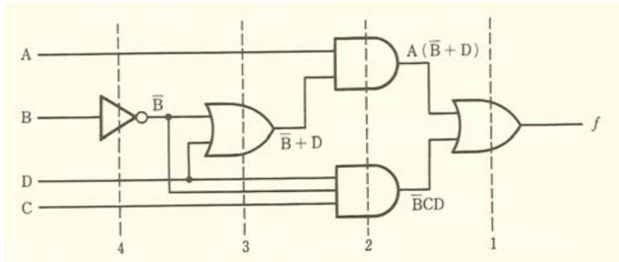
担当：　福井大学　大学院工学研究科
情報・メディア工学専攻
森　眞一郎　(moris@u-fukui.ac.jp)

組み合わせ論理回路の解析(1)

一般系

与えられた論理回路の入力から出力へ向かって、
各ゲートごとにその出力を順次書いていく

ごく自然な解析法！！

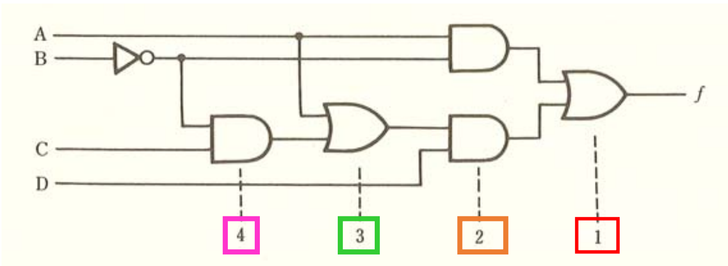


組み合わせ論理回路の合成

多段論理式の場合

今日は地道に変換する方法

$$F = A\overline{B} + D(A + \overline{B}C)$$



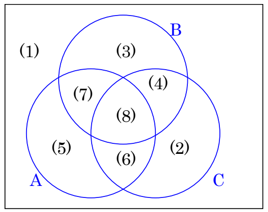
最小項と最大項（1）

[用語の定義]最小項と最大項

- 最小項:
  - 全ての変数が真または偽の形で含まれている論理積項
- 最大項:
  - 全ての変数が真または偽の形で含まれている論理和項

例　3変数の場合

A	B	C	Y
0	0	0	(1)
0	0	1	(2)
0	1	0	(3)
0	1	1	(4)
1	0	0	(5)
1	0	1	(6)
1	1	0	(7)
1	1	1	(8)



ベン図

最小項はベン図上での8つの区画のうち
いずれか一つの区画に対応

⇕

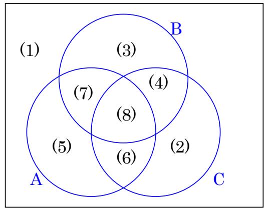
最大項はベン図上での8つの区画のうち
いずれか一つの区画を除いた区画に対応

最小項と最大項（２）

[用語の定義]最小項と最大項
- 最小項:
  - 全ての変数が真または偽の形で含まれている論理積項
- 最大項:
  - 全ての変数が真または偽の形で含まれている論理和項

例　3変数の場合

A	B	C	Y
0	0	0	(1)
0	0	1	(2)
0	1	0	(3)
0	1	1	(4)
1	0	0	(5)
1	0	1	(6)
1	1	0	(7)
1	1	1	(8)



ベン図

最小項はベン図上での8つの区画のうちいずれか一つの区画に対応

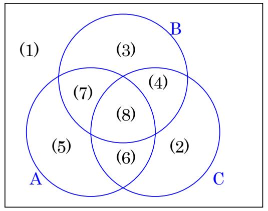
A･(～B)･C

最大項はベン図上での8つの区画のうちいずれか一つの区画を除いた区画に対応

(～A)＋B＋(～C)

....抜いた区画はA･(～B)･C

最小項と最大項（３）



ベン図

最小項はベン図上での8つの区画のうちいずれか一つの区画に対応

$A \bullet \overline{B} \bullet C$

最大項はベン図上での8つの区画のうちいずれか一つの区画を除いた区画に対応

$\overline{A} + B + \overline{C}$

....除いた区画は $A \bullet \overline{B} \bullet C$

対応関係は「ド･モルガンの定理」を使えば一意にわかる！！

$$\overline{X \bullet Y \bullet Z} = \overline{X} + \overline{Y} + \overline{Z}$$

$$\overline{X + Y + Z} = \overline{X} \bullet \overline{Y} \bullet \overline{Z}$$

標準形（１）

[用語の定義]積和形と和積形

- 積和形(sum of products form):
  - 論理関数が論理積項の和として論理式表現された場合、その論理式は積和形表現であると呼ぶ。
  - 特に、全ての論理積項が最小項で表現されている場合は、
    標準積和形　（あるいは加法標準形）と呼ぶ。
- 和積形(product of sums form):
  - 論理関数が論理和項の積として論理式表現された場合、その論理式は和積形表現であると呼ぶ。
  - 特に、全ての論理和項が最大項で表現されている場合は、
    標準和積形　（あるいは乗法標準形）と呼ぶ。

標準形（２）

[用語の定義]積和形と和積形

- 積和形(sum of products form):
  - 積和形表現
    f（A、B、C）= A･B + B･C + A･C　　... 積 の 和
  - 標準積和形表現（あるいは加法標準形表現）
    f（A、B、C）= A･B･C＋A･B･(～C)＋A･(～B)･C＋(～A)･B･C
    ... 最小項 の 和
- 和積形(product of sums form):
  - 和積形表現
    g(A,B,C) = (~A + ~B)(~B + ~C)(~A + ~C)　... 和 の 積
  - 標準和積形　（あるいは乗法標準形）
    g(A,B,C) = (~A + ~B + ~C)(~A + ~B + C)(~A + B + ~C)(A + ~B + ~C)
    ... 最大項の積

組み合わせ論理回路の合成

積和型あるいは和積型論理式の場合(2段)

積和型の場合 AND-OR型ゲート構成 で合成	和積型の場合 OR-AND型ゲート構成 で合成

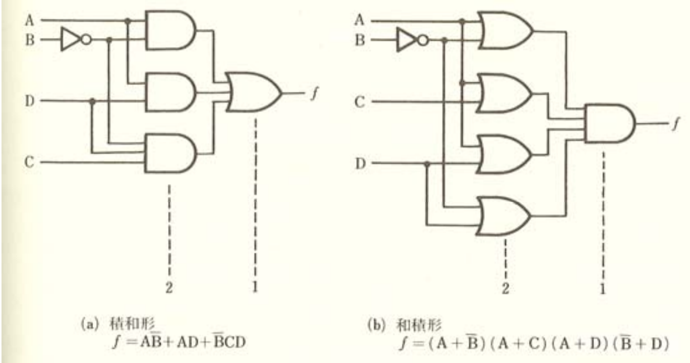
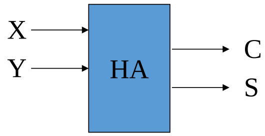


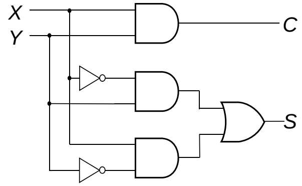
図 4.16 2段構成のゲート回路の例

半加算器(HA:Half Adder)

- 入力X、Yが与えられたときに、その桁の和S (Sum)と桁上がりC (Carry)を出力する回路



X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



$C = X \bullet Y$

$S = \overline{X} \bullet Y + X \bullet \overline{Y} = X \oplus Y$

[復習2]2進数の加算(その2)

- n桁の場合

000 + 000 = 0000
000 + 001 = 0001
…
001 + 011 = 0100
…
101 + 011 = 1000
…
111 + 111 = 1110

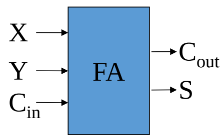
i桁目の計算では、入力Xi, Yiと i-1桁目からの桁上げC_{i-1}から、その桁の和Siと桁上げCiを生成する

101+011= (0100+0010)+(1+1) 1桁目
= (0100+0010)+ 10
= (0100+0000)+(00+10+10) 2桁目
= (0100+0000)+(100)
…

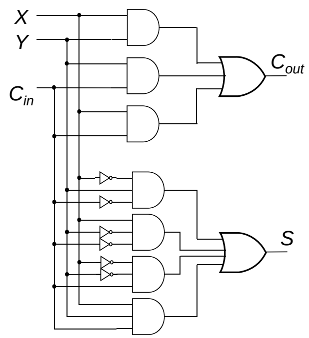
2項の和では C_{i-1} は 0 or 1

全加算器(FA:Full Adder)

- 入力X, Yと下位桁からの桁上げCinから、その桁の和Sと桁上げCoutを生成する回路



X	Y	C_{in}	C_{out}	S
0	0	0	0	0
0	1	0	0	1
1	0	0	0	1
1	1	0	1	0
0	0	1	0	1
0	1	1	1	0
1	0	1	1	0
1	1	1	1	1



$C_{out} = X \bullet Y + Y \bullet C_{in} + X \bullet C_{in}$

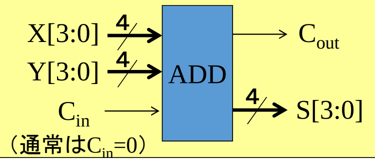
$S = \overline{X} \bullet Y \bullet \overline{C_{in}} + X \bullet \overline{Y} \bullet \overline{C_{in}} + \overline{X} \bullet \overline{Y} \bullet C_{in} + X \bullet Y \bullet C_{in}$

C_{out}は多数決、
Sは3変数の排他的論理和

Nビット加算器(ADD:Adder)

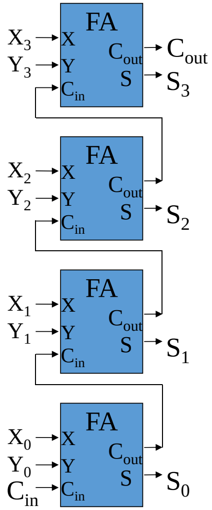
教科書p.205

N=4の例

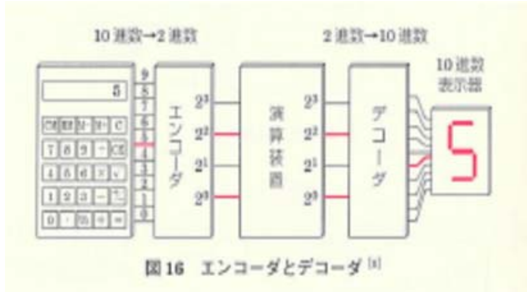


- リプルキャリ方式(右図)
  - FAをN個直列に接続して構成
    - i桁目の桁上げ出力C_{out}をi+1桁目の桁上げ入力C_{in}に接続
  - 回路は簡単であるが、桁上げ信号(Carry)が逐次的に伝播するため遅い

高速化手法の例
　桁上げ選択方式、桁上げ先見方式



復号器(Decoder)/符号器(Encoder)



2進化10進符号器
(BCDエンコーダ)

入力	出力
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

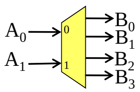
出力サイズが変化する
符号器の例 (モールス符号の親戚)

入力	出力
A	101100
B	111010100
C	111011100
D	1110100
E	100
F	101011100
G	1111100
…	
T	1100
…	
Z	111110100

デコーダ(decoder)/デマルチプレクサ(demultiplexer)

2^N個の出力のうち、Nビットの入力Aに対応する出力のみ1をそれ以外の出力には0を出す回路

「例1」2-to-4デコーダ(2入力デマルチプレクサ)



A_1	A_0	B_3	B_2	B_1	B_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

回路図は自分で考えて見ましょう

教科書p.102

エンコーダ(符号化器、encoder)

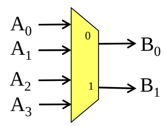
Nビットの入力を、Mビットの出力に符号化する回路

・一般のエンコーダはN＝2^Mとしてデコーダの逆の動作を示す。
　(この場合、入力禁止の組み合わせが生じる)
・より広義には、符号化ルールを定めると当該ルールに対応した
　任意のエンコーダを作ることができる。

「例1」4-to-2エンコーダ

A_3	A_2	A_1	A_0	B_1	B_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

入力制限1:入力が1となるのは1ビットのみ
入力制限2:必ずいづれかの入力ビットが1



$$B_0 = A_3 + A_1$$
$$B_1 = A_3 + A_2$$

[レポート課題] (論理式と論理回路)

(1) 以下の論理式を証明せよ

(a) $A \bullet C + \overline{A} \bullet B = (A + B) \bullet (\overline{A} + C)$

(b) $\overline{A \bullet \overline{B \bullet \overline{C}}} + B \bullet C = \overline{A} + B$

(2) 上記(a)の左辺の論理式に対応する論理回路、ならびに
右辺の論理式に対応する論理回路を設計して図示せよ。

(3) 教科書P.117　問題7の (1)~(3)の論理式に対応する論理回路を設計せよ。

(A4　レポート用紙提出のこと。表紙をつける必要はないが、1枚目の上側余白に学生番号、氏名を記入のこと
両面を使って解答してよいが、複数ページに跨る場合は必ず　ホッチキス留め　すること。)