論理設計
東野担当５回目
授業スライド
１１月４日

基礎工学部情報科学科　東野輝夫

授業時間変更＆中間試験のお知らせ

- 長谷川先生のご都合で，長谷川先生担当の水曜３限の計算機言語の授業と水曜４限の東野の論理設計の授業を下記のように交換して実施します．
- １１月２５日（水）
  - ３限　計算機言語　→　論理設計　の授業に変更
  - ４限　論理設計　　（この日は２コマ続けて論理設計の授業を実施）
- １２月　９日（水）
  - ３限　計算機言語　（この日は２コマ続けて計算機言語の授業を実施）
  - ４限　論理設計　→　計算機言語　の授業に変更

- 東野担当の論理設計の中間試験について
  - １１月２５日（水）の４限（15:10 開始）に東野担当の論理設計の中間試験を実施します．
  - この日は３限に普通の授業を実施し，４限に試験を実施します．試験はCLE上で実施します．詳細は別途連絡しますが，下記で実施予定．
    - 15:10　試験の方法を説明（CLEビデオ or Zoom）
    - 15:20　CLEに試験問題を掲示し，制限時間を決めて時間内にレポート課題提出と同じ方法で答案をuploadしてもらうことで答案回収します．

授業計画の変更

長谷川先生との授業の入れ替えに伴い，章の説明の順番を変更します

- 授業計画：東野担当の授業計画を下記のように変更します．
  1. ドントケアを含む論理関数の簡単化（６章）
  2. フリップフロップとレジスタ（１０章）
  3. 同期式順序回路（Mealy型, Moore型順序回路）（１１章）
  4. カルノー図を用いた論理関数の簡単化（１章から５章の復習）
  5. 組合せ論理回路設計、よく用いられる組み合わせ回路（７章，８章）
  6. 加減算器とALU、順序回路の簡単化（９章，１２章前半）
  7. 演習
  8. 順序回路の簡単化、カウンタ（１２章後半，１３章）
  9. 中間試験（１章～１１章）
  10. ＩＣを用いた順序回路の実現（１５章）
  11. 演習
  12. ＣＰＵの設計（付録）
  13. ＣＰＵの設計，演習
  14. 乗算器と除算器（１４章）
  15. 期末試験（１２章～１５章，付録）

８コマ目の授業を11/25の３限に実施
９コマ目の試験を11/25の４限に実施
中間試験の範囲を１１章までに変更
期末試験の範囲を１２章以降に変更

質問について

- メールで随時問い合わせや質問にお答えしますので，何かあれば，higashino@ist.osaka-u.ac.jp までメールで質問して下さい.
- また、時間を決めてZoomなどを用いて質問にお答えすることも可能ですので、まずはメールで疑問点や問い合わせ事項などを連絡して下さい。

お願い

本テキストや授業のビデオなどの電子ファイルを他人に転送したり、ネットへアップロードすることなどを禁止します。

著作権保護

- この授業のテキスト（教科書）や授業スライド、授業ビデオの著作権保護に努めて下さい.
- この授業のビデオやスナップショットを録画したり，それらを他の人に転送したり，インターネット上で公開したりすることを禁止します.
- この授業で利用するスライドにはオーム社の教科書の図などが含まれているので，著作権保護の観点から，この授業スライドの公開につながる行為は謹んでください.
- 来年度は CLE を使ったメディア授業でなく，対面の授業ができることを期待していますが，今年度の演習課題の解答が事前に公開されたりすると，来年度の授業で同じ演習課題が使えなくなり，授業テキストの大幅な修正が必要になるため，協力をお願いします.

OSAKA UNIVERSITY

前回の授業の補足

演習問題

- 論理式α, β, R を次の論理式とする.

  $\alpha = (p \vee q \vee r) \wedge (p \rightarrow r) \wedge (r \rightarrow w)$　　$\beta = (r \wedge w) \vee (q \wedge \neg r)$

  $R = \alpha \rightarrow \beta$

- このとき，¬R を和積形で表し ¬R のリゾルベントをとっていって空節 0 を導出する方法 によって，R の恒真性を示せ.

考慮時間

- 5-10分間程度で問題を解いてみてください．その間，ビデオを止めてください.

- この頁は30秒程度で次の頁に移行します.

演習問題解答

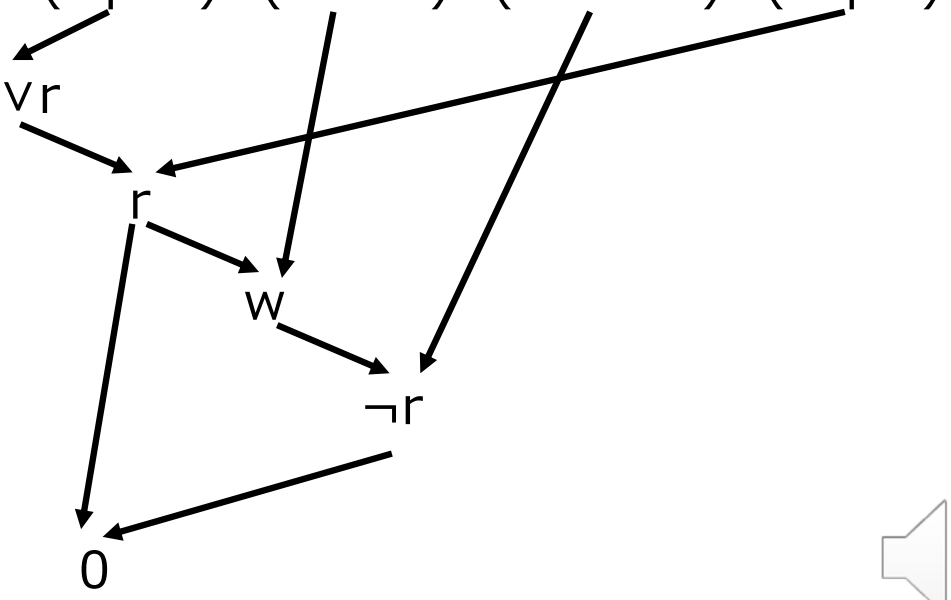
- 論理式α, β, R を次の論理式とする.

  $\alpha = (p \vee q \vee r) \wedge (p \rightarrow r) \wedge (r \rightarrow w)$　　$\beta = (r \wedge w) \vee (q \wedge \neg r)$

  $R = \alpha \rightarrow \beta$

- このとき，¬R を和積形で表し ¬R のリゾルベントをとっていって空節 0 を導出する方法 によって，R の恒真性を示せ.

$R = \alpha \rightarrow \beta$　　$\neg R = (p \vee q \vee r) \wedge (\neg p \vee r) \wedge (\neg r \vee w) \wedge (\neg r \vee \neg w) \wedge (\neg q \vee r)$



$R = \alpha \rightarrow \beta$ なら
$\neg R = \alpha \wedge \neg\beta$

$\beta = (r \wedge w) \vee (q \wedge \neg r)$ なら
$\neg\beta = \neg(r \wedge w) \wedge \neg(q \wedge \neg r)$
$= (\neg r \vee \neg w) \wedge (\neg q \vee r)$

上図より ¬R＝0 となり，Rは恒真である.

第７章
組合せ論理回路設計

第７章　組合せ論理回路設計

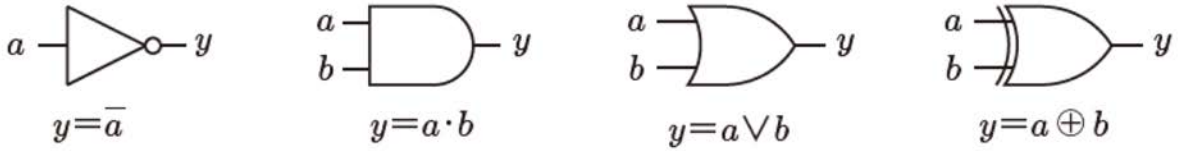
- この章のねらい

7章 組合せ論理回路設計

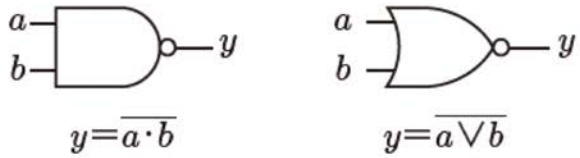
５章, ６章では論理式の簡単化を学んだ．本章では論理式から組合せ論理回路を実現する方法について学ぶ．組合せ論理回路を構成する基本要素である論理ゲートを紹介した後，それらを用いて論理式から組合せ論理回路を実現する．回路最適化や多出力関数の簡単化についても概要を紹介する．

組合せ論理回路

- 組合せ論理回路（Combinational logic circuit）
  - 組合せ論理回路は n 個の入力と m 個の出力を持つ．各出力値は入力の論理変数を変数とする論理関数の計算値であり，出力値は入力値にのみ依存して一意に定まる．つまり，m 個の論理関数の出力を計算する回路である．
  - 組合せ論理回路は，図７・１のように，NOTゲート，ANDゲー ト，ORゲートなどの論理ゲートを接続して構成する．



（a） NOT ゲート （b） AND ゲート （c） OR ゲート （d） XOR ゲート



（e） NAND ゲート （f） NOR ゲート

図 7・1 論理ゲートのシンボルと対応する論理関数

組合せ論理回路の実現

- AND, OR, NOT ゲートによる実現
  - 基本的には論理式で与えられた論理関数に対して，論理演算を論理ゲートに置き換えて回路を実現する.
- 例題７・１
  - 論理関数$f(x_2,x_1,x_0) = x_2x_1\bar{x}_0 \vee x_2(\bar{x}_1 \vee x_0)$ を実現する組合せ論理回路を AND, OR, NOT ゲートを用いて構成せよ.

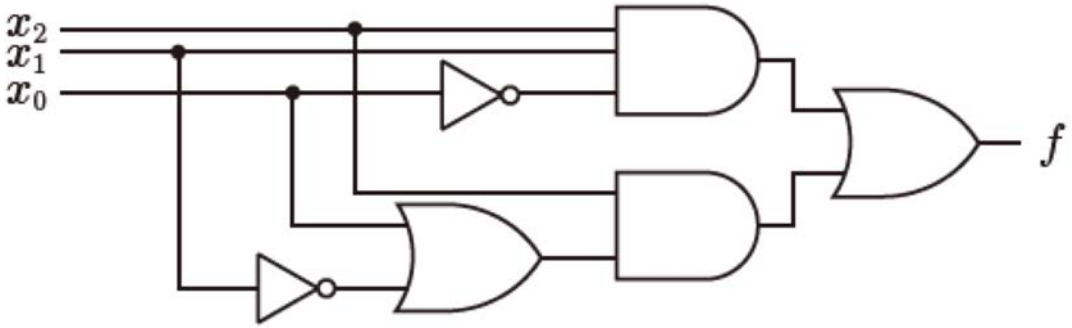


図 7・2　例題 7.1 の組合せ論理回路

NAND, NOT ゲートによる実現

- 現在のCMOS 論理回路では，AND, OR を用いた回路よりもNAND, NOR を用いた回路の方が動作速度や素子数の観点で好ましい（コラム参照）.
- NAND とNOR を比較した場合，NAND の方が動作速度の点で好ましい. そこで，AND, OR, NOT で構成された回路からNAND, NOT で構成された回路に変換する方法を紹介する.
- NAND ゲートには二つの表現があることに注目する（図7·3）．教科書は右の表現をすることもありますが，その回路図は左の表現と考えて取り扱って下さい.
- 図7·3(a) は通常の NANDゲートである．図7·3(b) は OR ゲートの入力部分に否定を意味する丸印（◦）がついている．ここでド・モルガン則である $\overline{a \cdot b} = \bar{a} \vee \bar{b}$ を考える.

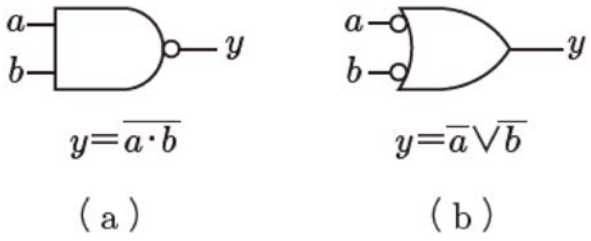


図 7・3　NAND ゲートの二つの表現

NAND, NOT ゲートによる実現

- 図7·3(a) は通常の NANDゲートである．図7·3(b) は OR ゲートの入力部分に否定を意味する丸印（◦）がついている．ここでド・モルガン則である $\overline{a \cdot b} = \bar{a} \vee \bar{b}$ を考える．
- 図 7·4(a) はAND ゲートの出力にOR ゲートが接続されており，$f(x_3,x_2,x_1,x_0) = x_3x_2 \vee x_1x_0$ は積和形で表されている．
- この回路に対して，AND ゲートの出力とOR ゲートの入力部分に否定を表す丸印（◦）をそれぞれつける（図7·4(b)）．
- ANDゲートはNANDゲートに，ORゲートも図7·3(b)のNANDゲートで構成されており，右端の図のようにAND-OR回路をNANDゲートのみの２段回路に変換できる（AND-OR回路は１段目のANDゲートも２段目のORゲートも，共にNANDゲートに置き換えれば良い）．

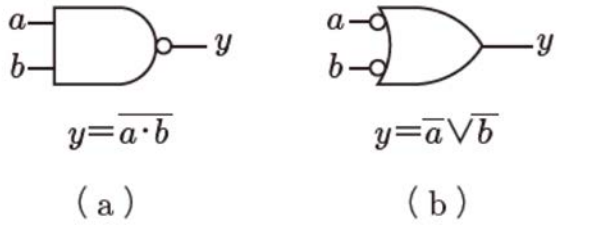


図 7・3 NAND ゲートの二つの表現

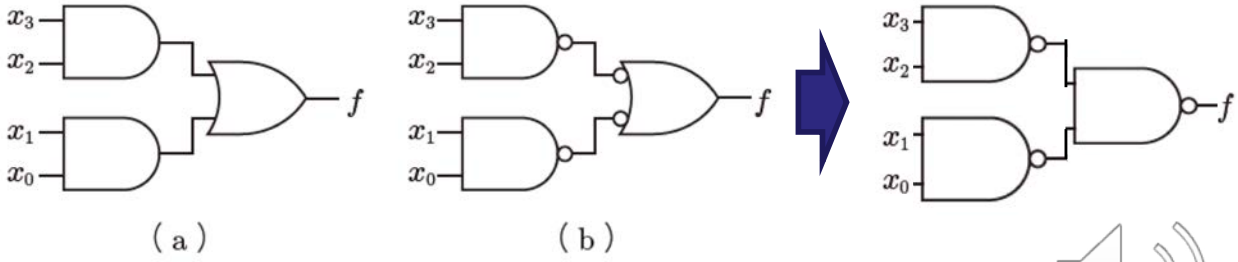


図 7・4 積和形部分の変換

NAND, NOT ゲートによる実現

例題７・２

- 図7·5(a), (b) の回路 A, B を NANDゲート，NOTゲートのみで実現せよ.

（ヒント）

- (a) は積和形の部分を見つけて置き換えるとよい.
- (b) は積和形の部分の置換えを行う際に，各配線が二重否定となるように，必要に応じてNOT ゲートを追加する.

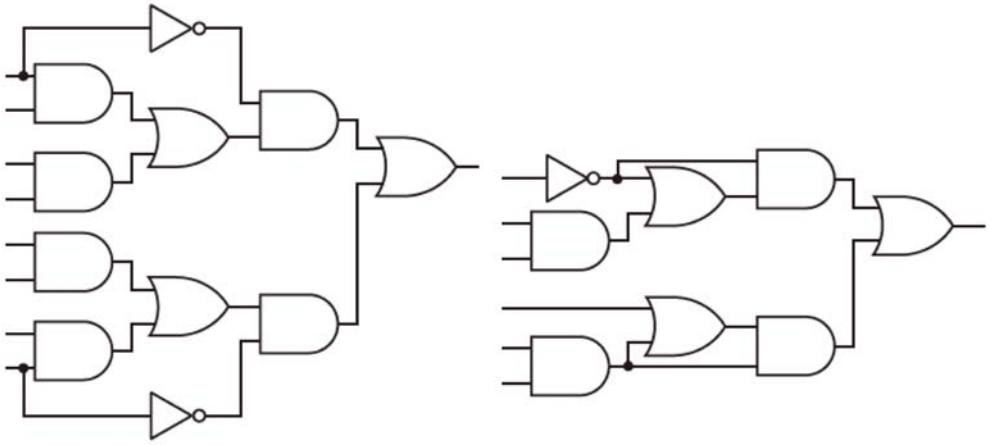


図 7・5 例題 7·2 の変換前の回路

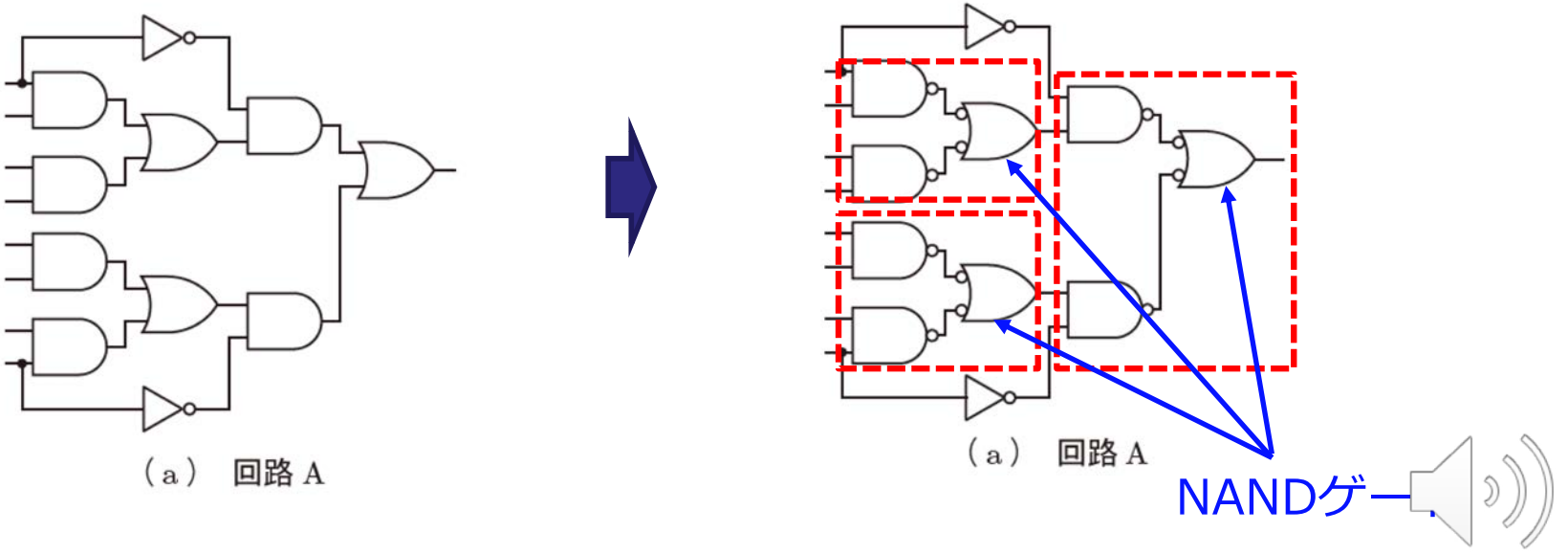
NAND, NOT ゲートによる実現

例題７・２

- 図7·5(a), (b) の回路 A, B を NANDゲート，NOTゲートのみで実現せよ.

（ヒント）

- (a) は積和形の部分を見つけて置き換えるとよい.
- (b) は積和形の部分の置換えを行う際に，各配線が二重否定となるように，必要に応じてNOT ゲートを追加する.



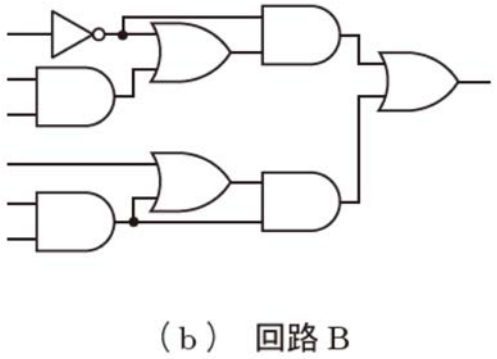
NAND, NOT ゲートによる実現

例題７・２

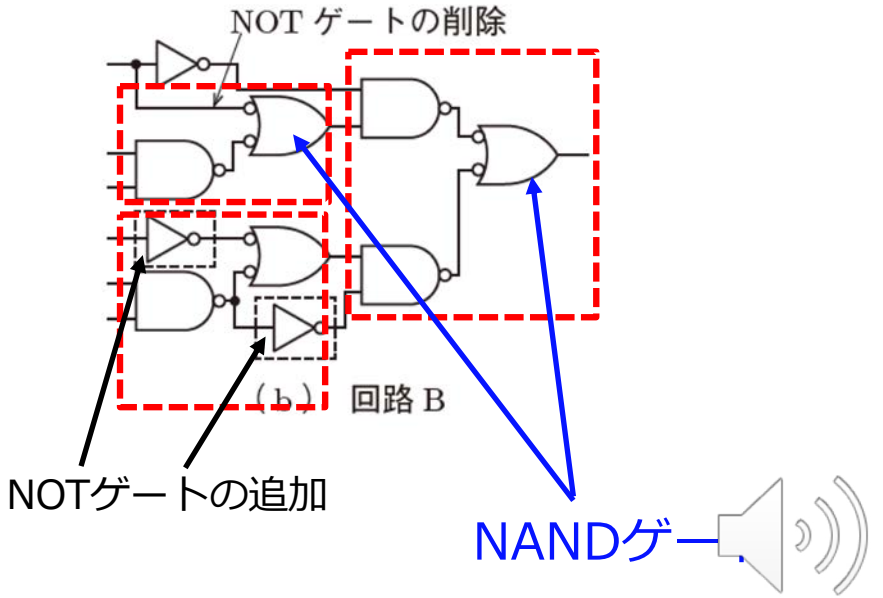
- 図7·5(a), (b) の回路 A, B を NANDゲート，NOTゲートのみで実現せよ.

（ヒント）

- (a) は積和形の部分を見つけて置き換えるとよい.
- (b) は積和形の部分の置換えを行う際に，各配線が二重否定となるように，必要に応じてNOT ゲートを追加する.



（b） 回路 B



組合せ論理回路の設計法

- 良い組合せ論理回路とは？
  - 動作が速くてゲート数が少ないものが良い（信号が論理ゲートを通過するには有限の遅延時間が必要であり，それらの時間の和が回路の動作速度の上限を与えるため）
  - 多段論理回路はこれまで考えてきた二段論理回路に比べてゲート数が少なくてすむことも多い（多段論理回路の設計法はこの教科書の範囲外，脚注の文献などを参照）
  - 二段論理回路で実現する場合，まず最簡積和形を求め，対応する組合せ回路で実現する
  - この際，ファンイン数や使えるゲートの種類に制限があれば，回路を変形する

ファンイン制限

- 論理ゲートの入力数をファンイン数と呼ぶ．論理関数を表す論理式を作成した場合，多変数の論理積や論理和が出てくることがある．
  - 多入力のANDゲートやORゲートに置き換える場合，現実的なファンイン数の上限は４程度であり，そのままでは組合せ論理回路として実現できないことがある．
  - このような場合には図7･7 に示すように，複数の論理ゲートを用いて同じ論理関数を実現する．
  - 図7･7は三つの４入力AND ゲートと一つの３入力AND ゲートを用いて12変数の論理積を実現した例である．

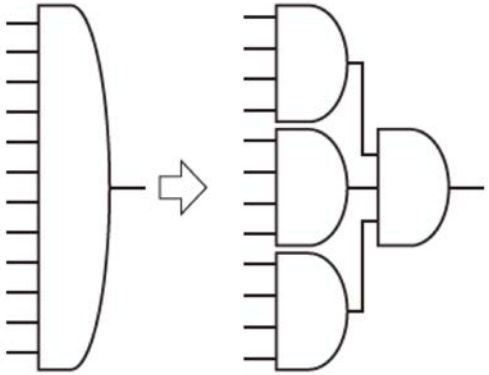


図 7・7　12 変数の論理積の実現例

回路の簡単化

- 回路の簡単化は多岐にわたるため，ここではわかりやすい回路の簡単化例を図7·8を用いて紹介するにとどめる
- 図7·8(a) の簡単化は二重否定の削除であり，冗長になっている二つのNOTゲートを削除している．
- 図7·8(b) の簡単化はド・モルガンの法則（ $\bar{x}_1 \cdot \bar{x}_0 = \overline{x_1 \vee x_0}$ ）を利用しており，入力側に二つあった NOT ゲートが出力側の一つに削減されている．

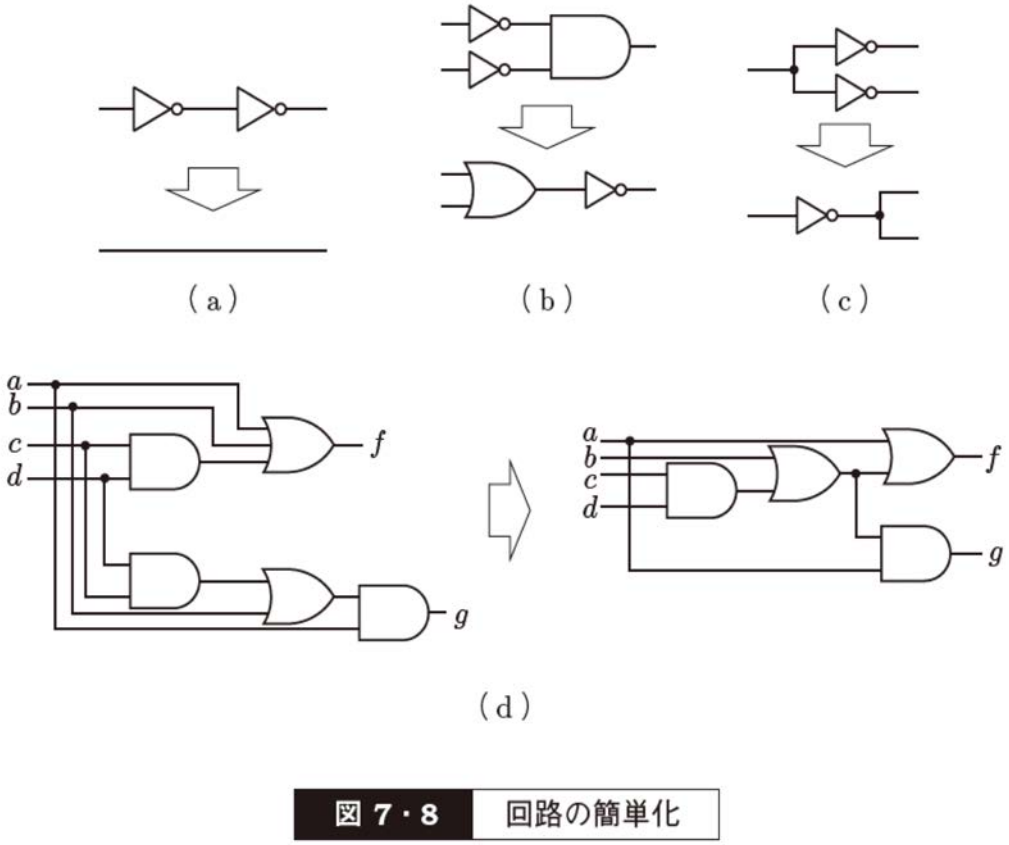


図 7・8　回路の簡単化

- 図7·8(c), (d) は回路の共有化である．(c)では分岐前に論理ゲートを移動することにより論理ゲートを削減している．(d) では，同じもしくは類似の論理式の計算を共有することによって必要な論理ゲート数を削減している．

多出力論理回路の簡単化

例題７・３

- 論理関数が複数あり，それぞれを積和形で表して，組合せ回路として実現することを考える．これらの積和形の論理式に共通の積項が存在すると，組合せ回路として実現するときにANDゲートが共有でき，必要なゲート数が削減できるので，全体の積項数の最小化が重要となる．
- 一方，個々の論理関数の簡単化では，複数の論理関数を実現するために必要な積項数が最小になるとは限らない．
- 複数の論理関数（多出力論理関数）の簡単化においては，各関数の主項のみならず，関数のすべての組合せについてそれらの積の主項も求め，これらの主項による各々の関数の最小被覆を求める．具体的な方法を次の例題（図７・９）を用いて説明する．
- f_a, f_b, f_cの最簡積和形は
  - $f_a = \bar{x}_1 x_0 \vee x_2 x_0 \vee \bar{x}_2 x_1 \bar{x}_0$
  - $f_b = x_1 \bar{x}_0 \vee x_2 \bar{x}_0 \vee \bar{x}_2 \bar{x}_1 x_0$
  - $f_c = x_1 \bar{x}_0 \vee x_2 \bar{x}_0 \vee x_2 \bar{x}_1$
  - 論理積７個，論理和３個

    （ $x_1 \bar{x}_0$，$x_2 \bar{x}_0$ は f_b, f_c で共用）

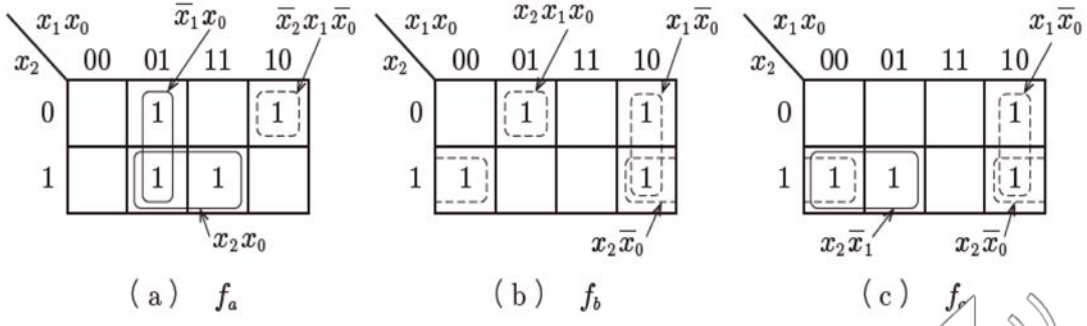


図 7・9　例題 7.3 の f_a, f_b, f_c のカルノー図

多出力論理回路の簡単化

例題７・３

- f_a, f_b, f_cの最簡積和形は
  - $f_a = \bar{x}_1 x_0 \vee x_2 x_0 \vee \bar{x}_2 x_1 \bar{x}_0$
  - $f_b = x_1 \bar{x}_0 \vee x_2 \bar{x}_0 \vee \bar{x}_2 \bar{x}_1 x_0$
  - $f_c = x_1 \bar{x}_0 \vee x_2 \bar{x}_0 \vee x_2 \bar{x}_1$
  - 論理積７個，論理和３個（ $x_1\bar{x}_0$，$x_2\bar{x}_0$ は f_b, f_c で共用）
- $x_1\bar{x}_0$，$x_2\bar{x}_0$のように f_a, f_b, f_cで共用できる主項が見つかれば積項数を削減できる
- $x_1\bar{x}_0$，$x_2\bar{x}_0$は $f_b \cdot f_c$ の主項になっている（図７・10参照）
- f_b の最簡積和形の候補： f_b，$f_a \cdot f_b$，$f_b \cdot f_c$，$f_a \cdot f_b \cdot f_c$ の主項

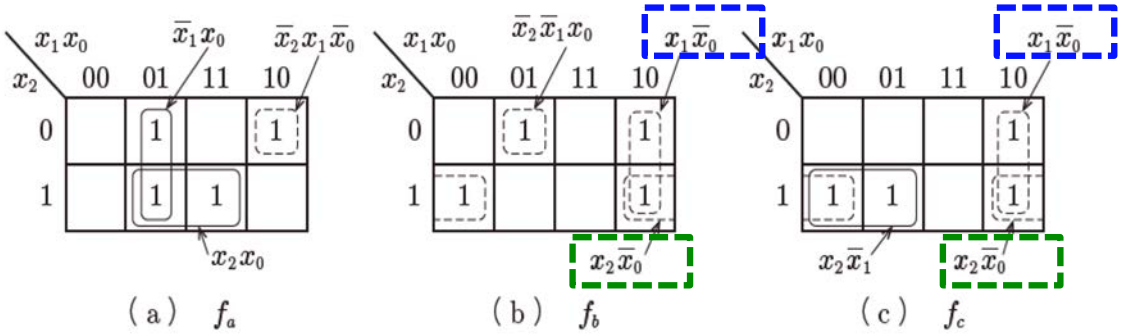


図 7・9　例題 7.3 の f_a, f_b, f_c のカルノー図

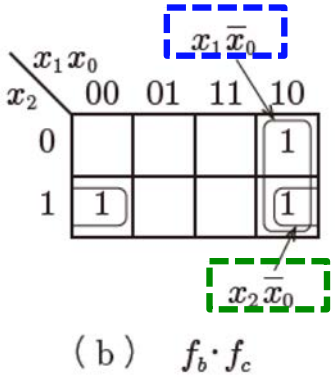


図 7・10

多出力論理回路の簡単化

例題７・３

- f_a, f_b, f_cの最簡積和形は
  - $f_a = \bar{x}_1x_0 \vee x_2x_0 \vee \bar{x}_2x_1\bar{x}_0$
  - $f_b = x_1\bar{x}_0 \vee x_2\bar{x}_0 \vee \bar{x}_2\bar{x}_1x_0$
  - $f_c = x_1\bar{x}_0 \vee x_2\bar{x}_0 \vee x_2\bar{x}_1$
  - 論理積７個，論理和３個

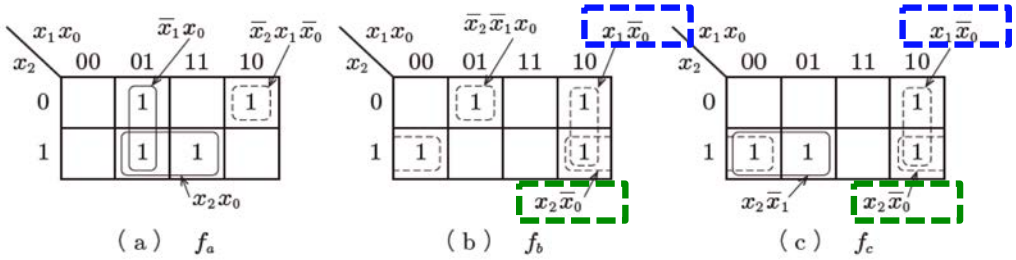


図 7・9　例題 7.3 の f_a, f_b, f_c のカルノー図

- f_a の最小項は，f_a, $f_a \cdot f_b$, $f_c \cdot f_a$, $f_a \cdot f_b \cdot f_c$ に含まれる主項を用いて被覆する．ここで $f_a \cdot f_b \cdot f_c$ に含まれる主項 $\bar{x}_2x_1\bar{x}_0$ は f_a, $f_a \cdot f_b$, $f_c \cdot f_a$ にも主項として含まれている．このとき，$f_a \cdot f_b \cdot f_c$ に含まれる主項 $\bar{x}_2x_1\bar{x}_0$は，f_a だけでなく f_b, f_cの被覆にも利用できるため，f_a, $f_a \cdot f_b$, $f_c \cdot f_a$に含まれる $\bar{x}_2x_1\bar{x}_0$ よりも優先して被覆に利用すべきである．
- このような優先関係により被覆に用いられない主項を図7·9，図7·10 では点線で示している．

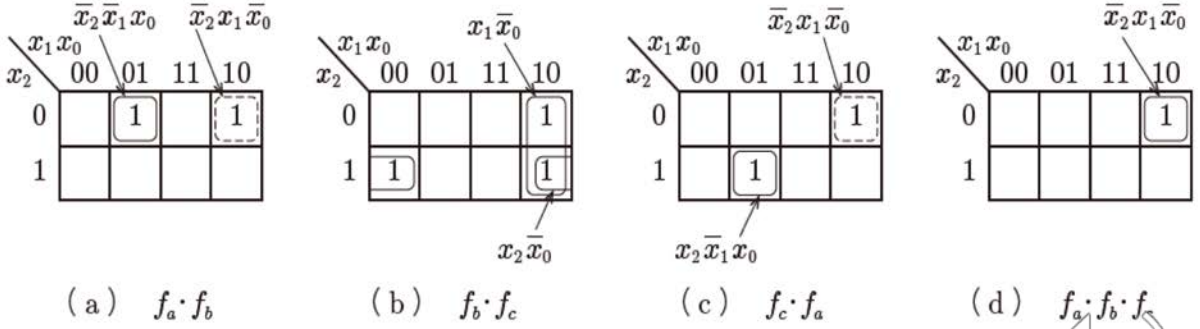


図 7・10　例題 7.3 の $f_a \cdot f_b, f_b \cdot f_c, f_c \cdot f_a, f_a \cdot f_b \cdot f_c$ のカルノー図

多出力論理回路の簡単化

例題７・３

- クワイン・マクラスキー法で用いた主項表による最小被覆の導出を適用する（図7·11）.
- f_a, f_b, f_c それぞれについて必須行を探し出して，最小被覆に加える.
- ここでは 図7·11(a) の左に必須と書いた行が該当する（縦の列に１が一つしか書かれていない場合，その最小項を覆う主項が必須項）.

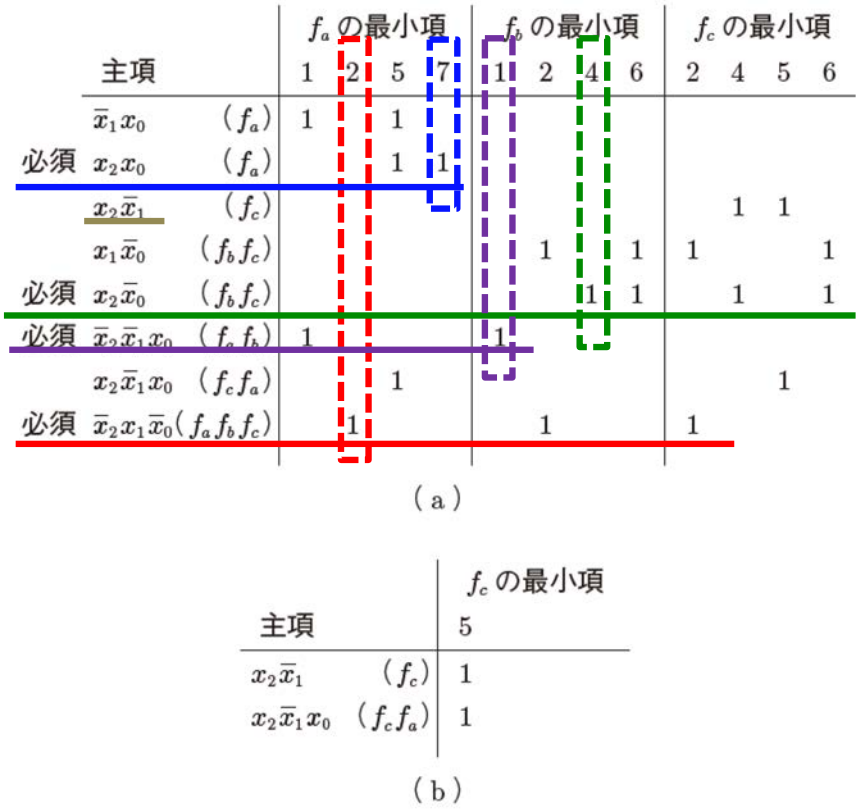


図 7・11　例題 7·3 の主項表を用いた最小被覆の導出

多出力論理回路の簡単化

例題７・３

- クワイン・マクラスキー法で用いた主項表による最小被覆の導出を適用する（図7·11）.
- f_a, f_b, f_c それぞれについて必須行を探し出して，最小被覆に加える.
- ここでは 図7·11(a) の左に必須と書いた行が該当する．これらの必須項に被覆された最小項を除くと，残された被覆すべき最小項は f_c の 5 のみである.
- この最小項を被覆可能な主項のうち，リテラル数の小さい $x_2\bar{x}_1$ を選択する.
- この結果 f_a, f_b, f_c の最簡多出力論理関数は以下のようになる（論理積５個，論理和３個に減少する）.
  - $f_a = x_2x_0 \vee \bar{x}_2\bar{x}_1x_0 \vee \bar{x}_2x_1\bar{x}_0$
  - $f_b = x_2\bar{x}_0 \vee \bar{x}_2\bar{x}_1x_0 \vee \bar{x}_2x_1\bar{x}_0$
  - $f_c = x_2\bar{x}_0 \vee x_2\bar{x}_1 \vee \bar{x}_2x_1\bar{x}_0$

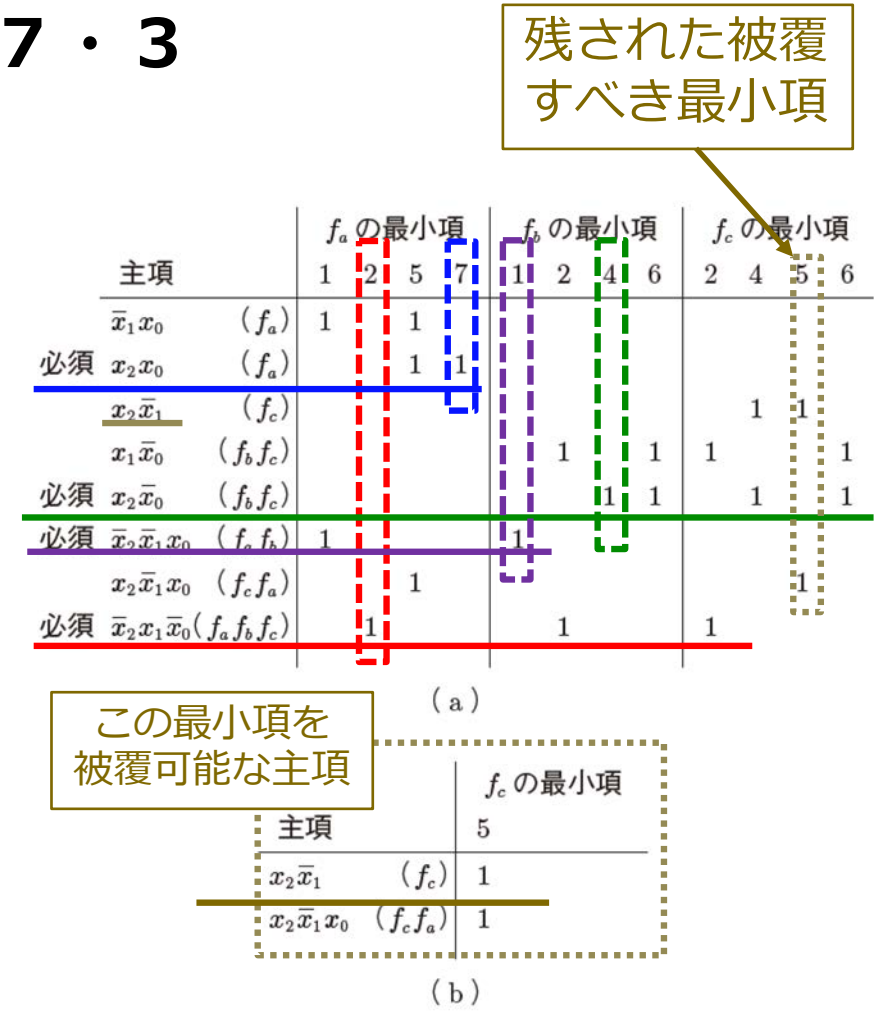


図 7・11　例題 7.3 の主項表を用いた最小被覆の導出

論理ゲートの相対的な遅延時間

- 論理ゲートの相対的な面積と遅延時間の具体例を 表7·1 に示す.
- NOTゲートの面積と遅延時間を 1 として表している. 2 入力の NAND ゲートの相対遅延時間は 1.2 で, 2 入力ANDゲートの相対遅延時間 2.2 と比較してかなり小さい.
- 同一の NANDゲートでも入力数が 4 となると相対遅延時間が 1.5 に増加している.

表 7・1　論理ゲートの相対的な面積と遅延時間の例

ゲートの種類	入力数	相対面積	相対遅延時間
NOT	1	1.0	1.0
NAND	2	1.3	1.2
NAND	3	1.7	1.3
NAND	4	2.3	1.5
NOR	2	1.3	1.2
NOR	3	1.7	1.3
NOR	4	2.3	1.6
XOR	2	2.7	1.8
XOR	3	7.0	4.0
AND	2	1.7	2.2
AND	3	2.3	2.4
AND	4	2.7	2.6
OR	2	1.7	2.5
OR	3	2.3	2.7
OR	4	2.7	3.0

OSAKA UNIVERSITY

演習問題

1 図 7·2 の組合せ回路を，NAND, NOT ゲートで実現せよ.

2 例題 7·3 の f_a, f_b, f_c に対してそれぞれ最簡積和形を求め，組合せ論理回路を AND, OR, NOT ゲートを用いて実現せよ.

3 例題 7·3 で求めた f_a, f_b, f_c に対する最簡多出力論理関数を用いて，組合せ論理回路を AND, OR, NOT ゲートを用いて実現せよ.

4 次の論理関数 f_a, f_b を同時に実現する最簡多出力論理関数を求めよ.

$$f_a = \overline{x}_2\overline{x}_0 \vee \overline{x}_2 x_1$$

$$f_b = x_1 x_0 \vee x_2 x_1$$

5 次の論理関数 f_a, f_b を同時に実現する最簡多出力論理関数を求めよ.

$$f_a = \overline{x}_2\overline{x}_0 \vee \overline{x}_2 x_1 \vee x_1\overline{x}_0$$

$$f_b = \overline{x}_2\overline{x}_1\overline{x}_0 \vee x_2 x_0 \vee x_2 x_1$$

演習問題 1

- 図7·2 の組合せ回路を，NAND, NOT ゲートで実現せよ.

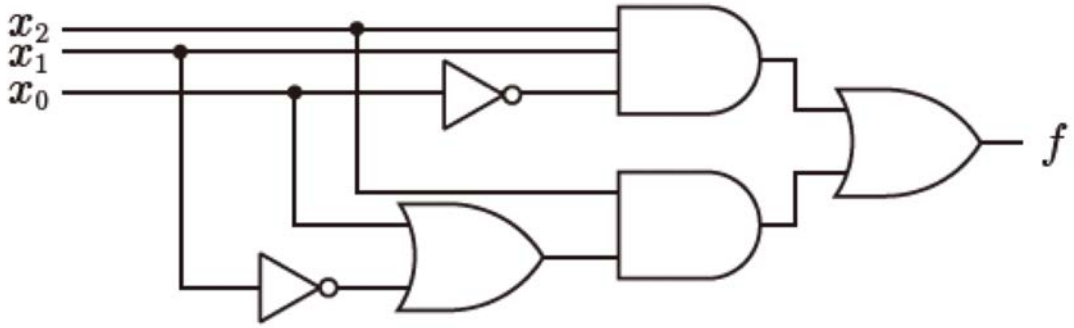


図 7・2　例題 7·1 の組合せ論理回路

考慮時間

- 5分間程度で問題を解いてみてください．その間，ビデオを止めてください.

- この頁は30秒程度で次の頁に移行します.

演習問題１解答

- 図7･2 の組合せ回路を，NAND, NOT ゲートで実現せよ.

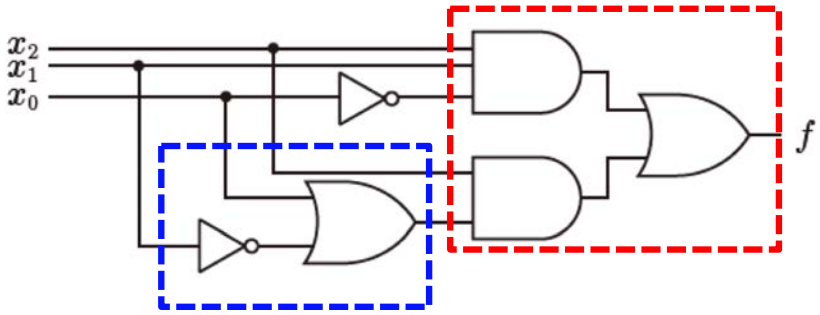


図 7・2　例題 7･1 の組合せ論理回路

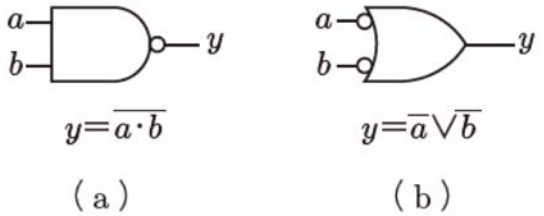
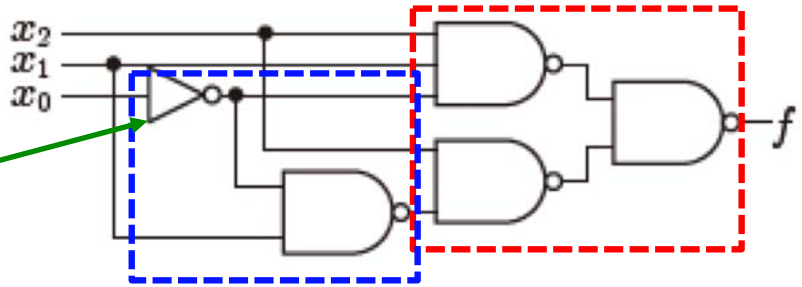


図 7・3　NAND ゲートの二つの表現

x_0の否定を
左下のNAND
ゲートの
上側の入力
になるように
変形



解図 7・1

演習問題 2

- 例題7·3 の f_a, f_b, f_c に対してそれぞれ最簡積和形を求め，組合せ論理回路を AND, OR, NOT ゲートを用いて実現せよ.

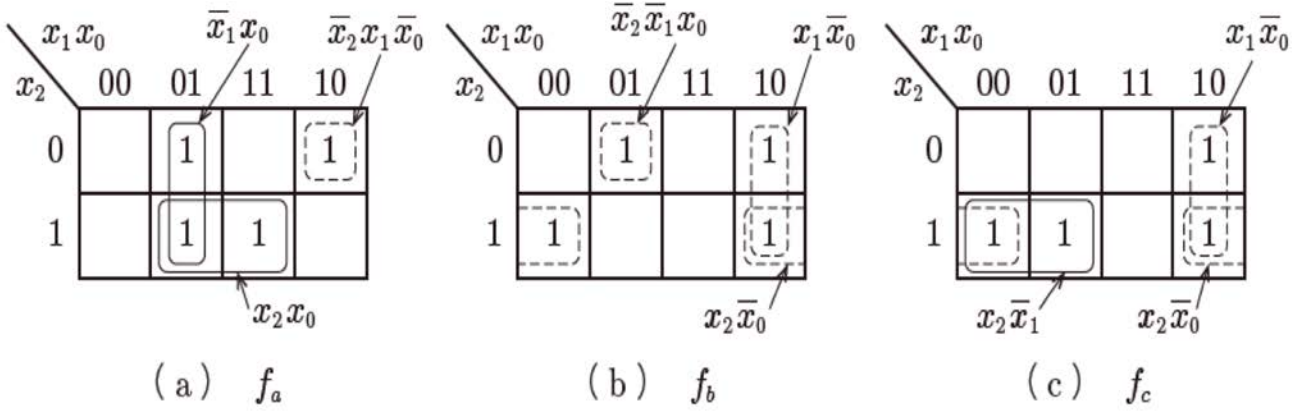


図 7・9 例題 7·3 の f_a, f_b, f_c のカルノー図

考慮時間

- 5分間程度で問題を解いてみてください．その間，ビデオを止めてください.

- この頁は30秒程度で次の頁に移行します.

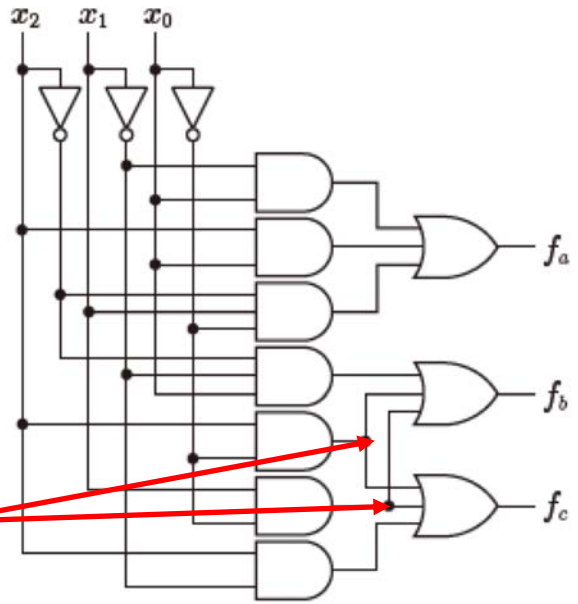
演習問題２解答

- 例題7·3 の f_a, f_b, f_c に対してそれぞれ最簡積和形を求め，組合せ論理回路を AND, OR, NOT ゲートを用いて実現せよ.

- f_a, f_b, f_cの最簡積和形は
  - $f_a = \bar{x}_1 x_0 \vee x_2 x_0 \vee \bar{x}_2 x_1 \bar{x}_0$
  - $f_b = x_1 \bar{x}_0 \vee x_2 \bar{x}_0 \vee \bar{x}_2 \bar{x}_1 x_0$
  - $f_c = x_1 \bar{x}_0 \vee x_2 \bar{x}_0 \vee x_2 \bar{x}_1$

  （論理積７個，論理和３個）
- その回路図は右図の通り

$x_1\bar{x}_0$と$x_2\bar{x}_0$は
f_bとf_cで共用



解図 7・2

演習問題 3

- 例題7·3 で求めた f_a, f_b, f_c に対する最簡多出力論理関数を用いて，組合せ論理回路を AND, OR, NOT ゲートを用いて実現せよ.

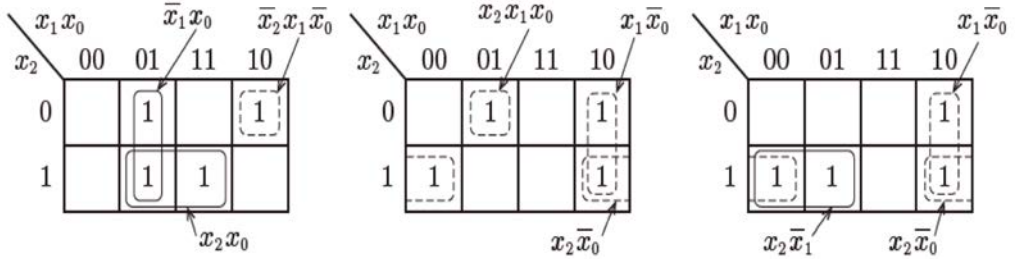


図 7・9 例題 7·3 の f_a, f_b, f_c のカルノー図

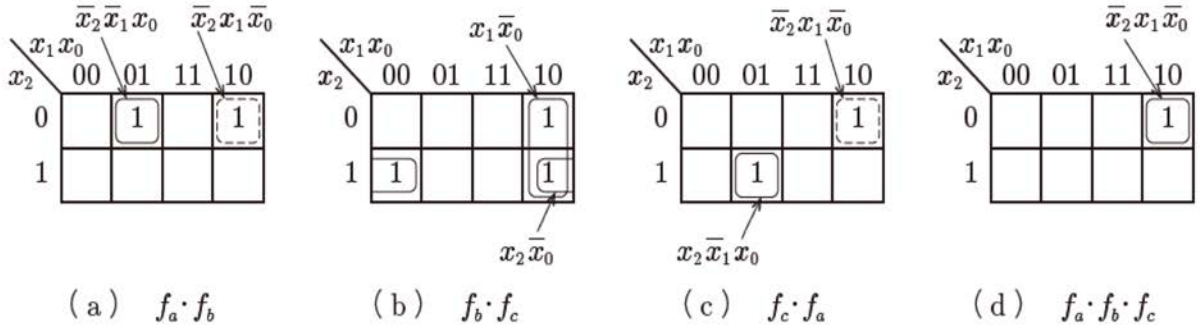


図 7・10 例題 7·3 の $f_a \cdot f_b, f_b \cdot f_c, f_c \cdot f_a, f_a \cdot f_b \cdot f_c$ のカルノー図

考慮時間

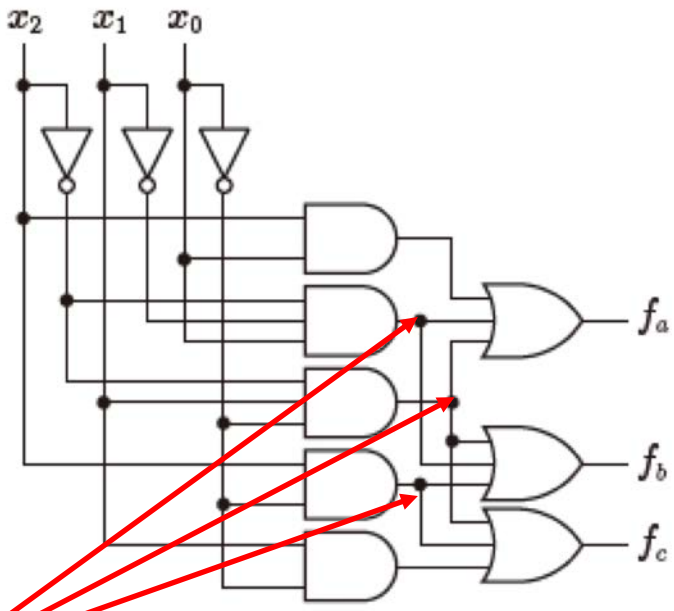
- 5分間程度で問題を解いてみてください．その間，ビデオを止めてください.

- この頁は30秒程度で次の頁に移行します.

演習問題３解答

- 例題7･3 で求めた f_a, f_b, f_c に対する最簡多出力論理関数を用いて，組合せ論理回路を AND, OR, NOT ゲートを用いて実現せよ.

- f_a, f_b, f_c の最簡多出力論理関数は下記（論理積５個，論理和３個）.
  - $f_a = x_2x_0 \vee \bar{x}_2\bar{x}_1x_0 \vee \bar{x}_2x_1\bar{x}_0$
  - $f_b = x_2\bar{x}_0 \vee \bar{x}_2\bar{x}_1x_0 \vee \bar{x}_2x_1\bar{x}_0$
  - $f_c = x_2\bar{x}_0 \vee x_2\bar{x}_1 \vee \bar{x}_2x_1\bar{x}_0$
- その回路図は右図の通り
  - $\bar{x}_2x_1\bar{x}_0$はf_aとf_bとf_cで共用
  - $\bar{x}_2\bar{x}_1x_0$はf_aとf_bで共用
  - $x_2\bar{x}$はf_bとf_cで共用



$\bar{x}_2x_1\bar{x}_0$と$\bar{x}_2\bar{x}_1x_0$と$x_2\bar{x}$は共用

解図 7・3

演習問題４

- 次の論理関数 f_a, f_b を同時に実現する最簡多出力論理関数を求めよ.
  - $f_a = \bar{x}_2\bar{x}_0 \vee \bar{x}_2x_1$
  - $f_b = x_1x_0 \vee x_2x_1$

考慮時間

- 5分間程度で問題を解いてみてください．その間，ビデオを止めてください.

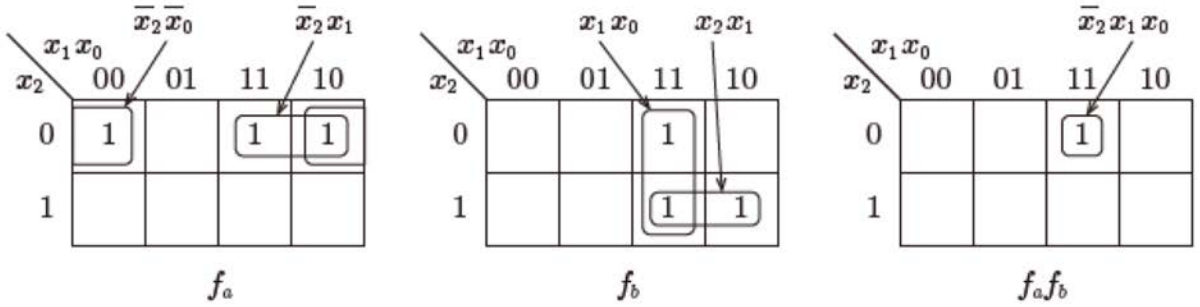
- この頁は30秒程度で次の頁に移行します.

演習問題４解答

- 次の論理関数 f_a, f_b を同時に実現する最簡多出力論理関数を求めよ.
  - $f_a = \bar{x}_2\bar{x}_0 \vee \bar{x}_2x_1$
  - $f_b = x_1x_0 \vee x_2x_1$

（解答）

- $f_a \cdot f_b = \bar{x}_2x_1x_0$
- クワイン・マクラスキー法で用いた主項表による最小被覆の導出を適用する（解表7･1&2）
- $f_a = \bar{x}_2\bar{x}_0 \vee \bar{x}_2x_1x_0$
- $f_b = x_2x_1 \vee \bar{x}_2x_1x_0$



解図 7・4

解表 7・1

	主項		最小項 f_a 0	2	3	最小項 f_b 3	6	7
必須	$\bar{x}_2\bar{x}_0$	(f_a)	1	1				
	$\bar{x}_2x_1$	(f_a)		1	1			
	x_1x_0	(f_b)				1		1
必須	x_2x_1	(f_b)					1	1
	$\bar{x}_2x_1x_0$	(f_af_b)			1	1		

解表 7・2

主項		最小項 f_a 3	f_b 3
$\bar{x}_2x_1$	(f_a)	1	
x_1x_0	(f_b)		1
$\bar{x}_2x_1x_0$	(f_af_b)	1	1

演習問題５

- 次の論理関数 f_a, f_b を同時に実現する最簡多出力論理関数を求めよ.
  - $f_a = \bar{x}_2\bar{x}_0 \vee \bar{x}_2x_1 \vee x_1\bar{x}_0$
  - $f_b = \bar{x}_2\bar{x}_1\bar{x}_0 \vee x_2x_0 \vee x_2x_1$

考慮時間

- 5分間程度で問題を解いてみてください．その間，ビデオを止めてください.

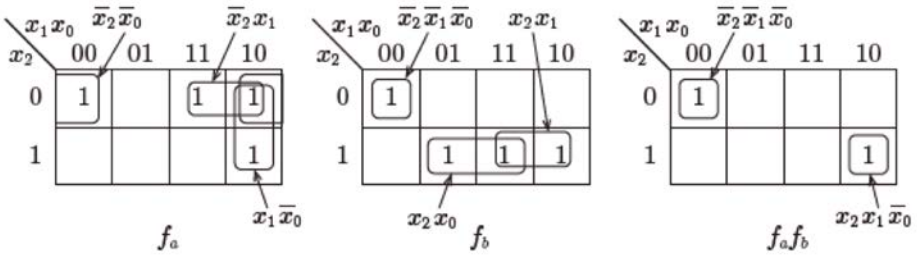
- この頁は30秒程度で次の頁に移行します.

演習問題５解答

- 次の論理関数 f_a, f_b を同時に実現する最簡多出力論理関数を求めよ.
  - $f_a = \bar{x}_2\bar{x}_0 \vee \bar{x}_2x_1 \vee x_1\bar{x}_0$
  - $f_b = \bar{x}_2\bar{x}_1\bar{x}_0 \vee x_2x_0 \vee x_2x_1$

（解答）

  - $f_a \cdot f_b = \bar{x}_2\bar{x}_1\bar{x}_0 \vee x_2x_1\bar{x}_0$
  - クワイン・マクラスキー法で用いた主項表による最小被覆の導出を適用する（解表7·3&4）.
  - $f_a = \bar{x}_2x_1 \vee x_2x_1\bar{x}_0 \vee \bar{x}_2\bar{x}_1\bar{x}_0$
  - $f_b = x_2x_0 \vee x_2x_1\bar{x}_0 \vee \bar{x}_2\bar{x}_1\bar{x}_0$



解図 7・5

解表 7・3

	主項	最小項 f_a 0	2	3	6	最小項 f_b 0	5	6	7
	$\bar{x}_2\bar{x}_0$ (f_a)	1	1						
必須	$\bar{x}_2x_1$ (f_a)		1	1					
	$x_1\bar{x}_0$ (f_a)		1		1				
	x_2x_1 (f_b)							1	1
必須	x_2x_0 (f_b)						1		1
支配されている	$\bar{x}_2\bar{x}_1\bar{x}_0$ (f_b)					1			
必須	$\bar{x}_2\bar{x}_1\bar{x}_0$ (f_af_b)	1				1			
	$x_2x_1\bar{x}_0$ (f_af_b)				1			1	

解表 7・4

	最小項 f_a	最小項 f_b
主項	6	6
$x_1\bar{x}_0$ (f_a)	1	
x_2x_1 (f_b)		1
$x_2x_1\bar{x}_0$ (f_af_b)	1	1

第８章
よく用いられる
組合せ回路

第８章　よく用いられる組合せ回路

- この章のねらい

8章 よく用いられる組合せ回路

プロセッサに代表される集積回路には，定番回路としてよく用いられる組合せ回路が存在する．これらの機能を知っておくことは，計算機システムの理解においても重要である．本章では，代表的な組合せ回路を取り上げて，その機能と回路構成を学ぶ．ただし，加減算器や算術論理演算ユニット（ALU）は次の９章で取り上げる．

2進デコーダ

- 2 進デコーダ（binary decoder）は，n 個の入力 $i_{n-1},\cdots, i_0$ に対して，2^n個の出力$o_{2^n-1},\cdots, o_0$ の出力をもつ．入力$(i_{n-1},i_{n-2}, \cdots, i_0)$を 2 進数と見なし，その 10 進数表記を k とすると，2 進デコーダは k 番目の出力 o_k のみに 1 を出力し，他の出力には 0 を出力する．
- n = 3 の場合の2 進デコーダについて，真理値表を 表8･1 に，回路図を 図8･1 に示す．

表 8・1　2 進デコーダ $(n = 3)$ の真理値表

i_2	i_1	i_0	o_7	o_6	o_5	o_4	o_3	o_2	o_1	o_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

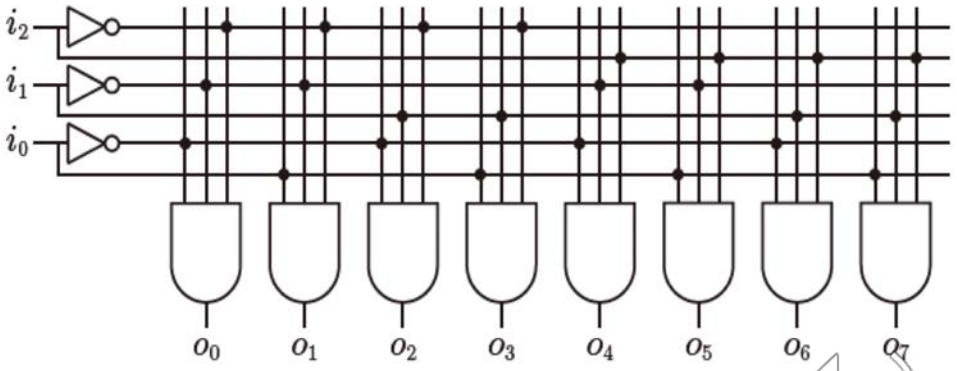


図 8・1　2 進デコーダの回路図

2進エンコーダ

- 2 進エンコーダ（binary encoder）は，2 進デコーダとは逆に，2^n 個の入力 $i_{2^n-1}, \cdots, i_0$ と n 個の出力 $o_{n-1}, \cdots, o_0$ をもつ．2^n 個の入力うち一つだけが 1 となり他は 0 であるとする．k 番目の入力が 1 のとき，k の 2 進数表現が $(o_{n-1}, o_{n-2}, \cdots, o_0)$ に出力される．
- 表8·2 に n = 3 の場合の 2 進エンコーダの真理値表を示す．出力 o_2, o_1, o_1 の論理式は以下のように表され，OR ゲートのみを用いて実現できることがわかる．
  - $o_2 = i_7 \vee i_6 \vee i_5 \vee i_4$
  - $o_1 = i_7 \vee i_6 \vee i_3 \vee i_2$
  - $o_0 = i_7 \vee i_5 \vee i_3 \vee i_1$

表 8・2　2 進エンコーダ ($n = 3$) の真理値表

i_7	i_6	i_5	i_4	i_3	i_2	i_1	i_0	o_2	o_1	o_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

2進エンコーダ

- 表8·2 のエンコーダは $i_{2^n-1}, \cdots, i_0$ のうち，一つしか 1 が入力されない場合しか用いることができない．複数の 1 が入力される場合があるとき，入力に優先順位を付けたプライオリティエンコーダが用いられる．表8·3 に i の添字が大きい入力が優先されるプライオリティエンコーダ（priority encoder）の真理値表を示す．また，図8·2 に回路図を示す．

表 8・3　2 進プライオリティエンコーダ ($n = 3$) の真理値表

i_7	i_6	i_5	i_4	i_3	i_2	i_1	i_0	o_2	o_1	o_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	1	X	X	X	X	1	0	0
0	0	1	X	X	X	X	X	1	0	1
0	1	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

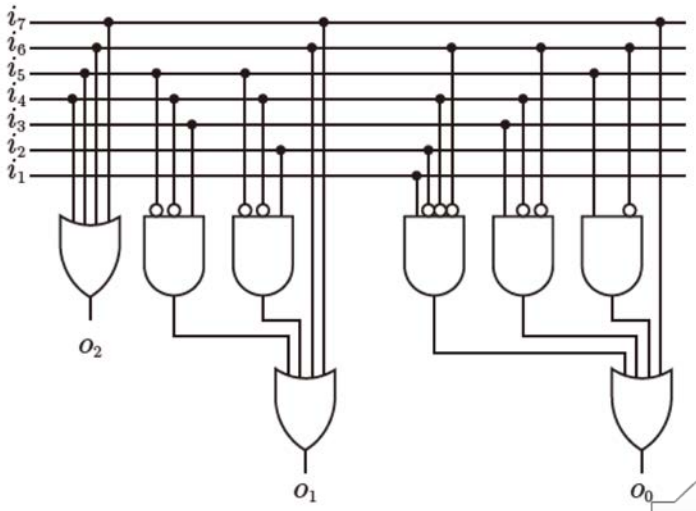


図 8・2　プライオリティエンコーダ ($n = 3$) の回路図

マルチプレクサ

例８・１

- マルチプレクサ（multiplexer）は，n ビットの制御入力 $s_{n-1},\cdots, s_0$ の値によって，2^n 個のデータの入力 $i_{2^n-1},\cdots, i_0$ のいずれかを出力o に出力する回路である．マルチプレクサはセレクタ（selector）とも呼ばれる．
- 例として n = 2 とし，$(s_1, s_0) = (0, 0), (0, 1), (1, 0), (1, 1)$ のときにそれぞれ i_0, i_1, i_2, i_3 を o に出力するマルチプレクサを考える．出力 o の積和形の論理式を求めよ．また，出力 o を実現する論理回路をNANDゲートとNOTゲートを用いて実現せよ．

（解答）　$o = i_0\bar{s}_1\bar{s}_0 \vee i_1\bar{s}_1s_0 \vee i_2s_1\bar{s}_0 \vee i_3s_1s_0$

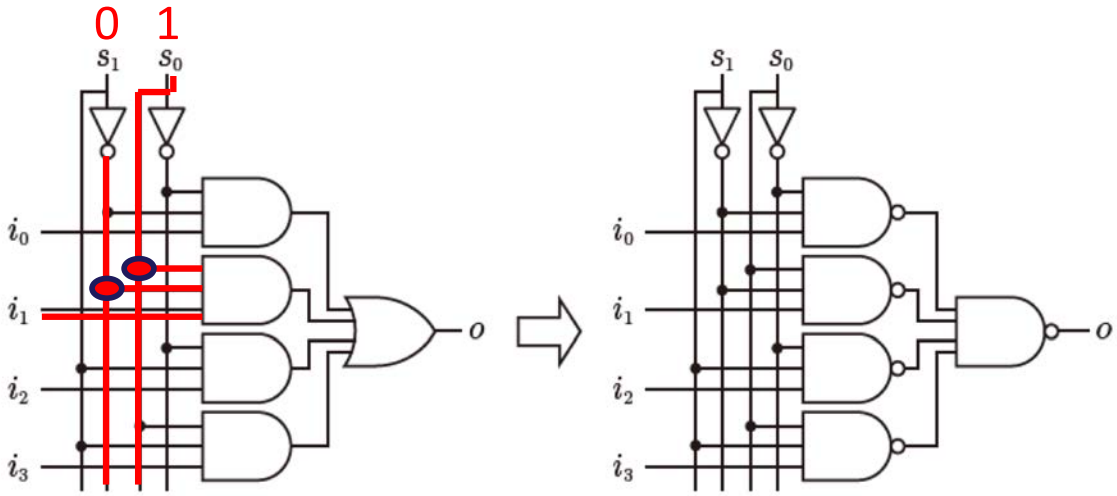


図 8・3　4 入力マルチプレクサの回路図

比較回路

- 二つの n ビットの 2 進数 $X = (x_{n-1},\cdots, x_0)$, $Y = (y_{n-1},\cdots, y_0)$ が入力され，X >Y のとき，出力 o に 1 を出力し，X ≤ Y のとき 0 を出力する比較回路（comparator circuit）を考える．2*n 個の入力変数をもつ組合せ回路を，5 章で学んだカルノー図を用いて最簡積和形を求めて設計することは，n が 3 以上になると難しい．そこで，二つの考え方で比較回路を設計する．

（考え方１）

- 中間変数 g_i, e_i を考える $(0 \le i \le n-1)$．g_i は，$x_i > y_i$ のときに 1 となり，それ以外のときは 0 である．ei は，$x_i = y_i$ のときに 1 となり，それ以外のときは 0 である．このとき，g_i, e_i はそれぞれ以下の論理式で表される．
  - $g_i = x_i\bar{y}_i$
  - $e_i = x_iy_i \vee \bar{x}_i\bar{y}_i$
- 例としてn = 3 の場合を考える．数の比較において上位の桁の大小関係がより重要であることを考えると，o は次のように表される．
  - $o = g_2 \vee e_2 \cdot (g_1 \vee e_1g_0)$
    $= x_2\bar{y}_2 \vee (x_2y_2 \vee \bar{x}_2\bar{y}_2) \cdot (x_1\bar{y}_1 \vee (x_1y_1 \vee \bar{x}_1\bar{y}_1) \cdot x_0y_0)$

比較回路

（考え方２）

- 各桁において，入力 x_i, y_i と下位桁の比較結果 c_i を用いて，その桁の比較結果 c_{i+1} を出力する 1 ビットの比較器を考える．
- 各桁では $x_i > y_i$ のとき c_{i+1} が 1，$x_i < y_i$ のとき c_{i+1} は 0 と，下位の桁の結果によらずc_{i+1} の値が決まる．一方，$x_i = y_i$ のときには，下位桁の比較結果 c_i が c_{i+1} の値となる．このc_{i+1} の真理値表は表8·4 で表され，最簡積和形は

$$c_{i+1} = x_i\bar{y}_i \vee c_i x_i \vee c_i\bar{y}_i$$

である．回路図は図8·4(a) で表される．
- この 1 ビット比較器を直列に接続すると，図 8·4(b) のように n ビットの比較器を構成できる．
- このように基本ブロックを組み合わせた回路設計は大きな回路や多入力回路の設計に適している

（nビットの比較回路の遅延時間は n に比例する）

表 8・4　1 ビット比較器の真理値表

c_i	x_i	y_i	c_{i+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

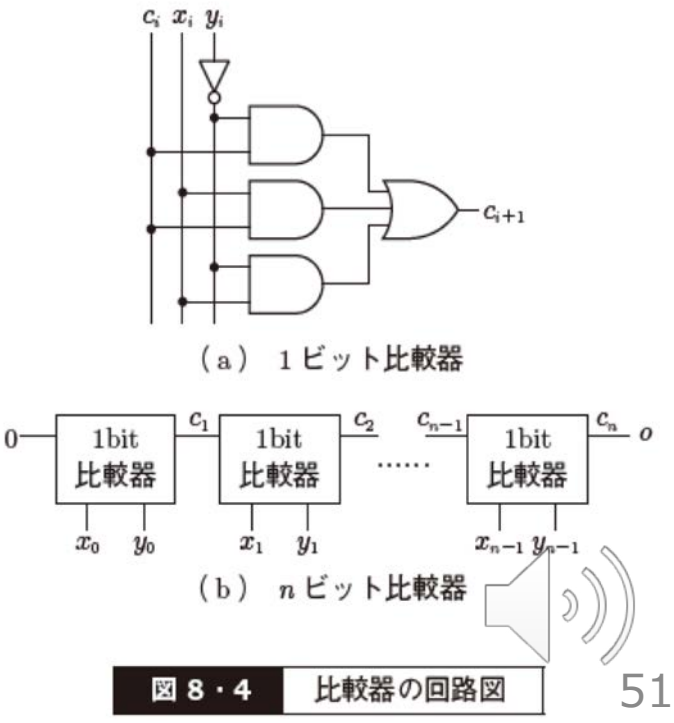


図 8・4　比較器の回路図

パリティ回路

- パリティビットは，データの保存や通信において利用される簡単な誤り検出符号である．与えられた n ビットのデータに対して 1 ビットのパリティビットを加え，(n+1) ビット中に含まれる 1 の数が偶数（あるいは奇数）となるように，パリティビットの値は定められる．通信時に利用した場合，受信側で (n + 1) ビット中の 1 の数を数え，偶数（あるいは奇数）であるかを確認すると，1 ビットエラーの有無が検出できる．4 ビットのデータに対してパリティビットを生成する回路，ならびに 1 ビットエラーを検出するパリティビットチェック回路を 図8･5 に示す．この例では，4ビットの排他的論理和（x1 ⊕ x2 ⊕ x3 ⊕ x4）が1 のとき，5 ビット目にパリティビットとして 1 を与えることで 5 ビット中に含まれる 1 の数が常に偶数になるようにパリティビットを生成している．

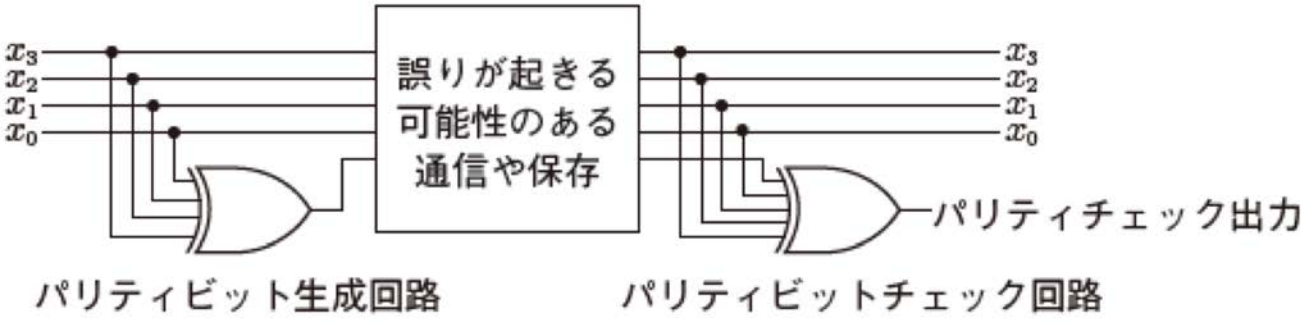


図 8・5 パリティビット生成回路とパリティビットチェック回路

OSAKA UNIVERSITY

演習問題

1 図 8-1 で示した $n = 3$ の 2 進デコーダにイネーブル信号 $enable$ を追加する．$enable = 0$ のとき，出力 $o_0 \sim o_7$ はすべて 0 となり，$enable = 1$ のとき出力はイネーブル信号を追加する前のデコーダと等しい．$n = 3$ のイネーブル付き 2 進デコーダの真理値表を示し，AND, NOT ゲートを用いて設計せよ．

2 $n = 2$ のイネーブル付き 2 進デコーダと $n = 3$ のイネーブル付き 2 進デコーダがある．これらを組み合わせて $n = 5$ のイネーブル付き 2 進デコーダを構成せよ．

3 4 入力マルチプレクサを用いて 16 入力マルチプレクサを構成せよ．

演習問題 1

- 図8·1 で示した n = 3 の 2 進デコーダにイネーブル信号 enable を追加する．enable = 0 のとき，出力o_0 ~ o_7 はすべて0 となり，enable = 1 のとき出力はイネーブル信号を追加する前のデコーダと等しい．n = 3 のイネーブル付き 2 進デコーダの真理値表を示し，AND, NOT ゲートを用いて設計せよ.

表 8・1　2 進デコーダ ($n = 3$) の真理値表

i_2	i_1	i_0	o_7	o_6	o_5	o_4	o_3	o_2	o_1	o_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

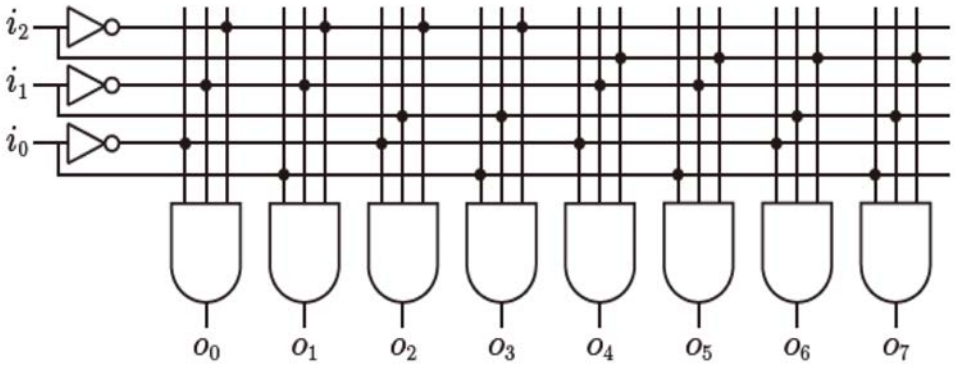


図 8・1　2 進デコーダの回路図

考慮時間

- 5分間程度で問題を解いてみてください．その間，ビデオを止めてください.

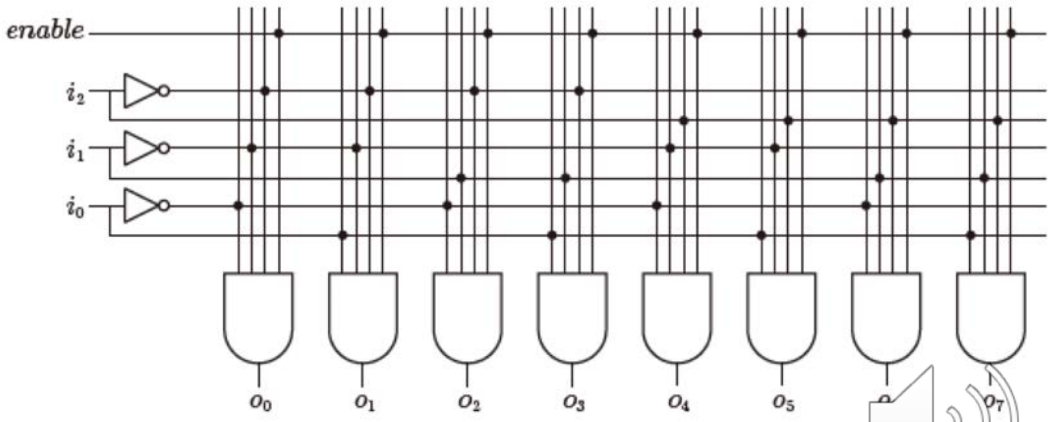
- この頁は30秒程度で次の頁に移行します.

演習問題１解答

- 図8·1 で示した n = 3 の 2 進デコーダにイネーブル信号 enable を追加する．enable = 0 のとき，出力 o0 ~ o7 はすべて0 となり，enable = 1 のとき出力はイネーブル信号を追加する前のデコーダと等しい．n = 3 のイネーブル付き 2 進デコーダの真理値表を示し，AND, NOT ゲートを用いて設計せよ．
- 各o_iの論理積回路にenable信号をさらに論理積すれば良い

解表 8・1

$enable$	i_2	i_1	i_0	o_7	o_6	o_5	o_4	o_3	o_2	o_1	o_0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0



解図 8・1

OSAKA UNIVERSITY

演習問題 2

- n = 2 のイネーブル付き 2 進デコーダと n = 3 のイネーブル付き 2 進デコーダがある．これらを組み合わせて n = 5 のイネーブル付き 2 進デコーダを構成せよ.
- 5 ビットのデータを $i = (i_4,i_3,i_2,i_1,i_0)$ で考えてみて下さい.

考慮時間

- 5分間程度で問題を解いてみてください．その間，ビデオを止めてください.

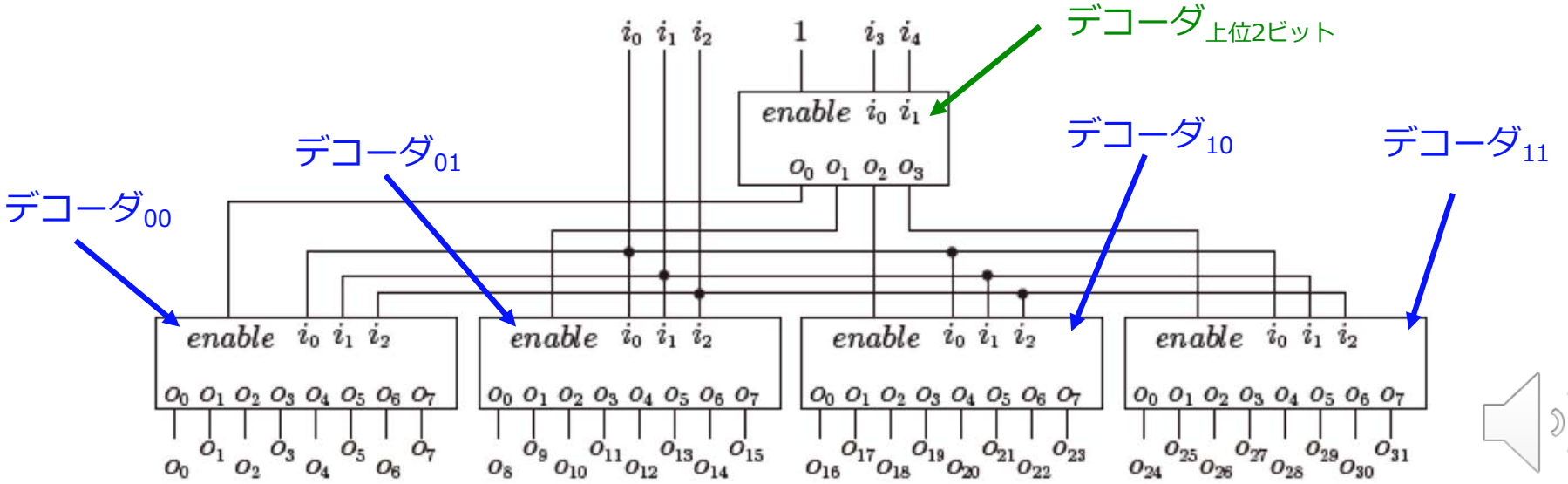
- この頁は30秒程度で次の頁に移行します.

演習問題２解答

- n = 2 のイネーブル付き 2 進デコーダと n = 3 のイネーブル付き 2 進デコーダを組み合わせて n = 5 のイネーブル付き 2 進デコーダを構成せよ.
- 5 ビットのデータを i = (i_4,i_3,i_2,i_1,i_0) で考えてみて下さい.

（解答）

- 0～7は(0,0,i_2,i_1,i_0)，8～15は(0,1,i_2,i_1,i_0)，16～23は(1,0,i_2,i_1,i_0)，24～31は(1,1,i_2,i_1,i_0) なので，(i_4,i_3)=(0,0)のときデコーダ$_{00}$ のenable信号が 1 になるようにn = 2 のイネーブル付き 2 進デコーダ デコーダ$_{上位2ビット}$を作れば良い．(i_4,i_3)=(0,1)ならデコーダ$_{01}$ のenable信号が 1 になる.



演習問題３

- ４ 入力マルチプレクサを用いて 16 入力マルチプレクサを構成せよ.

考慮時間

- 5分間程度で問題を解いてみてください．その間，ビデオを止めてください.

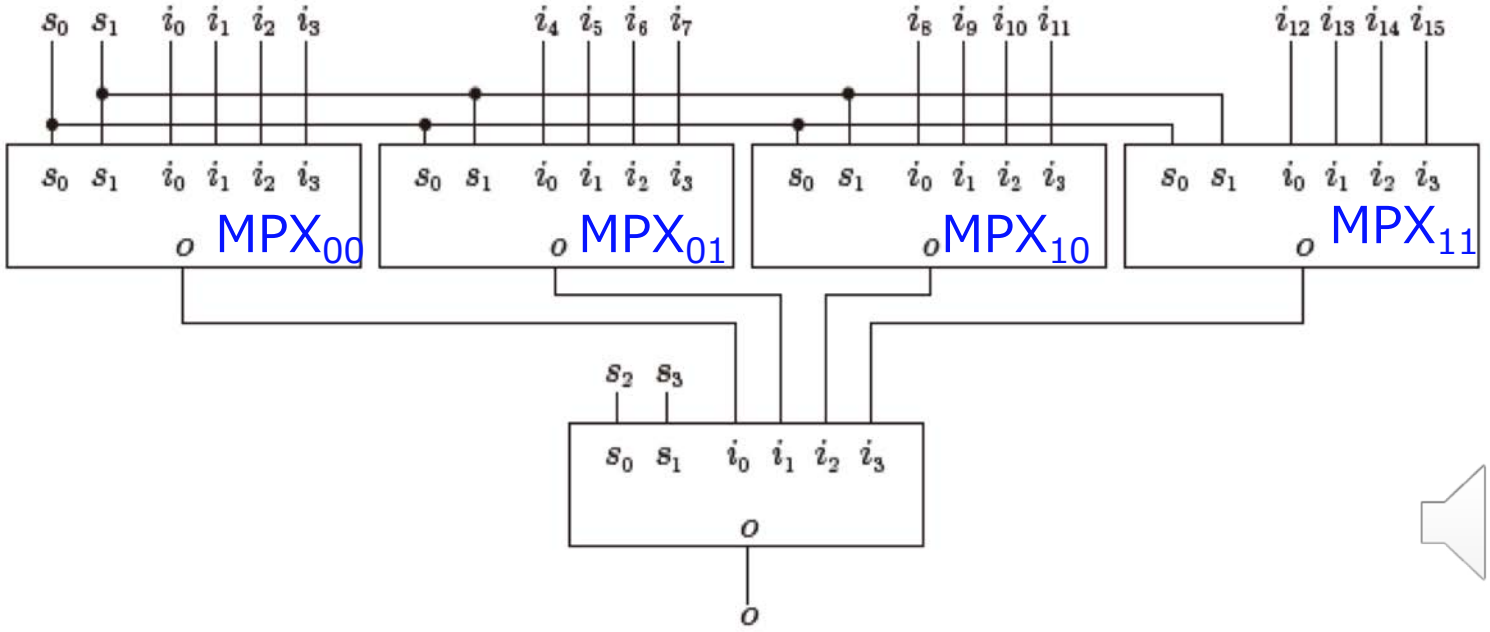
- この頁は30秒程度で次の頁に移行します.

演習問題３解答

- ４入力マルチプレクサ(MPX)を用いて 16 入力MPXを構成せよ.

(解答)

- ４ビットの制御入力を s =(s_3,s_2,s_1,s_0)とし，s の値によって i_{15}～i_0が選択されるとする．下記の2段回路を作れば，上段の回路で例えば (s_1,s_0) = (0,1)なら４つのMPXが i_1,i_5,i_9,i_{13} を選択する．次に下段のMPXの (s_1,s_0) に (s_3,s_2) の値を入力すれば，その値に従って i_1,i_5,i_9,i_{13} の一つが選択される（例えば(s_3,s_2)= (1,0) ならi_9が選ばれ， o = i_9 になる）.



Column

プログラマブル論理回路

- 専用集積回路よりも性能が劣るものの，製造後にユーザの手元で論理機能が定義変更できるプログラマブル集積回路も存在する．代表的なプログラマブル集積回路としてPLA（programmable logic array）とFPGA（field programmable gate array）がある．
  - PLA: PLA は積和形の論理関数の実現に適したプログラマブル論理回路である．図8·6に PLA の概略図を示す．AND プレーンに入力論理変数とその否定を縦配線で入力し，交点を選択的に接続することで各横配線に積項を実現する．OR プレーンでは，AND プレーンで生成した積項配線との交点を選択的に接続することで各縦配線に所望の積和形論理関数を実現する．

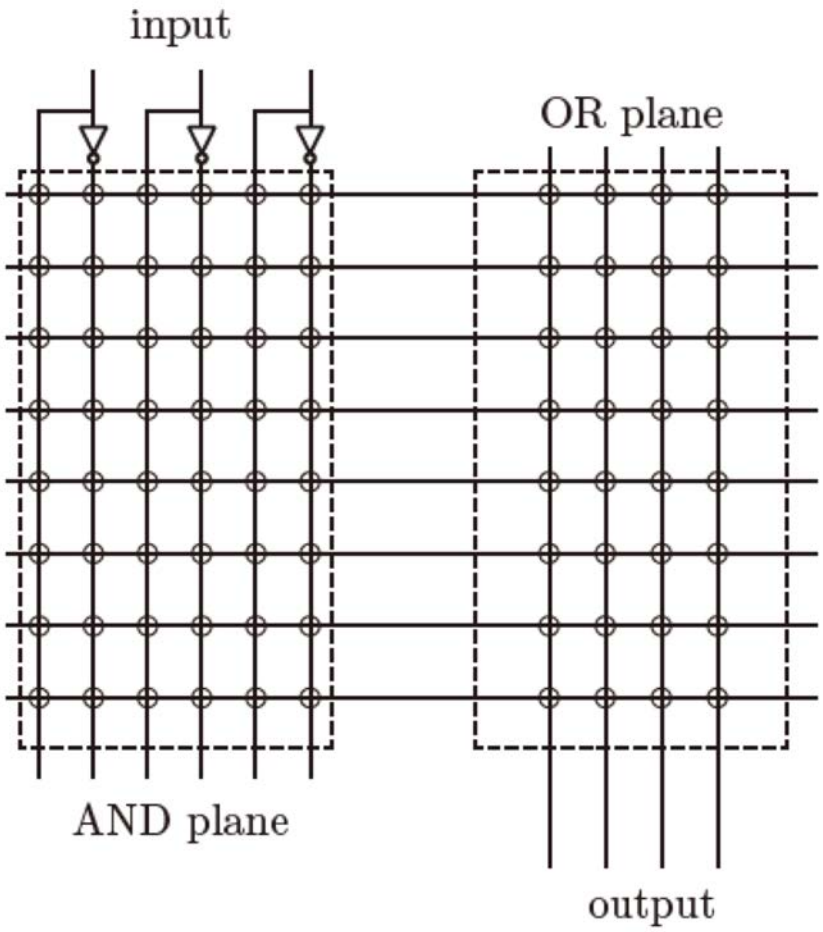
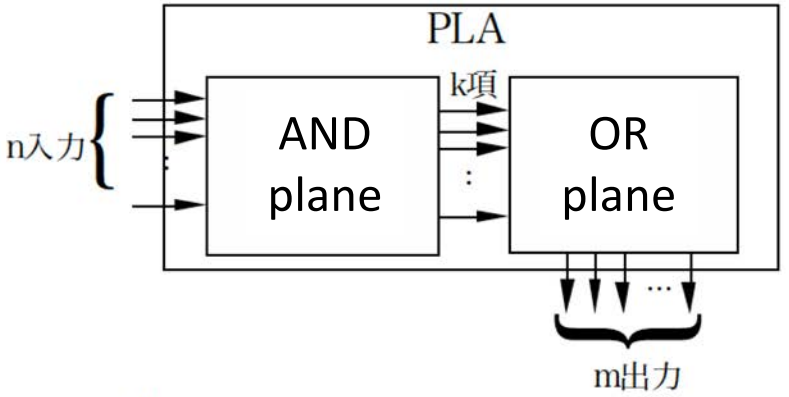
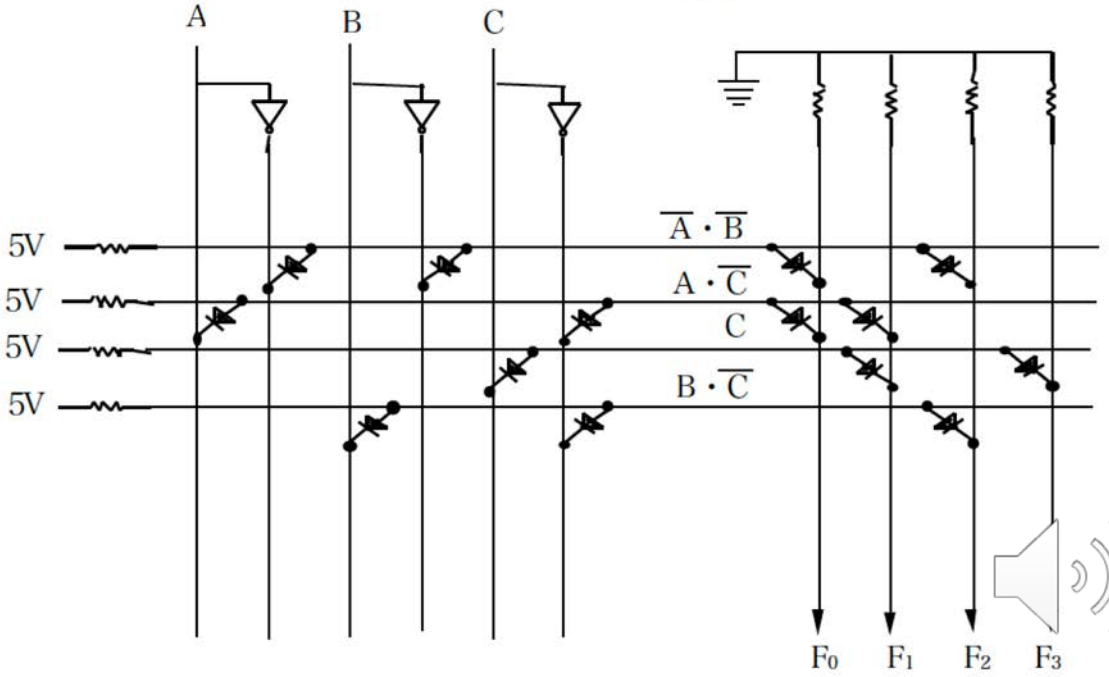


図 8・6 PLA 回路

PLA (Programmable Logic Array)

- 横軸の5Vに繋がる抵抗は抵抗値の大きな抵抗をセットする.
- 右図のようにAND plane の交点に片方向にしか電流が流れないダイオードを繋ぐと，繋いだダイオードの両方に電流が流れない限り，AND plane の下側に電流が流れていくので，OR Arrayの電圧が 0 になる．両方に電流が流れると，下側に電流が流れなくなるので，横軸の電圧は 5V のままになり，ANDが実現する



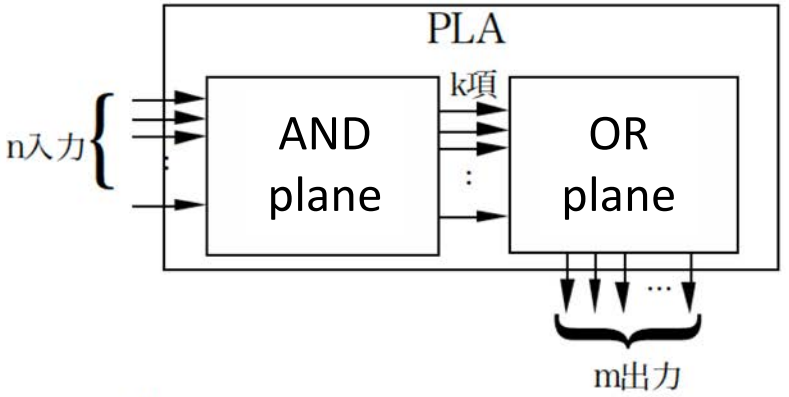


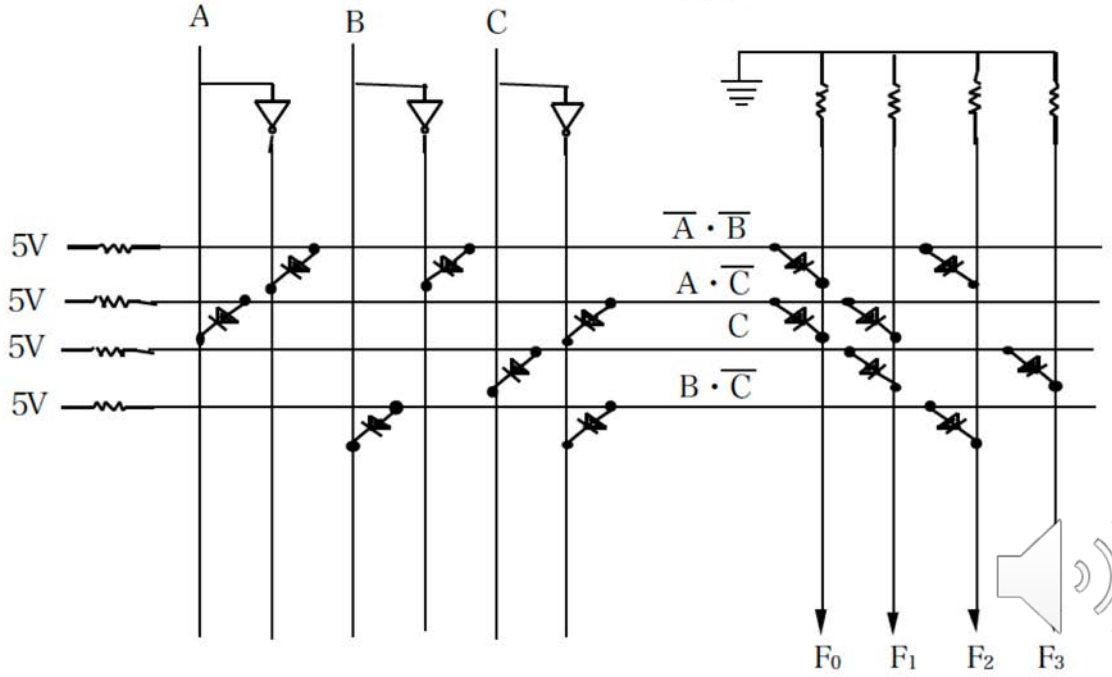
PLA (OR plane)

- 右図では，一番上の横軸が $\overline{A}\cdot\overline{B}$ に相当し，２番めの横軸が $A\cdot\overline{C}$ に相当する.
- これらのいずれかに電流が流れると，F_0に電流が流れる．結果

  $F_0 = \overline{A}\cdot\overline{B} \vee A\cdot\overline{C}$

  のOR回路が実現できる.





Column

プログラマブル論理回路

- FPGA: FPGA では 図8･7 に示したルックアップテーブル回路と呼ばれる回路を用いて論理関数を実現する.
- ルックアップテーブルは真理値表を直接的に実現する回路である．各最小項の値(s_0 ~ s_7) を記憶素子に格納し，入力信号 (x_0, x_1, x_2) の値に応じて該当する最小項の値をセレクタで選択して y に出力する．図では 3 変数の論理関数を実現するルックアップテーブル回路であるため，8 個の 1 ビットの記憶素子と，8 入力マルチプレクサを用いて実現されている.

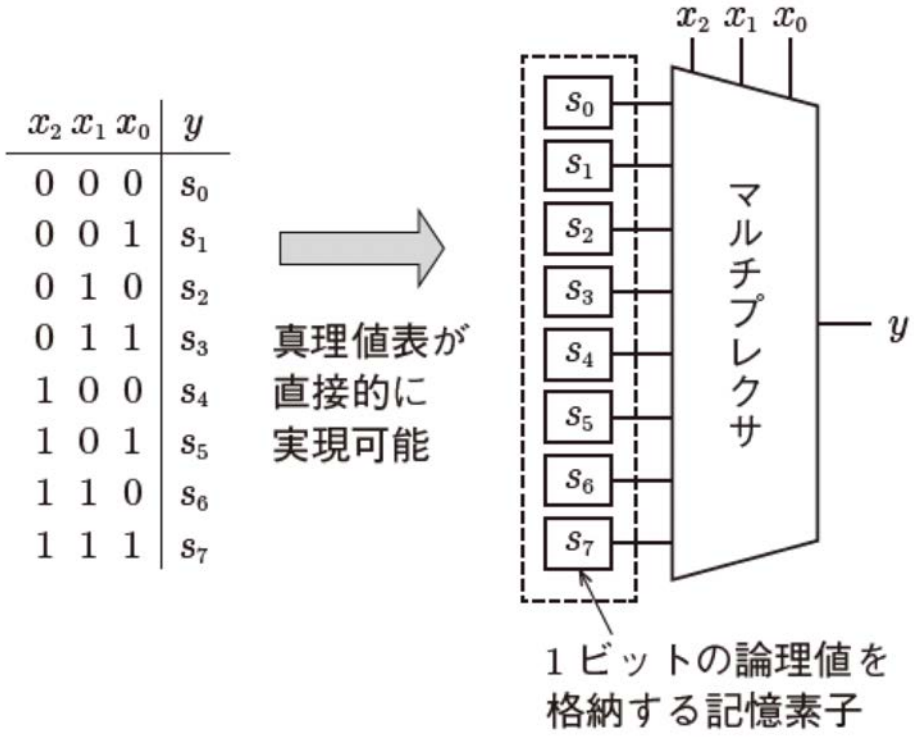


図 8・7 ルックアップテーブル回路

５回目の授業のレポート課題

5回目の授業のレポート課題

レポート課題

（課題５）

- 3変数論理関数 $f_1(x_2,x_1,x_0)$ は (x_2,x_1,x_0) を３ビットの２進数（x_2が上位ビット，x_0が下位ビット）と見なした際に 1,4,5,7のとき $f_1=1$、そうでないとき $f_1=0$ とする．また，$f_2(x_2,x_1,x_0)$は、1,2,3 のとき $f_2=1$、そうでないとき $f_2=0$ とする．
  1. f_1，f_2および $f_1 \cdot f_2$ の最簡積和形をそれぞれ求めよ。
  2. 上の1.で求めた主項を用いて、 $f_1(x_2,x_1,x_0)$， $f_2(x_2,x_1,x_0)$を同時に実現する最簡多出力論理関数を求めよ。
  - カルノー図に主項を明記して， f_1，f_2および $f_1 \cdot f_2$ の最簡積和形をどのように求めたかや，なぜ解答の関数が最簡多出力論理関数になるかの理由を明記してください．
- 提出先：wordやpower pointなどで電子的に作成するか，紙に書いた解答をスマホで写真を取ったりスキャナーなどで読み取り、pdf や jpeg, gif などの形式で電子化して CLE にアップして下さい．
- 締切：11月10日(火) 23:59（次の授業の前日迄）．
- ６回目の授業は 11月11日(水) の15:10 から実施します．授業のビデオは当日の12:30以降に視聴できるようになります．

OSAKA UNIVERSITY

５回目の授業終了

授業終了

皆さん
レポート提出してくださいね！