

電気工学通論Ⅱ (4)

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

- ・ スケジュール
- ・ 組合せ回路の構成法
- ・ 組合せ回路の実例

はじめに

■ 本講義の目的

- デ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. デジタル回路入門

デジタルとアナログ、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、アドレッシング、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**

3. 組合せ回路の構成法(その2)

■ 復習 ー組合せ回路の簡単化ー

ー ブール代数による方法

ブール代数の規則を利用して簡単化する。特に「論理的隣接性」によって簡単化する。

ー カルノー図による方法

1. カルノー図の上で、隣接した「1」を探し、これをループで囲む。ループは縦・横ともに2のべきの大きさをもつものとする。
2. ループの大きさは(重なってもいいから)なるべく大きくし、ループの数はなるべく少なくする
3. 2.で得られたループにしたがって、回路を記述する

例題1： フィボナッチ数(カルノー図)

- 問題： 4桁の2進数がフィボナッチ数であれば1を返す組合せ回路を書け（前回課題）
- カルノー図による解
 - 加法標準形は $\Sigma(0001, 0010, 0011, 0101, 1000, 1101)$
 - カルノー図

| XY \ ZW | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00 | | 1 | 1 | 1 |
| 01 | | 1 | | |
| 11 | | 1 | | |
| 10 | 1 | | | |

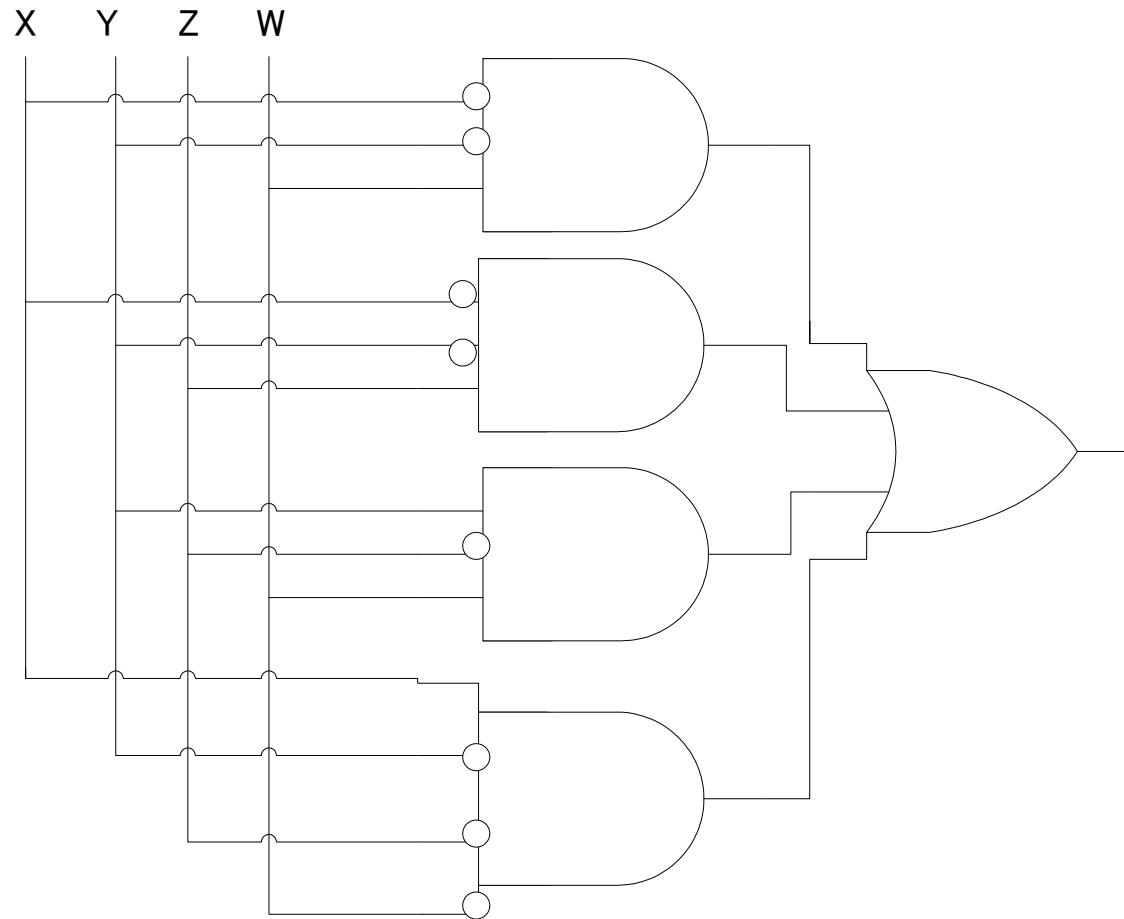
■ 解答

$$\begin{aligned} & X \cdot Y \cdot W + \bar{X} \cdot \bar{Y} \cdot Z + Y \cdot \bar{Z} \cdot W + X \cdot \bar{Y} \cdot \bar{Z} \cdot \bar{W} \\ & \bar{X} \cdot \bar{Z} \cdot W + \bar{X} \cdot Y \cdot Z + Y \cdot \bar{Z} \cdot W + X \cdot \bar{Y} \cdot \bar{Z} \cdot \bar{W} \end{aligned}$$

主項と必須主項

- 主項 (prime applicant)
 - 論理関数によって覆われる最大の積項
 - 簡単化したときに残る「候補」
 - 前ページの○●○
- 必須主項 (essential prime applicant)
 - 最小項を覆う唯一の主項
 - 簡単化の最後に「必ず」残る
 - 前ページの○
- 簡単化
 - 必須主項○を求める
 - 必須主項に含まれない最小項について、これを含む主項をそれぞれ1つずつ選ぶ(●または○)

フィボナッチ数判定回路 (MIL記法)



カルノー図の問題点

1. 入力数が5(6)までしか使えない
 2. 計算機でアルゴリズムック(Algorithmic)に処理しにくい
 3. 複数入力・複数出力の単純化ができない
 - 途中まで同じ回路を利用できる場合などに対応できない
- 解決策: **クワイン・マクラスキーの方法**
- 教科書を読んでみてください

4. 組合せ回路の実例

■ 内容

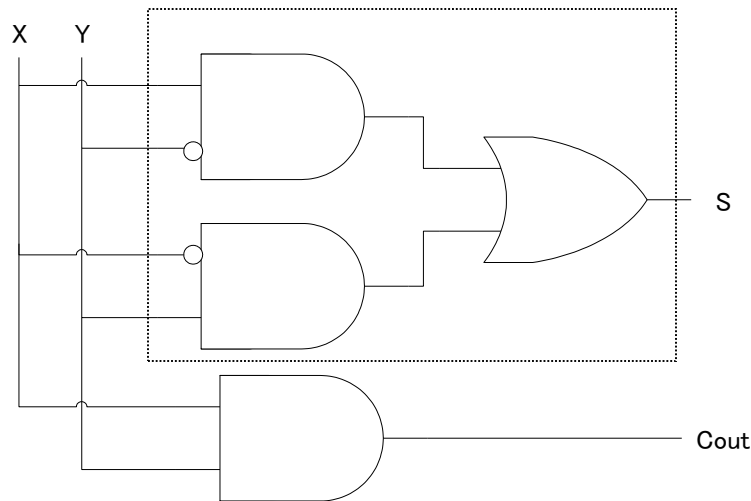
- 加算器
- 減算器
- ALU
- デコーダ
- セレクタ
- PLA


4.1 加算器

■ 半加算器(half adder)

- 入力A、Bを受け、その桁の和Sと、桁上がりC(Carry, キャリ)を作り出す2進1桁の加算回路

| X | Y | S | C |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |



 : XOR

$$S = \bar{X}Y + X\bar{Y} = X \oplus Y, \quad \text{Cout} = XY$$

全加算器

■ 1ビット全加算器(full adder)

- 入力A、Bおよび一つ下の桁からの桁上がりCinを受け、その桁の和Sと、桁上がりCoutを作り出す2進1桁の加算回路

| X | Y | Cin | S | Cout |
|---|---|-----|---|------|
| 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 |

真理値表

| Cin \ XY | 00 | 01 | 11 | 10 |
|----------|----|----|----|----|
| 0 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |

S

| Cin \ XY | 00 | 01 | 11 | 10 |
|----------|----|----|----|----|
| 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 |

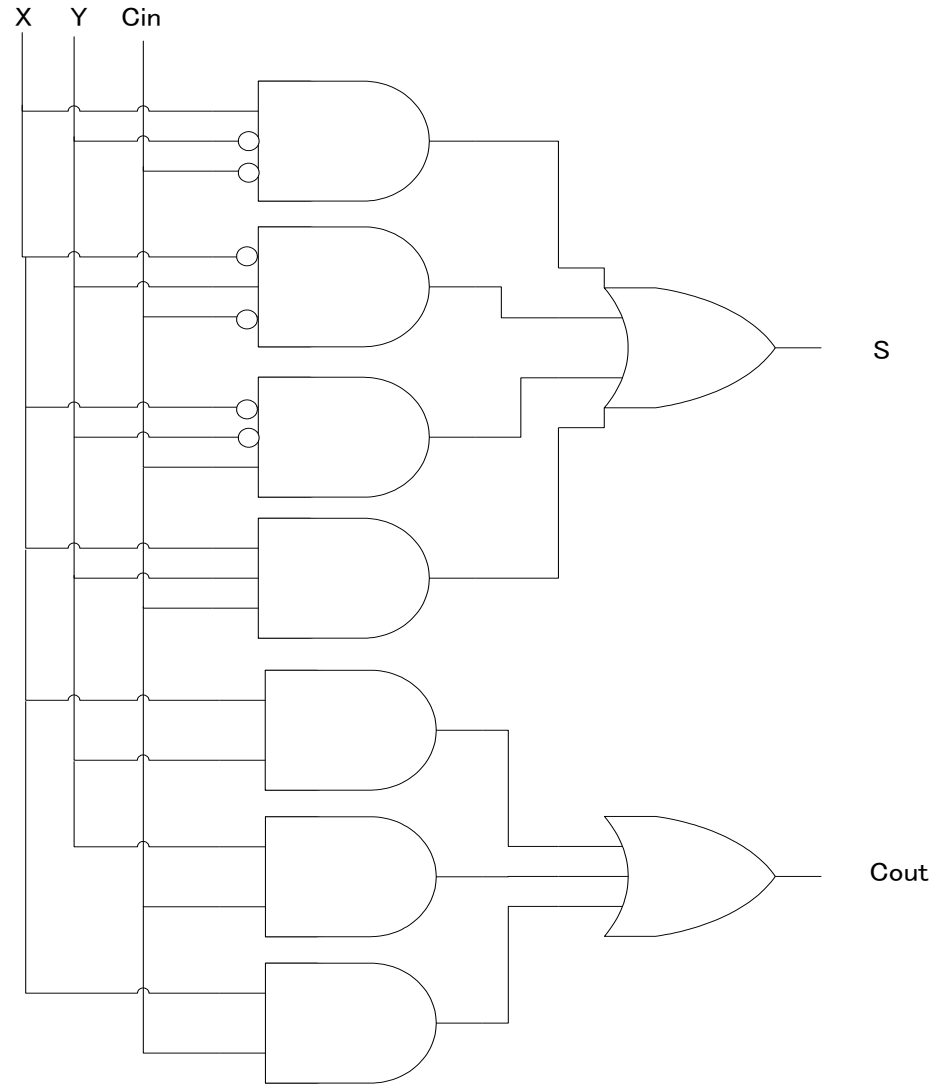
Cout

カルノー図

$$\begin{aligned}
 S &= XYC_{in} + X\bar{Y}\bar{C}_{in} + \bar{X}Y\bar{C}_{in} + \bar{X}\bar{Y}C_{in} \\
 &= X \oplus Y \oplus C_{in}
 \end{aligned}$$

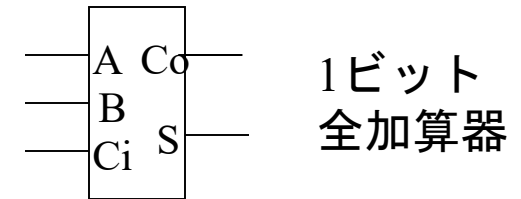
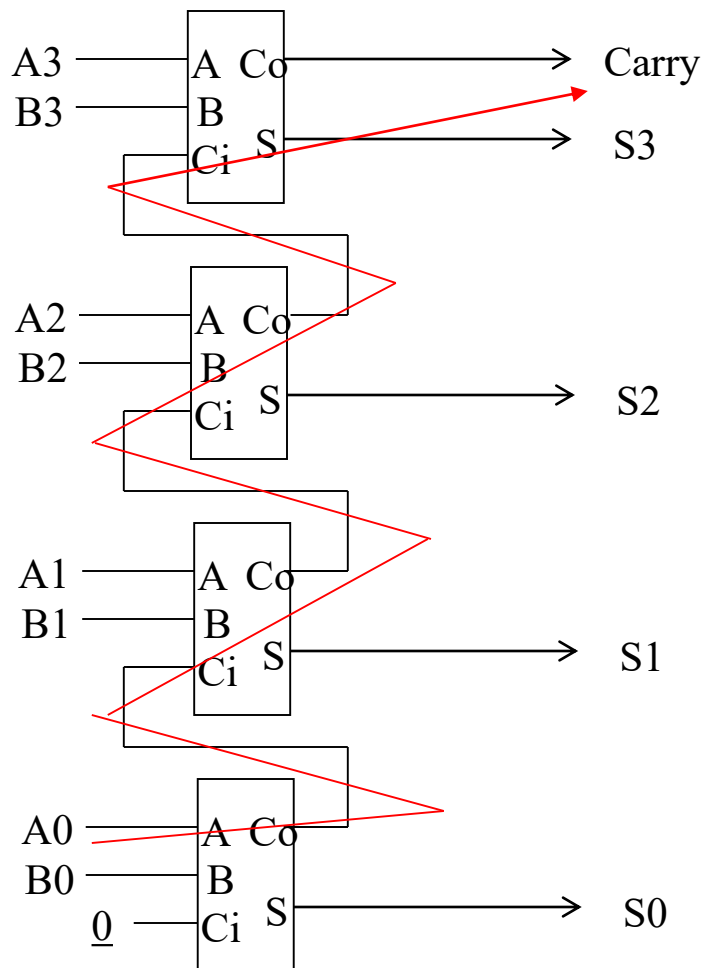
$$Cout = XY + YC_{in} + XC_{in}$$

全加算器の回路図 (MIL記法)



Nビット全加算器(1)

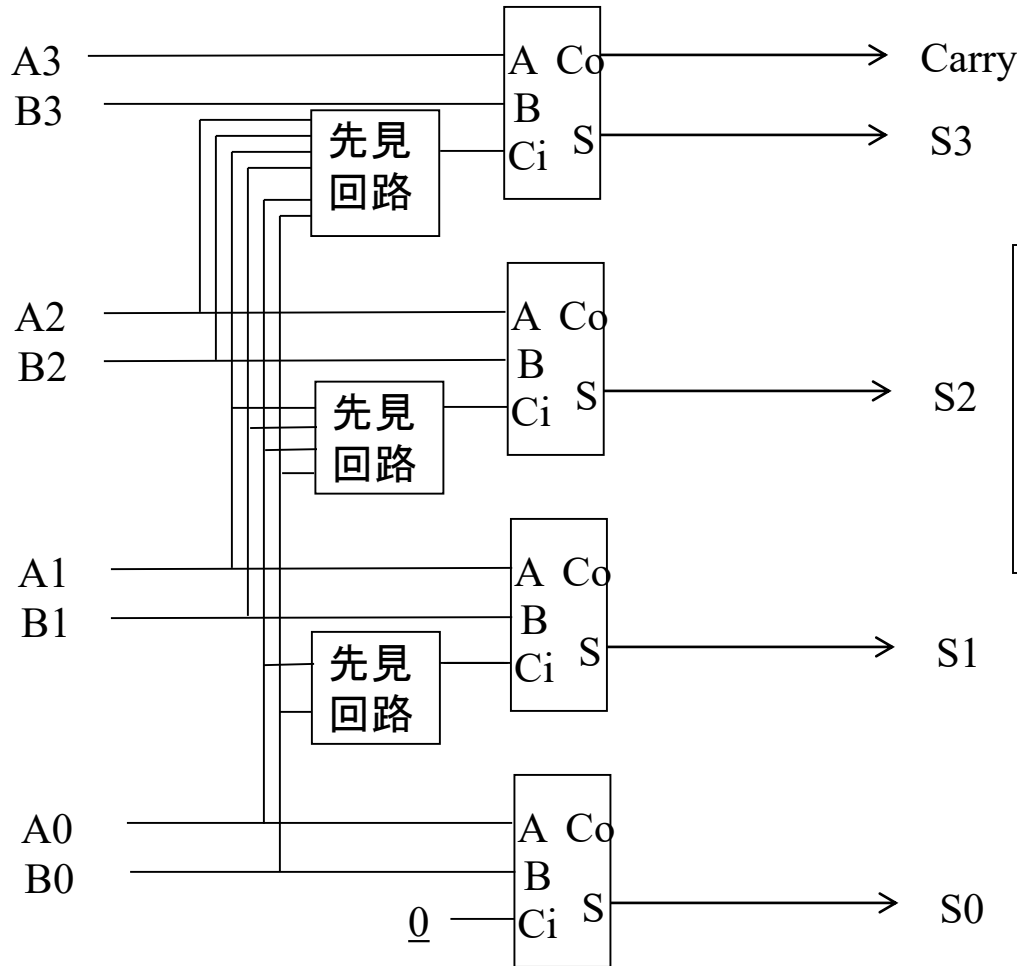
■ リプルキャリ方式



回路は簡単だが、桁上り
信号（Carry）の伝播に
時間がかかってしまう

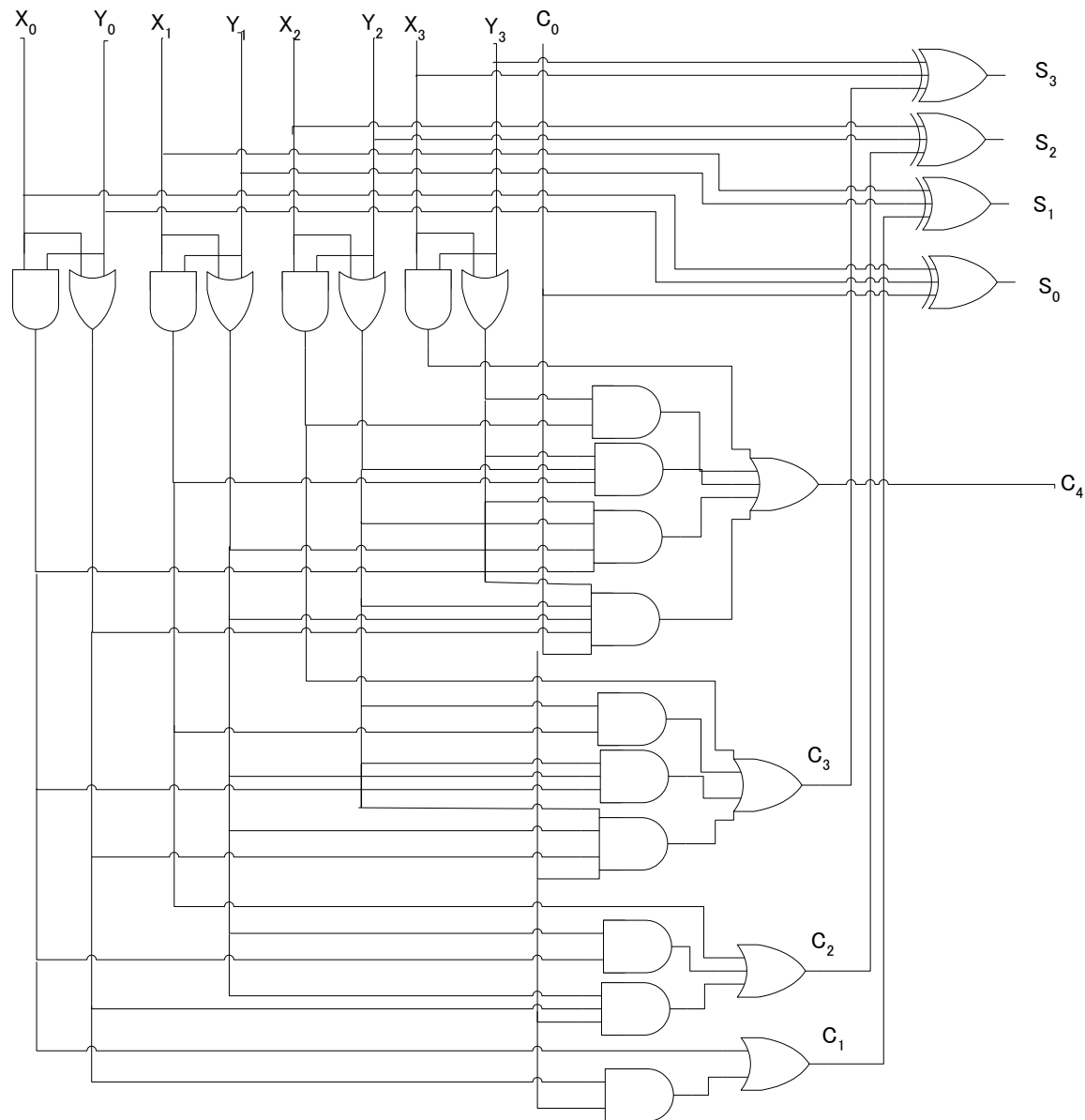
Nビット全加算器(2)

■ キャリルックahead方式



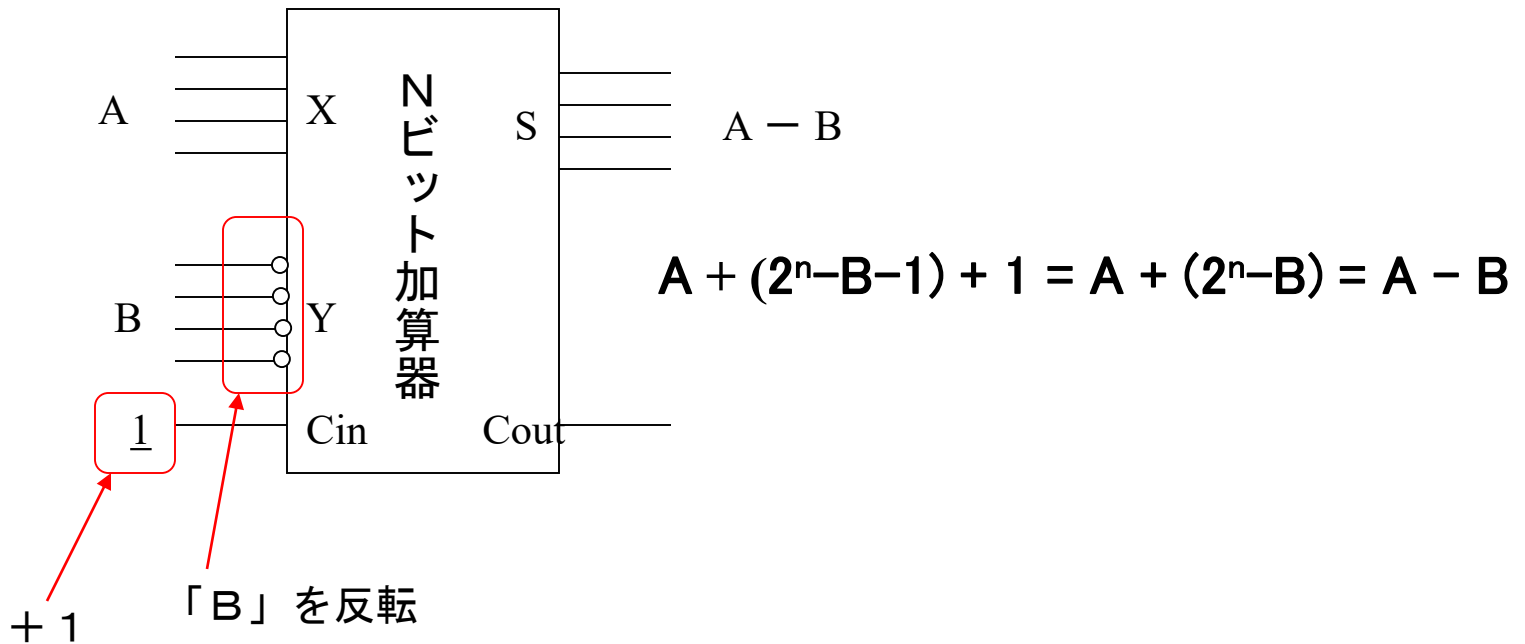
桁上げを先取りすることで、遅延を小さくする。上の桁ほど回路は複雑になる

4ビットキャリルックahead加算器



4.3 減算器

- $A - B$: B の全ビットを反転して1を加えたものを、 A に加える



1 を加える ← 加算器の最下位の桁のCarryInに 1 を入れる

問題

1. 論理関数 $\Sigma(1, 5, 6, 7, 9, 13, 14)$ をカルノー図によって簡単化せよ。
2. キャリルックアヘッド加算器の回路の一般的な設計法を示せ
3. 4ビット加減算器の回路を書け。ただし、4ビット加算器は既存のものとして使ってよい
 - 入力: $A_3-A_0, B_3-B_0, A/S$
 - 出力: $S: \begin{matrix} A/S=1 \text{ のとき } X+Y \\ A/S=0 \text{ のとき } X-Y \end{matrix}$
Carry

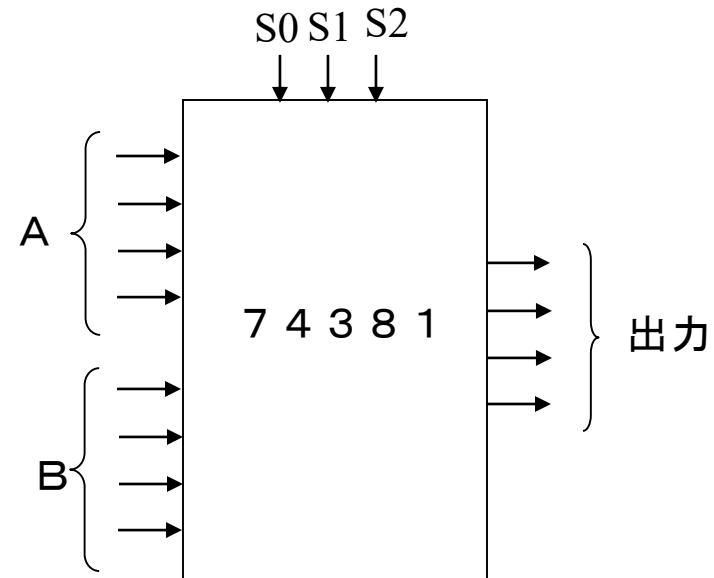
4.4 ALU

■ ALU = 算術論理ユニット(Arithmetic Logic Unit)

制御信号によって、どのような演算を行うかを決める組合せ回路

例. 74381 (という IC) by TI

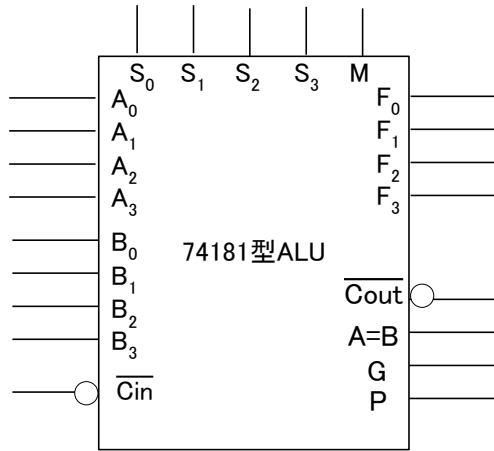
| 制御信号 | | | 出力 |
|------|----|----|--------------|
| S2 | S1 | S0 | |
| 0 | 0 | 0 | 0000 |
| 0 | 0 | 1 | B-A |
| 0 | 1 | 0 | A-B |
| 0 | 1 | 1 | A+B |
| 1 | 0 | 0 | $A \oplus B$ |
| 1 | 0 | 1 | A OR B |
| 1 | 1 | 0 | A AND B |
| 1 | 1 | 1 | 1111 |



例2. 74181

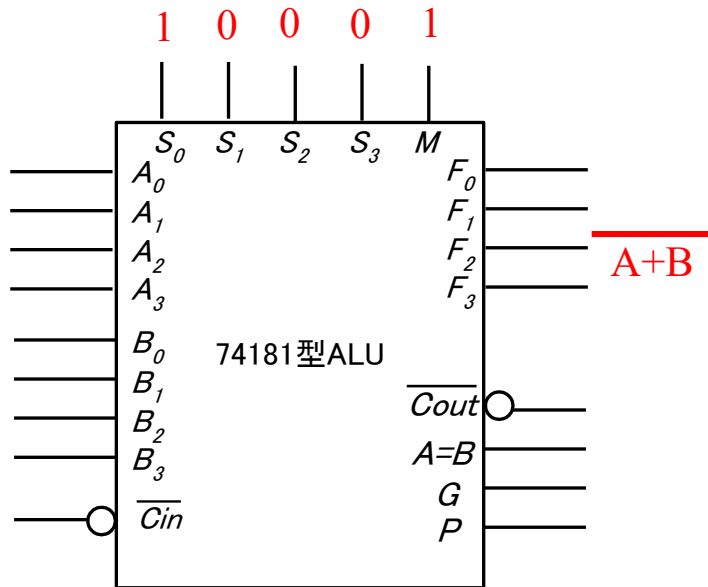
A+B, A-B, \overline{A} , \overline{B} , $A \oplus B$, A+1, A-1, A OR B, A AND B, 1111, 0000, A, B,

74181型ALU



| 制御信号 S ₃ S ₂ S ₁ S ₀ | M=1: 論理演算 | M=0: 算術演算 | |
|---|--------------------------|---------------------------------|--|
| | | $\overline{\text{Cin}}=0$ | $\overline{\text{Cin}}=1$ |
| 0 0 0 0 | $F=\overline{A}$ | F=A | F=A PLUS 1 |
| 0 0 0 1 | $F=\overline{A + B}$ | F=A+B | F=(A+B) PLUS 1 |
| 0 0 1 0 | $F=\overline{A \cdot B}$ | F=A + \overline{B} | F=(A + \overline{B}) PLUS 1 |
| 0 0 1 1 | F=0 | F= 1111 | F=ZERO |
| 0 1 0 0 | $F=\overline{A \cdot B}$ | F=A PLUS A · \overline{B} | F=A PLUS A · \overline{B} PLUS 1 |
| 0 1 0 1 | $F=\overline{B}$ | F=(A+B) PLUS A · \overline{B} | F=(A+B) PLUS A · \overline{B} PLUS 1 |
| 0 1 1 0 | $F=A \oplus B$ | F=A MINUS B MINUS 1 | F=A MINUS B |
| 0 1 1 1 | $F=A \cdot \overline{B}$ | F=A · \overline{B} MINUS 1 | F=A · \overline{B} |
| 1 0 0 0 | $F=\overline{A} + B$ | F=A PLUS A · B | F=A PLUS A · B PLUS 1 |
| 1 0 0 1 | $F=A \oplus B$ | F=A PLUS B | F= A PLUS B PLUS 1 |
| 1 0 1 0 | F=B | F=(A + \overline{B}) PLUS AB | F= (A + \overline{B}) PLUS A · B PLUS 1 |
| 1 0 1 1 | F=A · B | F=A · B MINUS 1 | F=A · B |
| 1 1 0 0 | F=1 | F=A PLUS A | F=A PLUS A PLUS 1 |
| 1 1 0 1 | $F=A + \overline{B}$ | F=(A+B) PLUS A | F=(A+B) PLUS A PLUS 1 |
| 1 1 1 0 | F=A + B | F=(A + \overline{B}) PLUS A | F=(A + \overline{B}) PLUS A PLUS 1 |
| 1 1 1 1 | F=A | F=A MINUS 1 | F=A |

ALU (算術論理演算ユニット)



©Shuichi Sakai

A=BはAとBが等しいときに1
G,Pはキャリルックアヘッド用

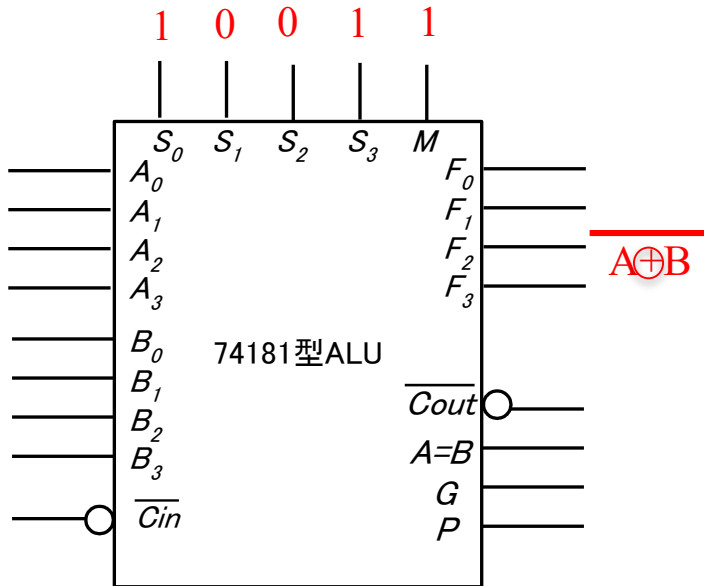


| 制御信号 S ₃ S ₂ S ₁ S ₀ | M=1: 論理演算 | M=0: 算術演算 | |
|---|---------------------------|--|--|
| | | $\overline{Cin}=0$ | $\overline{Cin}=1$ |
| 0 0 0 0 | $F=\overline{A}$ | $F=A$ | $F=A \text{ PLUS } 1$ |
| 0 0 0 1 | $F=\overline{A+B}$ | $F=A+B$ | $F=(A+B) \text{ PLUS } 1$ |
| 0 0 1 0 | $F=\overline{A \cdot B}$ | $F=A + \overline{B}$ | $F=(A+B) \text{ PLUS } 1$ |
| 0 0 1 1 | $F=0$ | $F=1111$ | $F=ZERO$ |
| 0 1 0 0 | $F=\overline{A \cdot B}$ | $F=A \text{ PLUS } A \cdot \overline{B}$ | $F=A \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$ |
| 0 1 0 1 | $F=\overline{B}$ | $F=(A+B) \text{ PLUS } A \cdot \overline{B}$ | $F=(A+B) \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$ |
| 0 1 1 0 | $F=A \oplus B$ | $F=A \text{ MINUS } B \text{ MINUS } 1$ | $F=A \text{ MINUS } B$ |
| 0 1 1 1 | $F=A \cdot \overline{B}$ | $F=A \cdot \overline{B} \text{ MINUS } 1$ | $F=A \cdot \overline{B}$ |
| 1 0 0 0 | $F=\overline{A+B}$ | $F=A \text{ PLUS } A \cdot B$ | $F=A \text{ PLUS } A \cdot B \text{ PLUS } 1$ |
| 1 0 0 1 | $F=\overline{A \oplus B}$ | $F=A \text{ PLUS } B$ | $F=A \text{ PLUS } B \text{ PLUS } 1$ |
| 1 0 1 0 | $F=B$ | $F=(A + \overline{B}) \text{ PLUS } AB$ | $F=(A + \overline{B}) \text{ PLUS } A \cdot B \text{ PLUS } 1$ |
| 1 0 1 1 | $F=A \cdot B$ | $F=A \cdot B \text{ MINUS } 1$ | $F=A \cdot B$ |
| 1 1 0 0 | $F=1$ | $F=A \text{ PLUS } A$ | $F=A \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 0 1 | $F=A + \overline{B}$ | $F=(A+B) \text{ PLUS } A$ | $F=(A+B) \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 1 0 | $F=A + B$ | $F=(A + \overline{B}) \text{ PLUS } A$ | $F=(A + \overline{B}) \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 1 1 | $F=A$ | $F=A \text{ MINUS } 1$ | $F=A$ |

(b) 動作: 重要なものは太字とした

図 1.10 74181 型 ALU (4 ビット) (前ページからの続き)

ALU (算術論理演算ユニット)



©Shuichi Sakai

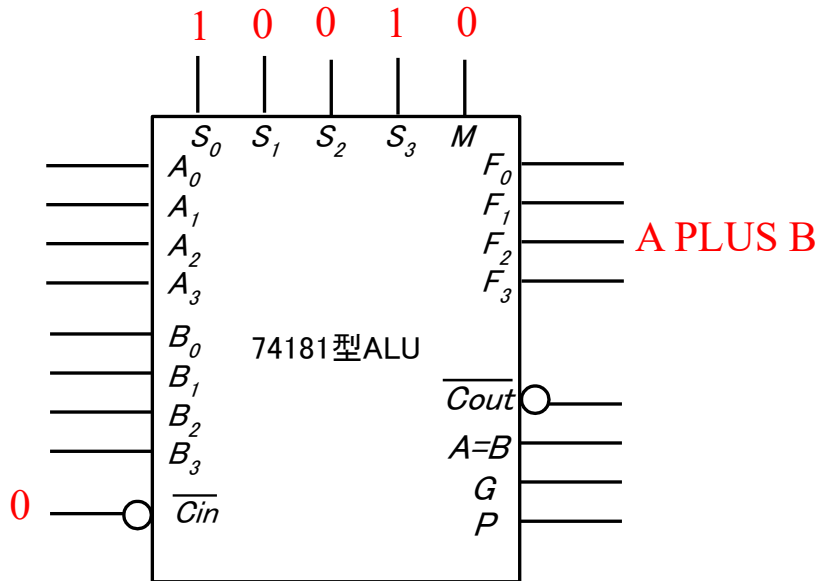
A=BはAtoBが等しいときに1
G,Pはキャリルックアヘッド用

| 制御信号 S ₃ S ₂ S ₁ S ₀ | M=1: 論理演算 | M=0: 算術演算 | |
|---|---------------------------|--|--|
| | | $\overline{Cin}=0$ | $\overline{Cin}=1$ |
| 0 0 0 0 | $F=\overline{A}$ | $F=A$ | $F=A \text{ PLUS } 1$ |
| 0 0 0 1 | $F=\overline{A+B}$ | $F=A+B$ | $F=(A+B) \text{ PLUS } 1$ |
| 0 0 1 0 | $F=\overline{A \cdot B}$ | $F=A + \overline{B}$ | $F=(A+B) \text{ PLUS } 1$ |
| 0 0 1 1 | $F=0$ | $F=1111$ | $F=ZERO$ |
| 0 1 0 0 | $F=\overline{A \cdot B}$ | $F=A \text{ PLUS } A \cdot \overline{B}$ | $F=A \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$ |
| 0 1 0 1 | $F=\overline{B}$ | $F=(A+B) \text{ PLUS } A \cdot \overline{B}$ | $F=(A+B) \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$ |
| 0 1 1 0 | $F=A \oplus B$ | $F=A \text{ MINUS } B \text{ MINUS } 1$ | $F=A \text{ MINUS } B$ |
| 0 1 1 1 | $F=A \cdot \overline{B}$ | $F=A \cdot \overline{B} \text{ MINUS } 1$ | $F=A \cdot \overline{B}$ |
| 1 0 0 0 | $F=\overline{A+B}$ | $F=A \text{ PLUS } A \cdot B$ | $F=A \text{ PLUS } A \cdot B \text{ PLUS } 1$ |
| 1 0 0 1 | $F=\overline{A \oplus B}$ | $F=A \text{ PLUS } B$ | $F=A \text{ PLUS } B \text{ PLUS } 1$ |
| 1 0 1 0 | $F=B$ | $F=(A + \overline{B}) \text{ PLUS } AB$ | $F=(A + \overline{B}) \text{ PLUS } A \cdot B \text{ PLUS } 1$ |
| 1 0 1 1 | $F=A \cdot B$ | $F=A \cdot B \text{ MINUS } 1$ | $F=A \cdot B$ |
| 1 1 0 0 | $F=1$ | $F=A \text{ PLUS } A$ | $F=A \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 0 1 | $F=A + \overline{B}$ | $F=(A+B) \text{ PLUS } A$ | $F=(A+B) \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 1 0 | $F=A + B$ | $F=(A + \overline{B}) \text{ PLUS } A$ | $F=(A + \overline{B}) \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 1 1 | $F=A$ | $F=A \text{ MINUS } 1$ | $F=A$ |

(b) 動作: 重要なものは太字とした

図 1.10 74181 型 ALU (4 ビット) (前ページからの続き)

ALU (算術論理演算ユニット)



©Shuichi Sakai

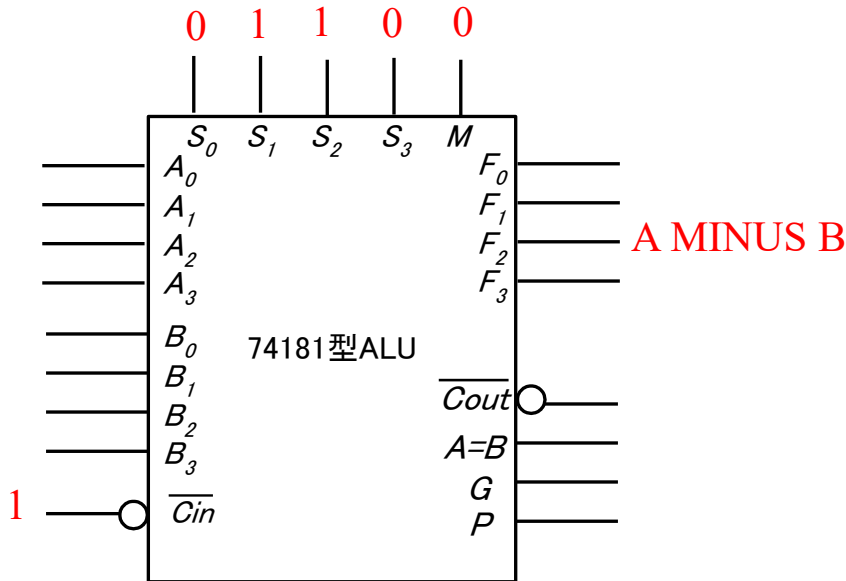
A=BはAtoBが等しいときに1
G,Pはキャリルックアヘッド用

| 制御信号 $S_3 S_2 S_1 S_0$ | $M=1$: 論理演算 | $M=0$: 算術演算 | |
|---------------------------|---------------------------|--|--|
| | | $\overline{Cin}=0$ | $\overline{Cin}=1$ |
| 0 0 0 0 | $F=\overline{A}$ | $F=A$ | $F=A \text{ PLUS } 1$ |
| 0 0 0 1 | $F=\overline{A+B}$ | $F=A+B$ | $F=(A+B) \text{ PLUS } 1$ |
| 0 0 1 0 | $F=\overline{A \cdot B}$ | $F=A + \overline{B}$ | $F=(A+B) \text{ PLUS } 1$ |
| 0 0 1 1 | $F=0$ | $F=1111$ | $F=ZERO$ |
| 0 1 0 0 | $F=\overline{A \cdot B}$ | $F=A \text{ PLUS } A \cdot \overline{B}$ | $F=A \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$ |
| 0 1 0 1 | $F=\overline{B}$ | $F=(A+B) \text{ PLUS } A \cdot \overline{B}$ | $F=(A+B) \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$ |
| 0 1 1 0 | $F=A \oplus B$ | $F=A \text{ MINUS } B \text{ MINUS } 1$ | $F=A \text{ MINUS } B$ |
| 0 1 1 1 | $F=A \cdot \overline{B}$ | $F=A \cdot \overline{B} \text{ MINUS } 1$ | $F=A \cdot \overline{B}$ |
| 1 0 0 0 | $F=\overline{A+B}$ | $F=A \text{ PLUS } A \cdot B$ | $F=A \text{ PLUS } A \cdot B \text{ PLUS } 1$ |
| 1 0 0 1 | $F=A \oplus \overline{B}$ | $F=A \text{ PLUS } B$ | $F=A \text{ PLUS } B \text{ PLUS } 1$ |
| 1 0 1 0 | $F=B$ | $F=(A + \overline{B}) \text{ PLUS } AB$ | $F=(A + \overline{B}) \text{ PLUS } A \cdot B \text{ PLUS } 1$ |
| 1 0 1 1 | $F=A \cdot B$ | $F=A \cdot B \text{ MINUS } 1$ | $F=A \cdot B$ |
| 1 1 0 0 | $F=1$ | $F=A \text{ PLUS } A$ | $F=A \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 0 1 | $F=A + \overline{B}$ | $F=(A+B) \text{ PLUS } A$ | $F=(A+B) \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 1 0 | $F=A + B$ | $F=(A + \overline{B}) \text{ PLUS } A$ | $F=(A + \overline{B}) \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 1 1 | $F=A$ | $F=A \text{ MINUS } 1$ | $F=A$ |

(b) 動作: 重要なものは太字とした

図 1.10 74181 型 ALU (4 ビット) (前ページからの続き)

ALU (算術論理演算ユニット)



©Shuichi Sakai

A=BはAtoBが等しいときに1
G,Pはキャリルックアヘッド用

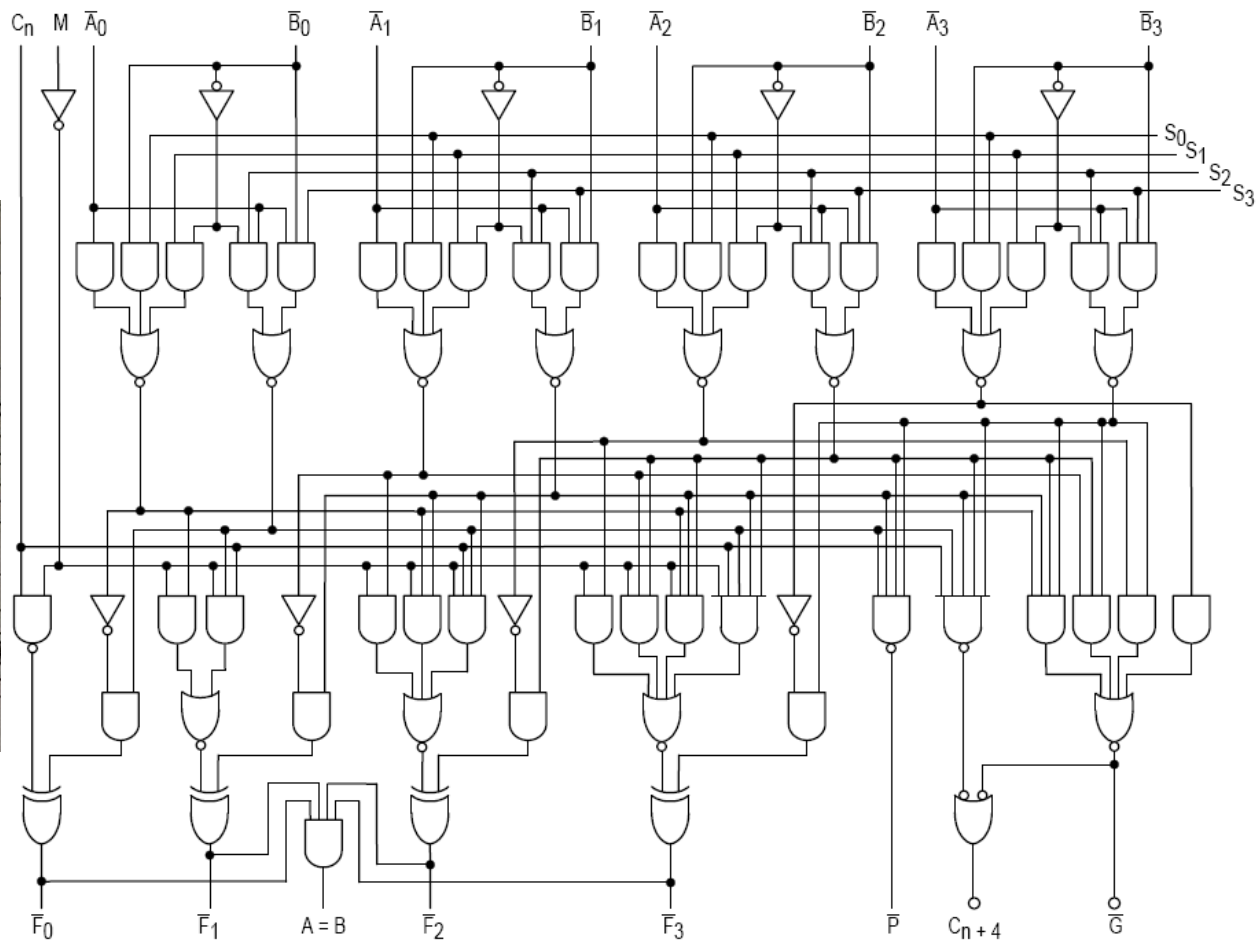
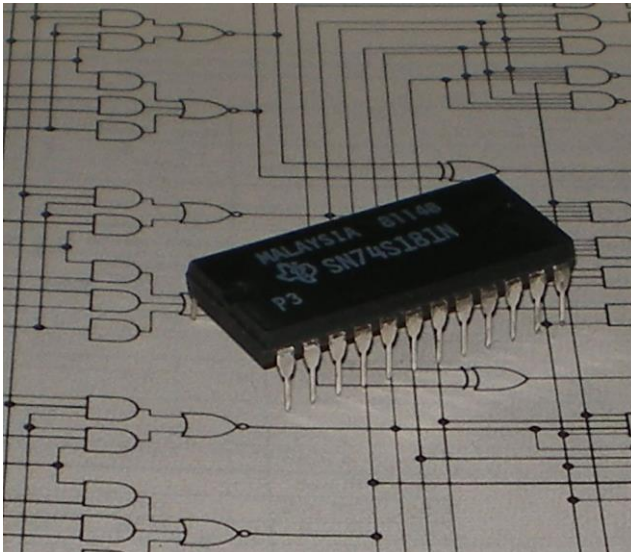
| 制御信号 $S_3 S_2 S_1 S_0$ | $M=1$: 論理演算 | $M=0$: 算術演算 | |
|---------------------------|---------------------------|--|--|
| | | $\overline{Cin}=0$ | $\overline{Cin}=1$ |
| 0 0 0 0 | $F=\overline{A}$ | $F=A$ | $F=A \text{ PLUS } 1$ |
| 0 0 0 1 | $F=\overline{A+B}$ | $F=A+B$ | $F=(A+B) \text{ PLUS } 1$ |
| 0 0 1 0 | $F=\overline{A \cdot B}$ | $F=A + \overline{B}$ | $F=(A+B) \text{ PLUS } 1$ |
| 0 0 1 1 | $F=0$ | $F=1111$ | $F=ZERO$ |
| 0 1 0 0 | $F=\overline{A \cdot B}$ | $F=A \text{ PLUS } A \cdot \overline{B}$ | $F=A \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$ |
| 0 1 0 1 | $F=\overline{B}$ | $F=(A+B) \text{ PLUS } A \cdot \overline{B}$ | $F=(A+B) \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$ |
| 0 1 1 0 | $F=A \oplus B$ | $F=A \text{ MINUS } B \text{ MINUS } 1$ | $F=A \text{ MINUS } B$ |
| 0 1 1 1 | $F=A \cdot \overline{B}$ | $F=A \cdot \overline{B} \text{ MINUS } 1$ | $F=A \cdot \overline{B}$ |
| 1 0 0 0 | $F=\overline{A+B}$ | $F=A \text{ PLUS } A \cdot B$ | $F=A \text{ PLUS } A \cdot B \text{ PLUS } 1$ |
| 1 0 0 1 | $F=A \oplus \overline{B}$ | $F=A \text{ PLUS } B$ | $F=A \text{ PLUS } B \text{ PLUS } 1$ |
| 1 0 1 0 | $F=B$ | $F=(A + \overline{B}) \text{ PLUS } AB$ | $F=(A + \overline{B}) \text{ PLUS } A \cdot B \text{ PLUS } 1$ |
| 1 0 1 1 | $F=A \cdot B$ | $F=A \cdot B \text{ MINUS } 1$ | $F=A \cdot B$ |
| 1 1 0 0 | $F=1$ | $F=A \text{ PLUS } A$ | $F=A \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 0 1 | $F=A + \overline{B}$ | $F=(A+B) \text{ PLUS } A$ | $F=(A+B) \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 1 0 | $F=A + B$ | $F=(A + \overline{B}) \text{ PLUS } A$ | $F=(A + \overline{B}) \text{ PLUS } A \text{ PLUS } 1$ |
| 1 1 1 1 | $F=A$ | $F=A \text{ MINUS } 1$ | $F=A$ |

(b) 動作: 重要なものは太字とした

図 1.10 74181 型 ALU (4 ビット) (前ページからの続き)

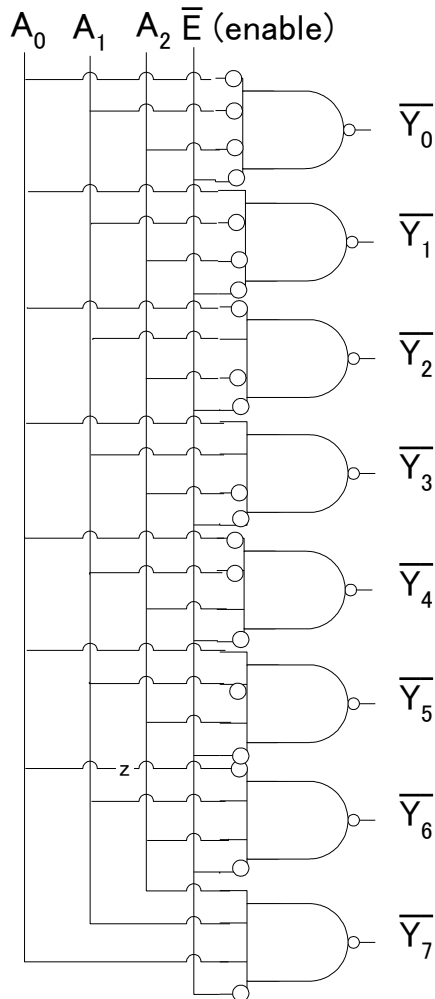
74181の回路図

- CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=168473>
- By 74s181 at English Wikipedia – Transferred from en.wikipedia to Commons by audriusa., Public Domain, <https://commons.wikimedia.org/w/index.php?curid=18363509>



4.5 デコーダ (複号器、decoder)

- 2進符号 i (n -bit) \rightarrow i 本目の信号線をonにする、という論理回路

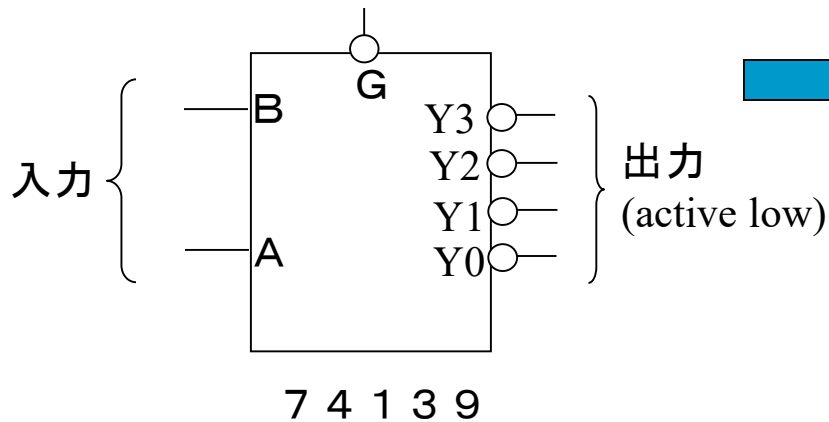


普通、出力はonのときに、0にする(active low)

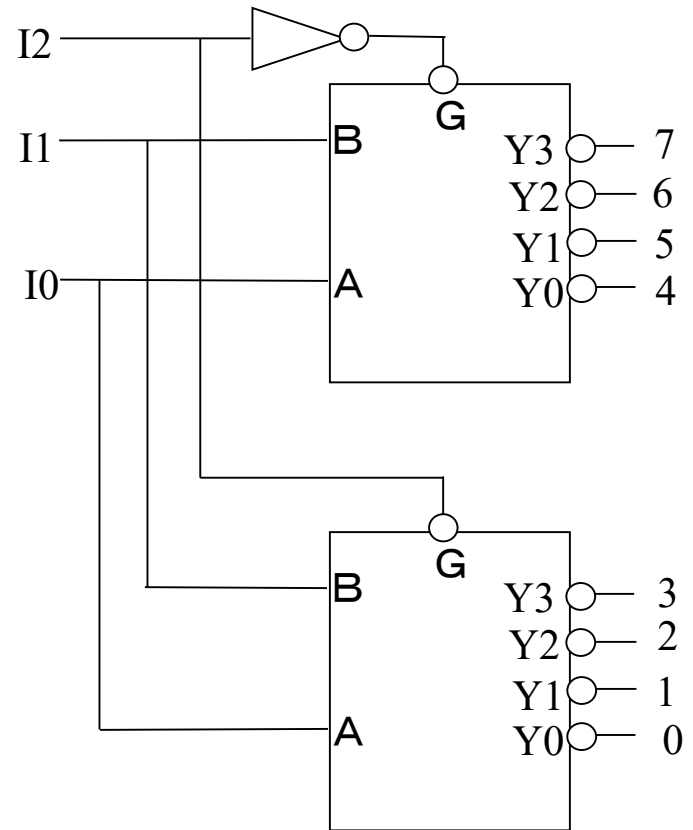
cf. active high: onのとき 1

e.g. $A_2A_1A_0=010$ のとき、 Y_2 が0になる

デコーダの連結



2-4 デコーダ



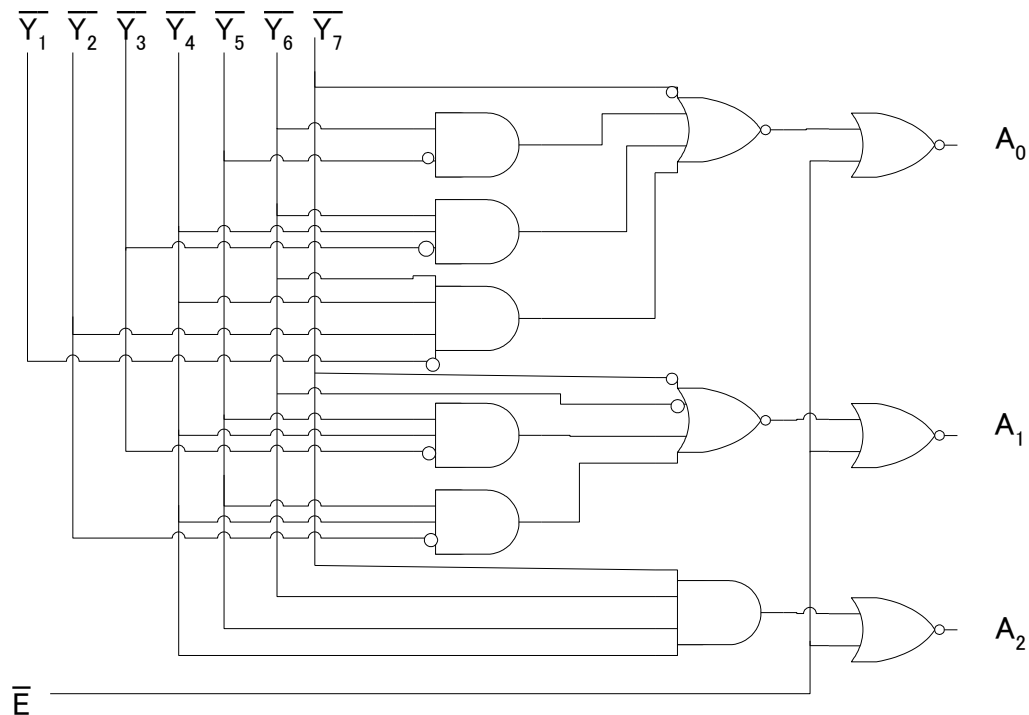
3-8 デコーダ

4.6 エンコーダ (符号化器、encoder)

i 本目の信号線が on \rightarrow 2進符号 i (n -bit) を生成する論理回路

| 入力 | | | | | | | | | 出力 | | |
|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-----------|-------|-------|-------|
| \bar{Y}_0 | \bar{Y}_1 | \bar{Y}_2 | \bar{Y}_3 | \bar{Y}_4 | \bar{Y}_5 | \bar{Y}_6 | \bar{Y}_7 | \bar{E} | A_0 | A_1 | A_2 |
| * | * | * | * | * | * | * | 0 | 0 | 1 | 1 | 1 |
| * | * | * | * | * | * | 0 | 1 | 0 | 0 | 1 | 1 |
| * | * | * | * | * | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| * | * | * | * | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| * | * | * | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 |
| * | * | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| * | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| * | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| * | * | * | * | * | * | * | * | 1 | 0 | 0 | 0 |

(a) 真理値表

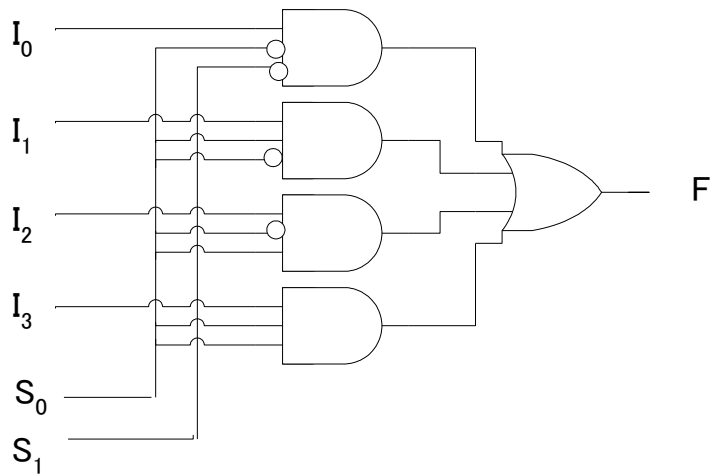


(b) 回路図

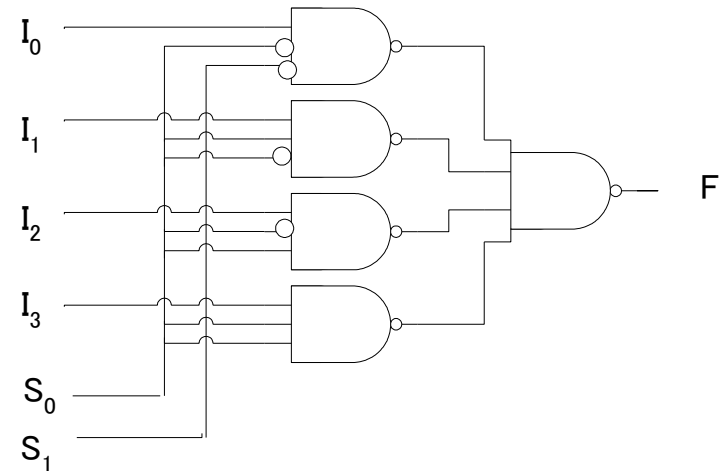
複数の入力 \rightarrow 一つ (最小値) を出力する論理回路
 \rightarrow 優先度付きエンコーダ (Priority Encoder)

4.7 マルチプレクサ(多重化装置、multiplexer)

- 複数の入力線からのデータを選択的に一本の共通線に接続する論理回路。各入力をチャンネル(channel)という



(a) 積和による実現



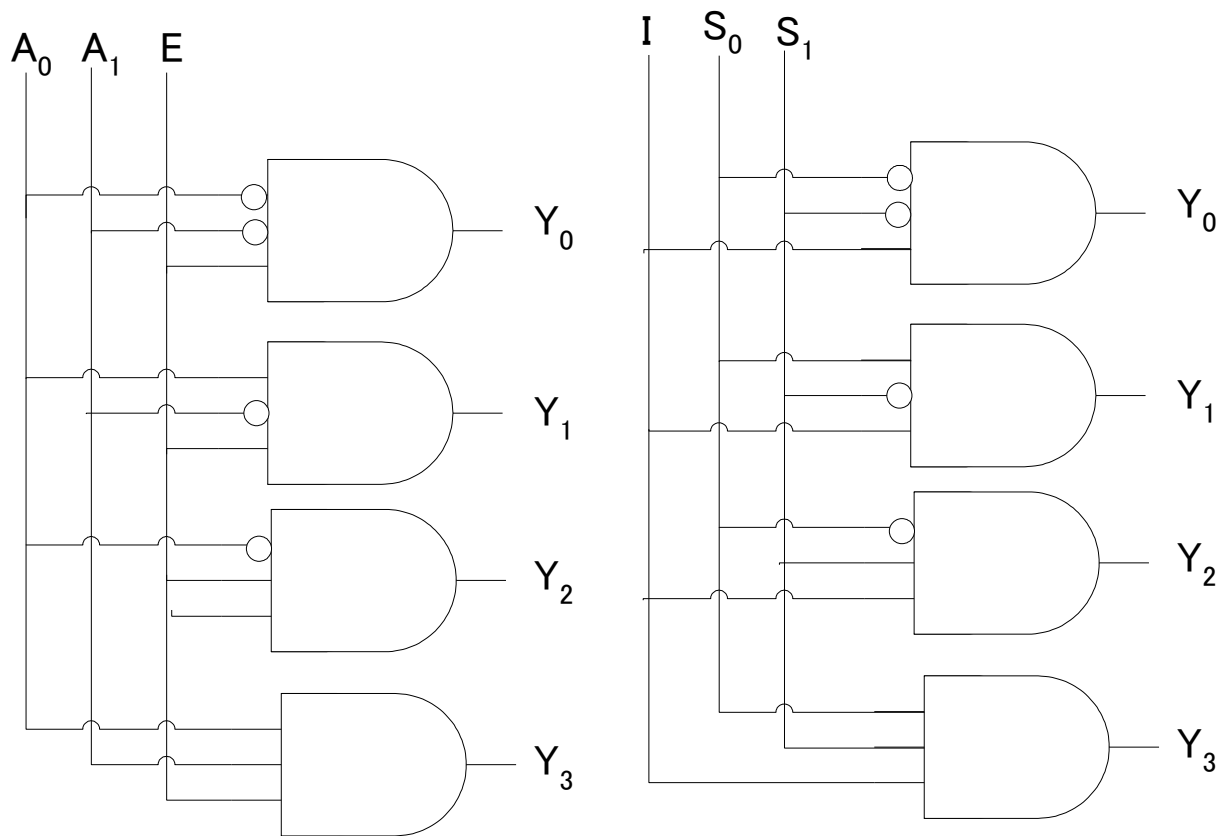
(b) NANDによる実現

4入力マルチプレクサ

マルチプレクサを**セレクタ**ともいう

4.8 デマルチプレクサ(多重分離装置、demultiplexer)

- マルチプレクサの反対で、一本の共通線からの信号を**選択的に**複数個の出力線に接続する論理回路 = **イネーブル付きのデコーダと等価**

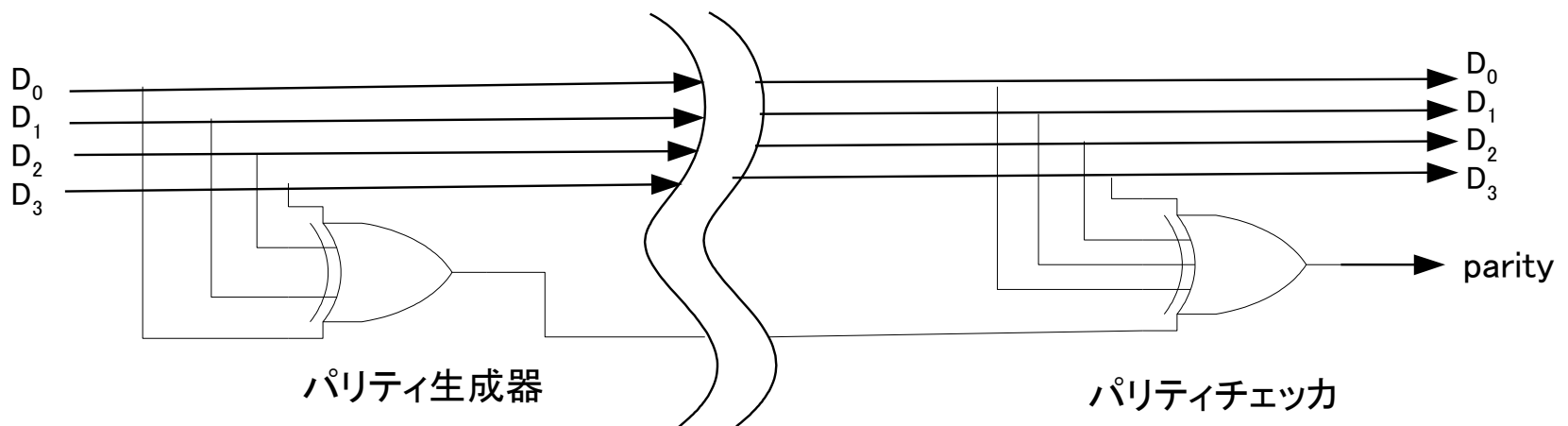


(a) デコーダ

(b) デマルチプレクサ

4.9 パリティ生成器(parity generator)とパリティチェッカ (parity checker)

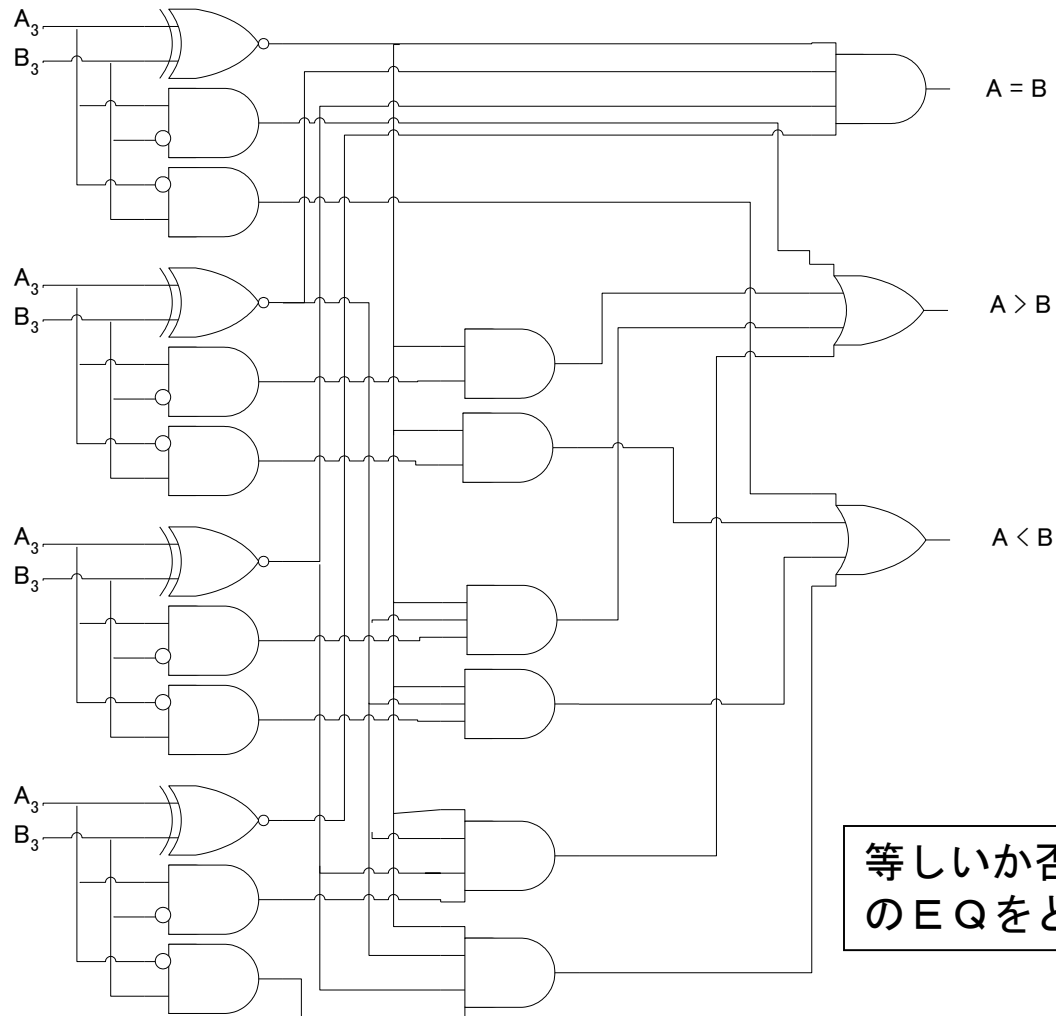
- パリティ: 入力の中で、1の数が奇数個のときに1, 偶数個のときに0
- パリティ生成器: パリティを作る論理回路
- パリティチェッカ: パリティをチェックする論理回路



パリティ生成器は n 入力の排他的論理和になり、パリティチェッカは $n + 1$ 入力の排他的論理和になる
⇒ データ転送のエラーチェックに使われる

4.10 コンパレータ (比較器、comparator)

■ 2つの2進数の大小の比較を行う論理回路



等しいか否かだけなら各ビットのEQをとってANDすればよい

問題

1. 4ビットの2進数2つの積をとる回路を設計せよ。
。1ビット全加算器を基本回路として使ってよい
2. 8入力の優先度つきエンコーダ (図4.15) 2つといくつかの素子を組み合わせて、16入力の優先度つきエンコーダを作れ。
3. 減算器 (加算器とNOTの組合せ。前回の講義参照) を使って、コンパレータを作れ
 $(A, B) \rightarrow (A < B, A = B, A > B)$
4. ここで述べたパリティチェッカでは1ビットの誤りを検出することはできるが、どのビットが誤ったかを検出することができず、正しいデータに修復することができない。
1ビットの誤りを**修正する回路**を作るにはどうしたらよいか？