

4. 組合せ回路

4-1 組合せ回路とは

組合せ回路：出力がその時点の入力の組合せだけで決まる回路

順序回路：出力がその時点の入力の組合せと過去の出力状態で決まる回路

加算器
減算器
比較器
データエンコーダ
データデコーダ
マルチプレクサ
デマルチプレクサ
符号変換回路

組合せ回路の設計手順

回路の入出力関係を真理値表で表わす

真理値表の出力「1」に着目した加法標準形
または
真理値表の出力「0」に着目した乗法標準形
の論理関数を求める

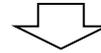
論理関数を簡単化する

基本論理素子を用い、加法形であればAND-OR回路
乗法形であればOR-AND回路を構成する
必要であれば、NANDまたはNORのみの回路構成とする

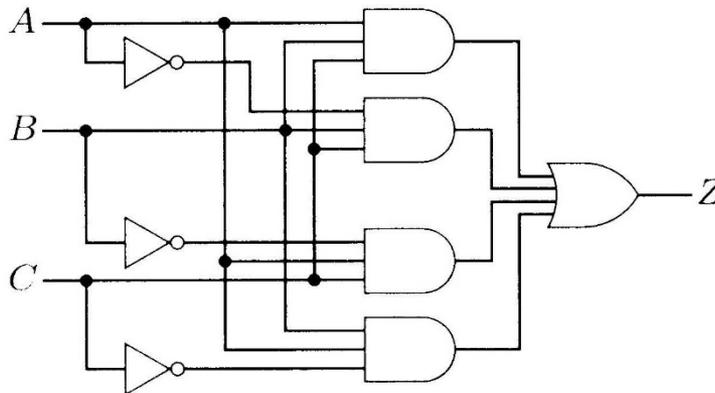
真理値表 → 複数の論理関数が構成可能

$$Z = ABC + ABC\bar{C} + \bar{A}BC + A\bar{B}C \quad (4-1) \leftarrow \text{複雑} = \text{素子数, 結線数多い}$$

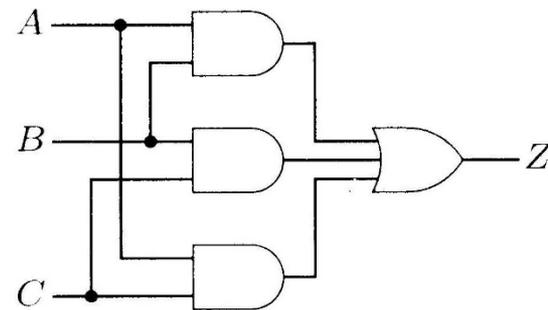
$$= AB + BC + CA \quad (4-2) \leftarrow \text{簡単化} = \text{素子数, 結線数減少}$$



安価
実装スペース節約
信頼性向上



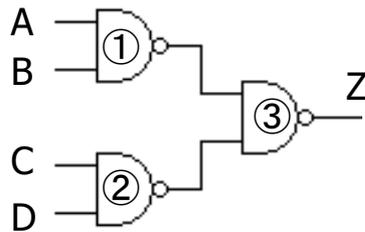
(4-1)で構成された論理回路



(4-2)で構成された論理回路

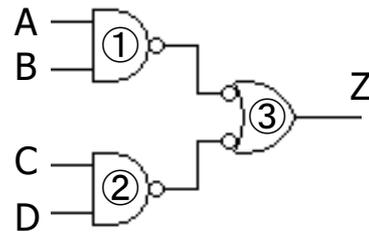
4-2 組合せ回路の構成

ゲートの出力と次段の入力のアクティブ状態をできるだけ一致させた形で記述する



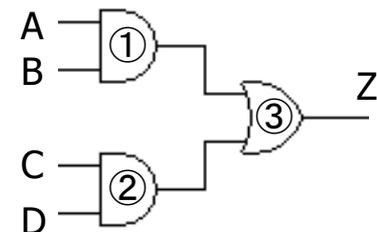
(a)

一見してZがいつ「1」になるのか不明



(b)

ゲート③のどちらかの入力が「0」であれば, Z=「1」



(c)

ANDとORの組合せと見ることもできる

(1) AND-OR構成

多段回路の変形

1段構成: $Z_1 = ABCDEFG$

2段構成: $Z_2 = ABCDEF + G$

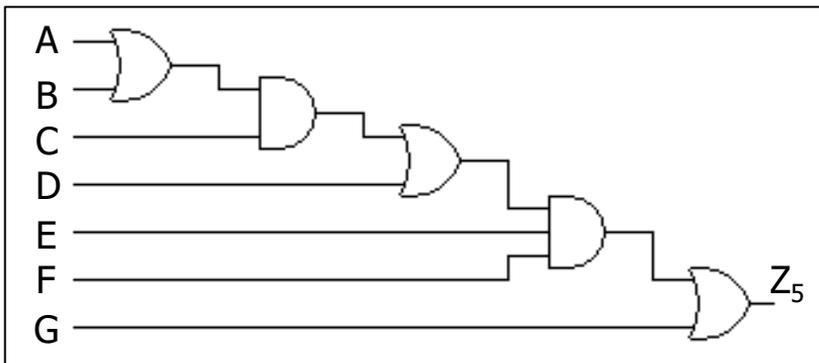
3段構成: $Z_3 = (A + B)CDEF + G$

4段構成: $Z_4 = (AB + C)DEF + G$

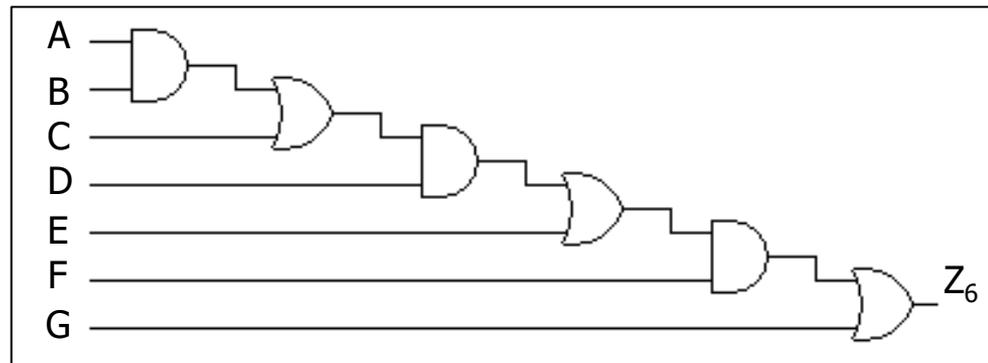
5段構成: $Z_5 = ((A + B)C + D)EF + G$

6段構成: $Z_6 = ((AB + C)D + E)F + G$

5段構成



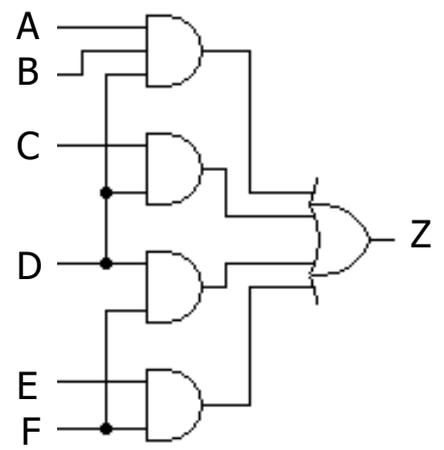
6段構成



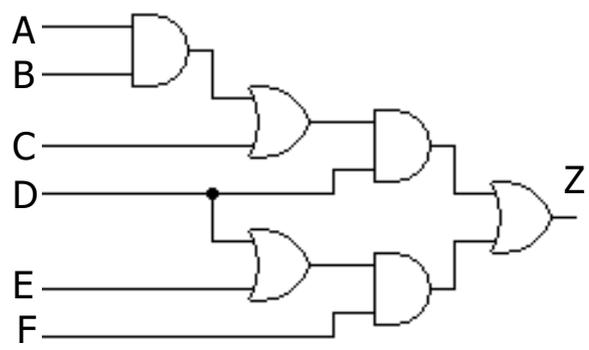
多段回路構成 (教p. 69 例題4. 1)

$Z = ABD + CD + DF + EF \rightarrow$ 2段構成

$= (AB + C)D + (D + E)F \rightarrow$ 4段構成



(a) 2段AND-OR構成



(b) 4段AND-OR構成

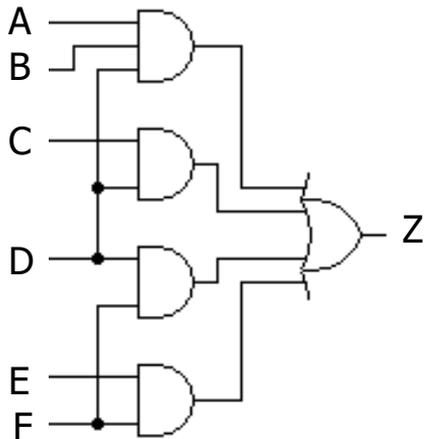
(2) NAND構成

加法形のNAND構成 (教p. 70)

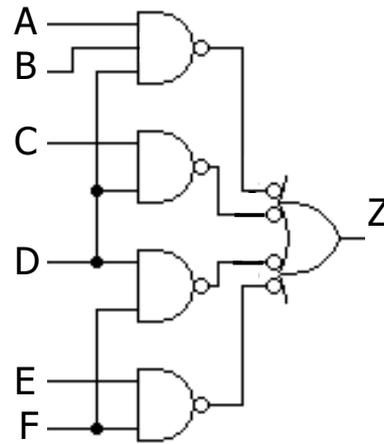
偶数段のゲート出力と奇数段のゲート入力に
状態表示記号を追加



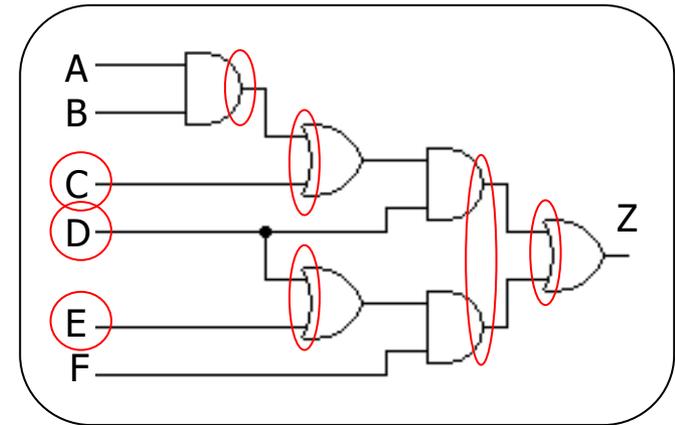
状態表示記号の追加によって論理が反転した
入力を元の状態に戻すため、インバータを追加



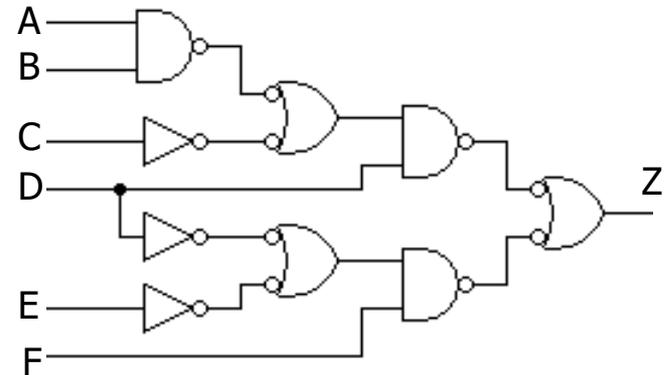
2段AND-OR構成



(a) 2段NAND構成



4段AND-OR構成(前ページ(b))

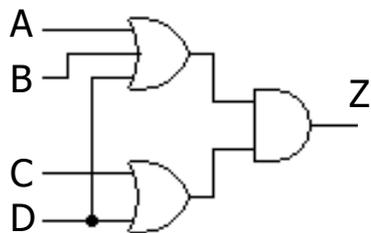


(b) 4段NAND構成

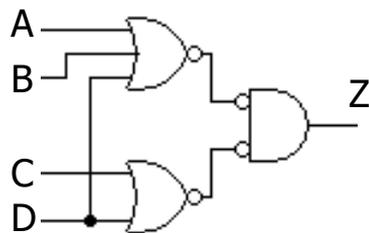
(3) NOR構成

「乗法形のNOR構成（教p. 70 例題4. 2）」

$$Z = (A + B + D)(C + D)$$

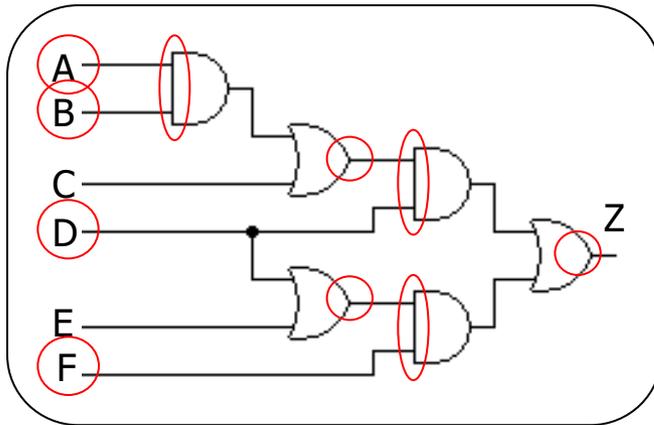


(a) OR-AND構成



(b) NOR構成

加法系のNOR構成 (教p. 71 例題4. 3)



4段AND-OR構成

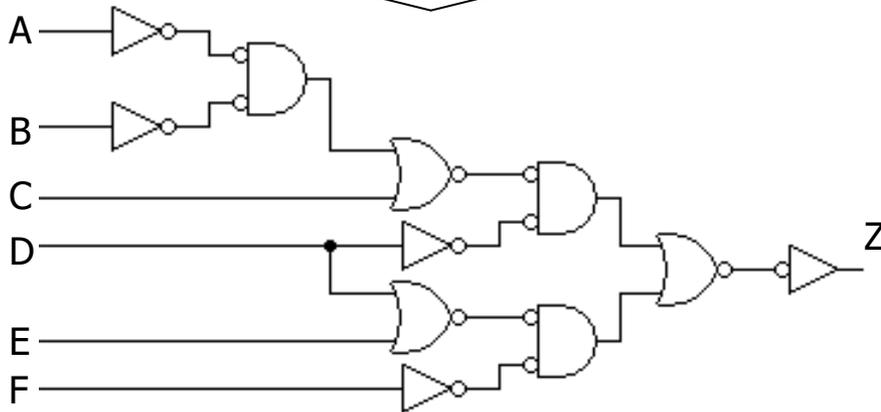
奇数段のゲート出力と偶数段のゲート入力に状態表示記号を追加



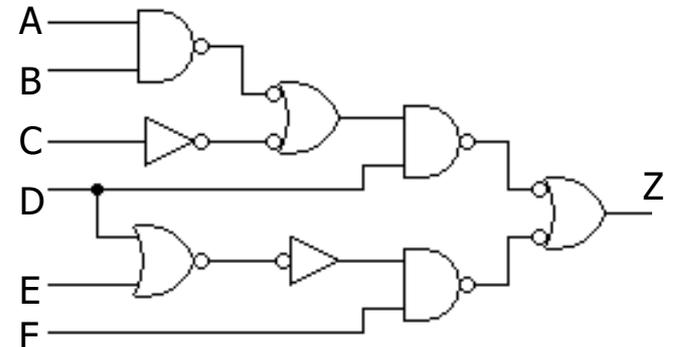
状態表示記号の追加によって論理が反転した入力を元の状態に戻すため、インバータを追加



最終出力段にインバータを追加



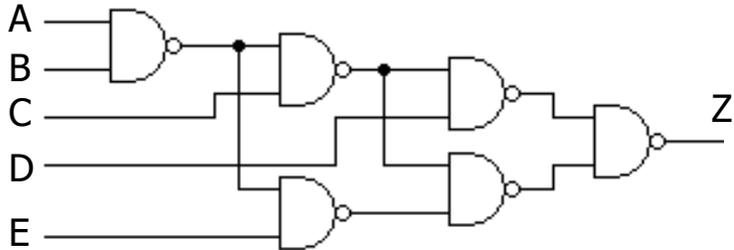
(a) 加法系のNOR構成



(b) 加法系のNAND-NOR構成

(4) 回路図の解析

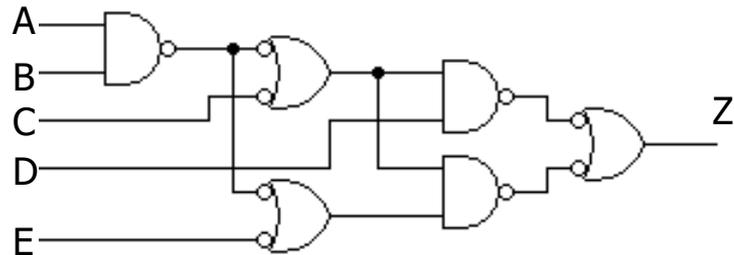
NAND回路構成の論理関数 (教p. 72 例題4. 4)



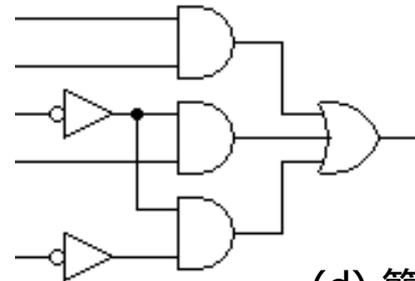
(a) NAND構成

図(c)より,

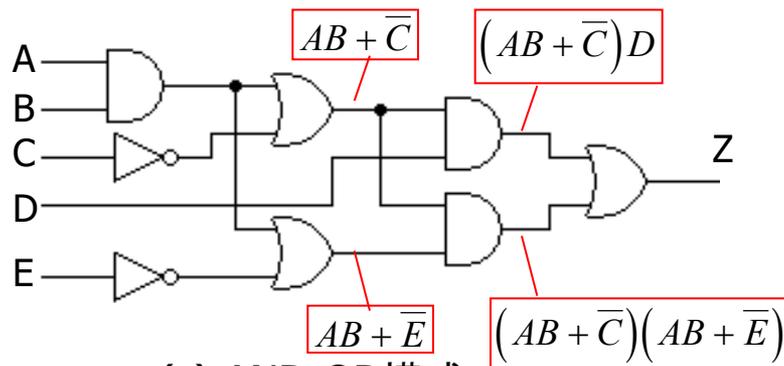
$$\begin{aligned} Z &= (AB + \bar{C})D + (AB + \bar{C})(AB + \bar{E}) \\ &= ABD + \bar{C}D + AB + AB\bar{E} + ABC\bar{E} + \bar{C}\bar{E} \\ &= AB(D + 1 + \bar{E} + \bar{C}) + \bar{C}D + \bar{C}\bar{E} = AB + \bar{C}D + \bar{C}\bar{E} \end{aligned}$$



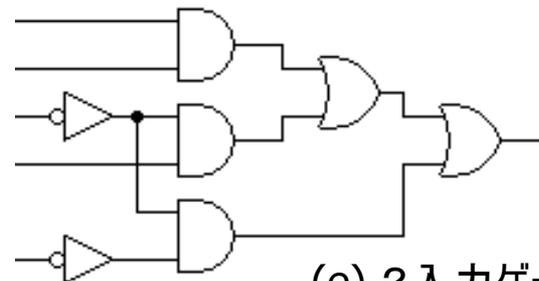
(b) 奇数段を負論理で記述



(d) 単純化したAND-OR構成



(c) AND-OR構成



(e) 2入力ゲートによるAND-OR構成

(2) 全加算器 (Full Adder; FA) ⇨ 前段からの桁上りを考慮する加算器

1ビット全加算器の構成 (教p. 74 例題4. 6)

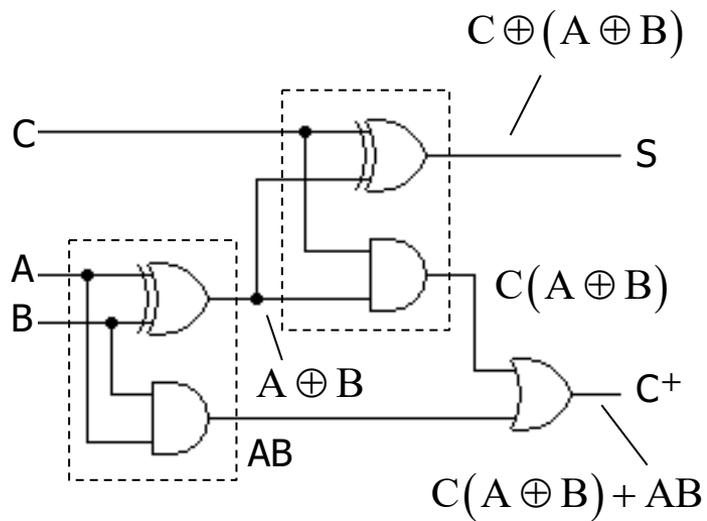
全加算器の真理値表

A	B	C	S	C ⁺
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

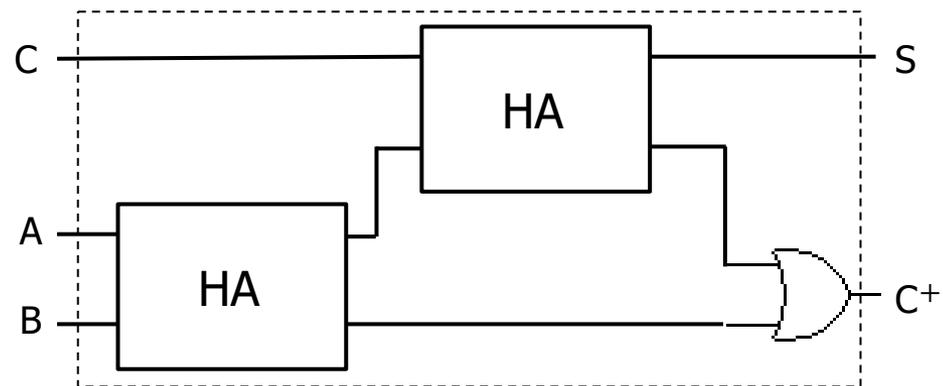
S と C⁺ を加法形で表すと,

$$\begin{aligned} S &= \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}\overline{C} + ABC \\ &= C(\overline{A}B + AB) + \overline{C}(\overline{A}B + A\overline{B}) \\ &= C(\overline{A \oplus B}) + \overline{C}(A \oplus B) \\ &= C \oplus (A \oplus B) \end{aligned}$$

$$\begin{aligned} C^+ &= \overline{A}BC + A\overline{B}C + ABC\overline{C} + ABC \\ &= C(\overline{A}B + A\overline{B}) + AB(C + \overline{C}) \\ &= C(A \oplus B) + AB \end{aligned}$$



(a) 加法形から構成したFA

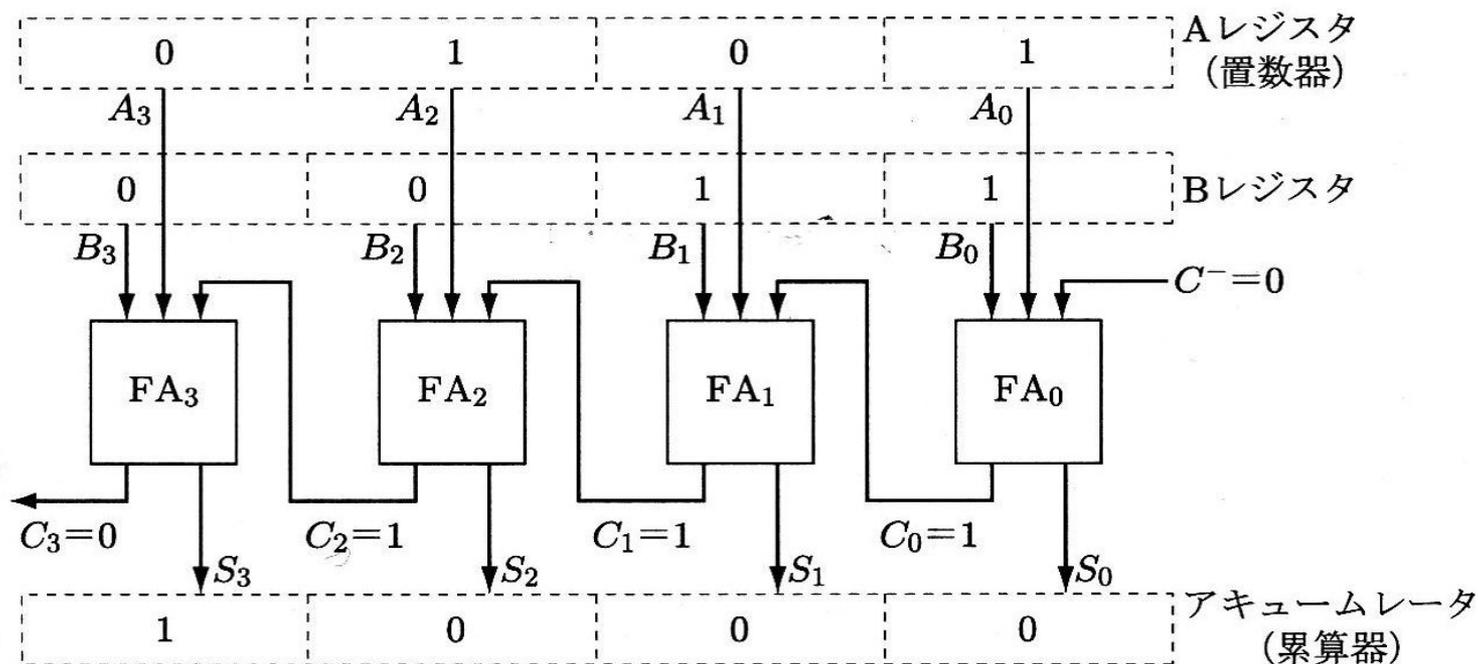


(b) 半加算器を用いた全加算器の構成

(3) 並列加算器(リプルキャリー型)

並列加算器: 全加算器を必要なビット数だけ並べ、前段の桁上がり出力を次段の桁上がり入力に接続する

リプルキャリー型: 最下位ビットの桁上がりが徐々に上位桁に伝搬するため、最上位ビットの桁上がりが得られるまで時間がかかる



全加算器の構成(リプルキャリー型)

(4) 並列加算器(キャリー・ルックアヘッド型)

キャリー・ルックアヘッド型: 最下位ビットからの桁上がりを待つことなく、最上位ビットの桁上がりを得られる

入力データ A_0, B_0 , 前段からの桁上がり C^- のとき, 次段への桁上がりは,

$$C_0 = C^-(A_0 \oplus B_0) + A_0B_0$$

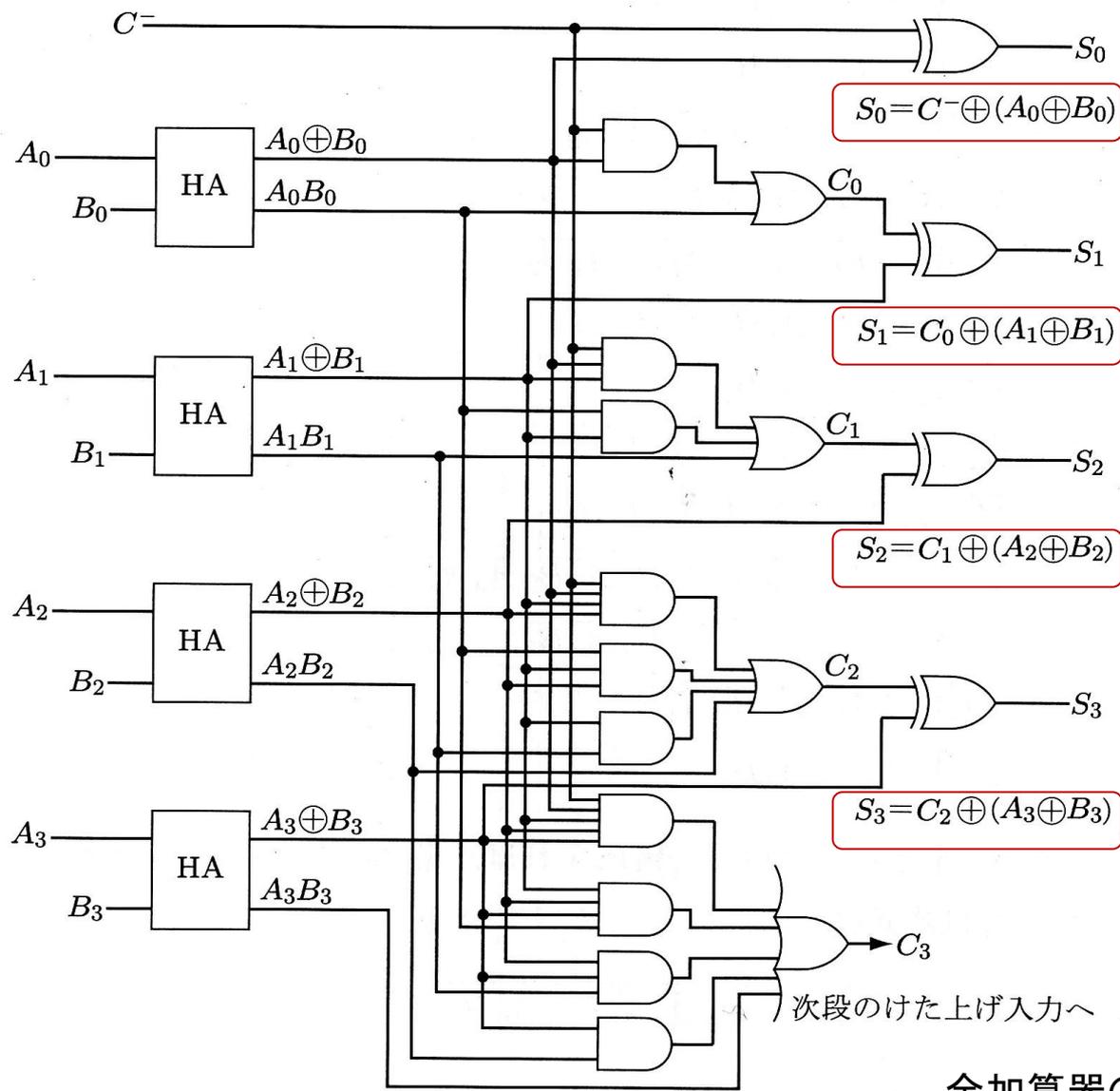
入力データ A_1, B_1 , 前段からの桁上がり C_0 のとき, 次段への桁上がりは,

$$\begin{aligned} C_1 &= C_0(A_1 \oplus B_1) + A_1B_1 \\ &= (C^-(A_0 \oplus B_0) + A_0B_0)(A_1 \oplus B_1) + A_1B_1 \\ &= C^-(A_0 \oplus B_0)(A_1 \oplus B_1) + A_0B_0(A_1 \oplus B_1) + A_1B_1 \end{aligned}$$

同様に, C_2, C_3 は,

$$\begin{aligned}C_2 &= C_1(A_2 \oplus B_2) + A_2B_2 \\ &= C^-(A_0 \oplus B_0)(A_1 \oplus B_1)(A_2 \oplus B_2) \\ &\quad + A_0B_0(A_1 \oplus B_1)(A_2 \oplus B_2) \\ &\quad + A_1B_1(A_2 \oplus B_2) \\ &\quad + A_2B_2\end{aligned}$$

$$\begin{aligned}C_3 &= C_2(A_3 \oplus B_3) + A_3B_3 \\ &= C^-(A_0 \oplus B_0)(A_1 \oplus B_1)(A_2 \oplus B_2)(A_3 \oplus B_3) \\ &\quad + A_0B_0(A_1 \oplus B_1)(A_2 \oplus B_2)(A_3 \oplus B_3) \\ &\quad + A_1B_1(A_2 \oplus B_2)(A_3 \oplus B_3) \\ &\quad + A_2B_2(A_3 \oplus B_3) \\ &\quad + A_3B_3\end{aligned}$$



全加算器の構成(キャリー・ルックアヘッド型)

(5) オーバーフロー検出回路

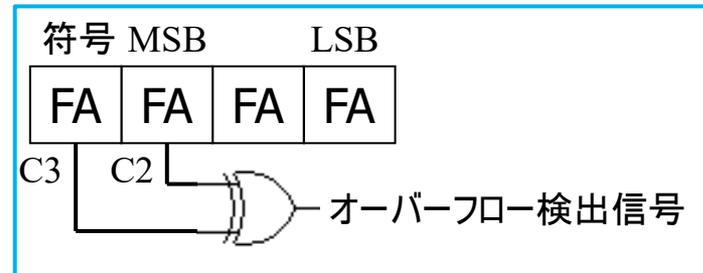
符号付き2進数の加算と減算(補数加算) → 4ビットの場合, 表示範囲は-8~+7

MSB: Most Significant Bit

LSB: Least Significant Bit

$$\begin{array}{r}
 0101 \dots 5 \\
 +) 0010 \dots 2 \\
 \hline
 0111 \dots 7 / \text{正答}
 \end{array}$$

(a) 符号ビットとMSBからの桁上がりなし



$$\begin{array}{r}
 1101 \dots -3 \\
 +) 1110 \dots -2 \\
 \hline
 11011 \dots -5 / \text{正答}
 \end{array}$$

(b) 符号ビットとMSBからの桁上がりがともにある

$$\begin{array}{r}
 0101 \dots 5 \\
 +) 0100 \dots 4 \\
 \hline
 1001 \dots -7 / \text{誤答}
 \end{array}$$

(c) MSBのみから桁上がりがある

$$\begin{array}{r}
 1011 \dots -5 \\
 +) 1100 \dots -4 \\
 \hline
 10111 \dots +7 / \text{誤答}
 \end{array}$$

(d) 符号ビットのみから桁上がりがある

符号ビットまたはMSBいずれかのみから桁上がりがある場合 → オーバーフロー

4-4 減算器

加算

全加算器で加算を実行

制御信号 C によって,
「1」→加算(ADD)
「0」→減算(SUB)
を切り替え

減算=補数加算

減数の1の補数(減数のビット反転) ... 補数器

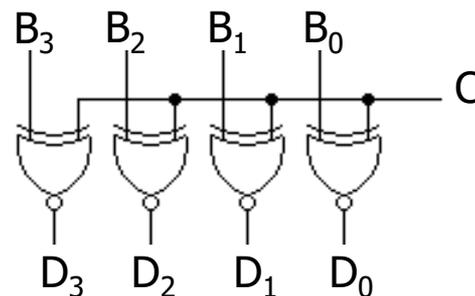
全加算器で被減数と減数(補数)を加算 ... 全加算器

最下位桁に+1 ... 全加算器の最下位桁に+1を入力

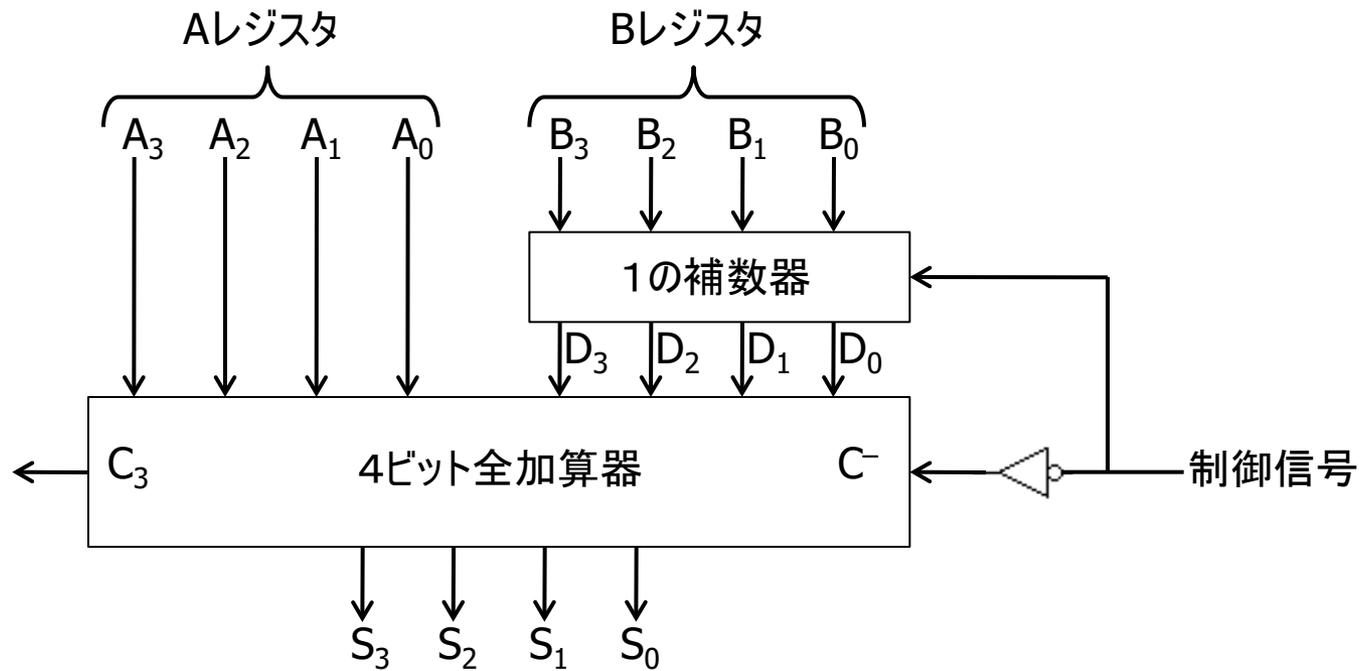
補数器の真理値表

B	C	D
0	1	0
1	1	1
0	0	1
1	0	0

$$D = BC + \overline{BC} = B \oplus C$$



補数器の回路構成



4ビット加減算器の回路構成

4-5 比較器

比較器：2つの1ビットデータ A , B を比較し, $A > B$, $A = B$, $A < B$ を判別(「1」を出力)する回路

比較器の設計 (教p. 82 例題4.7)

1ビット比較器の真理値表

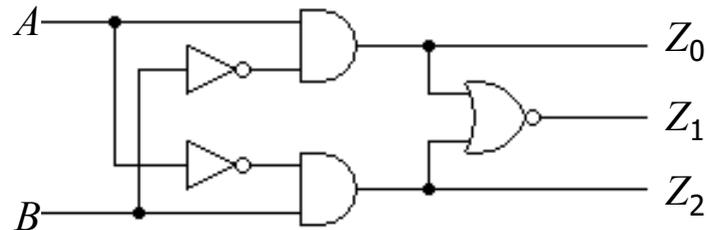
A	B	Z_0	Z_1	Z_2
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

$A > B$ $A = B$ $A < B$

$$Z_0 = \overline{A}B$$

$$Z_1 = \overline{A}\overline{B} + AB = \overline{A \oplus B} = \overline{\overline{A}B + A\overline{B}}$$

$$Z_2 = A\overline{B}$$

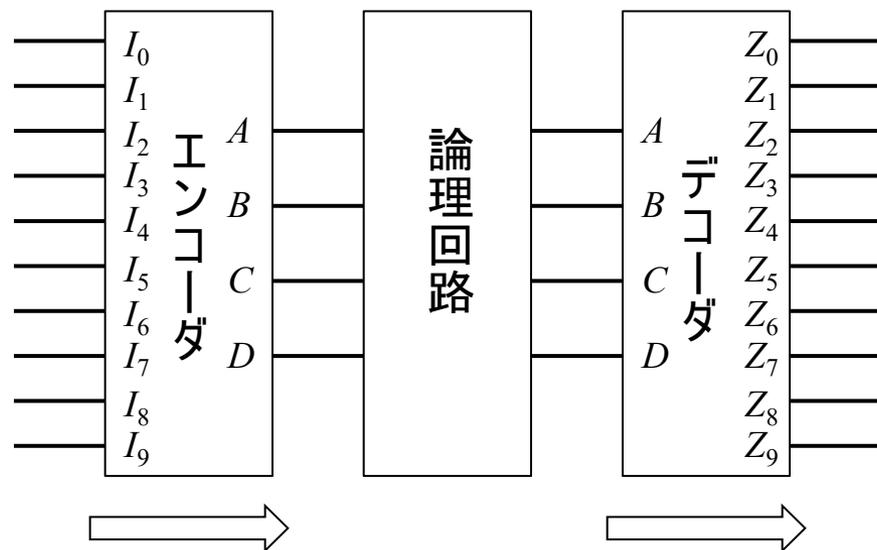


1ビットデータ比較器の回路構成

4-6 エンコーダとデコーダ

エンコーダ(符号器): アクティブ状態になった一つの入力に対応して2進符号を出力する回路
例えば、キーボードの'0'という数字を計算機内部で使用するアスキーコード(30)₁₆に変換する

デコーダ(復号器): 2進符号に対応する一つの出力をアクティブ状態にする回路
例えば、計算機内部でアスキーコードで表されている(30)₁₆という情報を元の'0'という数字に変換する



0~9までの数字を4ビットの
2進数(BCD符号)に変換

4ビットの2進数(BCD符号)を
対応する0~9までの数字に変換

エンコーダとデコーダの働き

(1) エンコーダ

4入力2ビット出力エンコーダの設計 (教p. 84 例題4.9)

4 to 2 エンコーダの真理値表

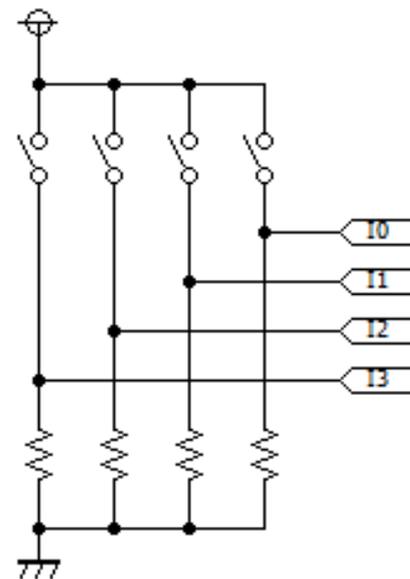
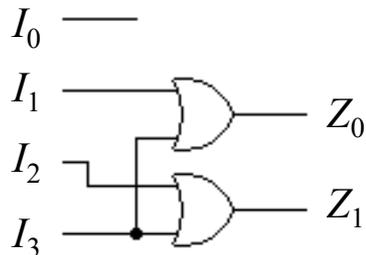
I_3	I_2	I_1	I_0	Z_1	Z_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

例えば、ここが1であれば、
数字の '0' が入力されたと考える

2進数で $(00)_2$ が
出力される

$$Z_0 = I_1 + I_3$$

$$Z_1 = I_2 + I_3$$



4入力2ビット出力エンコーダの回路構成

8入力3ビット出力エンコーダの例

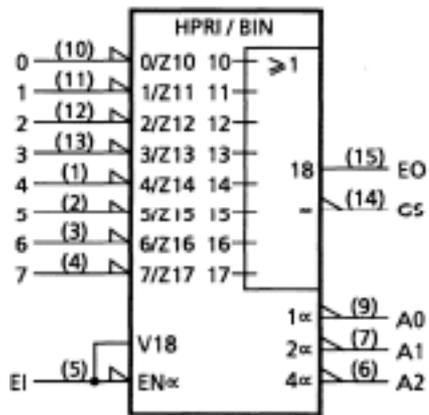
8 to 3ビットエンコーダの真理値表

A	B	C	D	E	F	G	H	Z ₂	Z ₁	Z ₀
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

74HC148 (8 to 3 Priority Encoder) の真理値表

Inputs									Outputs				
E1	0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H	X	X	X	X	X	X	X	X	H	H	H	H	H
L	H	H	H	H	H	H	H	H	H	H	H	H	L
L	X	X	X	X	X	X	X	L	L	L	L	L	H
L	X	X	X	X	X	X	L	H	L	L	H	L	H
L	X	X	X	X	L	H	H	H	L	H	L	L	H
L	X	X	X	L	H	H	H	H	H	L	L	L	H
L	X	X	L	H	H	H	H	H	H	L	H	L	H
L	X	L	H	H	H	H	H	H	H	H	L	L	H
L	L	H	H	H	H	H	H	H	H	H	H	L	H

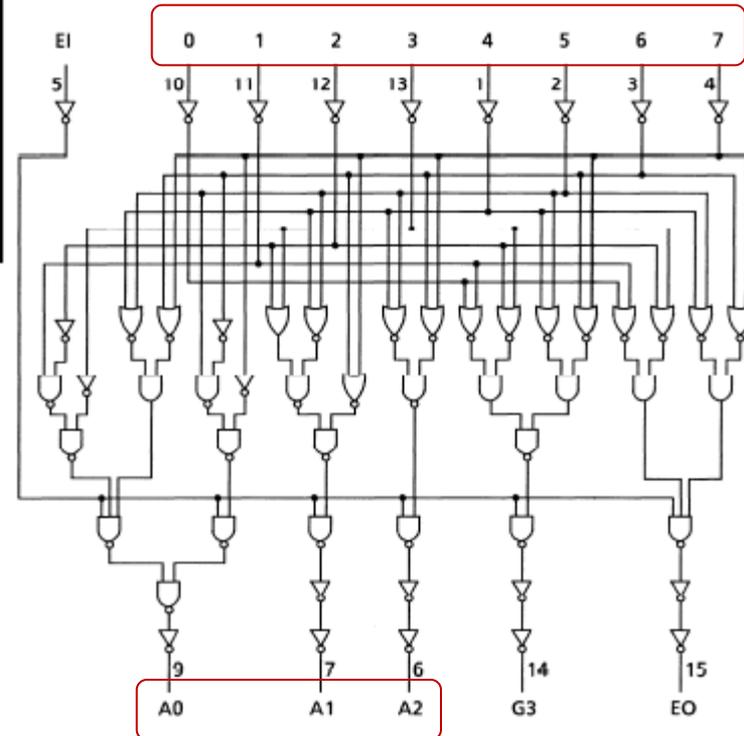
X: Don't care



74HC148の入出力信号

GS=「L」のときの
A₀=A₁=A₂=「H」が有効な符号

数値の大きいものほど優先順位が高い



74HC148の回路構成

(2) デコーダ

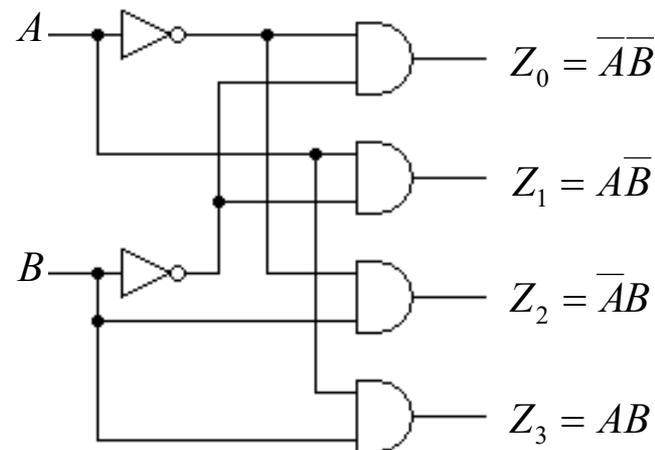
2ビット入力4出力デコーダの設計 (教p. 87 例題4. 11)

2 to 4 デコーダの真理値表

B	A	Z
0	0	Z_0
0	1	Z_1
1	0	Z_2
1	1	Z_3

例えば、2進数で $(00)_2$ が
入力された場合

数字の‘0’が出力され
たと考える



2ビット入力4出力エンコーダの回路構成

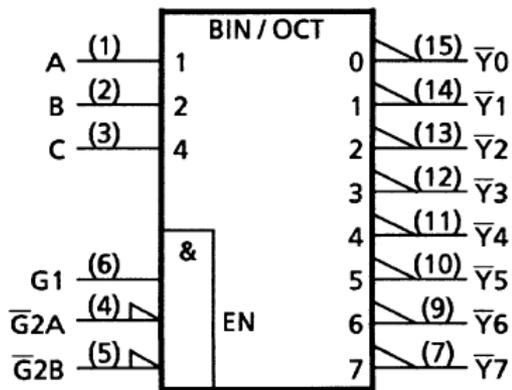
3ビット入力8出力デコーダの例

3ビット to 8 デコーダの真理値表

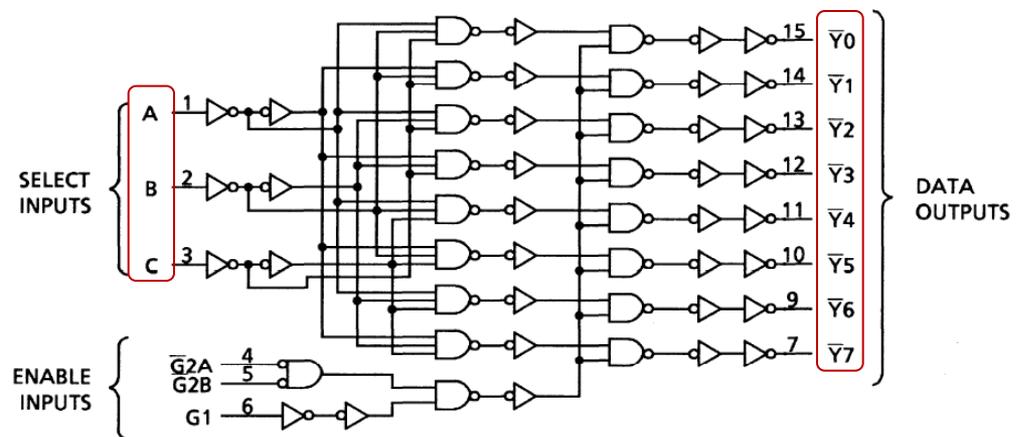
C	B	A	Z ₇	Z ₆	Z ₅	Z ₄	Z ₃	Z ₂	Z ₁	Z ₀
1	1	1	1	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	1

74HC138 (3 to 8 Decoder) の真理値表

Inputs						Outputs								Selected Output
Enable			Select			\bar{Y}_0	\bar{Y}_1	\bar{Y}_2	\bar{Y}_3	\bar{Y}_4	\bar{Y}_5	\bar{Y}_6	\bar{Y}_7	
G1	$\bar{G}2A$	$\bar{G}2B$	C	B	A									
L	X	X	X	X	X	H	H	H	H	H	H	H	H	None
X	H	X	X	X	X	H	H	H	H	H	H	H	H	None
X	X	H	X	X	X	H	H	H	H	H	H	H	H	None
H	L	L	L	L	L	L	H	H	H	H	H	H	H	\bar{Y}_0
H	L	L	L	L	H	H	L	H	H	H	H	H	H	\bar{Y}_1
H	L	L	L	H	L	H	H	L	H	H	H	H	H	\bar{Y}_2
H	L	L	L	H	H	H	H	H	L	H	H	H	H	\bar{Y}_3
H	L	L	H	L	L	H	H	H	H	L	H	H	H	\bar{Y}_4
H	L	L	H	L	H	H	H	H	H	H	L	H	H	\bar{Y}_5
H	L	L	H	H	L	H	H	H	H	H	H	L	H	\bar{Y}_6
H	L	L	H	H	H	H	H	H	H	H	H	H	L	\bar{Y}_7

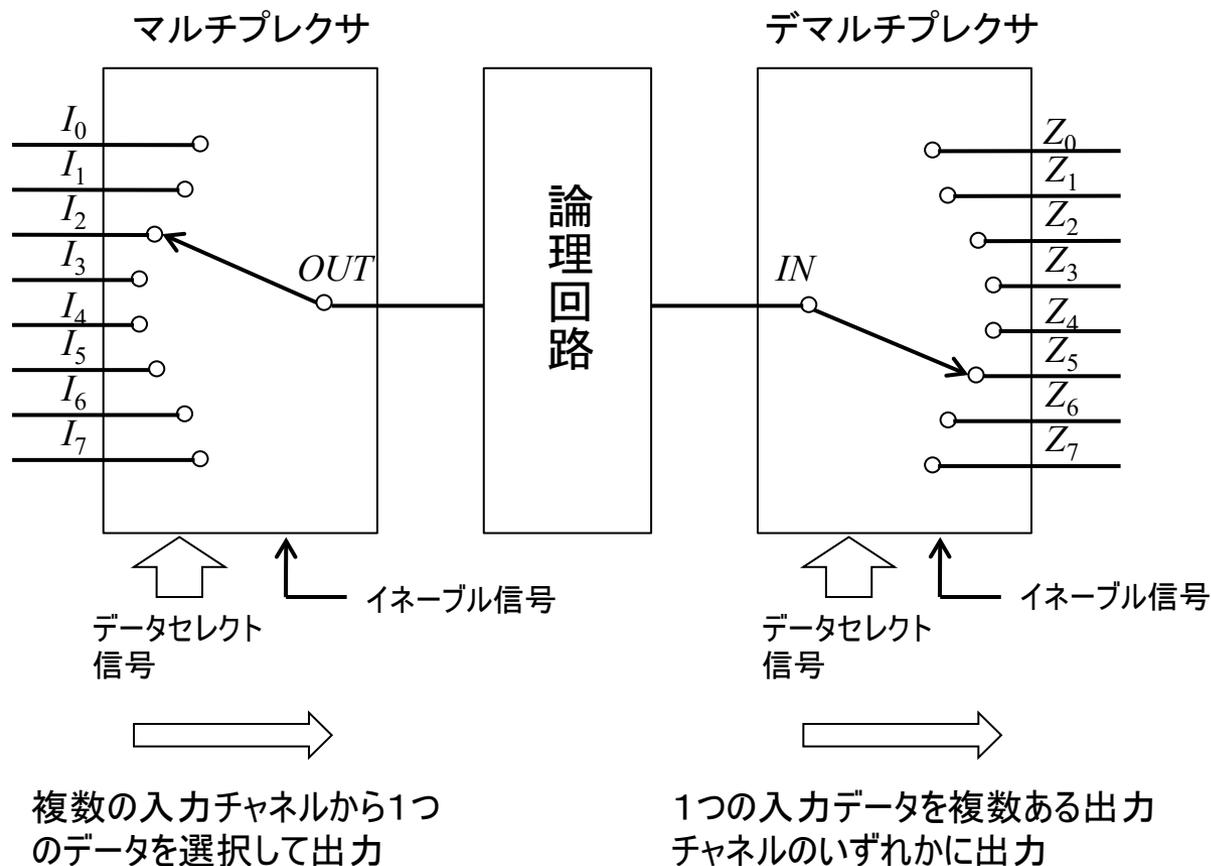


74HC138の入出力信号



74HC138の回路構成

4-7 マルチプレクサとデマルチプレクサ



マルチプレクサとデマルチプレクサの働き

(1) マルチプレクサ

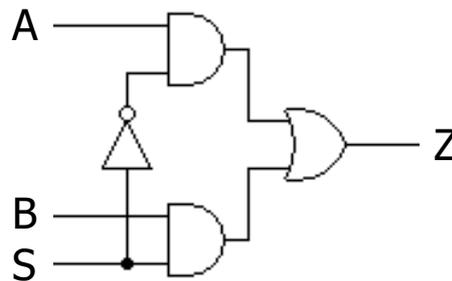
1ビット2チャンネルマルチプレクサ

$S = 0$ で A , $S = 1$ で B を出力

1ビット2チャンネルマルチプレクサ
真理値表

S	B	A	Z
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$$\begin{aligned} Z &= \bar{S}\bar{B}A + \bar{S}BA + S\bar{B}\bar{A} + SBA \\ &= \bar{S}A(\bar{B} + B) + SB(\bar{A} + A) \\ &= \bar{S}A + SB \end{aligned}$$



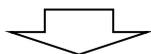
1ビット2チャンネルマルチプレクサ
の回路構成

1ビット4チャンネルマルチプレクサ (教p. 90 例題4. 14)

データセレクト信号 S_0, S_1 で入力 A, B, C, D のうちいずれか一つを出力

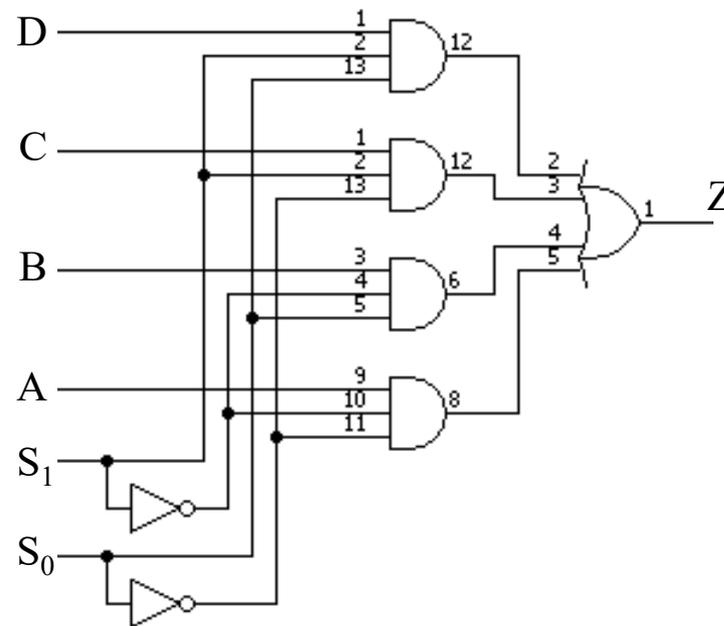
1ビット4チャンネルマルチプレクサの真理値表

S_1	S_0	D	C	B	A	Z
0	0	0	0	0	d_0	d_0
0	1	0	0	d_1	0	d_1
1	0	0	d_2	0	0	d_2
1	1	d_3	0	0	0	d_3



A, B, C, D がそれぞれ「1」のとき、 $Z = 1$ なので、真理値表より、

$$Z = \overline{S_1}\overline{S_0}A + \overline{S_1}S_0B + S_1\overline{S_0}C + S_1S_0D$$



1ビット4チャンネルマルチプレクサの回路構成

1ビット2チャンネルデマルチプレクサ

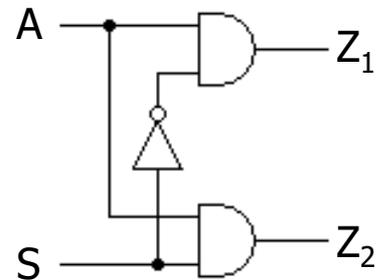
入力AをS = 0でZ₁に, S = 1でZ₂に出力

1ビット2チャンネルデマルチプレクサ
真理値表

S	A	Z ₁	Z ₂
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

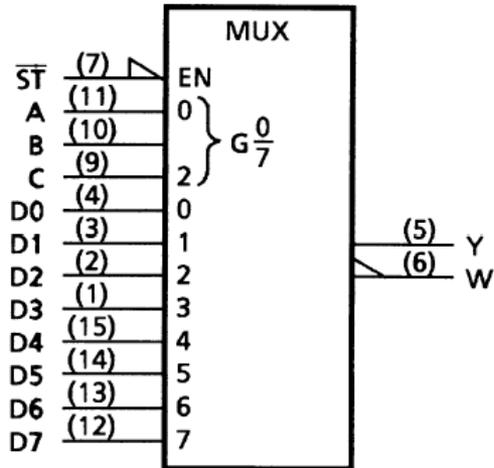
$$Z_1 = \bar{S}A$$

$$Z_2 = SA$$



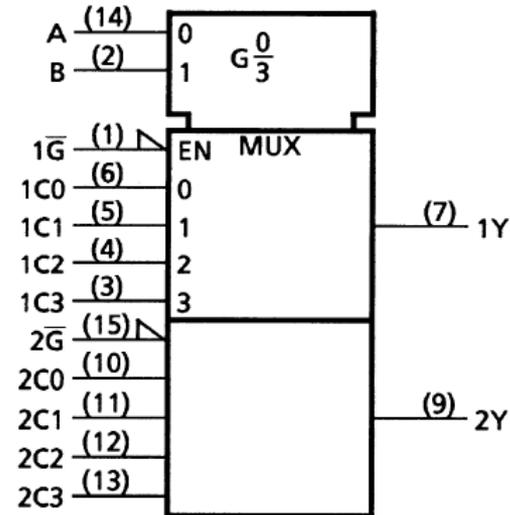
1ビット2チャンネルデマルチプレクサの回路構成

74HC151 (8 to 1 Multiplexer)



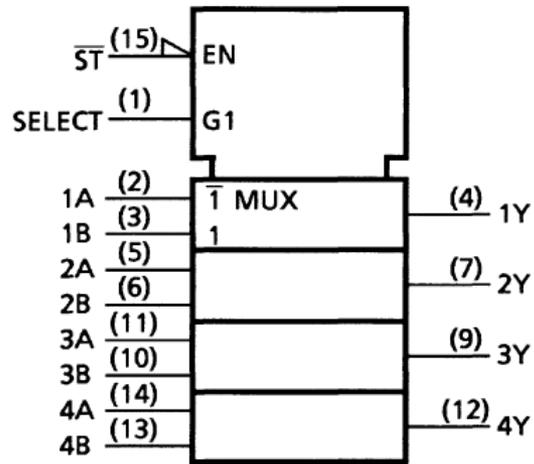
Inputs				Outputs	
Select			Strobe	Y	W
C	B	A	\overline{ST}		
X	X	X	H	L	H
L	L	L	L	D0	$\overline{D0}$
L	L	H	L	D1	$\overline{D1}$
L	H	L	L	D2	$\overline{D2}$
L	H	H	L	D3	$\overline{D3}$
H	L	L	L	D4	$\overline{D4}$
H	L	H	L	D5	$\overline{D5}$
H	H	L	L	D6	$\overline{D6}$
H	H	H	L	D7	$\overline{D7}$

74HC153 / HC253 (Dual 4 to 1 Multiplexer)

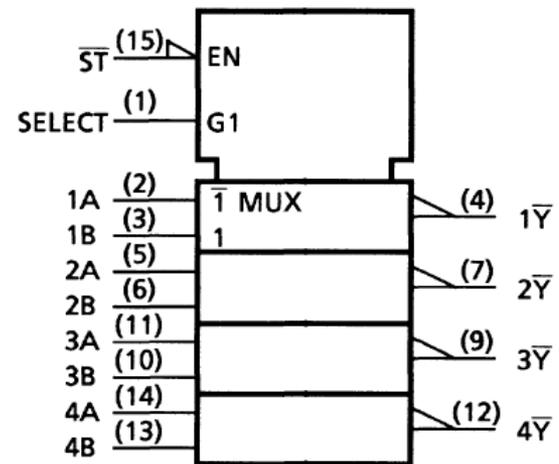


Select Inputs		Data Inputs				Strobe	Outputs Y	
B	A	C0	C1	C2	C3	\overline{G}	HC153A	HC253A
X	X	X	X	X	X	H	L	Z
L	L	L	X	X	X	L	L	L
L	L	H	X	X	X	L	H	H
L	H	X	L	X	X	L	L	L
L	H	X	H	X	X	L	H	H
H	L	X	X	L	X	L	L	L
H	L	X	X	H	X	L	H	H
H	H	X	X	X	L	L	L	L
H	H	X	X	X	H	L	H	H

74HC157 (Quad 2 to 1 Multiplexer)

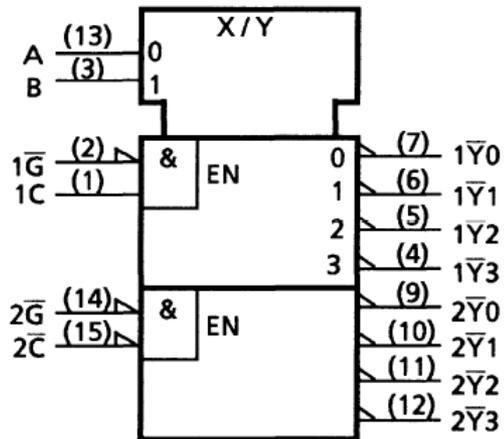


74HC158 (Quad 2 to 1 Multiplexer)



Inputs				Output	
\overline{ST}	SELECT	A	B	Y (157A)	\overline{Y} (158A)
H	X	X	X	L	H
L	L	L	X	L	H
L	L	H	X	H	L
L	H	X	L	L	H
L	H	X	H	H	L

74HC155 (Dual 2 to 4 / Single 3 to 8 Line decoder), Demultiplexer



1Cは'H'で1Y0～1Y3を制御(イネーブル)
 2Cは'L'で2Y0～2Y3を制御(イネーブル)
 1Cと2Cを接続すると3-8デコーダー

Inputs				Outputs			
B	A	1G	1C	1Y0	1Y1	1Y2	1Y3
X	X	H	X	H	H	H	H
L	L	L	H	L	H	H	H
L	H	L	H	H	L	H	H
H	L	L	H	H	H	L	H
H	H	L	H	H	H	H	L
X	X	X	L	H	H	H	H

Inputs				Outputs			
B	A	2G	2C	2Y0	2Y1	2Y2	2Y3
X	X	H	X	H	H	H	H
L	L	L	L	L	H	H	H
L	H	L	L	H	L	H	H
H	L	L	L	H	H	L	H
H	H	L	L	H	H	H	L
X	X	X	H	H	H	H	H