論理回路

第5回 代表的な組み合わせ論理回路

http://www.info.kindai.ac.jp/LC
E館3階E-331 内線5459
takasi-i@info.kindai.ac.jp

1

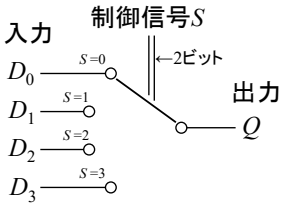
組み合わせ回路

- 定義: 組み合わせ回路
  - ある時刻の出力信号が、現在の入力信号だけで決まる回路
- 定義: 順序回路
  - ある時刻の出力信号が、現在の入力信号だけでなく、過去の入力信号の影響も受ける回路 (回路内にバッファ・メモリがある)

2

選択器 (multiplexor)

- 2^n本の入力から1本を選択し出力する回路
- 2^n+n 入力1出力
  - 入力: $D = (D_0,D_1,\ldots,D_{2^n-1})$, n ビット制御信号S
  - 出力: Q
  - ✓ 例: 2ビット選択器



S	S_1 S_0	Q
0	0 0	D_0
1	0 1	D_1
2	1 0	D_2
3	1 1	D_3

3

選択器

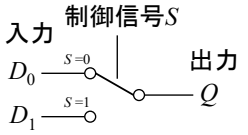
- $D = (D_0,D_1,\ldots,D_{2^n-1})$: 入力
- S : n ビット制御信号
- Q : 出力

```
MulPle (D,S ) {
        switch (S ) {
        case 0   : Q =D0   ; break;
        case 1   : Q =D1   ; break;
                   ⋮
        case 2^n-1: Q =D2^n-1; break;
        }
}
```

4

1ビット選択器

- 2本の入力$\{D_0,D_1\}$から1本を選択
  - 1ビット信号Sで制御



$$Q = \overline{S} \cdot D_0 + S \cdot D_1$$

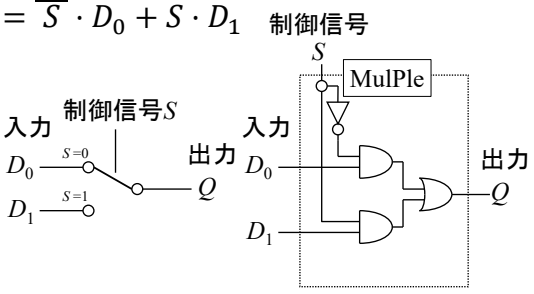
S	Q
0	D_0
1	D_1

D_i	S	Q
D_0 =0	0	0
D_0 =1		1
D_1 =0	1	0
D_1 =1		1

5

1ビット選択器の設計

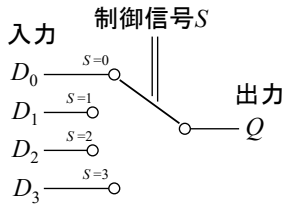
$$Q = \overline{S} \cdot D_0 + S \cdot D_1$$



6

2ビット選択器

- 2^2=4本の入力 D_0,D_1,D_2,D_3 から1本を選択
  - 2ビット信号 S=(S_1,S_0) で制御



S	S_1 S_0	Q
0	0 0	D_0
1	0 1	D_1
2	1 0	D_2
3	1 1	D_3

7

2ビット選択器

S_1 S_0	Q
0 0	D_0
0 1	D_1
1 0	D_2
1 1	D_3

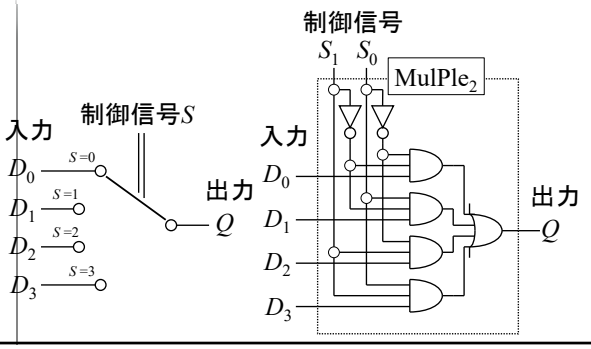
D_i	S_1 S_0	Q
D_0=0	0 0	0
D_0=1		1
D_1=0	0 1	0
D_1=1		1
D_2=0	1 0	0
D_2=1		1
D_3=0	1 1	0
D_3=1		1

$Q = S_1S_0D_0 + \bar{S}_1S_0D_1 + S_1\bar{S}_0D_2 + S_1S_0D_3$

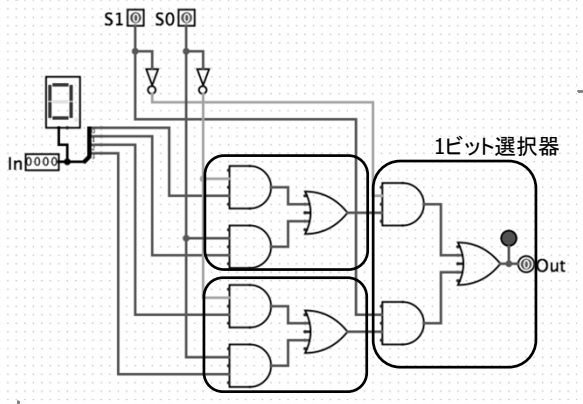
8

2ビット選択器の設計

$Q = S_1S_0D_0 + \bar{S}_1S_0D_1 + S_1\bar{S}_0D_2 + S_1S_0D_3$



9

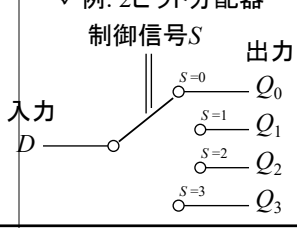


MP2.circ

10

分配器(demultiplexor)

- 信号を2^n本のうち1本に出力する回路
- n +1入力2^n出力
  - 入力: D, n ビット制御信号S
  - 出力: Q = (Q_0,Q_1,…,Q_{2^n-1})
  - ✓ 例: 2ビット分配器



S	S_1 S_0	Q_0 Q_1 Q_2 Q_3
0	0 0	D 0 0 0
1	0 1	0 D 0 0
2	1 0	0 0 D 0
3	1 1	0 0 0 D

11

分配器

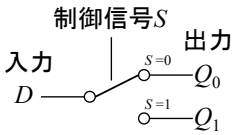
- D : 入力
- S : n ビット制御信号
- Q = (Q_0,Q_1,…,Q_{2^n-1}) : 出力

```
DeMulPle (D,S ) {
        switch (S ) {
        case 0    : Q0 =D ;    break;
        case 1    : Q1 =D ;    break;
                  ⋮
        case 2n-1: Q2n -1 =D ; break;
        }
}
```

12

1ビット分配器

- 信号を2本のうち1本に出力
  - 1ビット信号Sで制御



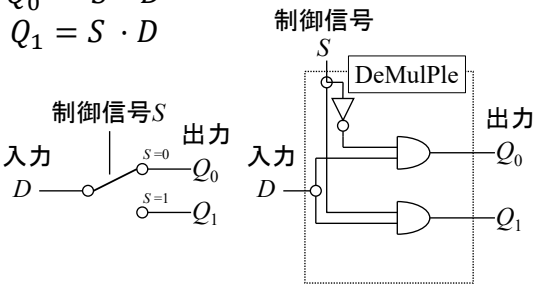
D	S	Q_0	Q_1
0	0	0	0
1		1	0
0	1	0	0
1		0	1

$$Q_0 = \overline{S} \cdot D$$
$$Q_1 = S \cdot D$$

13

1ビット分配器の設計

$$Q_0 = \overline{S} \cdot D$$
$$Q_1 = S \cdot D$$



14

比較器 (comparator)

- 入力の大小を比較する
  - 2入力3出力　　Z_X : X の方が大きい
    - 入力 : X,Y　　Z_Y : Y の方が大きい
    - 出力 : Z_X,Z_Y,Z_{eq}　　Z_{eq}: XとY が同じ

```
Comp (X,Y) {
        if (X>Y) Z_X = true;
        else if (Y>X) Z_Y = true;
        else Z_eq = true;
}
```

15

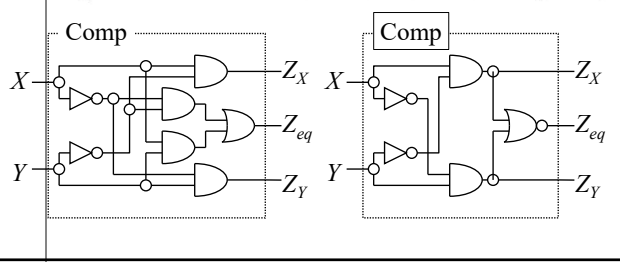
1ビット比較器の論理関数

X Y	Z_X	Z_Y	Z_{eq}
0 0	0	0	**1**
0 1	0	**1**	0
1 0	**1**	0	0
1 1	0	0	**1**

$$Z_X = X \cdot \overline{Y}$$
$$Z_Y = \overline{X} \cdot Y$$
$$Z_{eq} = \overline{X} \cdot \overline{Y} + X \cdot Y$$

16

1ビット比較器の設計



17

2ビット比較器の真理値表

- 2ビット×2入力3出力
  - 入力 : X=(X_1,X_0),Y=(Y_1,Y_0)
  - 出力 : Z_X,Z_Y,Z_{eq}

16通りの組み合わせ

X_1 X_0	Y_1 Y_0	Z_X	Z_Y	Z_{eq}
0 0	0 0	0	0	**1**
	0 1	0	**1**	0
	1 0	0	**1**	0
	1 1	0	**1**	0
0 1	0 0	**1**	0	0
	0 1	0	0	**1**
	1 0	0	**1**	0
	1 1	0	**1**	0
1 0	0 0	**1**	0	0
	0 1	**1**	0	0
	1 0	0	0	**1**
	1 1	0	**1**	0
1 1	0 0	**1**	0	0
	0 1	**1**	0	0
	1 0	**1**	0	0
	1 1	0	0	**1**

18

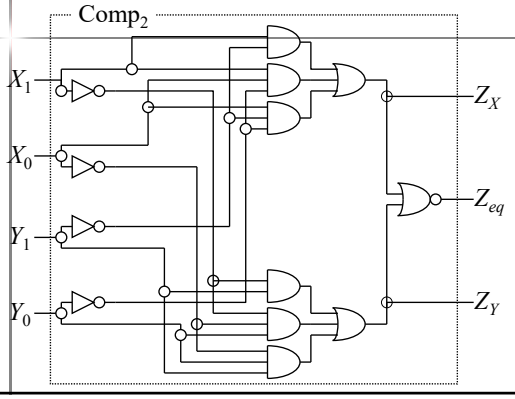
2ビット比較器の論理関数

$X_1 X_0$ \ $Y_1 Y_0$	0 0	0 1	1 1	1 0
0 0	=	<	<	<
0 1	>	=	<	<
1 1	>	>	=	>
1 0	>	>	<	=

- $>$: $X>Y$
- $=$: $X=Y$
- $<$: $X<Y$

19

2ビット比較器の設計



20

多ビット比較器の場合

- 比較器のビット数と出力の組み合わせ数
  1. 1ビット : $2^1 \times 2^1 = 4$通り
  2. 2ビット : $2^2 \times 2^2 = 16$通り
  3. 3ビット : $2^3 \times 2^3 = 64$通り
  4. 4ビット : $2^4 \times 2^4 = 256$通り
  5. 5ビット : $2^5 \times 2^5 = 1024$通り

Java の場合 : int 型 32 ビット long 型 64 ビット

ビット数が増えるにつれ膨大な組み合わせが必要

⇩

高ビット比較器の設計はとても無理！

21

複雑な回路の設計・製作

- 複雑な回路
  - 設計が難しい
  - 製作コストが高価

⇩

複雑な回路をより簡単な回路(設計済み)の組み合わせで作る

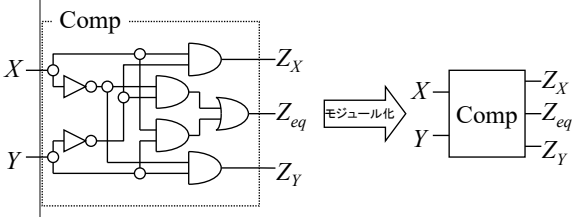
- 設計が簡単になる
- 回路の量産化で製作コスト削減

22

回路のモジュール化

- 回路全体を1つのゲートとみなす

例 : 比較器



23

大小比較

3桁の数の大小比較 (例 : 126 と 127)

1. 百の位を比較
2. 百の位が同じなら十の位を比較
3. 百の位と十の位が同じなら一の位を比較

大きくなる条件は

百の位が大きい

または 百の位が同じ かつ 十の位が大きい

または 百の位が同じ かつ 十の位が同じ かつ 一の位が大きい

等しくなる条件は

百の位が同じ かつ 十の位が同じ かつ 一の位が同じ

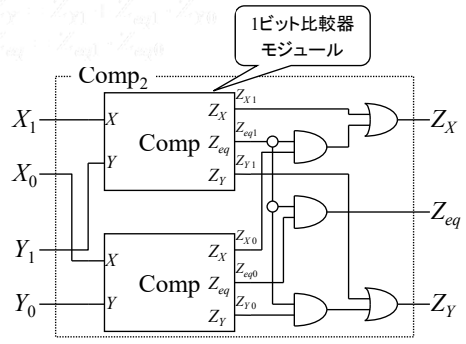
24

1ビット比較器モジュールを用いた 2ビット比較器

- $X=(X_1,X_0)$, $Y=(Y_1,Y_0)$ の大小比較
  - $X>Y$: $X_1>Y_1$ または ($X_1=Y_1$ かつ $X_0>Y_0$)
  - $X<Y$: $X_1<Y_1$ または ($X_1=Y_1$ かつ $X_0<Y_0$)
  - $X=Y$: $X_1=Y_1$ かつ $X_0=Y_0$
    - Z_{X1}, Z_{Y1}, Z_{eq1} : X_1, Y_1の比較結果
    - Z_{X0}, Z_{Y0}, Z_{eq0} : X_0, Y_0の比較結果

25

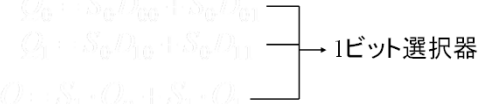
2ビット比較器の設計



26

1ビット選択器モジュールを用いた 2ビット選択器

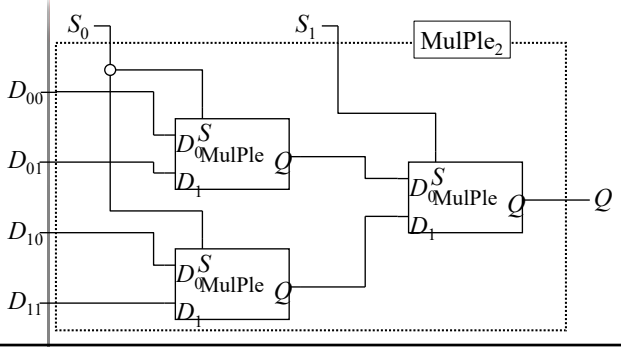
- 1ビット選択器: $D=(D_0,D_1)$, S
- 2ビット選択器: $D=(D_{00},D_{01},D_{10},D_{11})$, $S=(S_1,S_0)$



27

2ビット選択器の設計

- 1ビット選択器モジュール3個を用いて 2ビット選択器を設計

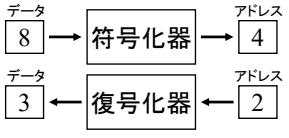


28

符号化器(encoder) 復号化器(decoder)

- 符号化 : 情報を数値コードに変える
- 復号化 : 数値コードを情報に戻す

✓例 : 情報をメモリに格納
- 情報 : データ
- コード : データのアドレス



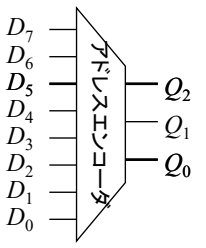
メモリ

アドレス	データ
0	1
1	8
2	**3**
3	5
4	**8**
5	
6	
7	

29

アドレス符号化器(address encoder)

- 入力: 2^n本の1ビット信号 $D=(D_{2^n-1},\ldots D_0)$
  - ただし、1本のみ 1,残りの2^n-1本は 0 が入力される
- 出力: n ビット信号 $Q=(D_{n-1},\ldots D_0)$



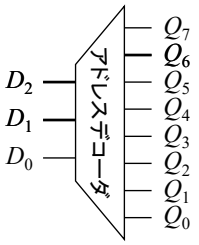
D	Q	D	Q
00000001	000	00010000	100
00000010	001	00100000	101
00000100	010	01000000	110
00001000	011	10000000	111

表に無い入力はドントケア

30

アドレス復号化器(address decoder)

– 入力: n ビット信号 $D=(D_{n-1},\ldots,D_0)$
– 出力: 2^n本の1ビット信号 $Q=(Q_{2^n-1},\ldots Q_0)$
  ■ ただし、1本のみ 1,残りの2^n-1本は 0 が出力される

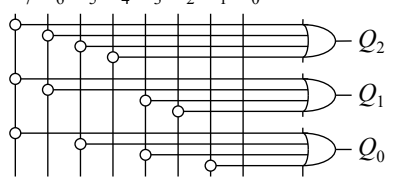


D	Q	D	Q
000	00000001	100	00010000
001	00000010	101	00100000
010	00000100	110	01000000
011	00001000	111	10000000

31

アドレスエンコーダ

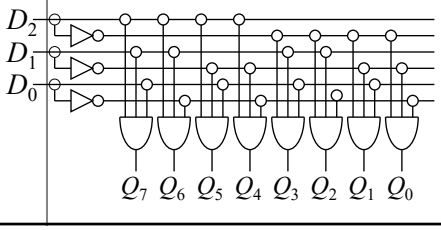
D	Q	D	Q
00000001	000	00010000	100
00000010	001	00100000	101
00000100	010	01000000	110
00001000	011	10000000	111



32

アドレスデコーダ

D	Q	D	Q
000	00000001	100	00010000
001	00000010	101	00100000
010	00000100	110	01000000
011	00001000	111	10000000



33

優先順位付符号化器(priority encoder)

– 入力: 2^n本の1ビット信号$D=(D_{2^n-1},\ldots D_0)$
– 出力: n ビット信号$Q=(Q_{n-1},\ldots,Q_0)$:

✓通常の符号化器は入力に1は1個のみ

✧入力に複数の1がある場合
  • 通常の符号化器 : ドントケア
  • 優先順位付符号化器 : 上位ビットを優先

✧入力に1が1つも無い場合
  • 通常の符号化器 : ドントケア
  • 優先順位付符号化器 : ドントケア

34

2ビット優先順位付符号化器の真理値表

$D_3D_2D_1D_0$	P	Q	$D_3D_2D_1D_0$	P	Q
0 0 0 0	-	-	1 0 0 0	3	3
0 0 0 1	0	0	1 0 0 1	3	-
0 0 1 0	1	1	1 0 1 0	3	-
0 0 1 1	1	-	1 0 1 1	3	-
0 1 0 0	2	2	1 1 0 0	3	-
0 1 0 1	2	-	1 1 0 1	3	-
0 1 1 0	2	-	1 1 1 0	3	-
0 1 1 1	2	-	1 1 1 1	3	-

P :優先順位付符号化器　Q : 符号化器

35

加算器(adder)

■ 2入力の和を計算
  – 入力 : 1桁の算術変数 X, Y
  – 出力 : S　$X+Y$ の1桁め
    C_{OUT} $X+Y$ の上位桁への繰り上がり

例 : $X=7$, $Y=8$ (10進数)
  • $S=5$
  • $C_{OUT}=1$

36

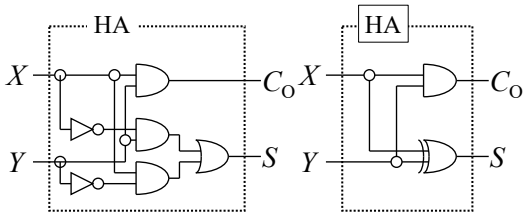
1ビット半加算器(half adder)

– 入力 : 1ビット変数 X, Y
– 出力 : S　$X+Y$ の1ビットめ
　　C_{OUT} $X+Y$ の上位ビットへの繰り上がり

$X\ Y$	C_O	S
0 0	0	0
0 1	0	1
1 0	0	1
1 1	1	0

37

1ビット半加算器の設計



38

1ビット全加算器(full adder)

– 入力 : 1ビット変数 X, Y
　　C_{IN}　下位ビットからの繰り上がり
– 出力 : S　$X+Y$ の1ビットめ
　　C_{OUT} $X+Y$ の上位ビットへの繰り上がり

$X\ Y$	C_I	C_O	S
0 0	0	0	0
	1	0	1
0 1	0	0	1
	1	1	0
1 0	0	0	1
	1	1	0
1 1	0	1	0
	1	1	1

39

1ビット全加算器の論理関数

S

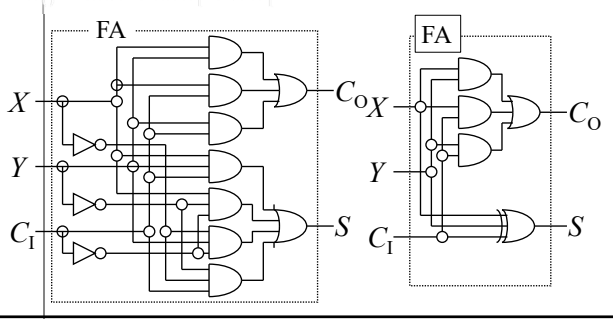
C_I \ $X\ Y$	0 0	0 1	1 1	1 0
0	0	1	0	1
1	1	0	1	0

C_O

C_I \ $X\ Y$	0 0	0 1	1 1	1 0
0	0	0	1	0
1	0	1	1	1

40

1ビット全加算器の設計



41

半加算モジュールを用いた全加算器

- 半加算器
- 全加算器

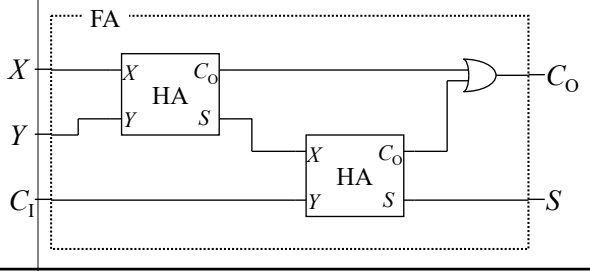
42

全加算器の設計

- 半加算器モジュール2個を用いて全加算器を設計

$S^{FA}(X,Y,C_I) = S^{HA}(S^{HA}(X,Y),C_I)$

$C_O^{FA}(X,Y,C_I) = C_O^{HA}(X,Y) + C_O^{HA}(S^{HA}(X,Y),C_I)$



43

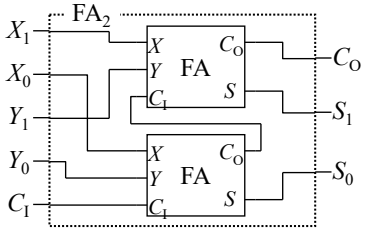
2ビット加算器

– 入力 : 2ビット変数 $X=(X_1,X_0)$, $Y=(Y_1,Y_0)$

C_{IN} 下位ビットからの繰り上がり

– 出力 : $S=(S_1,S_0)$, $X+Y$ の1,2ビットめ

C_{OUT} $X+Y$ の上位ビットへの繰り上がり



44

多数決器

– 入力 : n 変数 $X_1,X_2,\ldots,X_n$

– 出力 : $Z=$ { 1 (入力のうち1が半分以上) / 0 (入力のうち1が半分未満)

✓ 例 3変数多数決器

$X_1\ X_2\ X_3$	Z
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	1

$X_1\ X_2\ X_3$	Z
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	1

45

多数決器の論理関数

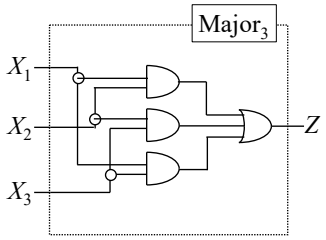
X_3 \ $X_1\ X_2$	0 0	0 1	1 1	1 0
0	0	0	1	0
1	0	1	1	1

$$Z = X_1 \cdot X_2 + X_2 \cdot X_3 + X_3 \cdot X_1$$

46

多数決回路

$$Z = X_1 \cdot X_2 + X_2 \cdot X_3 + X_3 \cdot X_1$$



47

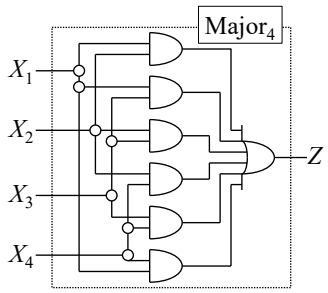
多数決器の論理関数(4変数)

$X_3\ X_4$ \ $X_1\ X_2$	0 0	0 1	1 1	1 0
0 0	0	0	1	0
0 1	0	1	1	1
1 1	1	1	1	1
1 0	0	1	1	1

$$Z = X_1X_2 + X_2X_3 + X_3X_4 + X_4X_1 + X_1X_3 + X_2X_4$$

48

多数決回路(4変数)



49

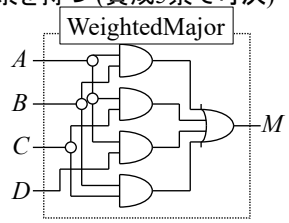
例題：重み付き多数決器

■ 各人が持つ票の重みが違う多数決

例題

– A 4票,B 3票,C 2票,D 1票を持つ (賛成5票で可決)

AB \ CD	00	01	11	10
00	0	3	**7**	4
01	1	4	**8**	**5**
11	3	**6**	**10**	**7**
10	2	**5**	**9**	**6**



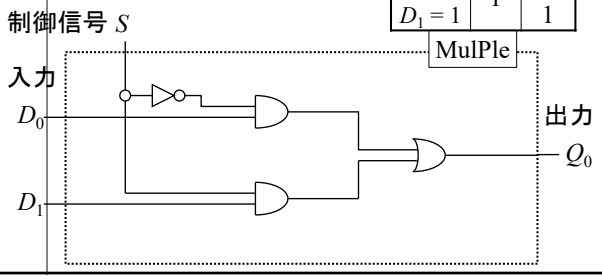
50

演習問題：選択器の設計

■ 1ビット選択器を設計せよ

$Q = \overline{S} \cdot D_0 + S \cdot D_1$

D_i	S	Q
$D_0 = 0$	0	0
$D_0 = 1$		1
$D_1 = 0$	1	0
$D_1 = 1$		1



51

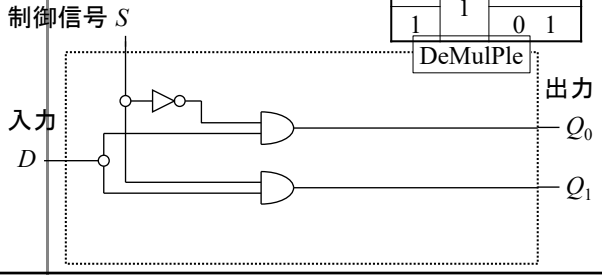
演習問題：分配器の設計

■ 1ビット分配器を設計せよ

$Q_0 = \overline{S} \cdot D$

$Q_1 = S \cdot D$

D	S	Q_0 Q_1
0	0	0 0
1		1 0
0	1	0 0
1		0 1



52

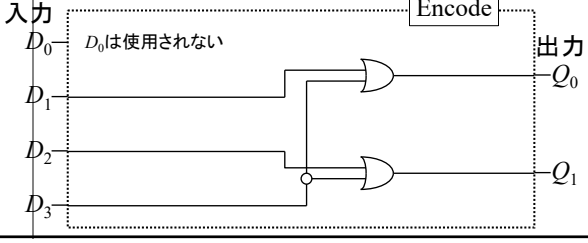
演習問題：符号化器の設計

■ 2ビット符号化器を設計せよ

$Q_0 = D_1 + D_3$

$Q_1 = D_2 + D_3$

$D_3 D_2 D_1 D_0$	Q_1 Q_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



53

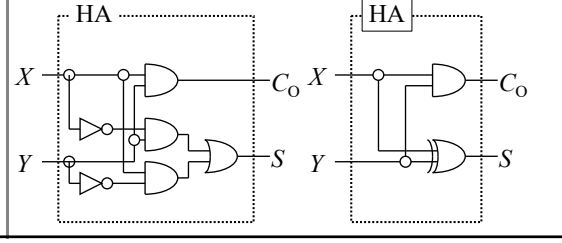
演習問題：半加算器の設計

■ 1ビット半加算器を設計せよ

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

$C_O = X \cdot Y$

X Y	C_O	S
0 0	0	0
0 1	0	1
1 0	0	1
1 1	1	0



54

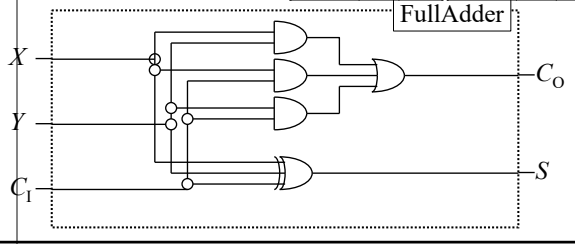
演習問題：全加算器の設計

- 1ビット全加算器を設計せよ

$S =$

$C_O =$

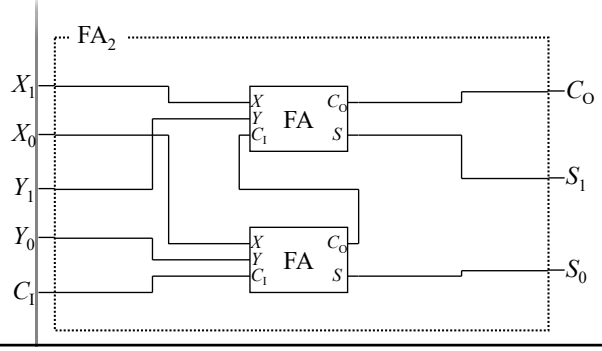
XYC_I	C_O	S	XYC_I	C_O	S
000	0	0	100	0	1
001	0	1	101	1	0
010	0	1	110	1	0
011	1	0	111	1	1



55

演習問題：2ビット全加算器の設計

- 1ビット全加算器モジュール2個を用いて2ビット全加算器を設計せよ



56