

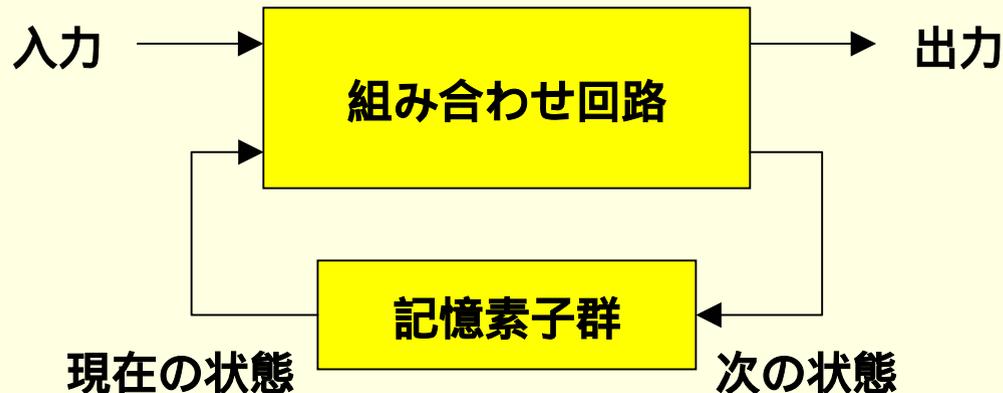
LSI概論(7)

組み合わせ論理回路

組み合わせ回路と順序回路

■ 論理回路

- 組み合わせ回路
回路の出力がそのときの入力値で決まる回路
- 順序回路
回路の出力は入力値だけで決まらず、回路の内部状態(記憶素子がある)に依存する回路



組み合わせ論理回路の構成

- 出力 Y_i は入力 X_i の関数式で表される

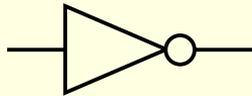
$$Y_i = F_i(X_1, X_2, X_3, \dots, X_n)$$



基本回路(復習)

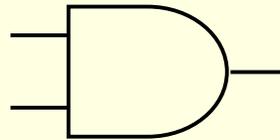
- 下記基本回路を組み合わせて回路を構成する

NOT



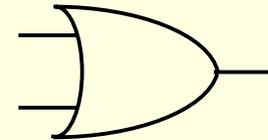
\overline{A}

AND



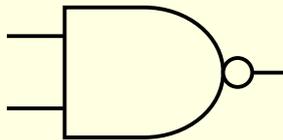
$A \cdot B$

OR



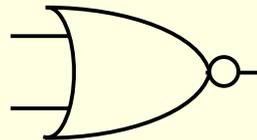
$A + B$

NAND



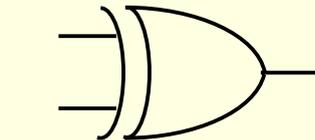
$\overline{A \cdot B}$

NOR



$\overline{A + B}$

XOR



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

真理値表, トランジスタ
を使った回路図はもう
理解できていますね!

2進数(復習)

- デジタルLSIの中は2値論理

Gnd=0, Vdd=1

- 演算は2進数を使用する

$$1001 \text{ (2進数)} = 2^3 + 2^0 = 9 \text{ (10進数)}$$

- 2進数の加算(正の数)

- $0 + 0 = 0$

- $0 + 1 = 1$

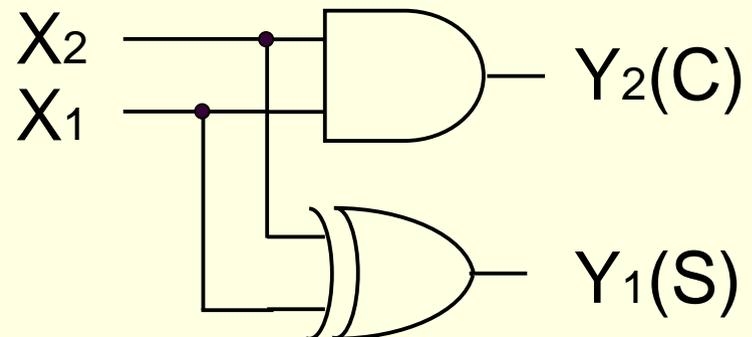
- $1 + 0 = 1$

- $1 + 1 = 10$

組み合わせ回路(1) 半加算器

- 2進数の足し算は下記の真理値表で示される
- 真理値表を論理式で表すと
 - $Y_1 = X_1 \cdot X_2 + X_1 \cdot \overline{X_2}$, $Y_2 = X_1 \cdot X_2$
- Y_1 はXOR, Y_2 はANDで実現できる
- したがって, 下記回路で2進の加算(半加算器: Half Adder)は実現できる.
- Y_1 は和(sum)ということでS, Y_2 は桁上げということでC(carry)で表現されることが多い

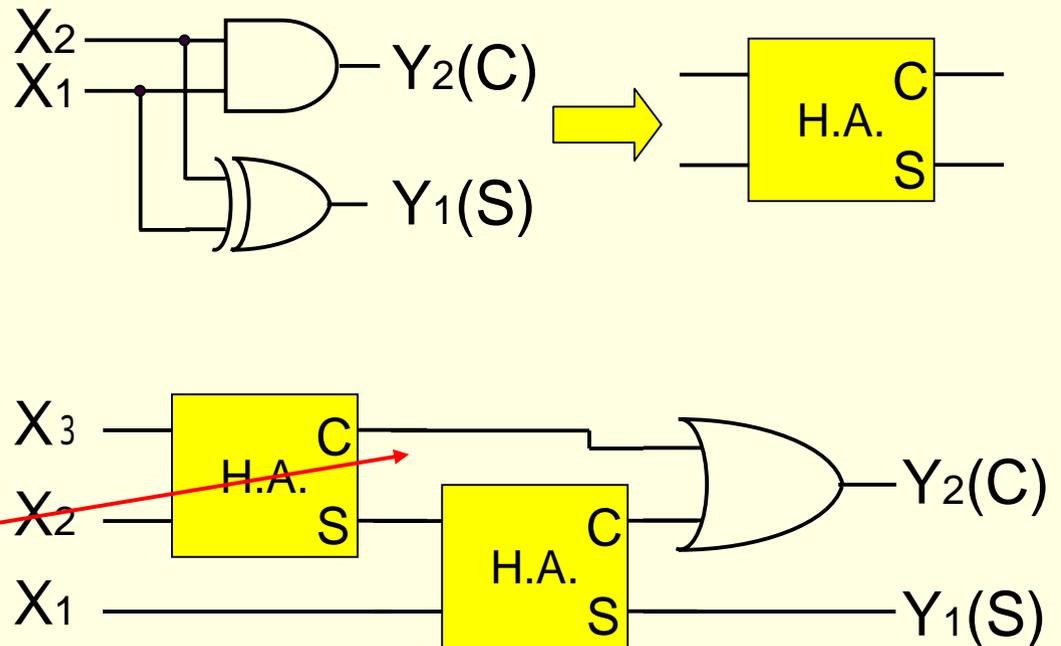
X_2	X_1	Y_2	Y_1
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



組み合わせ回路(2) 全加算器

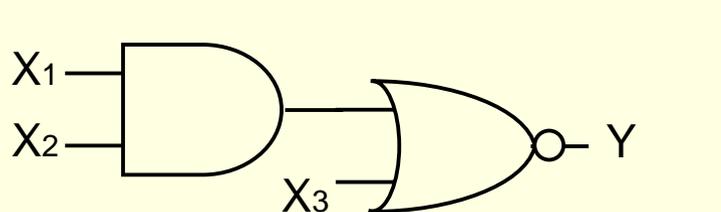
- 全加算器とは半加算器の入力が2入力から3入力となり、桁上げ入力も加算できるようにしている。
- 半加算器2個とORゲートで実現できる。

X_3	X_2	X_1	Y_2	Y_1	
0	+	0	+	0 = 0	0
0	+	0	+	1 = 0	1
0	+	1	+	0 = 0	1
0	+	1	+	1 = 1	0
1	+	0	+	0 = 0	1
1	+	0	+	1 = 1	0
1	+	1	+	0 = 1	0
1	+	1	+	1 = 1	1

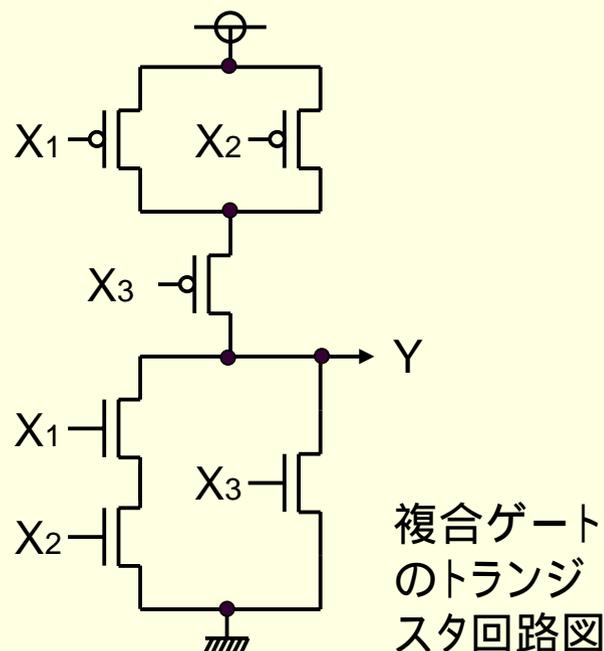
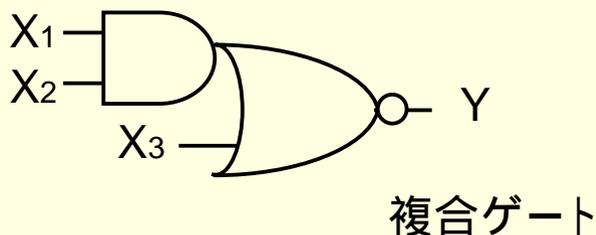
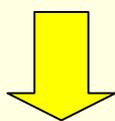


複合ゲート

- 下図のような積和型演算等の,いくつかの論理ゲートを組み合わせた論理を実現するのに,複合ゲートを使用する方法がある.
- 一般に,複合ゲートの方がスイッチングに要する遅延が少なく,レイアウト面積を小さくできる.



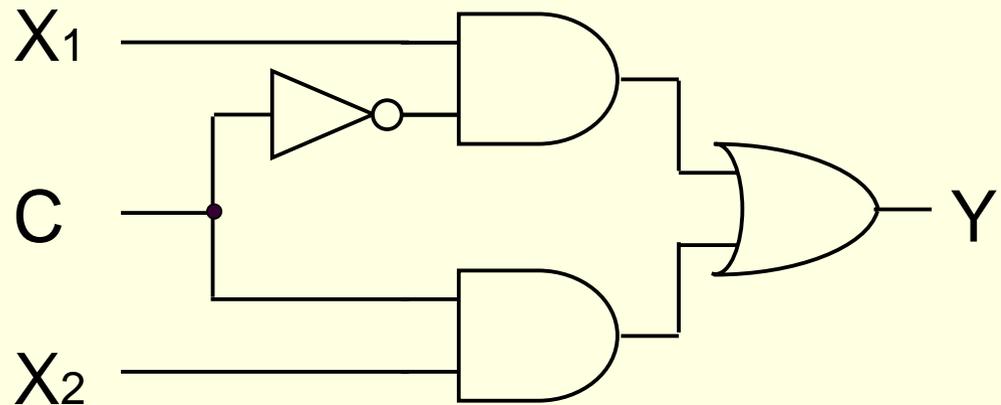
$$Y = \overline{X_1 \cdot X_2} + X_3$$



組み合わせ回路(3)マルチプレクサ

- セレクタとも呼ばれ、複数のデータの中から条件によって1つのデータを選択する回路
- C が0のとき $Y=X_1$, C が1のとき $Y=X_2$
 $Y = \bar{C} \cdot X_1 + C \cdot X_2$

C	X ₂	X ₁	Y
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

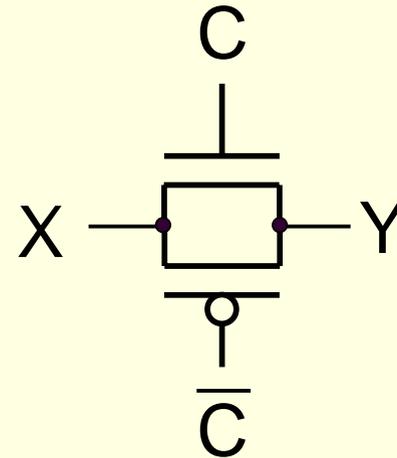


トランスマッションゲート(相補型)

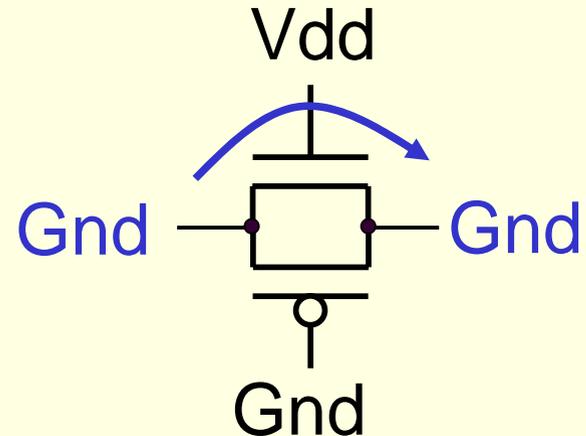
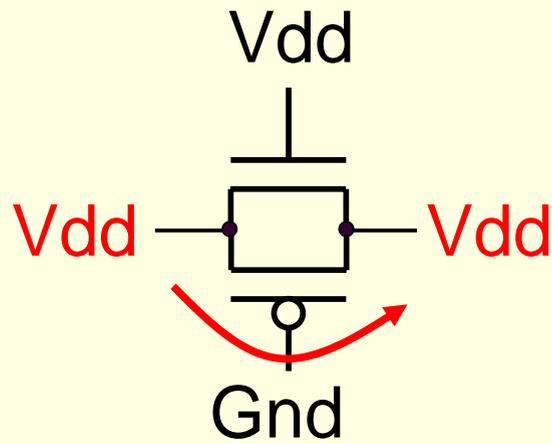
- PMOSとNMOSを並列に配置するとトランスマッションゲートができる

C	Y
0	Z
1	X

ハイインピーダンスと
いい出力が固定され
ないことを示す。



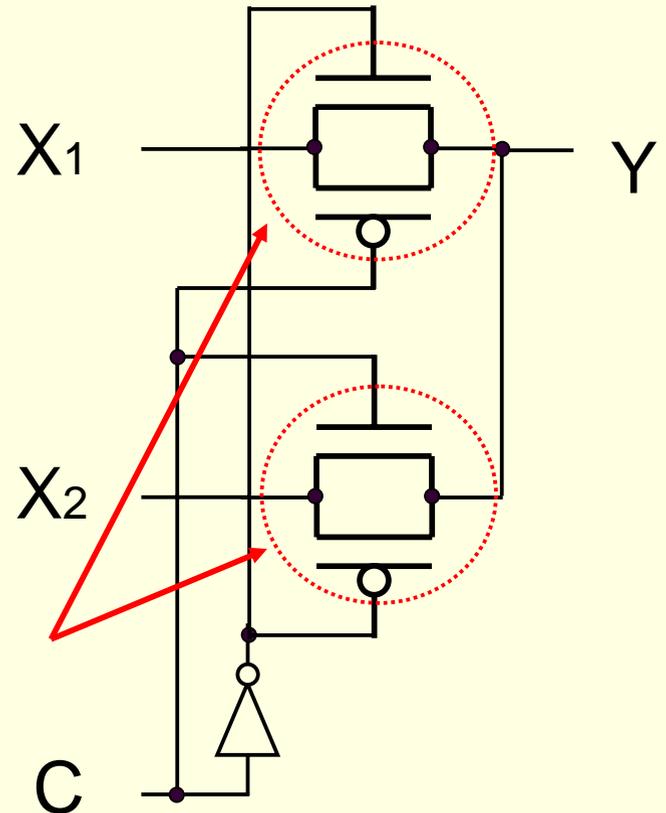
- VddデータはPMOS, GndデータはNMOSがデータを転送.



トランスマッションゲートで構成したマルチプレクサ

- マルチプレクサを実現するのにトランスマッションゲートを使用するとトランジスタ数を削減できる。(特に多入力の場合)
- $Y = \overline{C} \cdot X_1 + C \cdot X_2$
- ただし、出力Yの駆動能力は低いので注意する

どちらかが必ずONにならないと出力Yはハイインピーダンスのままになる。



講義内容

- | | | | |
|------|------------|----|-------------------------|
| (1) | 9月27日 (土) | 山内 | 導入(MELPEC, 講義ガイダンス, 実物) |
| (2) | 9月29日 (月) | 藤野 | MOSTランジスタとCMOS回路 |
| (3) | 10月 6日 (月) | 藤野 | 論理回路と論理ゲート |
| (4) | 10月11日 (土) | 藤野 | LSIの製造フロー |
| (5) | 10月20日 (月) | 藤野 | 基本レイアウト |
| (6) | 10月27日 (月) | 藤野 | CMOS回路の電気特性 |
| (7) | 11月10日 (月) | 藤野 | 組み合わせ論理回路 |
| (8) | 11月17日 (月) | 山内 | 演算回路1 (データ形式) |
| (9) | 12月 1日 (月) | 山内 | 演算回路2 (演算器) |
| (10) | 12月 8日 (月) | 山内 | 記憶回路(レジスタとレジスタファイル) |
| (11) | 12月15日 (月) | 藤野 | 大規模記憶回路(メモリ) |
| (12) | 12月22日 (月) | 藤野 | LSIの実装技術 |
| (13) | 12月23日 (火) | 山内 | EDAによるLSIの設計と製造フロー 1 |
| (14) | 1月 5日 (月) | 山内 | EDAによるLSIの設計と製造フロー 2 |