

# 集積回路設計 (INTEGRATED CIRCUIT DESIGN) 第8回

2020/05/29

# 階層設計と構造設計（4章）

2

- ランダム論理回路
  - ▣ 最小チップ面積で性能の良い論理設計
  - ▣ ALU, 加算器, レジスタ等の基本論理回路ブロック
    - ゲートレベル設計
    - トランジスタ設計
- 基本論理回路ブロック
  - ▣ 階層設計：基本論理回路ブロックを組み合わせて大規模なシステムを設計
    - ALUやレジスタ等 → プロセッサ
    - 基本セルやマクロセル → ゲートアレー
- PLA/ROM
  - ▣ PLA（プログラム可能な論理アレー）
  - ▣ レイアウト（チップ上の配置・配線）も考慮した規則的な構造

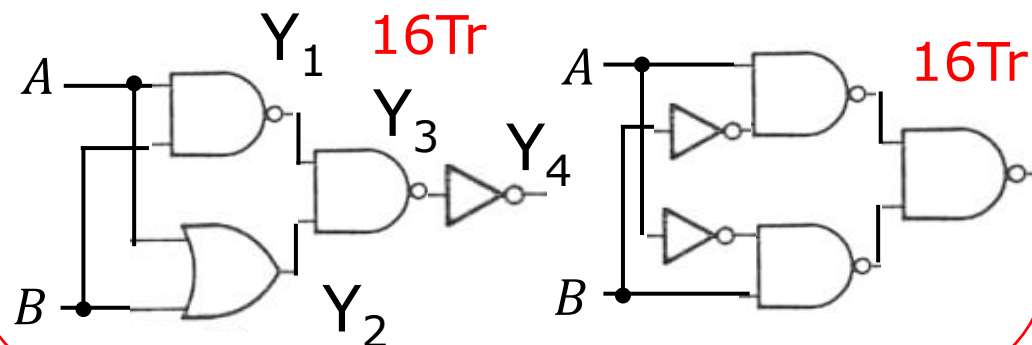
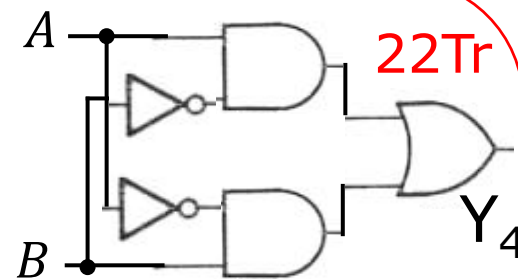
# 論理設計

3

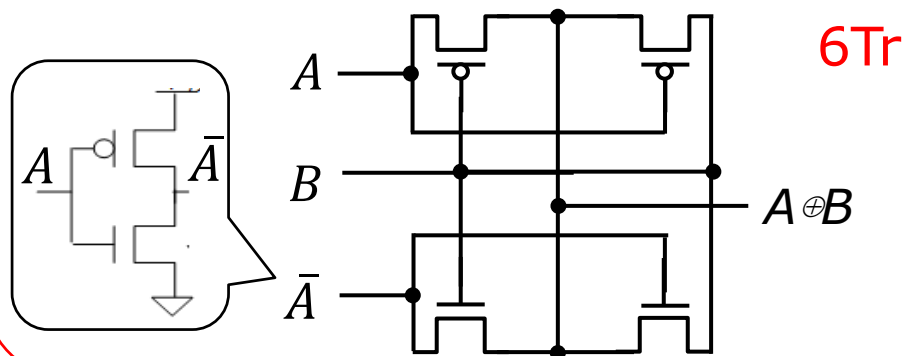
## 例) EXOR

A	B	$A \oplus B$	$Y_1$	$Y_2$	$Y_3$	$Y_4$
0	0	0	1	0	1	0
0	1	1	1	1	0	1
1	0	1	1	1	0	1
1	1	0	0	1	1	0

### ゲート論理設計



### トランジスタ設計



高  
↑  
抽象度  
↓  
低

IP設計

セル設計

ゲートレベル設計

トランジスタ設計

大  
↑  
規模  
↓  
小

# データパスと制御論理回路

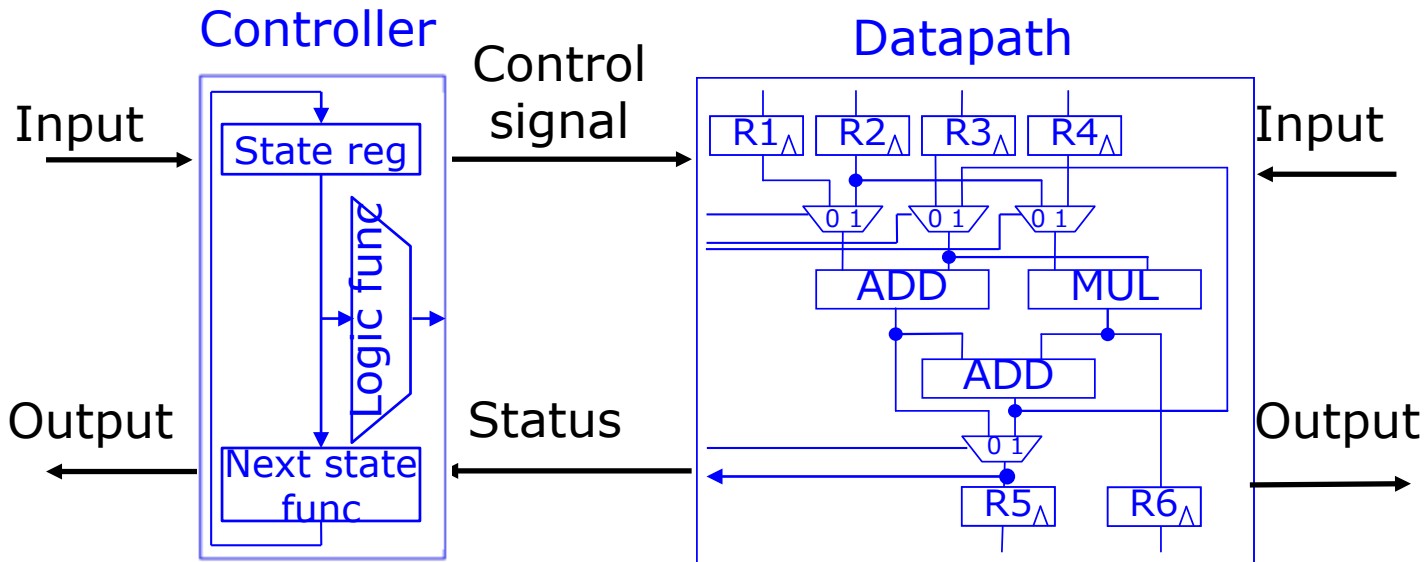
4

## □ データパス

- データ処理を行うための論理回路
- 個々の構成要素はランダム論理回路で設計
  - 算術論理演算（加減算・乗算・論理演算）
  - 記憶機能（メモリ・レジスタ）

## □ 制御論理回路

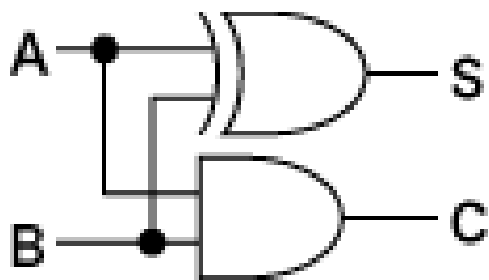
- データの流れを正しく制御する回路
- ランダム論理回路・PLA/ROMで設計



# 1-bit 加算器

5

- 半加算器 (=AND1個+EXOR1個)
  - $S = A \oplus B$ ,  $C = AB$
  - 下位ビットからの桁上げ無し
  - または全加算器 (次スライド) で  $C_{-1} = 0$



A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

# 1-bit 加算器

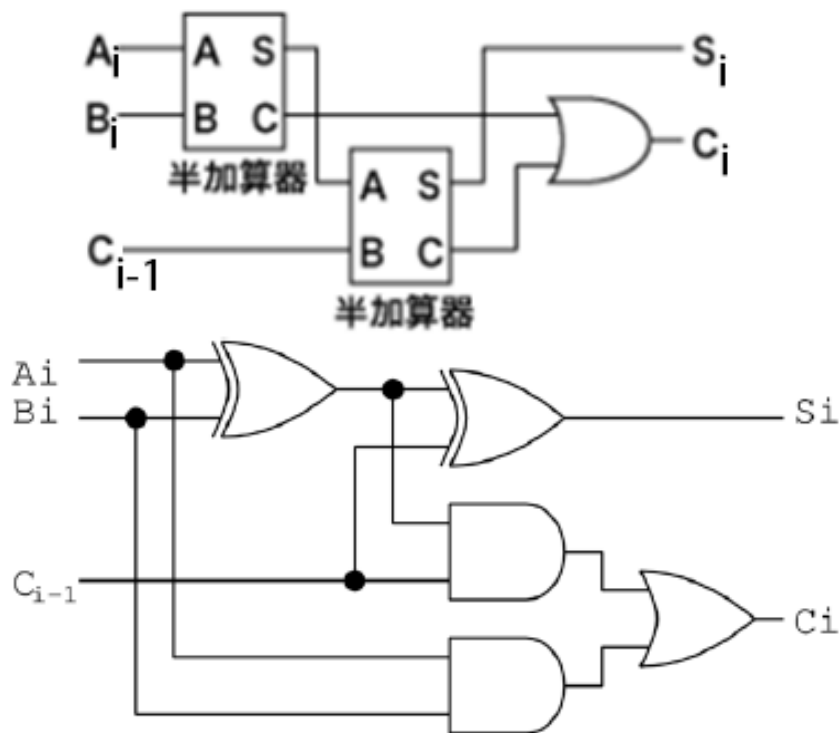
6

## 全加算器（＝半加算器2個＋OR1個）

$$S_i = (A_i \oplus B_i) \oplus C_i = A_i \oplus B_i \oplus C_{i-1}$$

$$C_i = A_i B_i + (A_i \oplus B_i) C_{i-1}$$

■ 下位ビットからの桁上げ

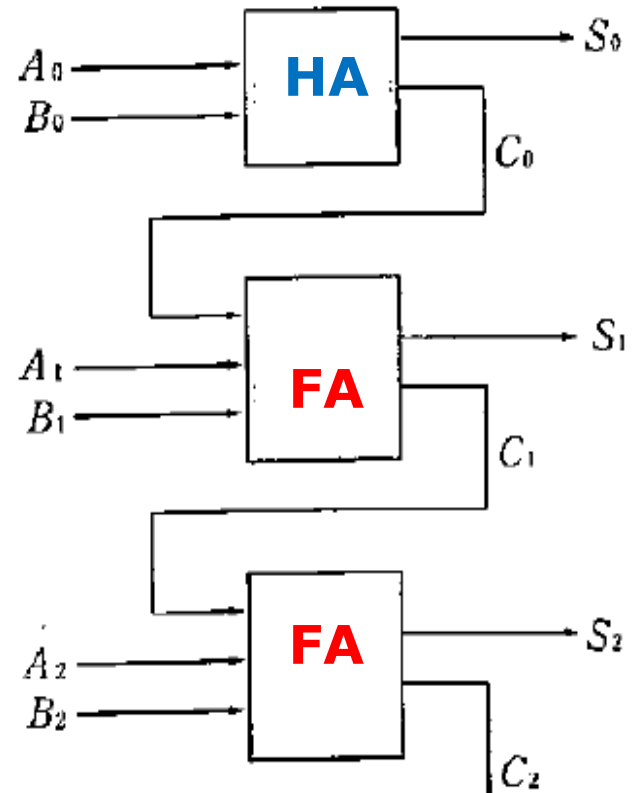
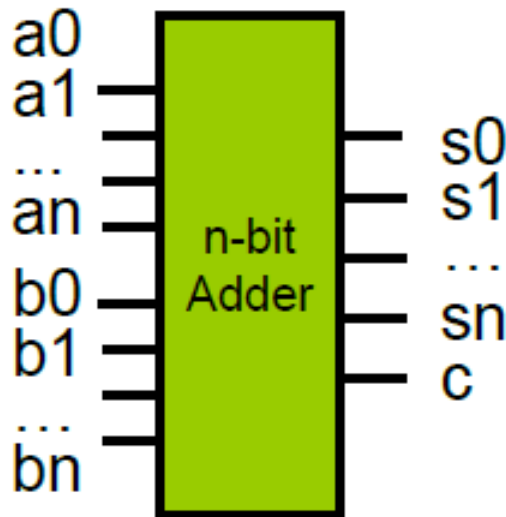


$A_i$	$B_i$	$C_{i-1}$	$C_i$	$S_i$
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

# n-bit 加算器

7

- $(A_n \dots A_i \dots A_1 A_0) + (B_n \dots B_i \dots B_1 B_0)$ 
  - ▣ 1-bit 加算器を接続（階層設計）
- Ripple Carry Adder (RCA)
  - ▣ 回路面積：小
  - ▣ 遅延：大 ( $A_0(B_0) \rightarrow C$ )



# n-bit 加算器

8

## □ Carry Look Ahead (CLA)

### □ 桁上げ先読み

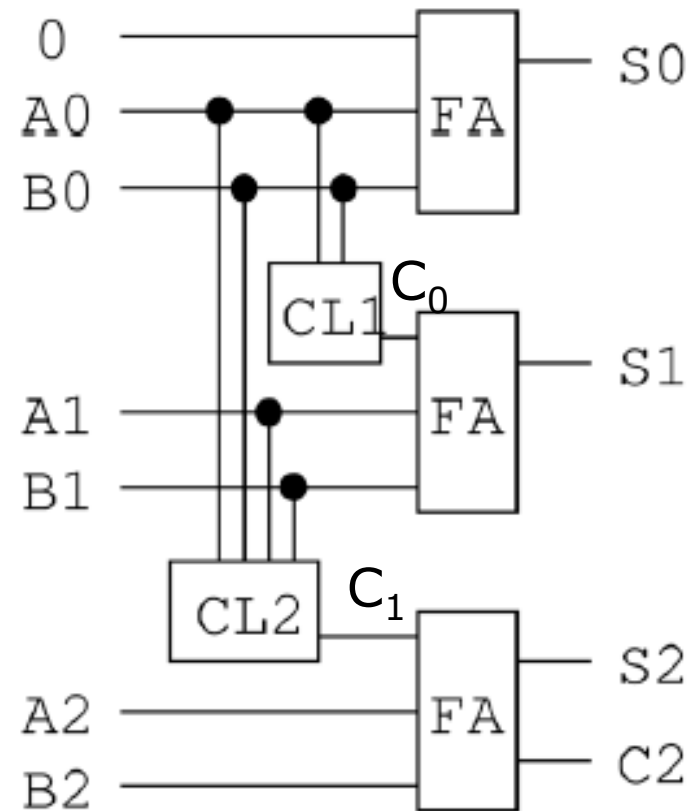
■  $CL_1 = A_0B_0$

■  $CL_2 = A_1B_1 + C_0A_1 + C_0B_1$   
 $= A_1B_1 + A_0B_0A_1 + A_0B_0B_1$

桁上げが起こる条件：  
3入力の内2つ以上が1

### □ 回路面積：大

### □ 遅延：小





# ALU (Arithmetic Logic Unit)

9

## □ 算術演算回路

### ▣ 一般的なプロセッサに組み込まれている演算器

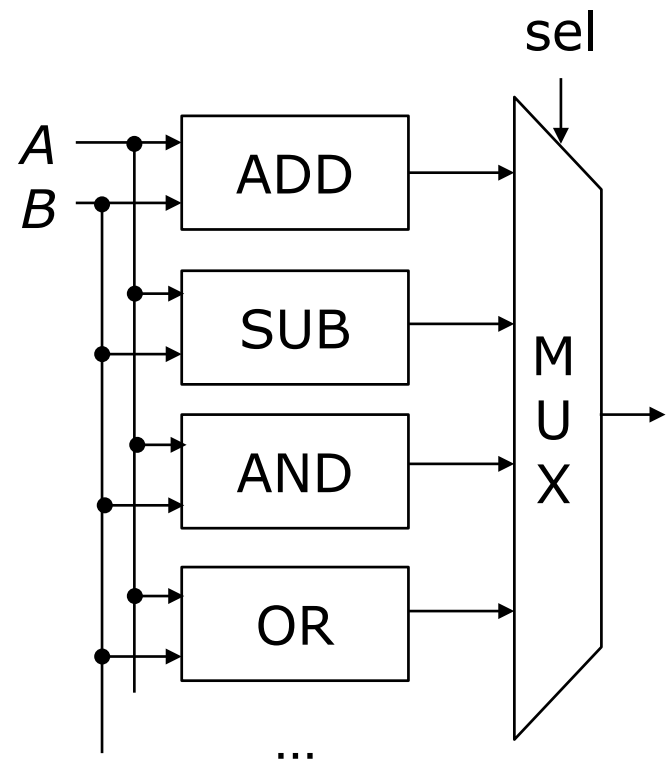
- 加算
- 減算
- 論理演算
- etc...

## □ 設計方法

### ▣ 独立に設計 & 組み合わせることはいらない

- 面積
- 電力
- etc...

### ▣ 共通部分を抽出して計算結果は使い回すように構築



# ALU (Arithmetic Logic Unit)

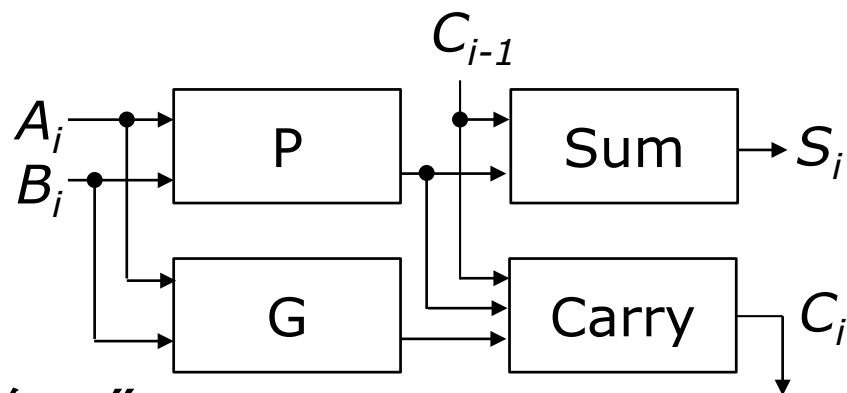
10

## □ ALUの基本構造

- P, Gブロックの機能を制御信号で切り替え  
→異なる演算

## □ 加算と減算

- 加算:  $A+B$
- 減算:  $A-B=A+(-B)$ 
  - “負の値をどう表現するか?”
  - 1の補数, 2の補数



# ALU (Arithmetic Logic Unit)

11

## □ Ripple Carry Adder (RCA)

□ 例)  $1-2=1+(-2)=-1$

- 符号+数値:  $001+110=111(=-3) \rightarrow 101(=-1)$ にならない
- 1の補数:  $001+101=110$
- 2の補数:  $001+110=111$

## □ 1の補数

- 正の値をビット反転
- $-(2^n-1) \sim 2^n-1$
- “Aの2進表現”+ “Aの1の補数の2進表現”で**すべてのビットが1**  
→ オール1にするために補う数=1の補数

## □ 2の補数

- 正の値をビット反転+1
- $-2^n \sim 2^n-1$
- “Aの2進表現”+ “Aの2の補数の2進表現”で **$2^n$**  →  $2^n$ にするために補う数=2の補数

			符号なし	符号あり		
$a_2$	$a_1$	$a_0$		符号+数値	1の補数	2の補数
0	0	0	0	0	0	0
0	0	1	1	1	1	1
0	1	0	2	2	2	2
0	1	1	3	3	3	3
1	0	0	4	-0	-3	-4
1	0	1	5	-1	-2	-3
1	1	0	6	-2	-1	-2
1	1	1	7	-3	-0	-1

# ALU (Arithmetic Logic Unit)

12

## □ 加算

$$\square S_i = A_i \oplus B_i \oplus C_{i-1} = P_i \oplus C_{i-1}$$

$$\square C_i = A_i B_i + (A_i \oplus B_i) C_{i-1} = G_i + P_i C_{i-1} \quad (C_{-1} = 0)$$

S: 1が奇数個で1

C: 1が2つ以上  
で1 (キャリー)  
= 多数決

## □ 減算

$$\square A - B = A + (-B)$$

$$\square S_i = P_i \oplus C_{i-1}$$

$$\square C_i = G_i + P_i C_{i-1} \quad (C_{-1} = 1; 2の補数)$$

$$P_i = A_i \oplus \overline{B_i}, \quad G_i = A_i \overline{B_i}$$

## □ 論理演算

$$\square P_i = (A_i \text{ と } B_i \text{ の論理演算})$$

$$\square G_i = 0$$

# Pブロック

13

	加算	減算	AND	OR
P	$A_i \oplus B_i$	$A_i \oplus \bar{B}_i$	$A_i B_i$	$A_i + B_i$
$C_{-1}$	0	1	0	0

## □ Pブロック:CMOSドミノ回路

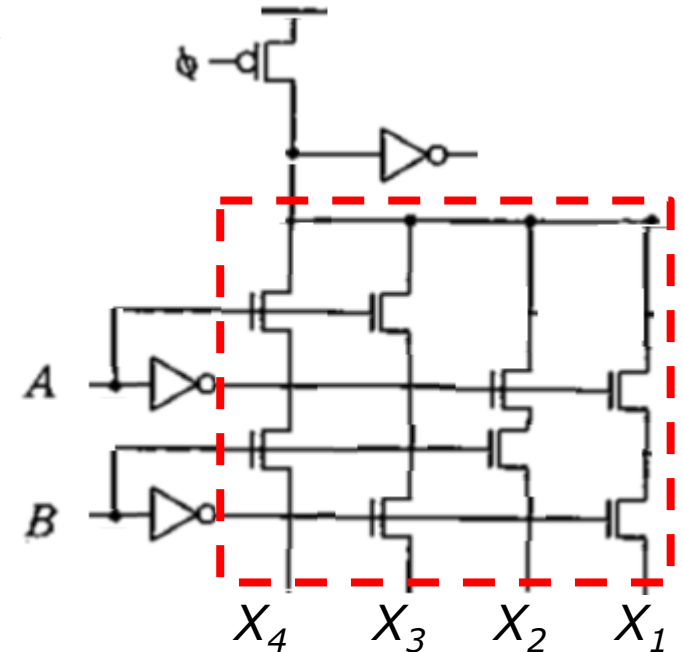
■ 加算～ORを個々に作って繋げると膨大

### ■ ドミノ回路

■ メイン部分：8 Tr

■ プリチャージ：1 Tr

■ 入出力：3 NOT



# Pブロックの制御信号 (1/2)

14

## □ 制御信号 $X_i (i=1\sim 4)$ を操作

### □ $\phi=0$ の時

- 点Eはプリチャージ (論理1;  $V_{DD}$ )
- 制御入力 $X_i$ も  $V_{DD}$  (放電パスを形成しない)

### □ $\phi=1$ の時

- $(A,B)=(0,0) \rightarrow$  パス①
- $(A,B)=(0,1) \rightarrow$  パス②
- $(A,B)=(1,0) \rightarrow$  パス③
- $(A,B)=(1,1) \rightarrow$  パス④

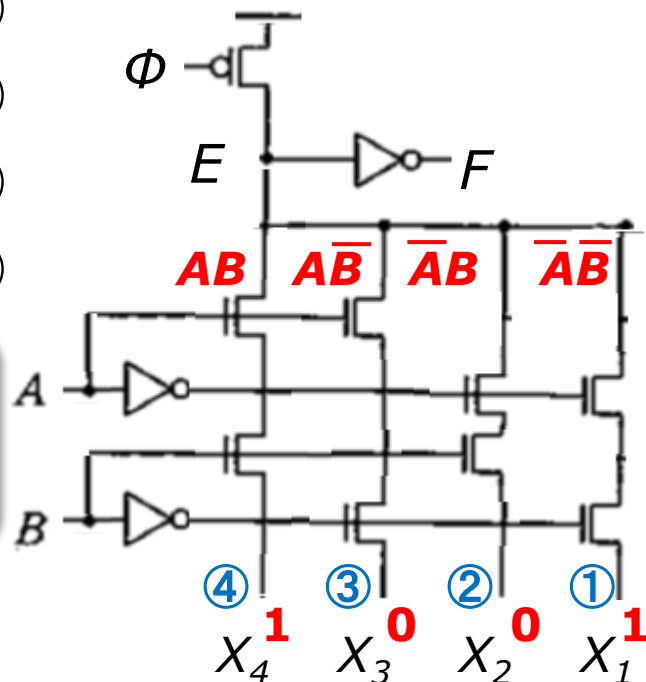
加算(1001)の時  
制御信号1  
→点Eの電荷保持  
→ $F=0$

加算(1001)の時  
制御信号0  
→点Eの電荷放電  
→ $F=1$

$$P_i = A_i \oplus B_i$$

加算( $A+B$ )のとき

$$P_i = A_i \oplus B_i = \bar{A}_i B_i + A_i \bar{B}_i$$



# Pブロックの制御信号 (2/2)

15

	加算	減算	AND	OR
P	$A_i \oplus B_i$	$A_i \oplus \bar{B}_i$	$A_i B_i$	$A_i + B_i$
$C_{-1}$	0	1	0	0
Pの制御信号 ( $X_4 X_3 X_2 X_1$ )	1001	0110	0111	0001

## □ Pブロック:CMOSドミノ回路

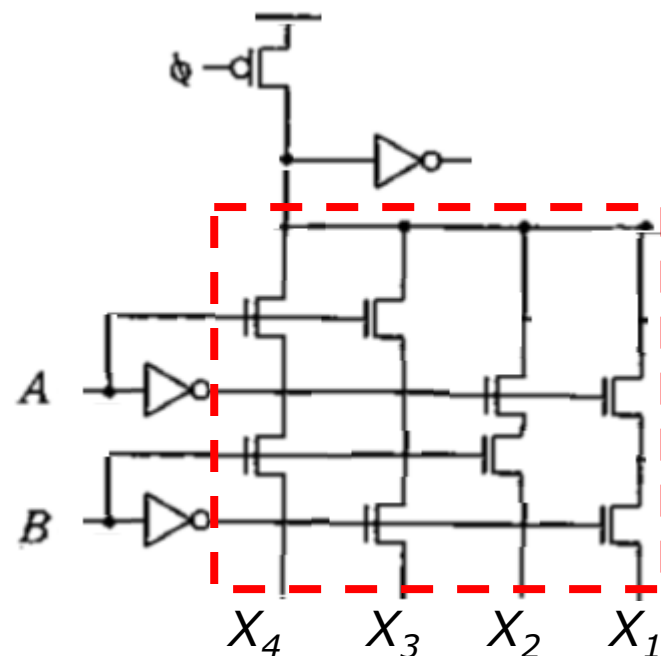
■ 加算～ORを個々に作って繋げると膨大

### ■ ドミノ回路

■ メイン部分：8 Tr

■ プリチャージ：1 Tr

■ 入出力：3 NOT



# キャリー生成論理

16

## □ キャリー生成論理

- $C_0 = C_{-1}P_0 + G_0$

- $C_1 = C_0P_1 + G_1$

- $C_2 = C_1P_2 + G_2$

- ...

クリティカルパス( $C_{-1} \sim C_n$ )

□ “クリティカルパスを短く抑えたい”

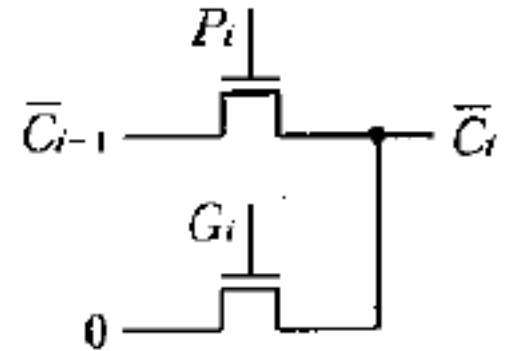


# マンチェスターキャリーチェーン

17

## □ パス論理とプリチャージ方式を使ったCMOSドミノ論理回路で実現

- $\overline{C}_{i-1}=0, P_i=1$  のとき  $\overline{C}_i=0$
- $G_i=1$  のとき  $\overline{C}_i=0$
- それ以外の組み合わせの時に1になるようにプリチャージ方式と組み合わせる



## □ 4ビットの例

