

## 第2章 コンピュータの設計

論理回路は理解できた！

しかし、コンピュータの設計ができない

なぜか

レジスタトランスファレベルの理解が足りない

制御されるもの：データパス

演算器、レジスタなどの機能装置とデータ通信路

制御するもの：制御装置

複数の状態

状態遷移して動作

各状態で実行する操作が定義

## 2.1 ブール代数

- $\{0, 1\}$ 上の代数系

基本演算

$\cdot$  : AND,  $+$  : OR,  $\sim$  : NOT

$$0 \cdot 0 = 0 \quad 1 + 1 = 1$$

$$0 \cdot 1 = 0 \quad 1 + 0 = 1$$

$$1 \cdot 0 = 0 \quad 0 + 1 = 1$$

$$1 \cdot 1 = 1 \quad 0 + 0 = 0$$

$$\sim 0 = 1 \quad \sim 1 = 0$$

- 定理

ドモルガンの定理

結合則、交換則、分配則など

# ブール代数の定理

## ドモルガンの双対定理

$$\overline{X_1 \cdot X_0} = \overline{X_1} + \overline{X_0}$$

$$\overline{X_1 + X_0} = \overline{X_1} \cdot \overline{X_0}$$

$$X \cdot 0 = 0, X + 1 = 1$$

$$X \cdot 1 = X, X + 0 = X$$

$$X \cdot X = X, X + X = X$$

$$X \cdot \overline{X} = \overline{X} \cdot X = 0, X + \overline{X} = \overline{X} + X = 1$$

## 結合則

$$(X_2 \cdot X_1) \cdot X_0 = X_2 \cdot (X_1 \cdot X_0)$$

$$(X_2 + X_1) + X_0 = X_2 + (X_1 + X_0)$$

## 吸收則

$$X_1 + X_1 \cdot X_0 = X_1$$

$$X_1 \cdot (X_1 + X_0) = X_1$$

## 交換則

$$X \cdot Y = Y \cdot X, X + Y = Y + X$$

## 分配則

$$X_2 \cdot (X_1 + X_0) = X_2 \cdot X_1 + X_2 \cdot X_0$$

$$X_2 + X_1 \cdot X_0 = (X_2 + X_1) \cdot (X_2 + X_0)$$

## 証明の仕方: 場合の数え上げ

$X_1$	$X_0$	$\sim(X_1 \cdot X_0)$	$\bar{X}_1 + \bar{X}_0$	$\sim(X_1 + X_0)$	$\bar{X}_1 \cdot \bar{X}_0$
0	0	1	1	1	1
0	1	1	1	0	0
1	0	1	1	0	0
1	1	0	0	0	0

一致
一致

図 2.1 ドモルガンの定理

$X_2$	$X_1$	$X_0$	$X_2 \cdot (X_1 + X_0)$	$X_2 \cdot X_1 + X_2 \cdot X_0$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

図 2.2 分配則の証明

## 2.2 組合わせ回路

### 最小項と積和標準形

$$\bar{X}_{n-1} \cdot \bar{X}_{n-2} \cdots \bar{X}_1 \cdot \bar{X}_0, \quad \bar{X}_{n-1} \cdot \bar{X}_{n-2} \cdots \bar{X}_1 \cdot X_0$$

$$\bar{X}_{n-1} \cdot \bar{X}_{n-2} \cdots X_1 \cdot \bar{X}_0, \quad \bar{X}_{n-1} \cdot \bar{X}_{n-2} \cdots X_1 \cdot X_0$$

$$\vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots$$

$$X_{n-1} \cdot X_{n-2} \cdots X_1 \cdot \bar{X}_0, \quad X_{n-1} \cdot X_{n-2} \cdots X_1 \cdot X_0$$

## N変数論理関数とその個数

$$\begin{aligned} F = & F(0, 0, \dots, 0, 0) \cdot \bar{X}_{n-1} \cdot \bar{X}_{n-2} \cdot \dots \cdot \bar{X}_1 \cdot \bar{X}_0 + \\ & F(0, 0, \dots, 0, 1) \cdot \bar{X}_{n-1} \cdot \bar{X}_{n-2} \cdot \dots \cdot \bar{X}_1 \cdot X_0 + \\ & F(0, 0, \dots, 1, 0) \cdot \bar{X}_{n-1} \cdot \bar{X}_{n-2} \cdot \dots \cdot X_1 \cdot \bar{X}_0 + \\ & F(0, 0, \dots, 1, 1) \cdot \bar{X}_{n-1} \cdot \bar{X}_{n-2} \cdot \dots \cdot X_1 \cdot X_0 + \\ & F(1, 1, \dots, 1, 0) \cdot X_{n-1} \cdot X_{n-2} \cdot \dots \cdot X_1 \cdot \bar{X}_0 + \\ & F(1, 1, \dots, 1, 1) \cdot X_{n-1} \cdot X_{n-2} \cdot \dots \cdot X_1 \cdot X_0 \end{aligned}$$

最小項の数:  $2^n$

その係数: 0, 1 の2通り

論理関数の個数:  $2$  の  $2^n$



# 2変数の論理関数

$$F = f_0 \overline{X_1} \overline{X_0} + f_1 \overline{X_1} X_0 + f_2 X_1 \overline{X_0} + f_3 X_1 X_0$$

$$F = f_0 \overline{X_1} \overline{X_0} + f_1 \overline{X_1} X_0 + f_2 X_1 \overline{X_0} + f_3 X_1 X_0$$

2変数論理関数: 16通!

$f_3$	$f_2$	$f_1$	$f_0$	論理関数	
0	0	0	0	0	
0	0	0	1	$\overline{X_1} \cdot \overline{X_0} : (\sim (X_1 + X_0), \text{NOR})$	
0	0	1	0	$\overline{X_1} \cdot X_0$	
0	0	1	1	$\overline{X_1} \cdot \overline{X_0} + \overline{X_1} \cdot X_0 : (\overline{X_1})$	
0	1	0	0	$X_1 \cdot \overline{X_0}$	
0	1	0	1	$\overline{X_1} \cdot \overline{X_0} + X_1 \cdot \overline{X_0} : (\overline{X_0})$	
0	1	1	0	$\overline{X_1} \cdot X_0 + X_1 \cdot \overline{X_0} : (\text{XOR})$	
0	1	1	1	$\overline{X_1} \cdot \overline{X_0} + \overline{X_1} \cdot X_0 + X_1 \cdot \overline{X_0}$ $(\overline{X_1} + \overline{X_0}, \sim (X_1 \cdot X_0), \text{NAND})$	
1	0	0	0	$X_1 \cdot X_0 : (\text{AND})$	(2.22)
1	0	0	1	$\overline{X_1} \cdot \overline{X_0} + X_1 \cdot X_0 : (\text{Coincidence, 一致})$	
1	0	1	0	$\overline{X_1} \cdot X_0 + X_1 \cdot X_0 : (X_0)$	
1	0	1	1	$\overline{X_1} \cdot \overline{X_0} + \overline{X_1} \cdot X_0 + X_1 \cdot X_0 : (\overline{X_1} + X_0)$	
1	1	0	0	$X_1 \cdot \overline{X_0} + X_1 \cdot X_0 : (X_1)$	
1	1	0	1	$\overline{X_1} \cdot \overline{X_0} + X_1 \cdot \overline{X_0} + X_1 \cdot X_0 : (X_1 + \overline{X_0})$	
1	1	1	0	$\overline{X_1} \cdot X_0 + X_1 \cdot \overline{X_0} + X_1 \cdot X_0 : (X_1 + X_0, \text{OR})$	
1	1	1	1	1	

# 組み合わせ回路の例

入 力			出 力	
X	Y	$C_{in}$	S	$C_{out}$
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

図 2.3 加算器設計のための真理値表

真理値表  
簡単化

積和標準形

$$S = \bar{X} \cdot \bar{Y} \cdot C_{in} + \bar{X} \cdot Y \cdot \bar{C}_{in} + X \cdot \bar{Y} \cdot \bar{C}_{in} + X \cdot Y \cdot C_{in}$$

$$C_{out} = \bar{X} \cdot Y \cdot C_{in} + X \cdot \bar{Y} \cdot C_{in} + X \cdot Y \cdot \bar{C}_{in} + X \cdot Y \cdot C_{in}$$

$$S = X \oplus Y \oplus C_{in}$$

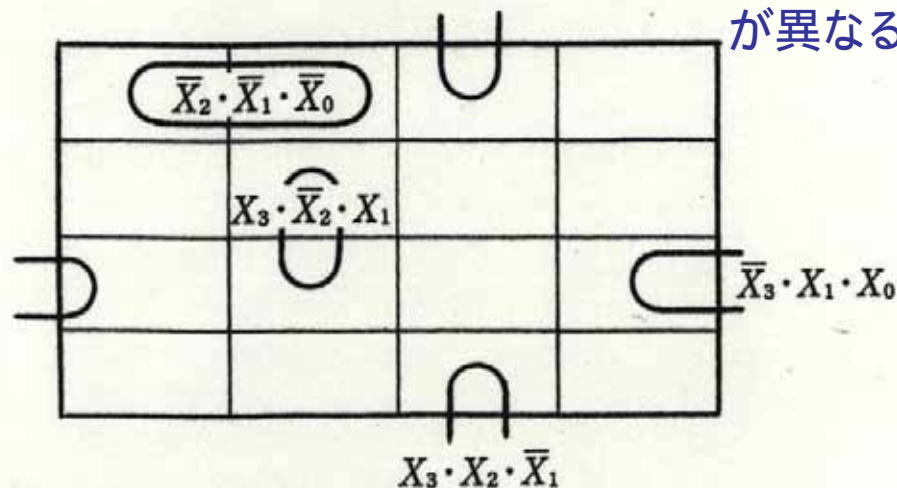
$$C_{out} = (X \oplus Y)C_{in} + XY$$

$\oplus$ : 排他的論理和

# 簡単化

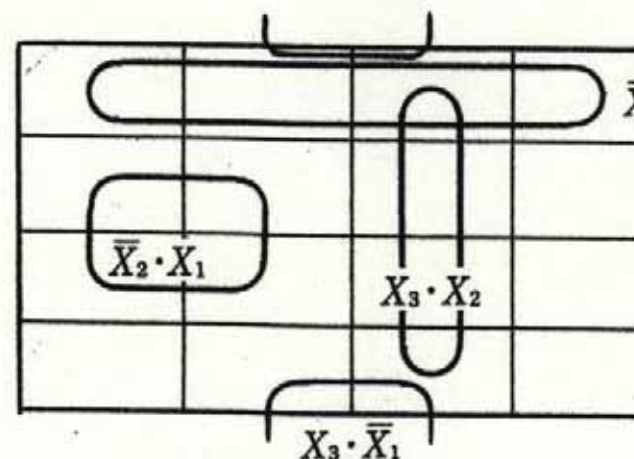
				$X_2$	
$X_0$	$\bar{X}_3 \cdot \bar{X}_2 \cdot \bar{X}_1 \cdot \bar{X}_0$	$X_3 \cdot \bar{X}_2 \cdot \bar{X}_1 \cdot \bar{X}_0$	$X_3 \cdot X_2 \cdot \bar{X}_1 \cdot \bar{X}_0$	$\bar{X}_3 \cdot X_2 \cdot \bar{X}_1 \cdot \bar{X}_0$	$X_1$
	$\bar{X}_3 \cdot \bar{X}_2 \cdot X_1 \cdot \bar{X}_0$	$X_3 \cdot \bar{X}_2 \cdot X_1 \cdot \bar{X}_0$	$X_3 \cdot X_2 \cdot X_1 \cdot \bar{X}_0$	$\bar{X}_3 \cdot X_2 \cdot X_1 \cdot \bar{X}_0$	
	$\bar{X}_3 \cdot \bar{X}_2 \cdot X_1 \cdot X_0$	$X_3 \cdot \bar{X}_2 \cdot X_1 \cdot X_0$	$X_3 \cdot X_2 \cdot X_1 \cdot X_0$	$\bar{X}_3 \cdot X_2 \cdot X_1 \cdot X_0$	
	$\bar{X}_3 \cdot \bar{X}_2 \cdot \bar{X}_1 \cdot X_0$	$X_3 \cdot \bar{X}_2 \cdot \bar{X}_1 \cdot X_0$	$X_3 \cdot X_2 \cdot \bar{X}_1 \cdot X_0$	$\bar{X}_3 \cdot X_2 \cdot \bar{X}_1 \cdot X_0$	
				$X_3$	

(a) 最小項の割付け

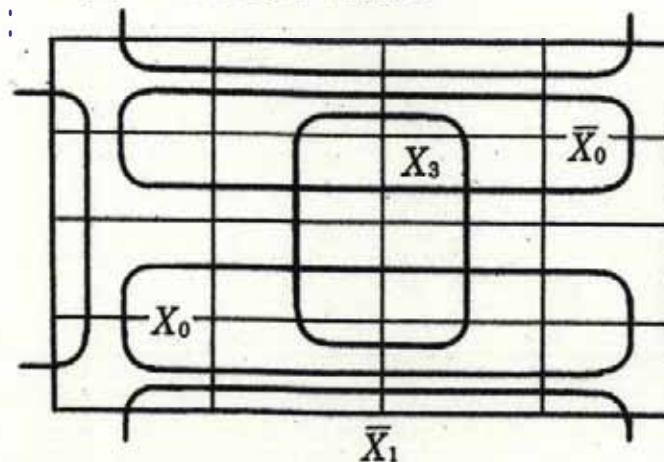


(b) 1変数の縮退

隣接する最小項:  
1箇所のみ変数  
が異なる



(c) 2変数の縮退



(d) 3変数の縮退

図 2.4 ベイチ図表 (4変数の場合)

$$\begin{aligned}
 F &= X_3 X_2 X_1 + \overline{X_3} X_2 X_1 + \overline{X_3} X_1 X_0 + \overline{X_3} \overline{X_2} X_1 \\
 &= X_2 X_1 + \overline{X_3} \overline{X_2} X_1 + \overline{X_3} X_1 X_0
 \end{aligned}$$

または

$$= X_2 X_1 + \overline{X_3} \overline{X_2} X_1 + \overline{X_3} \overline{X_2} X_0$$

ドントケアの場合

$$= X_2 + \overline{X_3} \overline{X_2} X_1 + \overline{X_3} X_1 X_0$$

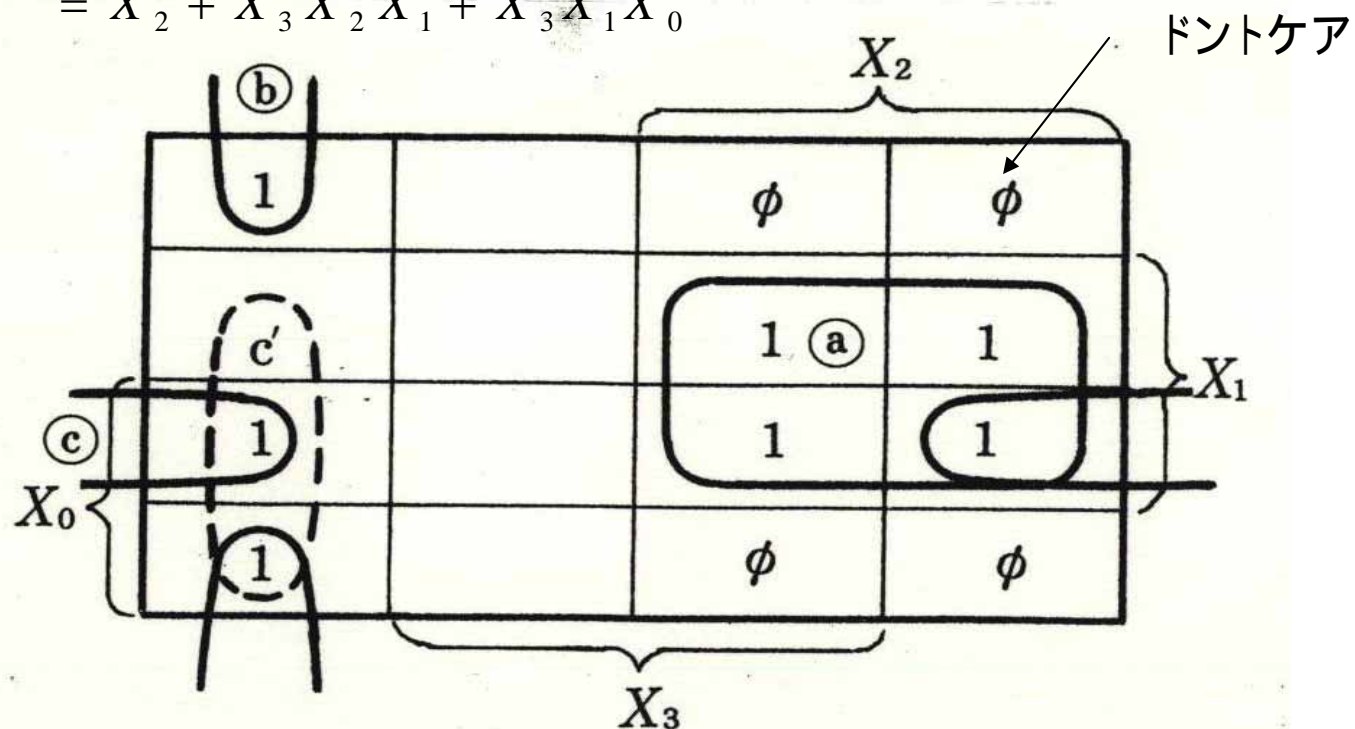
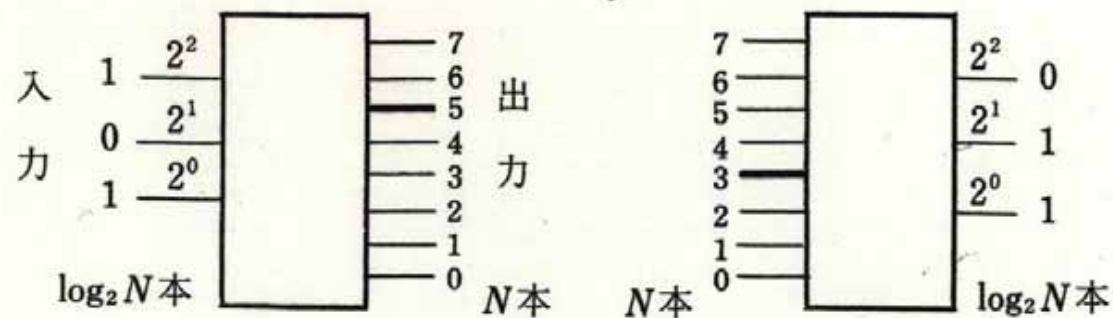


図 2.5 ベイチ図による論理回路の最小化



# 組合わせ回路の例



デコーダ

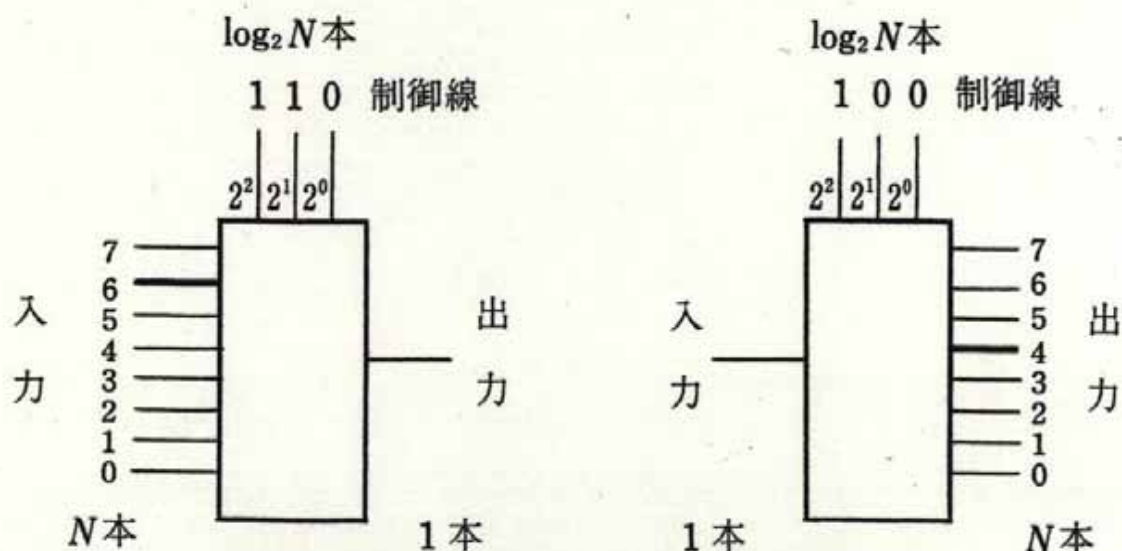
エンコーダ

入 力		出 力			
$X_1$	$X_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$Y_0 = \bar{X}_1 \cdot \bar{X}_0, Y_1 = \bar{X}_1 \cdot X_0$$

$$Y_2 = X_1 \cdot \bar{X}_0, Y_3 = X_1 \cdot X_0$$

(a) デコーダ



マルチプレクサ

デマルチプレクサ

入 力				出 力		
$X_3$	$X_2$	$X_1$	$X_0$	$E$	$Y_1$	$Y_0$
×	×	×	1	0	0	0
×	×	1	0	0	0	1
×	1	0	0	0	1	0
1	0	0	0	0	1	1
0	0	0	0	1	0	0

$$Y_0 = X_1 \cdot \bar{X}_0 + X_3 \cdot \bar{X}_2 \cdot \bar{X}_1 \cdot \bar{X}_0$$

$$Y_1 = X_2 \cdot \bar{X}_1 \cdot \bar{X}_0 + X_3 \cdot \bar{X}_2 \cdot \bar{X}_1 \cdot \bar{X}_0$$

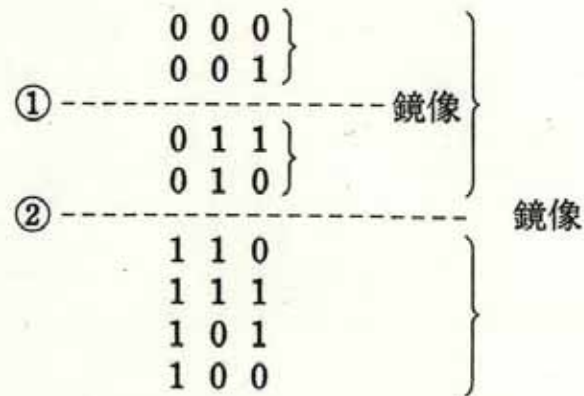
$$E = \bar{X}_3 \cdot \bar{X}_2 \cdot \bar{X}_1 \cdot \bar{X}_0$$

(b) プライオリティエンコーダ

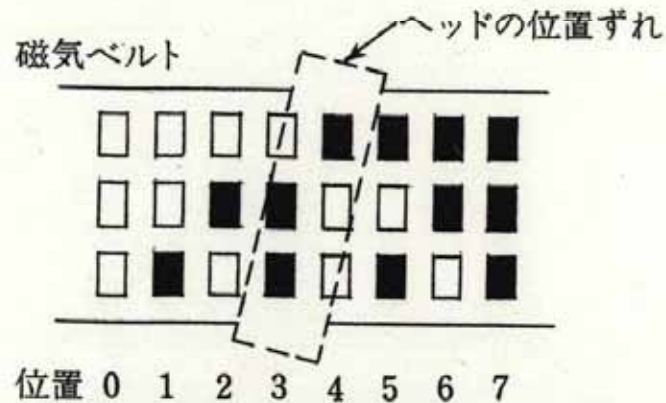
図 2.6 デコーダ/エンコーダとマルチプレクサ/デマルチプレクサ

図 2.7 デコーダとエンコーダ

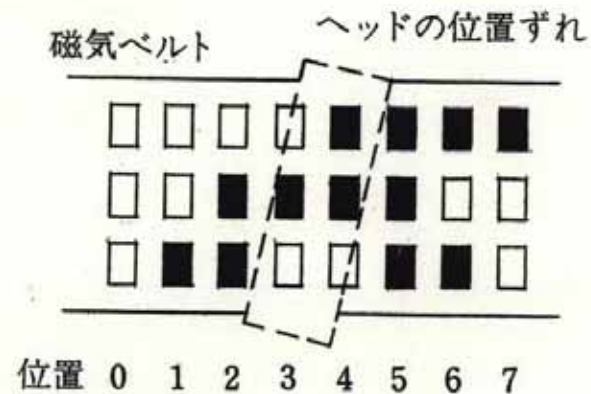
# 反射2進符号: 隣合う符号の ハミング距離1



(a) 反射2進符号の作り方



(b) 通常の2進符号による  
磁気ベルトの位置検出

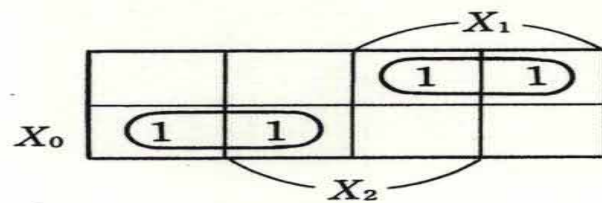


(c) 反射2進符号による磁  
気ベルトの位置検出

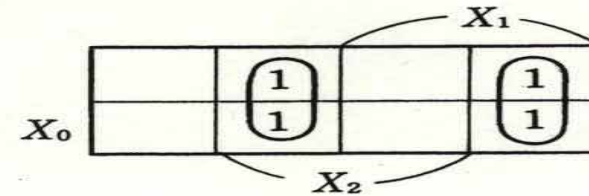
図 2.8 反射2進符号

入 力			出 力		
$X_2$	$X_1$	$X_0$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

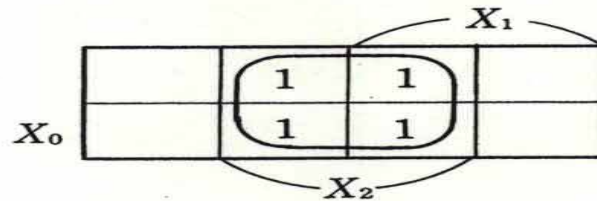
(a) 真理値表



$$Y_0 = X_1 \cdot \bar{X}_0 + \bar{X}_1 \cdot X_0$$



$$Y_1 = X_2 \cdot \bar{X}_1 + \bar{X}_2 \cdot X_1$$



$$Y_2 = X_2$$

(b) ベイチ図と論理関数

図 2.9 反射 2 進符号への変換



## 2.3 順序回路

状態遷移図

状態割付

論理設計

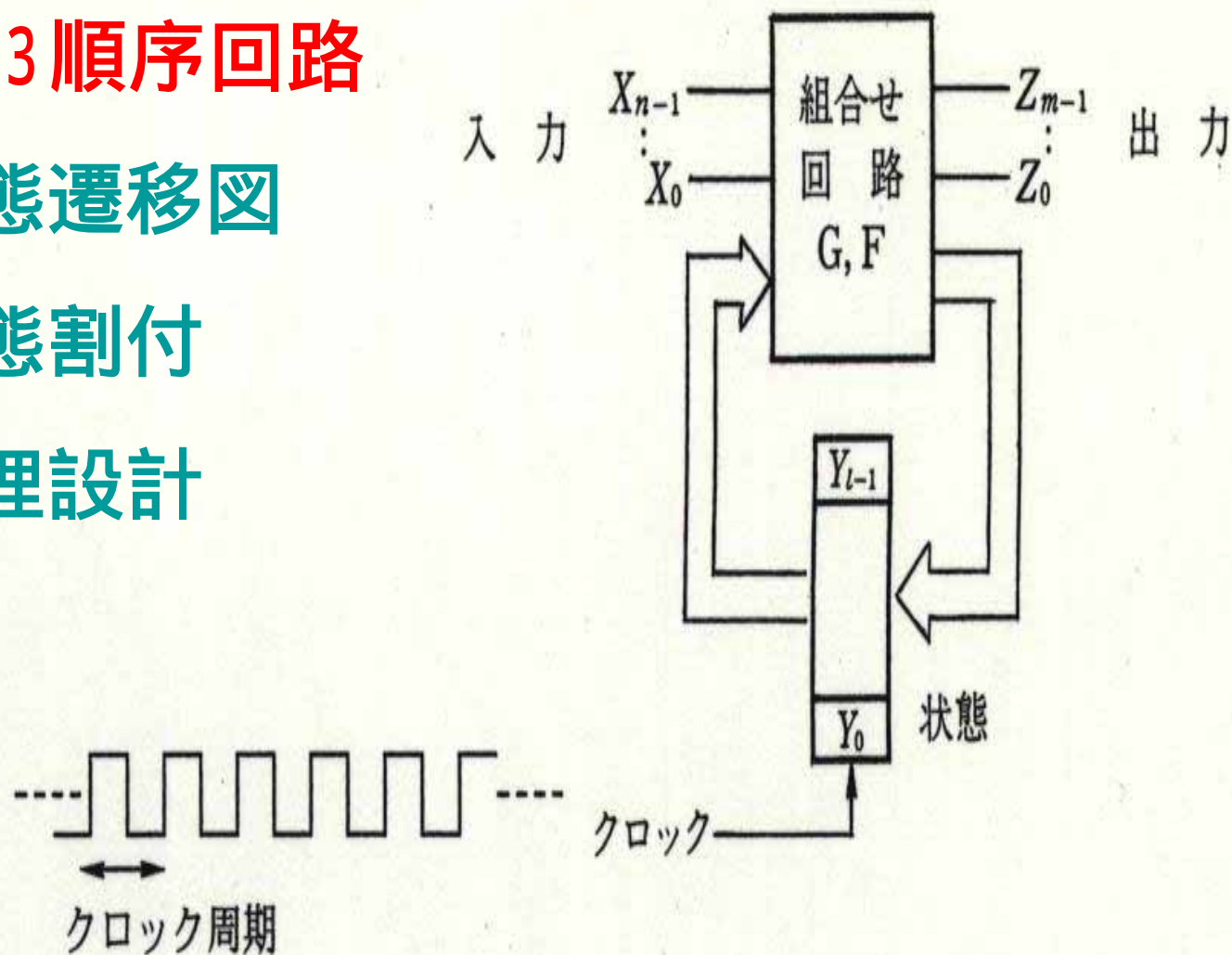


図 2.10 順序回路

# フリップフロップ

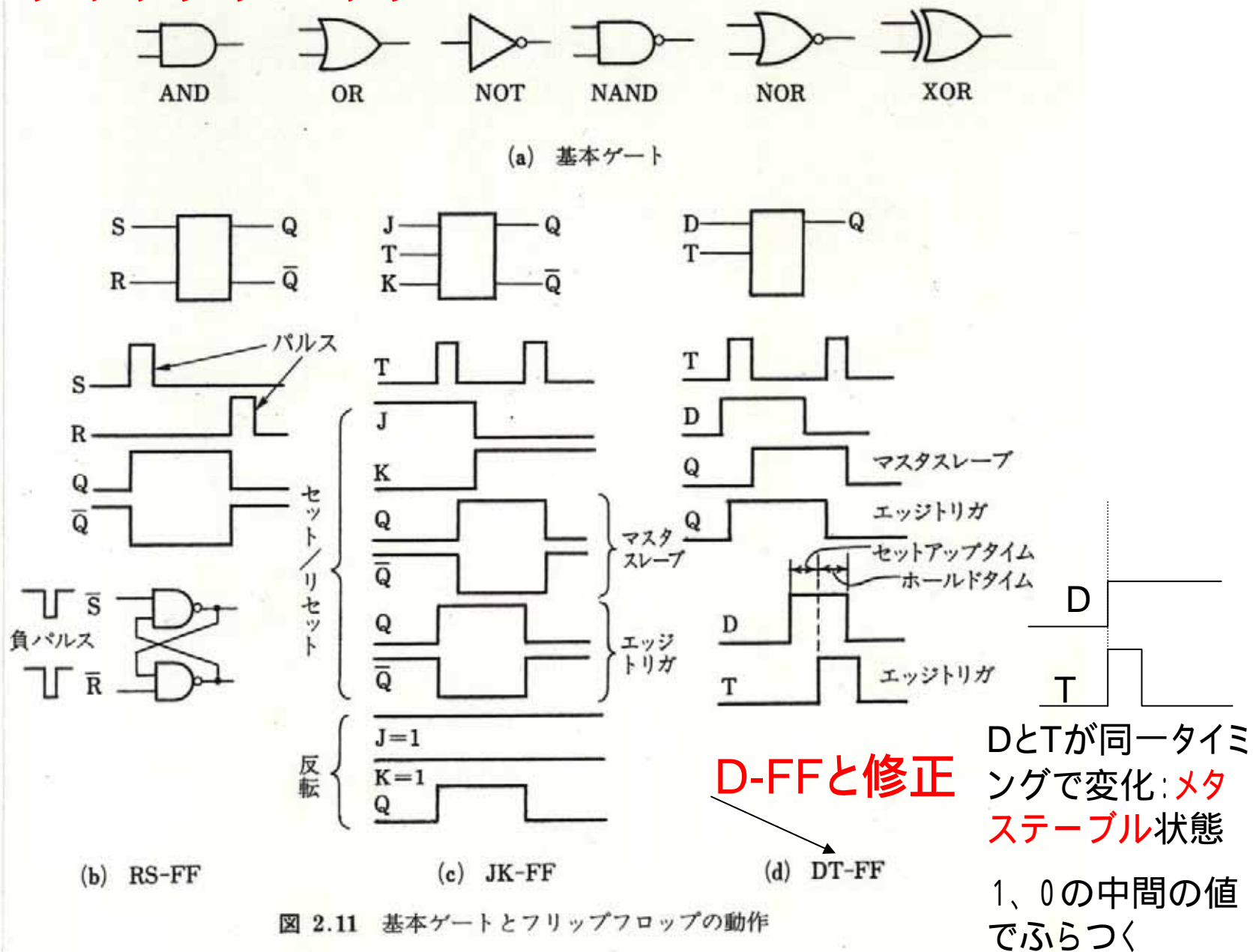
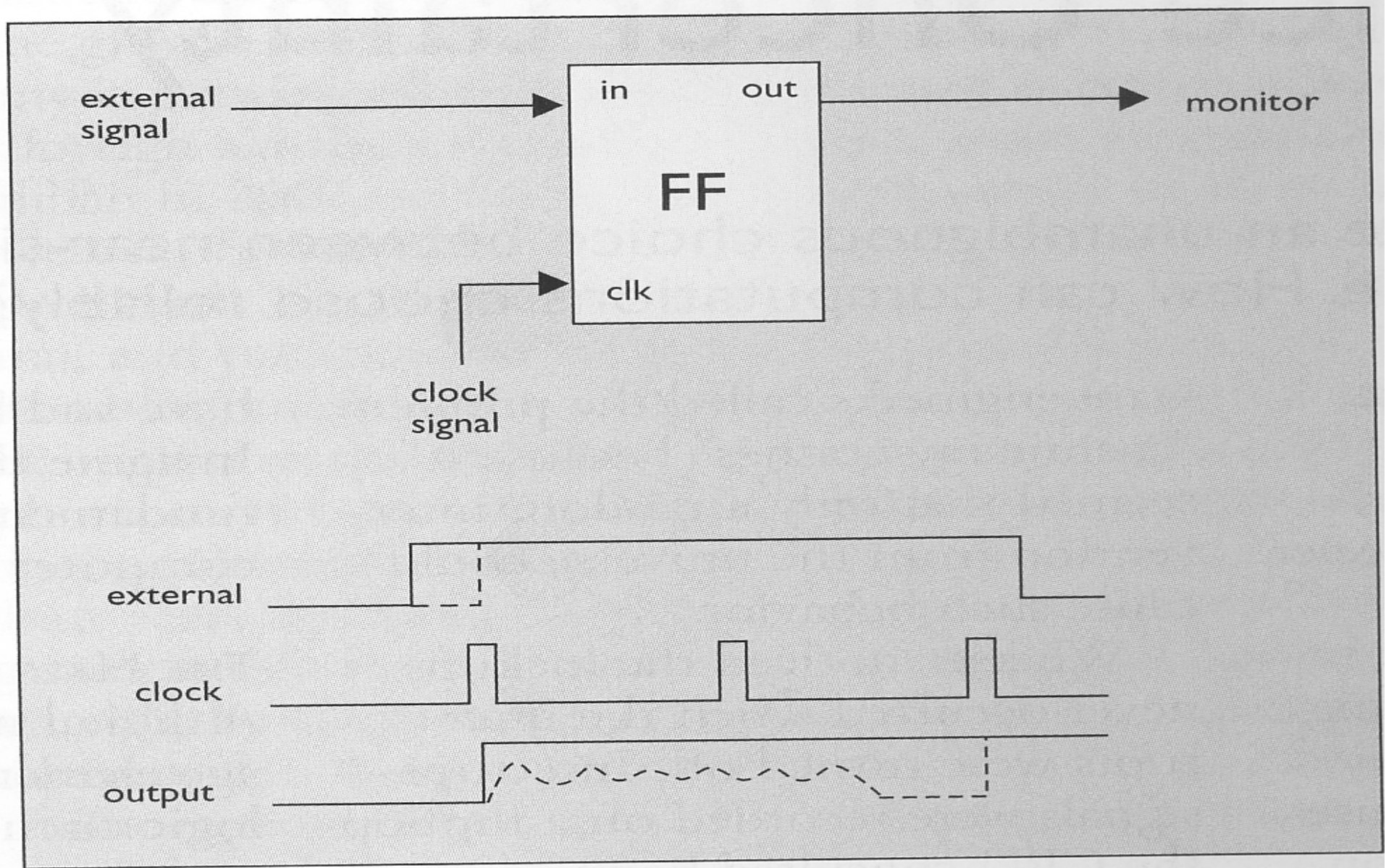
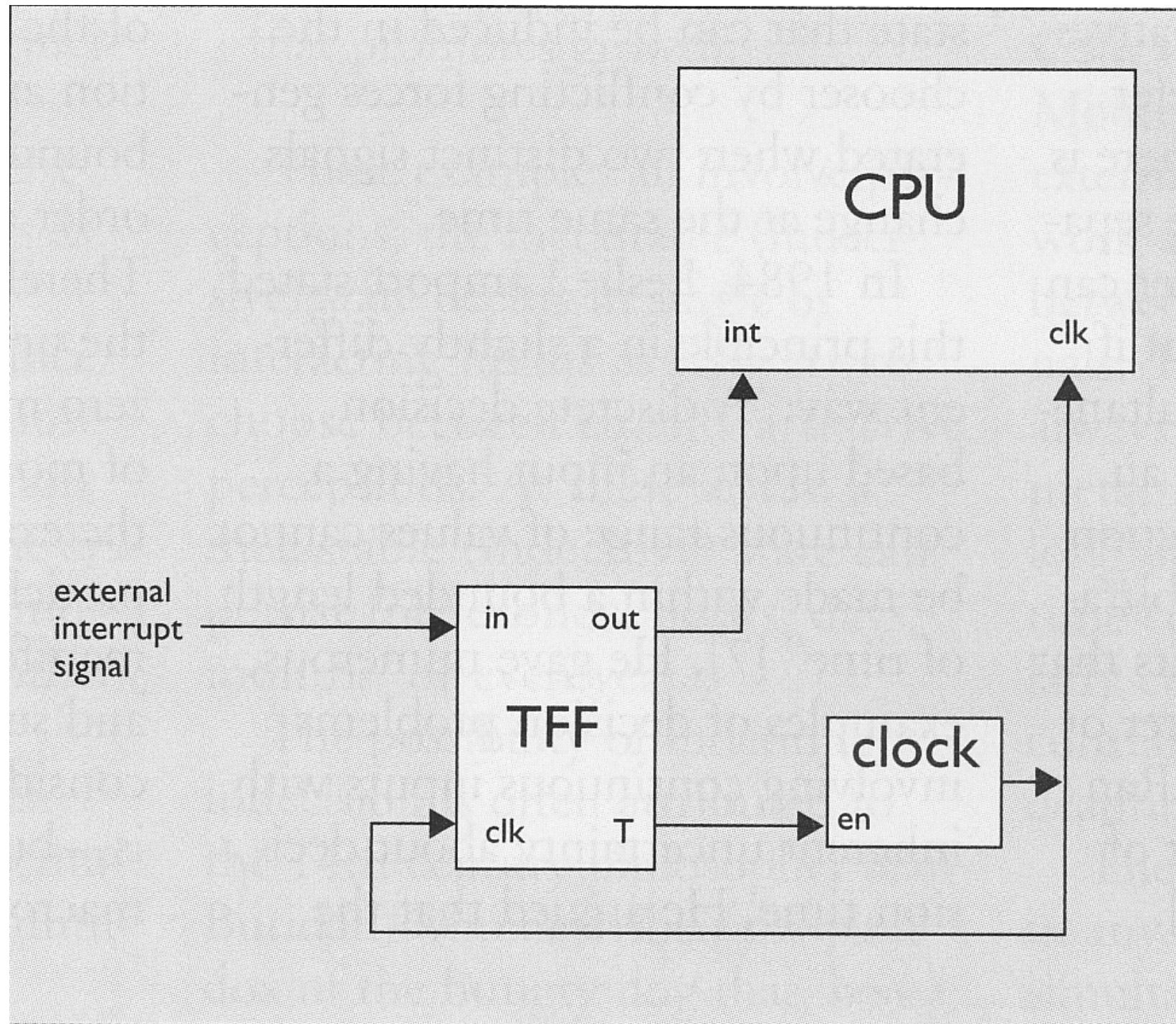


図 2.11 基本ゲートとフリップフロップの動作



P.J.Denning: The Choice Uncertainty Principle, CACM, Vol.50,No11,pp.9-14,2007



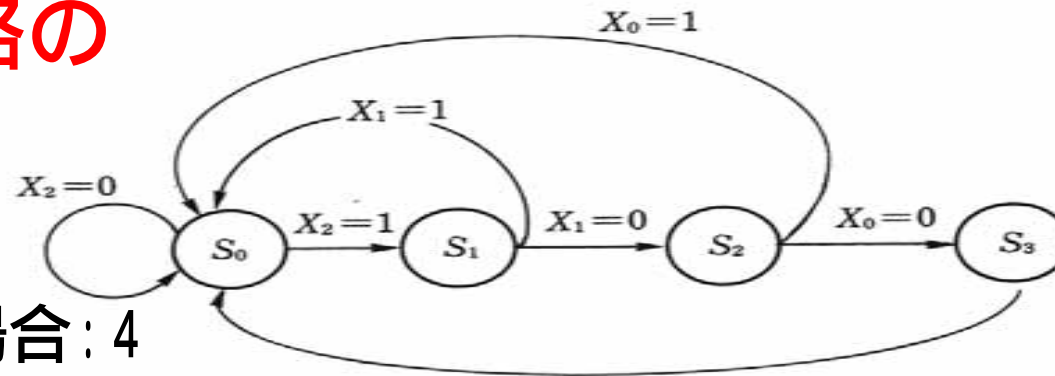


Xilinx.com reports that its modern flipflops have essentially zero chance of observing a metastable state when clock frequencies are 200MHz or less [1]. At these frequencies, the time between clock pulses (five nanoseconds) is longer than all metastable events. But in experiments with interrupt signals arriving 50 million times a sec-

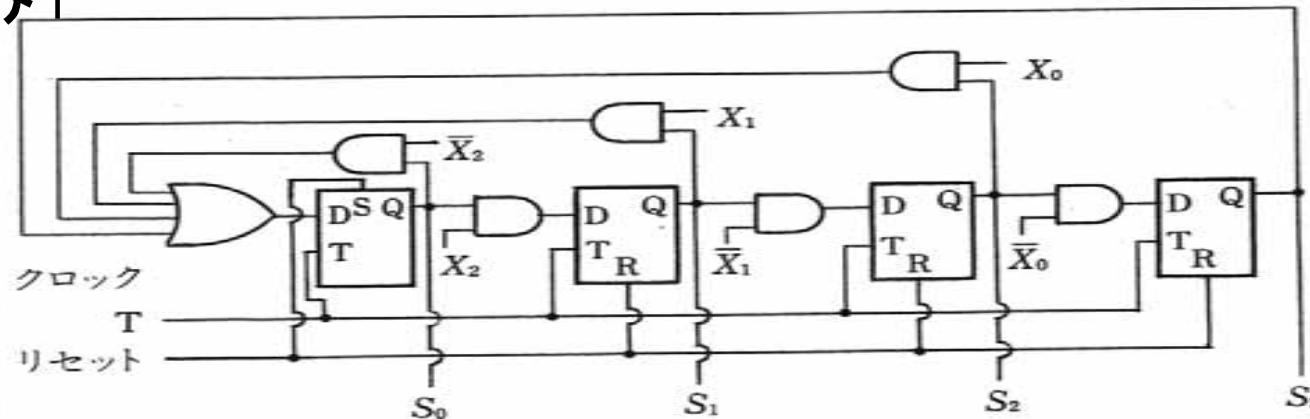
ond, a metastable state occurs about once a minute at clock frequency 300MHz and about once every two milliseconds at clock frequency 400MHz. In a computer system generating 500 interrupts per second, approximately 1/100,000 of the experimental rate, these extrapolate to one interrupt-caused metastable state about every two weeks at 300MHz and about every three minutes at 400MHz.

# 順序回路の 設計例

無符号化の場合：4  
ビットのうち、どれか  
1つのみ1



(a) 状態遷移図



(b) 無符号化による実現

図 2.12 状態遷移図と無符号化による実現

2ビットで符号化

入 力						出 力		
$X_2$	$X_1$	$X_0$	$Q_1$	$Q_0$ (現状態)		$D_1$	$D_0$ (次状態)	
1	×	×	0	0	$S_0$	0	1	$S_1$
0	×	×	0	0	$S_0$	0	0	$S_0$
×	1	×	0	1	$S_1$	0	0	$S_0$
×	0	×	0	1	$S_1$	1	0	$S_2$
×	×	1	1	0	$S_2$	0	0	$S_0$
×	×	0	1	0	$S_2$	1	1	$S_3$
×	×	×	1	1	$S_3$	0	0	$S_0$

次状態を与えるD-FFの入力

ひげ状(グリッチ)の状態S3

$$D_1 = \bar{X}_1 \cdot \bar{Q}_1 \cdot Q_0 + \bar{X}_0 \cdot Q_1 \cdot \bar{Q}_0$$
$$D_0 = X_2 \cdot \bar{Q}_1 \cdot \bar{Q}_0 + \bar{X}_0 \cdot Q_1 \cdot \bar{Q}_0$$

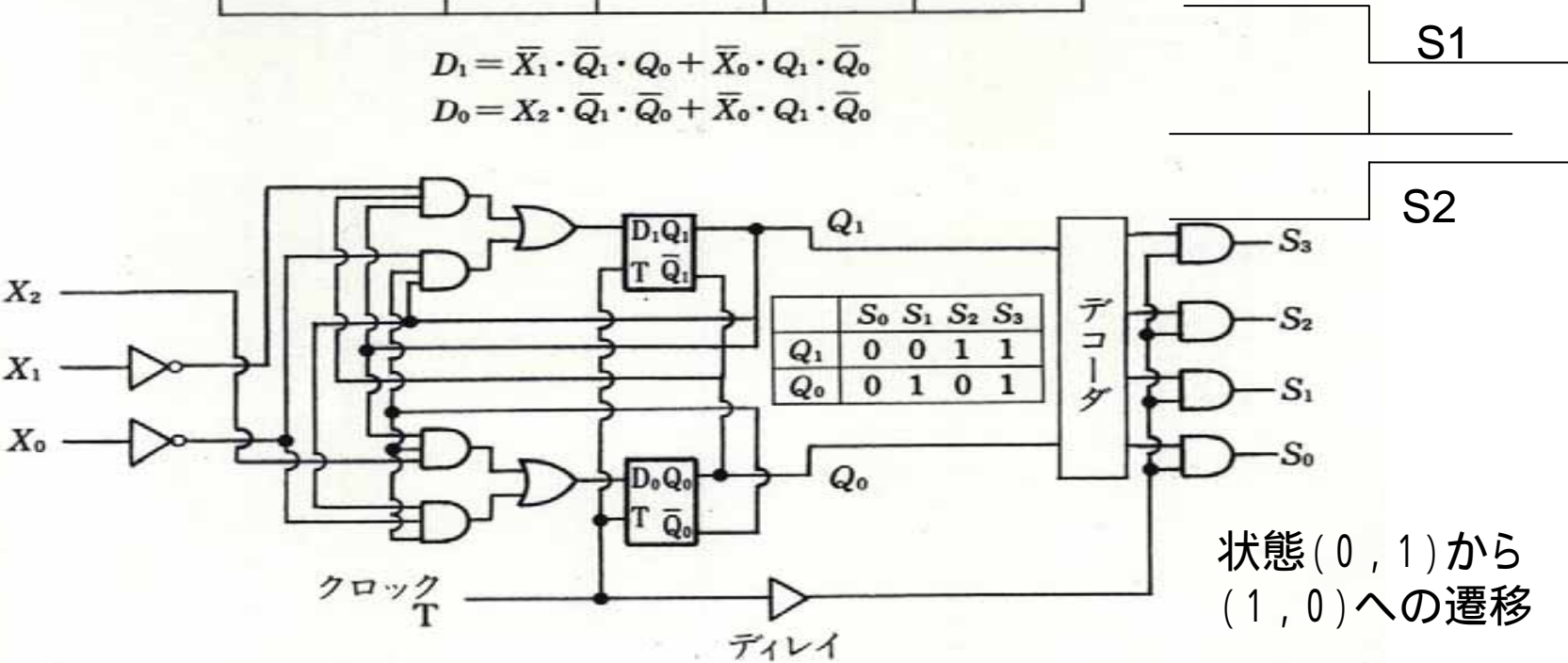


図 2.13 状態の符号化による実現

状態(0,1)から  
(1,0)への遷移  
(0,1) (1,1)  
(1,0)



# 簡単な順序回路の例

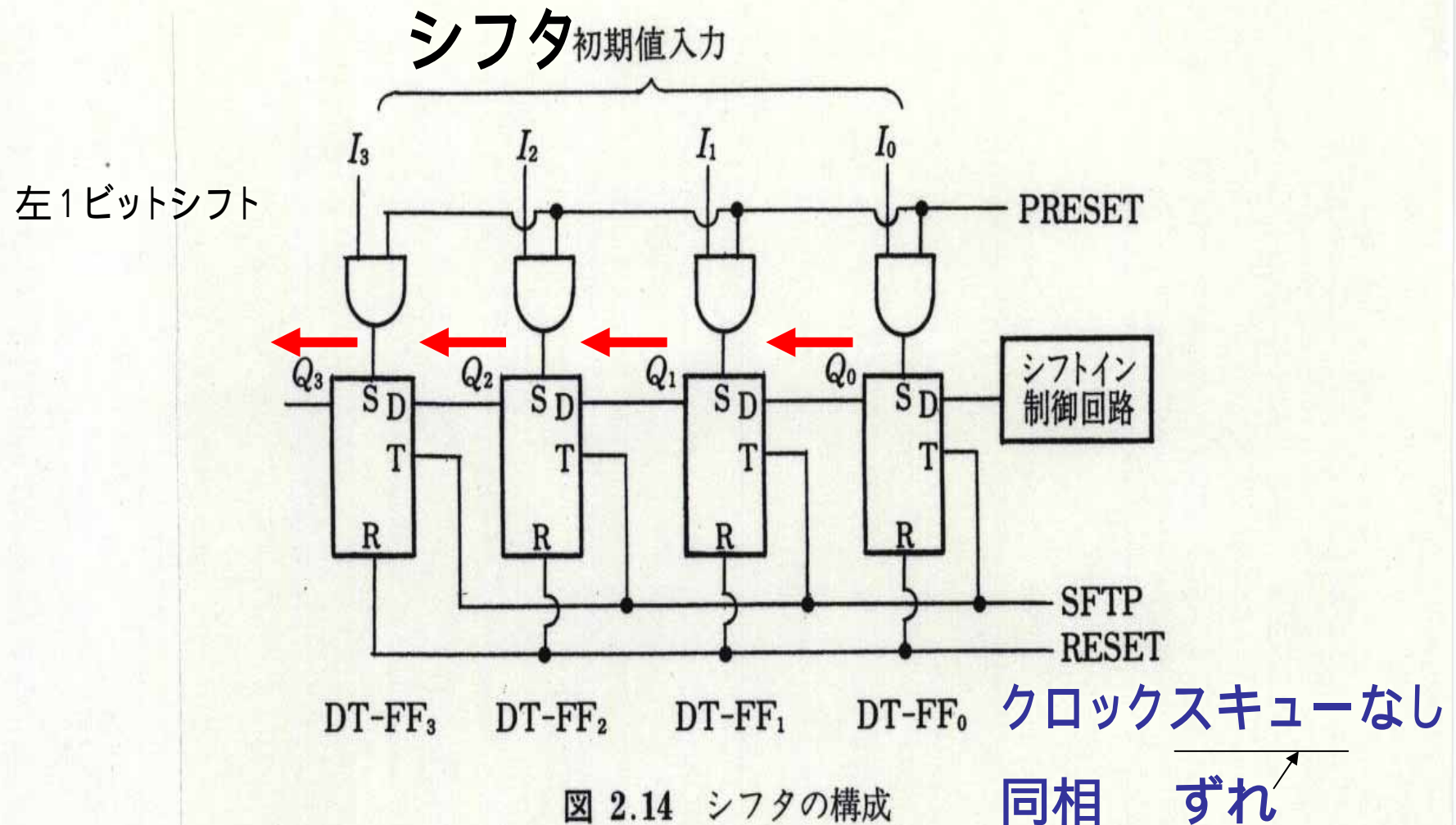
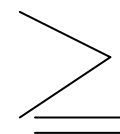


図 2.14 シフトの構成

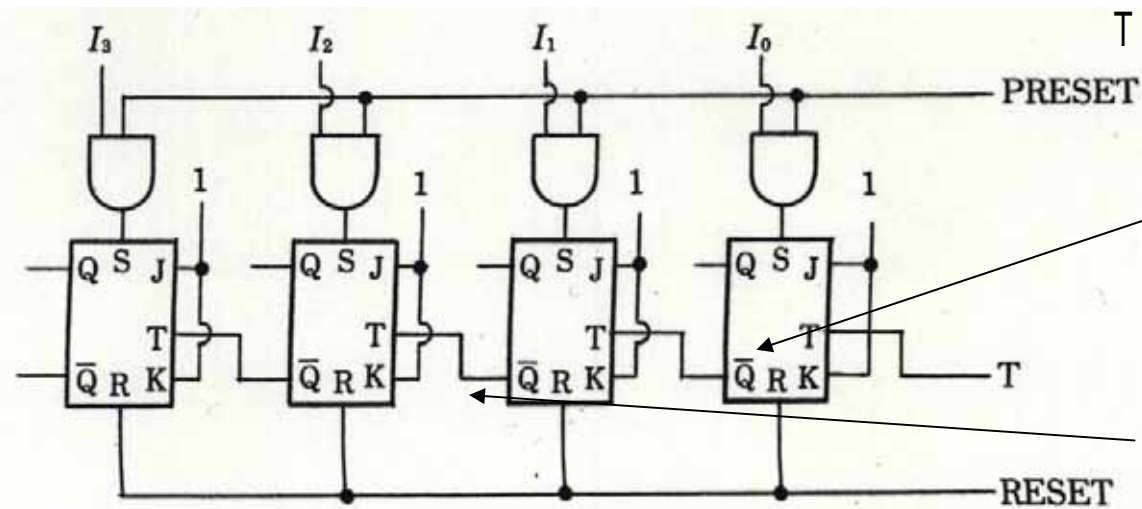
取り込んだ入力値が出力に  
現れるまでの時間



ホールドタイム



# カウンタ



(a) 非同期式カウンタ

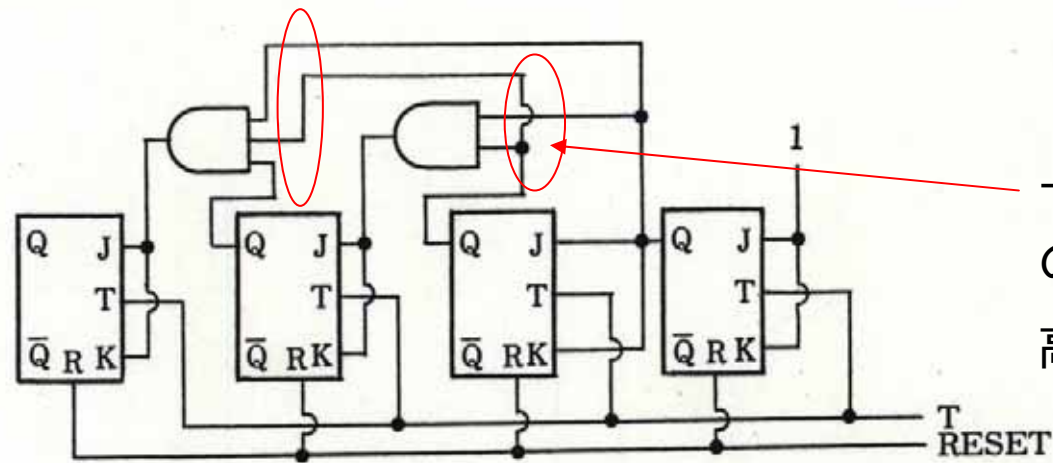
T: 立上がりで反転

2ビット目:

Q0が0,1,0  
となったとき  
反転

3ビット目:  
Q1が0,1,0  
となったとき  
反転

時間かかる



(b) 同期式カウンタ

下位ビットがオール1  
のとき反転

高速

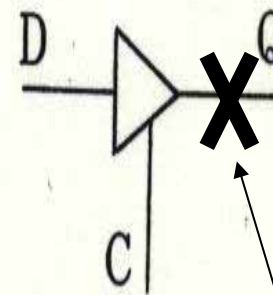
図 2.15 カウンタの構成

## 2.4 バス制御回路

### 3 状態素子 (Tri-State)

D	C	Q
1	0	Z
0	0	Z
1	1	1
0	1	0

真理値表



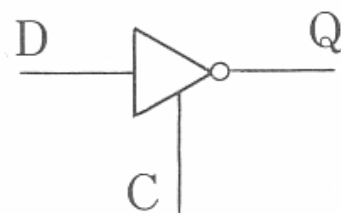
記号

Z状態: 出力が切れている状態

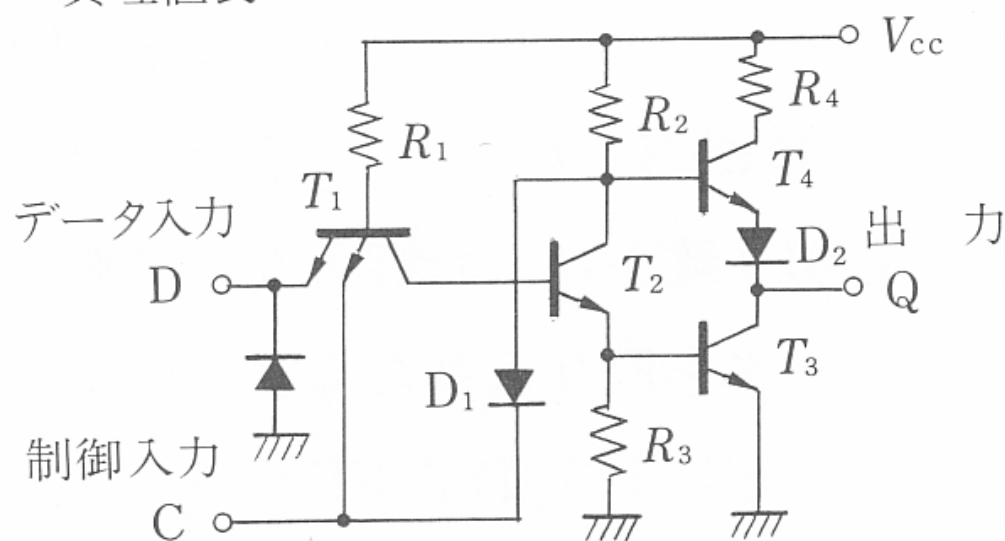
図 2.16 3 状態素子

D	C	Q
1	0	Z
0	0	Z
1	1	0
0	1	1

真理値表



記号

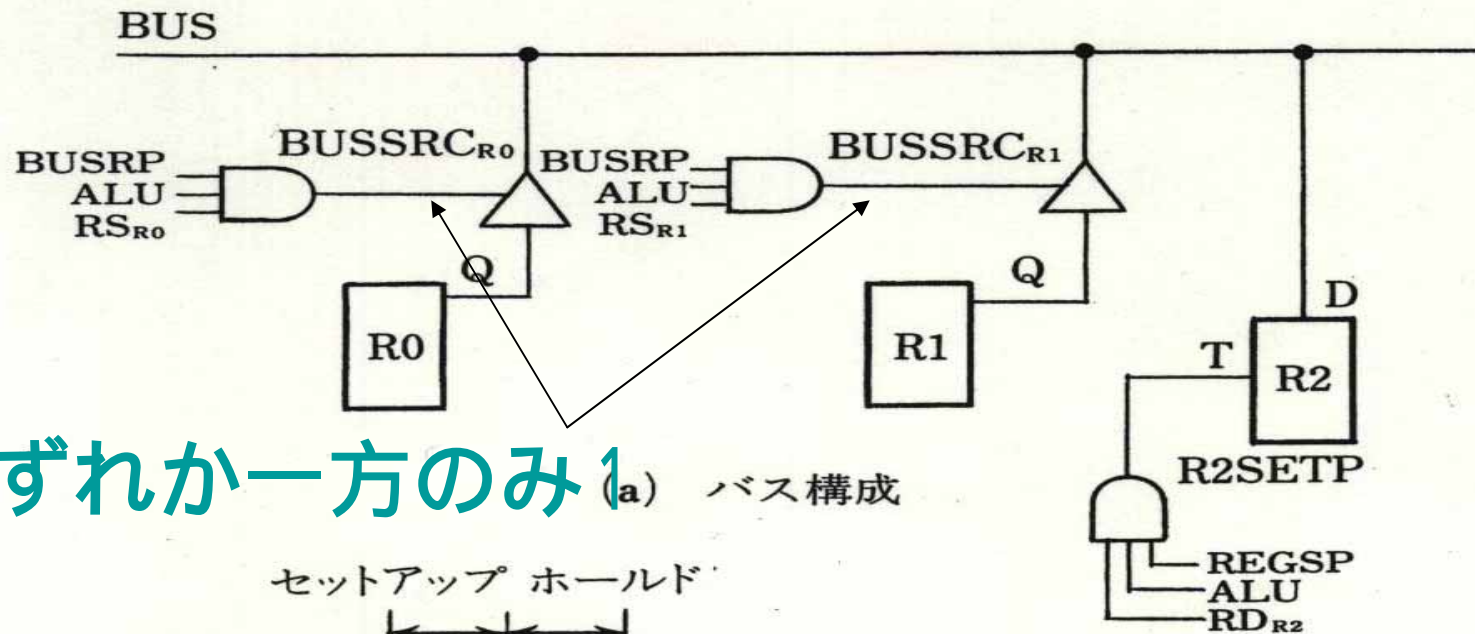


実際の回路

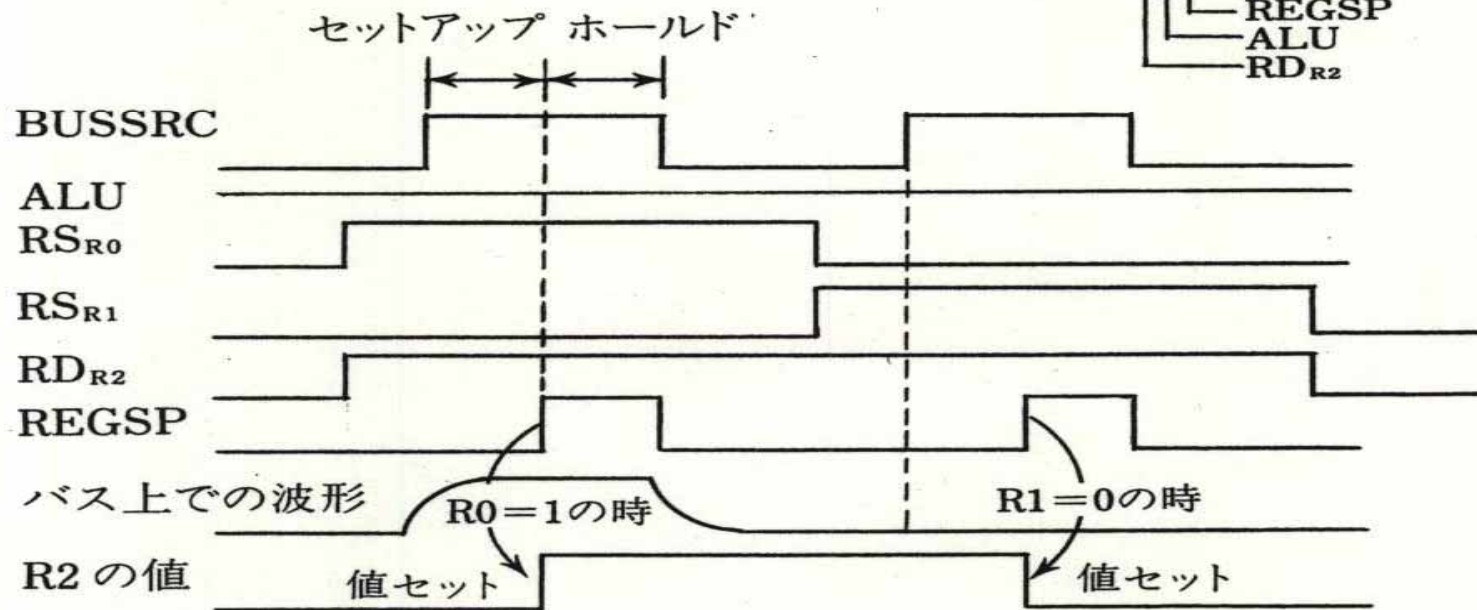
D	C	Q
H	L	Z
L	L	Z
H	H	L
L	H	H

実際の回路の入出力関係

図 2.19 3 状態素子



## いずれか一方のみ<sup>1</sup> (a) バス構成



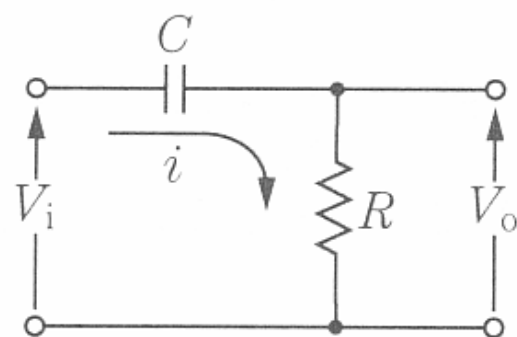
(b) 制御タイミング

図 2.17 バス制御方式

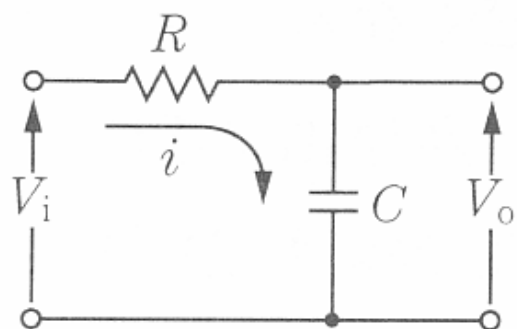
# バスの特徴

- 1本の線路で遠い位置に散在している機能装置を結合
- 配線面積を小さくできる
- ブロードキャスト(一斉通信)が容易
- 容量性の負荷
  - 信号遅延の増大
  - 大電力ドライバトランジスタ必要

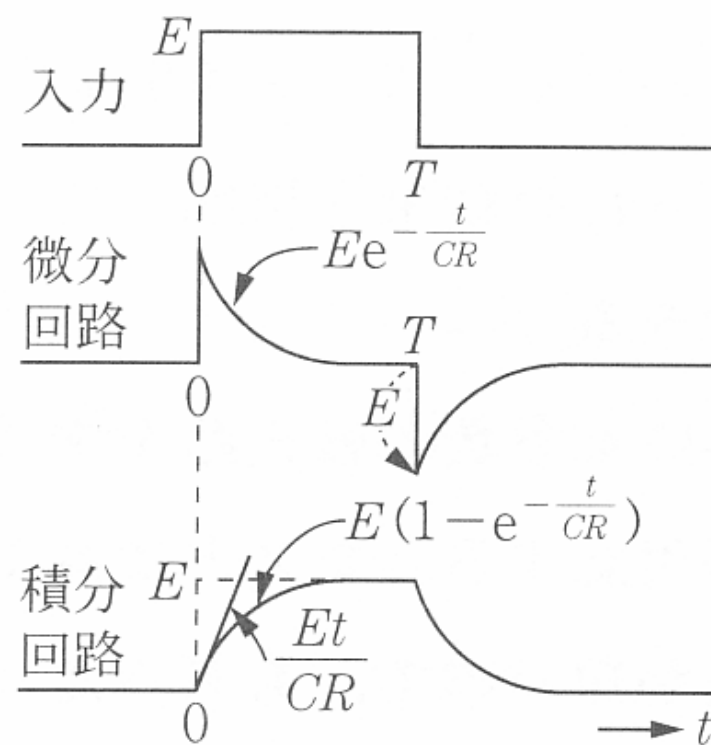
IC内での結合: マルチプレクサで1対1結合



(a) 微分回路



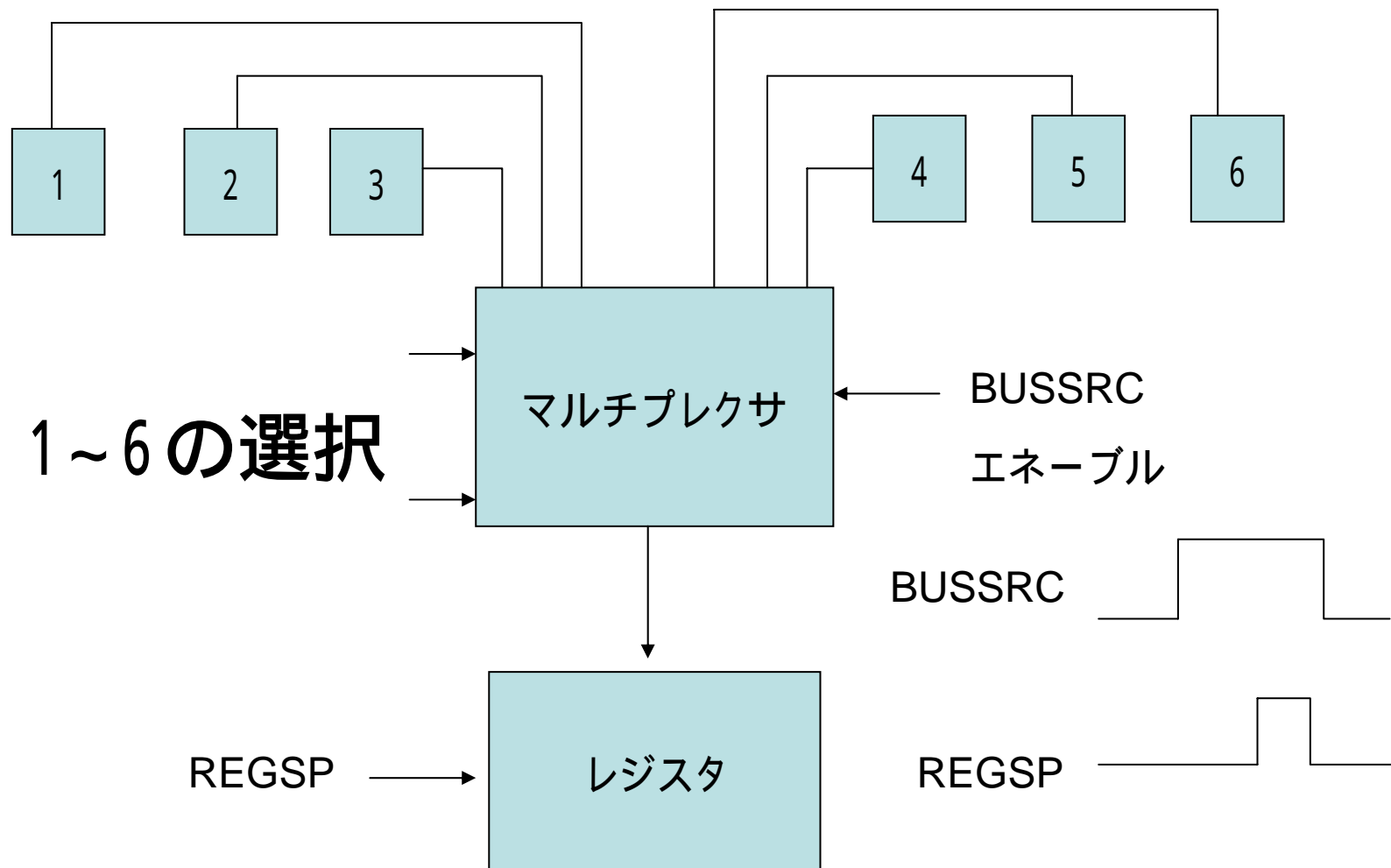
(b) 積分回路



(c) パルス応答

図 2.25 微分積分回路

ポイント-ポイント結合



## 8 ビットコンピュータの製作

### 2.5 機械命令セットの設計



## 2.3 機械命令セット

命令形式：

ロード/ストアアーキテクチャ

命令長：

1バイト命令と2バイト命令から構成

データタイプとデータ長：

1バイト整数

1バイト論理値

アドレッシングモード：

レジスタ  $R_i$ , ディスプレースメント  $M(R_i+D)$  をサポート

汎用レジスタ：

$R_0 \sim R_3$ ：汎用レジスタ

$R_0$  : BAL 命令のリターンアドレスの格納にも使用

## 特殊レジスタ

プログラムカウンタ：PC

条件コードレジスタ：CC

V：Overflow（オーバフロー）

C：Carry/Borrow（キャリー/ボロー）

Z：Zero（ゼロ）

N：Negative（負）

3	2	1	0
V	C	Z	N

## 命令レジスタ：IR

主記憶から読み出された機械命令を格納

デコードは主フィールド OP1 (I7-I4)，副フィールド OP2 (I3, I2)，OP3 (I1, I0) でなされる。下位バイトは変位である。

OP 1				OP 2		OP 3		変位 8 ビット							
I 7	I 6	I 5	I 4	I 3	I 2	I 1	I 0								

### 2.3.1 算術論理演算命令

第1章で命令の記法を定義した。本コンピュータは1バイト整数演算のみサポートするので、データタイプとデータ長を簡略化した記法を使用する。算術演算はレジスタレジスタ命令形式

$$Rd \leftarrow Rd \text{ op } Rs$$

でなされ、以下のものが用意されている。また、負の数の表現は2の補数を用いる。 $B_{MSB}$  は最上位桁へのボロー、 $B_{MSB-1}$  は最上位桁から次の桁へのボロー、 $C_{MSB}$  は最上位桁からのキャリー、 $C_{MSB-1}$  は最上位桁へのキャリーを表す。図2.24にオーバーフローの条件を示す(第3章を参照せよ)。



図 2.24 加算/減算におけるオーバーフロー

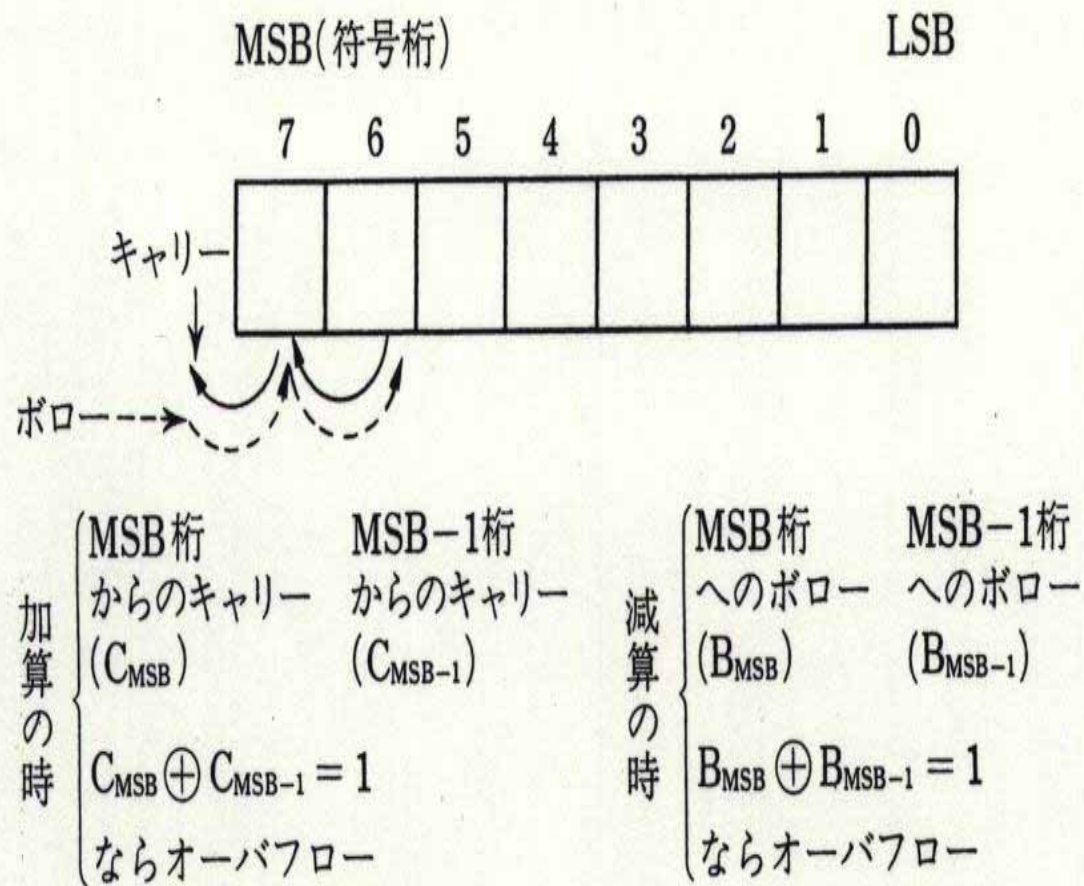


図 2.18 加算/減算におけるオーバーフロー

a. 算術命令

	4	2	2
SUB	0010	Rd	Rs

$V \leftarrow 1$   $B_{\text{MSB}} \oplus B_{\text{MSB}-1} = 1$  のとき

$C \leftarrow 1$   $B_{\text{MSB}} = 1$  のとき

$Z \leftarrow 1$  すべて 0 のとき

$N \leftarrow 1$   $\text{MSB} = 1$  のとき

	4	2	2
ADD	0011	Rd	Rs

$V \leftarrow 1$   $C_{\text{MSB}} \oplus C_{\text{MSB}-1} = 1$  のとき



$C \leftarrow 1$   $C_{MSB}=1$  のとき

$Z \leftarrow 1$  すべて 0 のとき

$N \leftarrow 1$  MSB=1 のとき

b. インクリメント/デクリメント

	4	2	2
DEC	0001	Rd	10

$Rd \leftarrow Rd - 1$

CC 設定は SUB と同様

	4	2	2
INC	0001	Rd	11

$Rd \leftarrow Rd + 1$

CC 設定は ADD と同様

c. 論理演算 論理演算には以下のものがあり，条件コードの設定は次のように行われる．

$V, C \leftarrow 0$

$Z \leftarrow 1$  すべて0のとき

$N \leftarrow 1$  すべて1のとき

	4	2	2
XOR	0100	Rd	Rs

	4	2	2
OR	0101	Rd	Rs

	4	2	2
AND	0110	Rd	Rs

### 2.3.2 シフト演算

シフトには論理シフト (SLL, SLLC, SRL, SRLC) と算術シフト (SLA, SLAC, SRA) があり，命令形式はレジスタ-レジスタである．

a. 論理シフト シフトアウトとシフトインを図 2.25 に示す．

	4	2	2
SLL	1000	Rd	00

論理左シフト

$LSB \leftarrow 0$

$V \leftarrow 0$

$C \leftarrow MSB$

$Z \leftarrow 1$  すべて 0 のとき

$N \leftarrow 1$  すべて 1 のとき

	4	2	2
SLLC	1000	Rd	01

論理左シフト (C 詰め)

$LSB \leftarrow C$

$V \leftarrow 0$



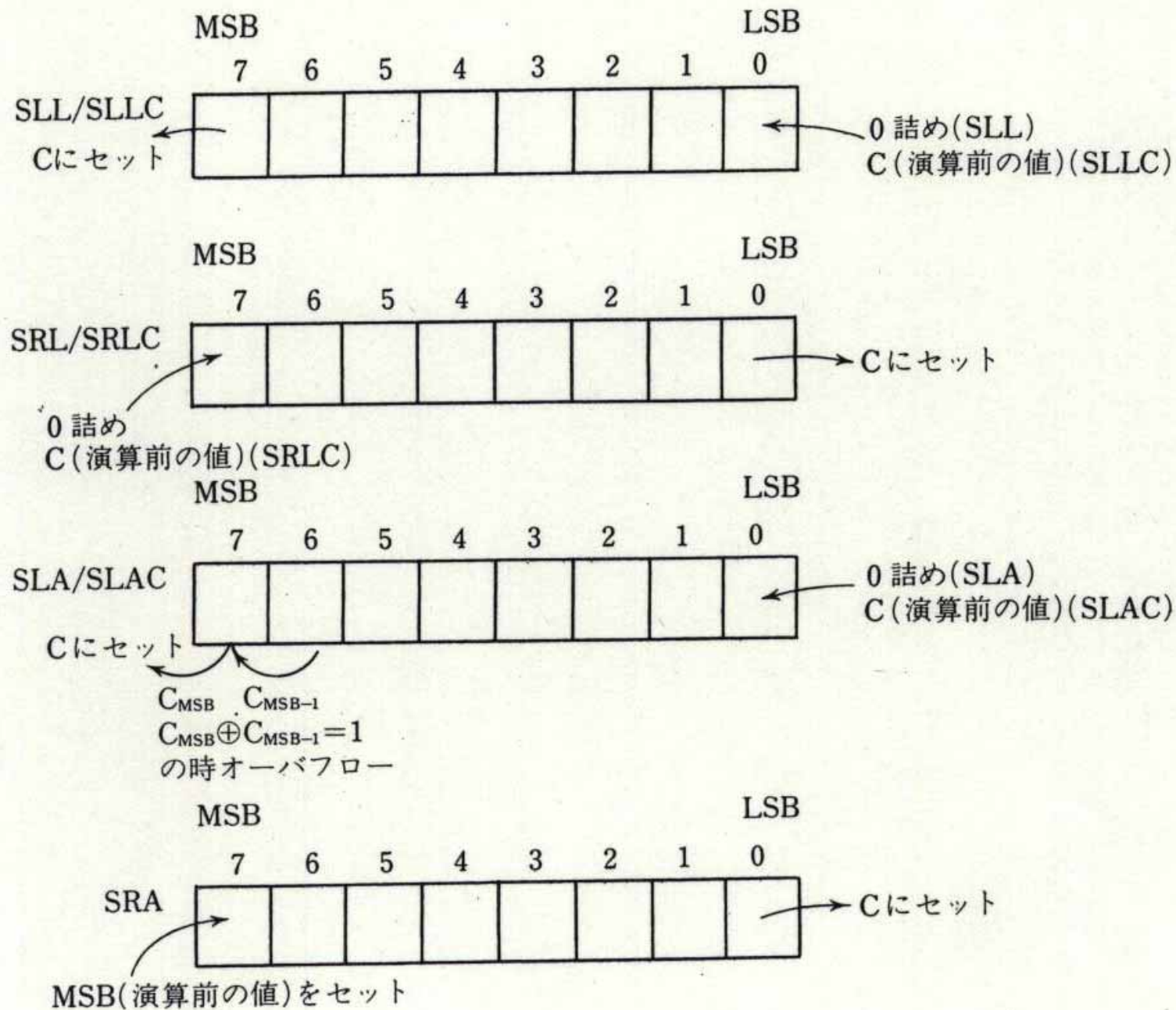


図 2.25 シフト演算

$C \leftarrow \text{MSB}$

$Z \leftarrow 1$  すべて0のとき

$N \leftarrow 1$  すべて1のとき

	4	2	2
SRL	1000	Rd	10

論理右シフト

$\text{MSB} \leftarrow 0$

$V \leftarrow 0$

$C \leftarrow \text{LSB}$

$Z \leftarrow 1$  すべて0のとき

$N \leftarrow 1$  すべて1のとき

	4	2	2
SRLC	1000	Rd	11

論理右シフト (C 詰め)

$\text{MSB} \leftarrow C$

$V \leftarrow 0$

$C \leftarrow \text{LSB}$

$Z \leftarrow 1$  すべて0のとき

$N \leftarrow 1$  すべて1のとき

b. 算術シフト

	4	2	2
SLA	1001	Rd	00

算術左シフト

$LSB \leftarrow 0$

$V \leftarrow 1 \quad C_{MSB} \oplus C_{MSB-1} = 1$  のとき

$C \leftarrow MSB$

$Z \leftarrow 1$  すべて 0 のとき

$N \leftarrow 1$   $MSB = 1$  のとき

	4	2	2
SLAC	1001	Rd	01

算術左シフト (C 詰め)

$LSB \leftarrow C$

$V \leftarrow 1 \quad C_{MSB} \oplus C_{MSB-1} = 1$  のとき

$C \leftarrow MSB$

$Z \leftarrow 1$  すべて 0 のとき

$N \leftarrow 1$   $MSB = 1$  のとき

	4	2	2
SRA	1001	Rd	10

算術右シフト

$V \leftarrow 0$

$C \leftarrow LSB, MSB \leftarrow MSB$

$Z \leftarrow 1$  すべて 0 のとき

$N \leftarrow 1$   $MSB = 1$  のとき

### 2.3.3 条件分岐 (BRC)

条件分岐命令 (BRC) は次の命令形式をとっている。BM は分岐モードを表す。

BM=00 のとき絶対アドレッシング A

IF V, C, Z, N=1 Then  $PC \leftarrow D$  Else  $PC \leftarrow PC+2$

BM=10 のとき自己相対アドレッシング S

IF V, C, Z, N=1 Then  $PC \leftarrow PC+D$  Else  $PC \leftarrow PC+2$

V, C, Z, N: 変更せず

BM の区別は個々の条件分岐命令の記号の後に A (絶対), S (自己) をつけて示す。

たとえば, BRCVA, BRCVS など。

	4	2	2	8
BRCV	1010	00	BM	D

	4	2	2	8
BRCC	1010	01	BM	D

	4	2	2	8
BRCZ	1010	10	BM	D

	4	2	2	8
BRCN	1010	11	BM	D



### 2.3.4 無条件分岐 (BRUC)

無条件分岐には次の2種類がある。BAL 命令はサブルーティンコール命令として使用する。その際リターン（復帰）アドレスは R0 に格納される。BRU 命令では無条件にオペランドアドレスに分岐する。BM は絶対、自己相対アドレッシングモードを示し、無条件分岐の記号のあとに A, S をつけて示す。BRUA, BRUS など。

	4	2	2	8
BRU	1011	00	BM	D

$PC \leftarrow D$  : BM=00 のとき

$PC \leftarrow PC + D$  : BM=10 のとき

	4	2	2	8
BAL	1011	01	BM	D

$R0 \leftarrow PC + 2$ ,  $PC \leftarrow D$  : BM=00 のとき

$R0 \leftarrow PC + 2$ ,  $PC \leftarrow PC + D$  : BM=10 のとき

RTN	1011	10	-	-
-----	------	----	---	---

PC R0

### 2.3.5 ロード/ストア

ロード/ストア命令には以下の3種がある。Xはディスプレイースメントアドレッシング、Aは絶対アドレッシング、Iはリテラルを表す。いずれもV, C, Z, Nは変更しない。

	4	2	2	8
LOADX	1100	Rd	Rb	D

$Rd \leftarrow M(Rb + D)$

	4	2	2	8
LOADI	1101	Rd	00	D

$Rd \leftarrow \#D$

	4	2	2	8
LOADA	1101	Rd	01	D

$$R_d \leftarrow M(D)$$

	4	2	2	8
STOREX	1110	Rs	Rb	D

$$M(R_b + D) \leftarrow R_s$$

	4	2	2	8
STOREA	1111	Rs	/	D

$$M(D) \leftarrow R_s$$

### 2.3.6 その他

停止命令として HLT (Halt) を用意する.

	4	2	2
HLT	0000	/	/



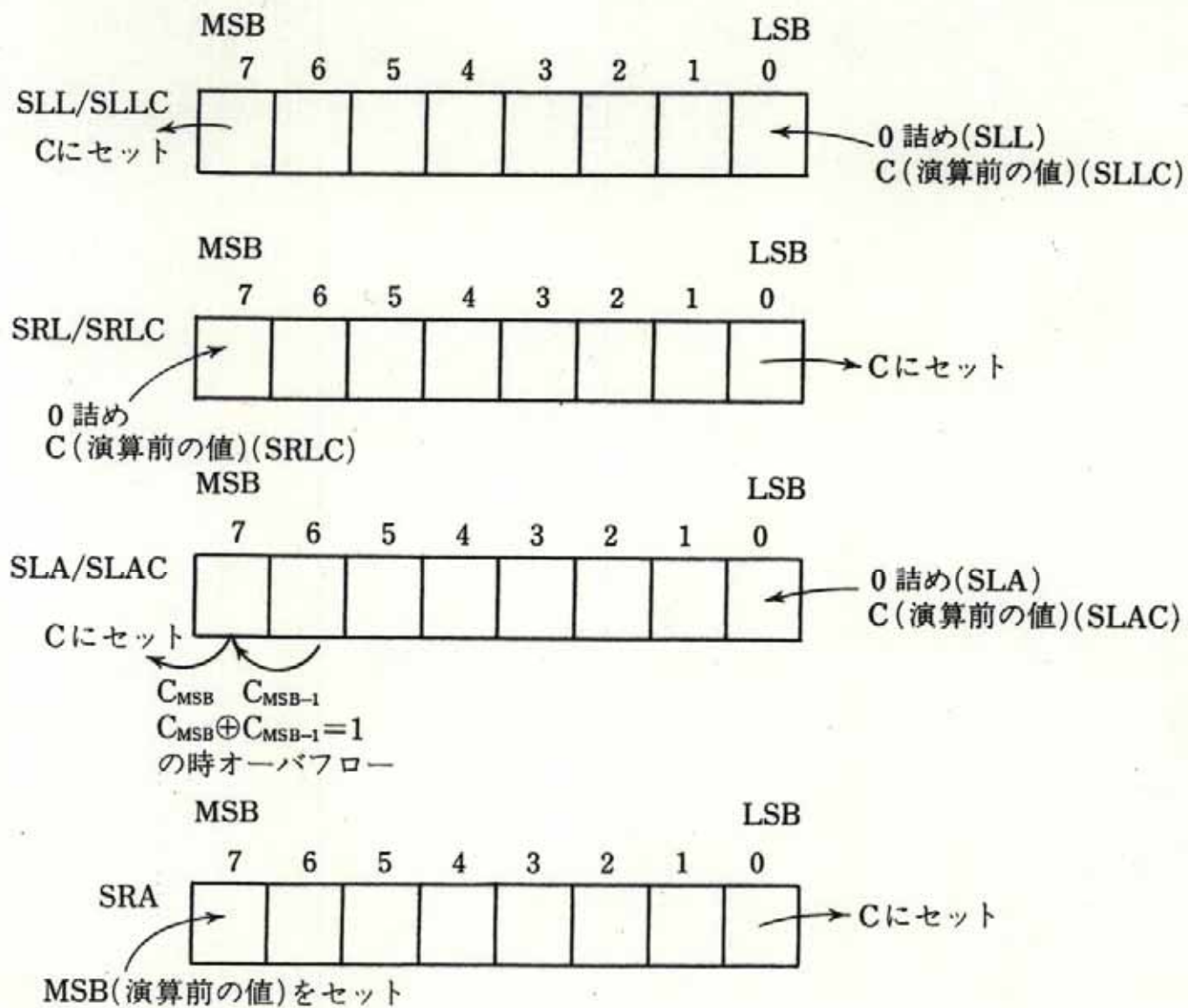
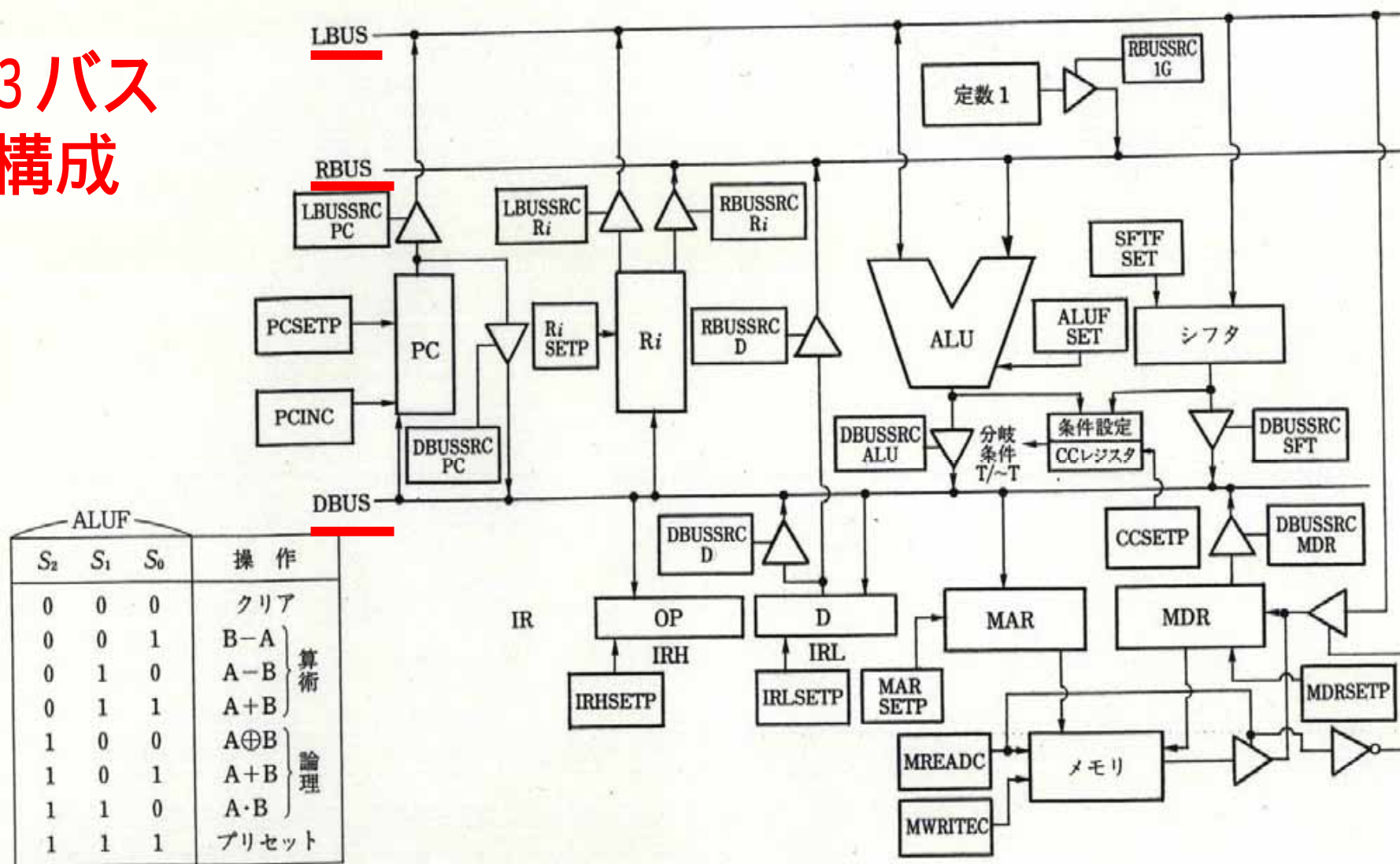


図 2.19 シフト演算

## 2 . 7 データパスの設計

# 3バス構成



算術論理演算装置

ALUなどの機能装置:ビルディングブロックとして用意されている

図 2.20 データバス

## 2 . 8 制御装置の設計

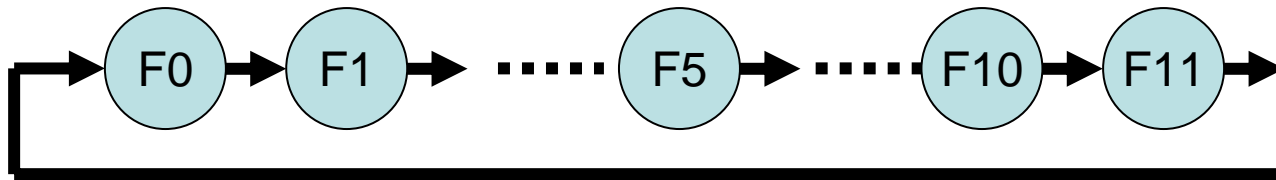
機械命令の実行

命令フェッチ

デコード ( 解読 )

演算実行 / 結果の格納

基本構造：12状態の順序回路  
で制御する



## 循環シフタで作成する

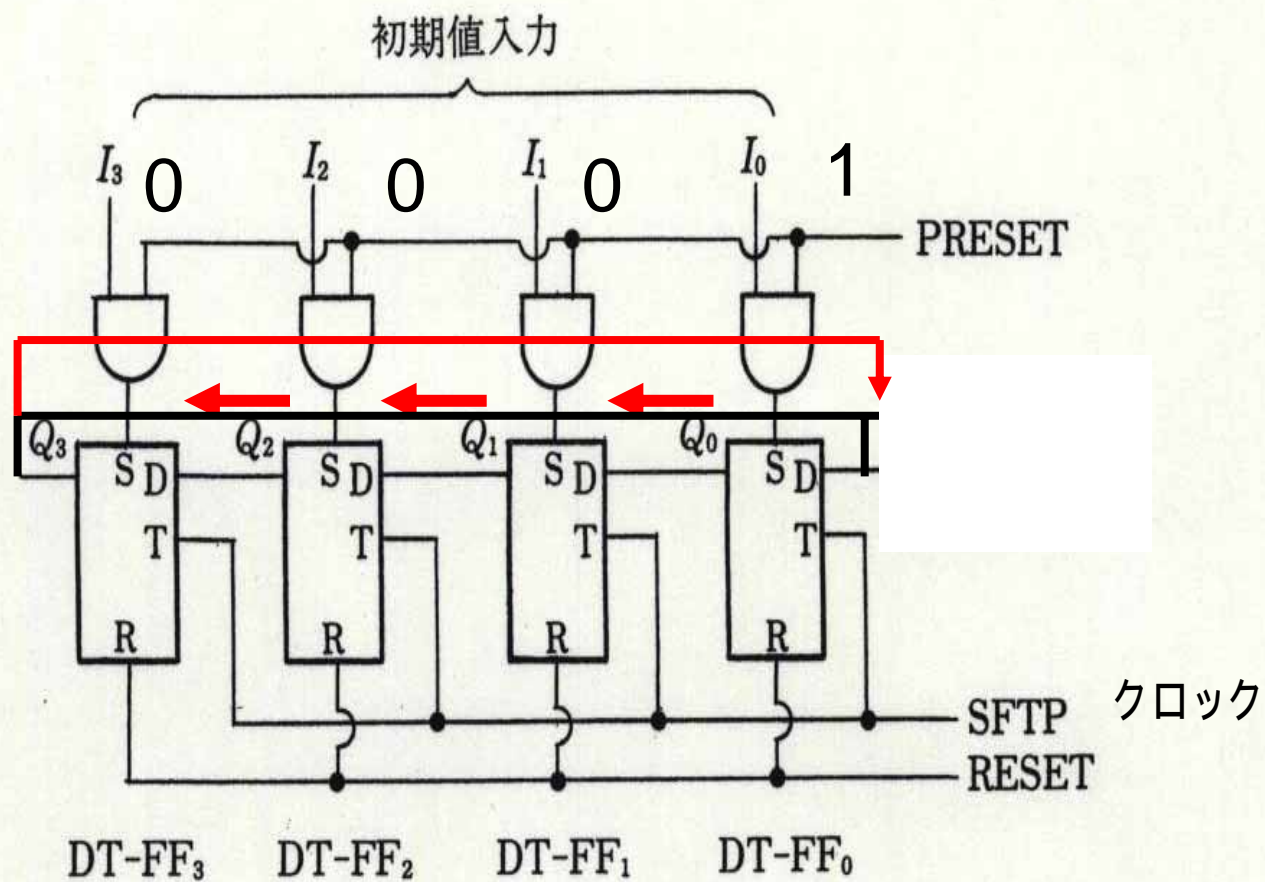
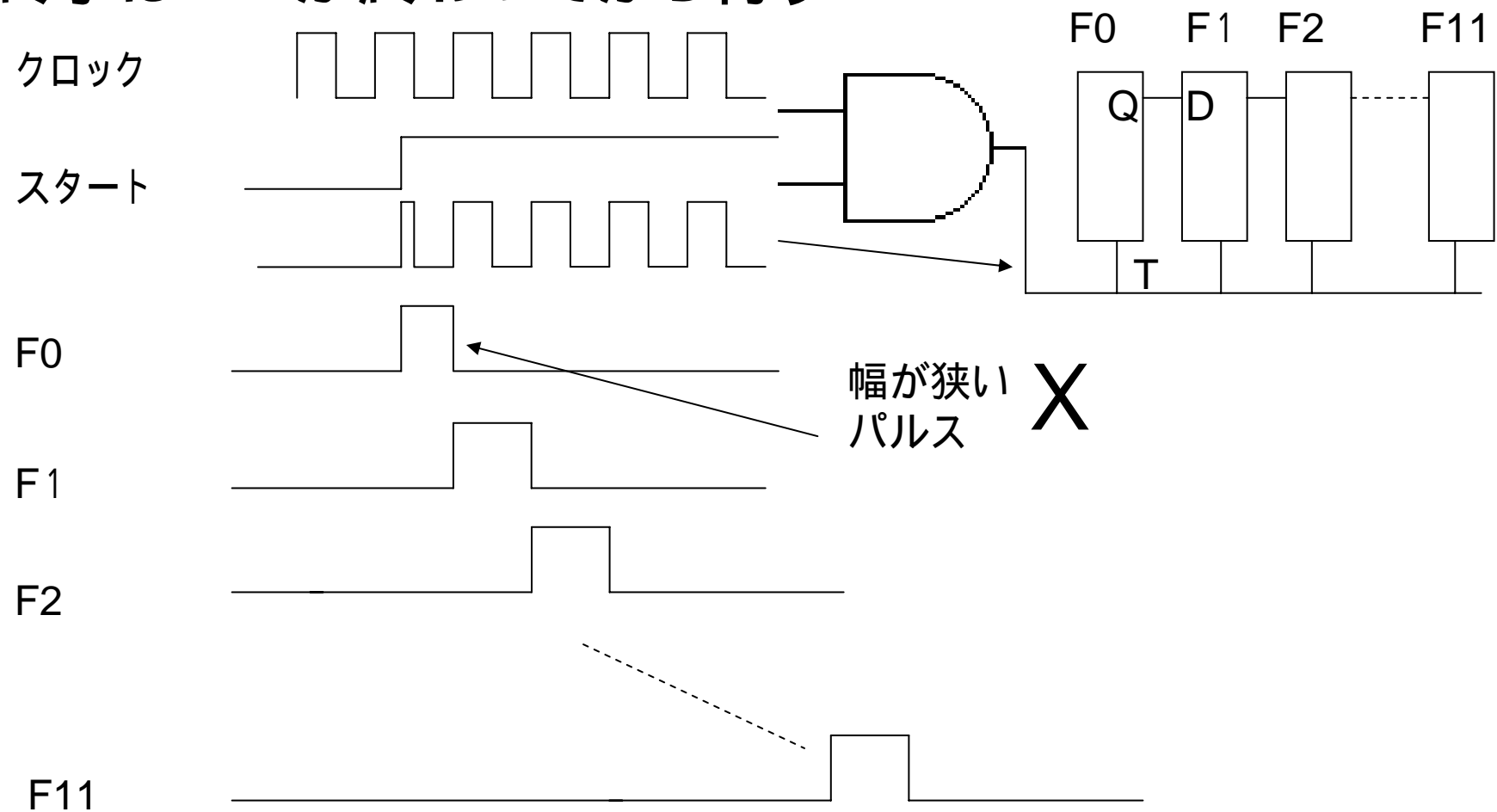


図 2.14 シフタの構成

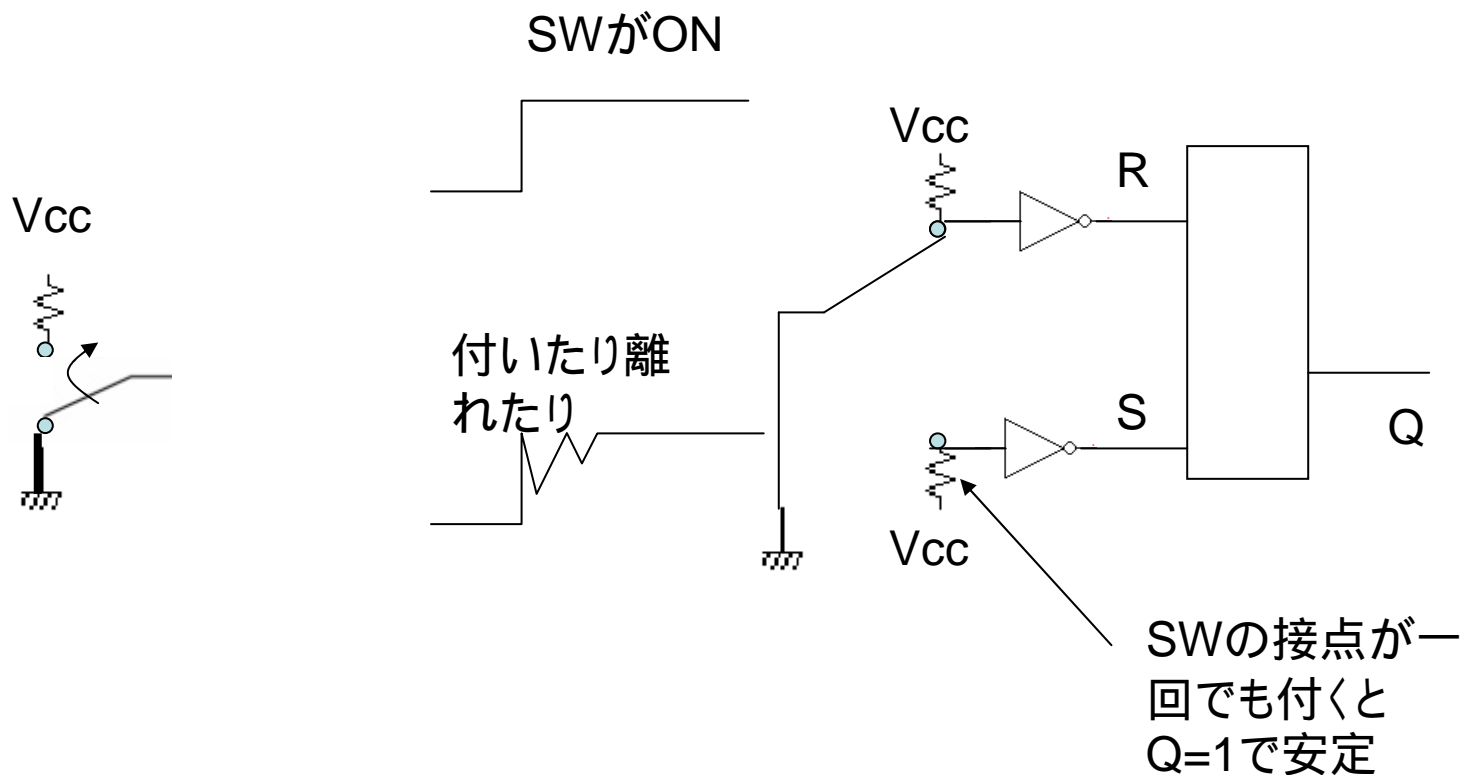
# スタート信号の入れ方が悪いとどうなる？

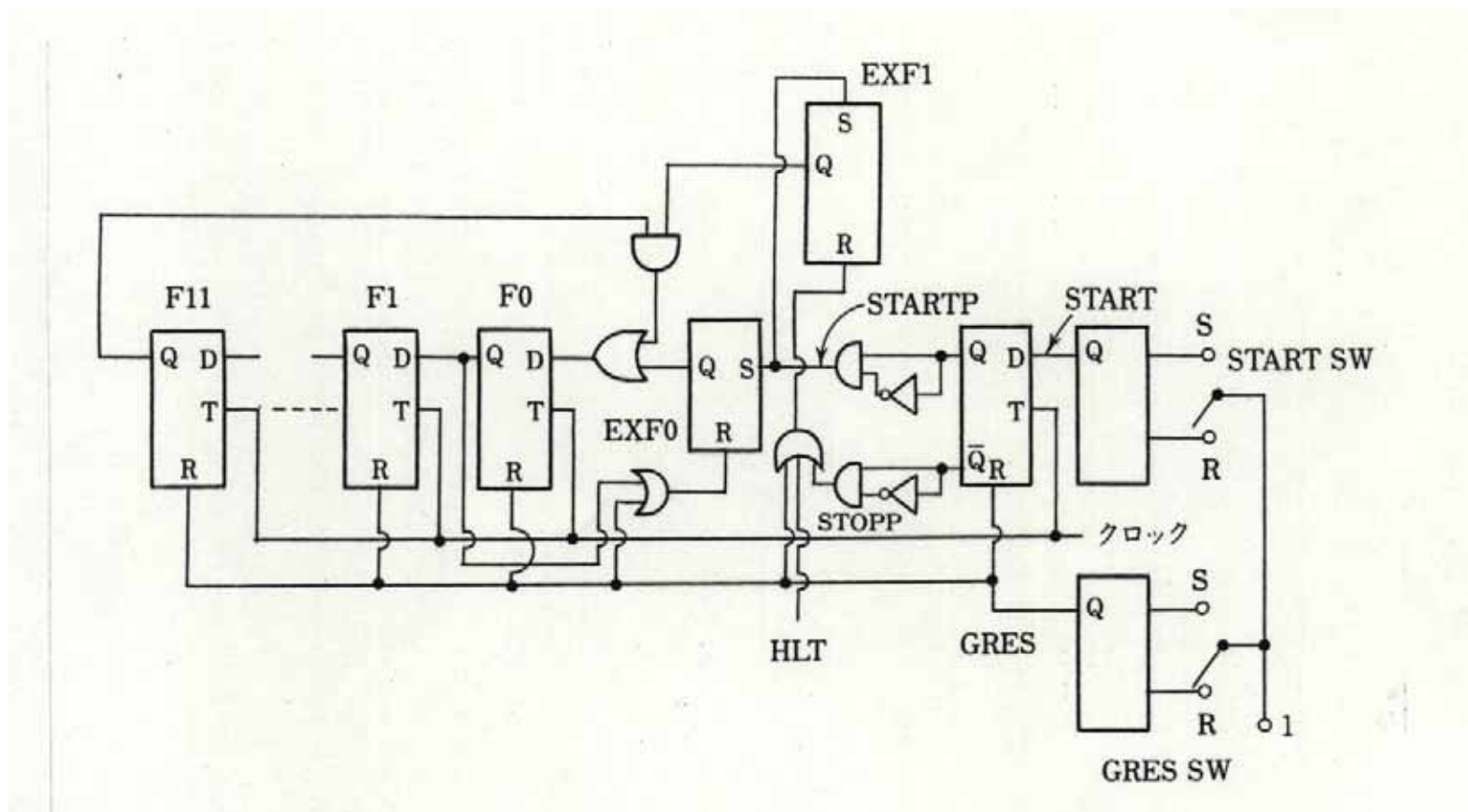
終了はF11が終わってから行う





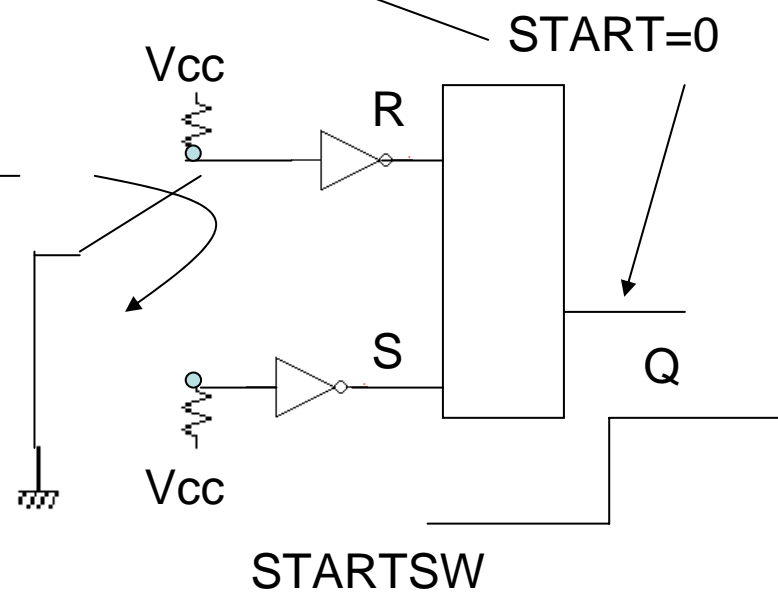
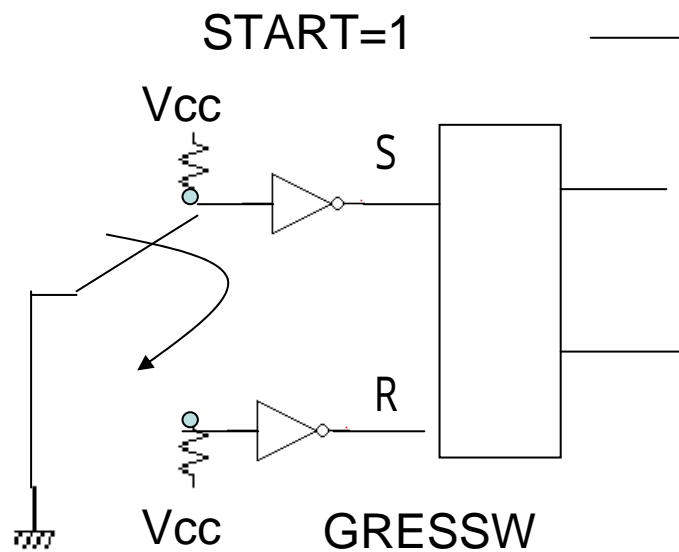
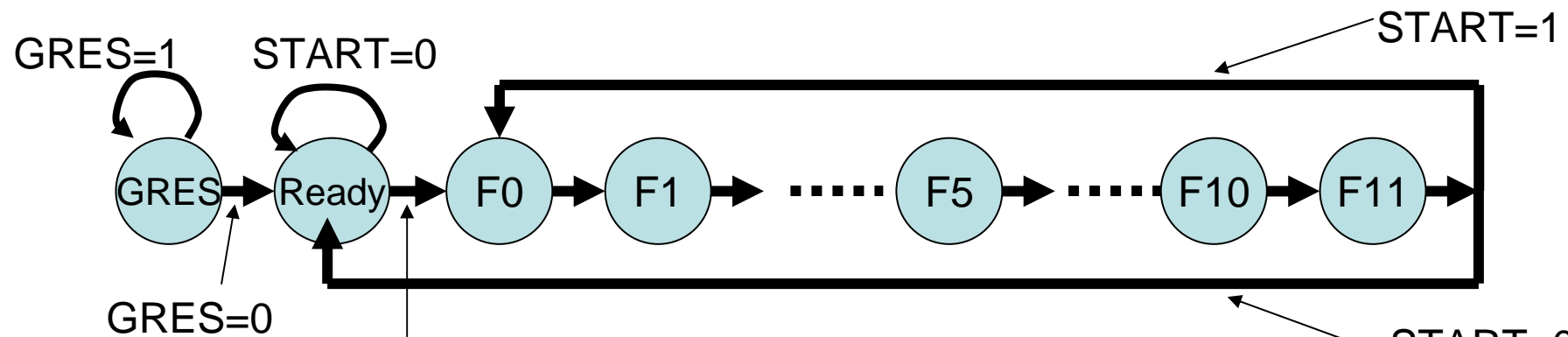
# スイッチ (SW) のチャタリング





循環シフト回路

スタートストップ回路



## 1 2 状態の制御装置

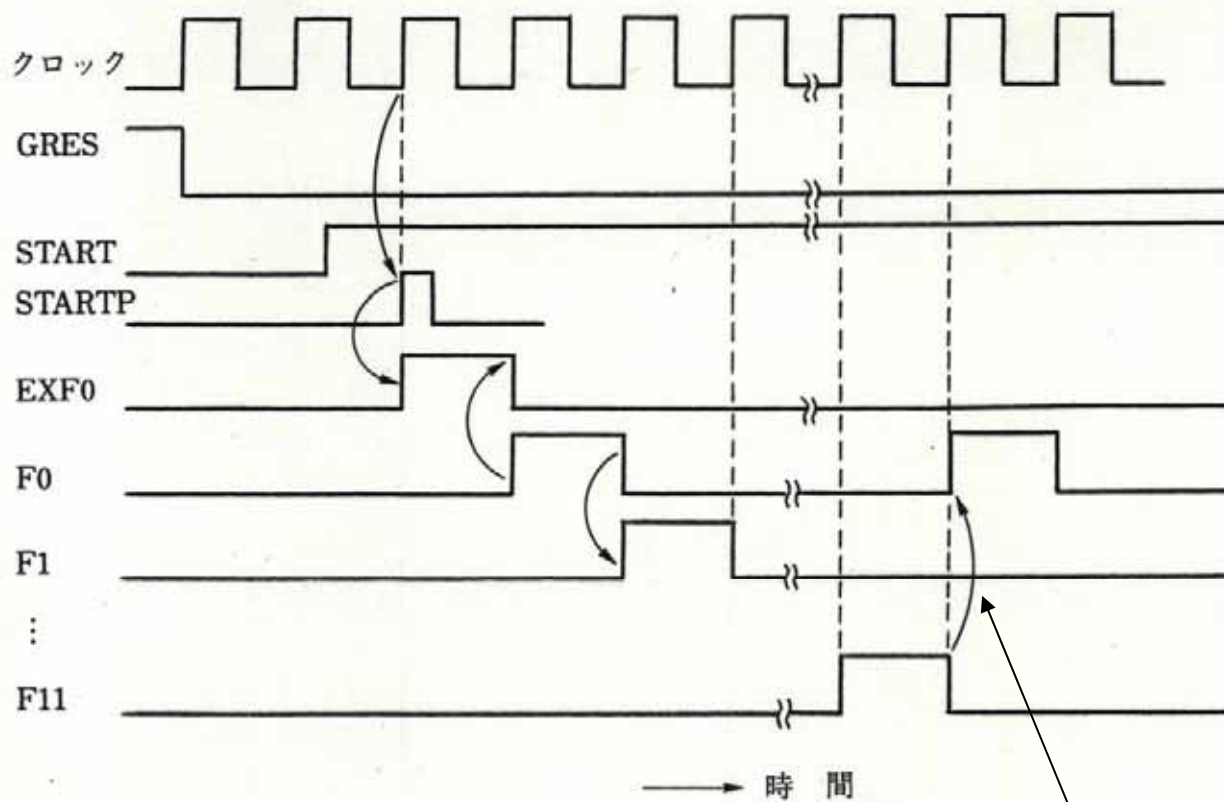


図 2.21 制御装置の構成

繰り返し



# 命令フェッチ:最初の1バイト

## 全命令共通

### 命令フェッチフェーズ

F0	F1	F2	F3
PC リード(D)	MAR にセット	M データを MDR にセット	IRH にセット
	M リード	MDR を D バスへ PCINC	

F3: デコード開始

主命令操作指定: OP1

副命令操作指定 OP2, OP3

セットアップ  
ホールドタイム  
に注意

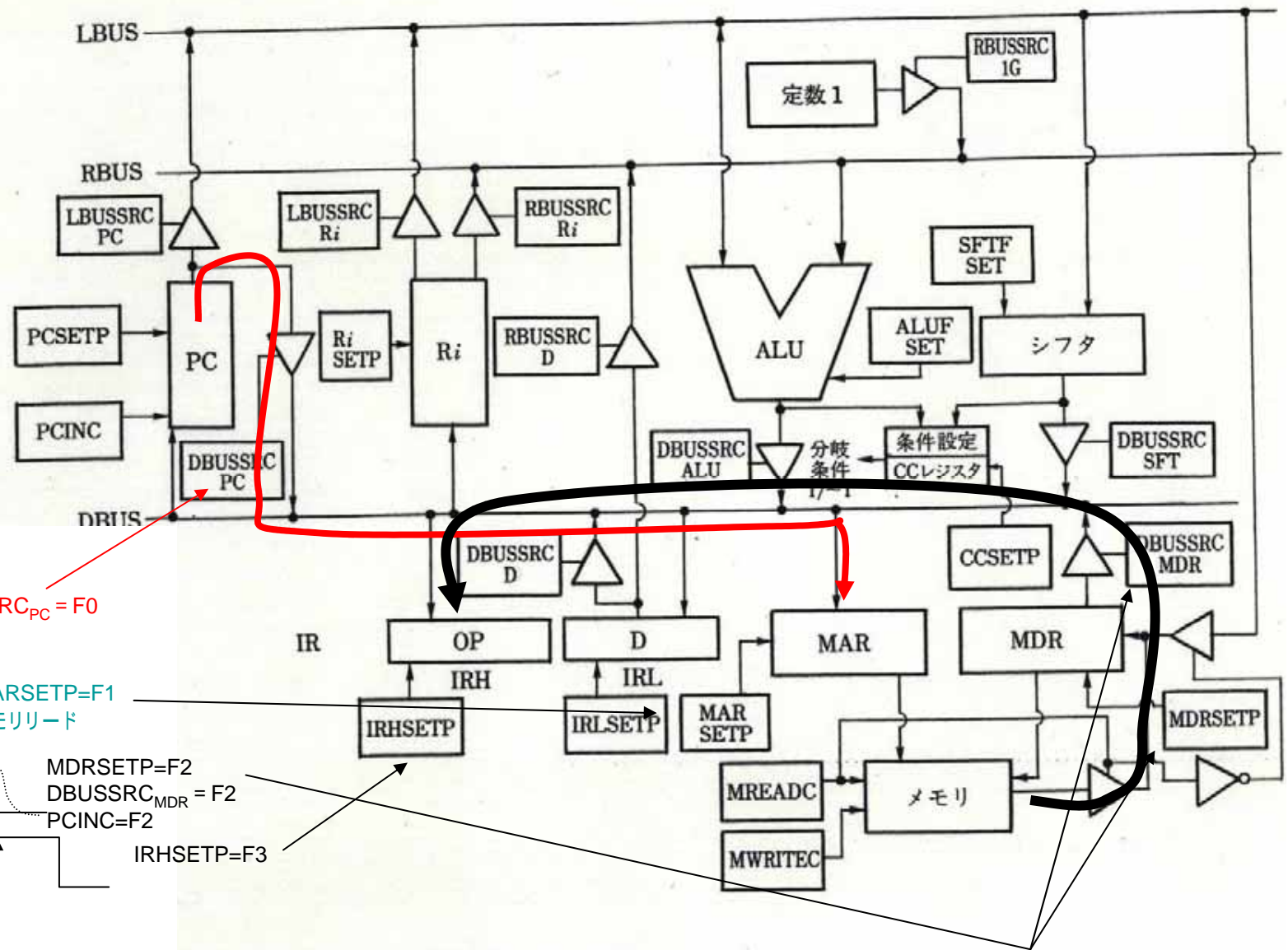
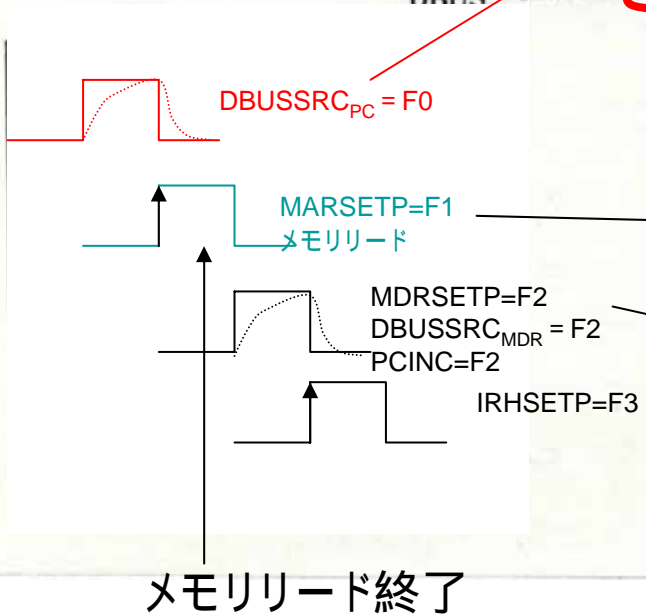


図 2.20 データバス

# デコード ALU演算の場合

F3: デコード開始

主命令操作指定: OP1

副命令操作指定 OP2, OP3

OP1 デコード結果 ADD

OP2 デコード結果  $RD_{Ri}$

OP3 デコード結果  $RS_{Ri}$

例: OP1: ADD, OP2: R0, OP3: R1

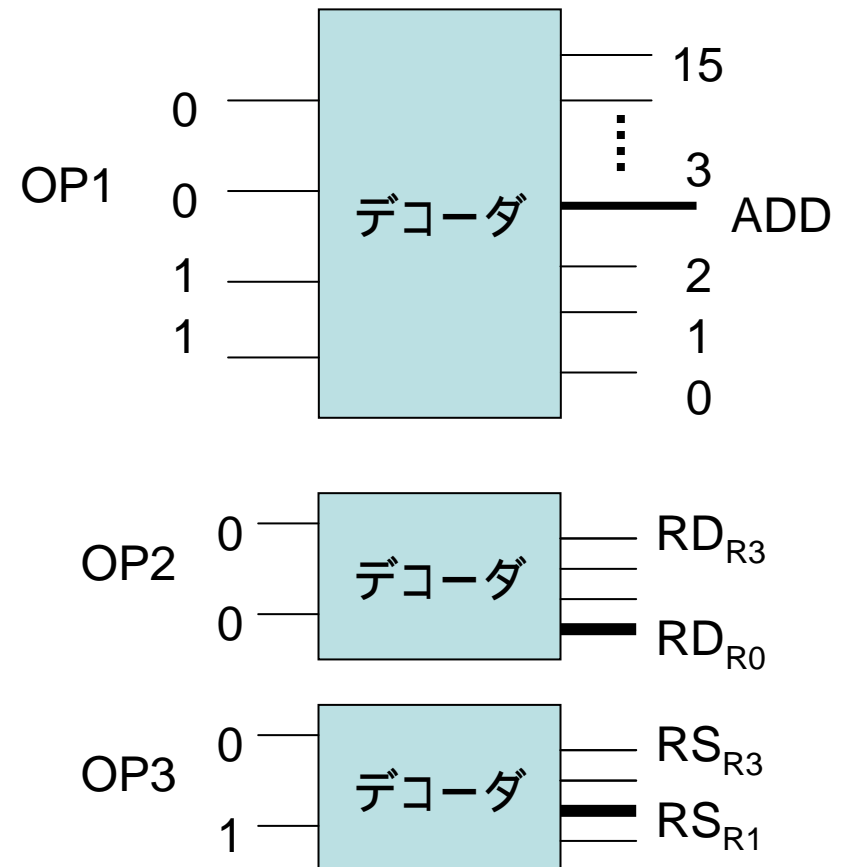
機械  
命令

OP1	OP2	OP3
0011	00	01

ADD R0 R1

SUB, XOR, OR, AND も同様

ALU = SUB + ADD + XOR + OR + AND





# デコード

## INC / DEC演算の場合

F3: デコード開始

主命令操作指定: OP1

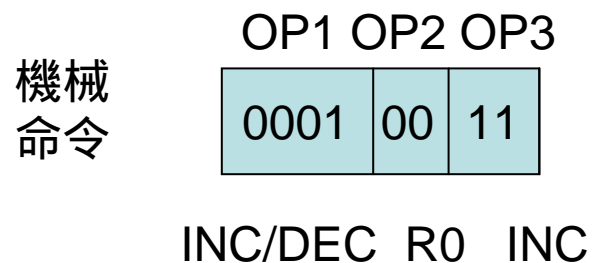
副命令操作指定 OP2, OP3

OP1 デコード結果 INC/DEC

OP2 デコード結果  $RD_{R_i}$

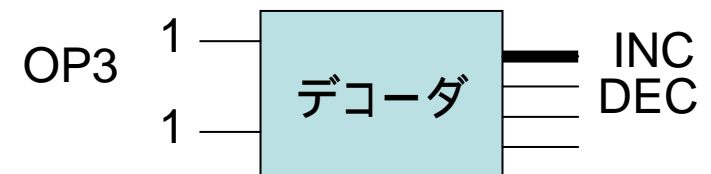
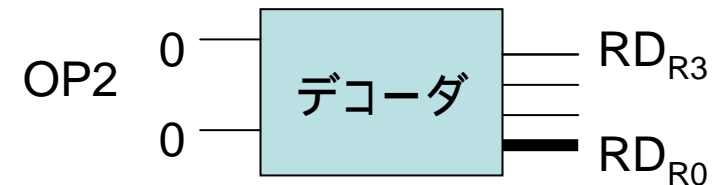
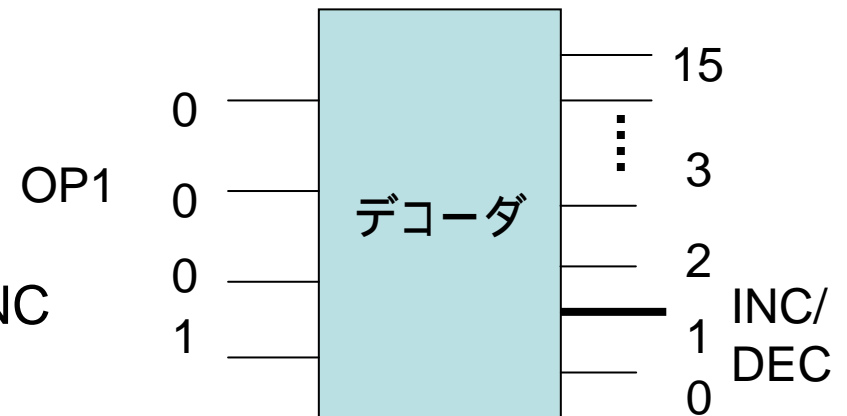
OP3 デコード結果 10: DEC, 11: INC

例: OP1: INC/DEC, OP2: R0, OP3: INC



INC: 011 ADD +1

DEC: 010 SUB -1



# デコード 条件分岐の場合

F3: デコード開始

主命令操作指定: OP1

副命令操作指定 OP2, OP3

OP1 デコード結果 AND

OP2 デコード結果 AND

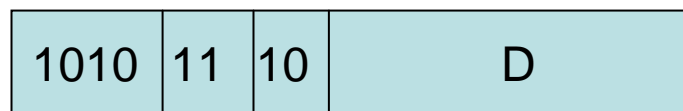
OP3 のデコード結果

例: OP1: BRC, OP2: N, OP3: S

BRCNS (Branch on Condition Negative

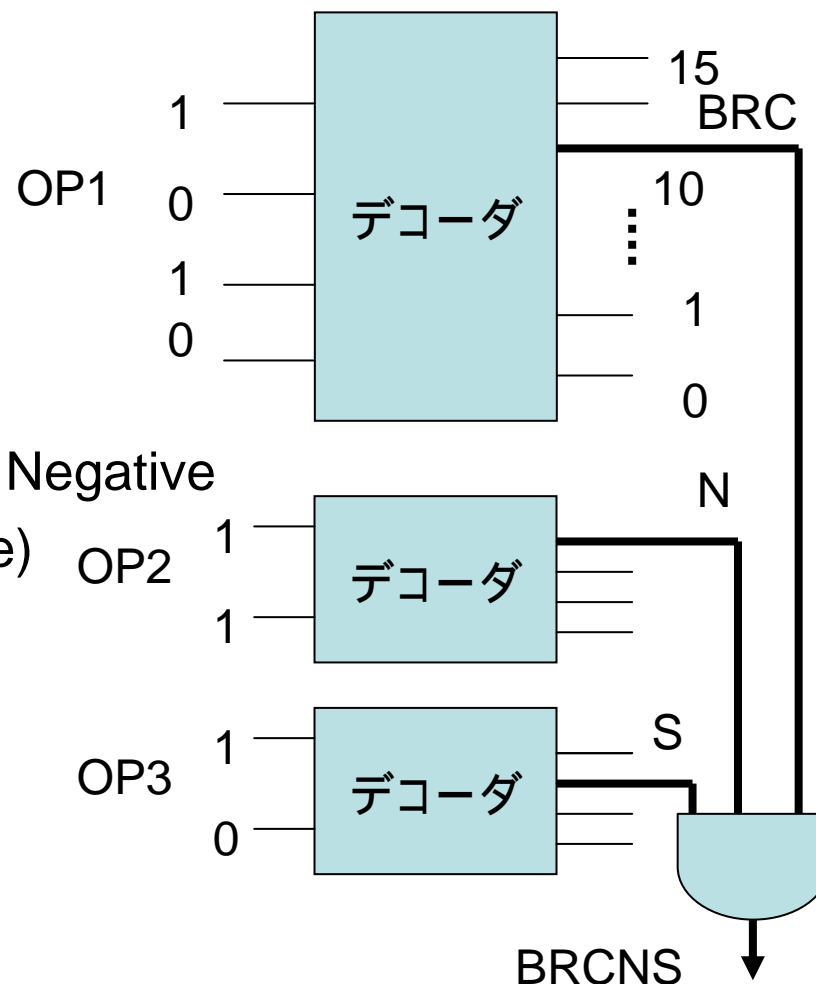
Self-Relative Addressing Mode)

OP1 OP2 OP3



BRC N S

命令が一意に決まる



# デコード BAL命令の場合

F3: デコード開始

主命令操作指定: OP1

副命令操作指定 OP2, OP3

OP1デコード結果 AND

OP2デコード結果 AND

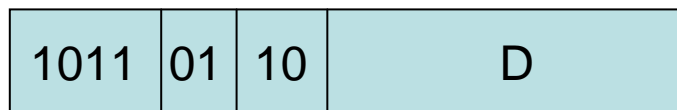
OP3のデコード結果

例: OP1: BRC, OP2: L, OP3: S

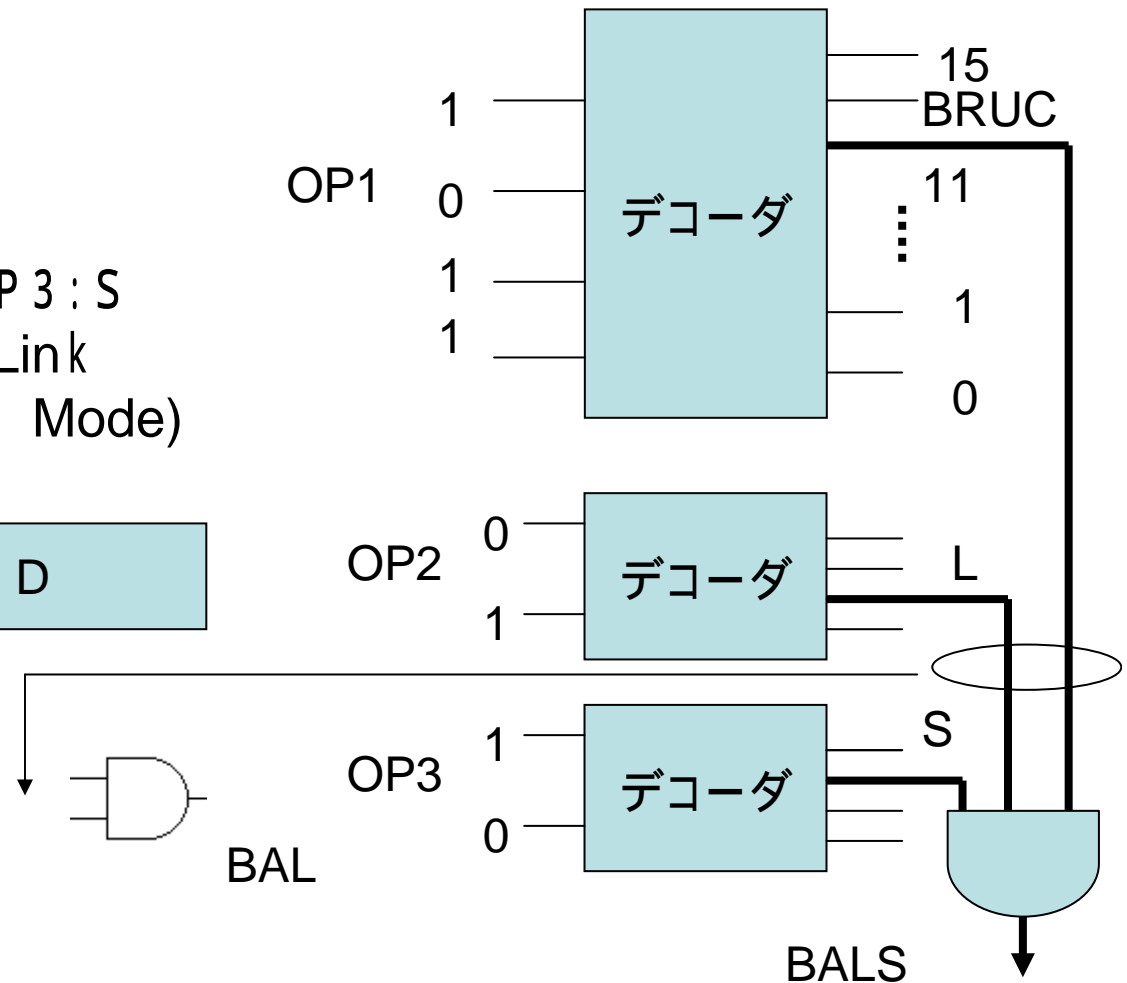
BAL (Branch and Link

Self-Relative Addressing Mode)

OP1 OP2 OP3



BRUC L S



OP1				デコード番号	
17	16	15	14		
1	1	1	1	15	STOREA
1	1	1	0	14	STOREX
1	1	0	1	13	LOADE
1	1	0	0	12	LOADX
1	0	1	1	11	BRUC
1	0	1	0	10	BRC
1	0	0	1	09	SFTA
1	0	0	0	08	SFTL
0	1	1	1	07	
0	1	1	0	06	AND
0	1	0	1	05	OR
0	1	0	0	04	XOR
0	0	1	1	03	ADD
0	0	1	0	02	SUB
0	0	0	1	01	INC/DEC
0	0	0	0	00	HLT

STORE=STOREA+STOREX  
 OP3 · A LOADA OP3 · I LOADI  
 LOADE=LOADI+LOADA  
 LOAD=LOADX+LOADE  
 OP2 · L BAL OP3 · A BALA OP3 · S BALS  
 OP2 · NL BRU OP3 · A BRUA OP3 · S BRUS  
 OP2 · N BRCN OP3 · A BRCNA OP3 · S BRCNS  
 OP2 · Z BRCZ OP3 · A BRCZA OP3 · S BRCZS  
 OP2 · C BRCC OP3 · A BRCCA OP3 · S BRCCS  
 OP2 · V BRCV OP3 · A BRCVA OP3 · S BRCVS  
 BRCA=BRC · A  
 BRCS=BRC · S  
 OP3 · RC SRAC · R SRA · LC SLAC · L SLA  
 OP3 · RC SRLC · R SRL · LC SLLC · L SLL  
 SFT=SFTL+SFTA

ALU=SUB+ADD+XOR+OR+AND  
 OP3 · I INC OP3 · D DEC

1 BYTE=HLT+INC/DEC+ALU+SFT  
 2 BYTE= ~ (HLT + INC/DEC + ALU + SFT)

OP2						
I3	I2		ALU/LOAD デコード番号	条件分岐 BRC	無条件分岐 BRUC	STORE
		↓				
1	1	3	RD <sub>R3</sub>	N		RS <sub>R3</sub>
1	0	2	RD <sub>R2</sub>	Z		RS <sub>R2</sub>
0	1	1	RD <sub>R1</sub>	C	L	RS <sub>R1</sub>
0	0	0	RD <sub>R0</sub>	V	NL	RS <sub>R0</sub>

OP3								
I1	I0		ALU	INC/ DEC	SFT	分岐モード (BM)	STOREX/ LOADX	LOADE
		↓						
			デコード番号					
1	1	3	RS <sub>R3</sub>	I	RC	S	RB <sub>R3</sub>	
1	0	2	RS <sub>R2</sub>	D	R	S	RB <sub>R2</sub>	
0	1	1	RS <sub>R1</sub>		LC	A	RB <sub>R1</sub>	A
0	0	0	RS <sub>R0</sub>		L	A	RB <sub>R0</sub>	I

図 2.23 機械命令のデコード

# 命令実行

## 命令実行前半フェーズ

### 1バイト命令

	F4	F5	F6	F7
ALU	Rs リード(R) Rd リード(L)	演算開始 結果 D バスへ	Rd ライト CC セット	NOP
INC/DEC	Rd リード(L) 定数 1 リード (R)	演算開始 結果 D バスへ	Rd ライト CC セット	NOP
SFT	Rd リード(L)	シフトセット (データセット)	シフト演算 結果 D バスへ	Rd ライト CC セット

### 2バイト命令

BRC	PC リード(D)	MAR にセット M リード	M データを MDR に MDR を D バスへ PCINC	IRL にセット
-----	-----------	-------------------	---	----------

BRUC  
BRU  
BAL  
LOADX  
LOADA  
LOADI  
STOREX  
STOREA

同上  
2 バイト命令の D フィールドの読出し



# 命令実行後半フェーズ

	F7	F8	F9	F10	F11
HLT	ストップ制御	NOP	NOP	NOP	NOP
ALU	NOP	NOP	NOP	NOP	NOP
INC/DEC	NOP	NOP	NOP	NOP	NOP
SFT	Rd ライト	NOP	NOP	NOP	NOP
以下の命令で IRL セット					
BRC	分岐テスト(T)				
	偽 BRC・~T のとき	NOP	NOP	NOP	NOP
	真 BRC・T の とき				
	BRCS のとき				
	PC, D リード (L, D)	加算 結果 D バスへ	PC にセット	NOP	NOP
BRUC BRU	BRCA のとき				
	D リード(D)	PC にセット	NOP	NOP	NOP
BRUC BRU	BRUS のとき				
	PC, D リード (L, R)	加算 結果 D バスへ	PC にセット	NOP	NOP
	BRUA のとき				
	D リード(D)	PC にセット	NOP	NOP	NOP



	F7	F8	F9	F10	F11
BAL	PC リード(D)	R0 にセット	BALS のとき PC, D リード (L, R) BALA のとき D リード(D)	加算 結果 D バスへ PC セット	PC セット NOP
LOADX	Rb, D リード (L, R)	加算 結果 D バスへ	MAR にセット M リード	M データ MDR へ MDR を D バスへ	Rd にセット
LOADA	NOP	D リード(D)	MAR にセット M リード	M データ MDR へ MDR を D バスへ	Rd にセット
LOADI	NOP	NOP	D リード(D)	Rd にセット	NOP
STOREX	Rb, D リード (L, R)	加算 結果 D バスへ	MAR にセット	Rs リード(L)	MDR セット M ライト
STOREA	NOP	D リード(D)	MAR にセット	Rs リード(L)	MDR セット M ライト

# 1バイト命令の実行: ADD R0 R1

OP1 OP2 OP3

0011 00 01

機械  
命令

命令実行前半フェーズ

ADD R0 R1

ALU演算SUB,XOR,OR,AND  
も同様

1バイト命令

	F4	F5	F6	F7
ALU	Rs リード(R) Rd リード(L)	演算開始 結果 D バスへ	Rd ライト CC セット	NOP
INC/DEC	Rd リード(L) 定数1 リード (R)	演算開始 結果 D バスへ	Rd ライト CC セット	NOP

# デコード ALU演算の場合

F3: デコード開始

主命令操作指定: OP1

副命令操作指定 OP2, OP3

OP1 デコード結果 ADD

OP2 デコード結果  $RD_{Ri}$

OP3 デコード結果  $RS_{Ri}$

例: OP1: ADD, OP2: R0, OP3: R1

機械  
命令

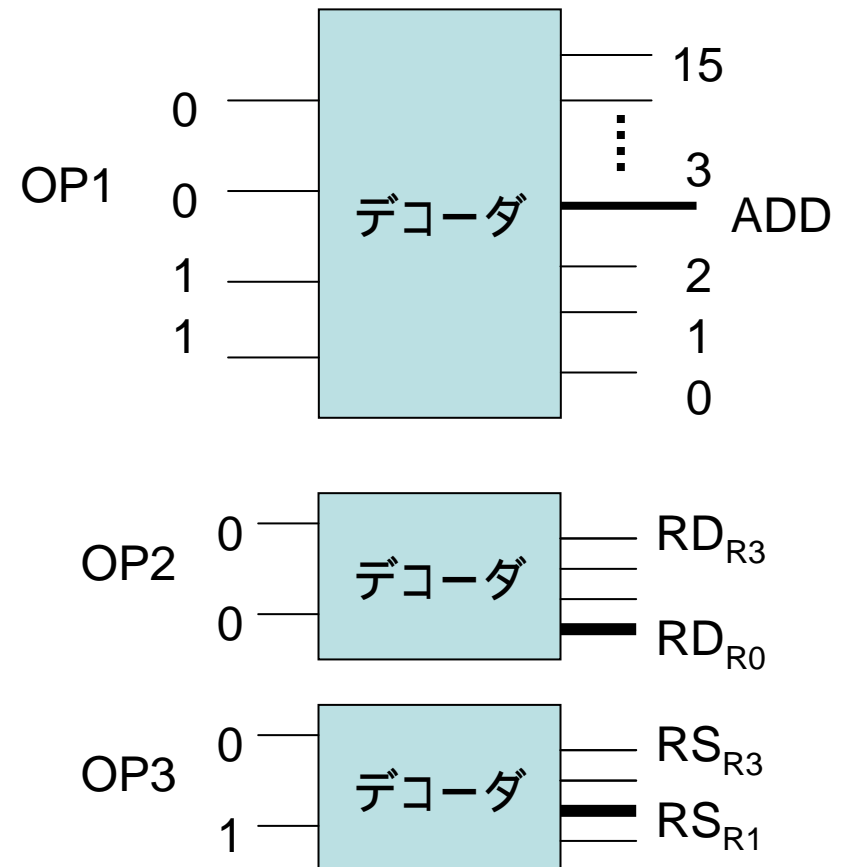
OP1 OP2 OP3

0011 00 01

ADD R0 R1

SUB, XOR, OR, AND も同様

ALU = SUB + ADD + XOR + OR + AND



## ALU演算

### 命令操作部(OP1)

SUB	0	0	1	0
ADD	0	0	1	1
XOR	0	1	0	0
OR	0	1	0	1
AND	0	1	1	0

ビット位置 17,16,15,14

### 使用する演算器

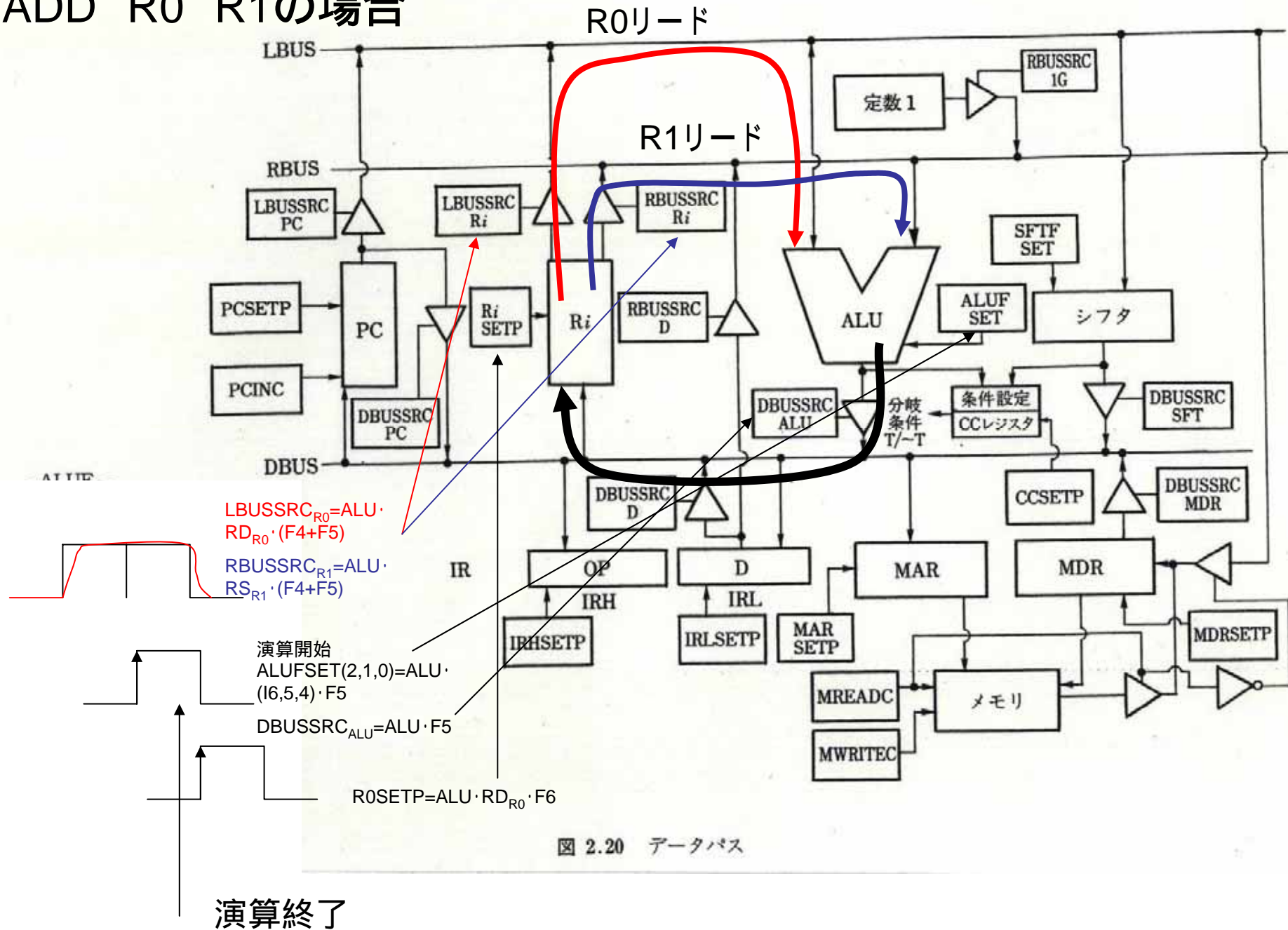
ALUF			DBUS-
S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	操 作
0	0	0	クリア
0	0	1	B-A
0	1	0	A-B
0	1	1	A+B
1	0	0	A⊕B
1	0	1	A+B
1	1	0	A·B
1	1	1	プリセット

算術論理演算装置

ALU演算の場合

命令操作部の3  
ビットをALUFに  
セット

# ADD R0 R1の場合



# INC / DEC演算の場合

F3: デコード開始

主命令操作指定: OP1

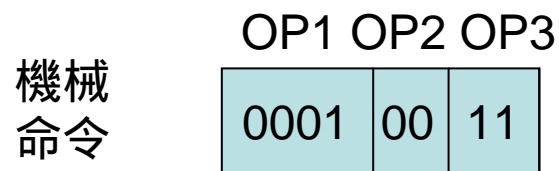
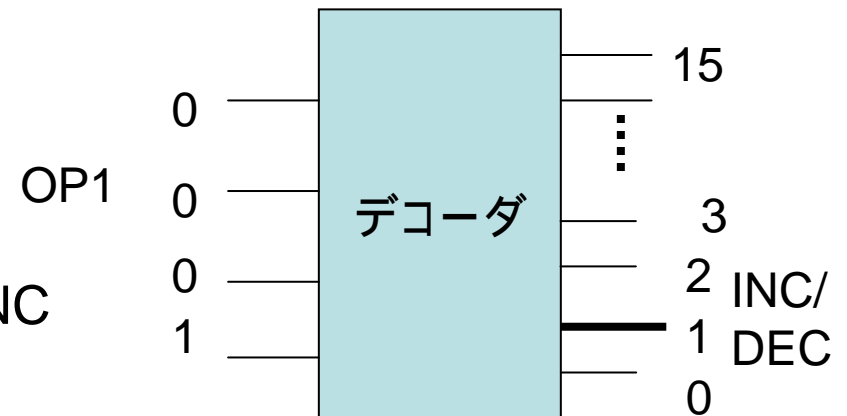
副命令操作指定 OP2, OP3

OP1 デコード結果 INC/DEC

OP2 デコード結果  $RD_{R_i}$

OP3 デコード結果 10: DEC、11: INC

例: OP1: INC/DEC、OP2: R0、OP3: INC



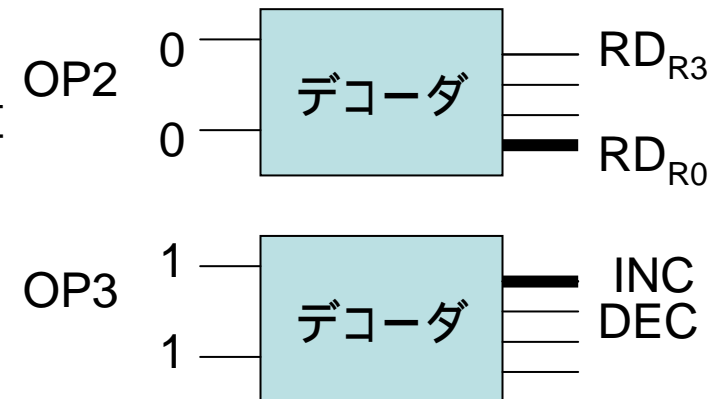
INC/DEC R0 INC

INC: 011 ADD R0 1

DEC: 010 SUB R0 1

0を付けて加算  
減算で代行

定数1の生成回路  
を利用





# INC/DEC R0の場合

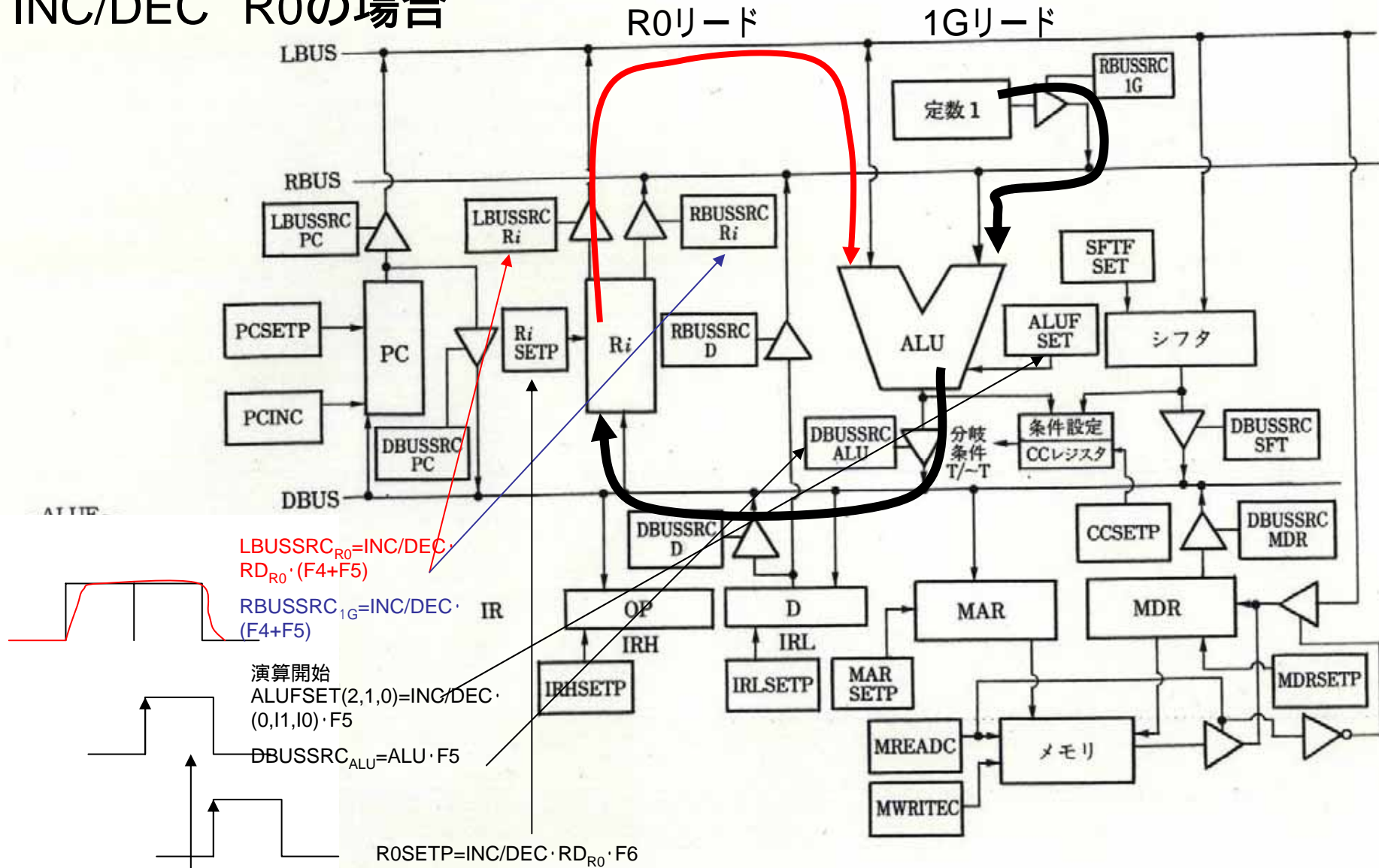


図 2.20 データバス



## 2バイト命令の実行:BALS

### 命令の下位バイトの読み出し

2バイト命令	F4	F5	F6	F7
BRC	PC リード(D)	MAR にセット	M データを MDR に MDR を D バスへ PCINC	IRL にセット
BRUC				
BRU				
BAL				
LOADX		同上		
LOADA		2 バイト命令の D フィールドの読出し		
LOADI				
STOREX				
STOREA				

TBI = BRC+BRUC+LOADX+LOADA+LOAODI+STOREX+STOREA

TBI: 2 バイト命令であることを示す信号線

# 命令の下位バイトの読み出しとIRLへのセット

セットアップ  
ホールドタイム  
に注意

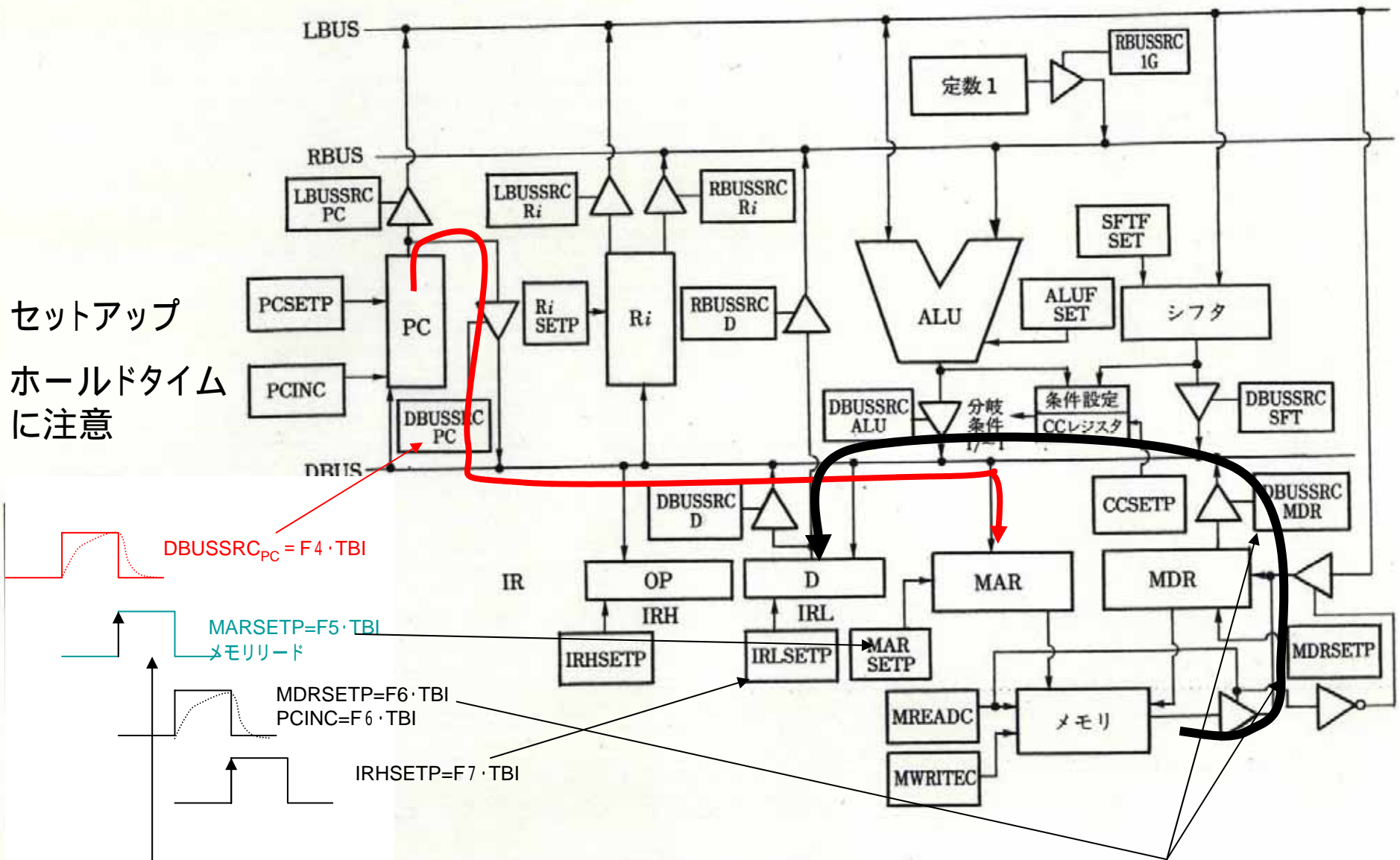


図 2.20 データバス

メモリリード終了

## BAL命令の実行

### リターンアドレスの設定とサブルーティンへの分岐

	F7	F8	F9	F10	F11
BAL	PC リード(D)	R0にセット	BALS のとき PC, D リード (L, R)	加算 結果 D バスへ	PC セット
			BALA のとき D リード(D)	PC セット	NOP

# デコード BALS命令の場合

F3: デコード開始

主命令操作指定: OP1

副命令操作指定 OP2, OP3

OP1 デコード結果 AND

OP2 デコード結果 AND

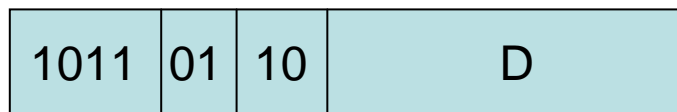
OP3 のデコード結果

例: OP1: BRC、OP2: L、OP3: S

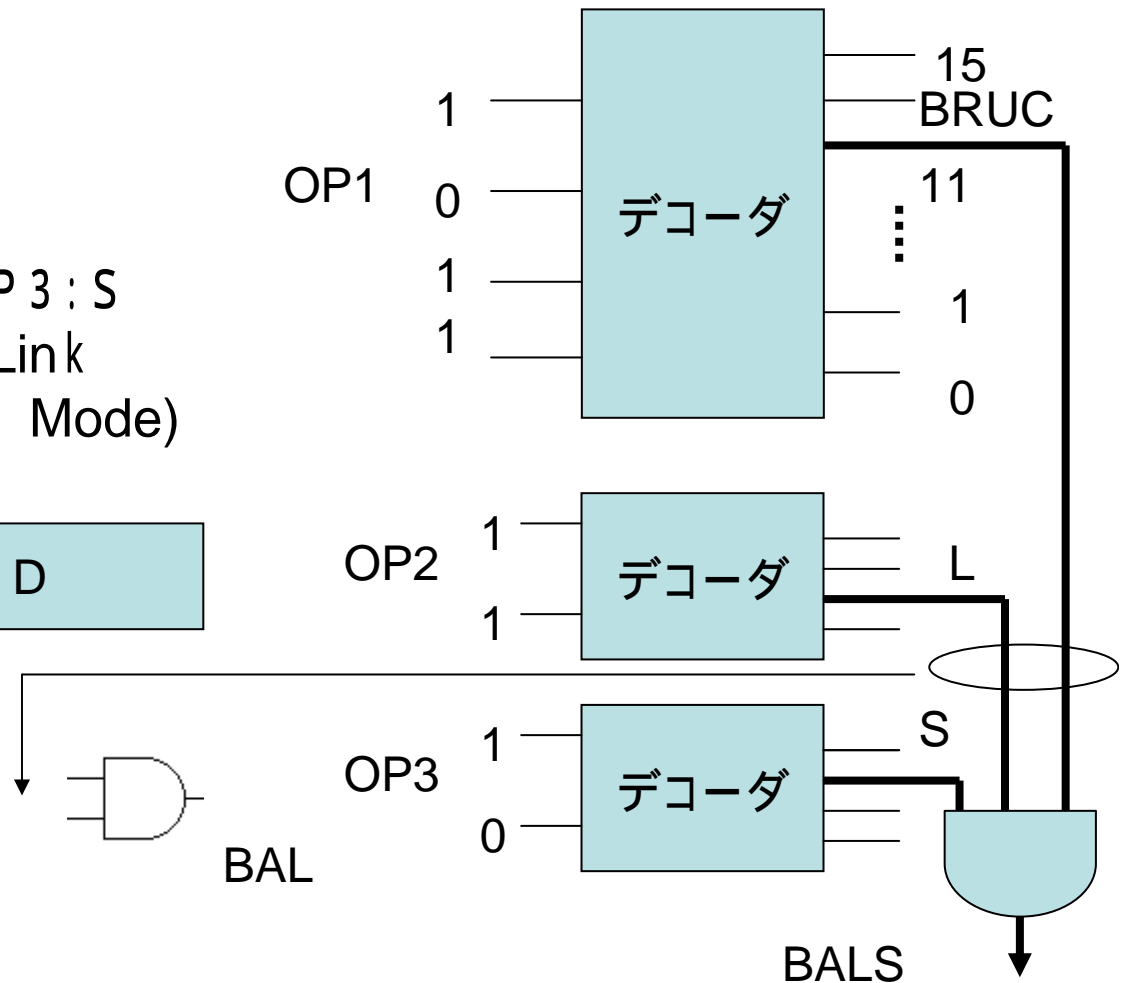
BAL (Branch and Link

Self-Relative Addressing Mode)

OP1 OP2 OP3



BRUC L S



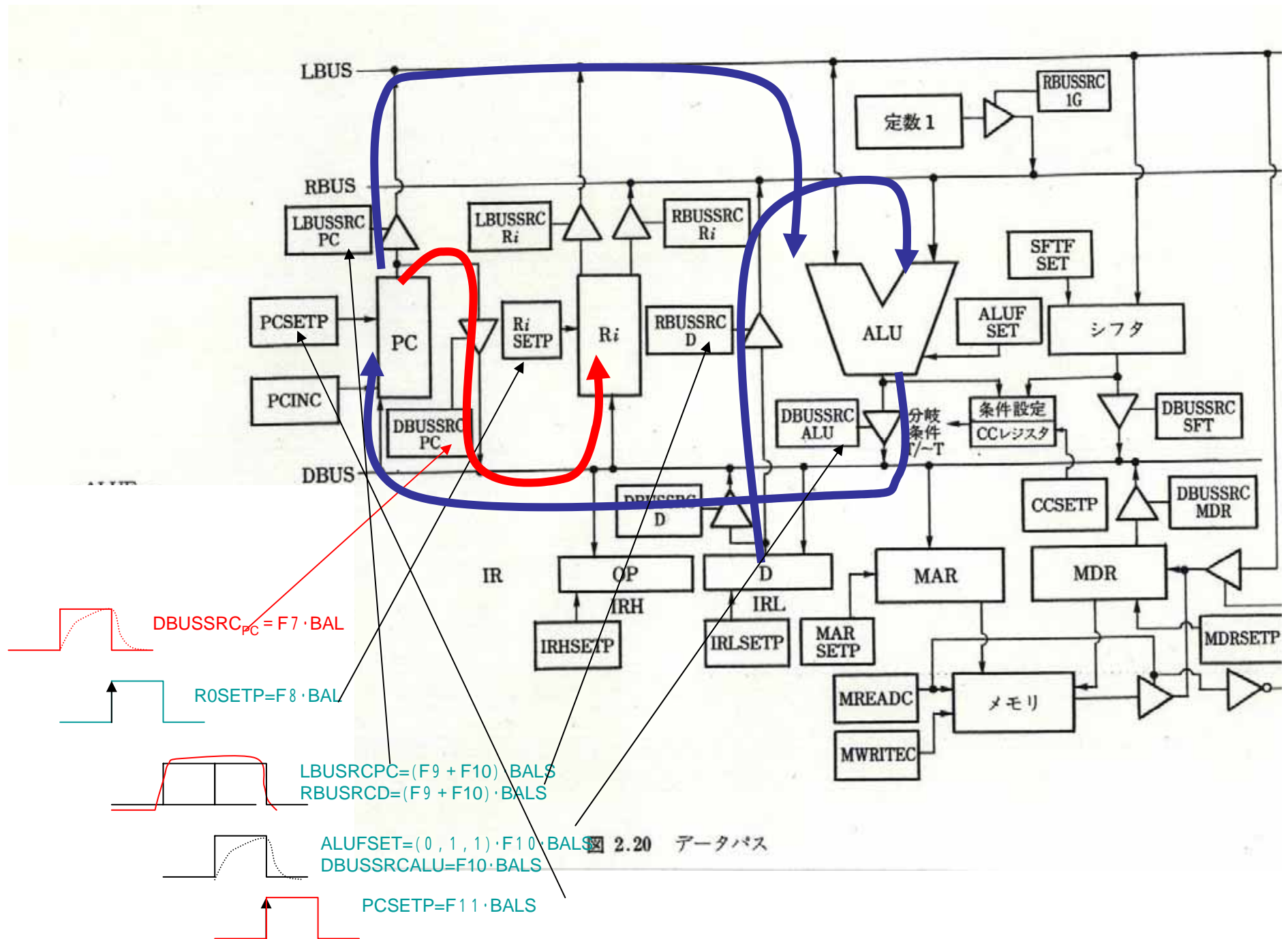


図 2.20 データバス



- ① Lバスにデータ供給する装置（添字で示す）とタイミング

$$\text{LBUSSRC}_{\text{PC}} = (\text{BRCS} \cdot \text{T} + \text{BRUS}) \cdot (\text{F7} + \text{F8}) + \text{BALS} \cdot (\text{F9} + \text{F10})$$

$$\text{LBUSSRC}_{\text{RI}} = (\text{ALU} + \text{SFT} + \text{INC/DEC}) \cdot \text{RD}_{\text{RI}} \cdot (\text{F4} + \text{F5}) + (\text{LOADX} + \text{STOREX}) \cdot \text{RB}_{\text{RI}} \cdot (\text{F7} + \text{F8}) + (\text{STOREX} + \text{STOREA}) \cdot \text{RS}_{\text{RI}} \cdot \text{F10}$$

- ② Rバスにデータ供給する装置（添字で示す）とタイミング

$$\text{RBUSSRC}_{\text{RI}} = \text{ALU} \cdot \text{RS}_{\text{RI}} \cdot (\text{F4} + \text{F5})$$

$$\text{RBUSSRC}_{\text{CD}} = (\text{BRCS} \cdot \text{T} + \text{BRUS}) \cdot (\text{F7} + \text{F8}) + \text{BALS} \cdot (\text{F9} + \text{F10}) + (\text{LOADX} + \text{STOREX}) \cdot (\text{F7} + \text{F8})$$

$$\text{RBUSSRC}_{\text{IG}} = \text{INC/DEC} \cdot (\text{F4} + \text{F5})$$

- ③ Dバスにデータ供給する装置（添字で示す）とタイミング

$$\text{DBUSSRC}_{\text{ALU}} = (\text{ALU} + \text{INC/DEC}) \cdot \text{F5} + (\text{BRCS} \cdot \text{T} + \text{BRUS}) \cdot \text{F8} + \text{BALS} \cdot \text{F10} + (\text{LOADX} + \text{STOREX}) \cdot \text{F8}$$

$$\text{DBUSSRC}_{\text{SFT}} = \text{SFT} \cdot \text{F6}$$

$$\text{DBUSSRC}_{\text{D}} = (\text{BRCA} \cdot \text{T} + \text{BRUA}) \cdot \text{F7} + \text{BALA} \cdot \text{F9} + \text{LOADA} \cdot \text{F8} + \text{LOADI} \cdot \text{F9} + \text{STOREA} \cdot \text{F8}$$

$$\text{DBUSSRC}_{\text{MDR}} = \text{F2} + (\sim (\text{HLT} + \text{ALU} + \text{INC/DEC} + \text{SFT})) \cdot \text{F6} + (\text{LOADX} + \text{LOADA}) \cdot \text{F10}$$

$$\text{DBUSSRC}_{\text{PC}} = \text{F0} + (\sim (\text{HLT} + \text{ALU} + \text{INC/DEC} + \text{SFT})) \cdot \text{F4} + \text{BAL} \cdot \text{F7}$$

④ 各装置（先頭の数文字で表す）へのセットパルス

$$\text{PCSETP} = \text{BAL} \cdot \text{F7} + (\text{BRCS} \cdot \text{T} + \text{BRUS}) \cdot \text{F9} + (\text{BRCA} + \text{BRUA}) \cdot \text{F8} + \text{BALS} \cdot \text{F11} \\ + \text{BALA} \cdot \text{F10}$$

$$\text{PCINC} = \text{F2} + (\sim (\text{HLT} + \text{ALU} + \text{INC/DEC} + \text{SFT})) \cdot \text{F6}$$

$$\text{IRHSETP} = \text{F2}$$

$$\text{IRLSETP} = (\sim (\text{HLT} + \text{ALU} + \text{INC/DEC} + \text{SFT})) \cdot \text{F7}$$

$$\text{RiSETP} = ((\text{ALU} + \text{INC/DEC}) \cdot \text{F6} + \text{SFT} \cdot \text{F7} + (\text{LOADX} + \text{LOADA}) \cdot \text{F11} + \\ \text{LOADI} \cdot \text{F10}) \cdot \text{RD}_{\text{Ri}}$$

$$\text{R0SETP} = ((\text{ALU} + \text{INC/DEC}) \cdot \text{F6} + \text{SFT} \cdot \text{F7} + (\text{LOADX} + \text{LOADA}) \cdot \text{F11} + \\ \text{LOADI} \cdot \text{F10}) \cdot \text{RD}_{\text{R0}} + \text{BAL} \cdot \text{F8}$$

$$\text{MARSETP} = \text{F1} + (\sim (\text{HLT} + \text{ALU} + \text{INC/DEC} + \text{SFT})) \cdot \text{F5} + (\text{LOADX} + \text{LOADA}) \cdot \text{F9} + \\ (\text{STOREX} + \text{STOREA}) \cdot \text{F9}$$

$$\text{MDRSETP} = \text{F2} + (\sim (\text{HLT} + \text{ALU} + \text{INC/DEC} + \text{SFT})) \cdot \text{F6} + ((\text{STOREX} + \text{STOREA}) \cdot \\ \text{F11} + (\text{LOADA} + \text{LOADX}) \cdot \text{F10})$$

$$\text{CCSETP} = (\text{ALU} + \text{INC/DEC}) \cdot \text{F6} + \text{SFT} \cdot \text{F7}$$

⑤ 演算装置やシフタでの操作指定

$$\text{ALUFSET}(2, 1, 0) = (\text{ALU} \cdot (\text{I7}, \text{I6}, \text{I5}) + \text{INC/DEC} \cdot (0, \text{I1}, \text{I0})) \cdot \text{F5} \\ + (0, 1, 1) (\text{BRCCS} \cdot \text{T} + \text{BRUS} + \text{LOADX} + \text{STOREX}) \cdot \text{F8} \\ + \text{BALS} \cdot \text{F10}$$

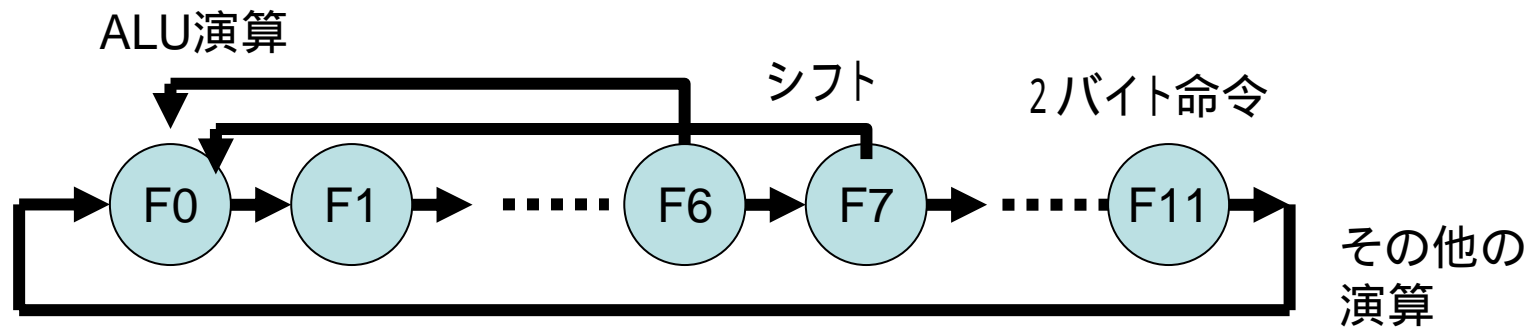
$$\text{SFTFSET}(2, 1, 0) = \text{SFT} \cdot (\text{I4}, \text{I1}, \text{I0}) \cdot \text{F6}$$

$$\text{SFTSETP} = \text{SFT} \cdot \text{F5}$$

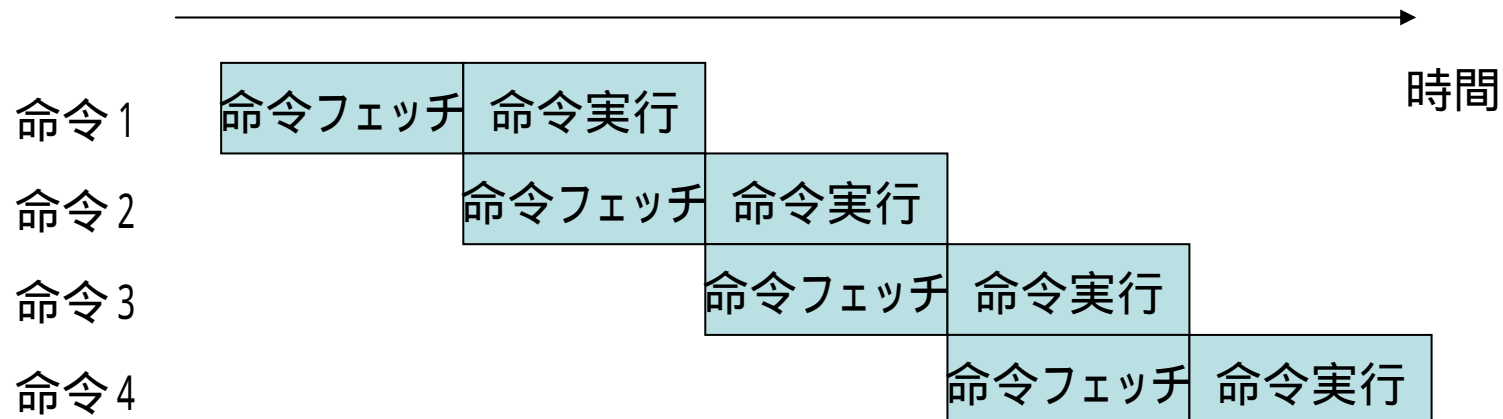
図 2.24 制御信号のタイミング一覧

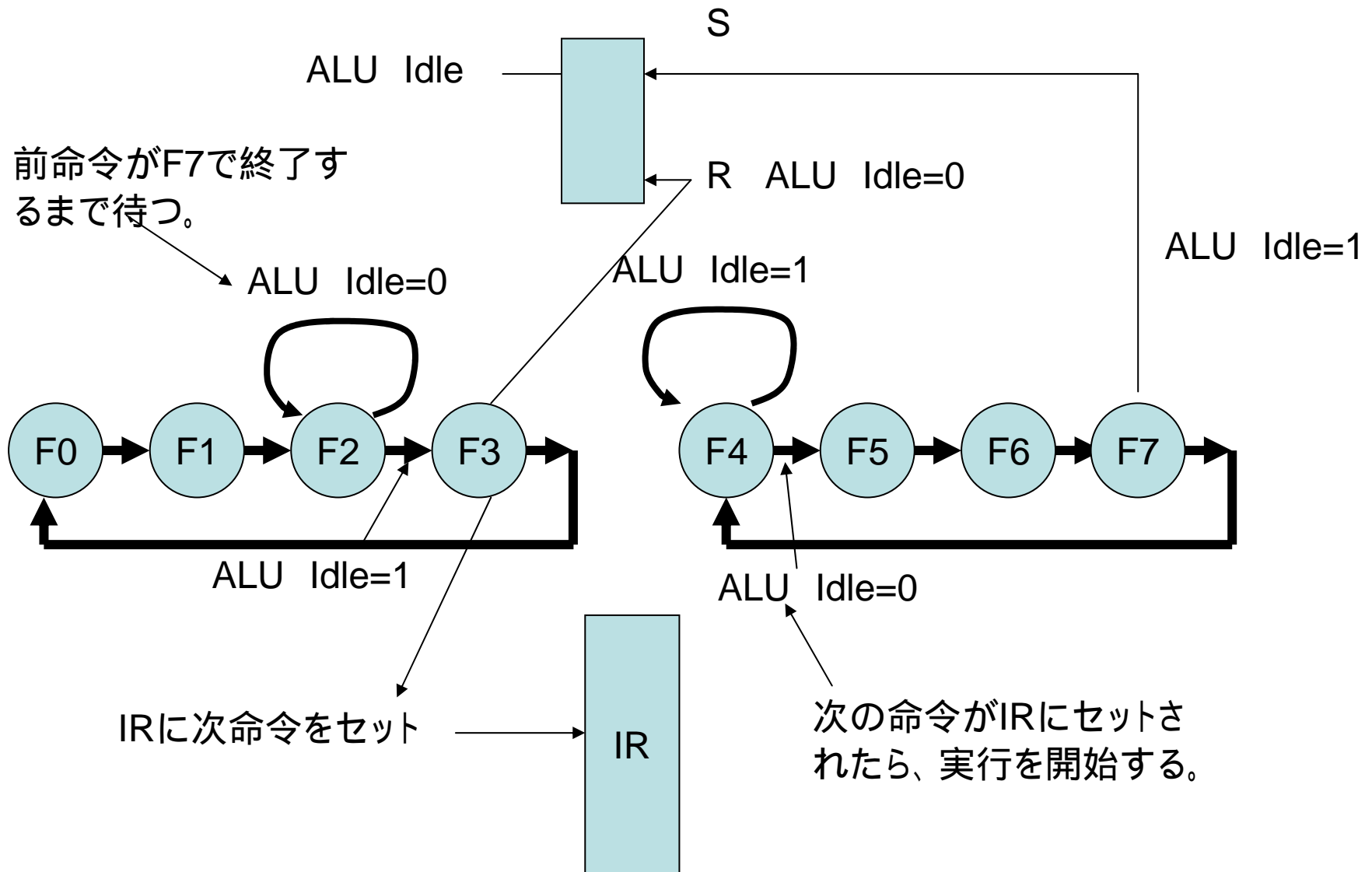


## 高速化1: 早く終了する命令は 次命令を早く実行



## 高速化2: 流れ作業(パイプライン)の導入





## 機械命令の2ステージパイプライン実行: 命令フェッチと演算実行

## 2.10 電子回路の基礎

# 電子回路

## (i) 微分回路

ステップ状の信号  $E$  ( $t < 0$  で  $0$ ,  $0 < t$  で  $E$ ) 入力

$$V_o = V_i - Q/C = iR$$

$$i = dQ/dt$$

が成り立つので,

$$dV_o/dt + V_o/(CR) = dV_i/dt$$

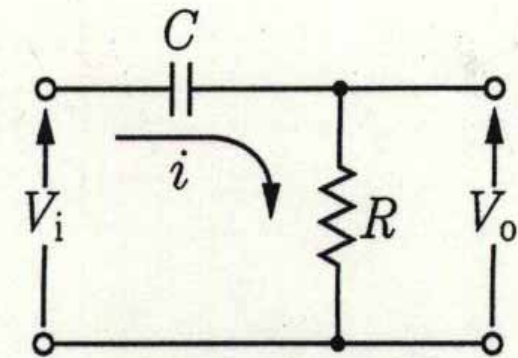
$t > 0$  で  $dV_i/dt = 0$ ,  $t = 0$  で  $V_o = E$  であるので

$$V_o = E \exp(-t/(CR))$$

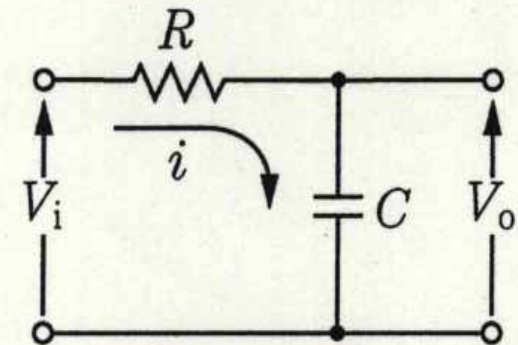
初期条件の与え方に注意

## (ii) 積分回路

$$V_o = E(1 - \exp(-t/(CR)))$$



(a) 微分回路



(b) 積分回路

## 積分回路

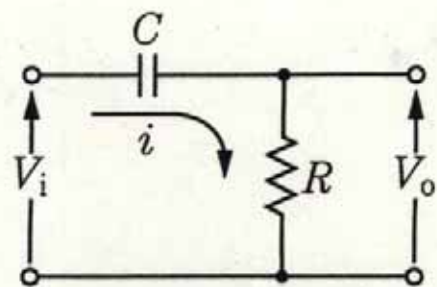
$$\text{電圧} : V_0 = E(1 - \exp(-\frac{t}{CR}))$$

$$\text{電流} : i = C \frac{dV_0}{dt} = \frac{E}{R} \exp(-\frac{t}{CR})$$

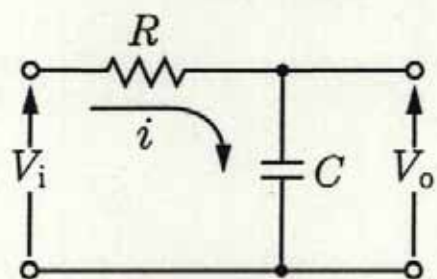
$$\text{消費電力} : P = \int_0^{\infty} E i dt = CE^2$$

$$\text{コンデンサ蓄積分} : \frac{CE^2}{2}$$

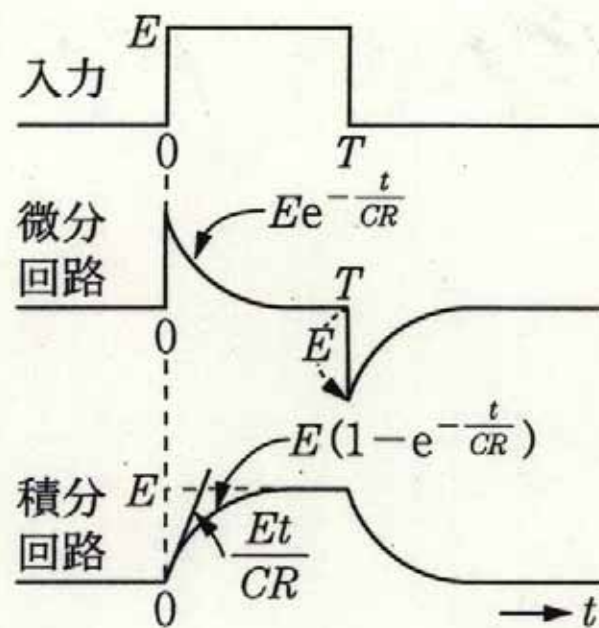
$$\text{抵抗消費分} : R \int_0^{\infty} i^2 dt = \frac{CE^2}{2}$$



(a) 微分回路



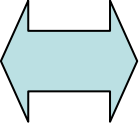
(b) 積分回路



(c) パルス応答

図 2.25 微分積分回路

伝送線路 1 : LC 線路(同軸、プリント基板)

分布定数回路  集中定数回路

$x$  の地点での電圧, 電流:  $v(x,t), i(x,t)$

- $\frac{\partial v}{\partial x} = -L \frac{\partial i}{\partial t} - Ri$
- $\frac{\partial i}{\partial x} = -C \frac{\partial v}{\partial t} - Gv$

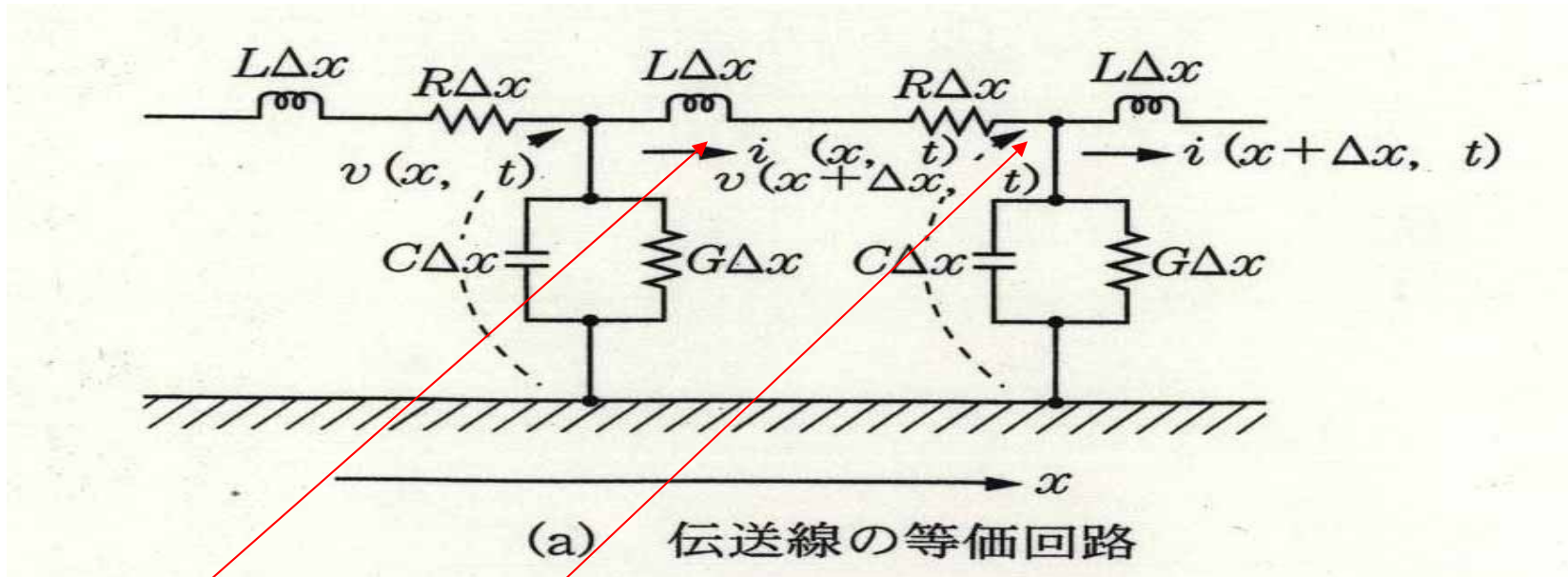
となる.  $R=G=0$  の損失のない理想伝送線路では

- $\frac{\partial^2 v}{\partial x^2} = -L \frac{\partial^2 v}{\partial t^2}$
- $\frac{\partial^2 i}{\partial x^2} = -C \frac{\partial^2 i}{\partial t^2}$

これは

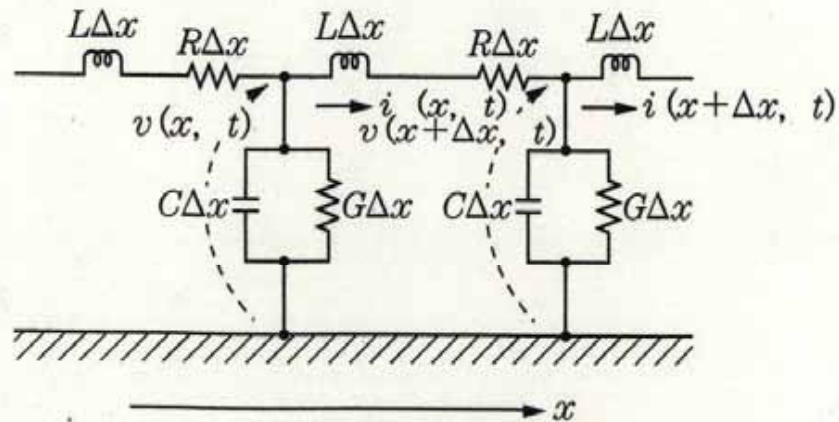
$$\frac{\partial^2 v}{\partial x^2} - LC \frac{\partial^2 v}{\partial t^2} = 0$$



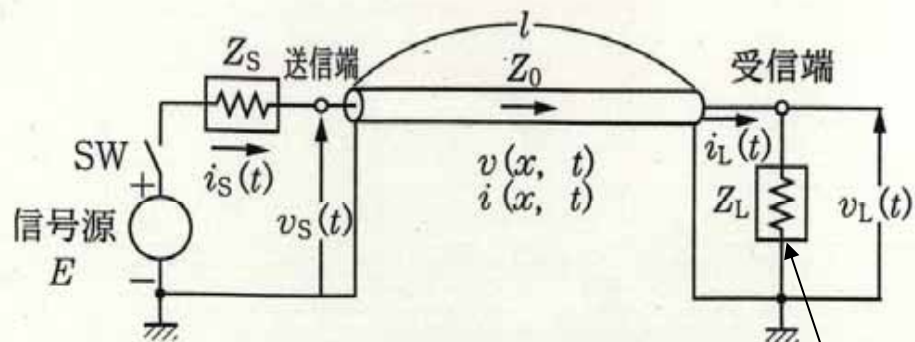


$$v(x, t) = L \frac{\partial i(x, t)}{\partial t} \Delta x + Ri(x, t) \Delta x + v(x + \Delta x, t)$$

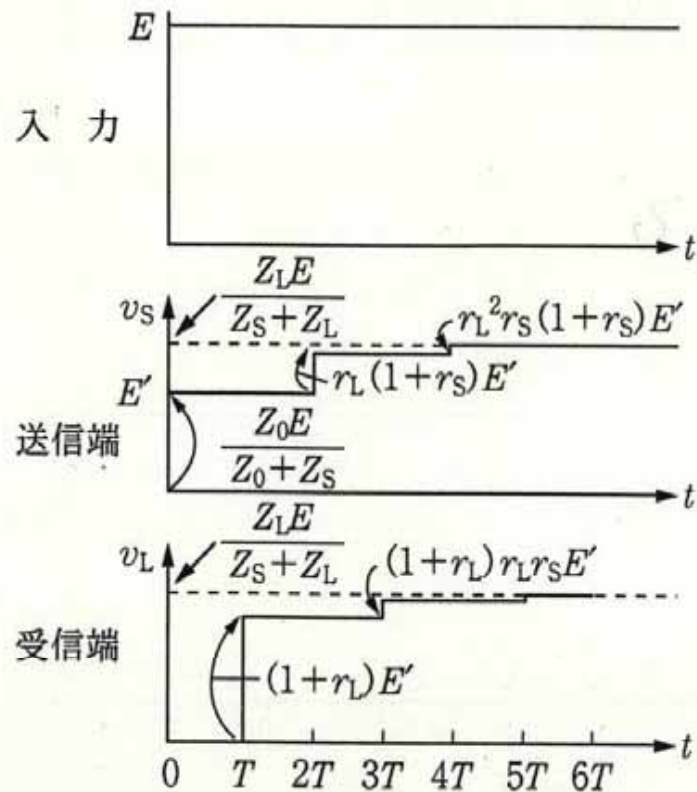
$$i(x, t) = C \frac{\partial v(x, t)}{\partial t} \Delta x + Gv(x, t) \Delta x + i(x + \Delta x, t)$$



(a) 伝送線の等価回路



(b) 伝送線の送受信端



(c) 反射による信号歪み

終端抵抗 (ターミネータ)

図 2.26 伝送線での反射

## 波動方程式

$$u: 1/\sqrt{LC}$$

$$\text{特性インピーダンス } Z_0: \sqrt{L/C}$$

解

$$v(x,t) = f(x-ut) + g(x+ut)$$

$$i(x,t) = (f(x-ut) - g(x+ut)) / Z_0$$

$f, g$  はおのこの正, 負方向への進行波である.

$x, t$  での  $f$  の値と  $f(x+x_0 - u(t+t_0))$  が同一:  $x_0 = ut_0$

$x, t$  での  $g$  の値と  $g(x-x_0 + u(t+t_0))$  が同一:  $x_0 = ut_0$

ステップ状の入力を加えた場合の受信端, 送信端での応答

$$- \quad v/ \quad x=L \quad i/ \quad t$$

$$- \quad i/ \quad x=C \quad v/ \quad t$$

**解**

$$v(x,t)=f(x-ut)+g(x+ut)$$

$$i(x,t)=(f(x-ut)-g(x+ut))/Z_0$$

$$\frac{\partial f(x-ut)}{\partial x} = \frac{\partial f(x-ut)}{\partial(x-ut)} \frac{\partial(x-ut)}{\partial x} = f'(x-ut)$$

$$\frac{\partial f(x-ut)}{\partial t} = \frac{\partial f(x-ut)}{\partial(x-ut)} \frac{\partial(x-ut)}{\partial t} = -uf'(x-ut)$$

## 受信端, 送信端での反射係数

$$r_L = (Z_L - Z_0) / (Z_0 + Z_L),$$

$$r_s = (Z_s - Z_0) / (Z_0 + Z_s)$$

信号Eを印加

送信端: 正方向の進行波のみが

生じ, それを  $f_0(t)$  とすると

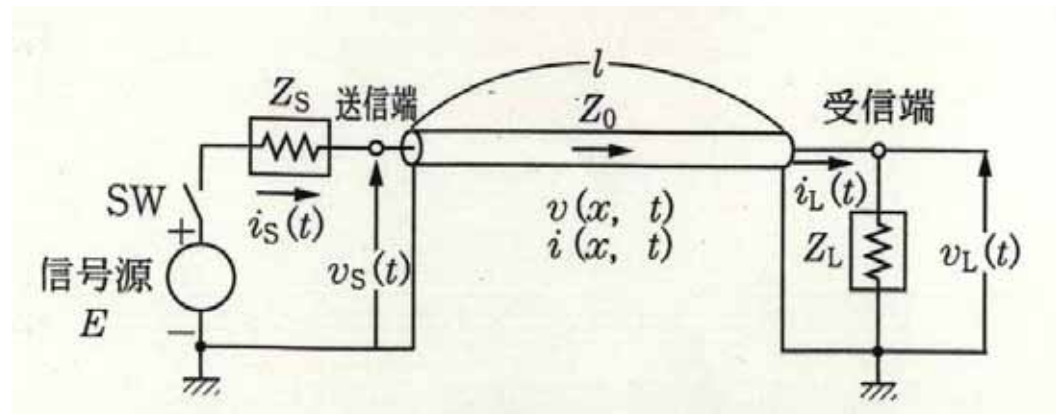
$$v_s(t) = f_0(t)$$

$$i_s(t) = f_0(t) / Z_0 = v_s(t) / Z_0$$

$$E = Z_s i_s + v_s(t)$$

$$v_s(t) = f_0(t) = E \quad (= Z_0 E / (Z_0 + Z_s))$$

$$v(x, t) = f(x - ut) + g(x + ut)$$
$$i(x, t) = (f(x - ut) - g(x + ut)) / Z_0$$



T 時間後受信端  $f_0$

正の進行波  $E$  を受けて負の進行波  $g$  を生じる

$$v_L = E + g_1$$

$$i_L = (E - g_1) / Z_0$$

$$v_L = Z_L i_L$$

$$E + g_1 = Z_L (E - g_1) / Z_0$$

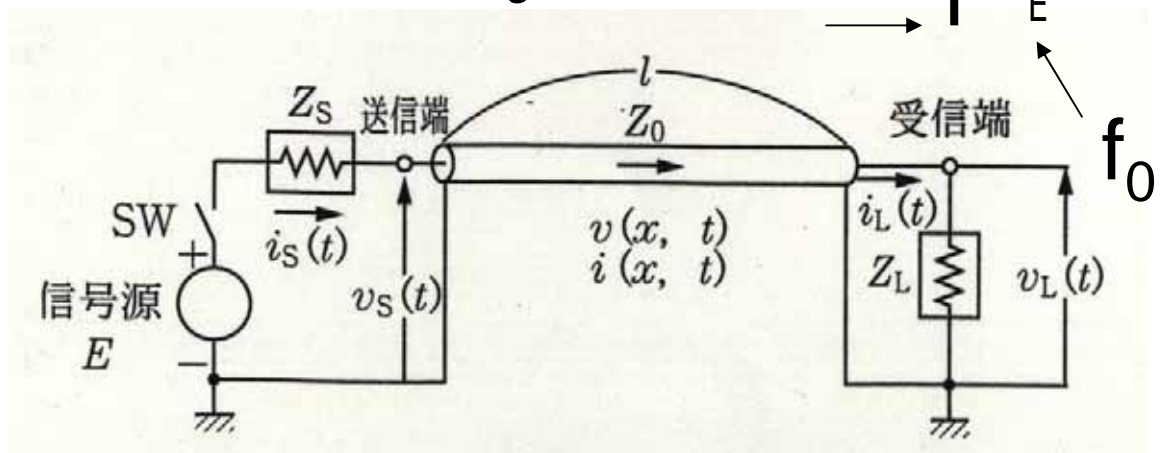
より

$$g_1 = \Gamma_L E$$

$$v_L = (1 + \Gamma_L) E$$

$$g_1 = \Gamma_L E'$$

$$v_L = (1 + \Gamma_L) E$$



$$\Gamma_L = (Z_L - Z_0) / (Z_0 + Z_L),$$

$$\Gamma_s = (Z_s - Z_0) / (Z_0 + Z_s)$$

2T時間後に送信端で

$\Gamma_L E'$ だけ反射

正方向進行波 $f_1$

送信端に  
だけ加算

$$v_s = f_0 + f_1 + \Gamma_L E'$$

$$i_s = (f_0 + f_1 - \Gamma_L E') / Z_0$$

$$v_s = -i_s Z_s + E$$

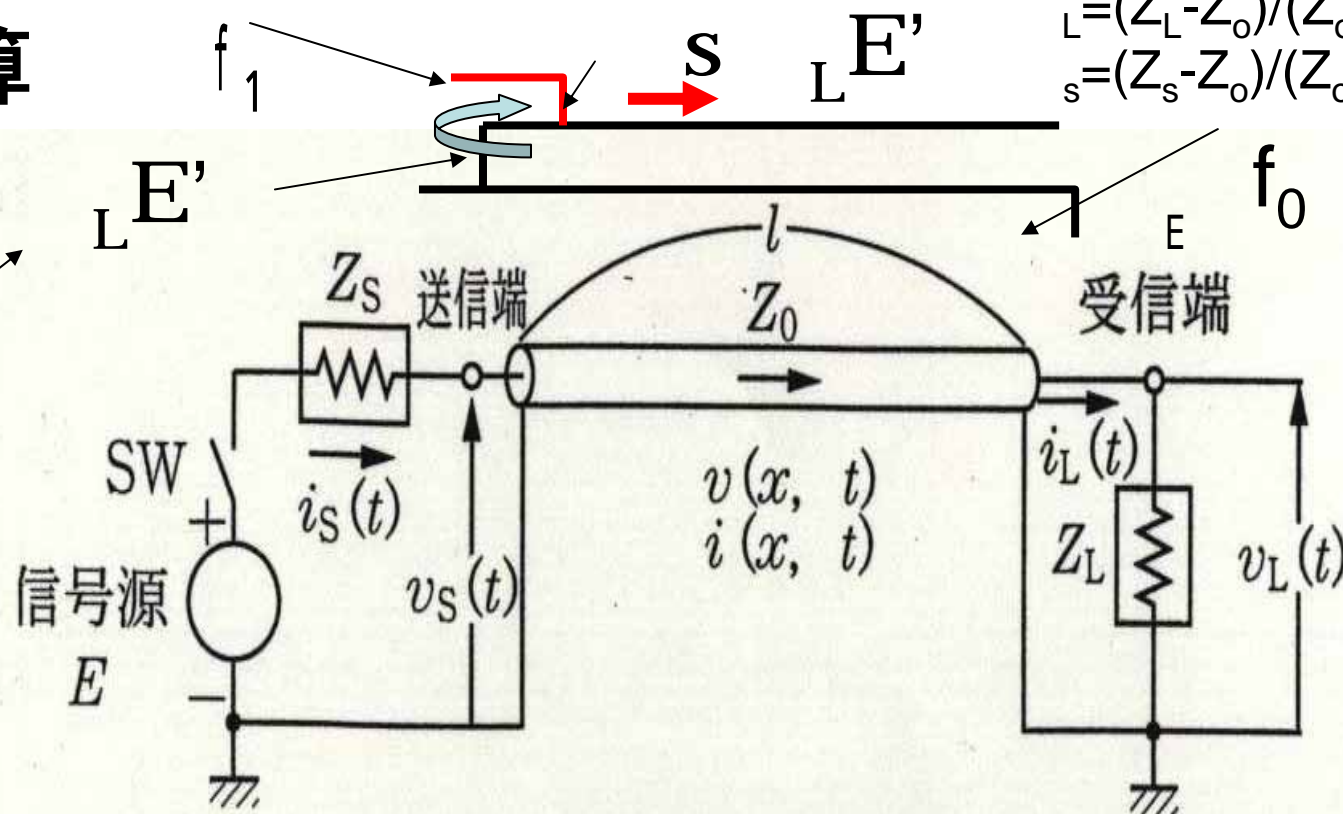
より

$$f_1 = \Gamma_s \Gamma_L E'$$

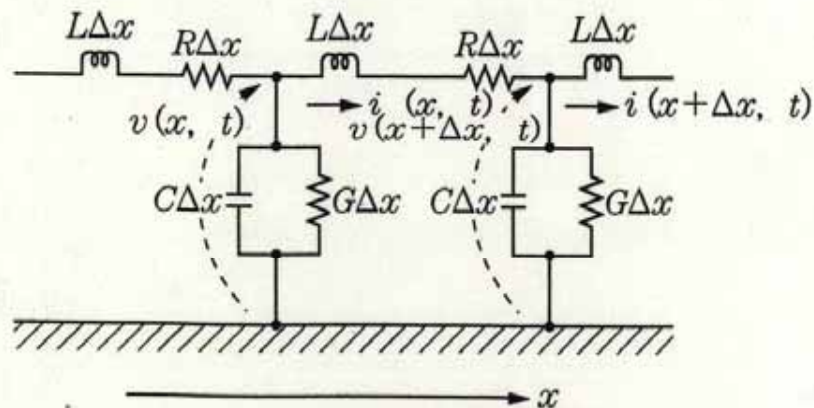
$$\Gamma_L = (Z_L - Z_0) / (Z_0 + Z_L),$$

$$\Gamma_s = (Z_s - Z_0) / (Z_0 + Z_s)$$

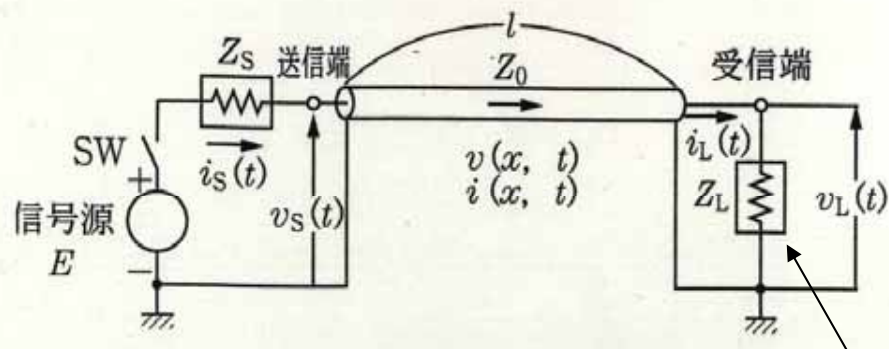
$\Gamma_1$



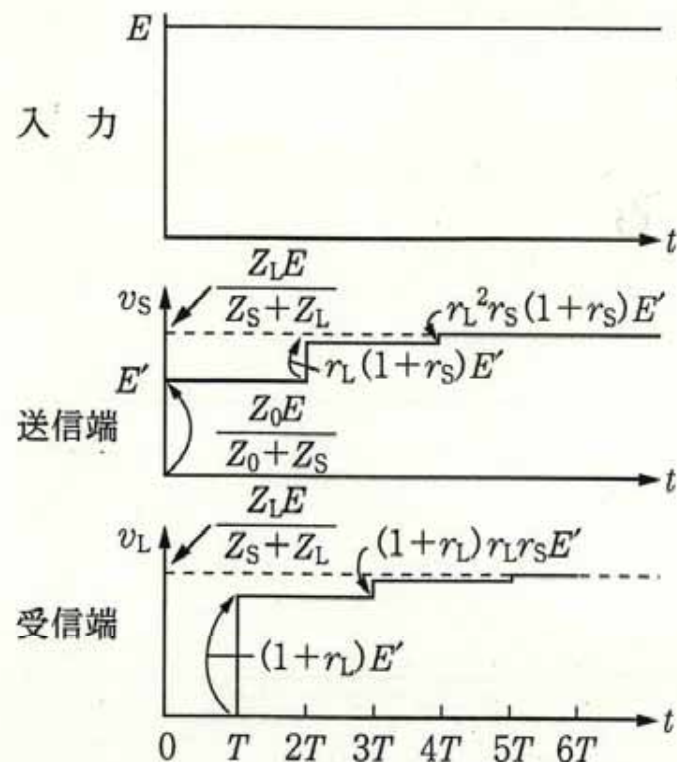




(a) 伝送線の等価回路



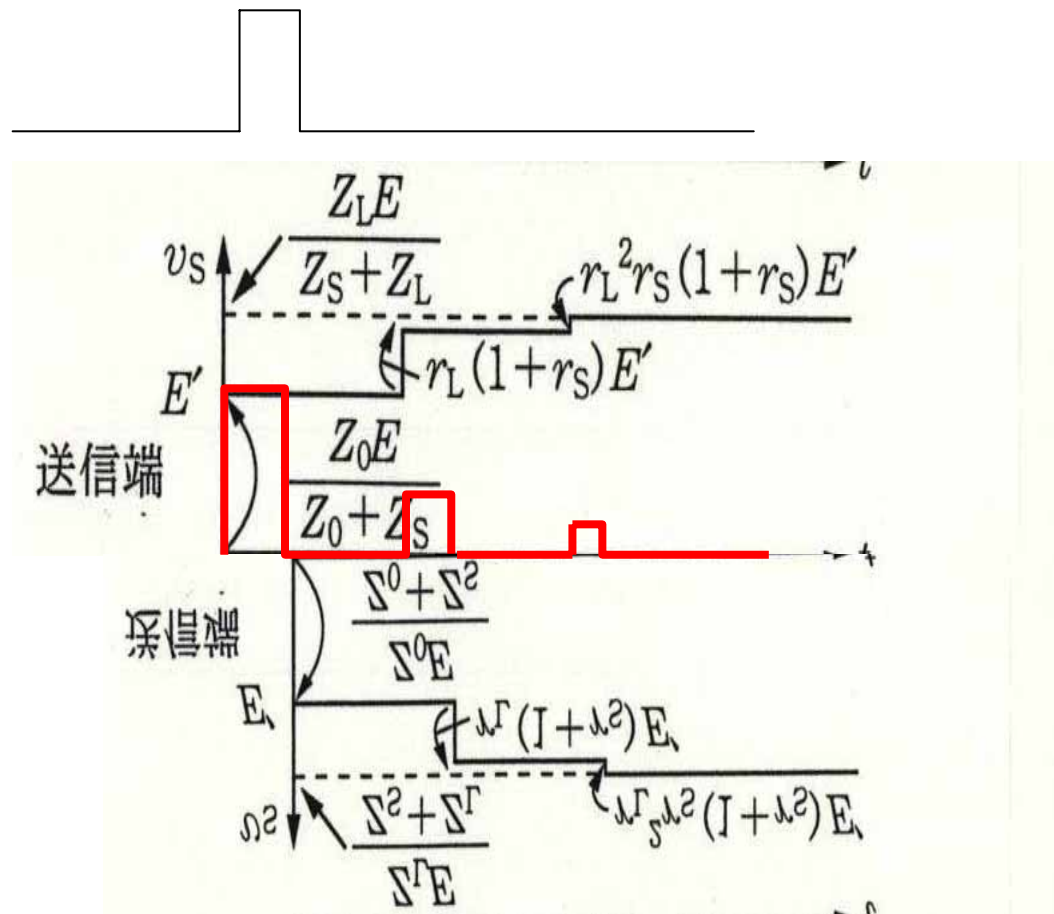
(b) 伝送線の送受信端



(c) 反射による信号歪み

**終端抵抗(ターミネータ)が必要！**

図 2.26 伝送線での反射



往復の伝播時間より幅の短いパルスを入力すると？

- 伝送線路2 : RC線路 (VLSI内)

$$-\frac{\partial v}{\partial x} = Ri$$

$$-\frac{\partial i}{\partial x} = C \frac{\partial v}{\partial t}$$

より

$$\frac{\partial^2 v}{\partial x^2} = RC \frac{\partial v}{\partial t}$$

これは拡散方程式

ラプラス変換して

$$\frac{d^2V}{dx^2} = RCsV$$

これを解いて

$$V = Ae^{-\sqrt{RCs}x} + Be^{\sqrt{RCs}x}$$

$$V(0) = \frac{E}{s}, B = 0 \quad (\times \text{ で有限値})$$

より

$$V = \frac{E}{s} e^{-\sqrt{RCs}x}$$

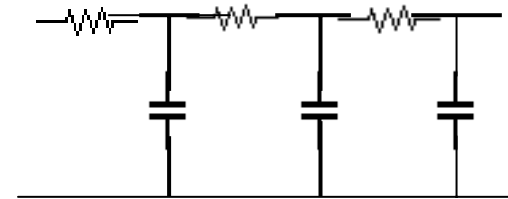
これを解くと

$$v(x, t) = E(t) \operatorname{erfc}\left(\frac{x}{2} \sqrt{\frac{RC}{t}}\right)$$

$$\operatorname{erfc}(y) = 1 - \operatorname{erf}(y) = \frac{2}{\sqrt{\pi}} \int_y^{\infty} e^{-\eta^2} d\eta$$

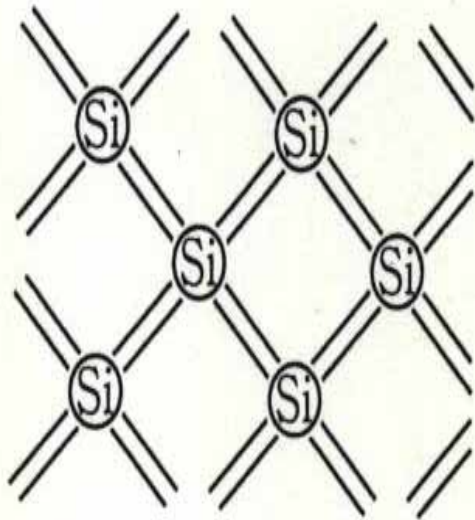
erf: エラー関数

$V(x, t) = \text{閾値 } Th$  となる時間は  
 $x^2$  と  $RC$  に比例する。



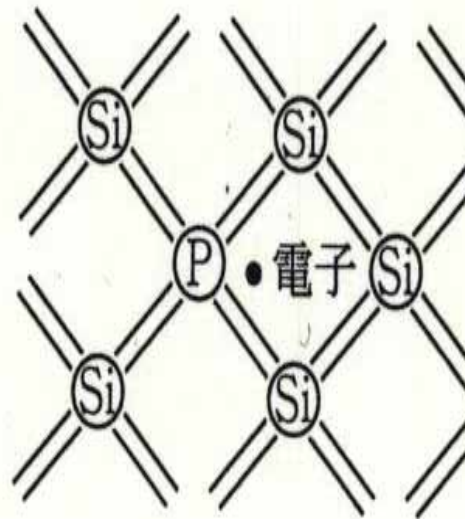
集中定数回路で近似

## 集積回路の基礎



Si : シリコン原子  
= : 共有結合

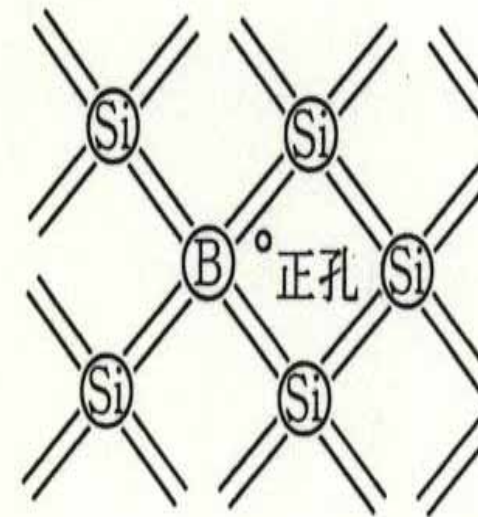
(a) 真正半導体



P : リン原子

第 属元素

(b) n型半導体



B : ほう素原子

第 属元素

(c) p型半導体

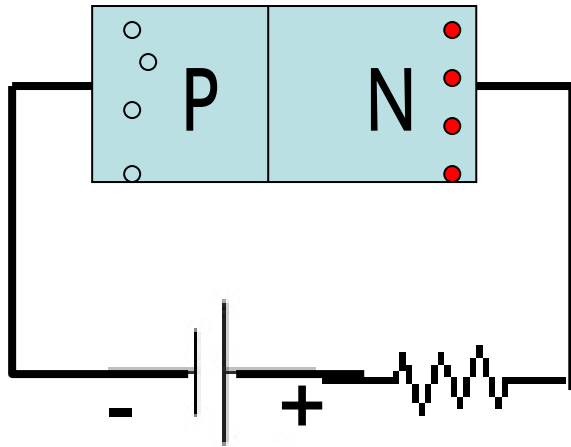
第 属元素

価電子 : 4

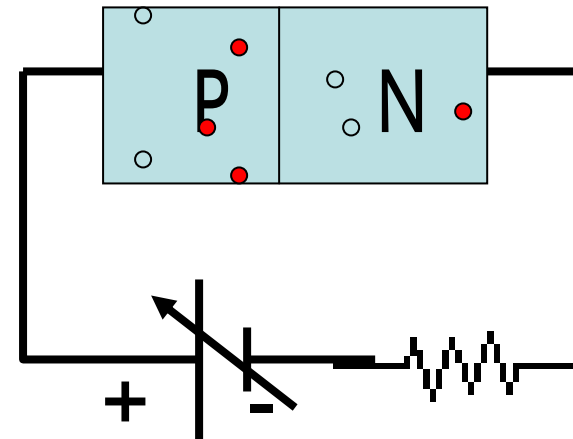
図 2.27 n型, p型半導体

# P N 接合

電流流れず



電流流れる



● 電子  
○ 正孔



# バイポーラトランジスタ: 電子、正孔いずれも関与

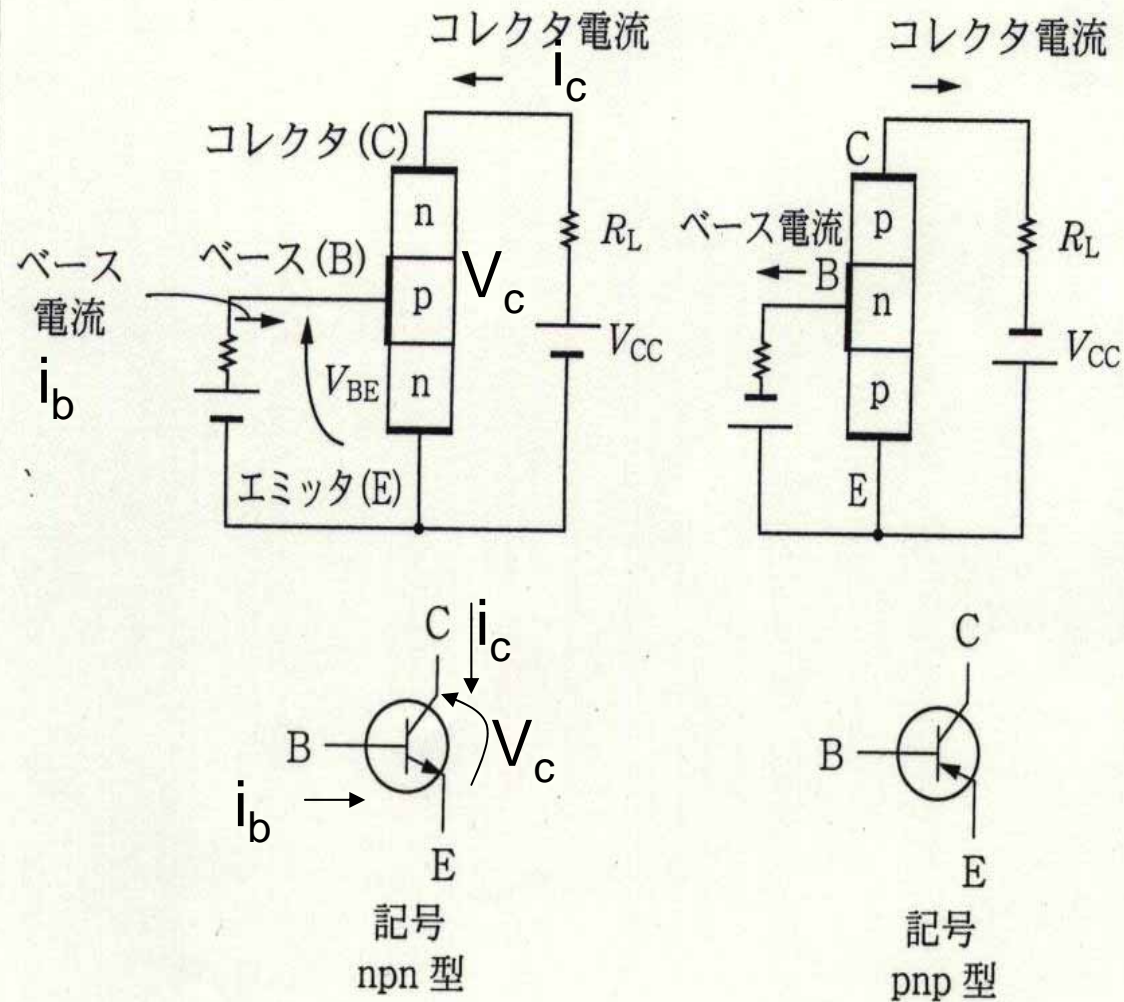
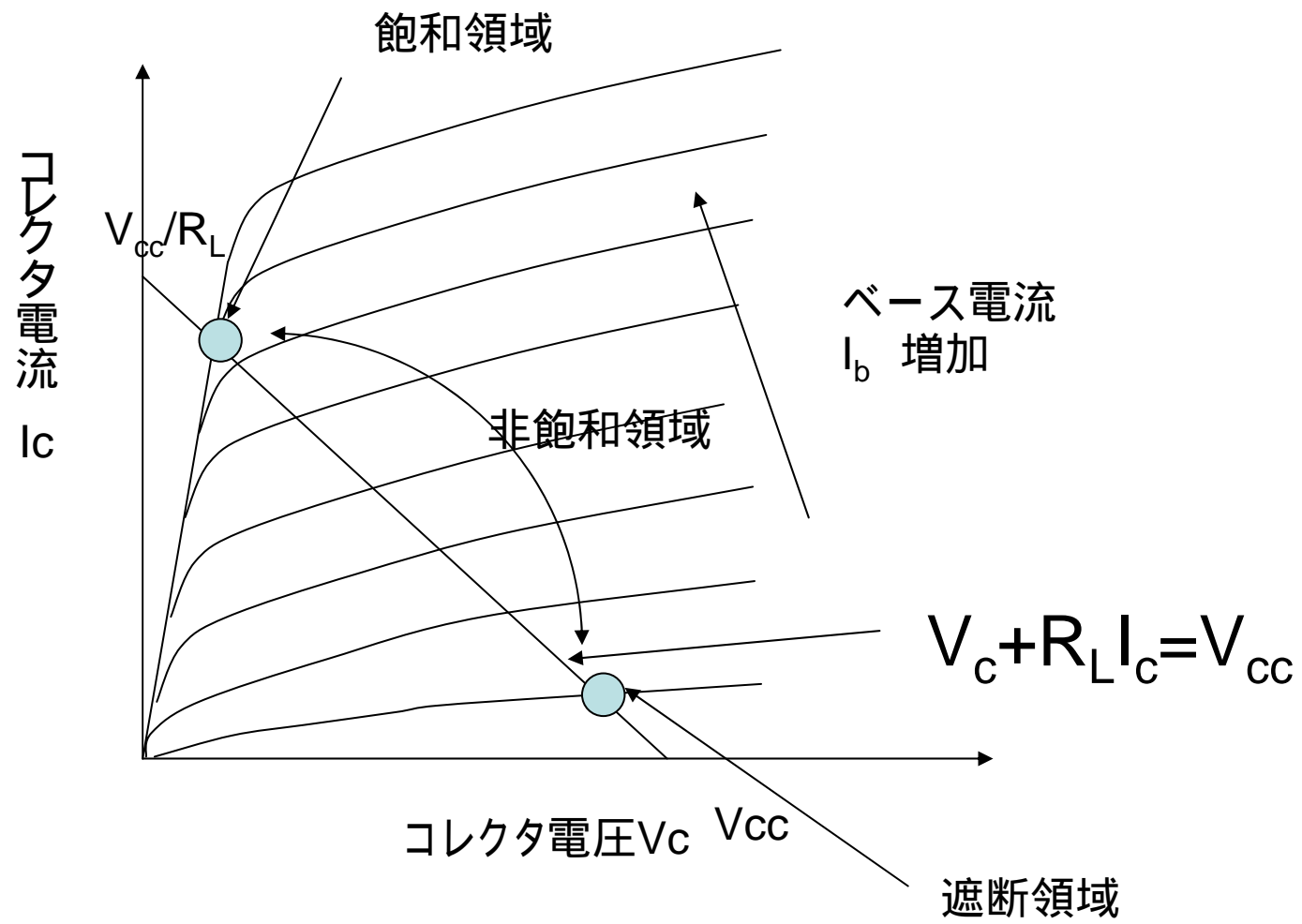
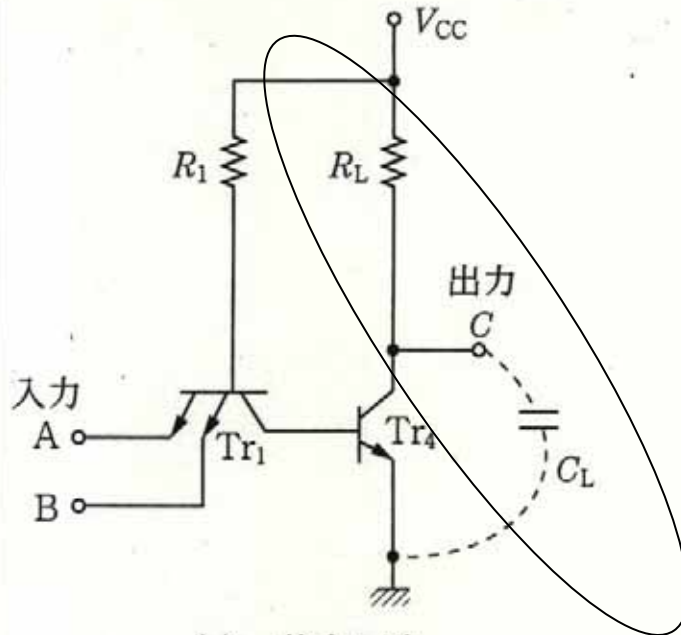


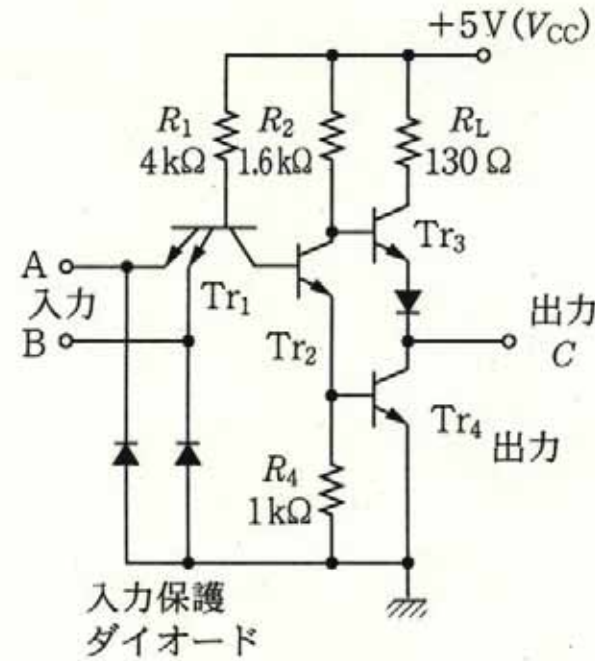
図 2.28 バイポーラトランジスタ



# TTL



(a) 基本回路

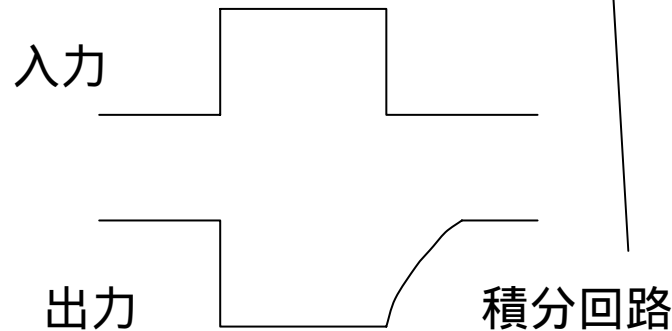


(b) トーテムポール方式

図 2.29 TTL 回路

飽和領域と

遮断領域を使用



	TR1	TR2	TR3	TR4
入力 L	ON	OFF	ON	OFF
入力 H	OFF	ON	OFF	ON

Transistor Transistor Logic

# ECL

Emitter Coupled Logic

非飽和領域を使用

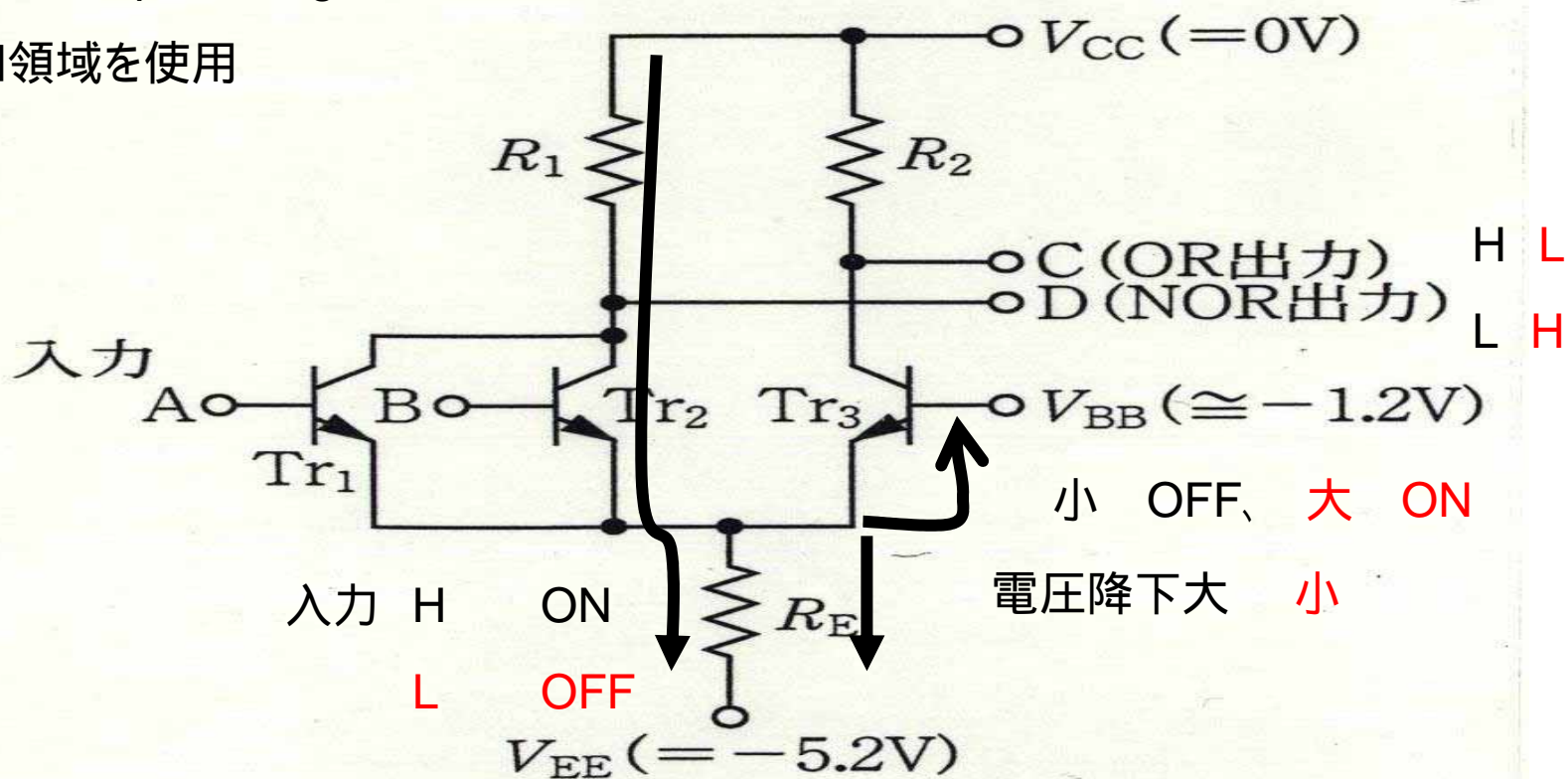
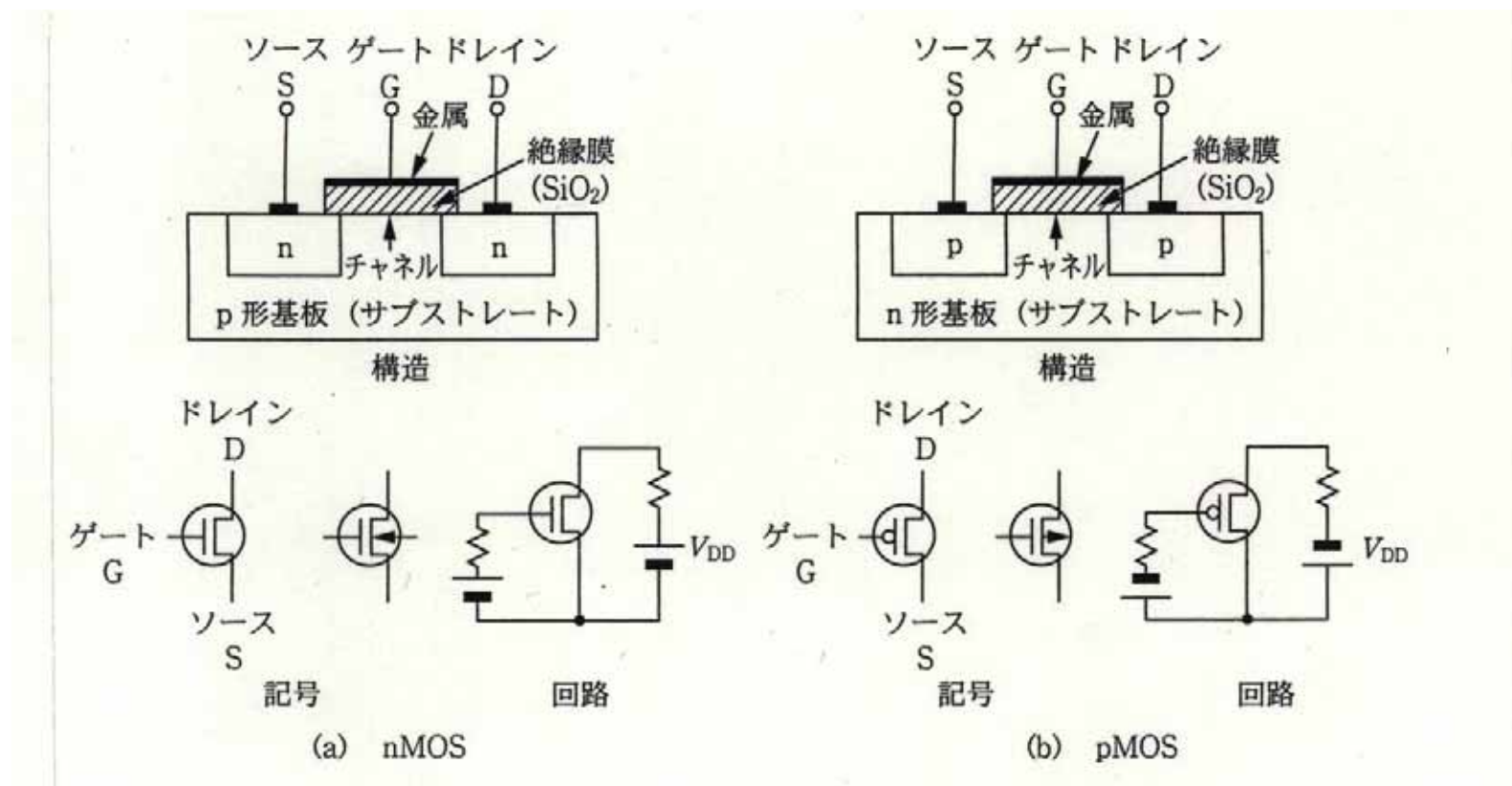


図 2.30 ECL の基本回路

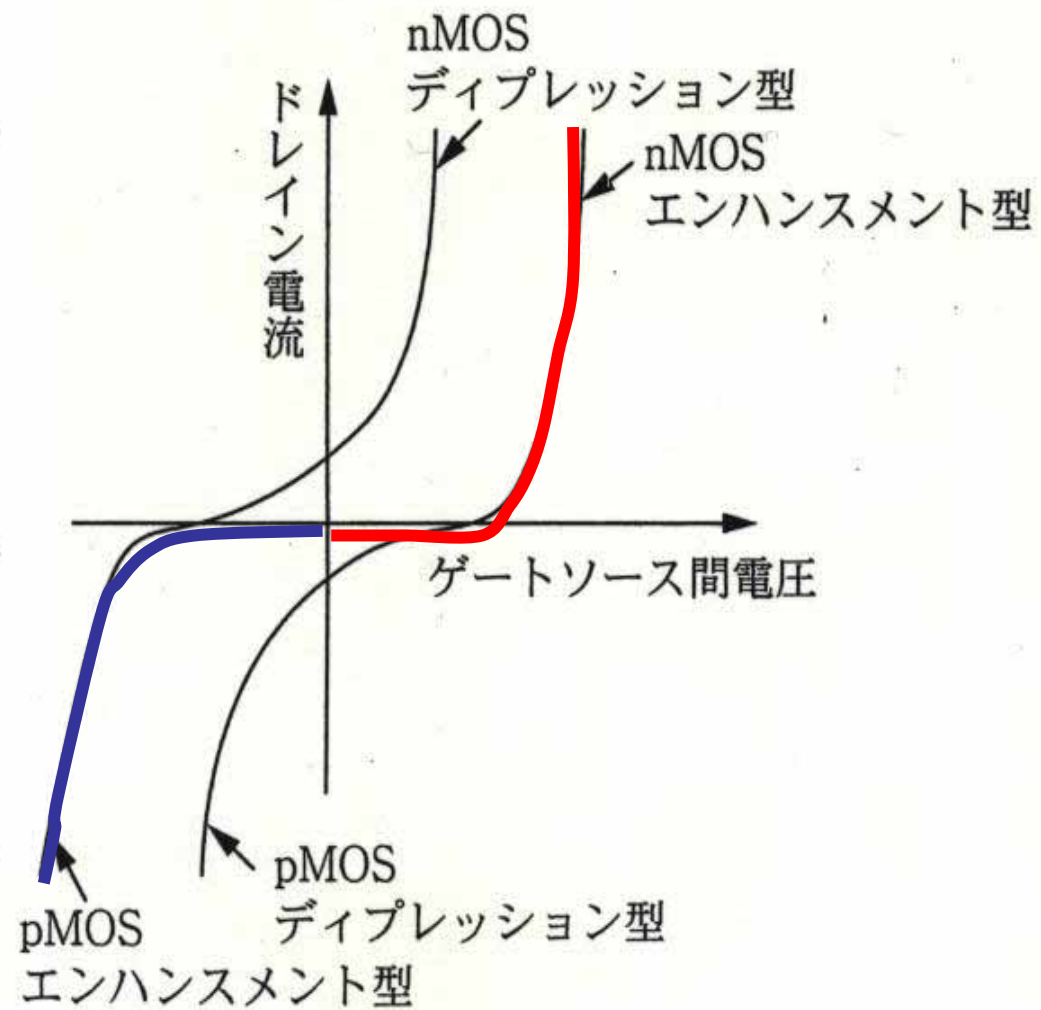
# MOS



電界効果を利用 (FET)

電子、正孔いずれか一方が関与

Metal Oxide Semiconductor



(c) 入出力特性

図 2.31 MOS 回路



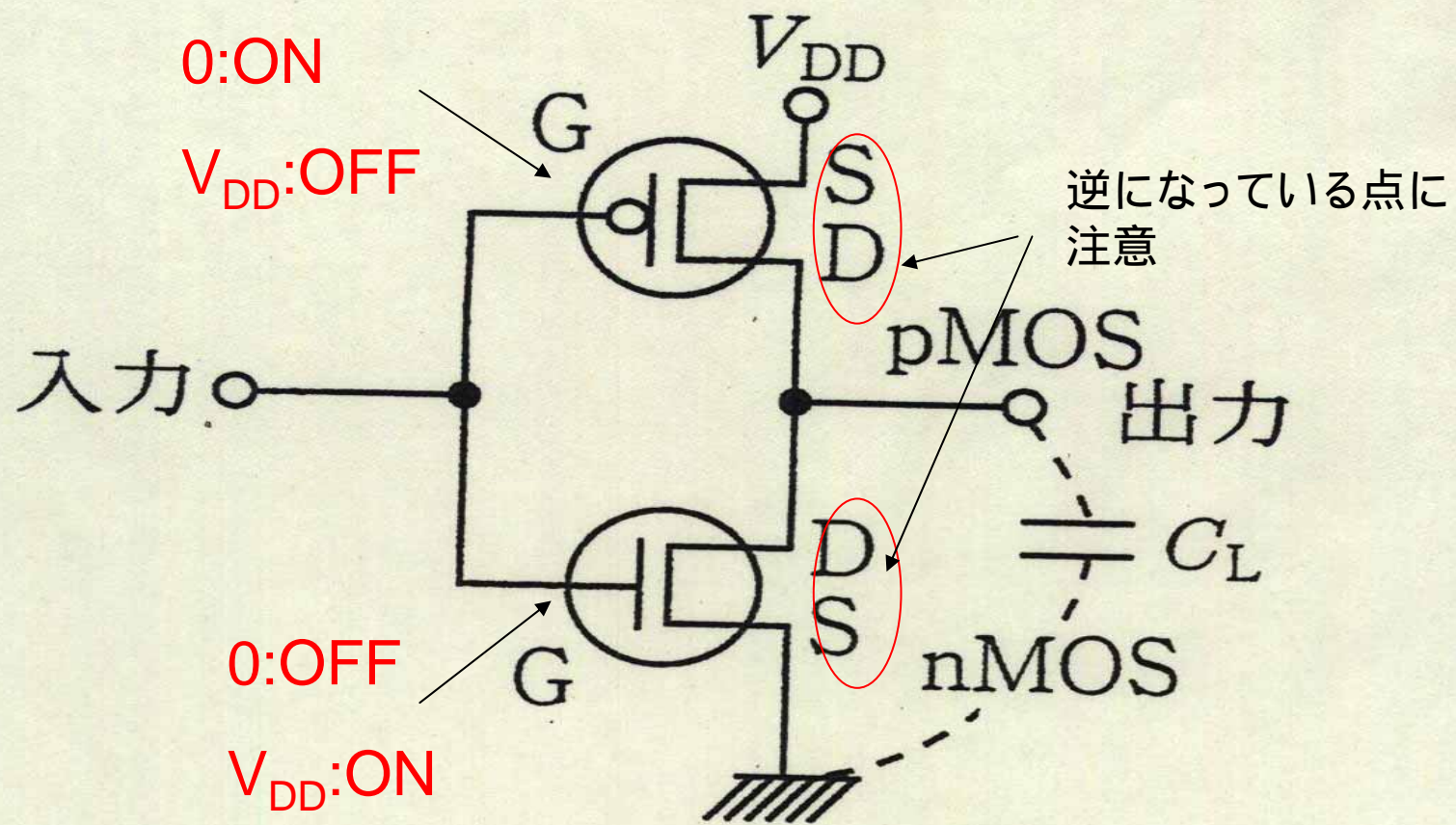
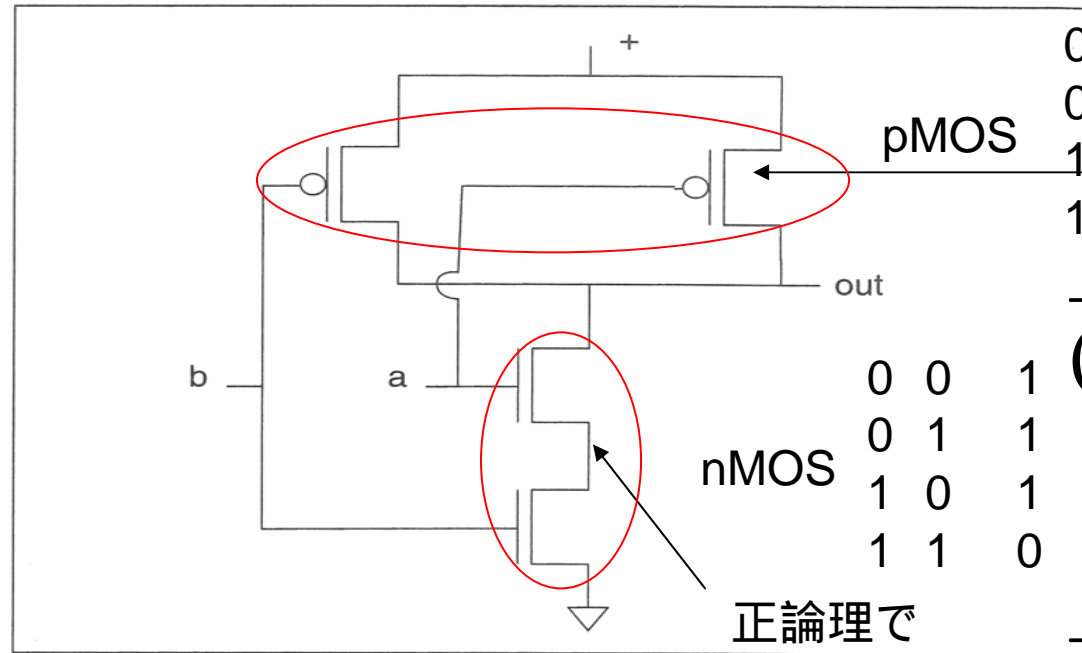


図 2.32 CMOS 否定回路



NAND



0	0	1
0	1	1
1	0	1
1	1	0

負論理  
でNOR

0	0	1
0	1	1
1	0	1
1	1	0

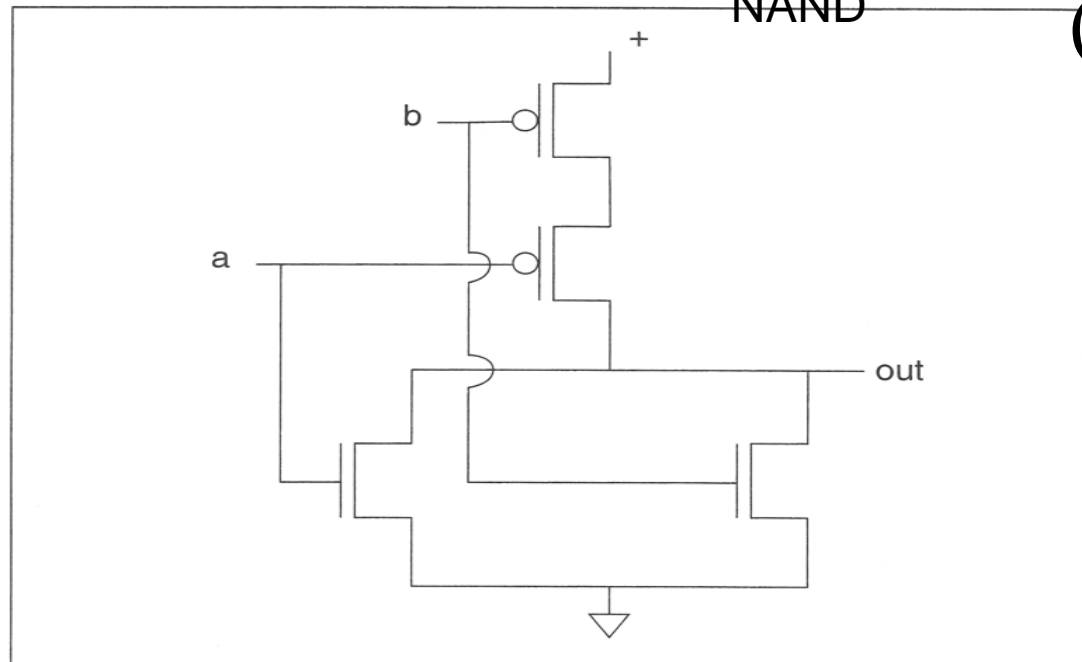
正論理で  
NAND

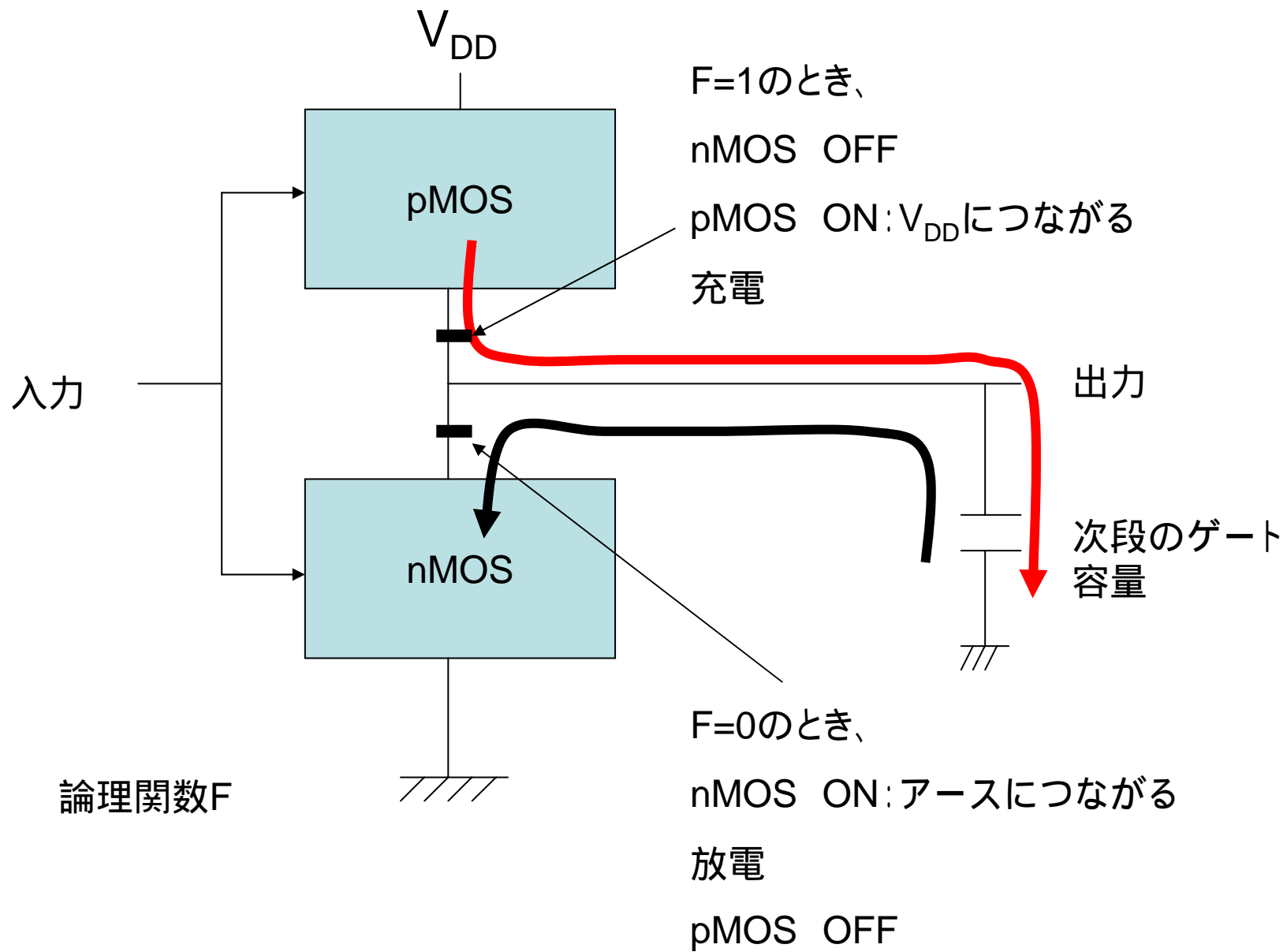
$$(a \cdot b) = \overline{\overline{a + b}}$$

$$= \overline{\overline{a + b}}$$

$$(a \cdot b)$$

NOR





## 正論理と負論理

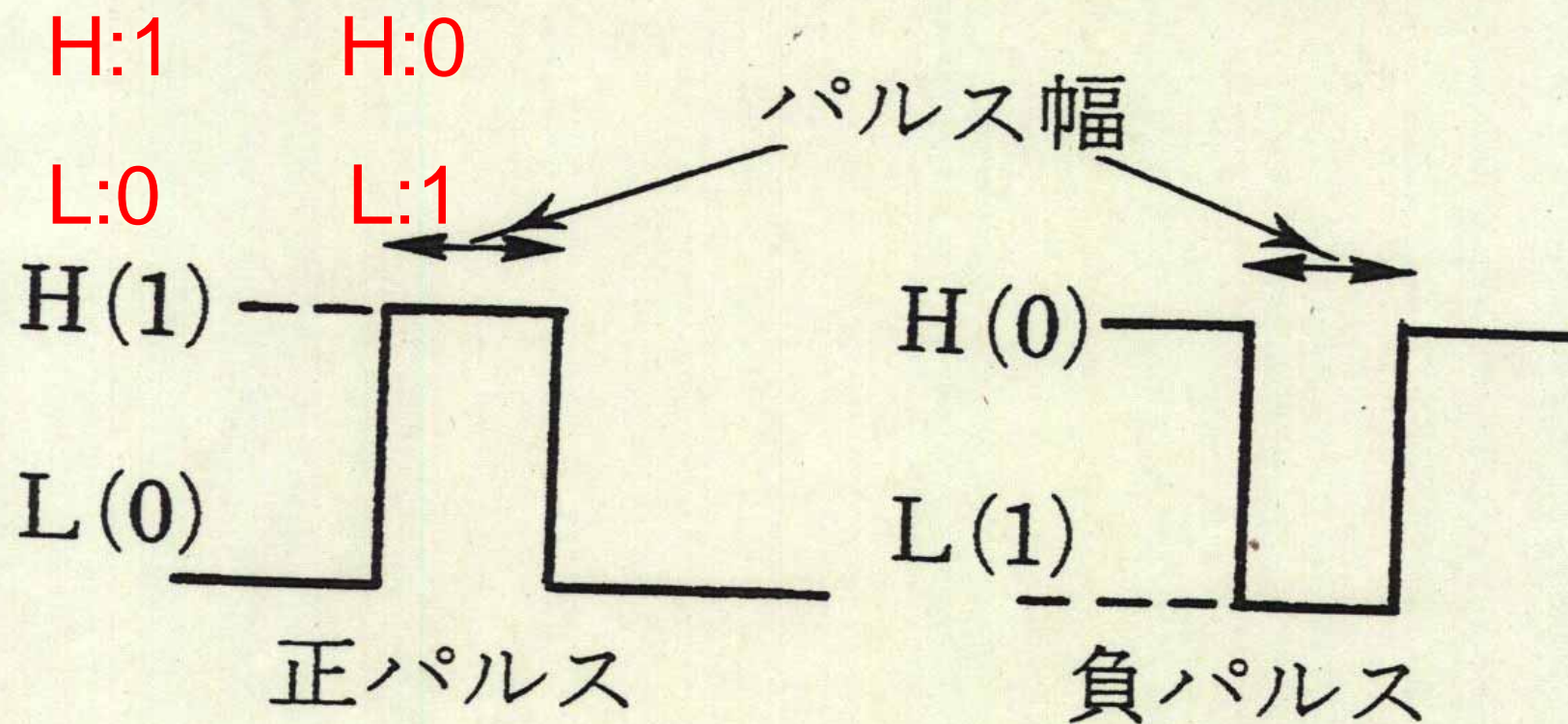
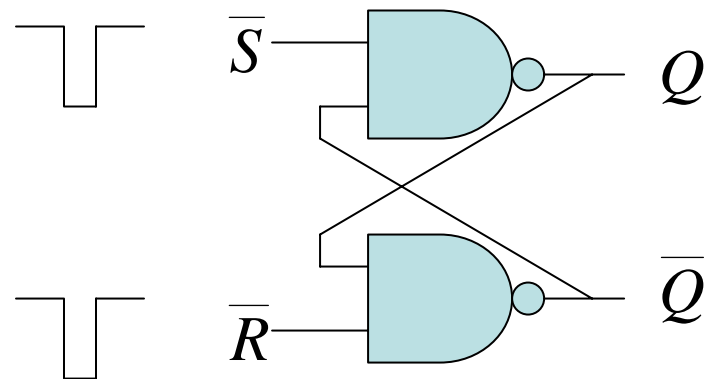


図 2.33 正パルスと負パルス

			正論理			負論理		
入力		出力						
H	H	H	1	1	1	0	0	0
H	L	L	1	0	0	0	1	1
L	H	L	0	1	0	1	0	1
L	L	L	0	0	0	1	1	1
			AND			OR		

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



# 省電力プロセッサ

## CMOSの電力消費

- ・動的

回路がON、OFFするとき  $f C_L V^2$

- ・漏れ電流  $V I_{\text{leakage}}$

- ・貫通電流  $f t_{\text{sc}} I_{\text{peak}} V$

pMOS、nMOSがスイッチング時  
同時ON

: ゲート動作率、 $f$ : 周波数、 $C$ : ゲート総容量、  
 $V$ : 電源電圧、 $t_{\text{st}}$ : スイッチング時間

# 基本式

$$P = f C_L V^2 + V I_{\text{leakage}} + f t_{\text{sc}} I_{\text{peak}} V$$

$$F_{\text{max}} = \frac{(V - V_{\text{threshold}})^2}{V}$$

$$I_{\text{leakage}} = \exp(-qV_{\text{threshold}}/kT)$$

## 各部での電力消費の割合 (%)

	クロック	データパス	メモリ	I/O
組み込み系	50	33	11	6
高速プロセッサ	30	25	40	5
MPEGASIC	22	34	22	22
ATMSWASIC	20	10	6	64

# 基本的な考え方

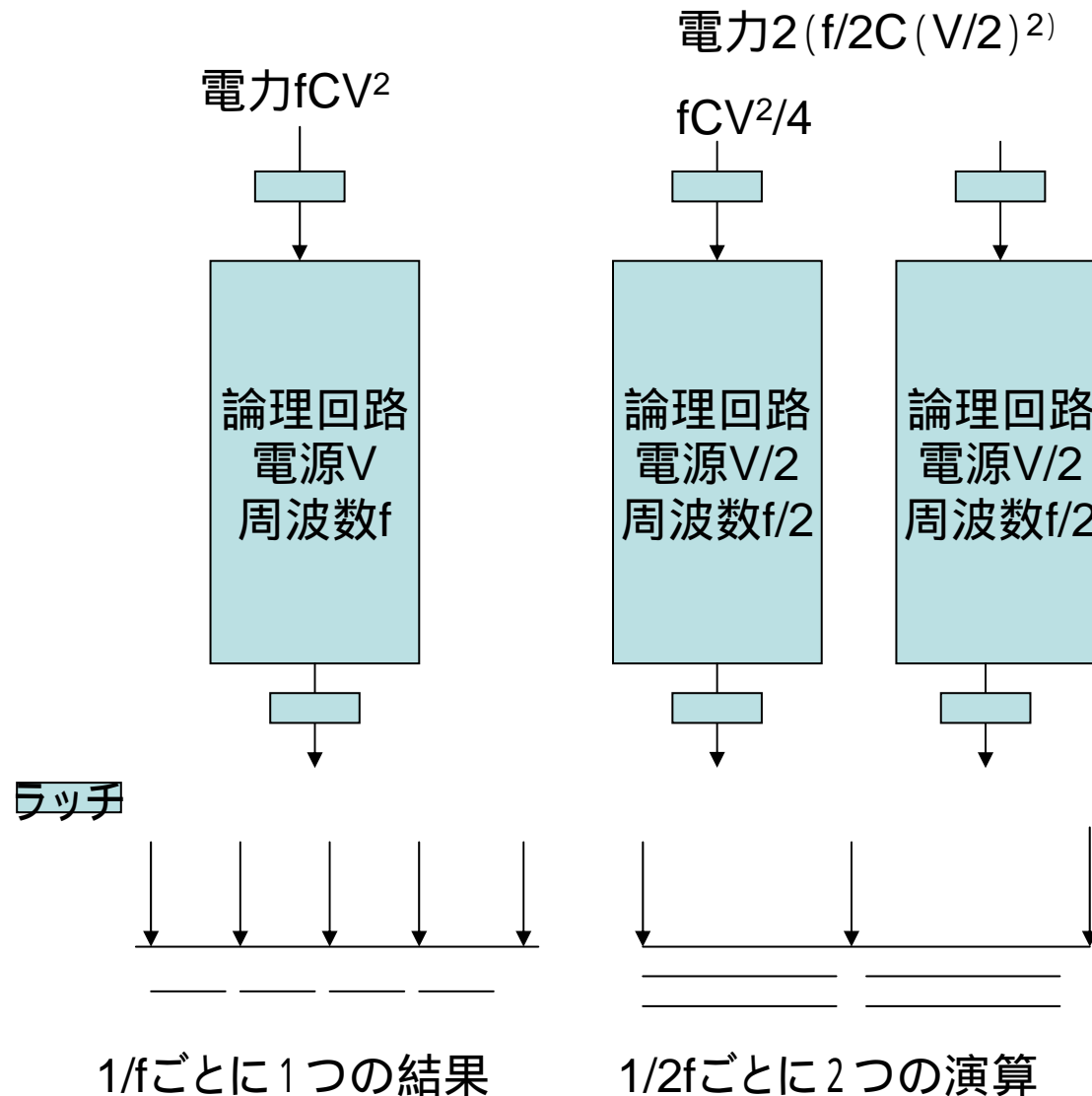
- スイッチング回数を少なくする
- 動作をしない(と予想される)回路には  
    クロック供給しない  
    電源を供給しない
- 電源電圧を制御して、必要十分な処理速度で実行
- 電源電圧、周波数を落として並列処理、パイプラインで行う



- デバイスレベル  
低電源電圧化、低ゲート容量化、低スレッショルド化
- 回路レベル  
パストランジスタ論理  
ゲート付きクロック  
グリッチの削減  
動的電源遮断  
非同期回路
- アーキテクチャレベル  
データパスの最適化: 必要な演算幅の決定など  
並列処理、パイプライン処理  
キャッシュメモリ  
バス  
    アドレスの反射2進符号化、データ圧縮
- OS、コンパイラ、アルゴリズムレベル  
符号化  
ビット変化の少ないコード生成  
動的電源電圧制御  
動的周波数制御

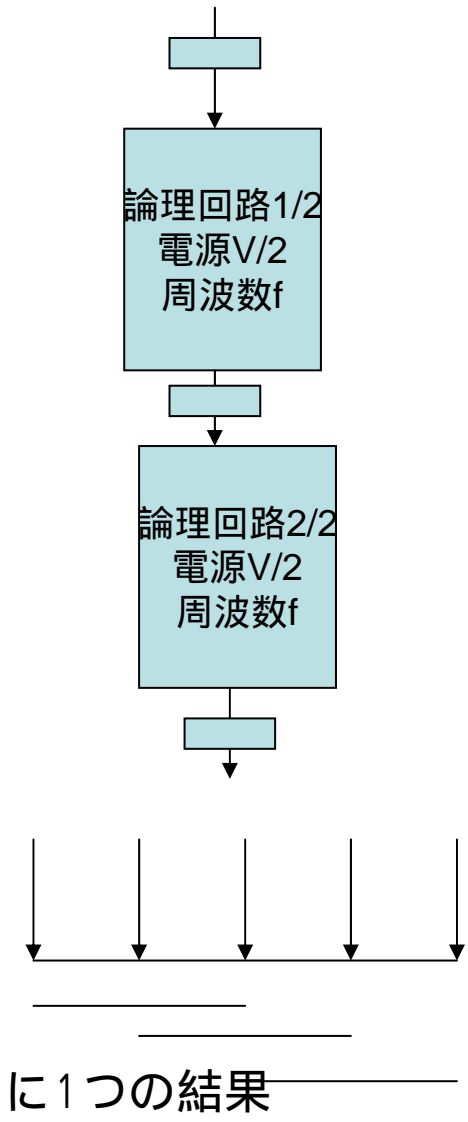
# 1 並列処理の導入

## 並列処理



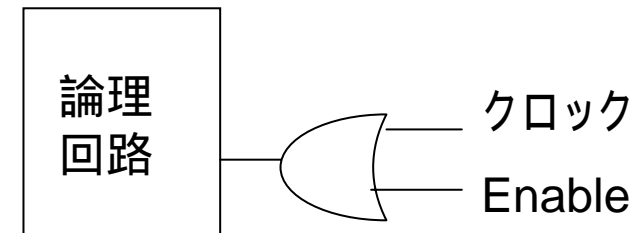
## パイプライン処理

電力  $fCV^2/2$

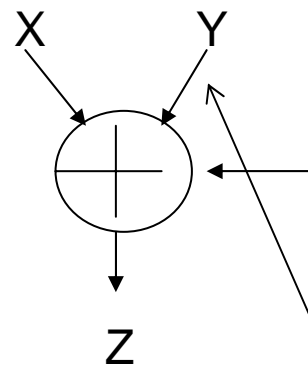


## 2 無駄なスイッチングを減らす

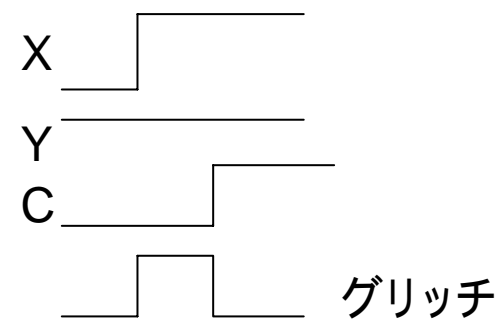
ゲート付き  
クロック



グリッチの  
削除



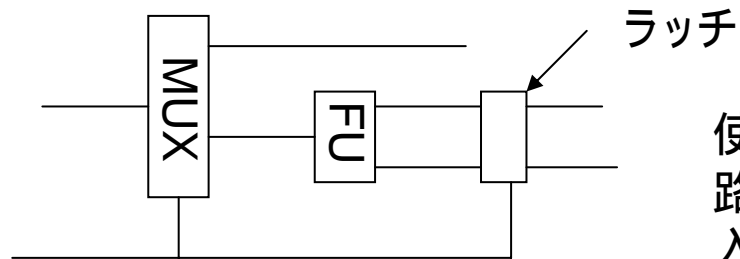
C遅れ



ディレイ挿入    パイプライン化

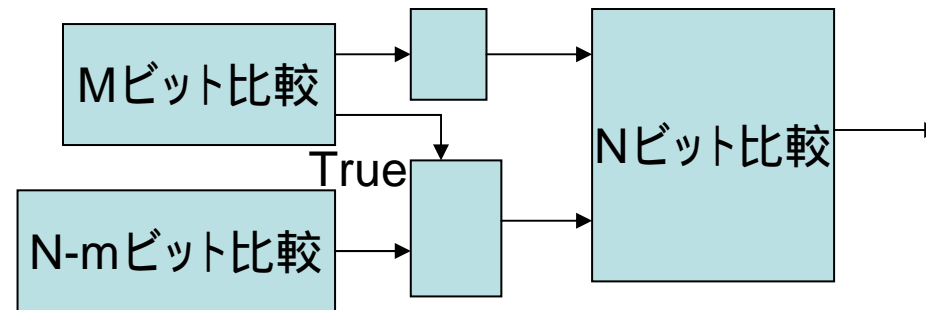
ガード付き計算

選択

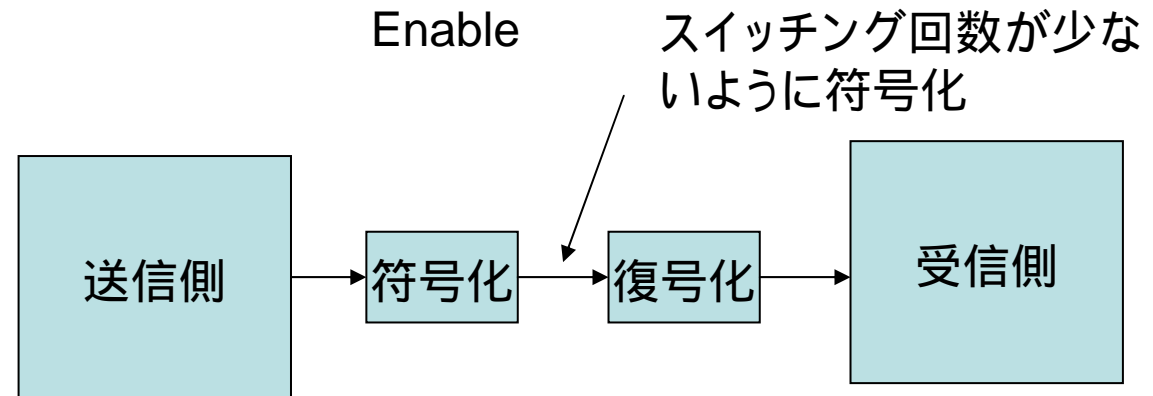


使わない回  
路には同じ  
入力

プレ計算



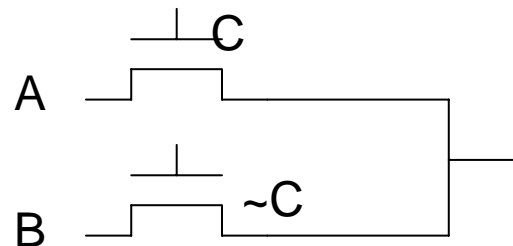
符号化



データの転送順、  
処理順の変更

正の数のグループ先、負の数のグループ後

パストラ  
ンジスタ



$$C \cdot A + \sim C \cdot B$$

トランジスタ数の削減

### 3 スイッチングの少ない符号化

2の補数表示: 符号反転操作大変

正+5      負-5

0 0 1 0 1      1 1 0 1 1

絶対値表示: 楽

0 0 1 0 1      1 0 1 0 1

カウンタ: 反射2進符号

2進符号   000 001 010 011 100 101 110 111

反射2進符号   000 001 011 010 110 111 101 100

1ビットずつ変化

- 2進符号と反射2進符号の相互変換

$$(b_n, b_{n-1}, b_{n-2}, \dots, b_1) \quad (g_n, g_{n-1}, g_{n-2}, \dots, g_1)$$

$$b_n=1, \quad g_n=1$$

$$b_n=0, \quad g_n=0$$

$$b_n=0, b_{n-1}=0 \text{ のとき } g_{n-1}=0$$

$$b_n=0, b_{n-1}=1 \text{ のとき } g_{n-1}=1$$

$$b_n=1, b_{n-1}=0 \text{ のとき } g_{n-1}=1$$

$$b_n=1, b_{n-1}=1 \text{ のとき } g_{n-1}=0$$

一般に

$$b_i \oplus b_{i-1} = 0 \text{ のとき } g_{i-1} = 0$$

$$b_i \oplus b_{i-1} = 1 \text{ のとき } g_{i-1} = 1$$

$$b_i \oplus b_{i-1} = g_{i-1}$$

## 4 メモリ・キャッシュメモリの省電力化

フィルタキャッシュ:L1キャッシュ前にバッファ設置

メモリのバンク化

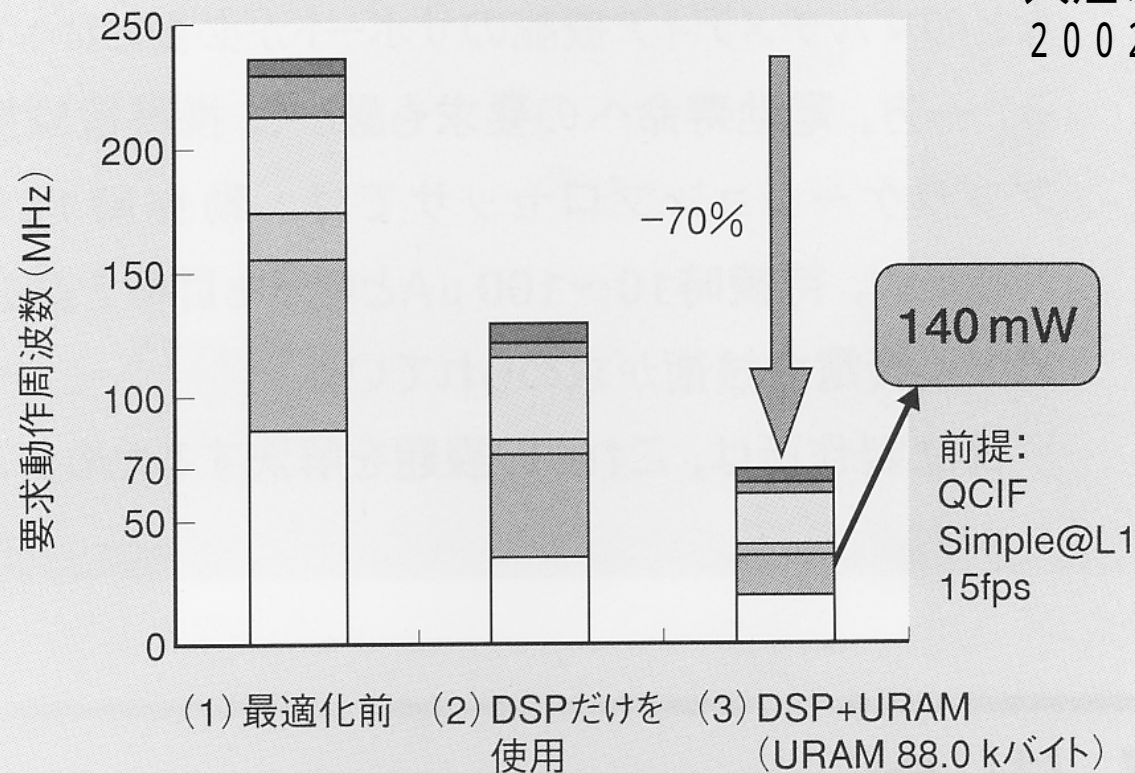
目的別領域化

リプレースなし領域:画像処理など

選択的ウェイクアクセス、投機的



入江:日立評論、  
2002年10月



注1: ■ MC(動き補償), ■ IDCT(逆離散コサイン変換), □ Q(量子化)/IQ(逆量子化)  
+VLC(可変長符号化), ■ DCT(離散コサイン変換), ■ Ctl(制御), □ ME(動き  
予測)

注2: 略語説明 QCIF(176×144画素のビデオ信号フォーマット), fps(Frames per  
Second), Simple@L1(低解像度で単純な動画像)

### 図1 MPEG-4符号化処理における要求動作周波数の低減

DSPとURAMを併用することで、一般的なCPUに比べて70%の動作周波数を低減することができる。

## 5 電源電圧制御

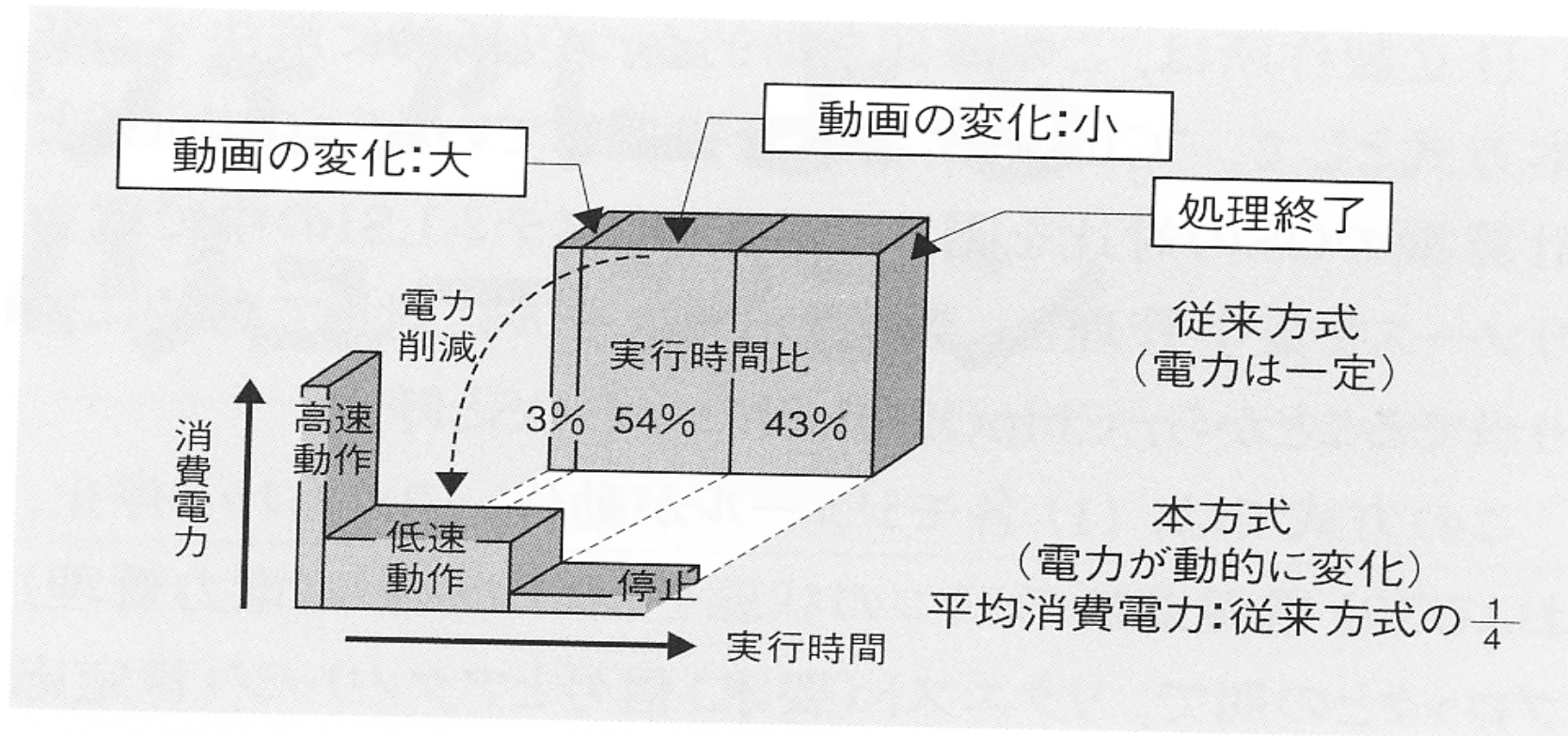


図2 MPEG-4動画デコーダでの消費電力削減

大半の処理は、低速・低電圧か停止（スリープ）である。このため、高速・高電圧を常に保持する処理の  $\frac{1}{4}$  の電力に低減することができる。

# 各メーカーでの名称

- Intel: SpeedStep
- Transmeta: Longrun
- AMD: PowerNow
- VIA: LongHaul