

コンピュータアーキテクチャ（1）

坂井 修一

東京大学大学院 情報理工学系研究科 電子情報学専攻
東京大学 工学部 電子情報工学科／電気電子工学科

- ・ はじめに
- ・ スケジュール
- ・ コンピュータアーキテクチャ入門

はじめに

- 本講義の目的
 - コンピュータアーキテクチャの基本を学ぶ
- 時間・場所
 - 火曜日 8:30 - 10:15 工学部2号館241
- ホームページ（ダウンロード可能）
 - url: <http://www.mtl.t.u-tokyo.ac.jp/~sakai/hard/>
- 教科書
 - 坂井修一『コンピュータアーキテクチャ』（コロナ社、電子情報レクチャーシリーズC-9）
 - 坂井修一『実践 コンピュータアーキテクチャ』（コロナ社）
- 参考書
 - D. Patterson and J. Hennessy, Computer Organization & Design、5th Ed.（邦訳『コンピュータの構成と設計』（第5版）上下（日系B P））
 - 馬場敬信『コンピュータアーキテクチャ』（改訂2版）、オーム社
 - 富田眞治『コンピュータアーキテクチャ I』、丸善
- 予備知識： 論理回路
 - 坂井修一『論理回路入門』、培風館
- 成績
 - 試験＋レポート＋出席

教科書



特典

- 理学部情報科学科 中田登志之教授とのコラボ
 - 同じ教科書
 - スライド共有
 - 高度／詳細な内容の追加
 - 細かい修正

©Toshiyuki Nakata とスライドに表示

注意

■ 成績評価基準(概略)

- 3年生(例): 実態をみての変更あり
 - ・ 出席(小テスト): 15~25%
 - ・ レポート: 35~45%
 - ・ 試験: 30~50% (試験欠席:未履修)
- 4年生以上
 - ・ レポート: 45~55%
 - ・ 試験: 45~55% (試験欠席:未履修)

講義の概要と予定（1／2）

1. コンピュータアーキテクチャ入門

ディジタルな表現、負の数、実数、加算器、ALU、フリップフロップ、レジスタ、計算のサイクル

2. データの流れと制御の流れ

主記憶装置、メモリの構成と分類、レジスタファイル、命令、命令実行の仕組み、実行サイクル、算術論理演算命令、シーケンサ、条件分岐命令

3. 命令セットアーキテクチャ

操作とオペランド、命令の表現形式、アセンブリ言語、命令セット、算術論理演算命令、データ移動命令、分岐命令、アドレッシング、サブルーチン、RISCとCISC

4. パイプライン処理（1）

パイプラインの原理、命令パイプライン、オーバヘッド、構造ハザード、データハザード、制御ハザード

5. パイプライン処理（2）

フォワーディング、遅延分岐、分岐予測、命令スケジューリング

6. キャッシュ

記憶階層と局所性、透過性、キャッシュ、ライトスルーとライトバック、ダイレクトマップ型、フルアソシアティブ型、セットアソシアティブ型、キャッシュミス

7. 仮想記憶

仮想記憶、ページフォールト、TLB、物理アドレスキャッシュ、仮想アドレスキャッシュ、メモリアクセス機構

講義の概要と予定 (2 / 2)

8. 基本CPUの設計

ディジタル回路の入力、Verilog HDL、シミュレーションによる動作検証、アセンブラ、基本プロセッサの設計、基本プロセッサのシミュレーションによる検証

9. 命令レベル並列処理(1)

並列処理、並列処理パイプライン、VLIW、スーパースカラ、並列処理とハザード

10. 命令レベル並列処理(2)

静的最適化、ループアンローリング、ソフトウェアパイプライン、トレーススケジューリング

11. アウトオブオーダー処理

インオーダーとアウトオブオーダー、フロー依存、逆依存、出力依存、命令ウィンドウ、リザベーションステーション、レジスタリネーミング、マッピングテーブル、リオーダーバッファ、プロセッサの性能

12. 入出力と周辺装置

周辺装置、ディスプレイ、二次記憶装置、ハードウェアインタフェース、割り込みとポーリング、アービタ、DMA、例外処理

試験: 7月

1. コンピュータアーキテクチャ入門

■ 内容

- デジタルな表現
 - ・ 2進数
 - ・ 自然数の表現
 - ・ 負の数の表現
 - ・ 実数の表現
- 計算する
 - ・ 加算、減算
 - ・ ALU
- 計算のサイクル
 - ・ フリップフロップ
 - ・ レジスタ
 - ・ レジスタとALUの結合
- 練習問題

デジタルな表現

- 2進数 $a_m a_{m-1} \dots a_1 a_0$

$$X = a_m 2^m + a_{m-1} 2^{m-1} + \dots + a_1 2 + a_0 \quad (a_i = 0, 1)$$

- 0,1の表し方

- ・ 0: 電圧の低い状態(L)
- ・ 1: 電圧の高い状態(H)

- ビット (bit, binary digit, **b**)

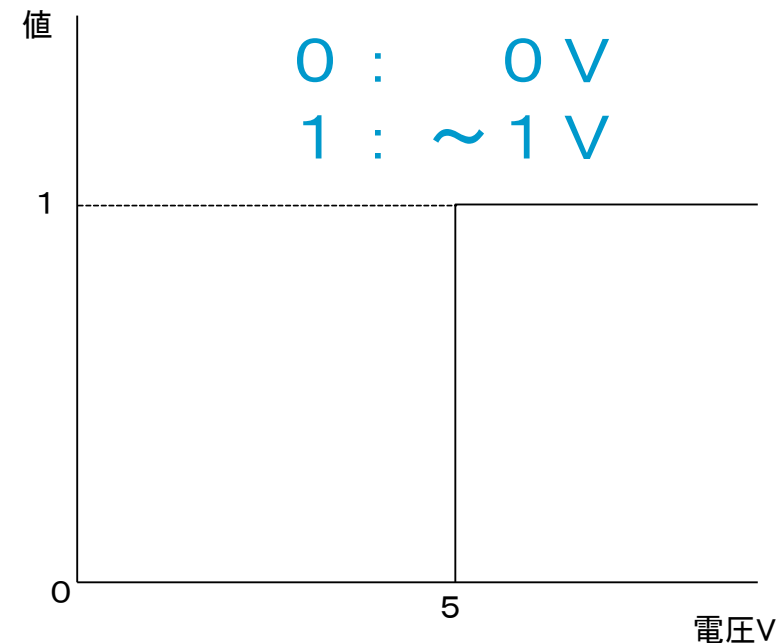
- 情報の単位。0か1の値を取る

- 電線で自然数を表現する

- 電線を束ねて、それぞれの線で「桁」を表現していると考え

- バイト(B)

- 8ビットのこと。2進数で8桁、16進数で2桁



ビットとバイト

1 ビット (bit) : 2 進数の 1 桁 (binary digit の略)

1 バイト (byte) : 8 ビット

- 1 ビットは, 0 と 1 の 2 通りの表現ができる
- 1 バイトは, 00000000 ~ 11111111 の
 $2^8 = 256$ 通りの表現ができる

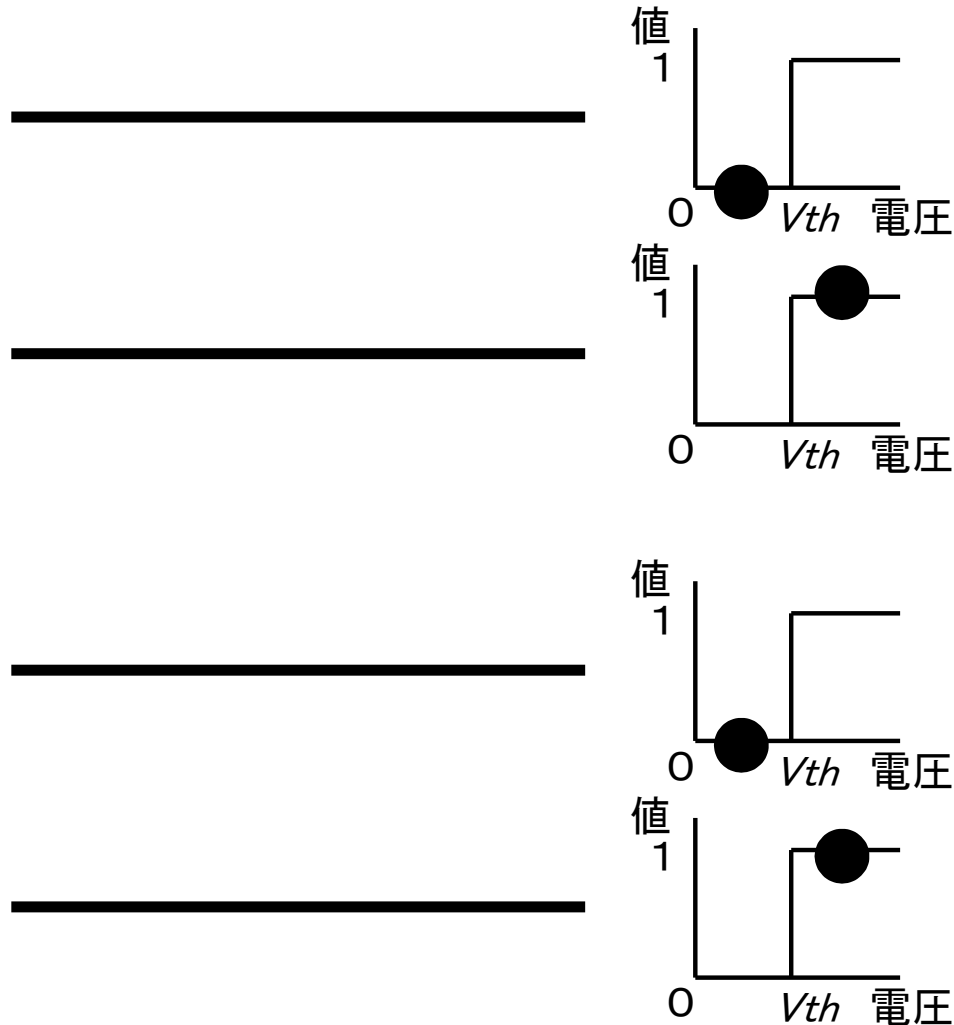
10 進数の 253 は $2 \times 10^2 + 5 \times 10^1 + 3 \times 10^0 = 253$

2 進数の 1101 は $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{(10)}$

ちなみに 0 と 1 だけだと表現しづらいので通常は 16 進数で表現する

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f 1 バイト 2 桁の 16 進数 eg 0x9a = $9 \times 16 + 10 = 154_{(10)}$

n本の線によるnビット自然数の表現



符号なし整数 (メモ)

4ビットで 0 ~ 15 の整数を表現できる

m ビットで $0 \sim 2^m - 1$ の整数を表現できる

$$7 + 4_{(10)} = 0111 + 0100 = 1011 = 11_{(10)}$$

$$7 - 4_{(10)} = 0111 - 0100 = 0011 = 3_{(10)}$$

数値	ビット列
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10(a)	1010
11(b)	1011
12(c)	1100
13(d)	1101
14(e)	1110
15(f)	1111

負の数の表現

- 負の数: 2の補数表現を使う

$$-X \rightarrow 2^n - X$$

(Xの各桁の1と0を反転し、結果に1を加えたもの)

符 号	X が正のとき X X が負のとき $2^n - X$
--------	--------------------------------------

2の補数表示

符号は、正のとき0, 負のとき1

-2^{n-1} から $2^{n-1} - 1$ までの数を表すことができる

符号あり整数 (メモ)

負の整数は2の補数で表現する

ビット反転して1を足す

4は0100だから, -4は0100→1011→1100

4ビットで -8 ~ 7 の整数を表現できる

m ビットで
 $-2^{m-1} \sim 2^{m-1} - 1$ の整数を表現できる

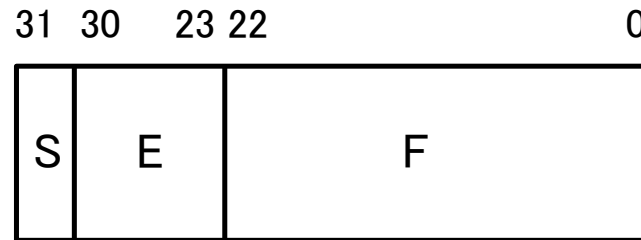
減算は加算で計算できる

$$\begin{aligned} 7 - 4_{(10)} &= 7 + (-4)_{(10)} \\ &= 0111 + 1100 = 10011 = 3_{(10)} \\ &\quad \text{※桁あふれは無視する} \end{aligned}$$

数値	ビット列
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111

実数の表現

- 固定小数点による表現
 - 整数の表現と同じ。何桁目かに小数点があると「約束」
 - 特別な回路を用意する必要なし
 - 演算のたびに小数点の位置合わせのためのシフト(shift, 桁移動)が必要。プログラムで。
- 浮動小数点による表現
 - 演算のために特別な回路を用意する。プログラムによる補正は不要。



メモ ©Toshiyuki Nakata

この考え方

正規化するために

Fの最上位ビットに1

が立つようにE/Fを

補正する。

Fの最上位ビットが1と

わかっているのを、

それを無くして、

Fをもう1ビット左シフトして

Fの桁数を稼ぐ

$$\left\{ \begin{array}{ll} E = 0 & \left\{ \begin{array}{ll} F = 0 & 0 \\ F \neq 0 & (-1)^S \times (0.F) \times 2^{-126} \end{array} \right. \\ 0 < E < 255 & (-1)^S \times (1.F) \times 2^{E-127} \\ E = 255 & \left\{ \begin{array}{ll} F = 0 & (-1)^S \times \infty \\ F \neq 0 & NaN \text{ (No Number)} \end{array} \right. \end{array} \right.$$

- 実数は近似値で表現している。誤差解析が必要

文字コード(文字とビット列との対応)

1バイトで表現

$2^8 = 256$ 個の表現が可能

- ASCII(アスキー) : 7ビットで表現
- JIS X 0201 : 8ビットでJIS仮名(半角カナ)も表現

2バイトで表現

- JIS
- シフト JIS ← Windows や Macintosh で使用
- EUC ← Linux 等の UNIX で使用

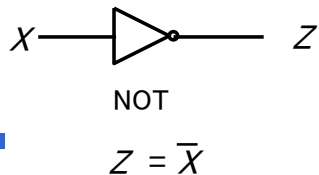
マルチバイトで表現

- Unicode : 世界中のほぼすべての言語をサポート

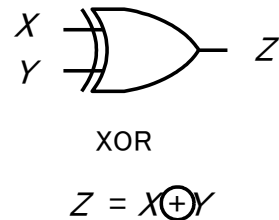
OS やソフトウェアの違い, 機種依存文字に注意

計算する

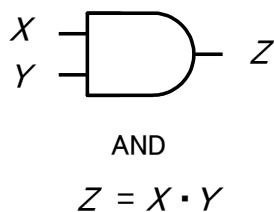
- 計算＝論理関数の実現
 - － 1個以上の2進数データから新たな2進数データを作成すること
- 組合せ論理回路
 - － 論理関数を実現する回路
 - － 数種類の基本素子で実現される



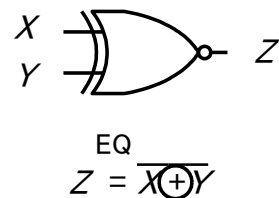
X	Z
0	1
1	0



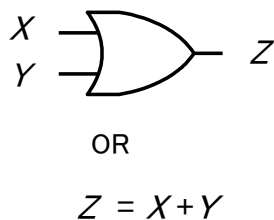
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0



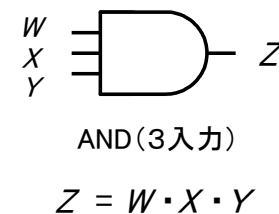
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1



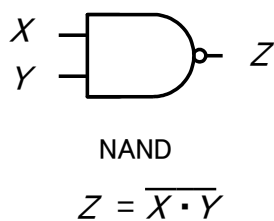
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1



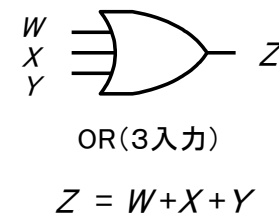
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1



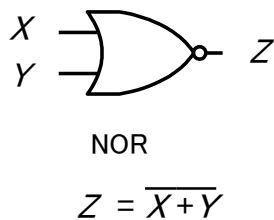
W	X	Y	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



X	Y	Z
0	0	1
0	1	1
1	0	1
1	1	0

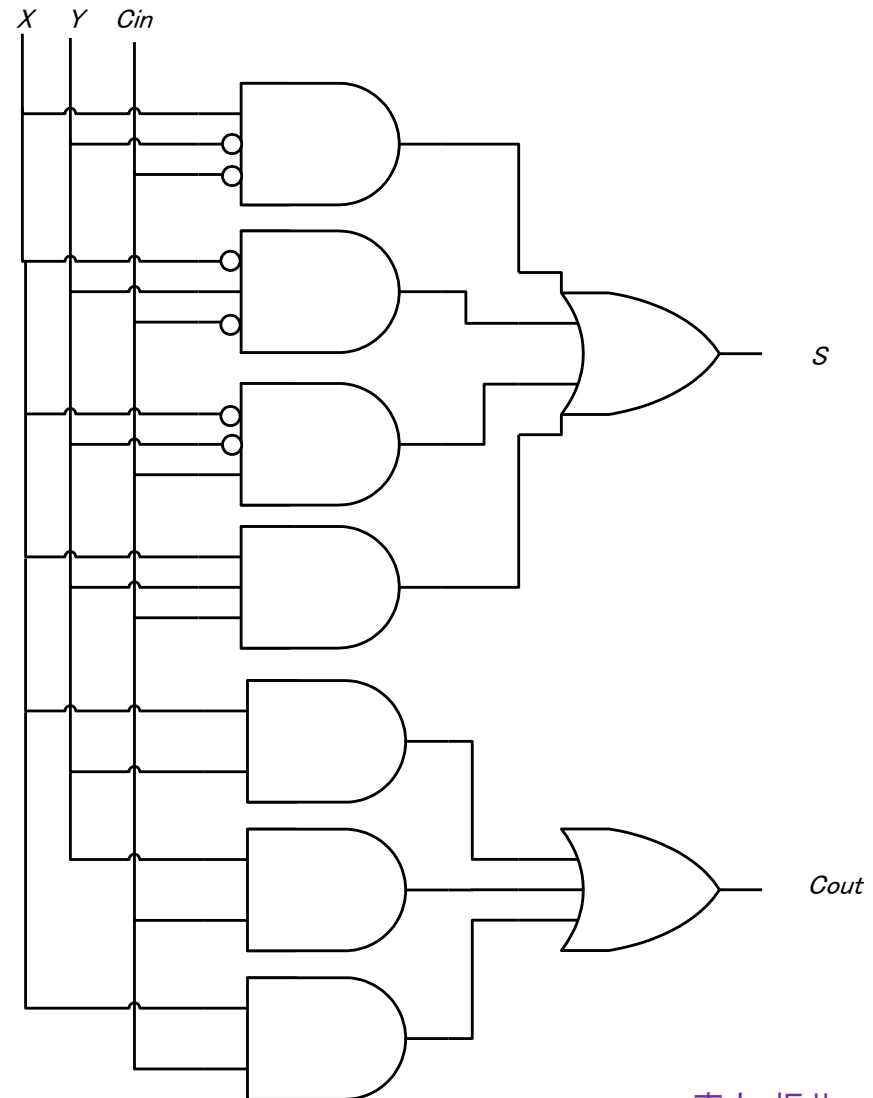
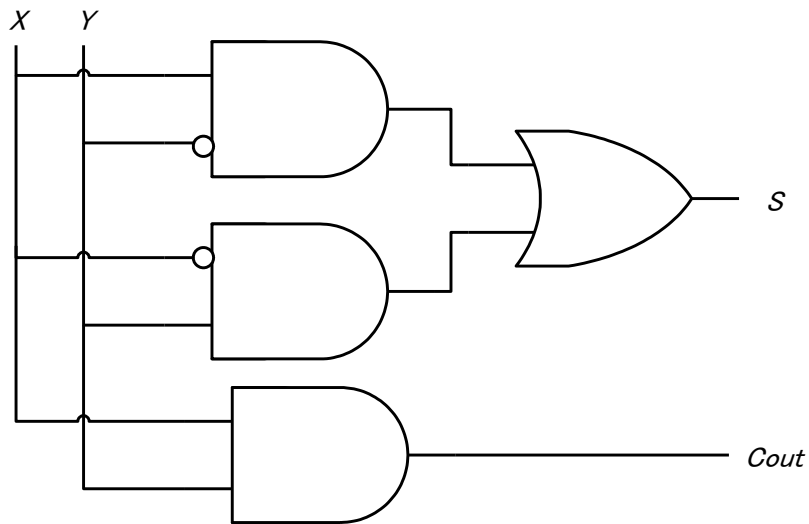


W	X	Y	Z
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

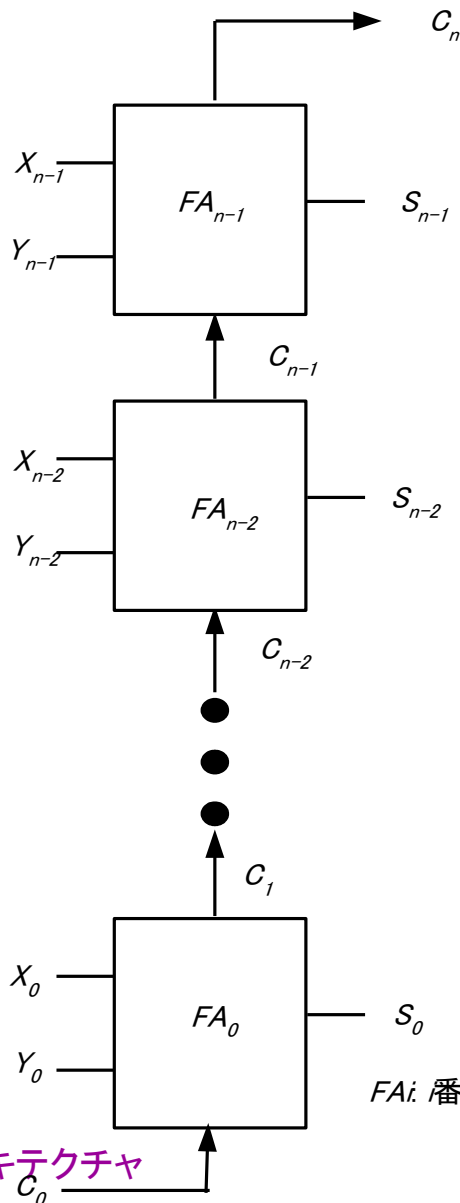


X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	0

演算回路(1): 1ビット加算器



演算回路(2): nビット加算器



リプルキャリ型加算器

メモ : ©Toshiyuki Nakata

C_0, Y_0, X_0 から C_1 が決まり

C_1, X_1, Y_1 から C_2 が決まり。。。

$C_{n-2}, X_{n-2}, Y_{n-2}$ から C_{n-1} が決まり、

$C_{n-1}, X_{n-1}, Y_{n-1}$ から C_n が決まるという感じで
 C_n の決定には最悪、全1ビット加算機の値が
決まらなないと決まらないう=>遅い！！

cf. キャリルックアヘッド

FA_i : i番目のビットの加算回路

キャリルックアヘッドの考え方(メモ)

$$\begin{aligned} \text{Cout} &= \text{Cin} * X + \text{Cin} * Y + X * Y \\ &= \text{Cin}(X + Y) + X * Y \end{aligned}$$

を各桁で展開

$$\text{Cout}_0 = X_0 * Y_0$$

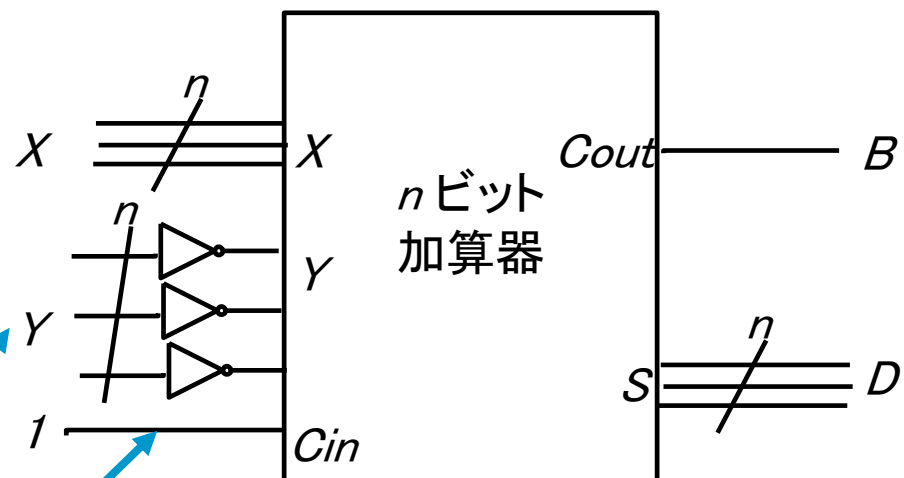
$$\begin{aligned} \text{Cout}_1 &= \text{Cout}_0(X_1 + Y_1) + X_1 * Y_1 \\ &= X_0 * Y_0 (X_1 + Y_1) + X_1 * Y_1 \end{aligned}$$

©Toshiyuki Nakata

$$\begin{aligned} \text{Cout}_2 &= \text{Cout}_1(X_2 + Y_2) + X_2 * Y_2 \\ &= (X_0 * Y_0 (X_1 + Y_1) + X_1 * Y_1) (X_2 + Y_2) + X_2 * Y_2 \end{aligned}$$

$$\begin{aligned} \text{Cout}_3 &= \text{Cout}_2(X_3 + Y_3) + X_3 * Y_3 \\ &= ((X_0 * Y_0 (X_1 + Y_1) + X_1 * Y_1) (X_2 + Y_2) + X_2 * Y_2) (X_3 + Y_3) + X_3 * Y_3 \end{aligned}$$

減算器



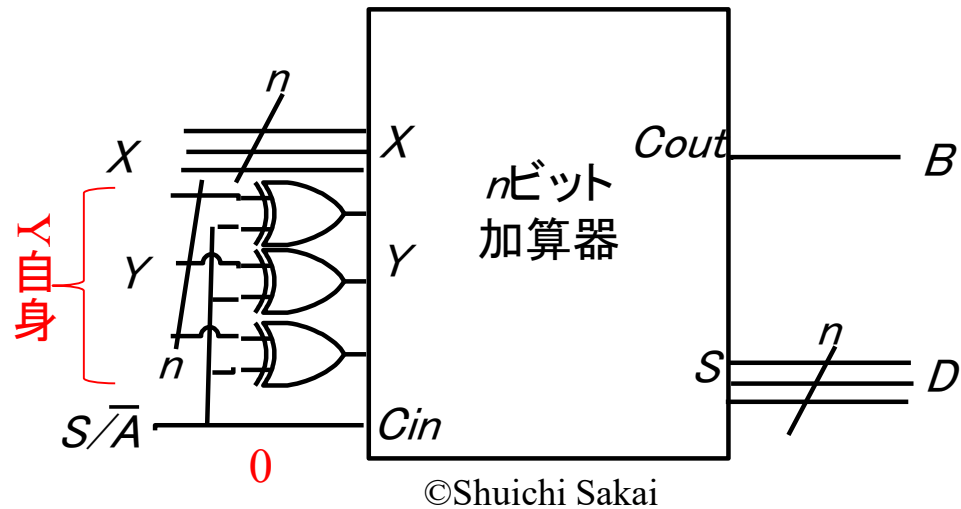
減算器

©Shuichi Sakai

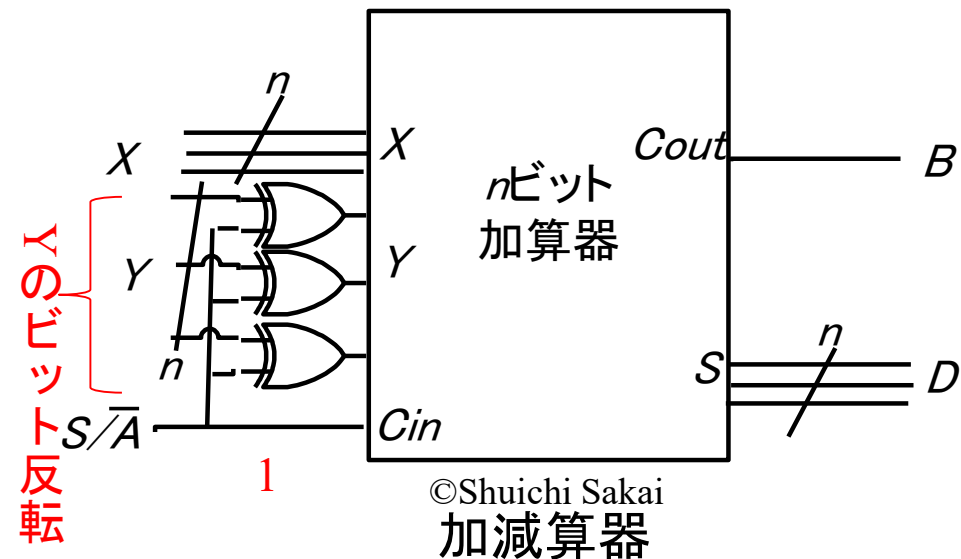
$X - Y = X + (-Y)$
 $(-Y) = Y$ の補数
= Y のビット反転+1

加減算器

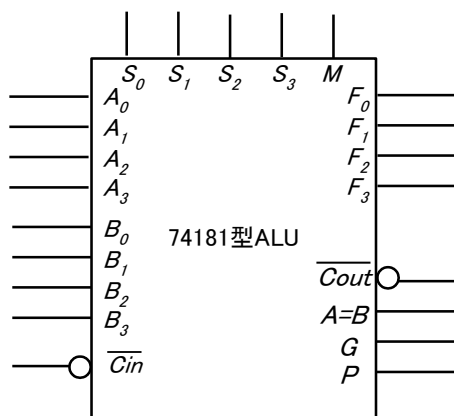
$S/\bar{A}=0 \Rightarrow$ 加算



$S/\bar{A}=1 \Rightarrow$ 減算



ALU

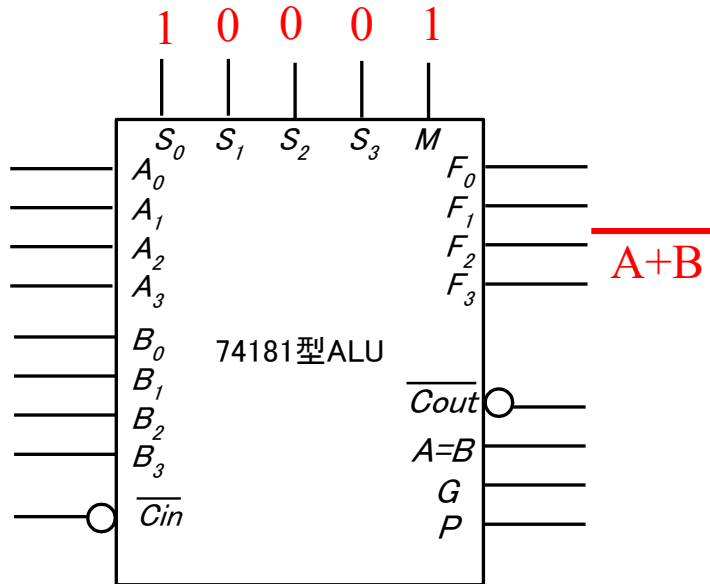


制御信号 $S_3 S_2 S_1 S_0$	$M=1$: 論理演算	$M=0$: 算術演算	
		$\overline{Cin}=0$	$\overline{Cin}=1$
0 0 0 0	$F=\overline{A}$	$F=A$	$F=A \text{ PLUS } 1$
0 0 0 1	$F=\overline{A+B}$	$F=A+B$	$F=(A+B) \text{ PLUS } 1$
0 0 1 0	$F=\overline{A} \cdot B$	$F=A+\overline{B}$	$F=(A+\overline{B}) \text{ PLUS } 1$
0 0 1 1	$F=0$	$F=1111$	$F=\text{ZERO}$
0 1 0 0	$F=\overline{A} \cdot \overline{B}$	$F=A \text{ PLUS } A \cdot \overline{B}$	$F=A \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$
0 1 0 1	$F=\overline{B}$	$F=(A+B) \text{ PLUS } A \cdot \overline{B}$	$F=(A+B) \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$
0 1 1 0	$F=A \oplus B$	$F=A \text{ MINUS } B \text{ MINUS } 1$	$F=A \text{ MINUS } B$
0 1 1 1	$F=A \cdot \overline{B}$	$F=A \cdot \overline{B} \text{ MINUS } 1$	$F=A \cdot \overline{B}$
1 0 0 0	$F=\overline{A} + B$	$F=A \text{ PLUS } A \cdot B$	$F=A \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 0 1	$F=\overline{A \oplus B}$	$F=A \text{ PLUS } B$	$F=A \text{ PLUS } B \text{ PLUS } 1$
1 0 1 0	$F=B$	$F=(A+\overline{B}) \text{ PLUS } AB$	$F=(A+\overline{B}) \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 1 1	$F=A \cdot B$	$F=A \cdot B \text{ MINUS } 1$	$F=A \cdot B$
1 1 0 0	$F=1$	$F=A \text{ PLUS } A$	$F=A \text{ PLUS } A \text{ PLUS } 1$
1 1 0 1	$F=A+\overline{B}$	$F=(A+B) \text{ PLUS } A$	$F=(A+B) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 0	$F=A+B$	$F=(A+\overline{B}) \text{ PLUS } A$	$F=(A+\overline{B}) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 1	$F=A$	$F=A \text{ MINUS } 1$	$F=A$

(b) 動作： 重要なものは太字とした

図 1.10 74181 型 ALU (4 ビット) (前ページからの続き)

ALU(算術論理演算ユニット)



©Shuichi Sakai

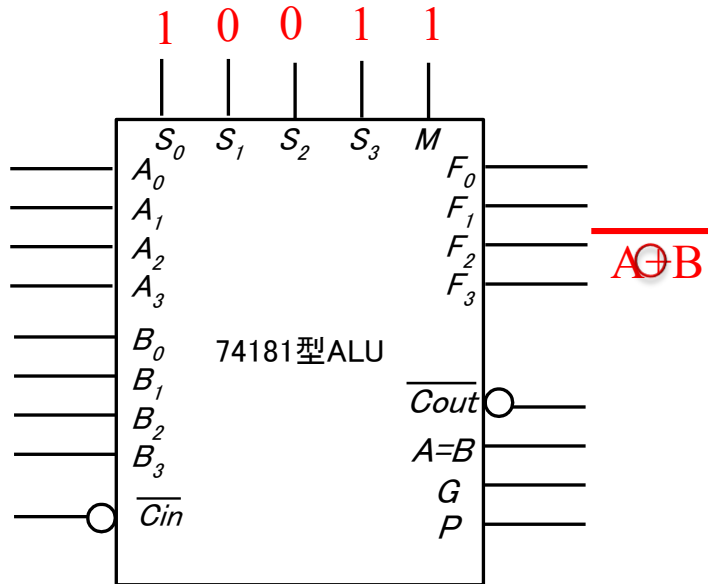
A=BはAtoBが等しいときに1
G,Pはキャリルックアヘッド用

制御信号 $S_3 S_2 S_1 S_0$	$M=1$: 論理演算	$M=0$: 算術演算	
		$\overline{Cin}=0$	$\overline{Cin}=1$
0 0 0 0	$F=\overline{A}$	$F=A$	$F=A \text{ PLUS } 1$
0 0 0 1	$F=\overline{A+B}$	$F=A+B$	$F=(A+B) \text{ PLUS } 1$
0 0 1 0	$F=\overline{A \cdot B}$	$F=A+\overline{B}$	$F=(A+\overline{B}) \text{ PLUS } 1$
0 0 1 1	$F=0$	$F=1111$	$F=\text{ZERO}$
0 1 0 0	$F=\overline{A \cdot B}$	$F=A \text{ PLUS } A \cdot \overline{B}$	$F=A \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$
0 1 0 1	$F=\overline{B}$	$F=(A+B) \text{ PLUS } A \cdot \overline{B}$	$F=(A+B) \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$
0 1 1 0	$F=A \oplus B$	$F=A \text{ MINUS } B \text{ MINUS } 1$	$F=A \text{ MINUS } B$
0 1 1 1	$F=A \cdot \overline{B}$	$F=A \cdot \overline{B} \text{ MINUS } 1$	$F=A \cdot \overline{B}$
1 0 0 0	$F=\overline{A+B}$	$F=A \text{ PLUS } A \cdot B$	$F=A \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 0 1	$F=\overline{A \oplus B}$	$F=A \text{ PLUS } B$	$F=A \text{ PLUS } B \text{ PLUS } 1$
1 0 1 0	$F=B$	$F=(A+\overline{B}) \text{ PLUS } AB$	$F=(A+\overline{B}) \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 1 1	$F=A \cdot B$	$F=A \cdot B \text{ MINUS } 1$	$F=A \cdot B$
1 1 0 0	$F=1$	$F=A \text{ PLUS } A$	$F=A \text{ PLUS } A \text{ PLUS } 1$
1 1 0 1	$F=A+\overline{B}$	$F=(A+B) \text{ PLUS } A$	$F=(A+B) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 0	$F=A+B$	$F=(A+\overline{B}) \text{ PLUS } A$	$F=(A+\overline{B}) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 1	$F=A$	$F=A \text{ MINUS } 1$	$F=A$

(b) 動作: 重要なものは太字とした

図 1.10 74181 型 ALU (4 ビット) (前ページからの続き)

ALU(算術論理演算ユニット)



©Shuichi Sakai

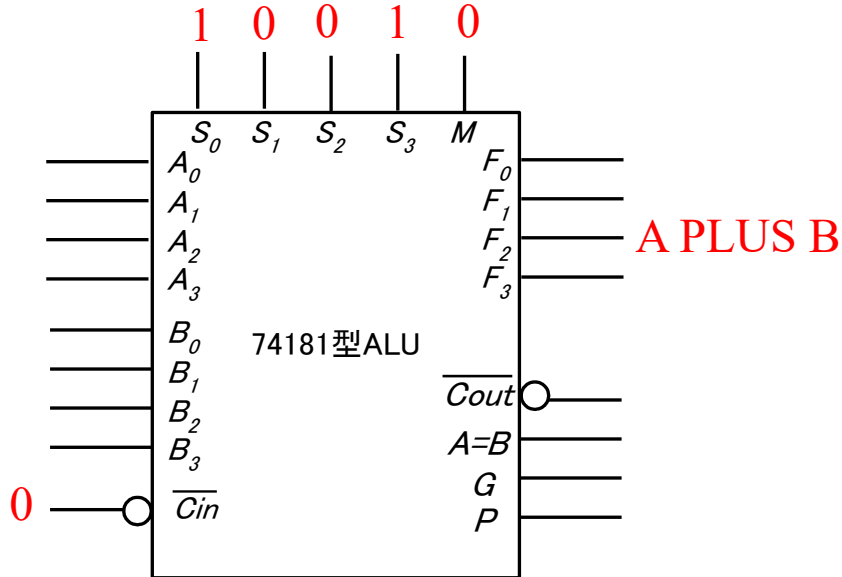
A=BはAtoBが等しいときに1
G,Pはキャリルックアヘッド用

制御信号 $S_3 S_2 S_1 S_0$	M=1: 論理演算	M=0: 算術演算	
		$\overline{Cin}=0$	$\overline{Cin}=1$
0 0 0 0	$F=\overline{A}$	$F=A$	$F=A \text{ PLUS } 1$
0 0 0 1	$F=\overline{A+B}$	$F=A+B$	$F=(A+B) \text{ PLUS } 1$
0 0 1 0	$F=\overline{A \cdot B}$	$F=A+\overline{B}$	$F=(A+\overline{B}) \text{ PLUS } 1$
0 0 1 1	$F=0$	$F=1111$	$F=\text{ZERO}$
0 1 0 0	$F=\overline{A \cdot B}$	$F=A \text{ PLUS } A \cdot \overline{B}$	$F=A \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$
0 1 0 1	$F=\overline{B}$	$F=(A+B) \text{ PLUS } A \cdot \overline{B}$	$F=(A+B) \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$
0 1 1 0	$F=A \oplus B$	$F=A \text{ MINUS } B \text{ MINUS } 1$	$F=A \text{ MINUS } B$
0 1 1 1	$F=A \cdot \overline{B}$	$F=A \cdot \overline{B} \text{ MINUS } 1$	$F=A \cdot \overline{B}$
1 0 0 0	$F=\overline{A+B}$	$F=A \text{ PLUS } A \cdot B$	$F=A \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 0 1	$F=\overline{A \oplus B}$	$F=A \text{ PLUS } B$	$F=A \text{ PLUS } B \text{ PLUS } 1$
1 0 1 0	$F=B$	$F=(A+\overline{B}) \text{ PLUS } AB$	$F=(A+\overline{B}) \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 1 1	$F=A \cdot B$	$F=A \cdot B \text{ MINUS } 1$	$F=A \cdot B$
1 1 0 0	$F=1$	$F=A \text{ PLUS } A$	$F=A \text{ PLUS } A \text{ PLUS } 1$
1 1 0 1	$F=A+\overline{B}$	$F=(A+B) \text{ PLUS } A$	$F=(A+B) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 0	$F=A+B$	$F=(A+\overline{B}) \text{ PLUS } A$	$F=(A+\overline{B}) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 1	$F=A$	$F=A \text{ MINUS } 1$	$F=A$

(b) 動作: 重要なものは太字とした

図 1.10 74181 型 ALU (4 ビット) (前ページからの続き)

ALU(算術論理演算ユニット)



©Shuichi Sakai

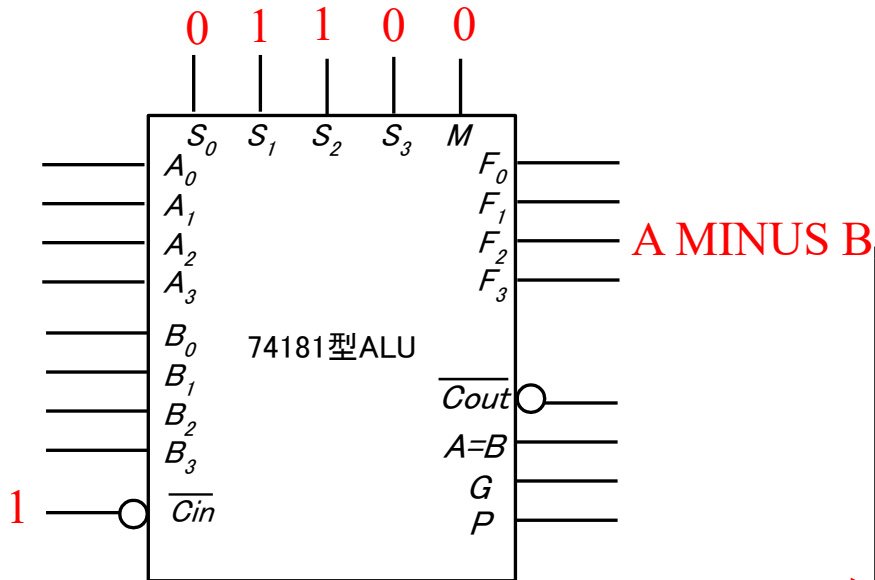
A=BはAtoBが等しいときに1
G,Pはキャリルックアヘッド用

制御信号 $S_3 S_2 S_1 S_0$	M=1: 論理演算	M=0: 算術演算	
		$\overline{Cin}=0$	$\overline{Cin}=1$
0 0 0 0	$F=\overline{A}$	$F=A$	$F=A \text{ PLUS } 1$
0 0 0 1	$F=\overline{A+B}$	$F=A+B$	$F=(A+B) \text{ PLUS } 1$
0 0 1 0	$F=\overline{A \cdot B}$	$F=A+\overline{B}$	$F=(A+\overline{B}) \text{ PLUS } 1$
0 0 1 1	$F=0$	$F=1111$	$F=\text{ZERO}$
0 1 0 0	$F=\overline{A \cdot B}$	$F=A \text{ PLUS } A \cdot \overline{B}$	$F=A \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$
0 1 0 1	$F=\overline{B}$	$F=(A+B) \text{ PLUS } A \cdot \overline{B}$	$F=(A+B) \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$
0 1 1 0	$F=A \oplus B$	$F=A \text{ MINUS } B \text{ MINUS } 1$	$F=A \text{ MINUS } B$
0 1 1 1	$F=A \cdot \overline{B}$	$F=A \cdot \overline{B} \text{ MINUS } 1$	$F=A \cdot \overline{B}$
1 0 0 0	$F=\overline{A+B}$	$F=A \text{ PLUS } A \cdot B$	$F=A \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 0 1	$F=A \oplus \overline{B}$	$F=A \text{ PLUS } B$	$F=A \text{ PLUS } B \text{ PLUS } 1$
1 0 1 0	$F=B$	$F=(A+\overline{B}) \text{ PLUS } AB$	$F=(A+\overline{B}) \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 1 1	$F=A \cdot B$	$F=A \cdot B \text{ MINUS } 1$	$F=A \cdot B$
1 1 0 0	$F=1$	$F=A \text{ PLUS } A$	$F=A \text{ PLUS } A \text{ PLUS } 1$
1 1 0 1	$F=A+\overline{B}$	$F=(A+B) \text{ PLUS } A$	$F=(A+B) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 0	$F=A+B$	$F=(A+\overline{B}) \text{ PLUS } A$	$F=(A+\overline{B}) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 1	$F=A$	$F=A \text{ MINUS } 1$	$F=A$

(b) 動作: 重要なものは太字とした

図 1.10 74181 型 ALU (4 ビット) (前ページからの続き)

ALU(算術論理演算ユニット)



©Shuichi Sakai

A=BはAtoBが等しいときに1
G,Pはキャリルックアヘッド用

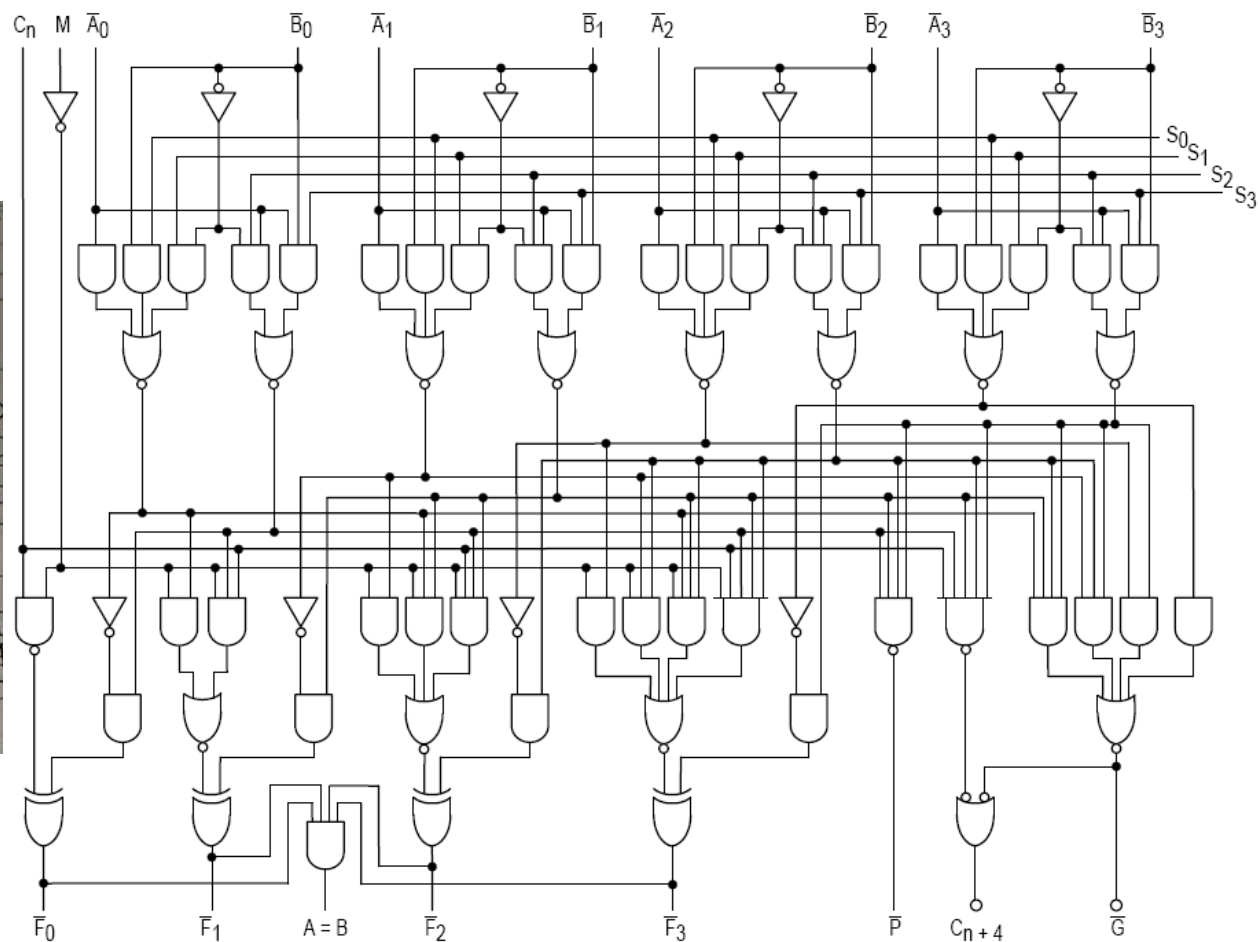
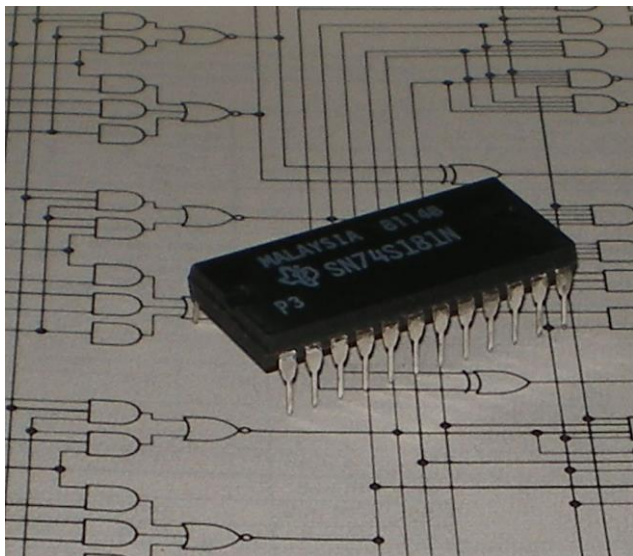
制御信号 $S_3 S_2 S_1 S_0$	$M=1$: 論理演算	$M=0$: 算術演算	
		$\overline{Cin}=0$	$\overline{Cin}=1$
0 0 0 0	$F=\overline{A}$	$F=A$	$F=A \text{ PLUS } 1$
0 0 0 1	$F=\overline{A+B}$	$F=A+B$	$F=(A+B) \text{ PLUS } 1$
0 0 1 0	$F=\overline{A \cdot B}$	$F=A+\overline{B}$	$F=(A+\overline{B}) \text{ PLUS } 1$
0 0 1 1	$F=0$	$F=1111$	$F=\text{ZERO}$
0 1 0 0	$F=\overline{A \cdot B}$	$F=A \text{ PLUS } A \cdot \overline{B}$	$F=A \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$
0 1 0 1	$F=\overline{B}$	$F=(A+B) \text{ PLUS } A \cdot \overline{B}$	$F=(A+B) \text{ PLUS } A \cdot \overline{B} \text{ PLUS } 1$
0 1 1 0	$F=A \oplus B$	$F=A \text{ MINUS } B \text{ MINUS } 1$	$F=A \text{ MINUS } B$
0 1 1 1	$F=A \cdot \overline{B}$	$F=A \cdot \overline{B} \text{ MINUS } 1$	$F=A \cdot \overline{B}$
1 0 0 0	$F=\overline{A+B}$	$F=A \text{ PLUS } A \cdot B$	$F=A \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 0 1	$F=\overline{A \oplus B}$	$F=A \text{ PLUS } B$	$F=A \text{ PLUS } B \text{ PLUS } 1$
1 0 1 0	$F=B$	$F=(A+\overline{B}) \text{ PLUS } AB$	$F=(A+\overline{B}) \text{ PLUS } A \cdot B \text{ PLUS } 1$
1 0 1 1	$F=A \cdot B$	$F=A \cdot B \text{ MINUS } 1$	$F=A \cdot B$
1 1 0 0	$F=1$	$F=A \text{ PLUS } A$	$F=A \text{ PLUS } A \text{ PLUS } 1$
1 1 0 1	$F=A+\overline{B}$	$F=(A+B) \text{ PLUS } A$	$F=(A+B) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 0	$F=A+B$	$F=(A+\overline{B}) \text{ PLUS } A$	$F=(A+\overline{B}) \text{ PLUS } A \text{ PLUS } 1$
1 1 1 1	$F=A$	$F=A \text{ MINUS } 1$	$F=A$

(b) 動作: 重要なものは太字とした

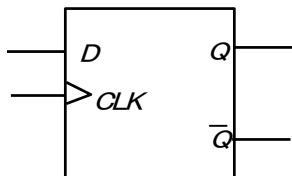
図 1.10 74181 型 ALU (4 ビット) (前ページからの続き)

74181の回路図

- CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=168473>
- By 74s181 at English Wikipedia – Transferred from en.wikipedia to Commons by audriusa., Public Domain, <https://commons.wikimedia.org/w/index.php?curid=18363509>



フリップフロップ

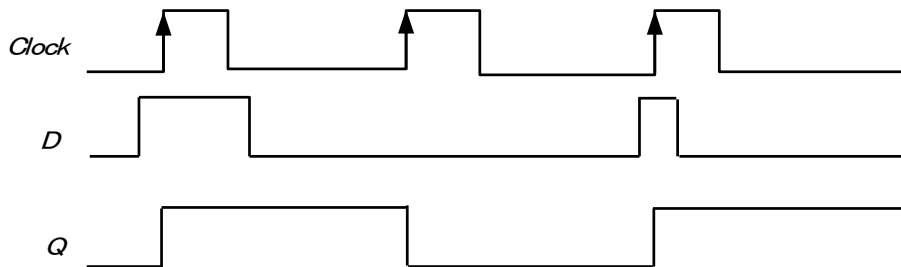


(a) 表記

D	CLK	Q	$\overline{Q}$
0	↑	0	1
1	↑	1	0
X	↑以外	Q	$\overline{Q}$

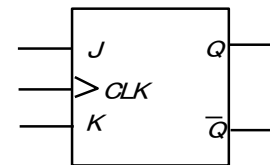
↑は信号の立ち上がりを表し、Xは入力が0でも1でもいいことを表す

(b) 動作規則



(c) 動作例

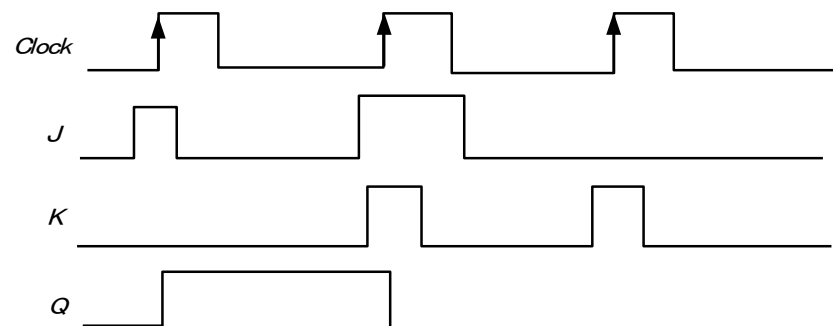
図 1.11 D フリップフロップ



(a) 表記

J	K	CLK	Q	$\overline{Q}$
0	0	↑	Q	$\overline{Q}$
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	$\overline{Q}$	Q
0	0	↑以外	Q	$\overline{Q}$

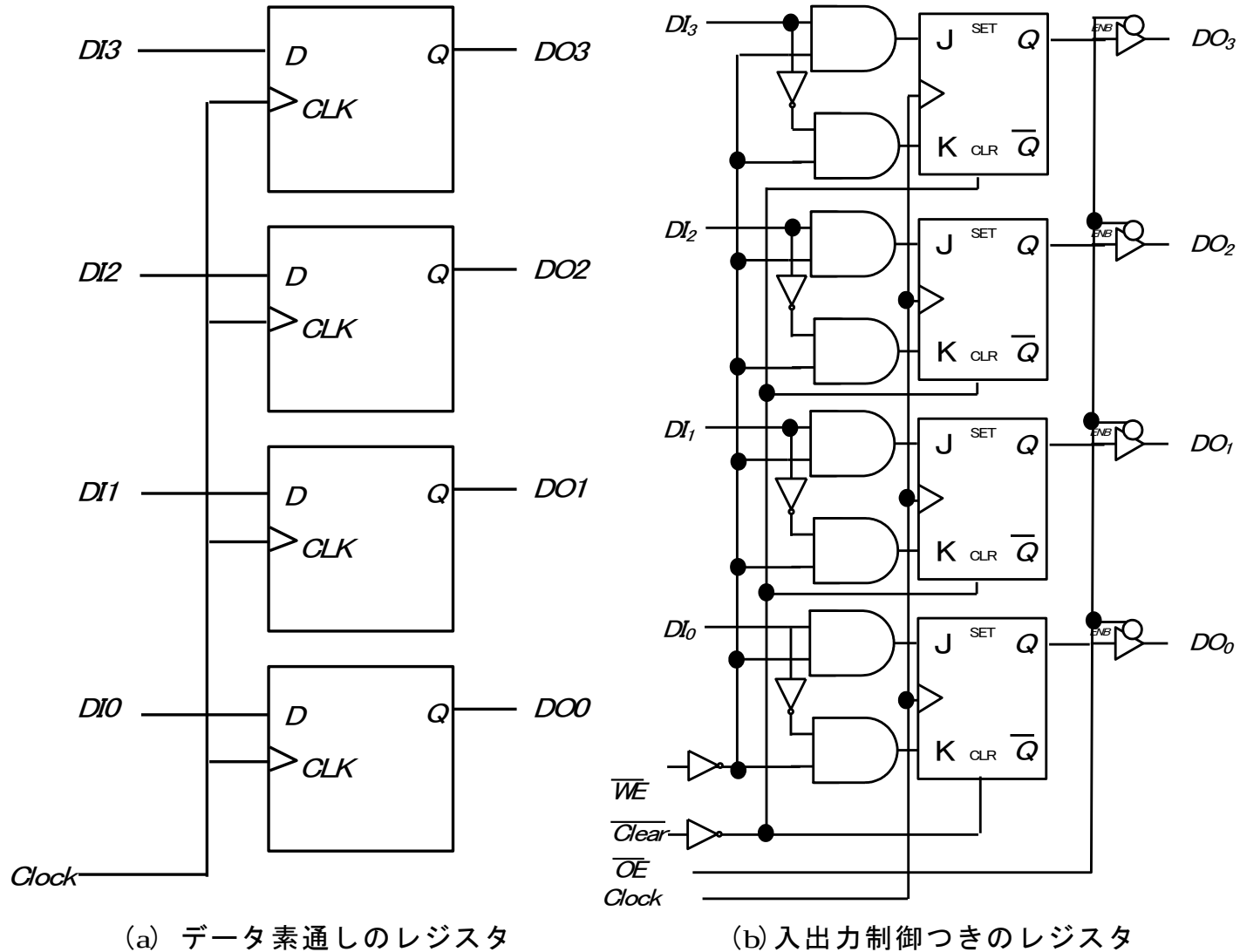
(b) 動作規則



(c) 動作例

図 1.12 JK フリップフロップ

レジスタ



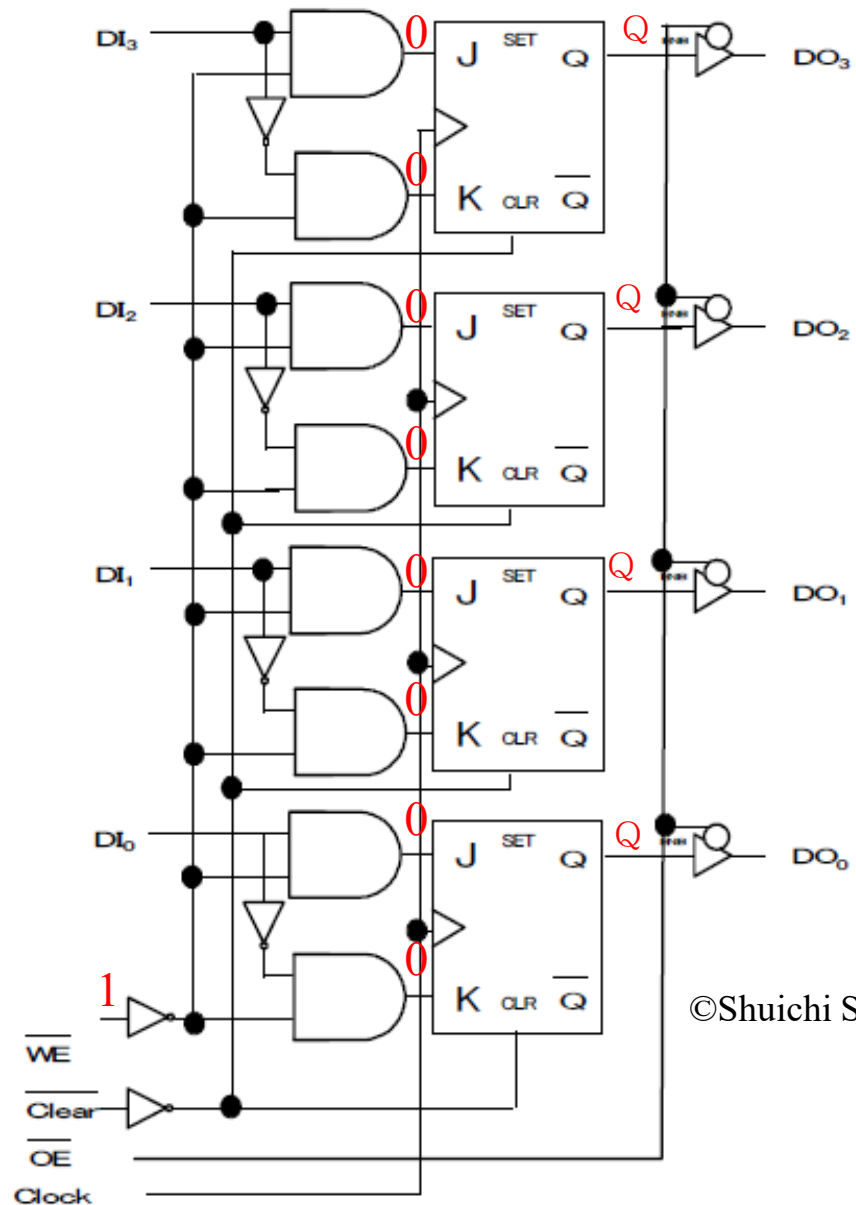
レジスタ

$\overline{WE}=1$ の時

→

J	K	CLK	Q	$\overline{Q}$
0	0	↑	Q	$\overline{Q}$
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	$\overline{Q}$	Q
0	0	↑以外	Q	$\overline{Q}$

(b) 動作規則



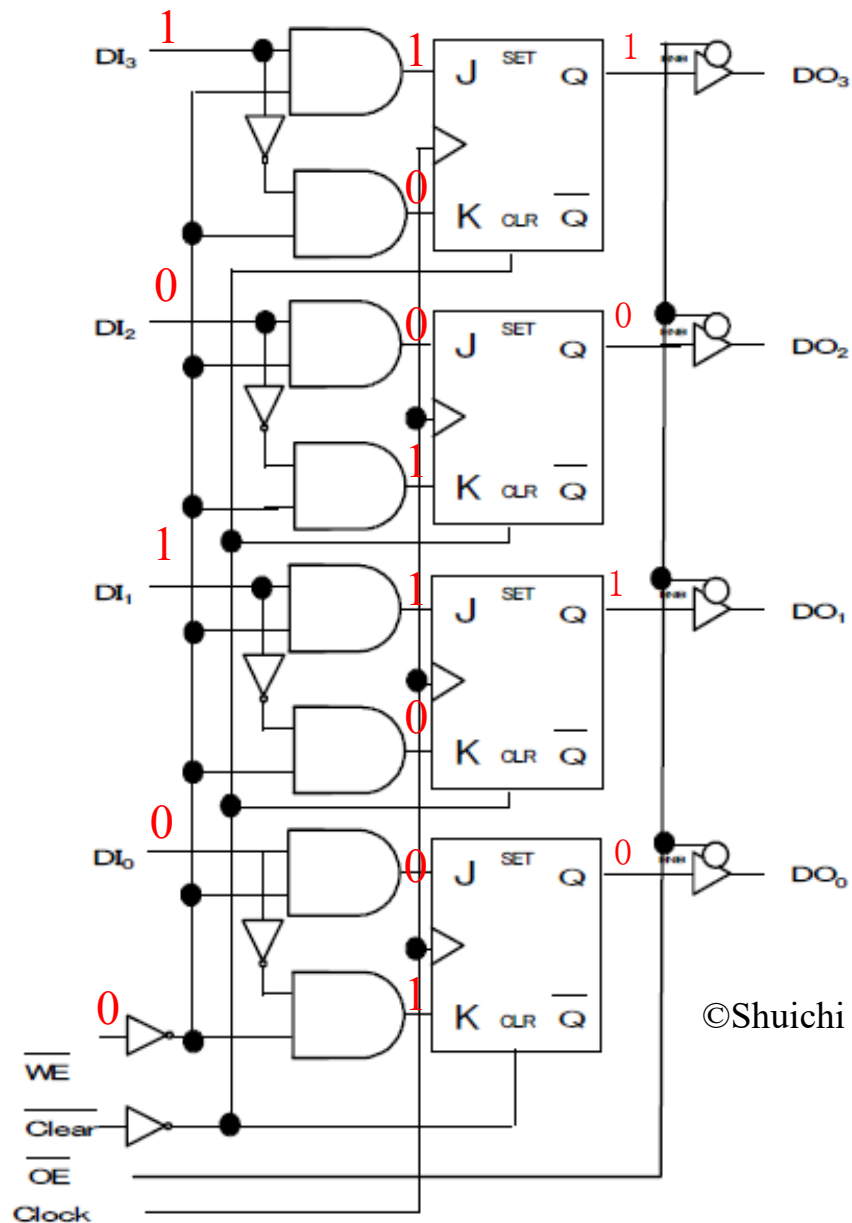
©Shuichi Sakai

レジスタ

$\overline{WE}=0$
D=1010の時

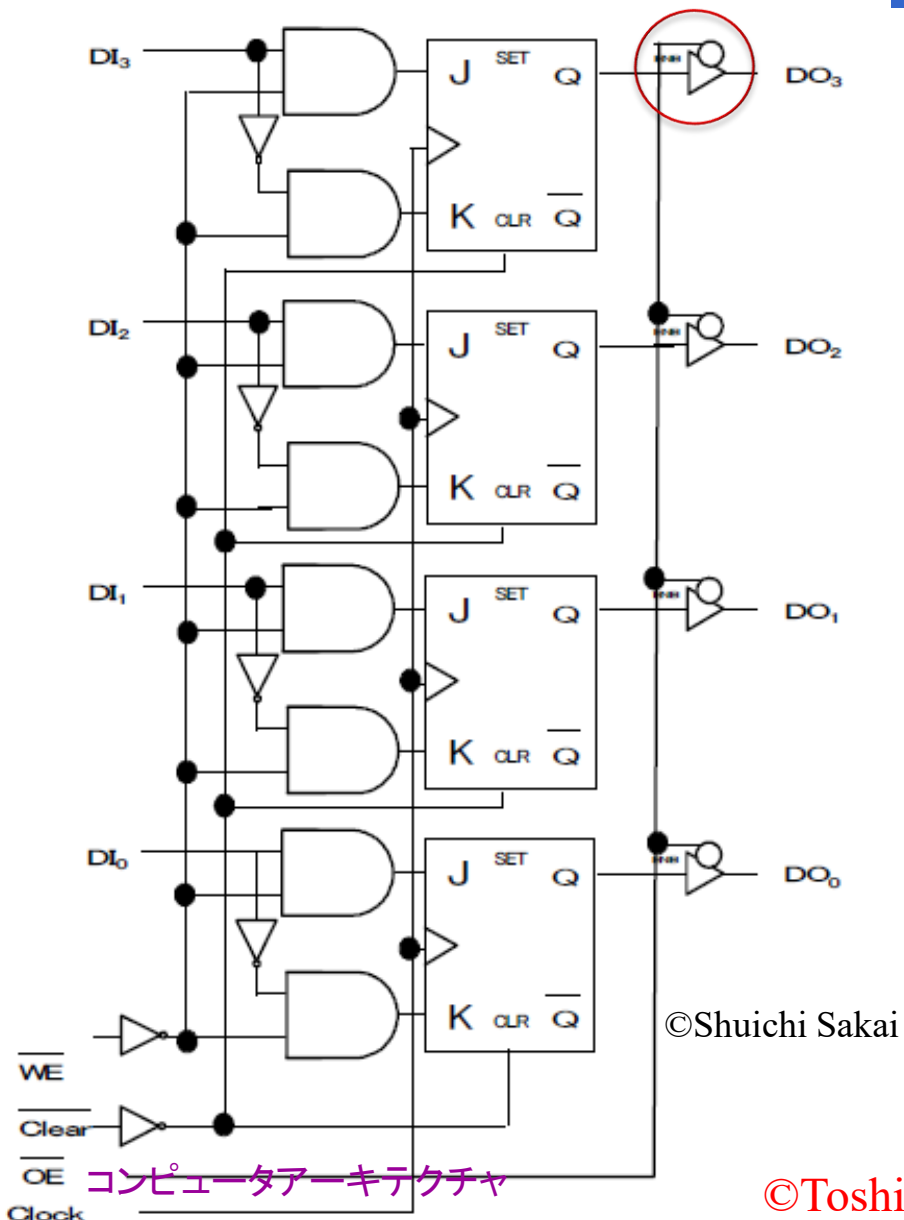
J	K	CLK	Q	$\overline{Q}$
0	0	↑	Q	$\overline{Q}$
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	$\overline{Q}$	Q
0	0	↑以外	Q	$\overline{Q}$

(b) 動作規則



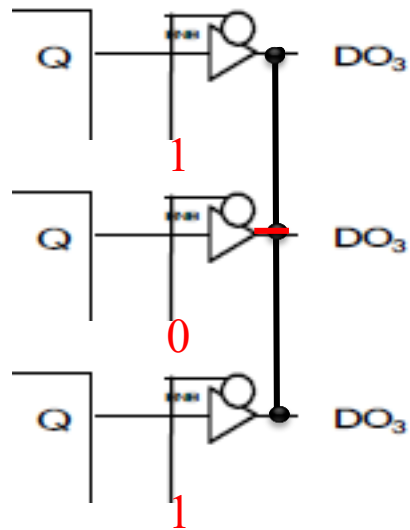
©Shuichi Sakai

Tri-Stateゲートについて(メモ)

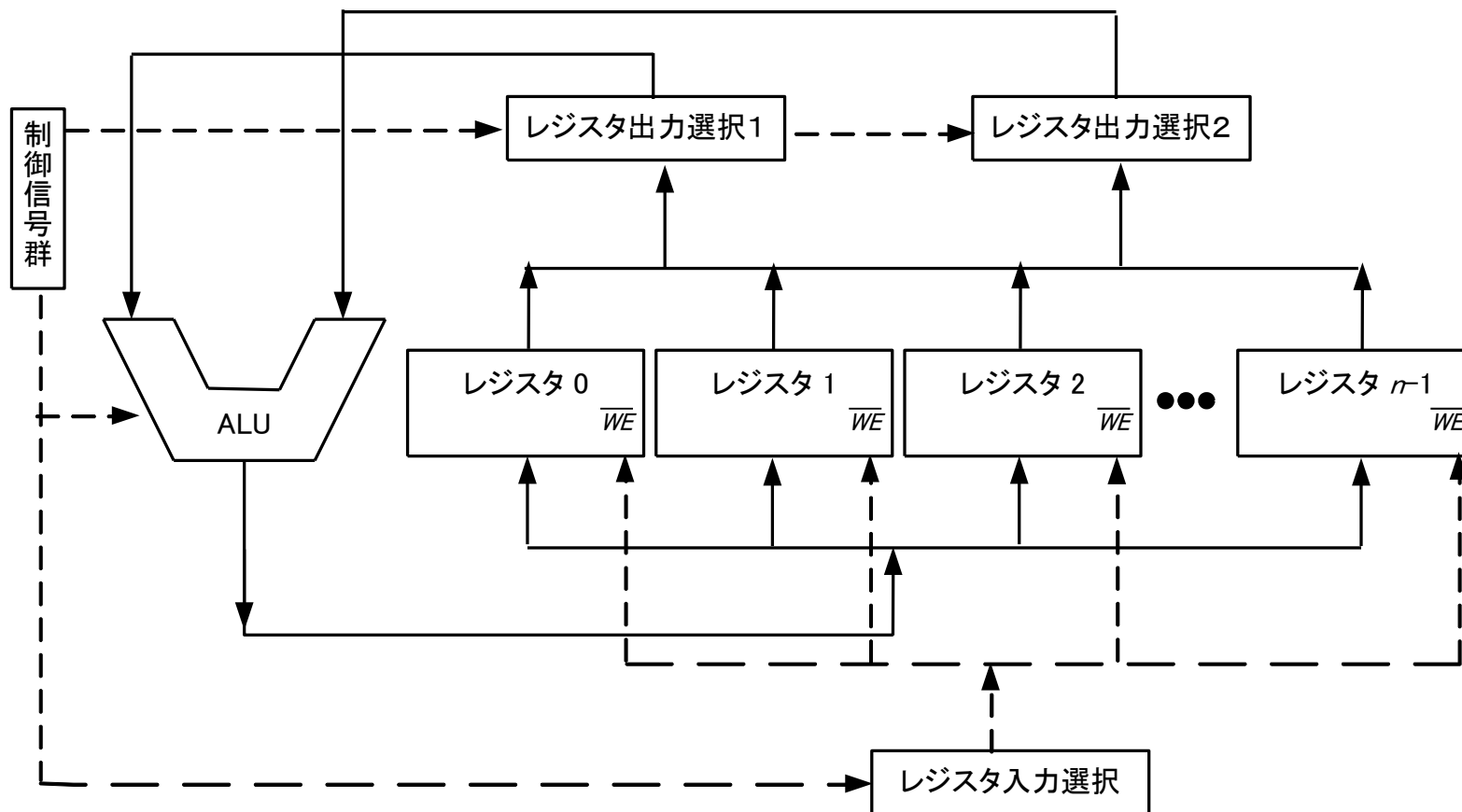


OE=10の時はトランジスタの出力が
Hi-Impedance（電流が流れない）状態に
なっているので、出力線に影響を与えない
。

従って複数のレジスタの出力を結線し、どれか一つだけOE=0にするとそのレジスタの出力が有効になる。ただし、実際の計算機回路では出力選択回路を用いる場合もある。(実際少し破綻を来す例も有り)

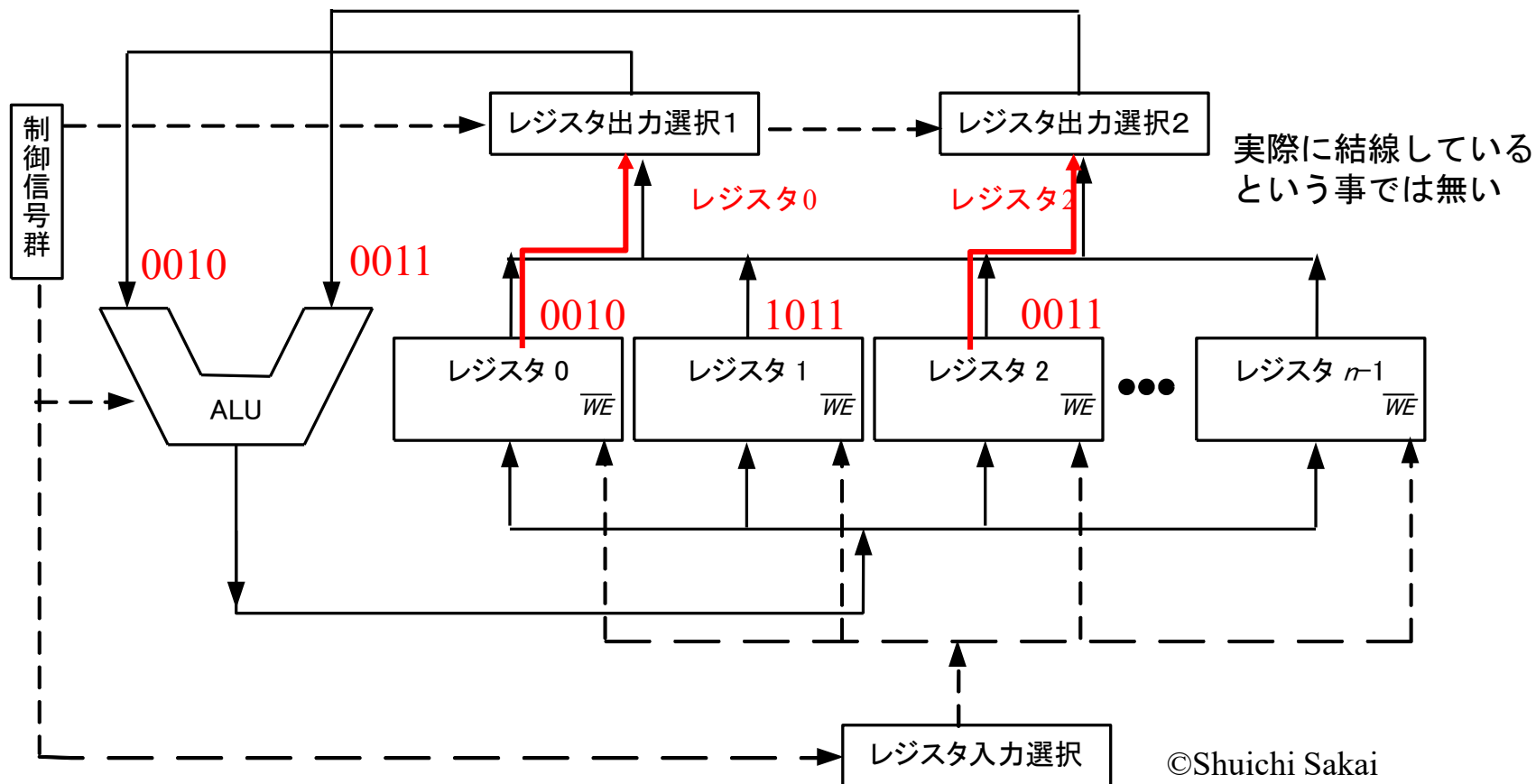


計算のサイクル(少し概念的)

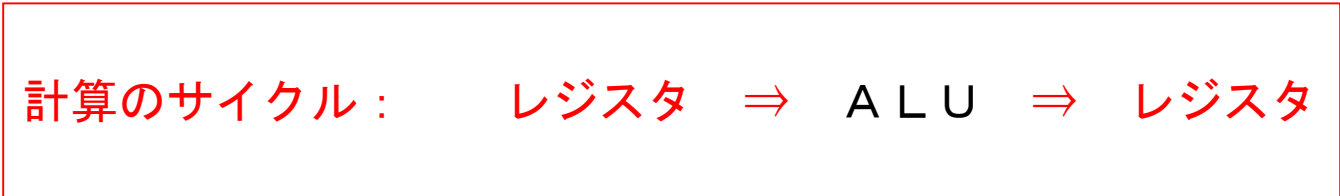


計算のサイクル： レジスタ $\Rightarrow$ ALU $\Rightarrow$ レジスタ

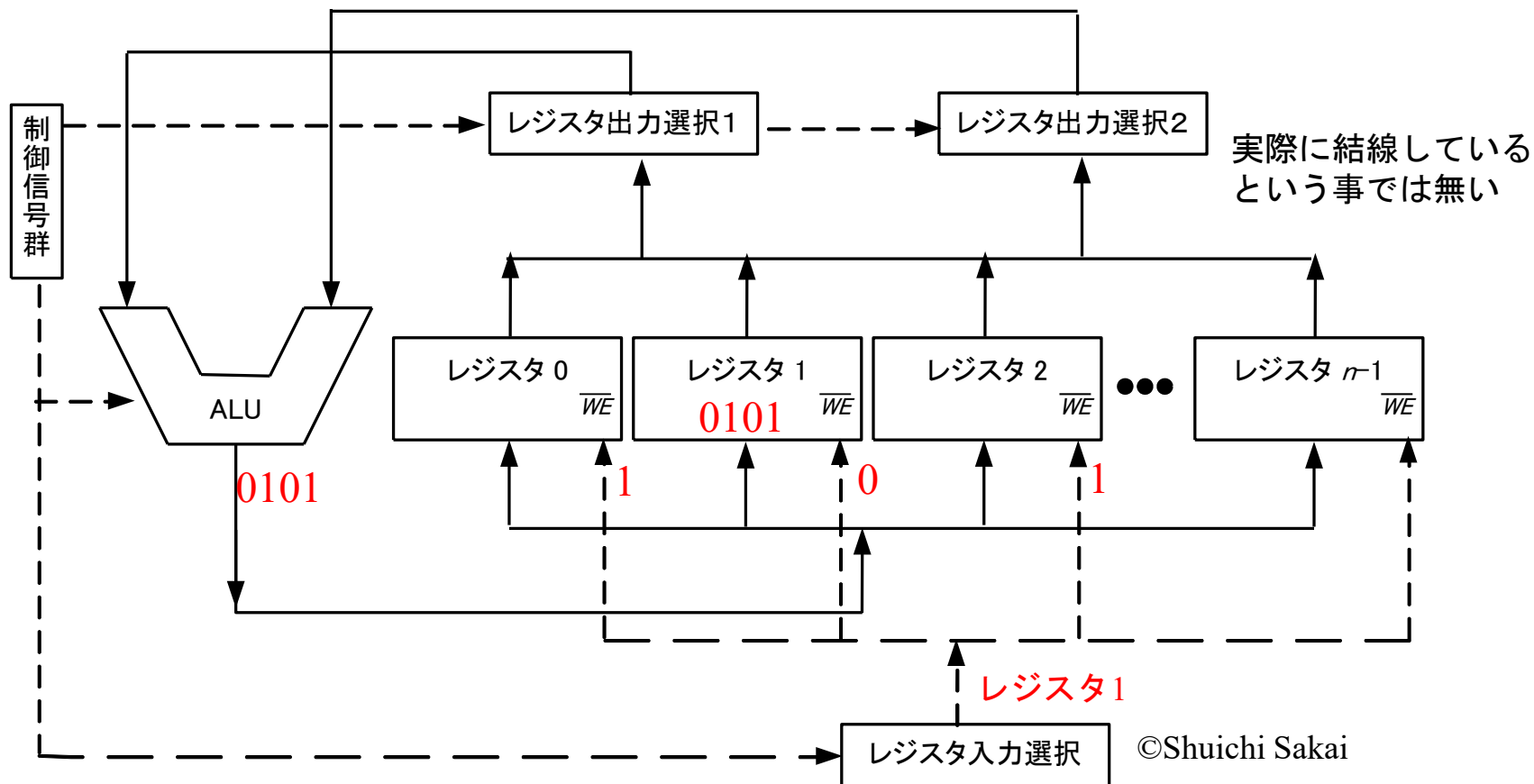
計算のサイクル(少し概念的)



計算のサイクル： レジスタ $\Rightarrow$ ALU $\Rightarrow$ レジスタ



計算のサイクル(少し概念的)



計算のサイクル： レジスタ ⇒ ALU ⇒ レジスタ