

# 情報理論及演習

2003 年 5 月 12 日

担当：池口 徹

埼玉大学 大学院 理工学研究科 情報数理科学専攻 助教授

Email : [tohru@ics.saitama-u.ac.jp](mailto:tohru@ics.saitama-u.ac.jp)

URL : <http://www.nls.ics.saitama-u.ac.jp/~tohru>

# 今日の講義の内容は？

コンピュータの心臓部のはなし

- トランジスタ
- メモリ
- マイクロチップ

# Chapter7

## How a Transistor Works

トランジスタとは?

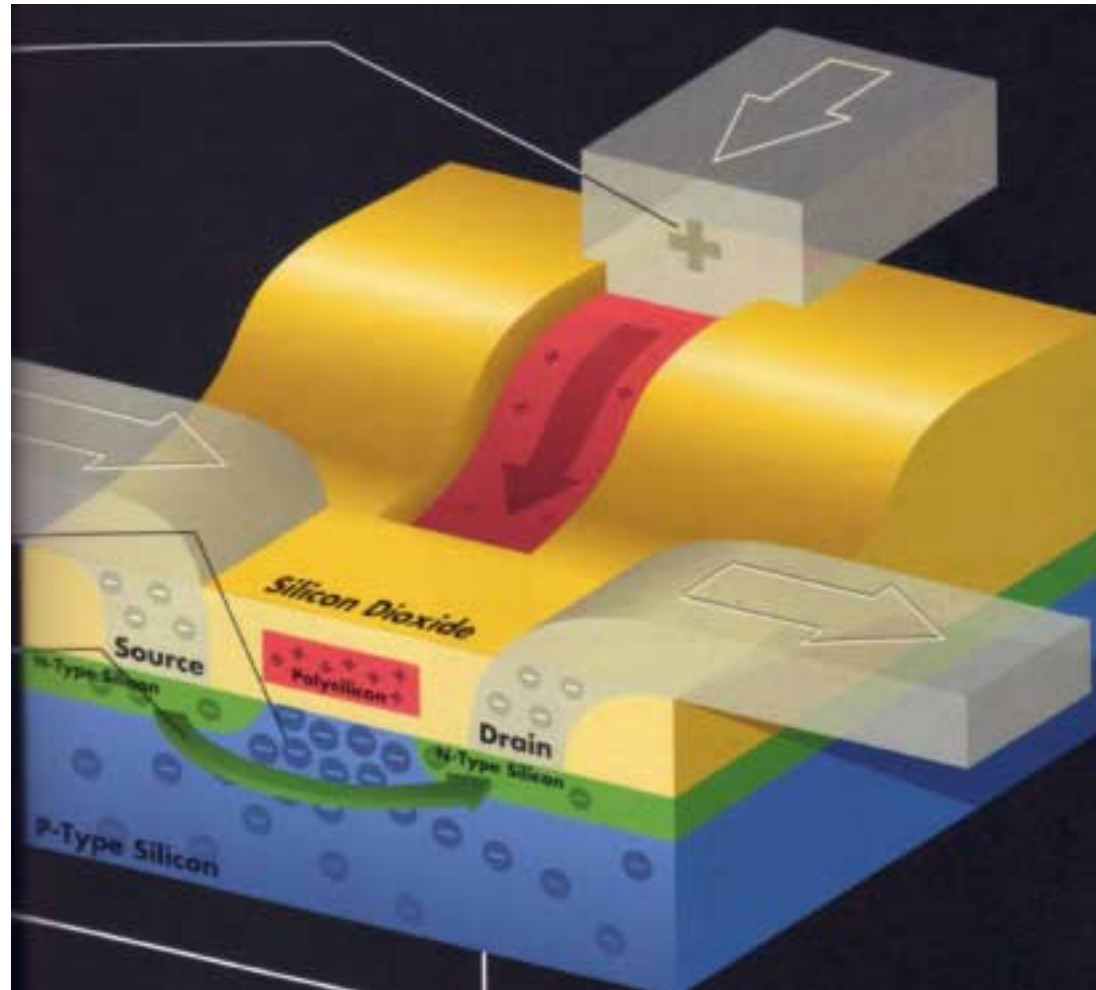
- 増幅機能を持った半導体 (Si) 素子
- 2 値状態 (on, off)  $\rightarrow$  0,1  $\rightarrow$  2 進数

10 進数	2 進数	10 進数	2 進数
0	0	6	110
1	1	7	111
2	10	8	1000
3	11	9	1001
4	100	10	1010
5	101	11	1011

# ビット bit

- 16 bits  $\rightarrow 2^{16} = 65,536$  (8088,80286)
- 32 bits  $\rightarrow 2^{32} = 4,294,967,296 = 42$  億 (80386,80486)
- ブール代数を実現
  - 1  $\rightarrow$  True (真)
  - 0  $\rightarrow$  False (偽)
- トランジスタの組み合わせにより, 様々な論理を実現

# トランジスタ

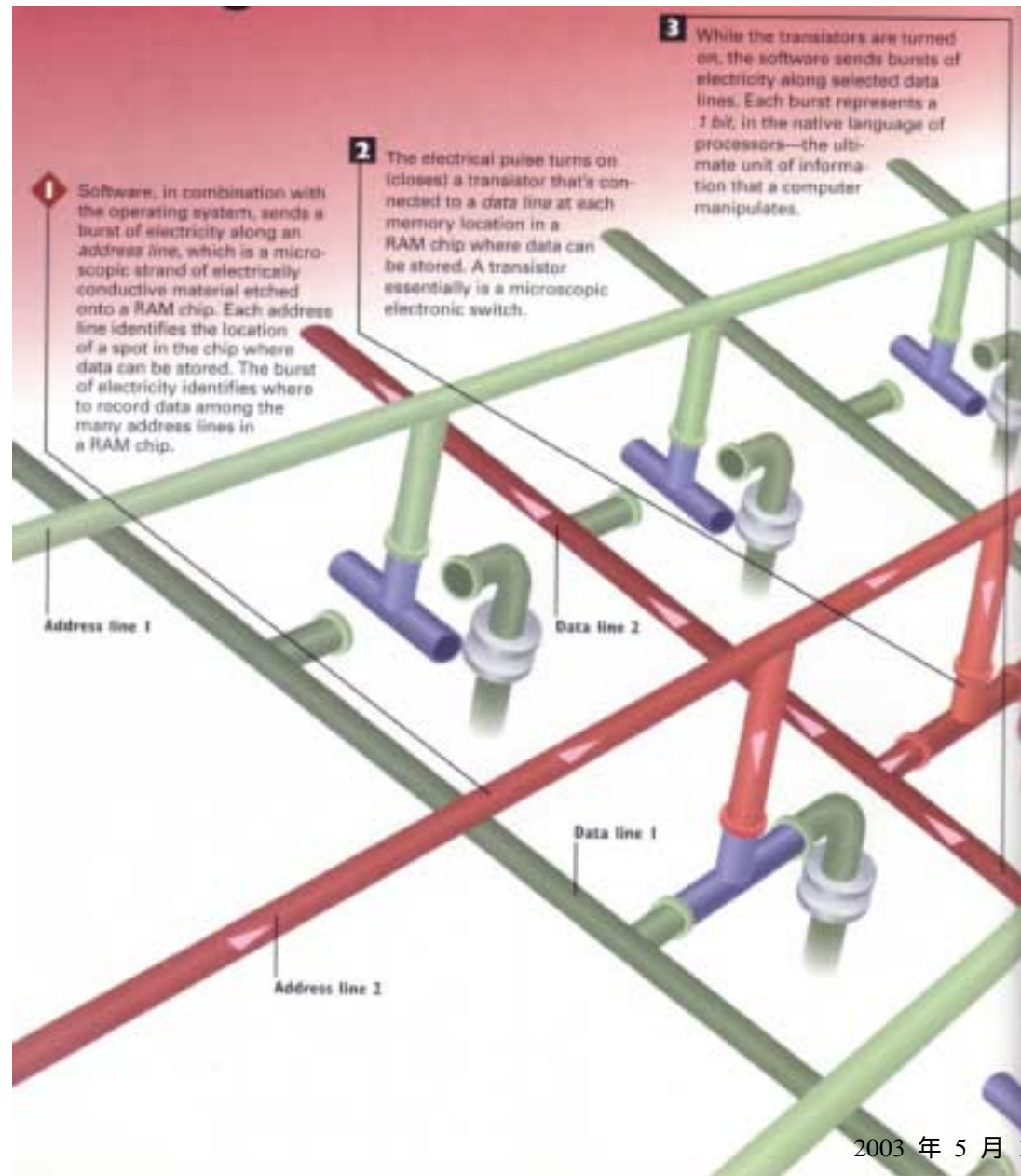


# Chapter 8

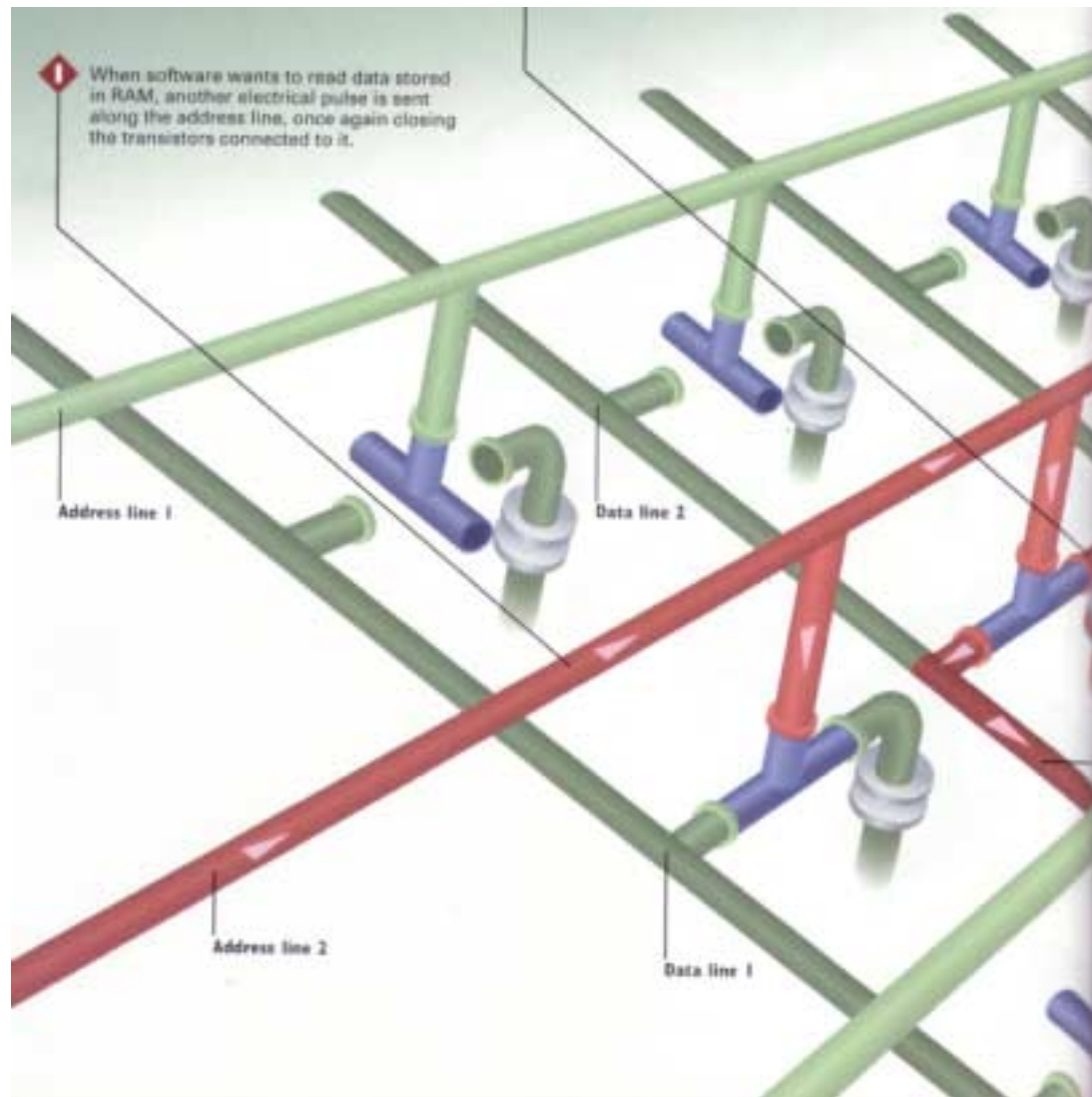
## How RAM Works

- Random Access Memory
- 様々な情報を貯めておくところ
- スイッチの集合
  - 開 → 0
  - 閉 → 1
- 電流によりトランジスタが ON → スイッチが閉じる
- これにより, コンデンサに電荷が溜る → 1 を表現

# RAMへのデータの書き込み



# RAMからのデータの読み出し





# RAMの種類

## ● DRAM

- 記憶内容を保持するためには常にデータの再書き込み (リフレッシュ操作) が必要 → 速度低下
- 高記録密度
- 記憶容量あたりの製造コストが安い
- メインメモリ

## ● SRAM

- リフレッシュ不要 → 高速
- キャッシュメモリ  
CPU とメインメモリ (DRAM) 間に配置

# RAMの種類

- Video RAM  
グラフィックス書込み，ディスプレイ表示
- Extended Data Out RAM  
データ出力のタイミングを改良 → 読み出し速度の向上
- Synchronous DRAM  
バスが一定周期のクロック信号に同期 → 高速化  
168 ピン DIMM
- Error Correcting Code メモリ  
データエラーをチェックし，自動修復
- Single In-Line Memory Module  
ピン数 30, 64, 72，片面一列，32 ビット，2 枚一組
- Dual In-Line Memory Module  
ピン数 168，両面並列，64 ビット

# Chapter 9

## How a Microprocessor Works

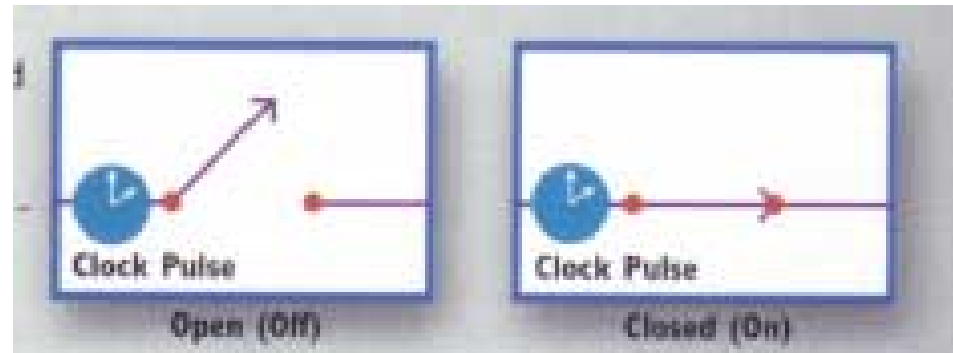
- マイクロプロセッサ  
演算処理等を行うチップ
  - Central Processing Unit
  - Coprocessor
  - 3D accelerator
- Intel
  - 8088, 80x86
  - Pentium
- Apple+IBM+Motorola
  - Power PC(RISC)

# 加算の仕組み

- 0,1 (1 ビット) による表現
- 基本的な演算
  1. NOT
  2. OR
  3. AND
  4. XOR
- 半加算器 (half adder)
- 全加算器 (full adder)

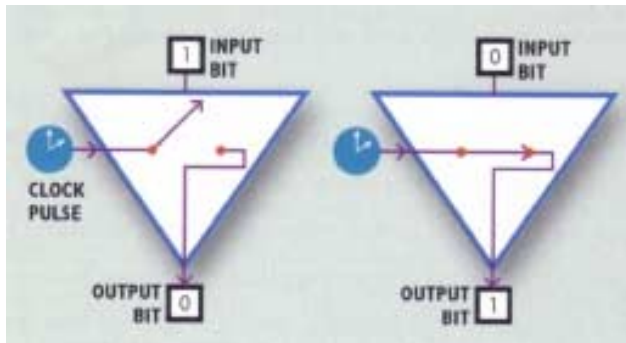
# データの表現

10 進数	2 進数
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010



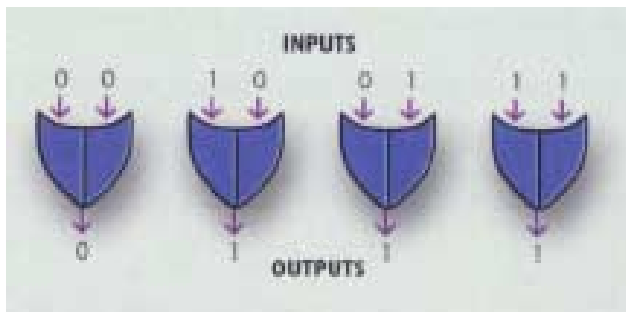
# 基本的な演算

## NOT(否定)



クロック	Input	Output
1	1	0
1	0	1

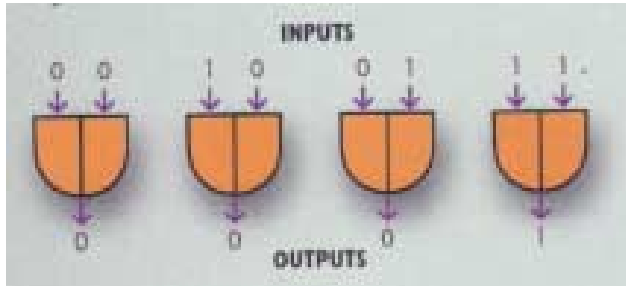
## OR (論理和)



1st Input	2nd Input	Output
0	0	0
1	0	1
0	1	1
1	1	1

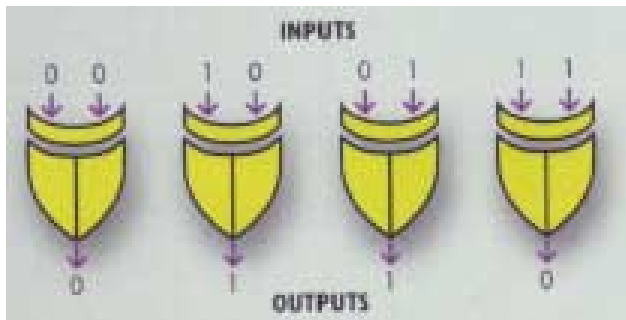
# 基本的な演算

## ● AND (論理積)



1st Input	2nd Input	Output
0	0	0
1	0	0
0	1	0
1	1	1

## ● XOR (排他的論理和)

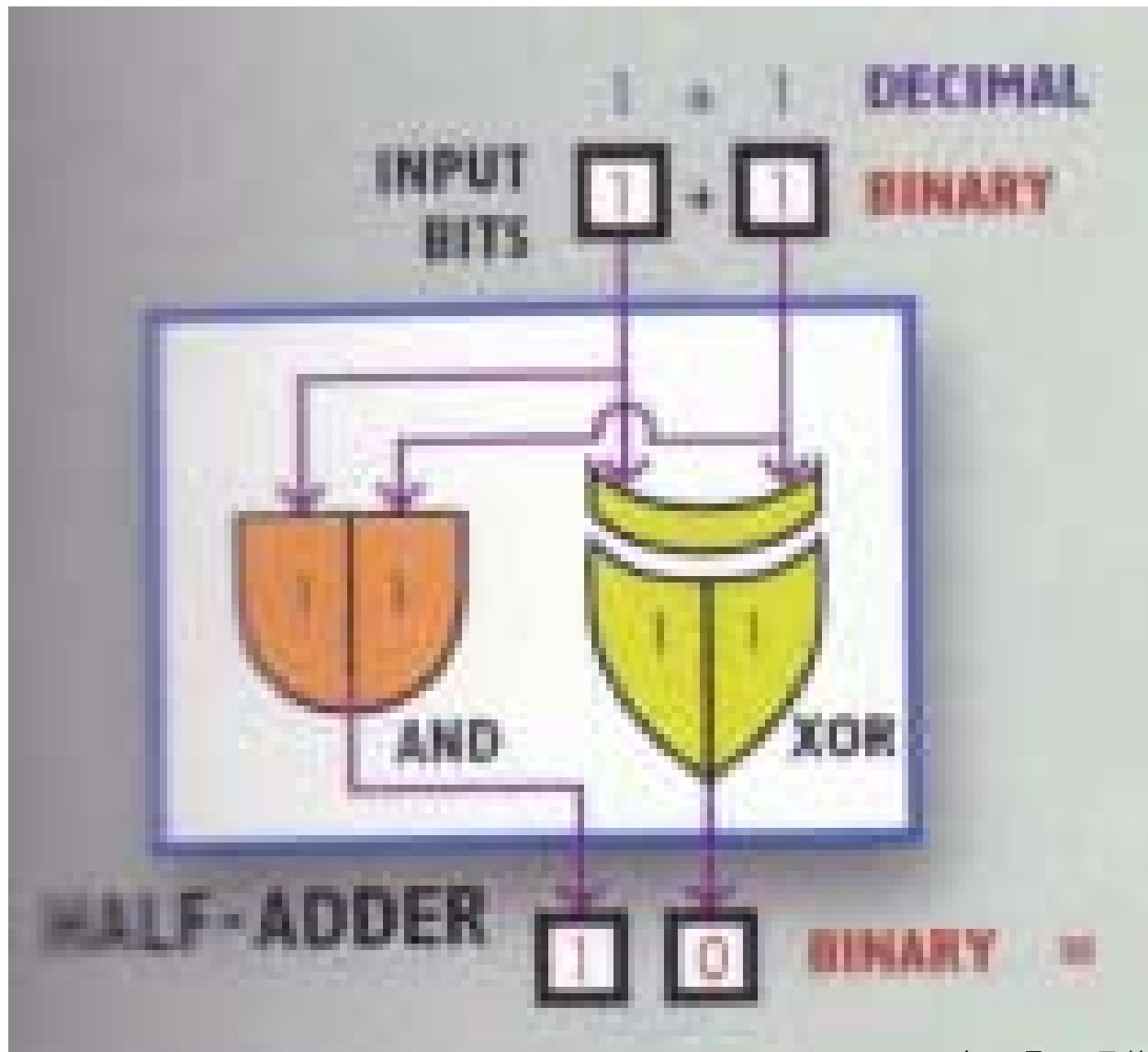


1st Input	2nd Input	Output
0	0	0
1	0	1
0	1	1
1	1	0

入力に差があると出力が 1

# 半加算器 (half-adder)

2 個の 1 桁の 2 進数の加算





# 全加算器 (full-adder)

