

---

# 計算機工学

東北大学 大学院情報科学研究科

鏡 慎吾

[swk\(at\)ic.is.tohoku.ac.jp](mailto:swk@ic.is.tohoku.ac.jp)

# 目標

- 計算機(コンピュータ)の構成と動作原理を学ぶ
- 何のため?
  - ソフトウェアの動き方をきっちりと理解する
    - 理解せずに書いて動かした(動いたように見える)プログラムは、すぐに動かなくなる
    - 高速化のためには計算機の動作原理の理解が必須
  - ハードウェアの構成要素・構造を知る
    - ごく一部の人には計算機を作る研究をする人
    - マイコン・FPGAなどでハードウェアを自作する人
    - 計算機を買う人
  - 計算機の動作や設計の背後にある「考え方」を学ぶ
    - 計算機以外にも応用可能な概念がたくさんある  
(例: 論理, 状態遷移)

## 例題 (おまけ)

ある人が狼, 羊, 牧草とともに旅をしていたところ, 川にさしかかった. 小さな舟を漕いで渡るしかない. 舟には, 漕ぎ手である人のほか, 狼, 羊, 牧草のいずれか高々1つしか乗せるスペースがない. ただし, 人がいないと狼は羊を食べてしまい, また羊は牧草を食べてしまう. 人, 狼, 羊, 牧草すべて無事に川を渡るにはどうすればよいか.

## 例題 (おまけ その2)

天国と地獄の分かれ道に門番が立っている. 門番は天国または地獄のどちらから派遣されているが, どちらかはわからない. 門番には「はい」または「いいえ」で答えることのできる質問を一つだけすることができる. ただし, 天国からきた門番は本当の答えを教えてくれるが, 地獄から来た門番は必ず嘘をつく. どのような質問をすればよいか.

# 予定

- (1) 04/07
- (2) 04/14
- (3) 04/21
- (4) 04/28
- (05/05 祝日)
- (5) 05/12
- (6) 05/19
- (7) 05/26
- (8) 06/02
- (9) 06/09
- (10) 06/14 (土)
- (11) 06/16
- (12) 06/23
- (13) 06/30
- (14) 07/07
- (15) 07/14 期末試験予定
- (07/21 祝日)
- (07/28 予備日)

- 講義はスライドと板書で進める. 必要に応じて資料を配布する
- 評価は期末試験により行う
- 参考書
  - パターソン, ヘネシー: コンピュータの構成と設計 — ハードウェアとソフトウェアのインタフェース, 日経BP社, 2011 (ほか多数. シラバスも参照)



いまや世界標準となった  
コンピュータ学の金字塔

コンピュータ界を牽引する世界中のリーダー達を育てた  
古典的教科書が並列処理の時代に先けて全面改訂

PEARSON  
ISBN 4-803-12101-8



並列処理の仕組みを基礎から  
身に付ける最良の教科書

マルチプロセッサと並列処理に関する知識に個人と企業  
両方から、目録を繰って買ったすべての人にとっても必須書に

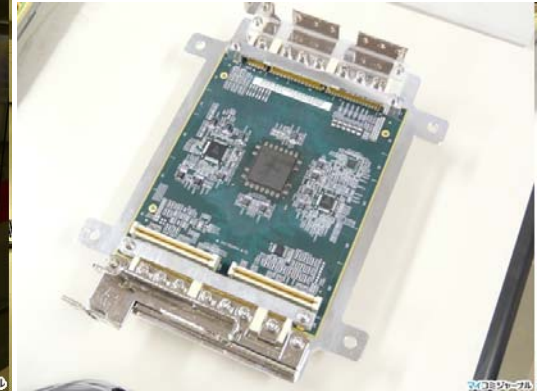
PEARSON 定価: 4,800円(税別)

---

# イントロダクション

# いろいろな計算機

いわゆる  
スーパーコンピュータ



<http://journal.mycom.co.jp/articles/2008/11/25/sx-9/>

いわゆる  
パーソナルコンピュータ



<http://www.bunkai.jp/note/panasonic/cf-t2.html>

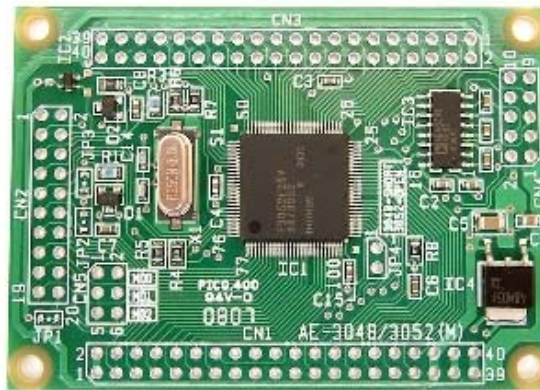
# いろいろな計算機

携帯電話  
（「スマートフォン」  
じゃなくても立派な  
コンピュータ）

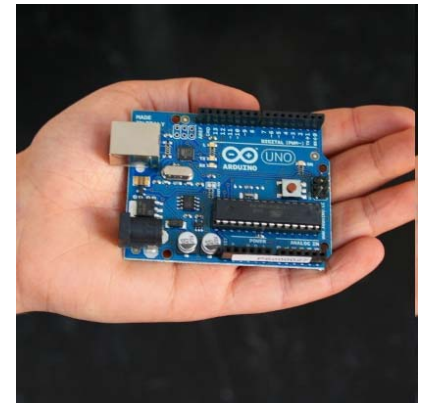


<http://itpro.nikkeibp.co.jp/article/COLUMN/20100708/350067/>

いわゆるマイコン  
（マイクロコントローラ）



<http://akizukidenshi.com/catalog/g/gK-00180/>



<http://www.arduino.cc/>

# いろいろな計算機

- スーパーコンピュータ, パーソナルコンピュータ, スマートフォンなどの「いかにもコンピュータ」なもの以外にも, 世の中にはコンピュータであふれている
- 自動車
  - 1台に数十個から百個程度のプロセッサ, ...
- 家電
  - テレビ, DVDプレイヤー, 炊飯器, ポット, エアコン, 掃除機, ...
- 通信機器
  - ルータ, スイッチ, 無線LANステーション, ...
- 社会インフラ
  - 銀行のオンラインシステム, 鉄道の運行システム, ...

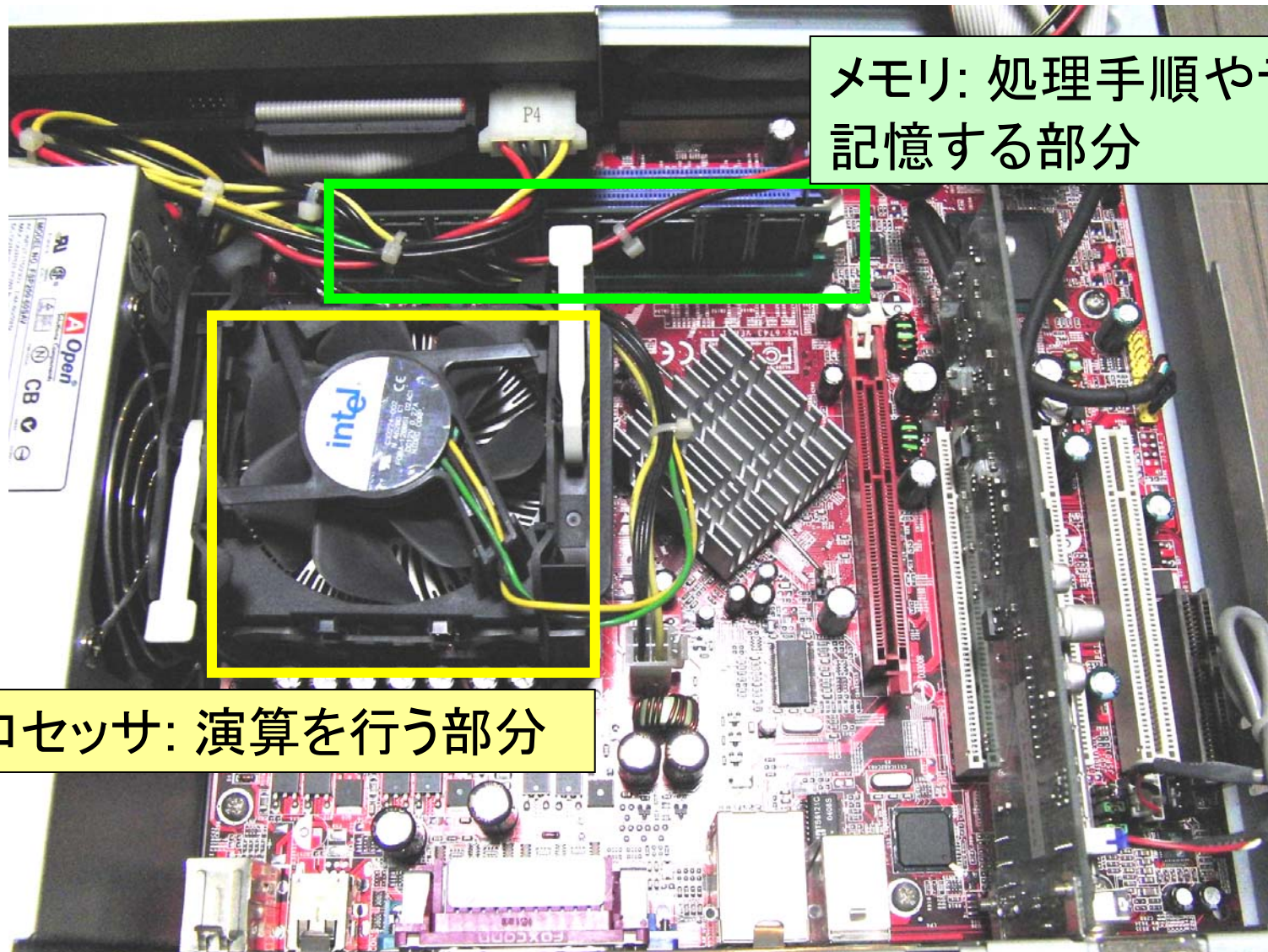


# パーソナルコンピュータ(PC)の中身



OS※5	Windows 8 Pro 64ビット		
CPU	<b>インテル® Core™ i5-3340M vPro™ プロセッサ</b> (動作周波数 2.70GHz、ターボ・ブースト2.0利用時は最大3.40GHz)	<b>インテル® Core™ i7-3540M vPro™ プロセッサ</b> (動作周波数 3.0GHz、ターボ・ブースト2.0利用時は最大3.70GHz)	<b>インテル® Core™ i5-3340M vPro™ プロセッサ</b> (動作周波数 2.70GHz、ターボ・ブースト2.0利用時は最大3.40GHz)
メモリー※6	標準4GB(最大8GB)		
SSD/HDD※7	SSD 128GB	HDD 640GB	
質量※8	標準バッテリーパック(L)装着時:約 <b>1.36kg</b> 軽量バッテリーパック(S)装着時:約 <b>1.15kg</b>	標準バッテリーパック(L)装着時:約 <b>1.40kg</b> 軽量バッテリーパック(S)装着時:約 <b>1.19kg</b>	標準バッテリーパック(L)装着時:約 <b>1.34kg</b> 軽量バッテリーパック(S)装着時:約 <b>1.13kg</b>
駆動時間※9	標準バッテリーパック(L)装着時:約 <b>19時間</b> 軽量バッテリーパック(S)装着時:約 <b>9.5時間</b>	標準バッテリーパック(L)装着時:約 <b>17.5時間</b> 軽量バッテリーパック(S)装着時:約 <b>8.5時間</b>	標準バッテリーパック(L)装着時:約 <b>18時間</b> 軽量バッテリーパック(S)装着時:約 <b>9時間</b>
液晶サイズ	12.1型HD+(1600×900ドット)		

# パーソナルコンピュータ(PC)の中身



メモリ: 処理手順やデータを記憶する部分

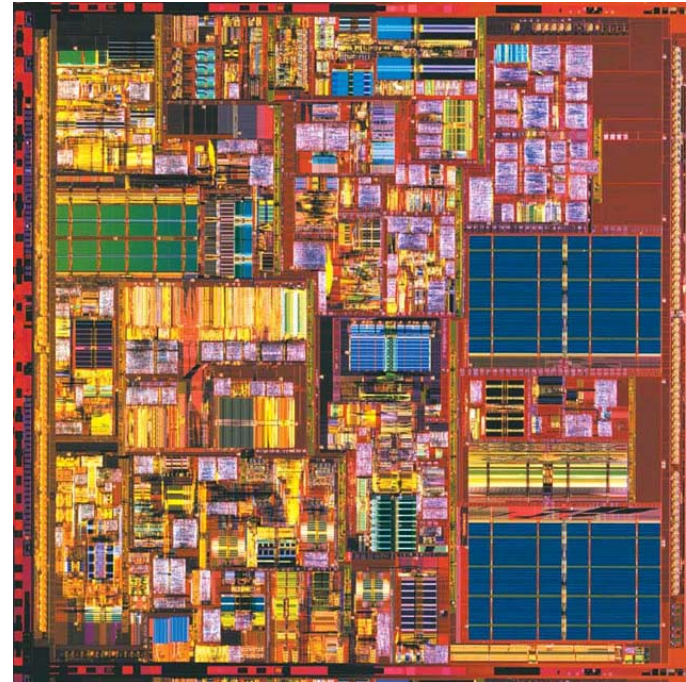
プロセッサ: 演算を行う部分

# プロセッサ

CPU (Central Processing Unit) または MPU (Microprocessor Unit)

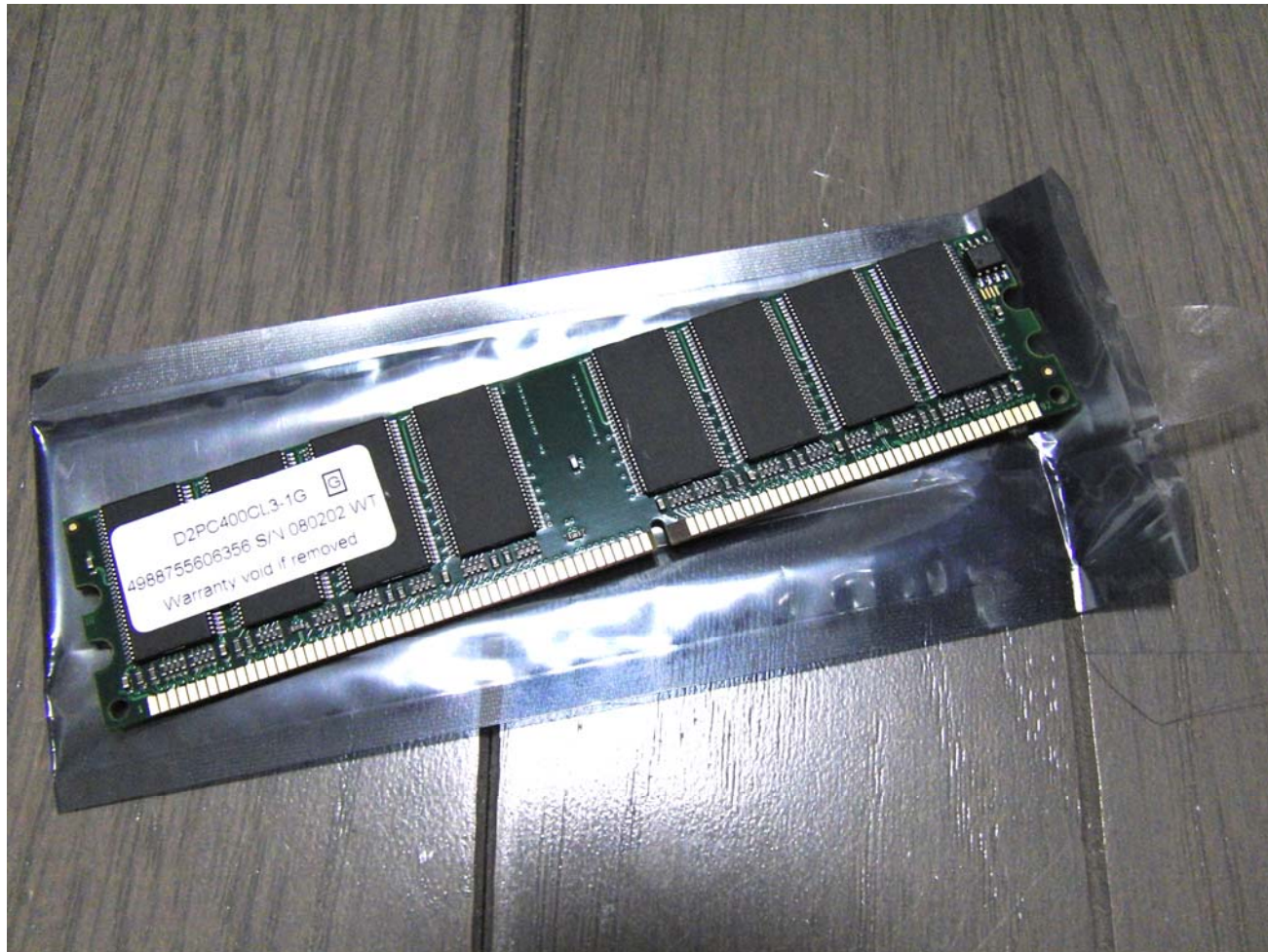


"Northwood" core Pentium 4 processor  
[http://ja.wikipedia.org/wiki/%E7%94%BB%E5%83%8F:Pentium4\\_northwood.png](http://ja.wikipedia.org/wiki/%E7%94%BB%E5%83%8F:Pentium4_northwood.png)

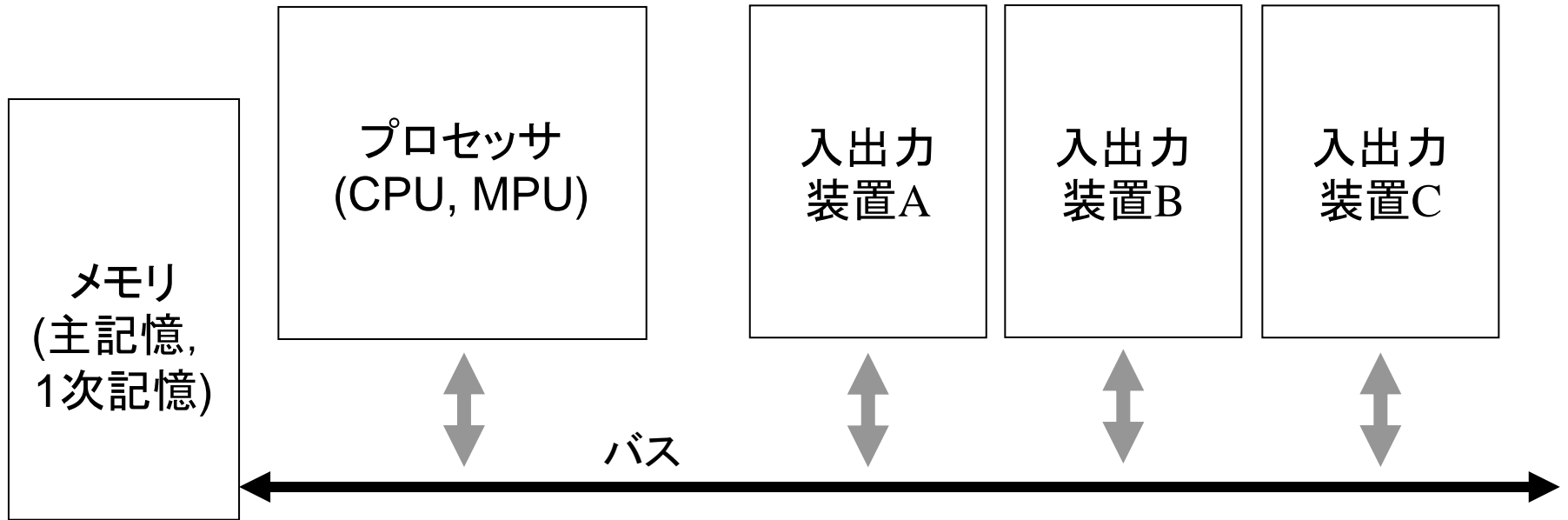


<http://journal.mycom.co.jp/articles/2005/02/22/p4/001.html>

# メモリ



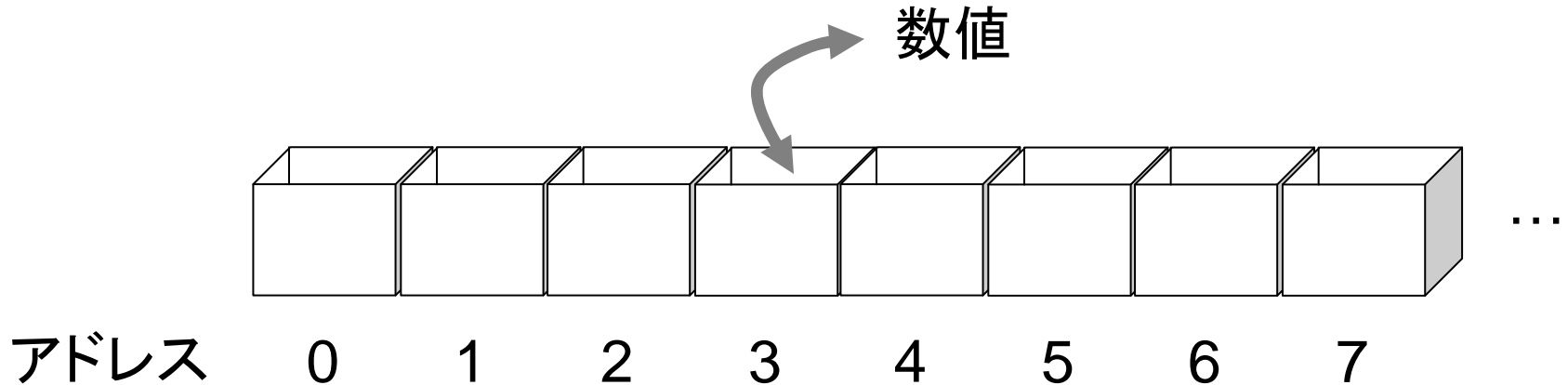
# 計算機の基本構成



## 入出力装置 (Input/Output, I/O) の例

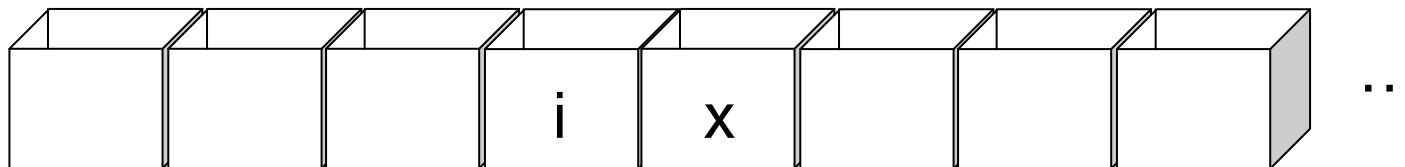
- 二次記憶 (外部記憶): ハードディスク, CD, DVD
- キーボード, マウス
- グラフィックス, ディスプレイ
- ネットワーク

# メモリの概念



- 数値を格納する区画が並んだもの
- 番号(**アドレス**)でインデックスづけられている
- アドレスを指定して、数値を書き込んだり、読み出したりする
- 書き込むことと読み出すことをあわせて「アクセスする」という
- **同じ大きさ**の区画に、**順番に**アドレスがついていることが重要

# プログラムの動作



アドレス 0 1 2 3 4 5 6 7

```
1: int i, x;
2:
3: i = 5;
4: x = 10;
5: i = i + 1;
6: if (i > 3) {
7:     x = x + i;
8: } else {
9:     x = x - i;
10: }
```

3: アドレス 3 に 5 を書き込む  
4: アドレス 4 に 10 を書き込む  
5: アドレス 3 から 5 を読み出す  
5 + 1 を計算して結果 6 を得る  
結果 6 をアドレス 3 に書き込む  
6: アドレス 3 から 6 を読み出す  
6 > 3 を計算して結果 true を得る  
true なので 7 行目に進んでよいと判断  
7: アドレス 4 から 10 を読み出す  
アドレス 3 から 6 を読み出す  
10 + 6 を計算して結果 16 を得る  
結果 16 をアドレス 4 に書き込む  
10 行目にジャンプする



# 数値の表現

「アドレス 3 に 5 を書き込む」:

3 や 5 などの数値はどのように表されるのか? → **2進数**

10 進数:

0, 1, 2, ... 9 の 10 個のシンボルを使って数を表す

1234 (decimal)

$$= 1 \times 10^3 + 2 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$$

同様に,

0, 1 の 2 個のシンボルを使って表したものを2進数と呼ぶ

1001 (binary)

$$= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

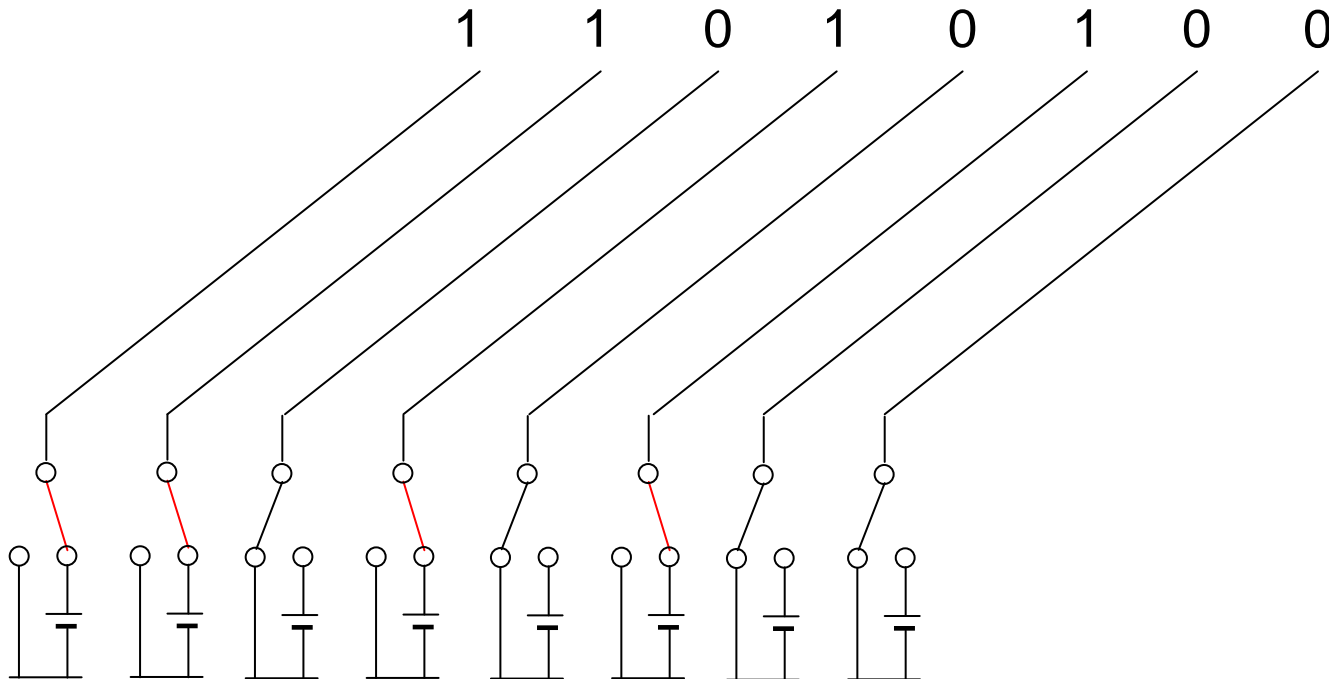
つまり計算機内の数値とは、0 と 1 が一定数並んだものである

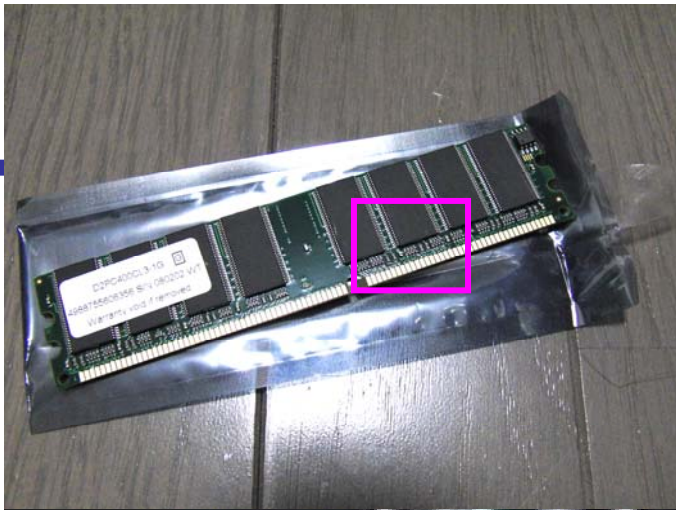
# なぜ2進数を使うのか

- 電圧が高い = 1
- 電圧が低い = 0

という2種類の物理状態をシンボルとして扱う

素子・配線を複数並べて、必要なサイズのデータを表現する





# 計算の実現

「5 + 1 を計算して結果 6 を得る」

計算はどのように行うのか? → 論理関数

$z = f(x, y)$  は,  $x$  も  $y$  も  $z$  も 8 桁の2進数だとすると,

$$z_1 = f_1(x_1, x_2, \dots, x_8, y_1, y_2, \dots, y_8)$$

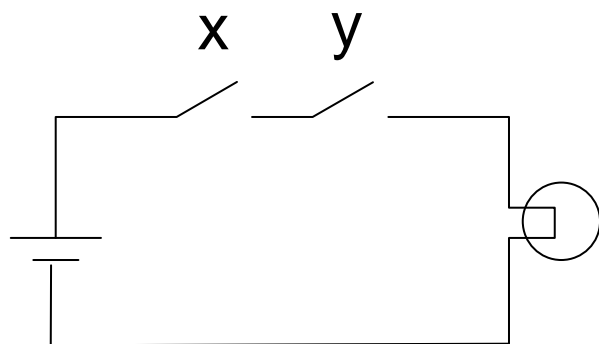
$$z_2 = f_2(x_1, x_2, \dots, x_8, y_1, y_2, \dots, y_8)$$

⋮

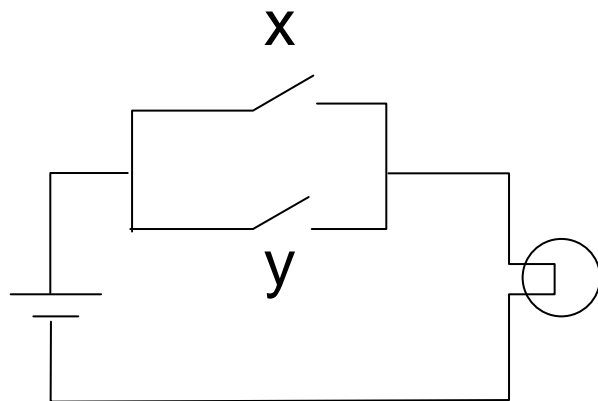
$$z_8 = f_8(x_1, x_2, \dots, x_8, y_1, y_2, \dots, y_8)$$

のような関数の組で表せる. ただし  $x_i, y_i, z_i \in \{0, 1\}$

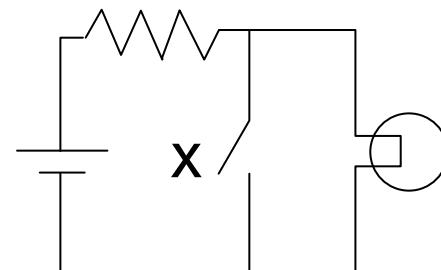
このような論理関数は必ず AND, OR, NOT の組合せで表せる



AND(x, y)



OR(x, y)



NOT(x)

AND, OR, NOT は、したがって、すべての演算（算術演算，論理演算，比較演算，...）はスイッチの組み合わせで実現できる

# プログラム

計算手順はどこからどのよ  
うに与えられるのか?

→ プログラム

- プログラム  
= 命令がメモリに並んだもの
- プロセッサは、命令をアドレス  
順に読み出して実行する
- 実行順を変える命令もある

3: アドレス 3 に 5 を書き込む  
4: アドレス 4 に 10 を書き込む  
5: アドレス 3 から 5 を読み出す  
5 + 1 を計算して結果 6 を得る  
結果 6 をアドレス 3 に書き込む  
6: アドレス 3 から 6 を読み出す  
6 > 3 を計算して結果 true を得る  
true なので 7行目に進んでよいと判断  
7: アドレス 4 から 10 を読み出す  
アドレス 3 から 6 を読み出す  
10 + 6 を計算して結果 16 を得る  
結果 16 をアドレス 4 に書き込む  
10行目にジャンプする

命令「t0 の内容に整数 14 を加算し, 結果を t1 に保存せよ」  
(t0 や t1 はプロセッサ内の記憶場所の名前)

00100001 00001001 00000000 00001110

addi \$t1, \$t0, 14

結局, 計算対象(データ)も, 計算手順(プログラム)も,  
メモリに 0 と 1 の羅列として保持されている  
(プログラム内蔵方式)

# プログラミング言語とコンパイラ

```
int i, x;  
i = 5; x = 10;  
i = i + 1;  
if (i > 3) {  
    x = x + i;  
} else {  
    x = x - i;  
}
```

ソースコード  
(C言語)

```
i に 5 を保存  
x に 10 を保存  
i + 1 を i に保存  
i と 3 を比較  
「>」でなければ L1 へ  
x + i を x に保存  
L2 へ  
L1: x - i を x に保存  
L2: 終了
```

アセンブリコード  
(アセンブリ言語)

```
00101100  
11010110  
11011101  
10110110  
00101001  
10010011  
...
```

オブジェクトコード  
(実際にメモリに入る  
数字の列)

コンパイラ

アセンブラ

さまざまな言語

コンピュータの種類によってさまざま



# 講義内容

- イン트로ダクション
- 算術演算 (2進数)
- 論理演算 (AND, OR, NOT)
- 計算機の動作
- ブール代数
- 組合せ論理回路
- 順序回路
- メモリ
- 計算機の構成

# コンピュータの歴史

大昔

指で数える

BC3000頃

abacus

バビロニアの算盤

(機械式計算機)

1640 頃

Pascal

Pascalene

機械式加減算機

1670 頃

Leibniz

Stepped Reckoner

機械式四則演算機

1832

Babbage

階差機関

機械式

Babbage

解析機関

実現せず

(電気式)

1904

Fleming

2極真空管発明

1906

de Forest

3極真空管発明

1938

Zuse

Z1

機械式+真空管

1943

Aiken

Harvard Mark I

機械式+リレー

# コンピュータの歴史

## (電子式へ)

1939	Atanasoff, Berry	ABC	真空管300本, コンデンサつきドラムによるメモリ
1943	Newman, Flowers	Colossus	真空管2500本. 暗号解読専用機
1945	von Neuman		プログラム内蔵型の概念を「発表」
1946	Mauchly, Eckert	ENIAC	真空管18000本
1948	Shockley		接合トランジスタ発明

## (プログラム内蔵型へ)

1948	William, Kilburn	SSEM (The Baby)	ブラウン管メモリの実験機
1949	Wilkes	EDSAC	真空管, 水銀遅延線
1951?	Mauchly, Eckert, von Neumann	EDVAC	真空管, 水銀遅延線
1951	Mauchly, Eckert	UNIVAC I	初の商用コンピュータ

# コンピュータの歴史

(汎用機・ミニコンピュータ)

1956

FORTRAN

1958 Kilby

集積回路(IC)発明

1958

IBM-7070

トランジスタ使用

1960

ALGOL, COBOL, LISP

1960

DEC PDP-1

1964

IBM-360

集積回路を使用

1965

BASIC

1965

ARPANET計画開始 → インターネットへ

1970

UNIX

1971

初の電子メール

# コンピュータの歴史

(PCの時代へ)

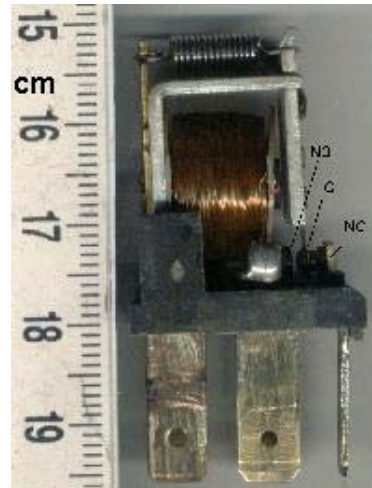
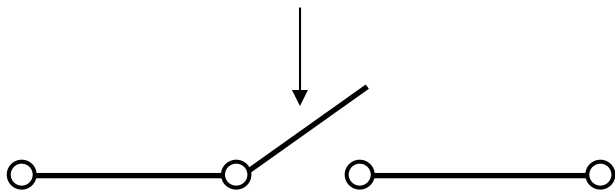
1971		マイクロプロセッサ4004開発
1972		C言語
1974		半導体メモリ(4kbit DRAM)商品化
1978		Intel 8086
1979		Motorola MC68000
1981	IBM PC	
1981		MS-DOS
1984	Apple Macintosh	
1985		Microsoft Windows
1989		World Wide Web

※年号は多少前後している可能性あり (稼動年, 発表年, ...)

# 計算機の構成方法の変遷

- スイッチをどのように実現するか
  - リレー, 真空管, トランジスタ (個別素子, IC, VLSI)
- 記憶素子をどのように実現するか
  - 原理上はスイッチで作れる (フリップフロップ). しかし高価
  - 他の手段: 水銀遅延線, 真空管, 磁気コア, 半導体メモリ
- 速く, 小さく, 壊れにくいデバイスの追求
- デバイスが変われば, それらの組み合わせ方 (= アーキテクチャ) も変わる

# スイッチの変遷



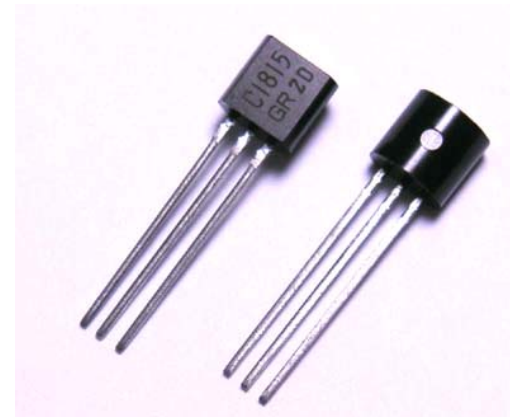
電磁リレー

<http://ja.wikipedia.org/wiki/%E7%94%BB%E5%83%8F:Relay.jpg>



真空管

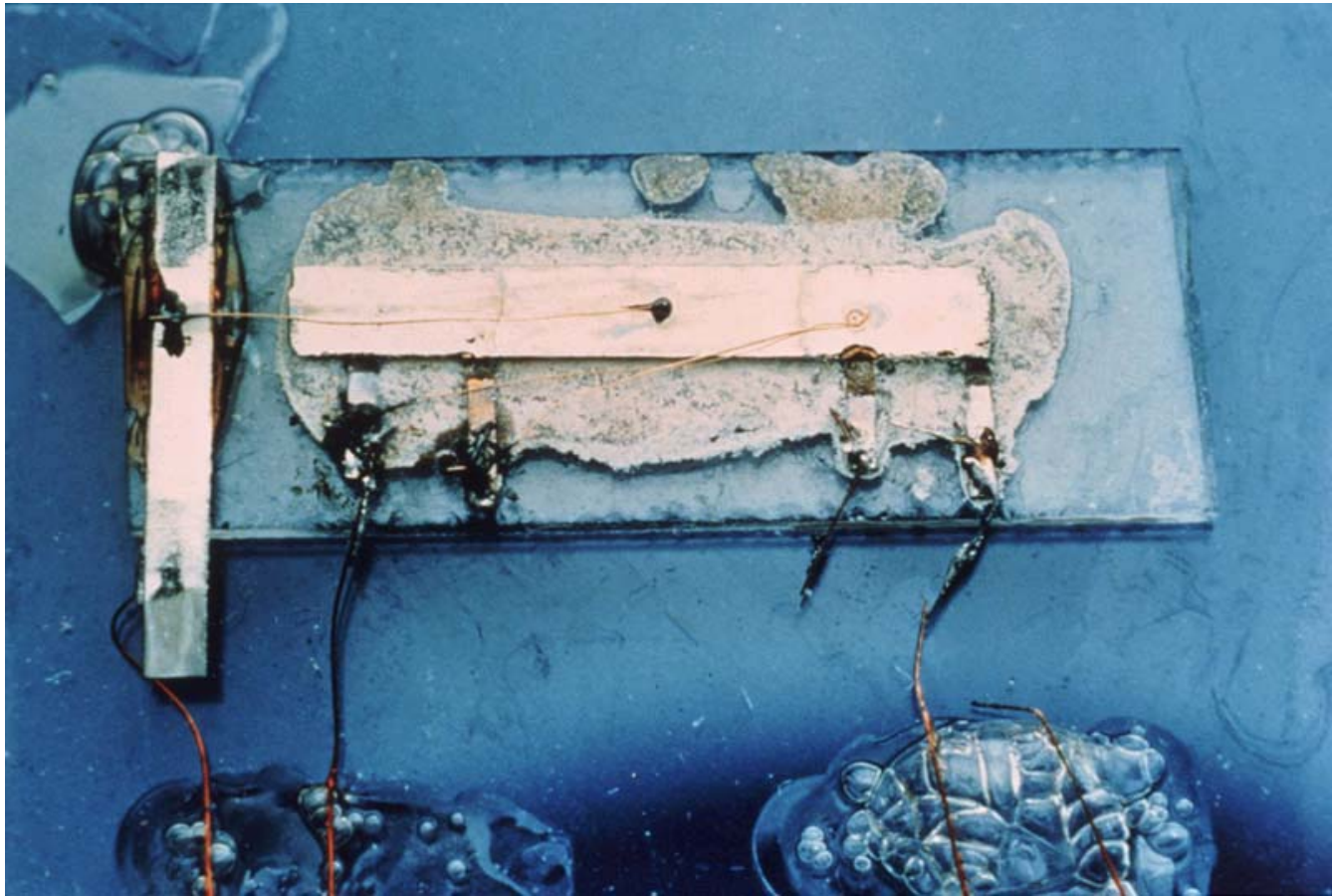
<http://ja.wikipedia.org/wiki/%E7%9C%9F%E7%A9%BA%E7%AE%A1>



トランジスタ

<http://ja.wikipedia.org/wiki/%E3%83%88%E3%83%A9%E3%83%B3%E3%82%B8%E3%82%B9%E3%82%BF>

# 最初の集積回路 (Integrated Circuit)

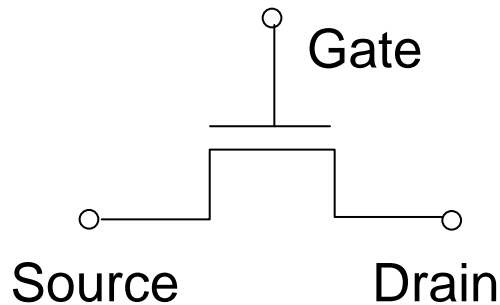


<http://www.ti.com/corp/docs/kilbyctr/downloadphotos.shtml>



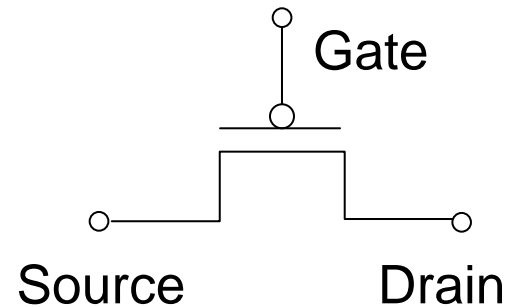
# MOS 型トランジスタ (Metal-Oxide-Semiconductor)

## NMOSトランジスタ



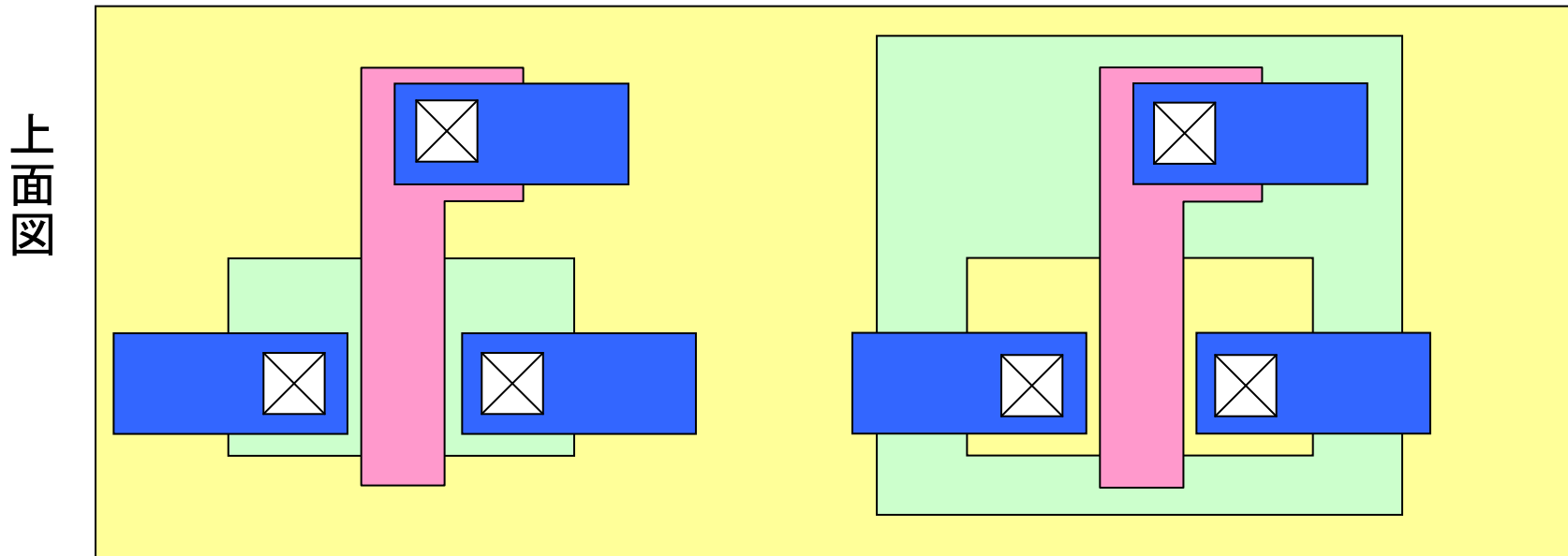
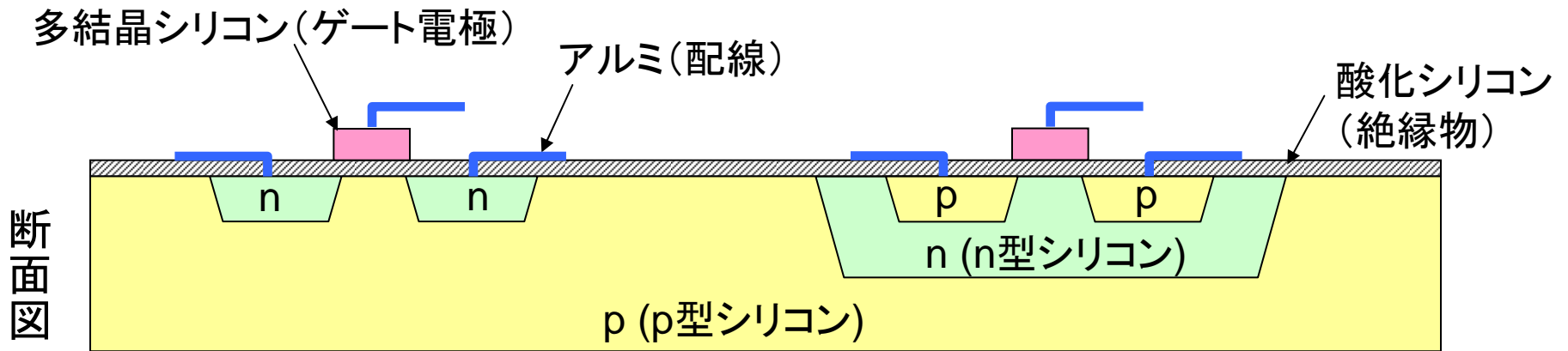
ゲートに電圧がかかっているとき (high, e.g. 3.3V), ソース-ドレイン間を電流が流れる

## PMOSトランジスタ

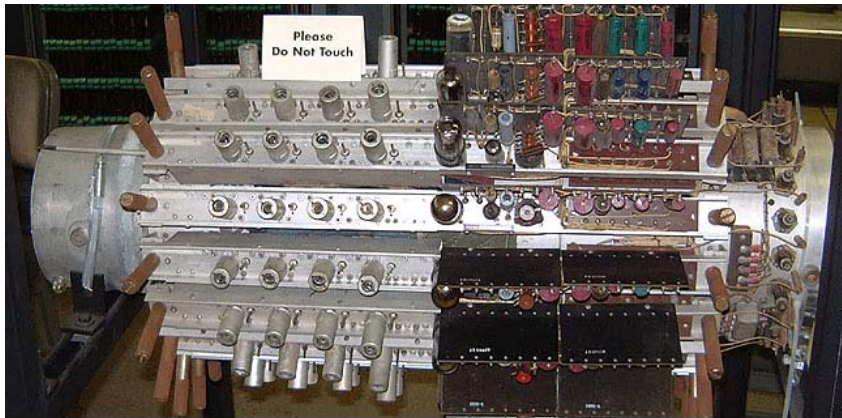


ゲートに電圧がかかっていないとき (low, 0V), ソース-ドレイン間を電流が流れる

# トランジスタの構造 (模式図)

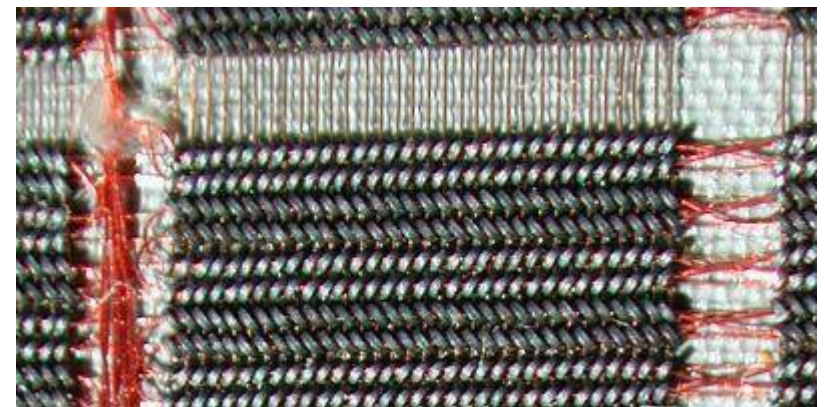
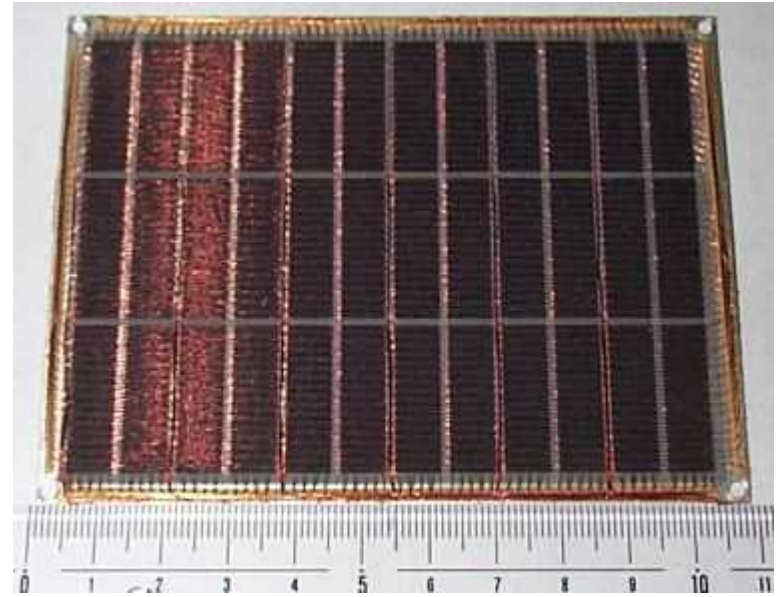


# 記憶素子の変遷



水銀遅延線

[http://ja.wikipedia.org/wiki/%E7%94%BB%E5%83%8F:Mercury\\_memory.jpg](http://ja.wikipedia.org/wiki/%E7%94%BB%E5%83%8F:Mercury_memory.jpg)



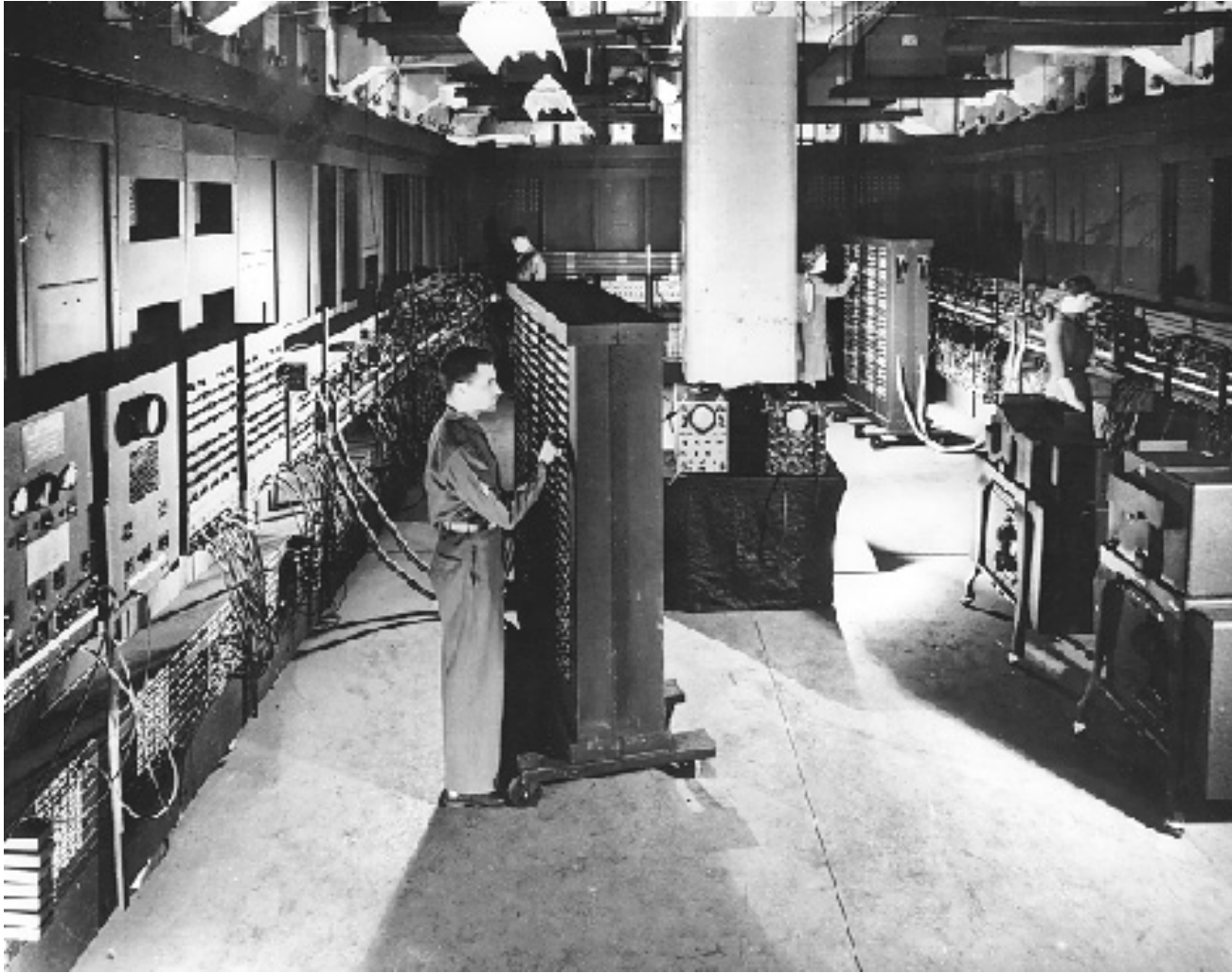
磁気コアメモリ

<http://www.st.rim.or.jp/~nkomatsu/pr/emicro/coremem.html>



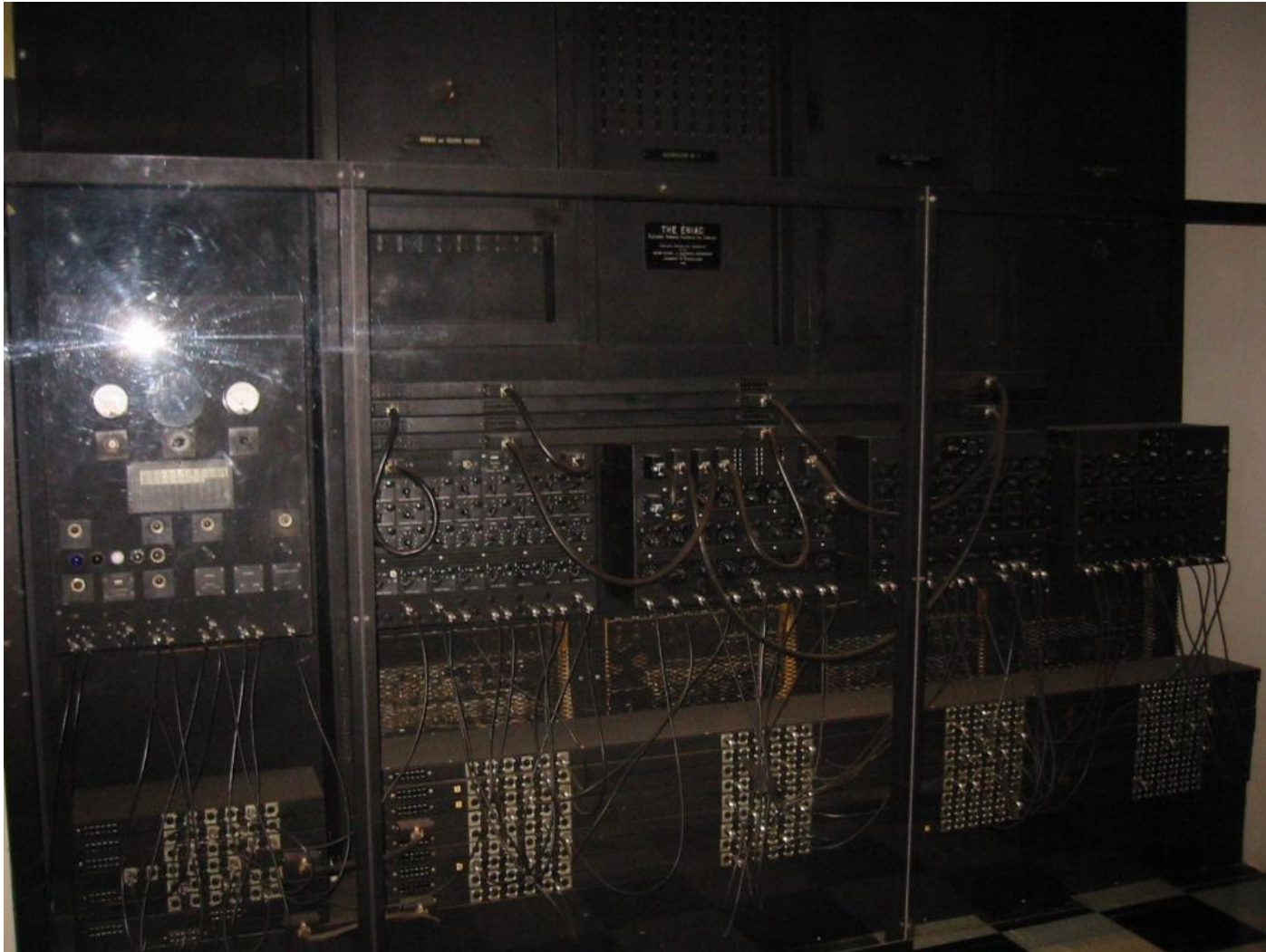
半導体メモリ

# ENIAC



<http://www.library.upenn.edu/exhibits/rbm/mauchly/jwm0-1.html>

# ENIAC



**(National Museum of American History, Washington, D.C.)**

# ENIAC



このパネルにケーブルを挿してプログラミングした  
(プログラム内蔵型ではなかった)

**(American History Museum)**

# UNIVAC



[http://news.com.com/1952+UNIVAC/2009-1006\\_3-6038974-2.html](http://news.com.com/1952+UNIVAC/2009-1006_3-6038974-2.html)

# IBM System/360

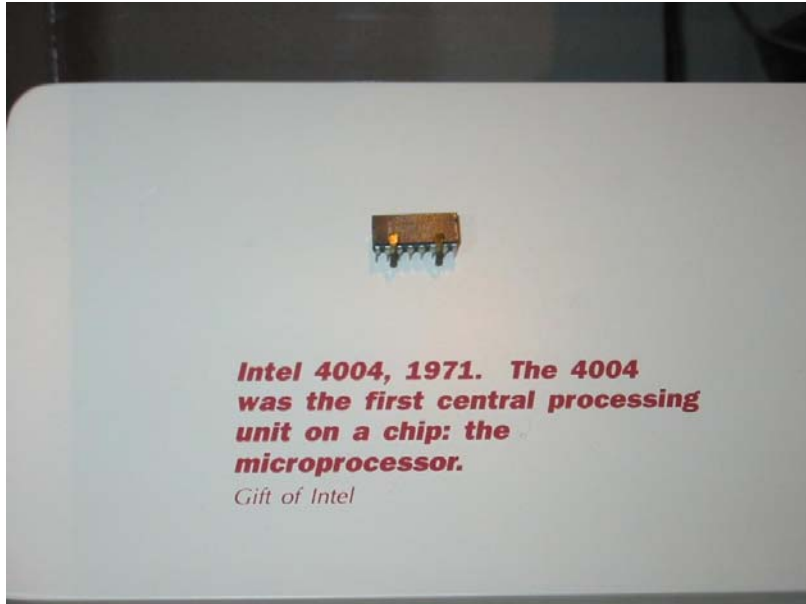


[http://upload.wikimedia.org/wikipedia/commons/8/8d/Bundesarchiv\\_B\\_145\\_Bild-F038812-0014%2C\\_Wolfsburg%2C\\_VW\\_Autowerk.jpg](http://upload.wikimedia.org/wikipedia/commons/8/8d/Bundesarchiv_B_145_Bild-F038812-0014%2C_Wolfsburg%2C_VW_Autowerk.jpg)

「コンピュータアーキテクチャ」という概念をおそらく最初に明確に導入した商用計算機. オペレーティングシステム(OS)を最初に導入した商用計算機でもある



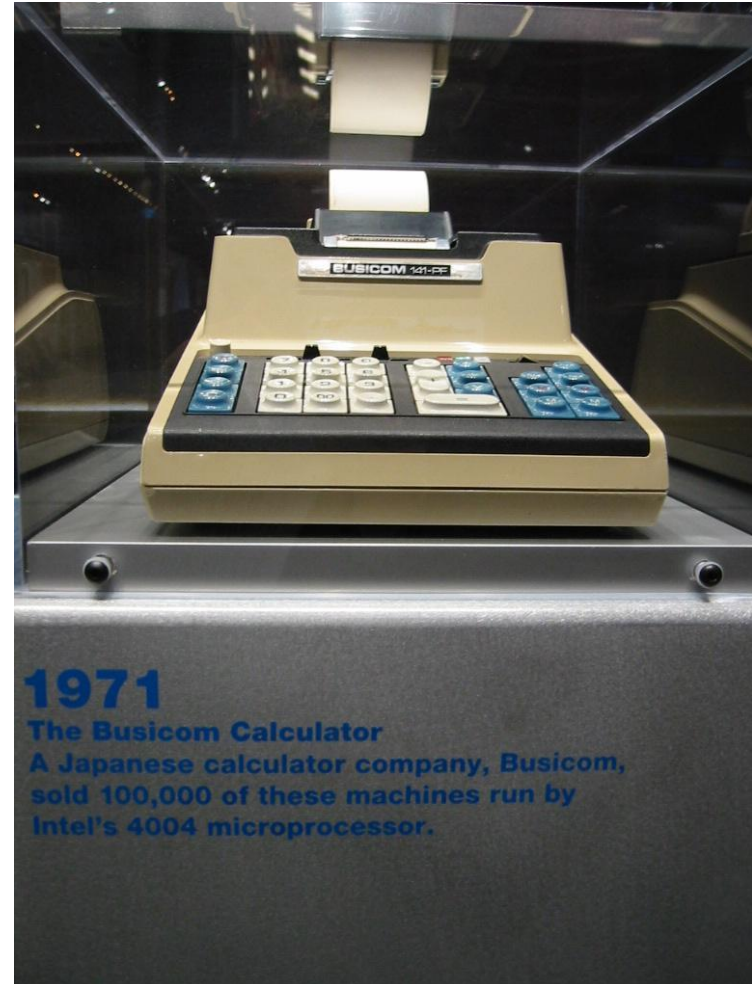
# intel 4004



**Intel 4004, 1971. The 4004 was the first central processing unit on a chip: the microprocessor.**

*Gift of Intel*

(American History Museum)



**1971**

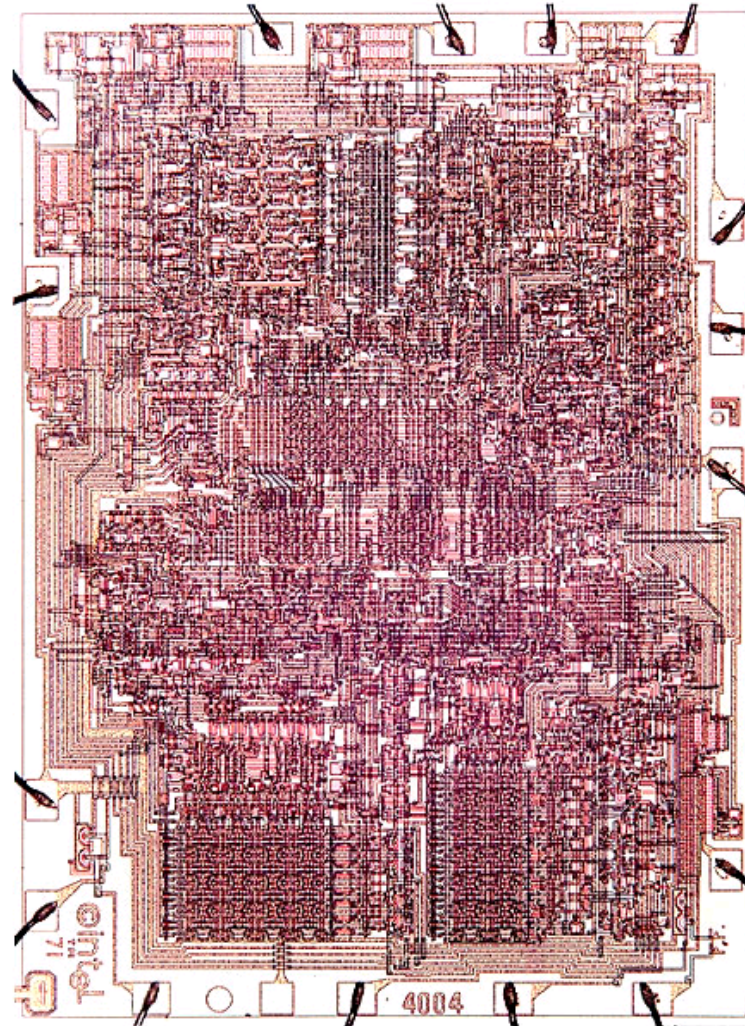
**The Busicom Calculator**

**A Japanese calculator company, Busicom, sold 100,000 of these machines run by Intel's 4004 microprocessor.**

Busicom 141-PF

(Intel Museum, Santa Clara)

# intel 4004



[http://news.com.com/1971+Intel+4004+processor/2009-1006\\_3-6038974-3.html](http://news.com.com/1971+Intel+4004+processor/2009-1006_3-6038974-3.html)

# 現在のPCやWSの原型



L to R: Macintosh, Sun 1, IBM PC  
(American History Museum)

# Intel 386

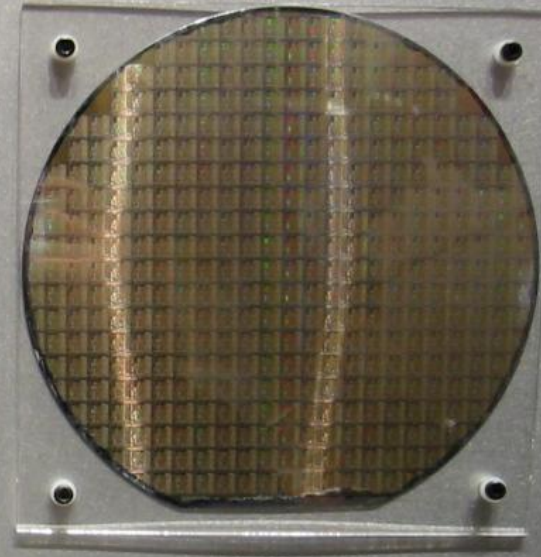
**1985**

## The Intel386™ Powers High-End Computers

This addition to Intel's growing family of microprocessors packed in 275,000 transistors, could process 5 million instructions per second, and could run all the operating systems popular at the time, including Windows\*. It could also "multitask," or run multiple programs at the same time.

In addition to its new architectural features, the Intel386 processor was built using a new multi layer, low-power, high-speed CHMOS process. With so much riding on new technologies, Intel386 chip designers were especially excited to find that the very first units that came off the manufacturing line worked.

*"The best time that I had at Intel*



(Intel Museum)

# Intel Pentium

sound, photographs, and video.



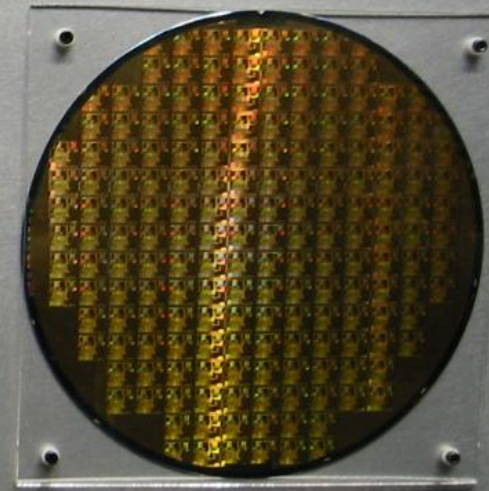
**1993**

## The Pentium® Brand Is Born

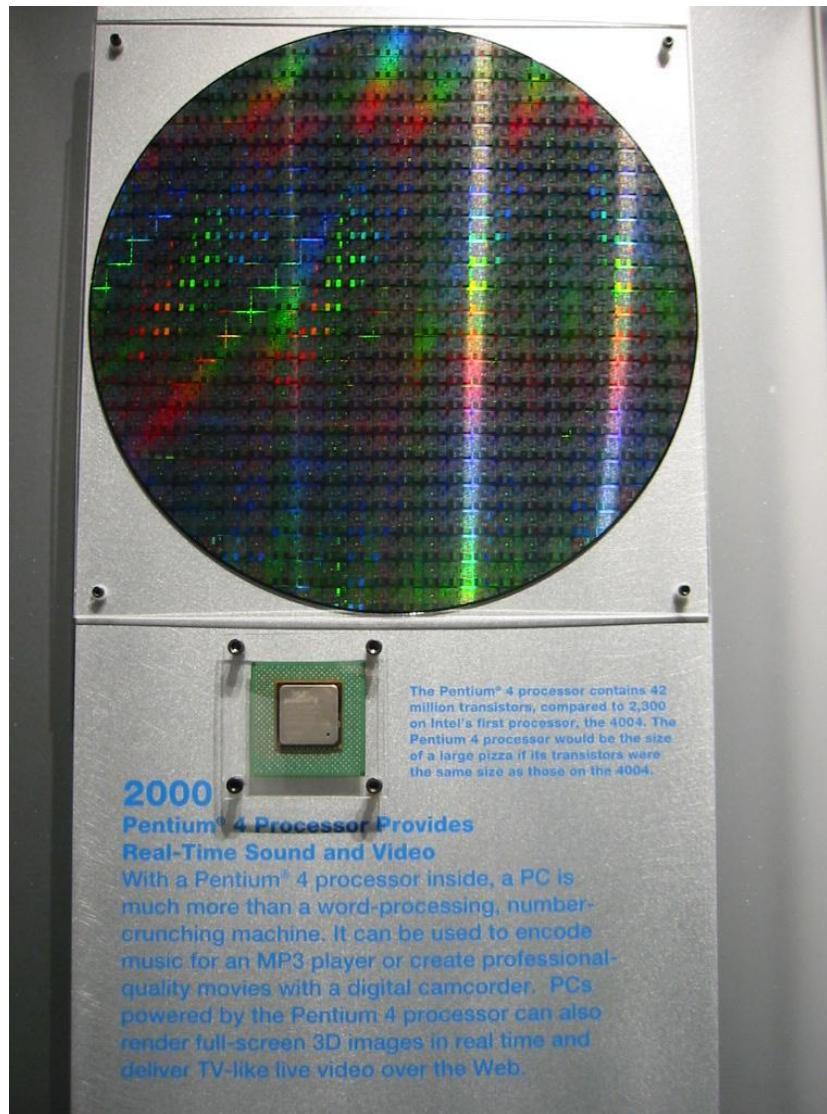
With 3.1 million transistors, the Pentium® processor brought faster performance, better graphics, and real-time speech and video to personal computers. Because it was fully compatible with earlier Intel microprocessors, the chip's expected name was Intel586. However, Intel cannot prevent imitators from using similar numbers to name their products, so the company decided to use a name it could protect with a trademark. "Pente" means five—a reminder that the chip was Intel's fifth-generation microprocessor.



Each Pentium chip from this 8-inch wafer is capable of executing more than 100 million instructions per second.



# Intel Pentium 4

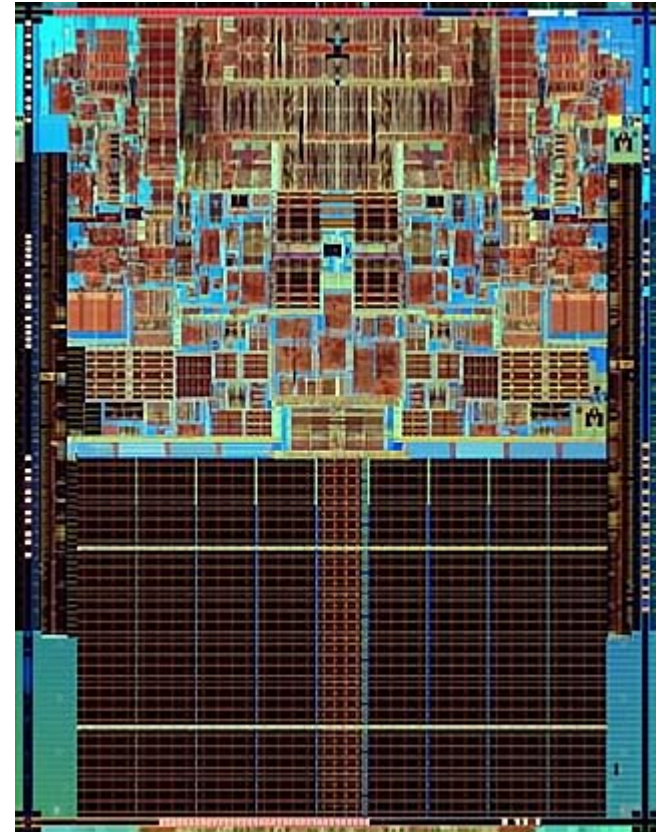


(Intel Museum)

# Intel Core 2

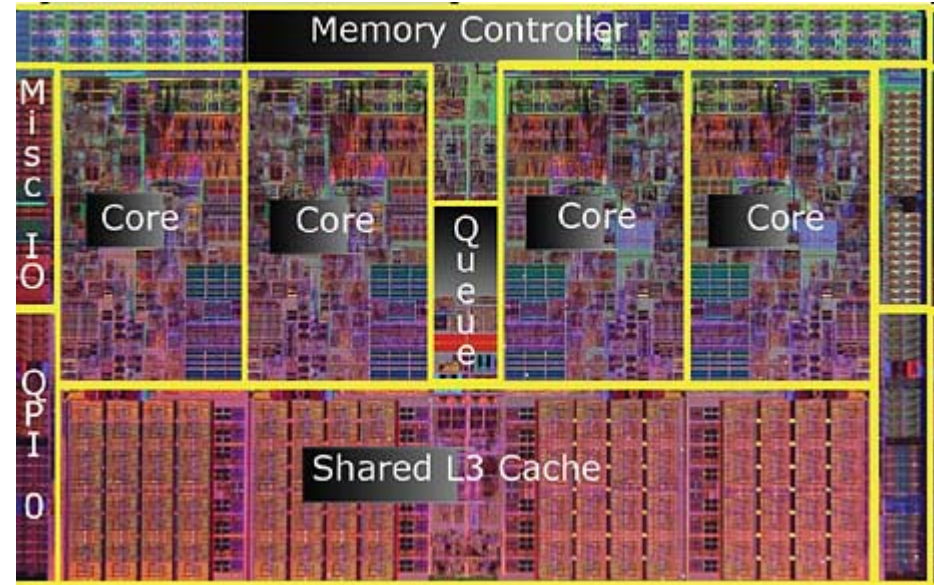


<http://ja.wikipedia.org/wiki/%E7%94%BB%E5%83%8F:IntelCore2DuoE6600.jpg>



[http://ja.wikipedia.org/wiki/%E7%94%BB%E5%83%8F:Conroe\\_die.png](http://ja.wikipedia.org/wiki/%E7%94%BB%E5%83%8F:Conroe_die.png)

# Intel Core i7



[http://ja.wikipedia.org/wiki/Intel\\_Core\\_i7](http://ja.wikipedia.org/wiki/Intel_Core_i7)

<http://www.atmarkit.co.jp/fsys/zunouhoudan/102zunou/corei7.html>