

東北大学 工学部 機械知能・航空工学科
2020年度 クラス C D

情報科学基礎 I

1. 序論

(教科書1.1節, 6.1節, 11章)

大学院情報科学研究科

鏡 慎吾

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
 - 計算機を作る人・分析する人
 - マイコン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)

例題の例

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。

(「ブール代数と論理回路」で扱う予定)

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。

(「順序回路の設計と応用」で扱う予定)

教科書・参考書

教科書

鏡, 佐野, 滝沢, 岡谷, 小林:
コンピュータ工学入門,
コロナ社, 2015

付録・章末問題解答はウェブで配布:

<http://www.coronasha.co.jp/np/isbn/9784339024920/>



参考書

パターンソン, 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, 2014



■ このスレッドは過去ログ倉庫に格納されています

自分の本買わせる教授www

- 1 : 風吹けば名無し@\(^o^)/ : 2015/04/07(火) 13:39:11
死ね
- 2 : 風吹けば名無し@\(^o^)/ : 2015/04/07(火) 13:39:21
わかる
死ね
- 3 : 風吹けば名無し@\(^o^)/ : 2015/04/07(火) 13:39:51
新入生あるある

<http://tomcat.2ch.sc/test/read.cgi/livejupiter/1428381553/>

講義内容

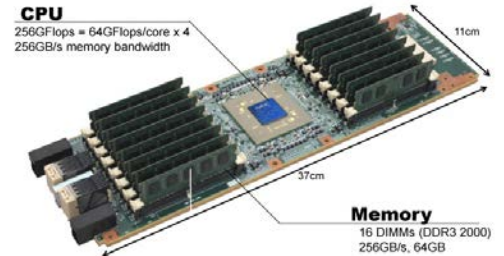
- 序論
- 算術演算 (2進数)
- 論理演算 (AND, OR, NOT)
- 計算機の構成と動作
- ブール代数
- 組合せ論理回路
- 順序回路
- メモリ
- 入出力・OS
- コンパイラ
- 浮動小数点演算
- ネットワーク

いろいろな計算機

いわゆる
スーパーコンピュータ



Node Packaging



Rated power consumption = 469W

18 © NEC Corporation, 2014 / HOTCHIPS26 SX-ACE Empowered by Innovation NEC

<http://www.tohoku.ac.jp/japanese/2015/03/news20150309-01.html>
<http://pc.watch.impress.co.jp/docs/column/kaigai/662027.html>

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)

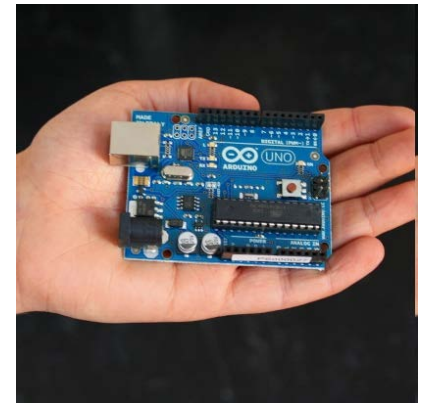


<http://itpro.nikkeibp.co.jp/article/COLUMN/20100708/350067/>

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

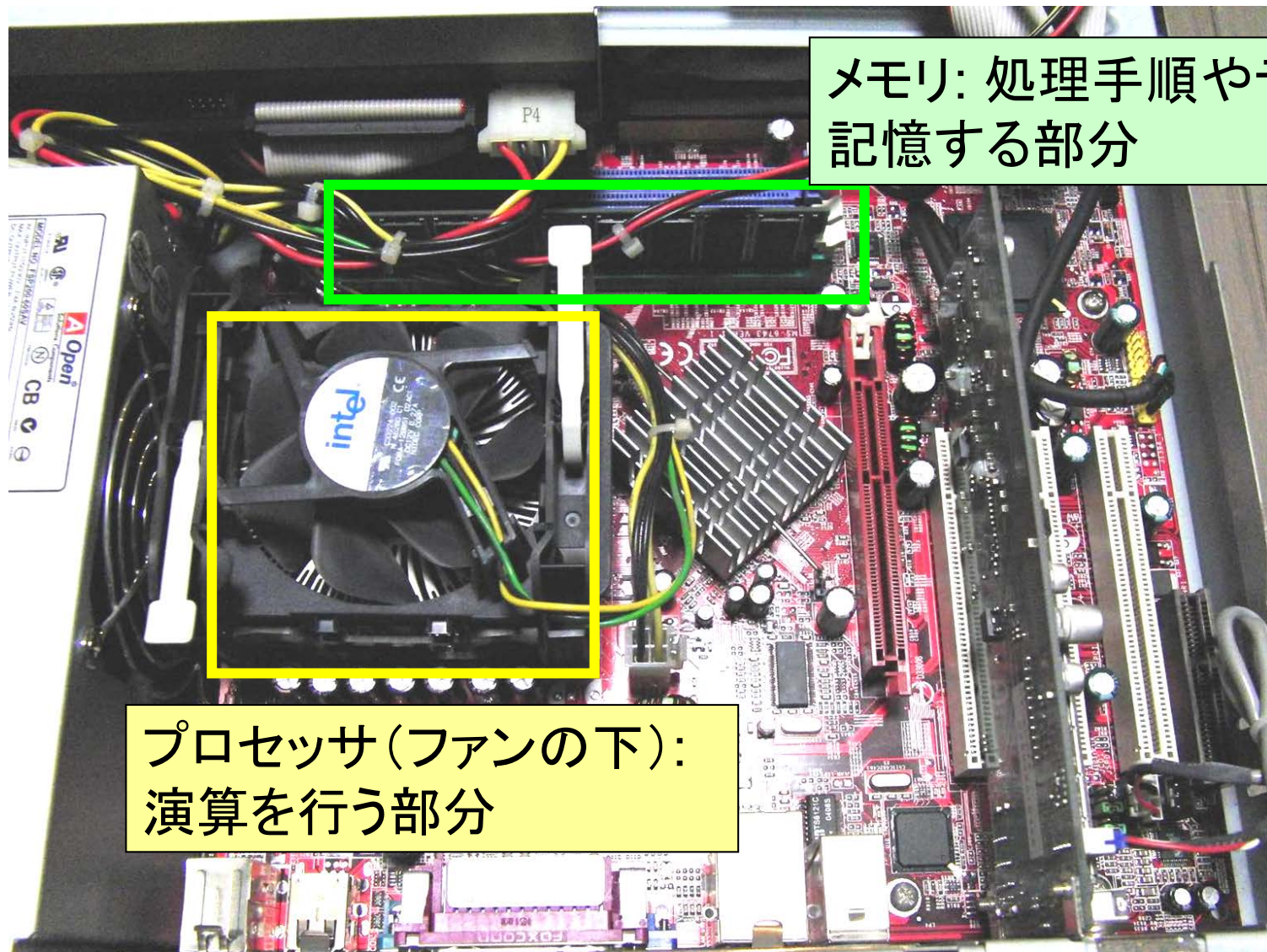
- スーパーコンピュータ, パーソナルコンピュータ, スマートフォンなどの「いかにもコンピュータ」なもの以外にも, 世の中にはコンピュータであふれている
- **自動車**
 - 1台に数十個から百個程度のプロセッサ, ...
- **家電**
 - テレビ, DVDプレイヤー, 炊飯器, ポット, エアコン, 掃除機, ...
- **通信機器**
 - ルータ, スイッチ, 無線LANステーション, ...
- **社会インフラ**
 - 銀行のオンラインシステム, 鉄道の運行システム, ...

構成要素

パーソナルコンピュータ(PC)の中身



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(ファンの下):
演算を行う部分

お届け目安	構成により異なります	構成により異なります	構成により異なります
型名	構成により異なります	構成により異なります	構成により異なります
OS	<ul style="list-style-type: none"> ・Windows 10 Home 64ビット版 ・Windows 10 Pro 64ビット版 	<ul style="list-style-type: none"> ・Windows 10 Home 64ビット版 ・Windows 10 Pro 64ビット版 	<ul style="list-style-type: none"> ・Windows 10 Home 64ビット版 ・Windows 10 Pro 64ビット版
CPU	<ul style="list-style-type: none"> ・インテル® Core™ i5-8265U ・インテル® Core™ i7-8565U 	<ul style="list-style-type: none"> ・インテル® Core™ i3-8145U ・インテル® Core™ i5-8265U ・インテル® Core™ i7-8565U 	<ul style="list-style-type: none"> ・インテル® Core™ i5-8265U ・インテル® Core™ i7-8565U
メモリ	<ul style="list-style-type: none"> ・8GB ・16GB 	<ul style="list-style-type: none"> ・4GB ・8GB ・16GB 	<ul style="list-style-type: none"> ・4GB ・8GB ・16GB
ストレージ	<ul style="list-style-type: none"> ・SSD 約256GB ・SSD 約512GB ・SSD 約1TB 	<ul style="list-style-type: none"> ・SSD 約128GB ・SSD 約256GB ・SSD 約512GB ・SSD 約1TB 	<ul style="list-style-type: none"> ・SSD 約128GB ・SSD 約256GB ・SSD 約512GB ・SSD 約1TB
光学ドライブ	<ul style="list-style-type: none"> ・なし ・DVDスーパーマルチ 	<ul style="list-style-type: none"> ・なし ・DVDスーパーマルチ 	<ul style="list-style-type: none"> ・なし ・DVDスーパーマルチ
ディスプレイ	13.3型(1920×1080)ノングレ	13.3型(1920×1080)ノングレ	13.3型(1920×1080)タッチ

<https://www.fujitsu-webmart.com/pc/webmart/ui2012.jsp?pos=ank04>

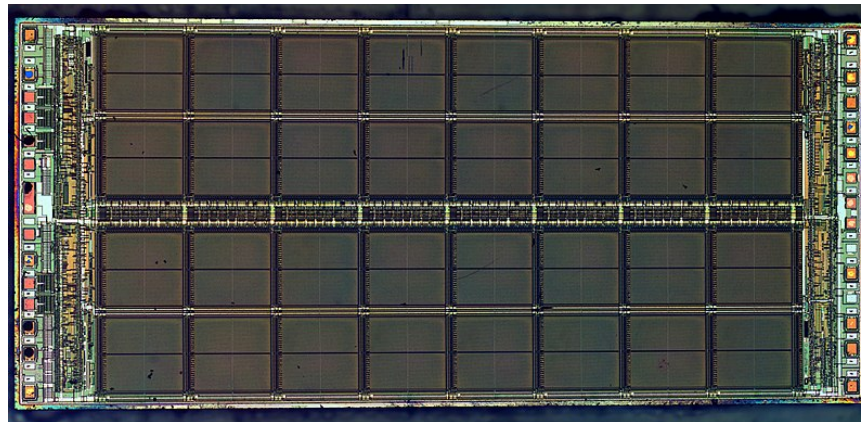
プロセッサ

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



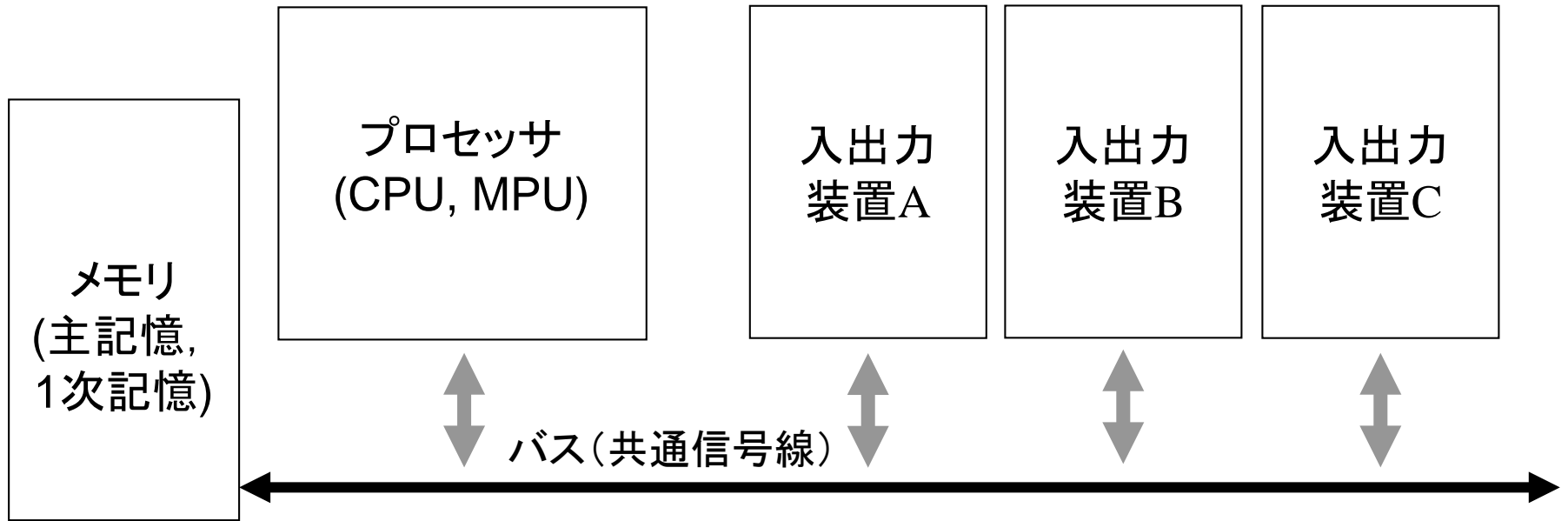
<https://www.facebook.com/hkepc/posts/10155027471233946>

メモリ



<https://commons.wikimedia.org/wiki/File:MT4C1024-HD.jpg>

計算機の基本構成



入出力装置 (Input/Output, I/O) の例

- 2次記憶 (外部記憶, ストレージ): ハードディスク, フラッシュメモリ, 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

単に「メモリ」という場合に,

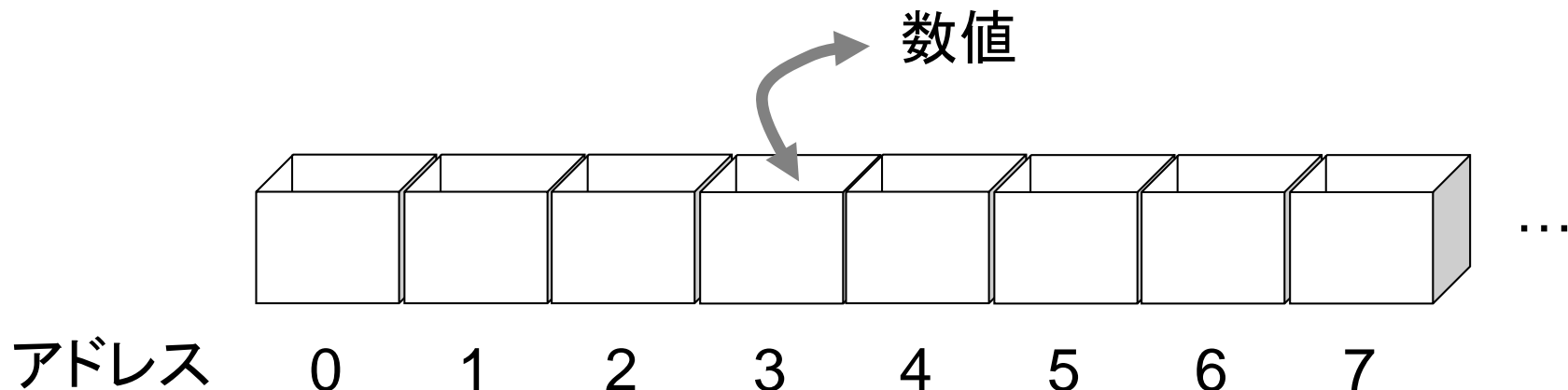
- 主記憶 (電源断で消える) を指すのか
- 2次記憶 (電源断で消えない) を指すのか
に注意すること

動作原理

計算機的基本的原理

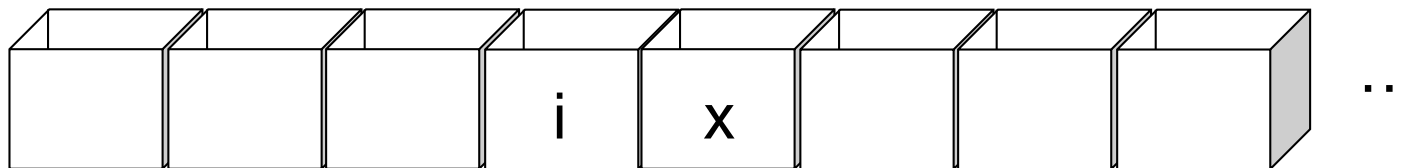
- 動作を指示する「命令」の列がメモリ内に並んでいる
- 計算の対象である「データ」もメモリ内に並んでいる
- プロセッサは、(原則として) **並んでいる順に命令を実行する**
- **命令実行によって、データが次々と書き換えられていく**

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: }
```

mem[3] に 5 を保存
mem[4] に 10 を保存
mem[3] + 1 を mem[3] に保存
mem[3] と 3 を比較
比較結果が「>」でなければ L1 へ
mem[4] + mem[3] を mem[4] に保存
L2 へ
L1: mem[4] - mem[3] を mem[4] に保存
L2: 終了

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)

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

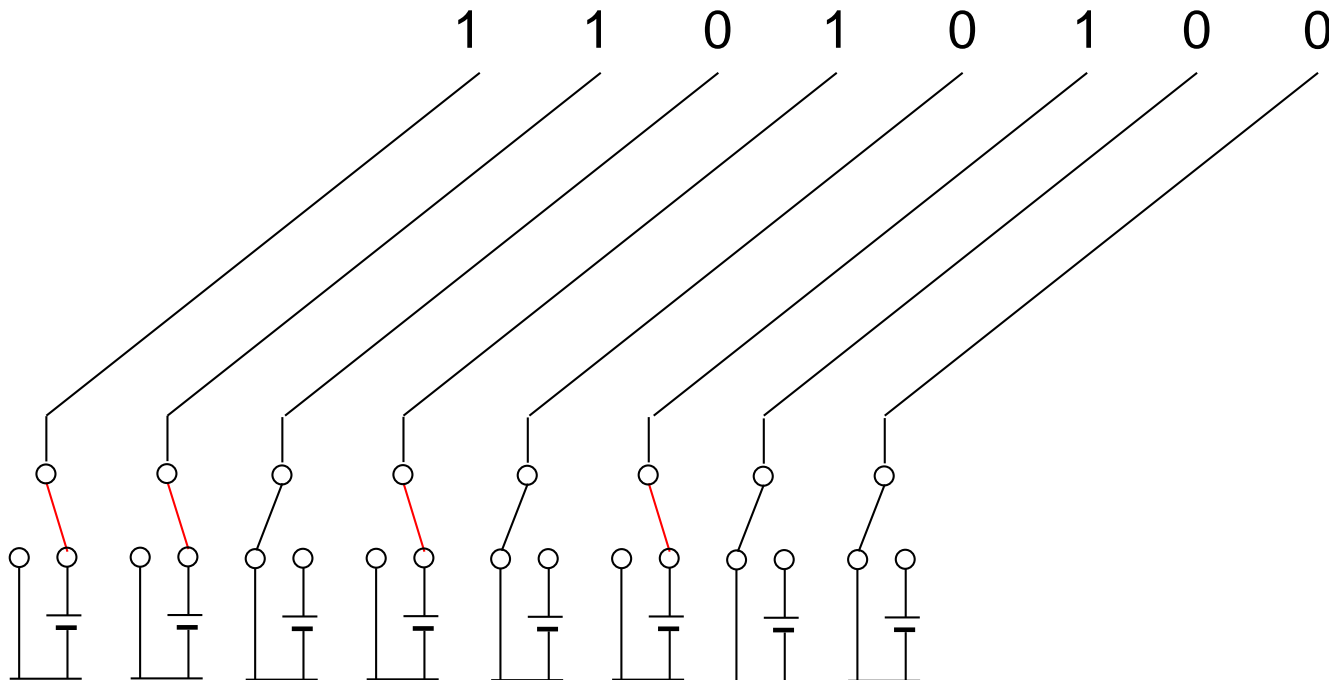
つまり計算機内の数値とは、0 と 1 が一定数並んだものである

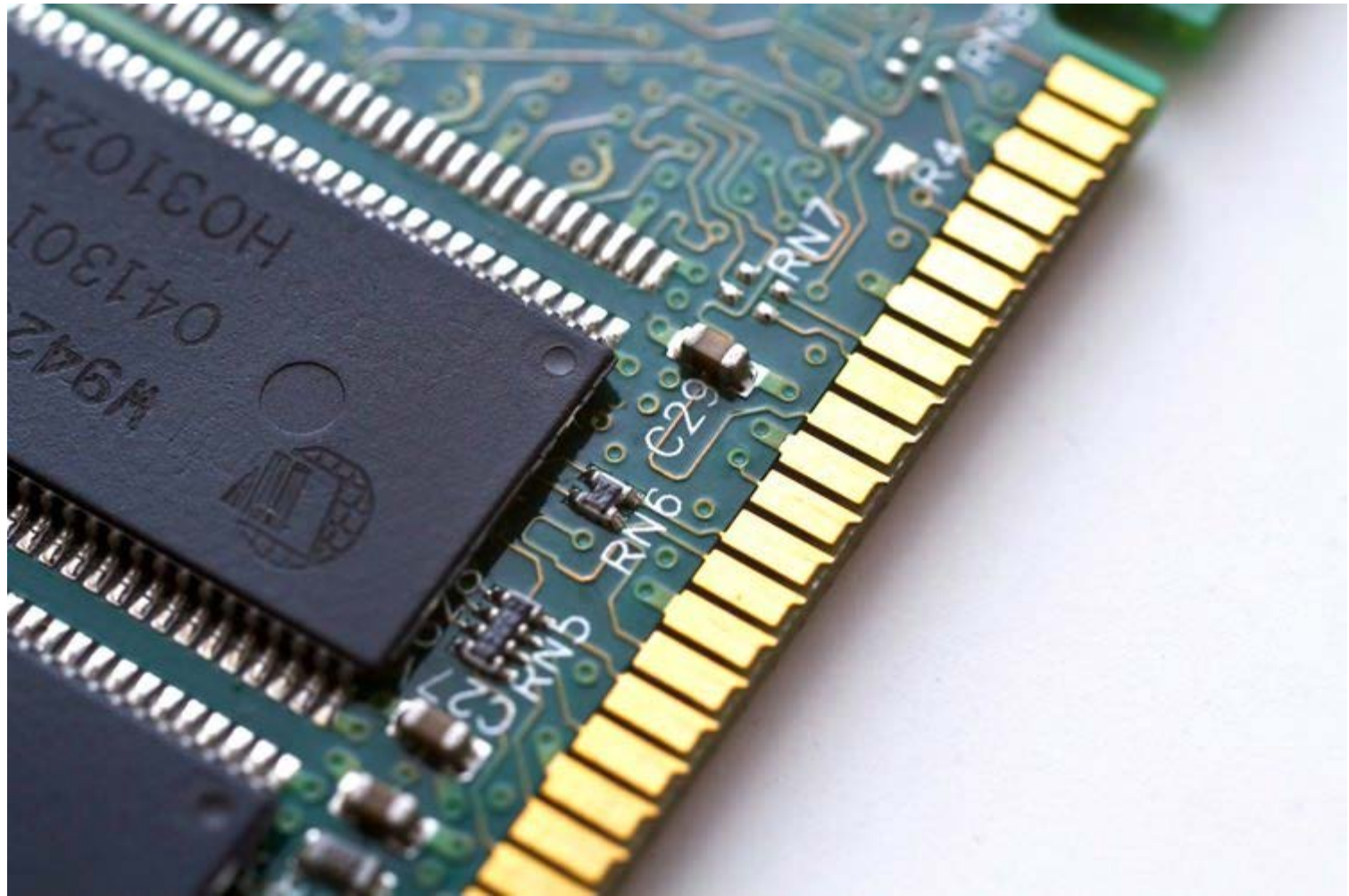
なぜ2進数を使うのか

- 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





さまざまなデータの表現

音声: 数値が1次元に並んだもの

画像: 数値が2次元に並んだもの

文字列: 各文字に数値を割り当てて、それを並べたもの

... → すべて2進数

コード	文字	コード	文字	コード	文字	コード	文字	コード	文字	コード	文字
32	空白	48	0	64	@	80	P	96	`	112	p
33	!	49	1	65	A	81	Q	97	a	113	q
34	"	50	2	66	B	82	R	98	b	114	r
35	#	51	3	67	C	83	S	99	c	115	s
36	\$	52	4	68	D	84	T	100	d	116	t
37	%	53	5	69	E	85	U	101	e	117	u
38	&	54	6	70	F	86	V	102	f	118	v
39	'	55	7	71	G	87	W	103	g	119	w
40	(56	8	72	H	88	X	104	h	120	x
41)	57	9	73	I	89	Y	105	i	121	y
42	*	58	:	74	J	90	Z	106	j	122	z
43	+	59	;	75	K	91	[107	k	123	{
44	,	60	<	76	L	92	\	108	l	124	
45	-	61	=	77	M	93]	109	m	125	}
46	.	62	>	78	N	94	^	110	n	126	~
47	/	63	?	79	O	95	_	111	o	127	DEL

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,

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

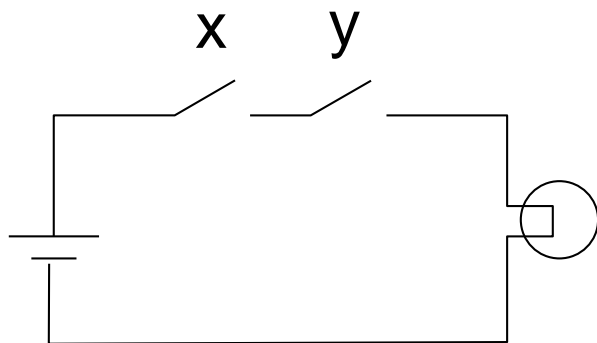
$$z_2 = f_2(x_1, x_2, \dots, x_8, y_1, y_2, \dots, y_8)$$

⋮

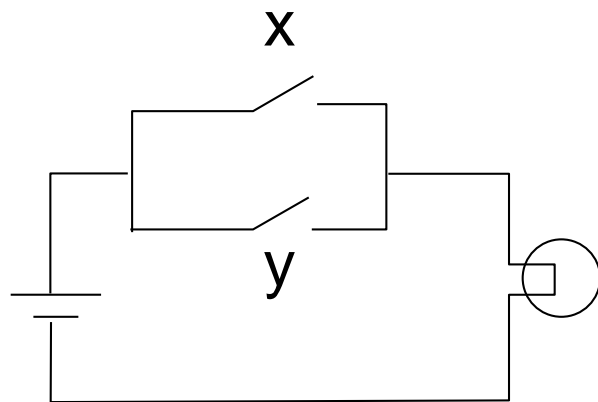
$$z_8 = f_8(x_1, x_2, \dots, x_8, y_1, y_2, \dots, y_8)$$

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

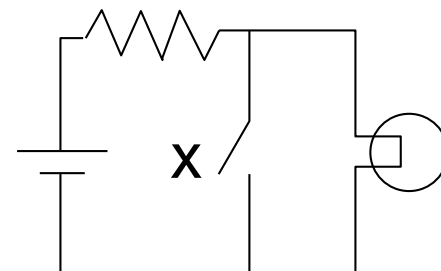
このような**論理関数**は必ず **AND, OR, NOT** の組合せで表せる



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

プログラムの構成

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

```
mem[3] に 5 を保存
mem[4] に 10 を保存
mem[3] + 1 を mem[3] に保存
mem[3] と 3 を比較
比較結果が「>」でなければ L1 へ
mem[4] + mem[3] を mem[4] に保存
L2 へ
L1: mem[4] - mem[3] を mem[4] に保存
L2: 終了
```

命令

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
(プログラム記憶方式, 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;  
}
```

```
mem[3] に 5 を保存  
mem[4] に 10 を保存  
mem[3] + 1 を mem[3] に保存  
mem[3] と 3 を比較  
比較結果が「>」でなければ L1 へ  
mem[4] + mem[3] を mem[4] に保存  
L2 へ  
L1: mem[4] - mem[3] を mem[4] に保存  
L2: 終了
```

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

ソースコード
(C言語)

アセンブリコード
(アセンブリ言語)

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

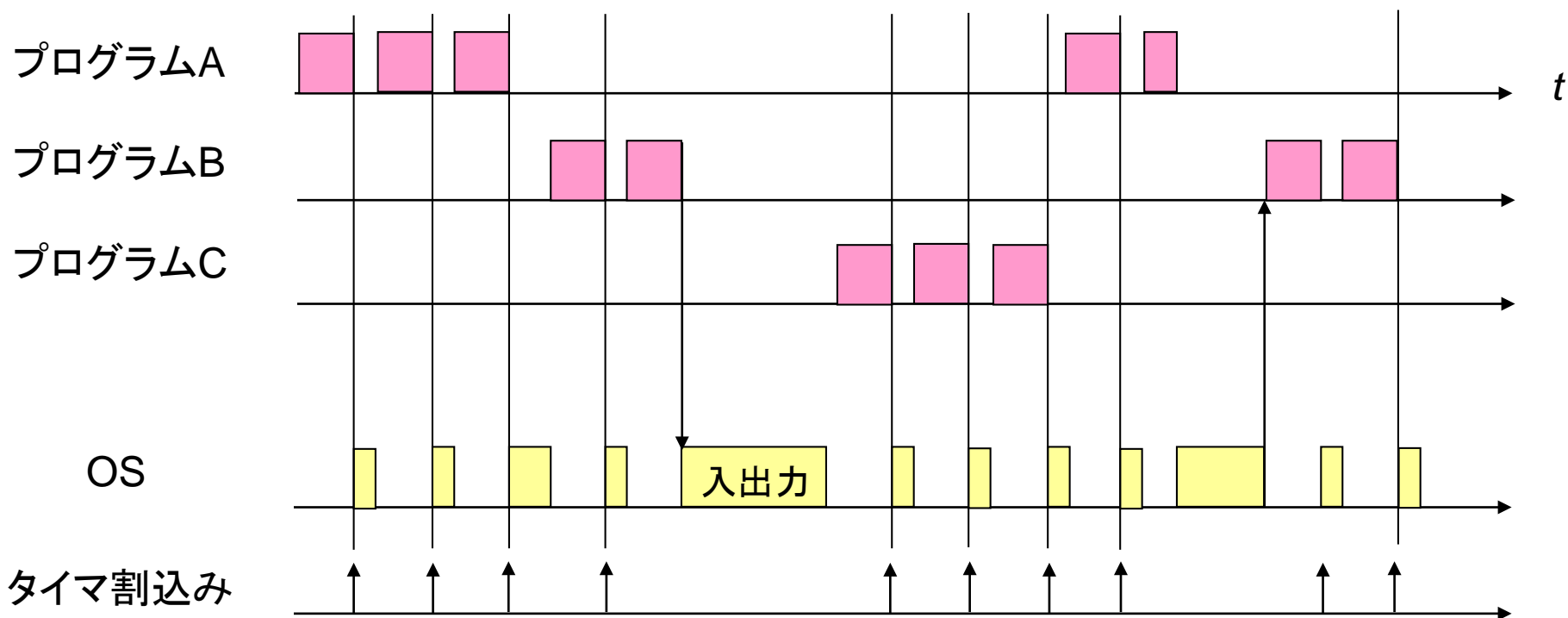
実際に試してみる: <https://godbolt.org/>

オペレーティングシステム (OS)

Q: 命令が順番に実行されているだけなのだとしたら, なぜ PC 上では複数のソフトウェアが同時に走っているように見えるのか?

A: オペレーティングシステム (OS) と呼ばれるソフトウェアが, 複数のプログラムの時分割切り替えを行っている

OSの例: Windows, macOS, Linux, ...



計算機の歴史(前編)

(後編は「命令セットアーキテクチャ」の回)

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も変わる

計算機アーキテクチャ (computer architecture):

- 計算機の構成方法を何らかの抽象的なレベルで記述したもの
- 「レベル」の例は追いつ追いつ説明する

計算機の歴史

真空管 (1904)

トランジスタ (1947)

集積回路 (1958)

マイクロプロセッサ
(1971)

磁気ドラムメモリ (1932)

陰極線管メモリ (1946?)

水銀遅延管 (1947?)

磁気コアメモリ (1949)

半導体メモリ (DRAM)
(1966)

機械式

電気機械式

電子式

(17~19c.)

Z1 (1938)

Z3 (1941)

Harvard Mk1
(1944)

ABC (1942)

Colossus (1943)

ENIAC (1946)

SSEM (1948)

EDSAC (1949)

EDVAC (1951)

PDP-1 (1960)

System/360 (1964)

プログラム
内蔵方式

計算機アーキテク
チャの概念

計算機の歴史

真空管 (1904)

トランジスタ (1947)

集積回路 (1958)

マイクロプロセッサ
(1971)

磁気ドラムメモリ (1932)

陰極線管メモリ (1946?)

水銀遅延管 (1947?)

磁気コアメモリ (1949)

半導体メモリ (DRAM)
(1966)

機械式

電気機械式

電子式

(17~19c.)

Z1 (1938)

Z3 (1941)

Harvard Mk1
(1944)

ABC (1942)

Colossus (1943)

ENIAC (1946)

SSEM (1948)

EDSAC (1949)

EDVAC (1951)

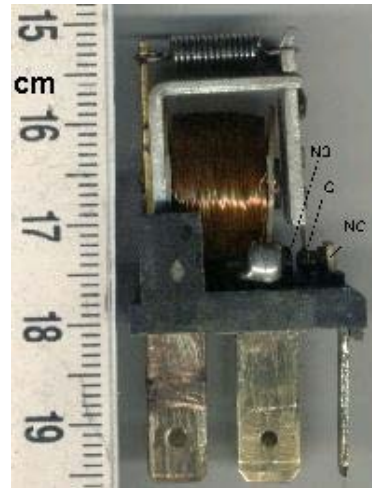
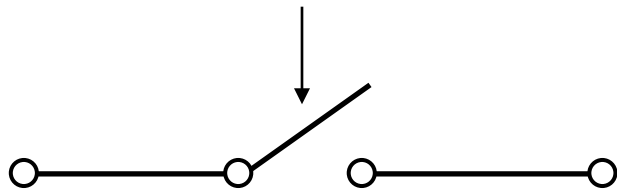
PDP-1 (1960)

System/360 (1964)

プログラム
内蔵方式

計算機アーキテク
チャの概念

スイッチ回路の変遷



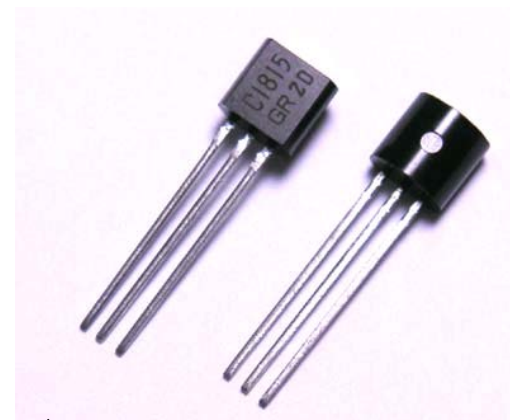
電磁リレー

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



真空管

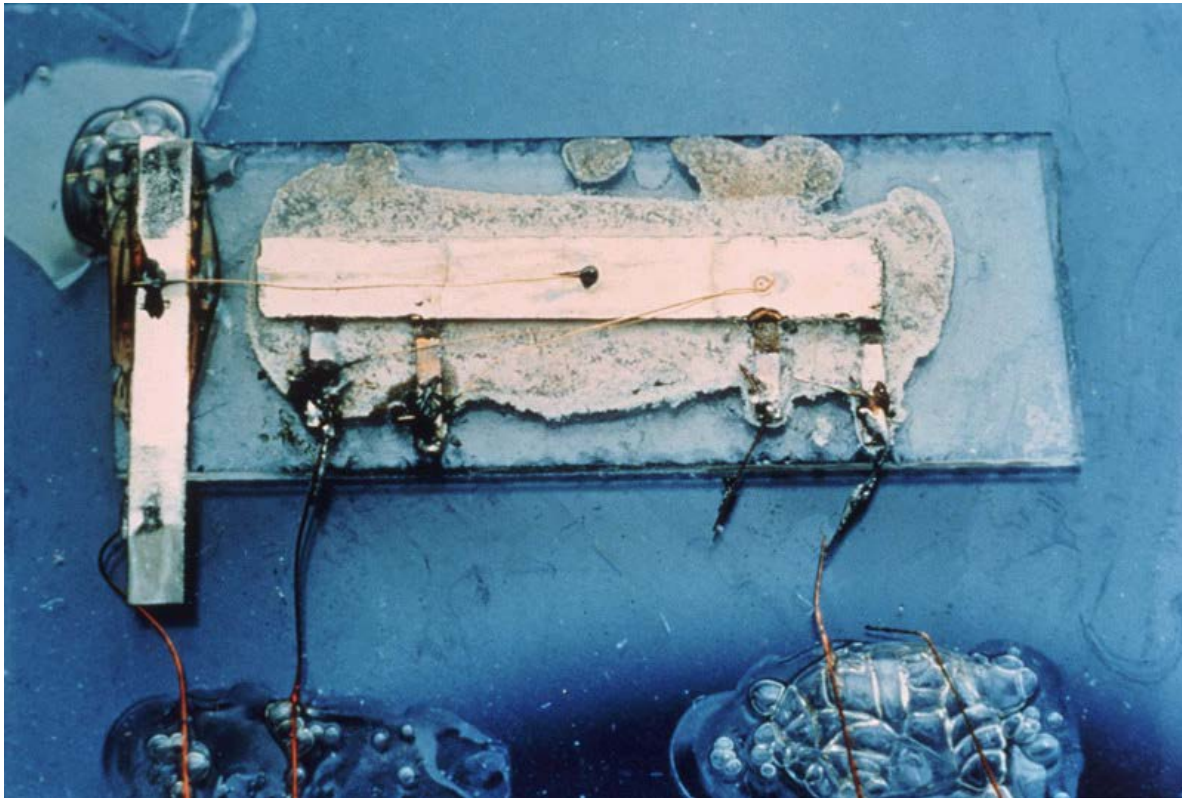
<http://ja.wikipedia.org/wiki/%E7%9C%9F%E7%A9%BA%E7%AE%A1>



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, IC)

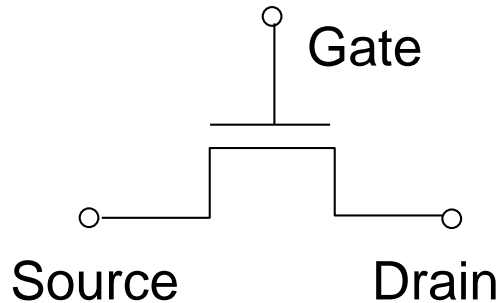


<http://www.ti.com/corp/docs/kilbyctr/downloadphotos.shtml>

- 個別素子(主にトランジスタ)を組み合わせて配線する代わりに, 単一の半導体基板上に複数の素子を形成
- 大規模化したものを LSI (Large Scale Integration), VLSI (Very LSI) などと呼ぶ

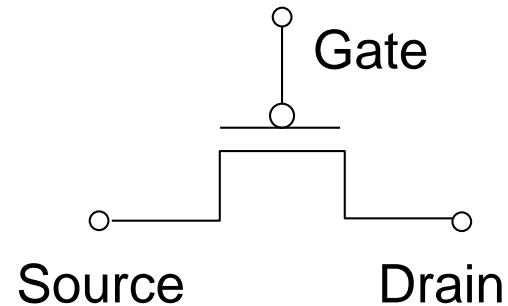
MOS 型トランジスタ (Metal-Oxide-Semiconductor)

NMOSトランジスタ



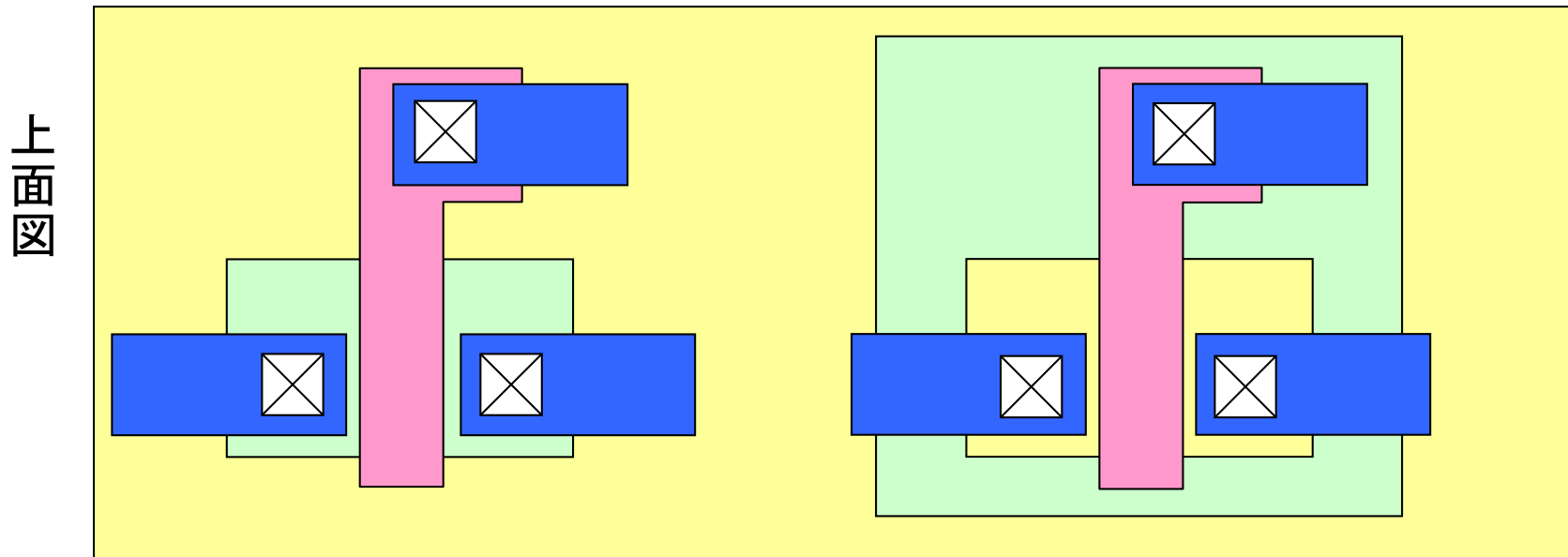
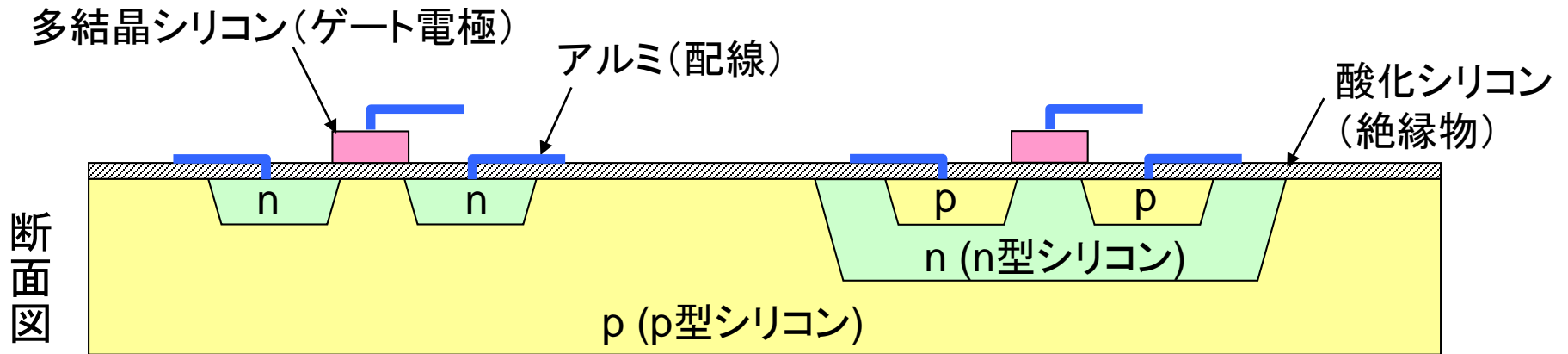
ゲートに電圧がかかっているとき (high, e.g. 3.3V), ソース-ドレイン間を電流が流れる

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

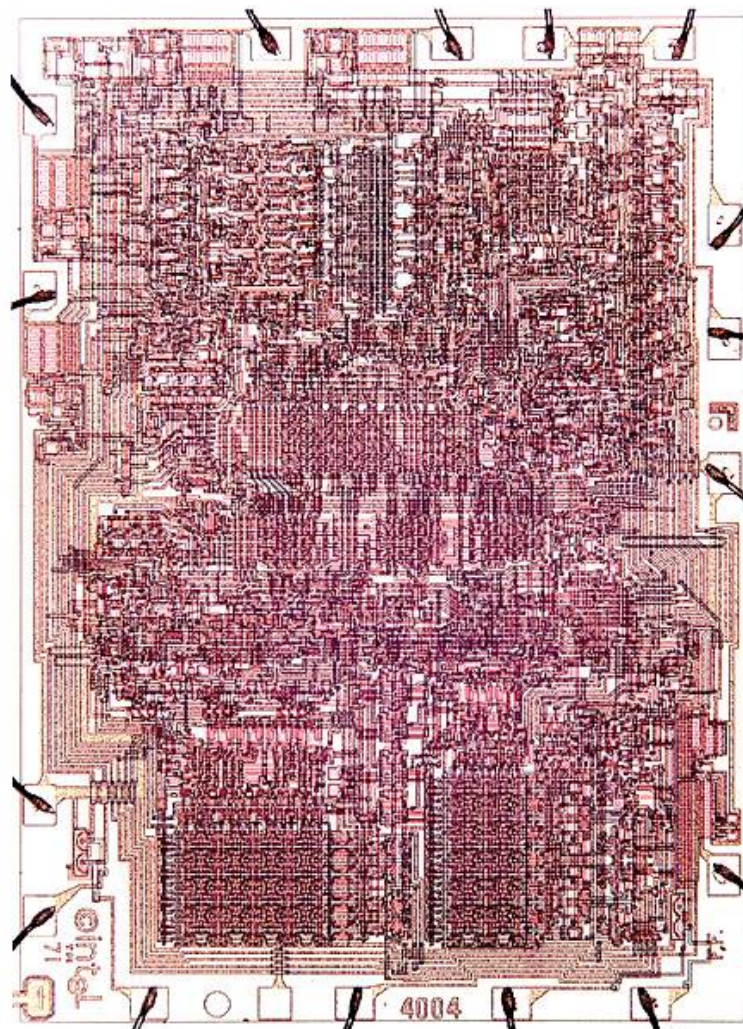
トランジスタの構造 (模式図)



NMOS

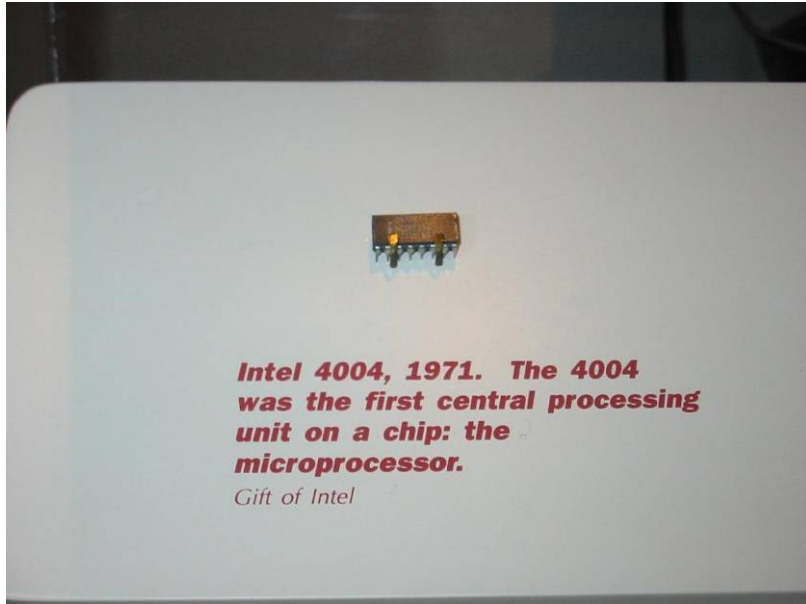
PMOS

Intel 4004 (1971)



http://news.com.com/1971+Intel+4004+processor/2009-1006_3-6038974-3.html

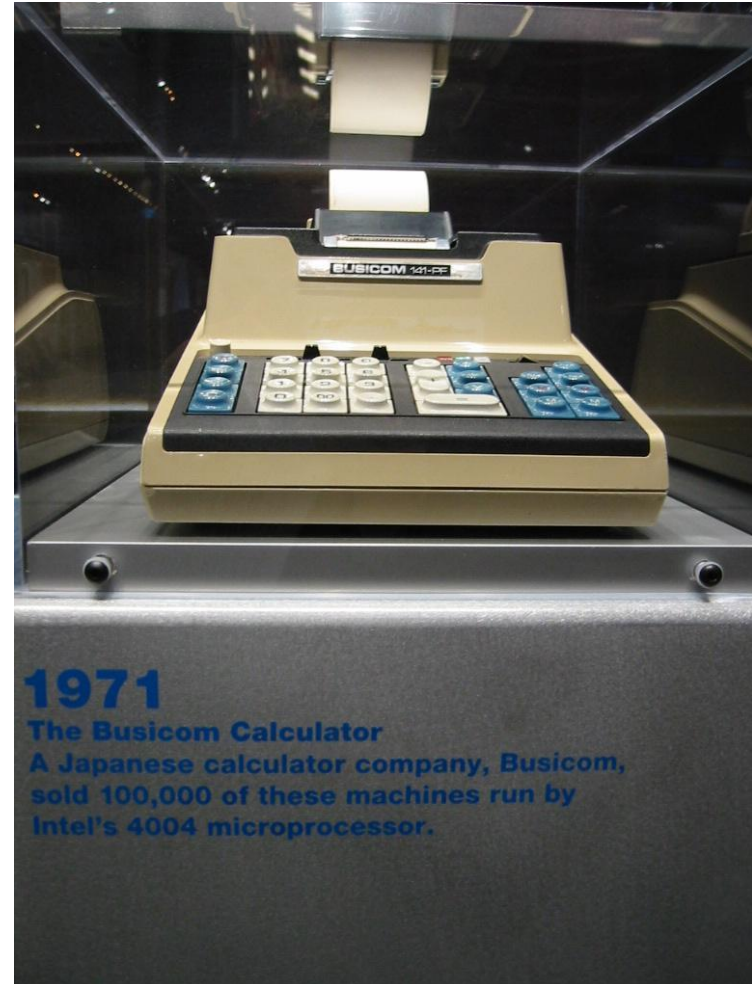
Intel 4004



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

Gift of Intel

(American History Museum)



1971

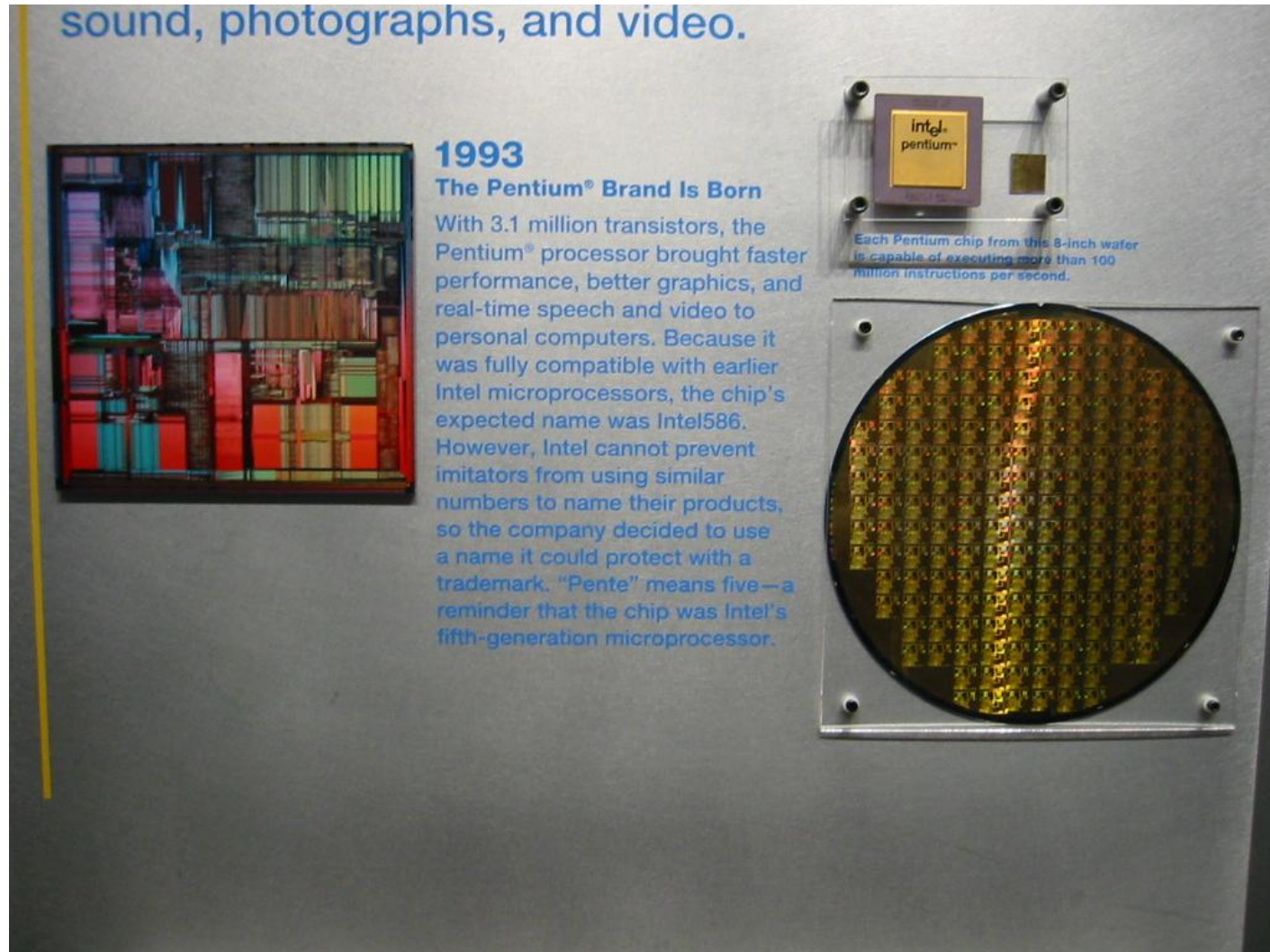
The Busicom Calculator

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

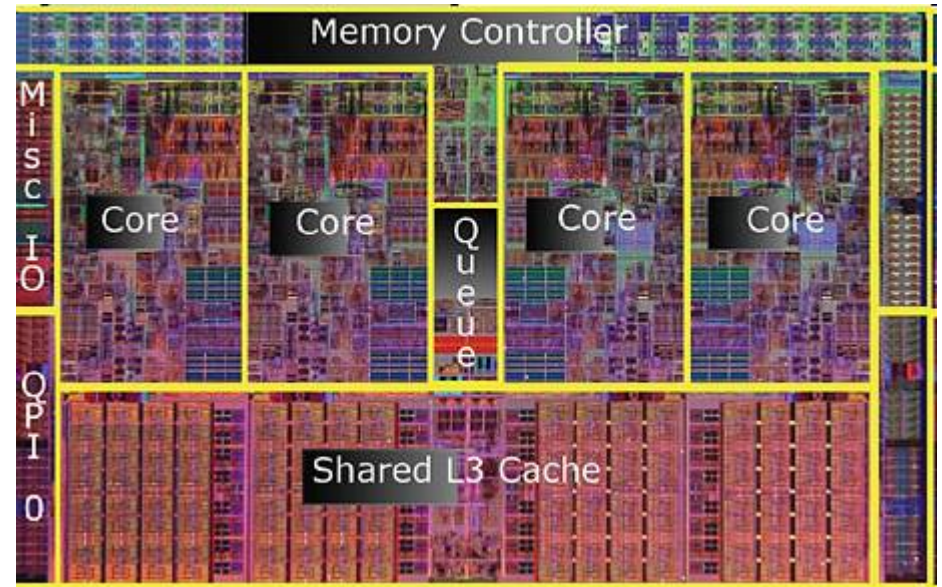
Busicom 141-PF

(Intel Museum, Santa Clara)

Intel Pentium (1993)



Intel Core i7 (2008-)



Intel Developer Forum 2008, Shanghai

http://ja.wikipedia.org/wiki/Intel_Core_i7

<https://www.atmarkit.co.jp/ait/articles/0811/27/news144.html>

計算機の歴史

真空管 (1904)

トランジスタ (1947)

集積回路 (1958)

マイクロプロセッサ
(1971)

磁気ドラムメモリ (1932)

陰極線管メモリ (1946?)

水銀遅延管 (1947?)

磁気コアメモリ (1949)

半導体メモリ (DRAM)
(1966)

機械式

電気機械式

電子式

(17~19c.)

Z1 (1938)

Z3 (1941)

Harvard Mk1
(1944)

ABC (1942)

Colossus (1943)

ENIAC (1946)

SSEM (1948)

EDSAC (1949)

EDVAC (1951)

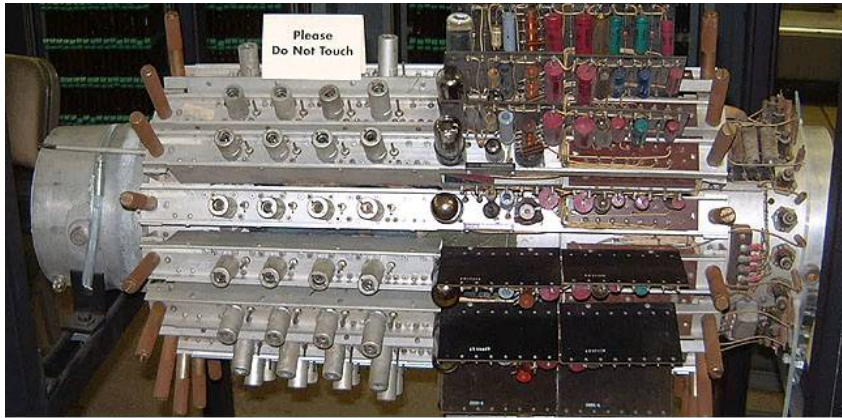
PDP-1 (1960)

System/360 (1964)

プログラム
記憶方式

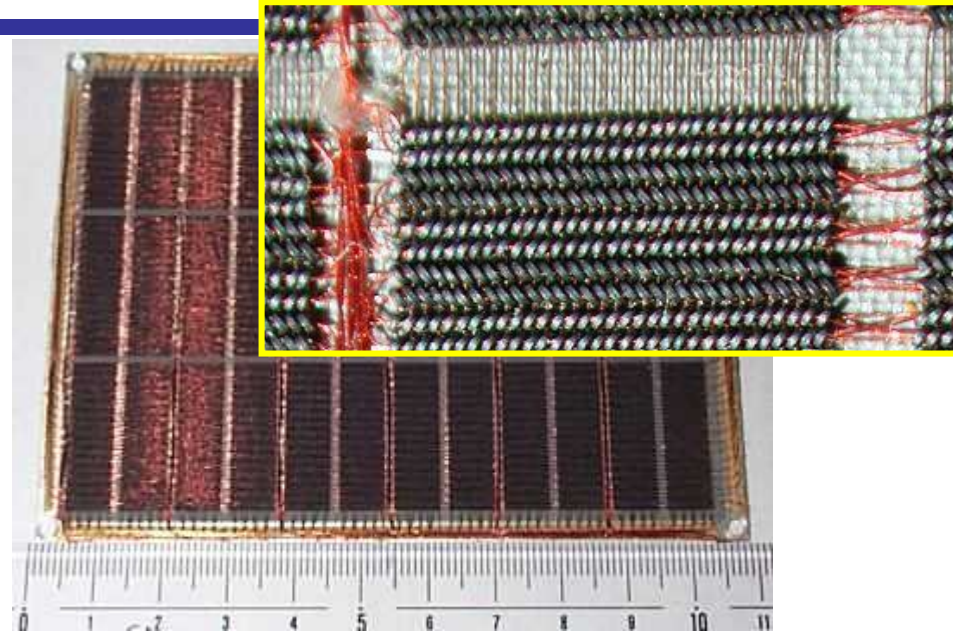
計算機アーキテク
チャの概念

記憶素子の変遷



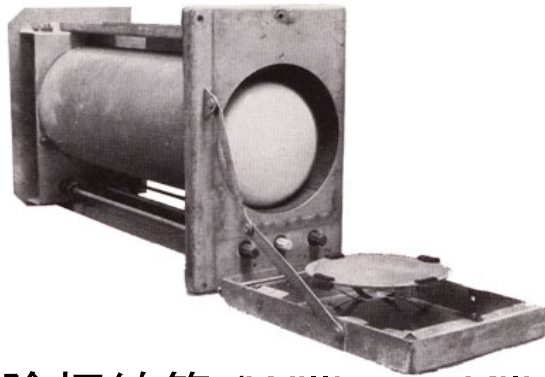
水銀遅延管

https://commons.wikimedia.org/wiki/File:Mercury_memory.jpg



磁気コアメモリ

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



陰極線管 (Williams-Kilburn管)

<https://en.wikipedia.org/wiki/File:Williams-tube.jpg>



半導体メモリ

計算機の歴史

真空管 (1904)

トランジスタ (1947)

集積回路 (1958)

マイクロプロセッサ
(1971)

磁気ドラムメモリ (1932)

陰極線管メモリ (1946?)

水銀遅延管 (1947?)

磁気コアメモリ (1949)

半導体メモリ (DRAM)
(1966)

機械式

電気機械式

電子式

(17~19c.)

Z1 (1938)

Z3 (1941)

Harvard Mk1
(1944)

ABC (1942)

Colossus (1943)

ENIAC (1946)

SSEM (1948)

EDSAC (1949)

EDVAC (1951)

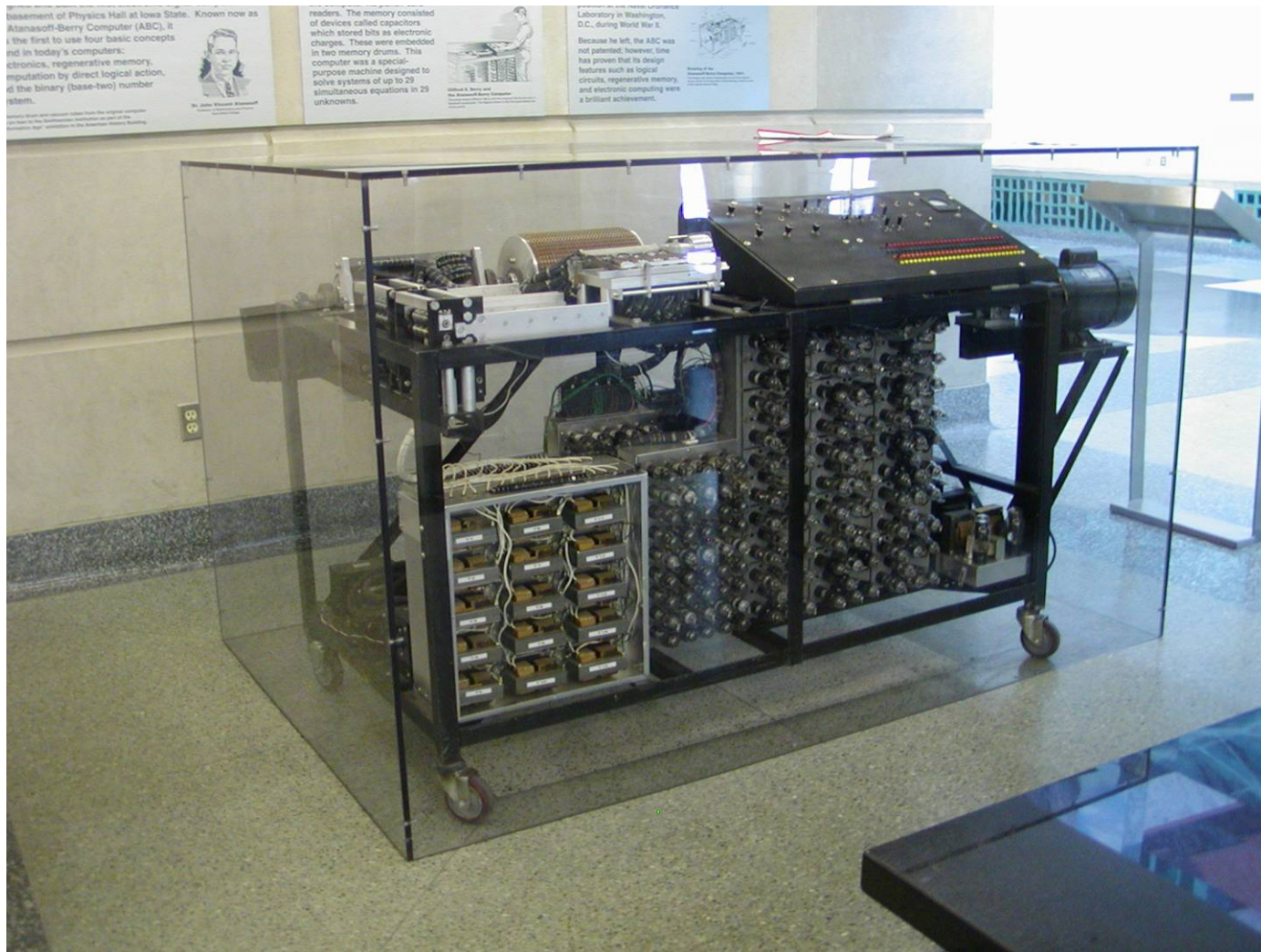
PDP-1 (1960)

System/360 (1964)

プログラム
記憶方式

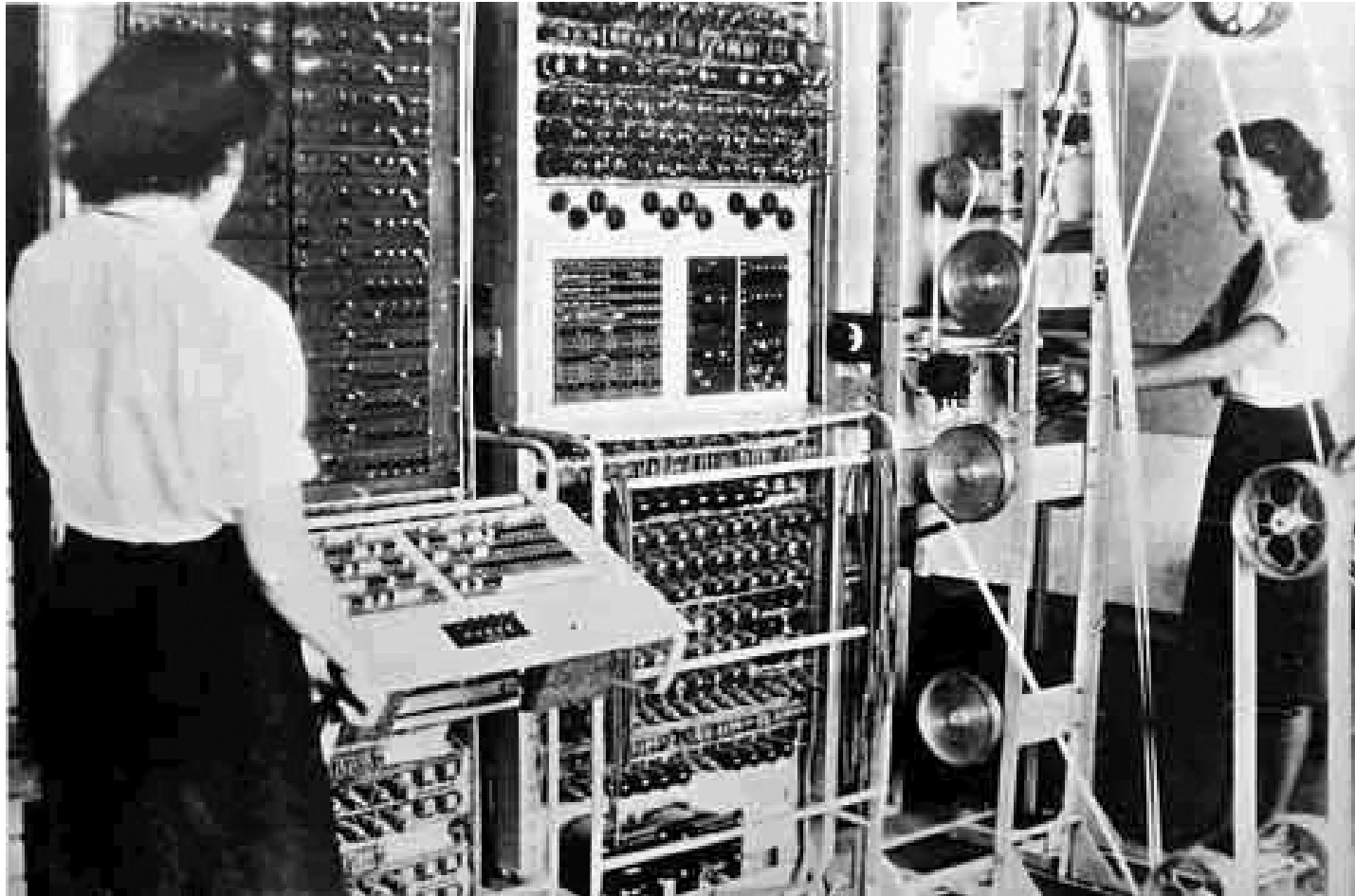
計算機アーキテク
チャの概念

ABC (1942)



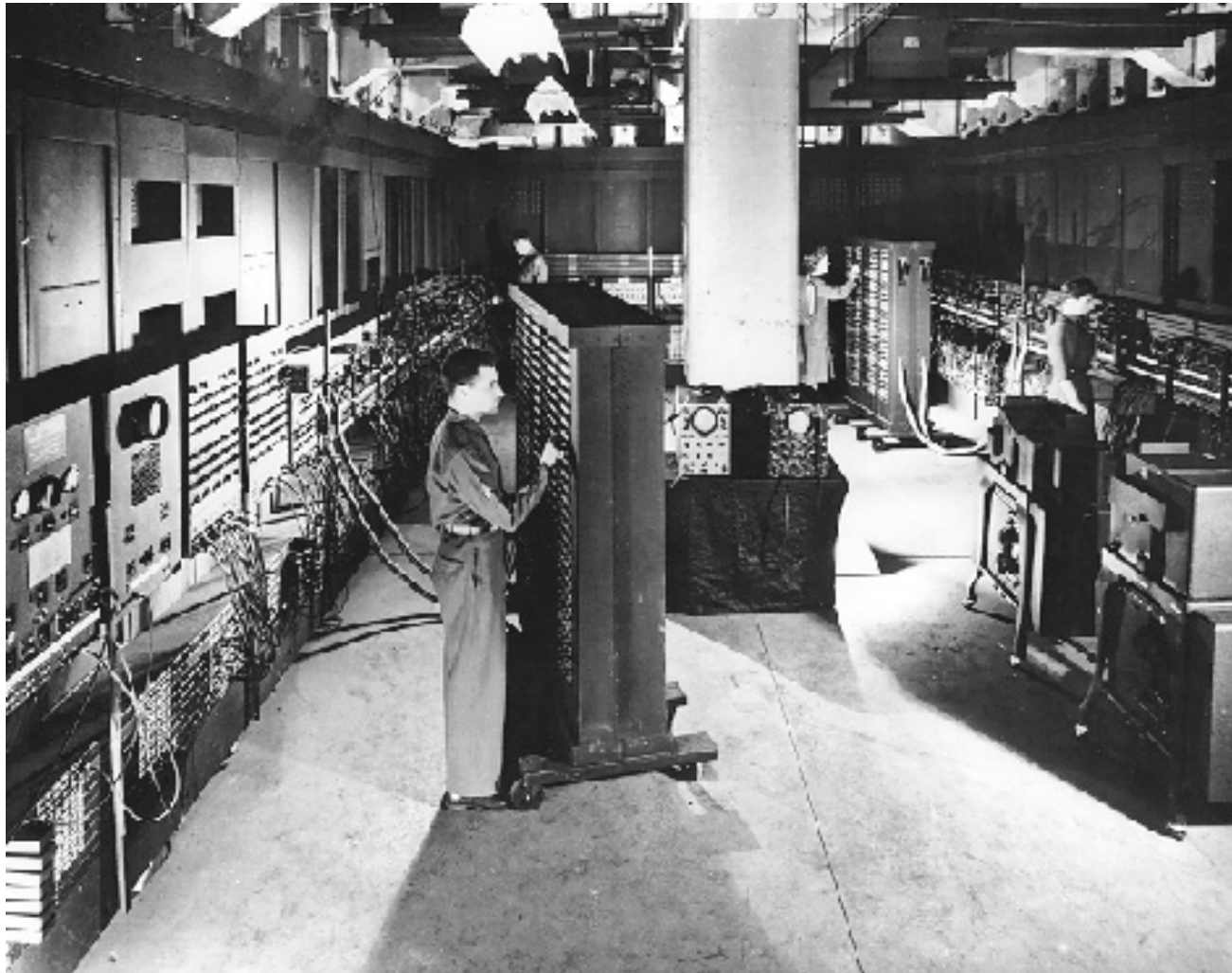
http://commons.wikimedia.org/wiki/File:Atanasoff-Berry_Computer.jpg

Colossus (1943)



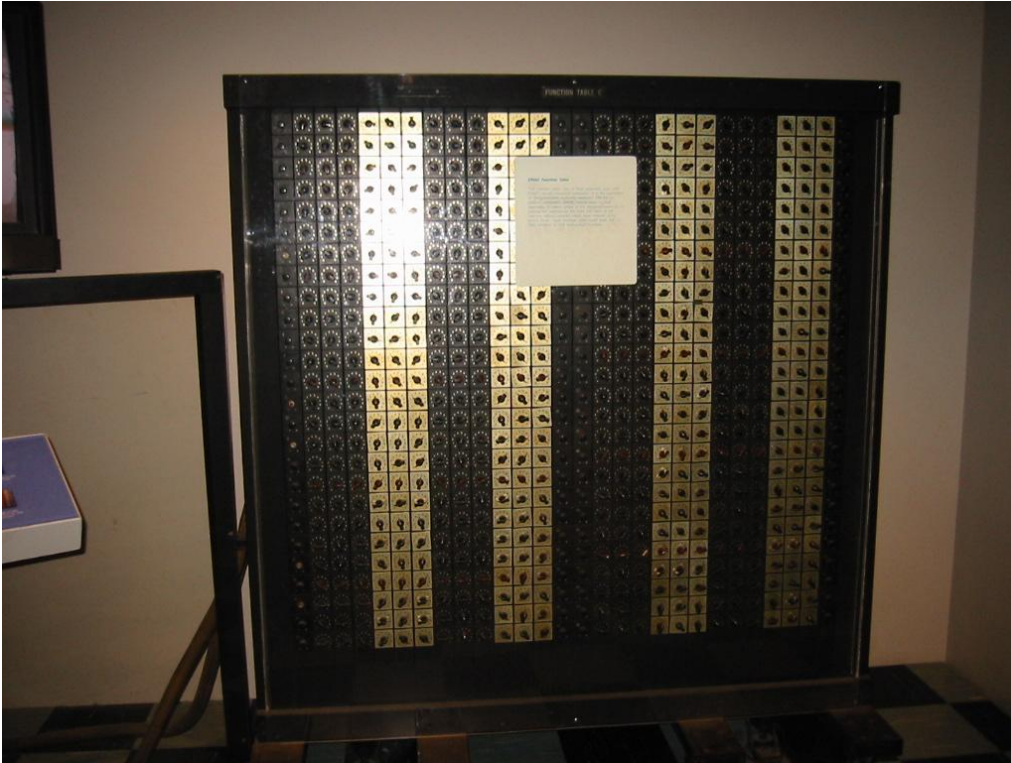
<http://commons.wikimedia.org/wiki/File:Colossus.jpg>

ENIAC (1946)



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

ENIACのプログラミングパネル



(American History Museum)

このパネルにケーブルを挿してプログラミングした
つまりまだプログラム記憶型ではなかった

計算機の歴史

真空管 (1904)

トランジスタ (1947)

集積回路 (1958)

マイクロプロセッサ
(1971)

磁気ドラムメモリ (1932)

陰極線管メモリ (1946?)

水銀遅延管 (1947?)

磁気コアメモリ (1949)

半導体メモリ (DRAM)
(1966)

機械式

電気機械式

電子式

(17~19c.)

Z1 (1938)

Z3 (1941)

Harvard Mk1
(1944)

ABC (1942)

Colossus (1943)

ENIAC (1946)

SSEM (1948)

EDSAC (1949)

EDVAC (1951)

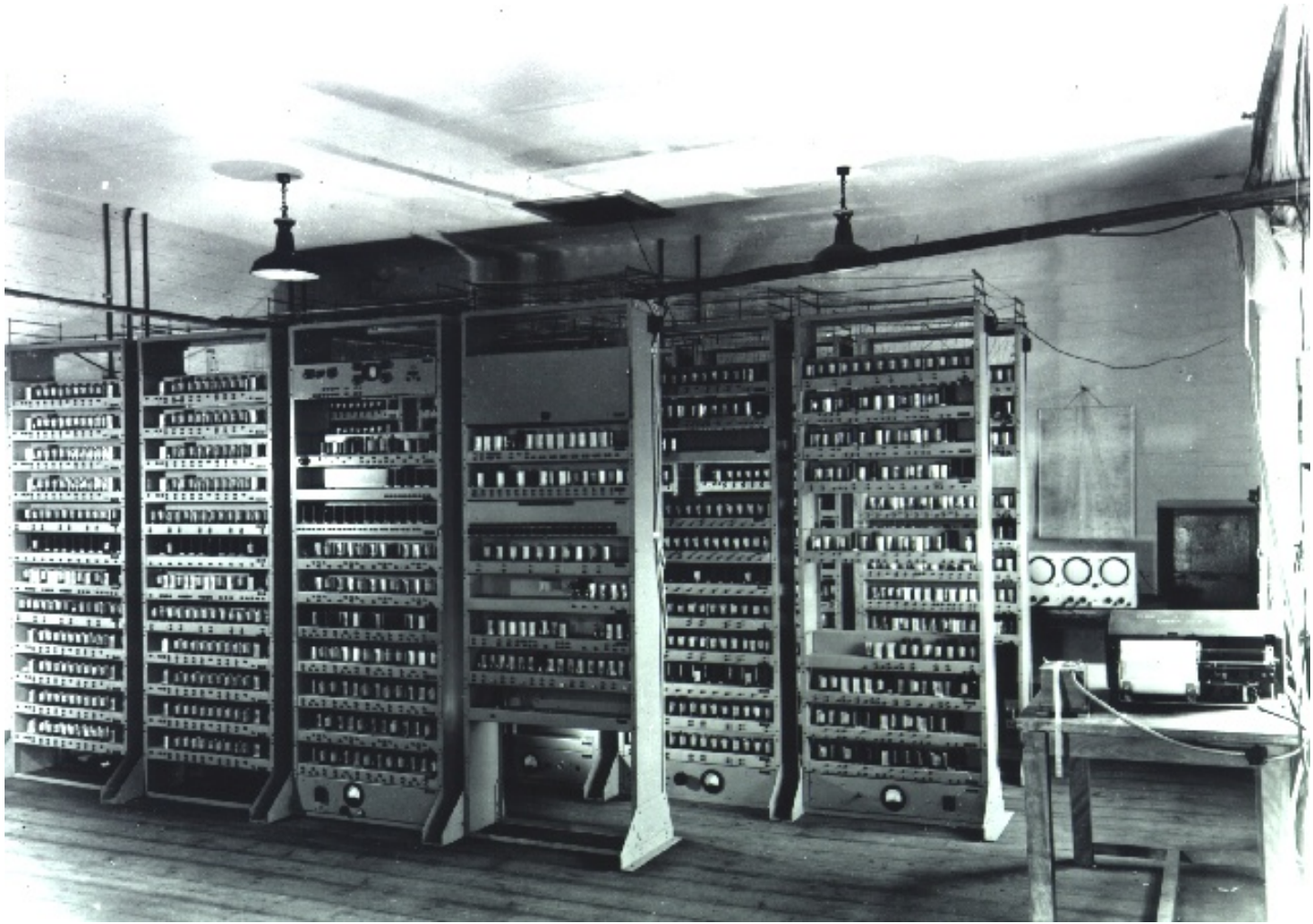
PDP-1 (1960)

System/360 (1964)

プログラム
記憶方式

計算機アーキテク
チャの概念

EDSAC (1949)



http://www.cl.cam.ac.uk/relics/archive_photos.html

EDVAC (1951)



<http://ftp.arl.mil/ftp/historic-computers/>

計算機の歴史

真空管 (1904)

トランジスタ (1947)

集積回路 (1958)

マイクロプロセッサ
(1971)

磁気ドラムメモリ (1932)

陰極線管メモリ (1946?)

水銀遅延管 (1947?)

磁気コアメモリ (1949)

半導体メモリ (DRAM)
(1966)

機械式

電気機械式

電子式

(17~19c.)

Z1 (1938)

Z3 (1941)

Harvard Mk1
(1944)

ABC (1942)

Colossus (1943)

ENIAC (1946)

SSEM (1948)

EDSAC (1949)

EDVAC (1951)

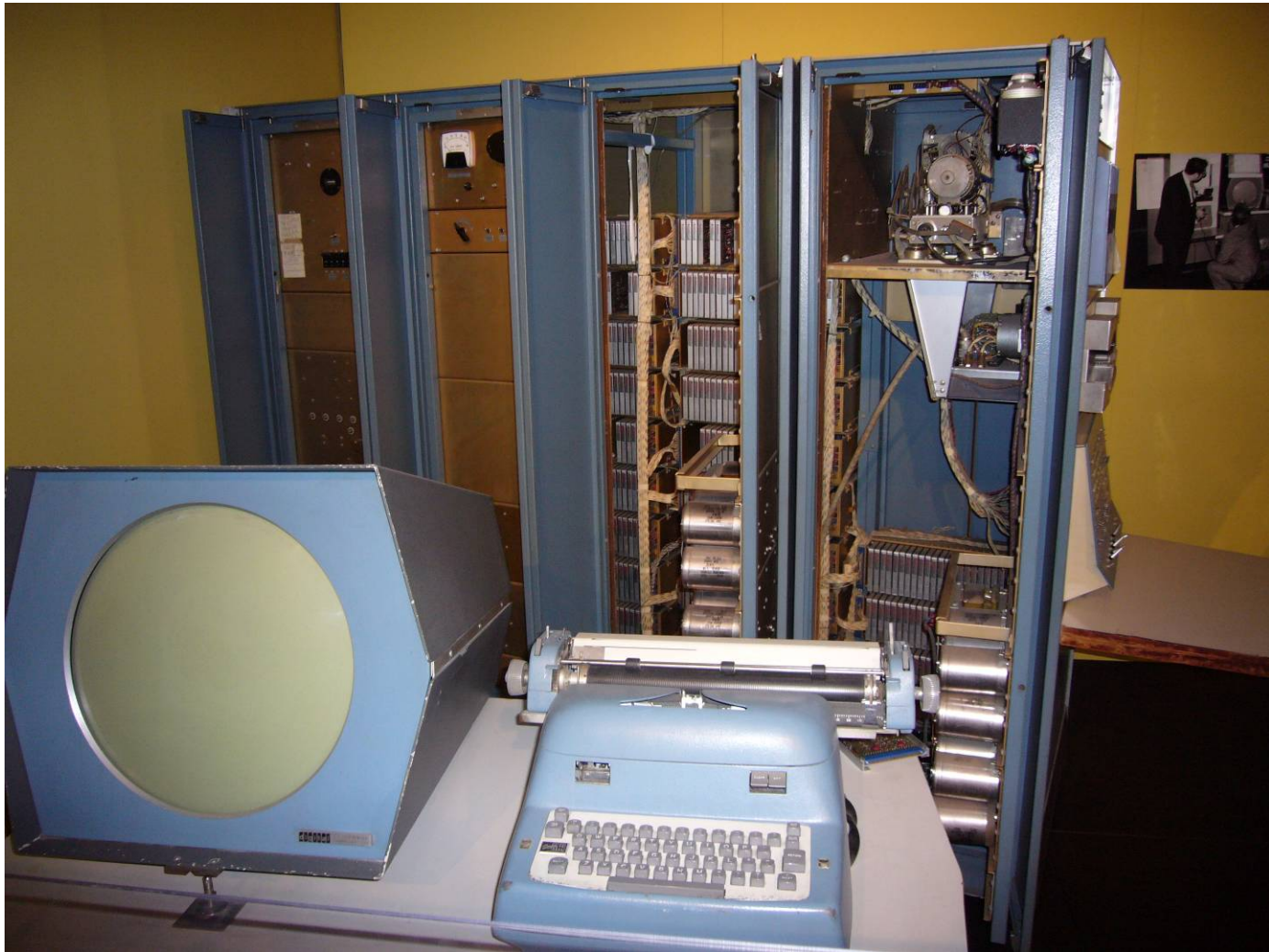
PDP-1 (1960)

System/360 (1964)

プログラム
記憶方式

計算機アーキテク
チャの概念

DEC PDP-1 (1960)



<http://commons.wikimedia.org/wiki/File:PDP-1.jpg>

IBM System/360 (1964)



http://upload.wikimedia.org/wikipedia/commons/8/8d/Bundesarchiv_B_145_Bild-F038812-0014%2C_Wolfsburg%2C_VW_Autowerk.jpg

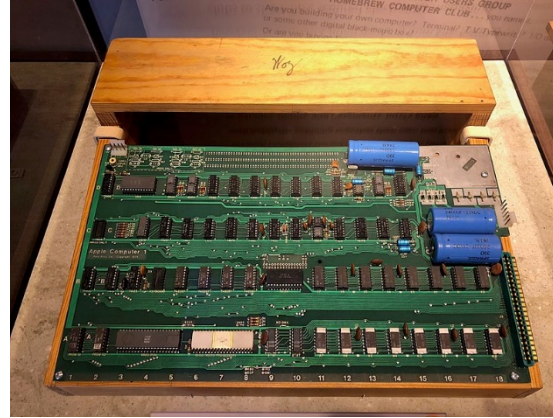
「コンピュータアーキテクチャ」という概念をおそらく最初に明確に導入した商用計算機. オペレーティングシステム(OS)を最初に導入した商用計算機でもある

パーソナルコンピュータの出現 (1970年代)



Altair 8800 (1974)

https://en.wikipedia.org/wiki/File:Altair_8800.jpg



Apple I (1976)

https://en.wikipedia.org/wiki/File:Apple_1_Woz_1976_at_CHM.agr.jpg



PC-8001 (1979)

https://commons.wikimedia.org/wiki/File:Nec_PC_8000_series.jpg

Apple Macintosh, Sun-1, IBM PC



Macintosh (1984)

Sun-1 (1982)

IBM PC (1981)

(American History Museum)

用語の確認

- プロセッサ (CPU, MPU)
- メモリ
- アドレス
- プログラム
- 命令
- プログラム記憶方式
- コンパイラ
- アセンブラ
- アセンブリ言語
- オペレーティングシステム (OS)
- MOSTランジスタ
- 集積回路 (IC, LSI, VLSI)