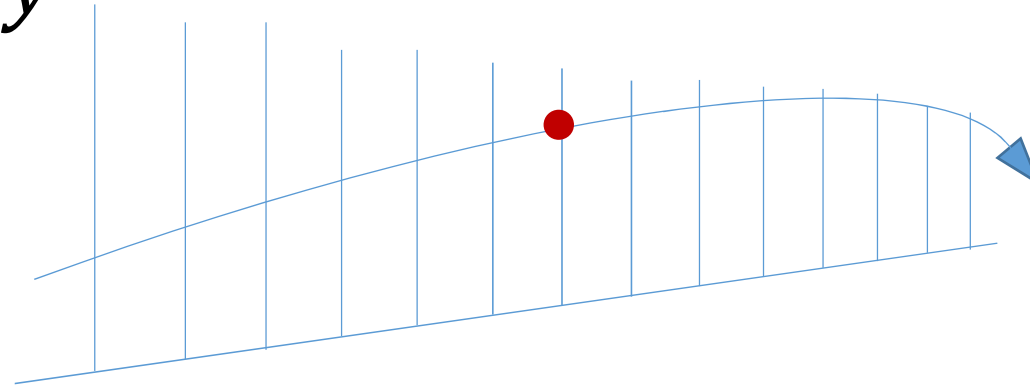


計算物理学



y



物理法則 $F = ma$

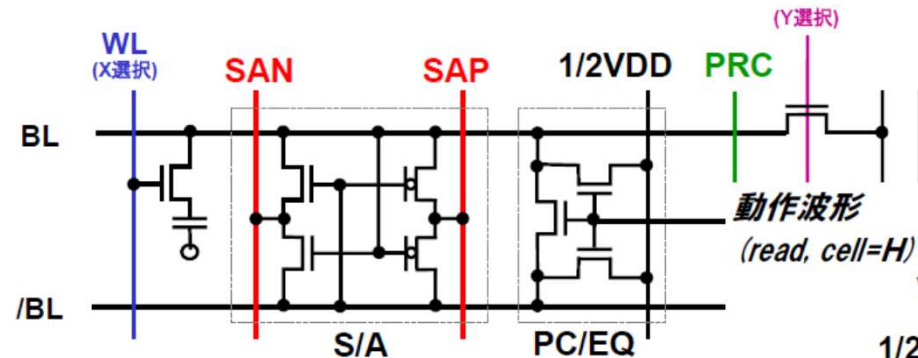
力と加速度

現象を数で表現する

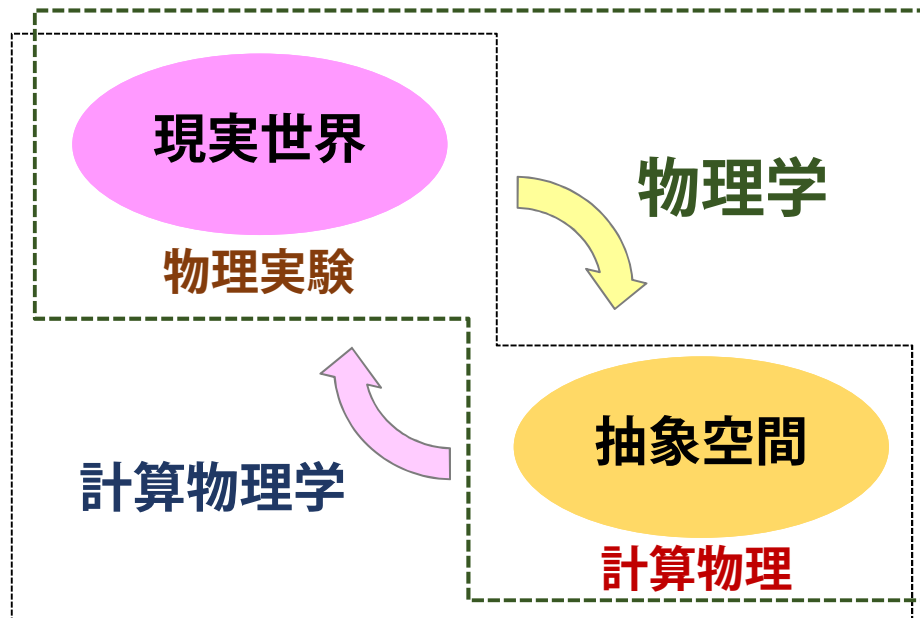
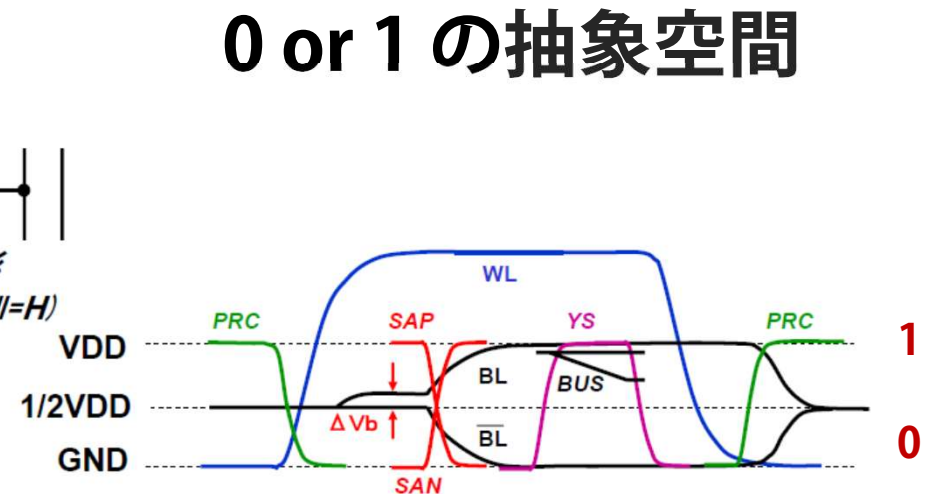
$$mg = m \frac{d^2 y}{dt^2}$$

抽象的座標空間に現象を射影し、数学を用いて理解する

計算機



動作波形
(read, cell=H)



0 or 1 の抽象空間

計算物理

仮想空間の中に
現実を再現

講義内容

- 計算機の概要
- プログラム言語
(Fortran, C, C++)
- 数値計算法
- 物理現象への応用

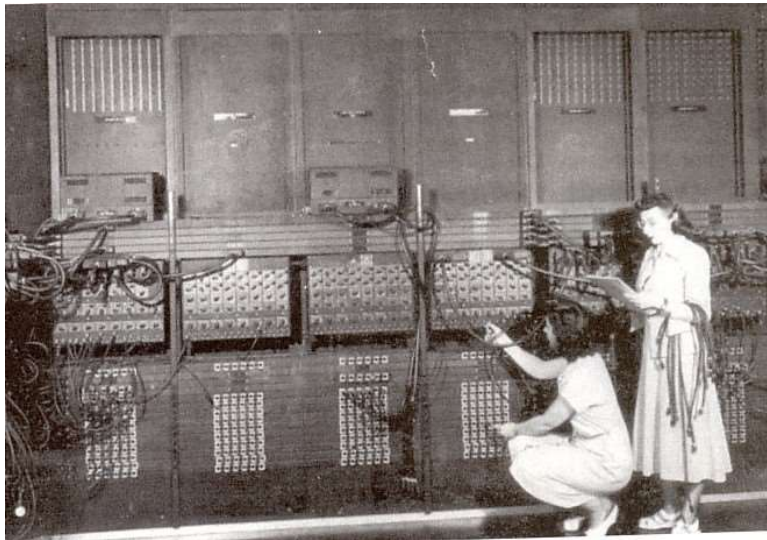
成績は出席とレポートで評価

レポートは 2~3 回を予定

電子計算機の誕生

E N I A C (Electric Numerical Integrator and Computer) 1946年

U.S. Army



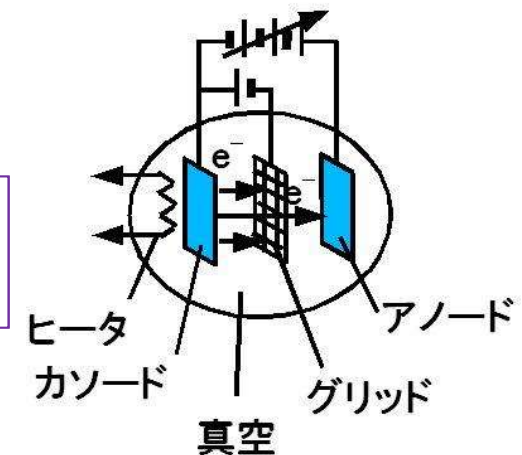
真空管 17,468 本
440 m³
174 kW
5,000 Flops

真空管



弾道解析の計算時間：40時間から30秒に短縮

Flops : FLoating point number Operations Per Second
一秒当りの浮動小数点演算の回数

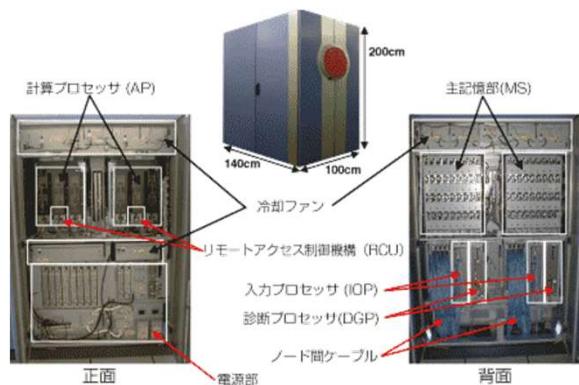


2002年世界最速の計算機

地球シミュレータ (2002)

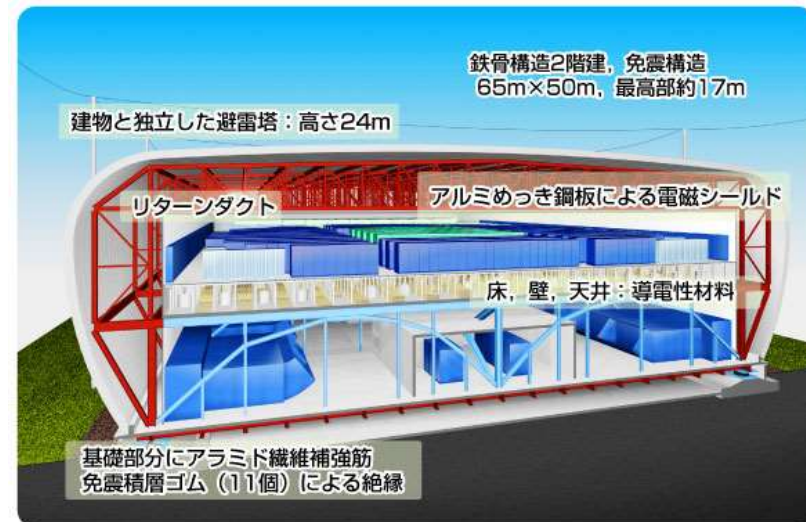


宇宙開発事業団、日本原子力研究所、
海洋科学技術センター (横浜)



5,120 NECベクトルプロセッサ

40 T = 40×10^{12} Flops



50×65 m²

電力: 5 MW 電気代: 年間 5 億円

2004年世界最速の計算機

IBM Blue Gene (2004)



IBM/米国エネルギー省
核安全保障局
Rochester, United States

32,768 PowerPC 440

90 T = 90×10^{12} Flops



電力：1 MW

2011年世界最速の計算機

京 (2011)

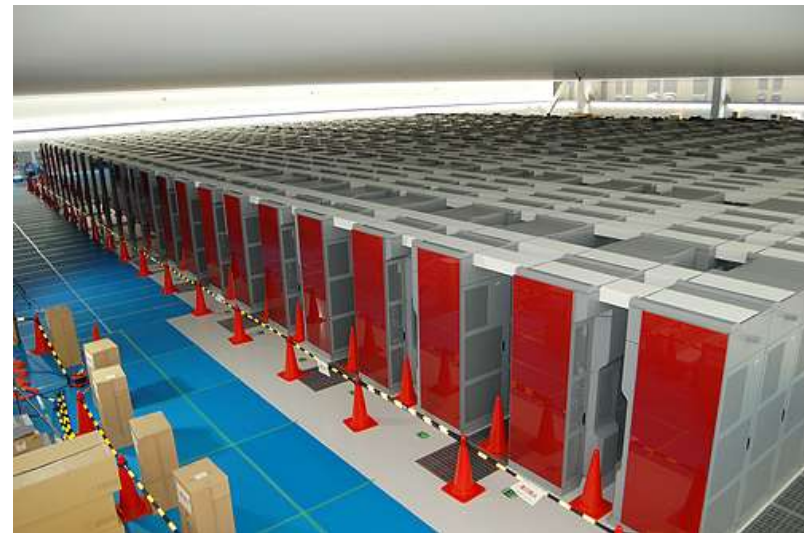


理化学研究所 (神戸)

68,544 スカラープロセッサ

$8162\text{ T} = 8162 \times 10^{12}\text{ Flops}$

8 コア / 1CPU



電力：12.6 MW 電気代：年間30億円

2013年世界最速の計算機

天河2 2013 年



中国国防科学技術大学
(湖南省長沙市)

16,000 Intel Xeon IvyBridge Xeon Phi

33,860 T = 33.86×10^{15} Flops 321万コア



電力 : 17.8 MW

Intel Xeon Phi
60 コア/1枚



江蘇省 無錫市



神威・太湖之光 2016 年



中国国家並列計算機工学技術研究センター
(無錫市)

2017年世界最速の計算機

40960 CPU SW26010 1,065万 コア

93,015 T = 93.01×10^{15} Flops



電力: 15.4 MW

1CPU当たり260コア



2018年世界最速の計算機



Summit 2018年6月
オークリッジ国立研究所（米国）

4,608ノード IBM POWER9 22C

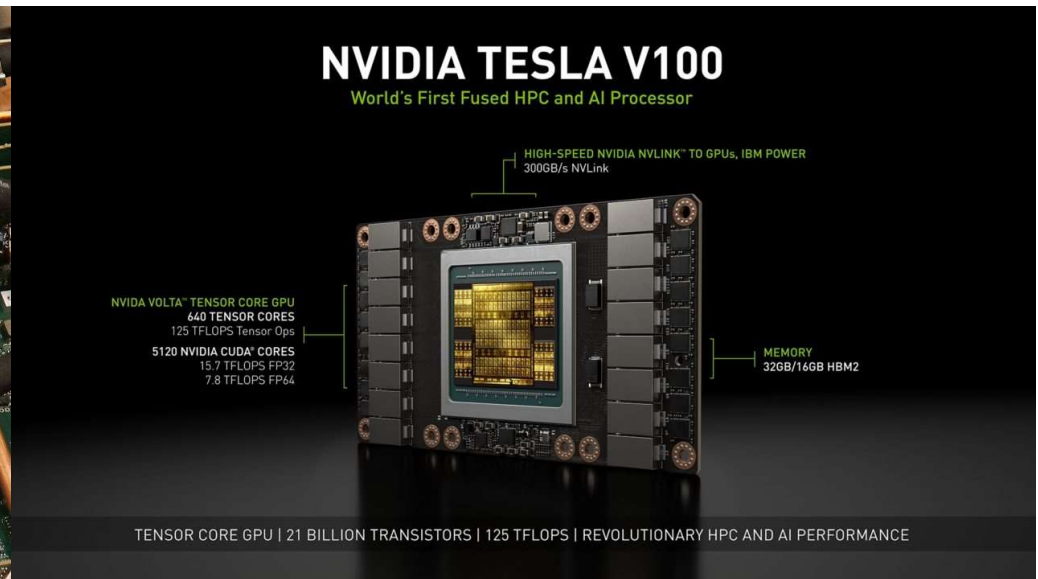
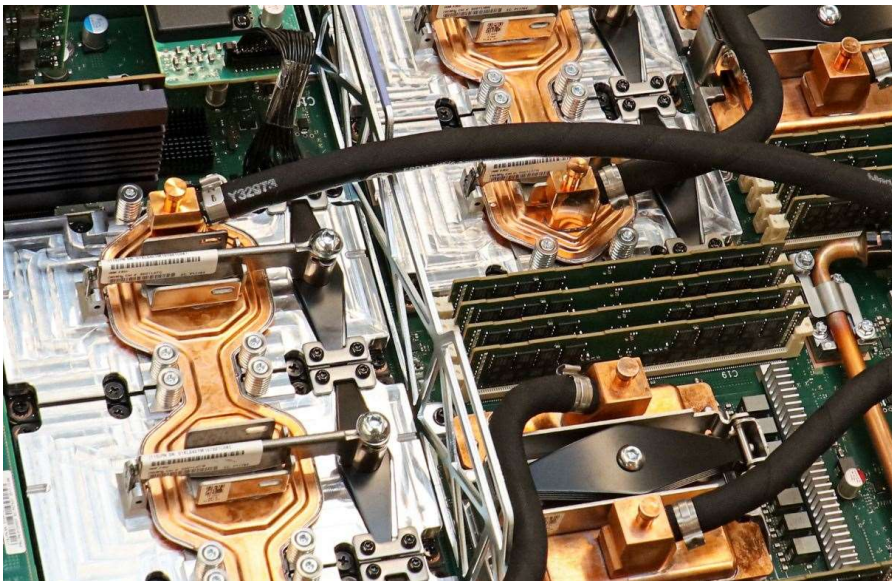
27,648 基の NVIDIA GPU

$122,300\text{ T} = 122.3 \times 10^{15}\text{ Flops}$

電力: 8.8 MW

27,648 基の NVIDIA GPU

1 基当たり 5120 GPUコア

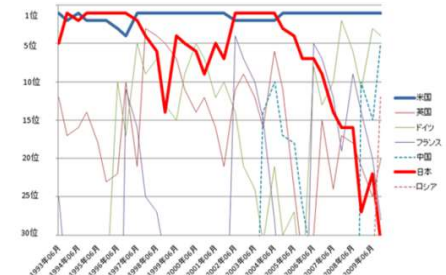


過去25年の歴代最速計算機

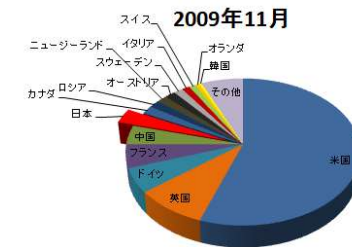
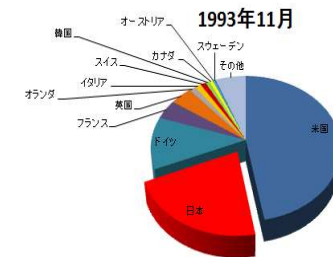
2018 年	2018.6 – (アメリカ合衆国) IBM Summit	122300 TFlops
	2016.6 – 2017.11 (中国) NRCPC 神威	93014 TFlops
2013 年	2013.6 - 2016.6 (中国) NUDT 天河二号	33863 TFlops
	2012.6 – 2012.11 (アメリカ合衆国) IBM Blue Gene/Q	16325 TFlops
	2011.6 – 2011.11 (日本) 富士通 京	10510 TFlops
2010 年	2010.11 - 2011.6 (中国) NUDT 天河一号A	2566 TFlops
	2009.11 - 2010.11 (アメリカ合衆国) Cray Jaguar	1759 TFlops
2005 年	2008.06 - 2009.11 (アメリカ合衆国) IBM Roadrunner	1105 TFlops
	2004.11 - 2008.06 (アメリカ合衆国) IBM Blue Gene/L	137 TFlops
	2002.06 - 2004.11 (日本) NEC 地球シミュレータ	36 TFlops
2000 年	2000.11 - 2002.06 (アメリカ合衆国) IBM ASCI White	5 TFlops
	1997.06 - 2000.11 (アメリカ合衆国) Intel ASCI Red	2 TFlops
1995 年	1996.11 - 1997.06 (日本) 日立 CP-PACS	0.4 TFlops
	1996.06 - 1996.11 (日本) 日立 SR2201	0.2 TFlops
	1994.11 - 1996.06 (日本) 富士通 数値風洞	0.17 TFlops
	1994.06 - 1994.11 (アメリカ合衆国) Intel Paragon XP/S 140	0.14 TFlops
1993 年	1993.11 - 1994.06 (日本) 富士通 数値風洞	0.04 TFlops

LINPACK

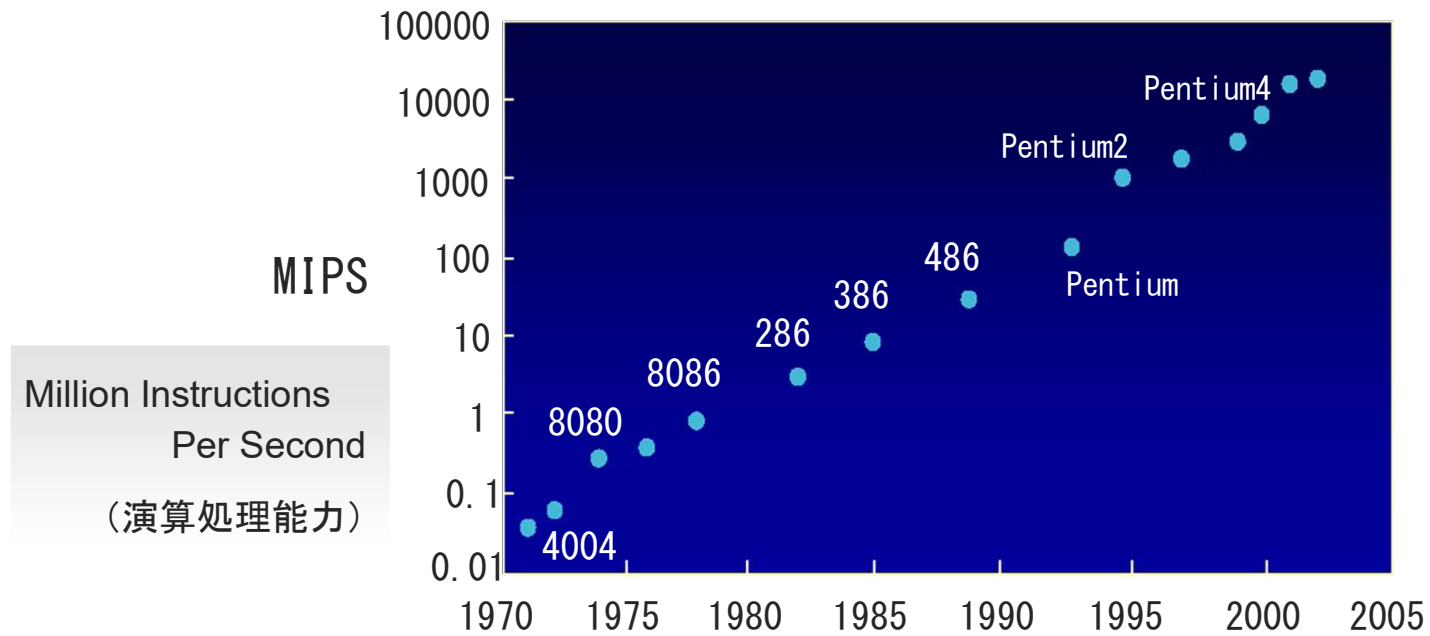
TOP500国別最高順位



TOP500の国別占有率

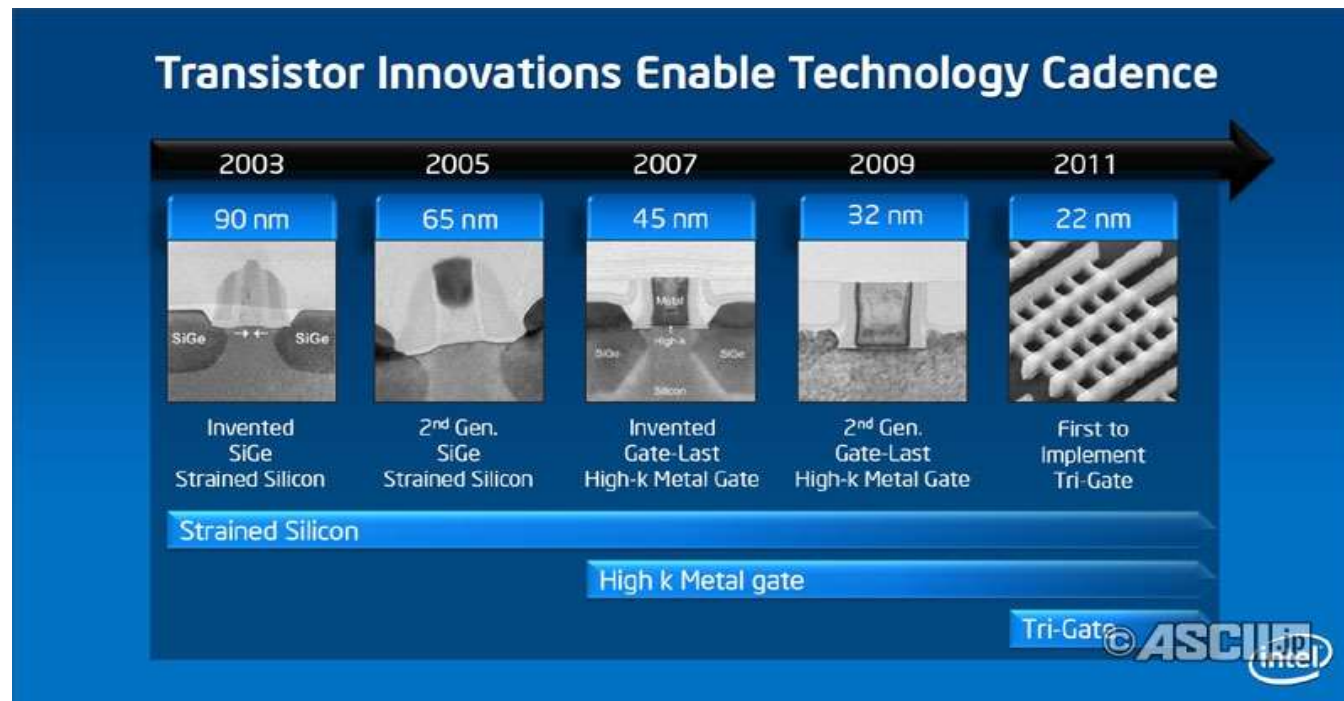


マイクロプロセッサの性能向上

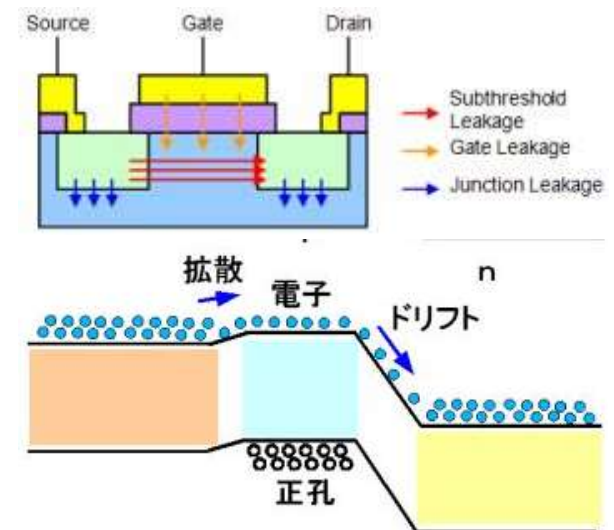


ムーアの法則：
集積回路の集積度が約2年で2倍

微細加工技術の進歩



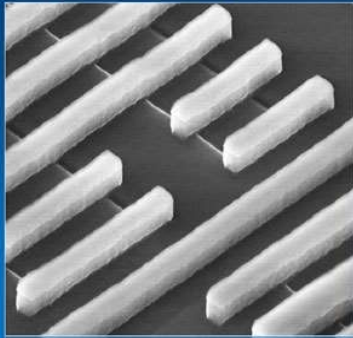
ゲート長			
Pentium 4	Willamett	180 nm	W=80nm (2000)
	Northwood	130 nm	W=60nm (2002)
	Prescott	90 nm	W=50nm (2004)
Core 2	Conroe	65 nm	W=30nm (2006)
	Nehalem	45 nm	W=20nm (2009)
Core i	Sandy Bridge	32 nm	W=15nm (2011)
	Ivy Bridge	22 nm	Tri-Gate (2012)
	Skylake	14 nm	Tri-Gate (2015)
	Cannon Lake	10 nm	Tri-Gate (2018)



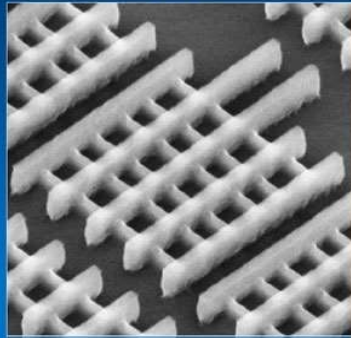
微細加工技術の進歩 Intel

2017年 設備投資費 約120億米ドル

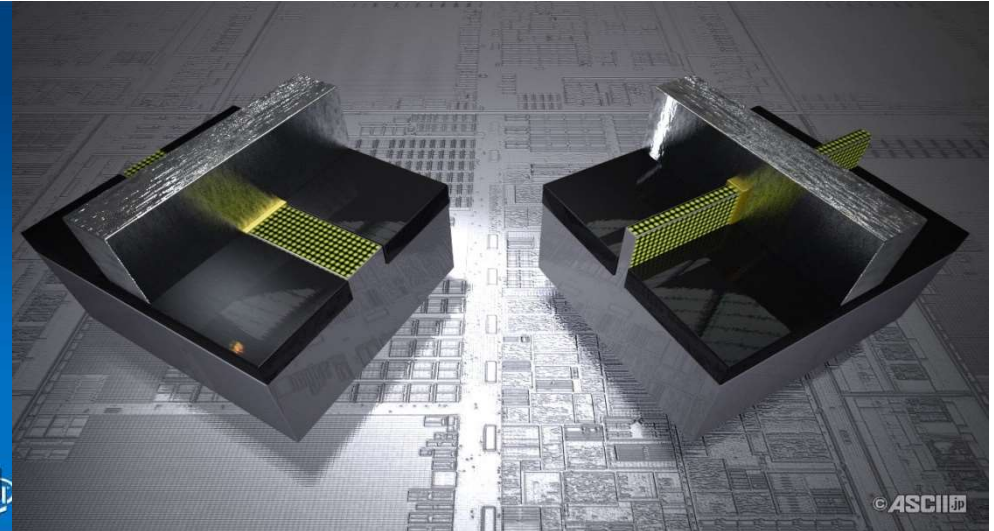
32 nm Planar Transistors



22 nm Tri-Gate Transistors



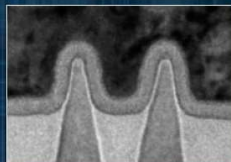
© ASCII



© ASCII

3RD GENERATION FINFETS

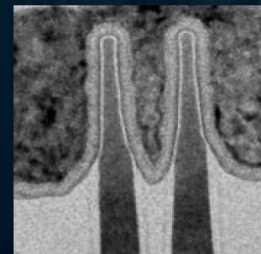
22 NM



14 NM



10 NM



10 nm Fins are ~25% taller and ~25% more closely spaced than 14 nm

TECHNOLOGY AND MANUFACTURING DAY



計算機の構造

Power Unit
電源



HDD
ハードディスク

CPU
中央演算装置

DRAM
メモリ

チップセット

I/O
ビデオ出力装置
ネットワーク

マザーボード

HDD 1 Gbit/mm²



1 bit : 30nm × 30nm



トランジスタ数

CPU	14億8,000万
DRAM	44億 (4Gbit)



C言語、Fortran のソースファイル

C言語 main.c

```
#include <stdio.h>
int main(void){

    int i, j=0;

    for (i=1; i<=10; i++){
        j+=i;
    }
    printf("%5d¥n", j);

    return 0;
}
```

C++言語 main.cpp

```
#include <cstdio>
using namespace std;
int main(void){

    int j=0;

    for (int i=1; i<=10; i++){
        j+=i;
    }
    printf("%5d¥n", j);

    return 0;
}
```

Fortran 77 main.f

```
integer i, j

j=0
do 100 i=1, 10
    j=j+i
    write (*,*) j
end
```

行番号 → 100 continue

Fortran 95 main.f95

```
integer :: i, j=0

do i=1, 10
    j=j+i
end do
write (*,*) j

end
```

文の最後には;を付ける = は代入演算子(等号ではない)

i++ は i=i+1 と同じ意味 j+=i は j=j+i と同じ意味

書式指定子 %5d : 整数型5桁で表示

エスケープシーケンス ¥n : 改行コード

write (*,*)

標準出力

書式指定省略

C 言語と Fortran の比較

型宣言文

	C言語	Fortran
0byte 無し	void	
1byte 整数型(8bit)	char	character
2byte 整数型(16bit)	int (short, short int)	integer
4byte 整数型(32bit)	long int (long)	integer*4
4byte 浮動小数点型(32bit)	float	real
8byte 浮動小数点型(64bit)	double	real*8
16byte 浮動小数点型(128bit)	(long double, __float128)	real*16

演算子

C、Fortran 共通	C言語	Fortran	論理演算子	C言語	Fortran
= 代入演算子			論理積	&&	.AND.
			論理和	 	.OR.
算術演算子	関係(比較)演算子		算術演算子		
* 積	$x < y$	$x.LT. y$	$x * = y$	$x = x * y$	
/ 商	$x \leq y$	$x.LE. y$	$x /= y$	$x = x / y$	
+ 和	$x > y$	$x.GT. y$	$x += y$	$x = x + y$	
- 差	$x \geq y$	$x.GE. y$	$x -= y$	$x = x - y$	
	$x == y$	$x.EQ. y$	べき乗	$pow(x, y)$	$x ** y$
	$x \neq y$	$x.NE. y$	剰余	$x \% y$	$mod(x, y)$

引数や返却値の型と内部表現

無し (NULL) **void**

表現する値

1byte 整数型(8bit) **char** 0 ~ 255 ASCII コード、shift-JIS コード



0100 0111 **G** 71
1011 1001 **ケ** 185

2byte 整数型(16bit) **int (short, short int)** -32,768 ~ 32,767



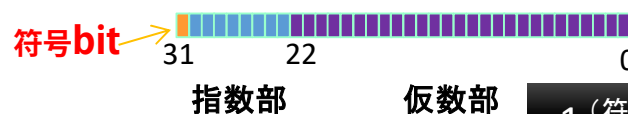
0000 0000 0100 0111 **71**
1111 1111 1011 1001 **-71**

4byte 整数型(32bit) **long int (long)** -2,147,483,648 ~ 2,147,483,647



0000 0000 0000 0000 0000 0000 0100 0111 **71**
1111 1111 1111 1111 1111 1111 1011 1001 **-71**

4byte 浮動小数点型(32bit) **float** $\pm 3.4028234 \times 10^{38} \sim \pm 1.175494351 \times 10^{-38}$



0011 1110 1000 0000 0000 0000 0000 0000 **0.25**
125

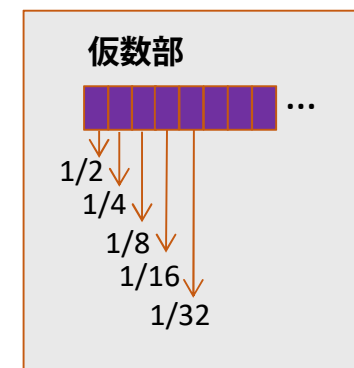
$-1 \text{ (符号bit)} \times (1 + \text{仮数部}) \times 2^{(\text{指数部}-127)}$ IEEE 754 形式

8byte 浮動小数点型(64bit) **double** $\pm 1.7976931348623157 \times 10^{308} \sim \pm 2.2250738585072014 \times 10^{-308}$



$-1 \text{ (符号bit)} \times (1 + \text{仮数部}) \times 2^{(\text{指数部}-1023)}$ IEEE 754 形式

16byte 浮動小数点型(128bit) **long double _float128**



文字コード表

ASCIIコード 1 byte コード(半角)

	00	10	20	30	40	50	60	70
00	NUL	DLE	SPACE	0	@	P	`	p
01	SOH	DC1	!	1	A	Q	a	q
02	STX	DC2	"	2	B	R	b	r
03	ETX	DC3	#	3	C	S	c	s
04	EOT	DC4	\$	4	D	T	d	t
05	ENQ	NAK	%	5	E	U	e	u
06	ACK	SYN	&	6	F	V	f	v
07	BEL	ETB	'	7	G	W	g	w
08	BS	CAN	(	8	H	X	h	x
09	HT	EM	)	9	I	Y	i	y
0A	LF	SUB	*	:	J	Z	j	z
0B	VT	ESC	+	;	K	[	k	{
0C	FF	FS	,	<	L	\	l	
0D	CR	GS	-	=	M	]	m	}
0E	SO	RS	.	>	N	^	n	~
0F	SI	US	/	?	O	_	o	DEL

シフトJISコード

1 byte コード(半角)

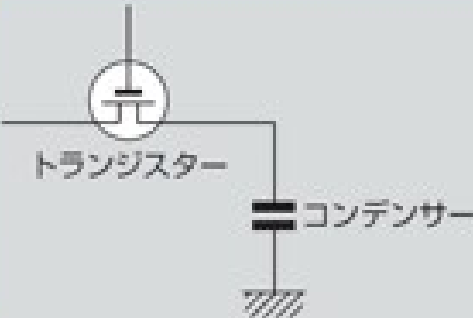
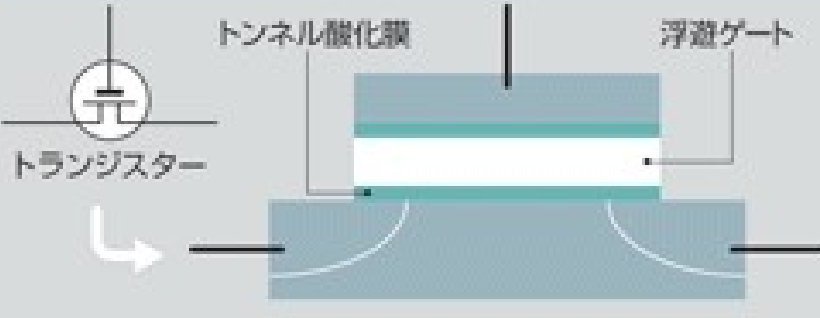
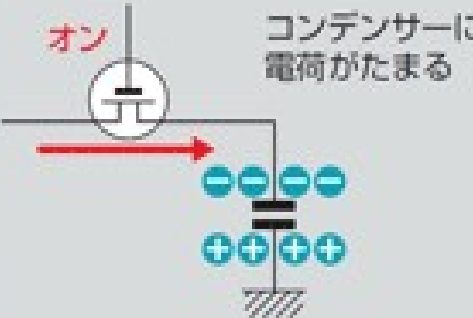
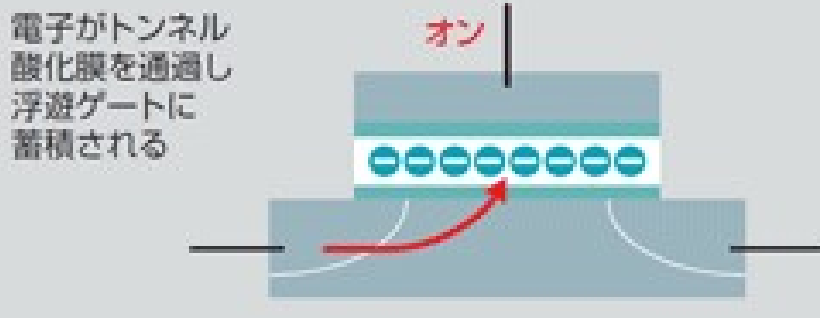
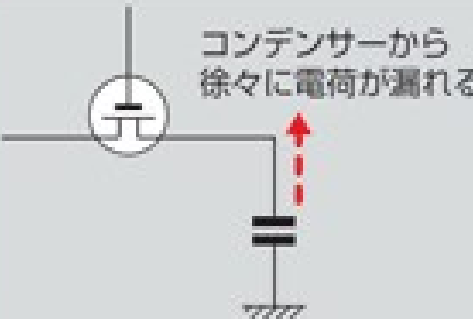
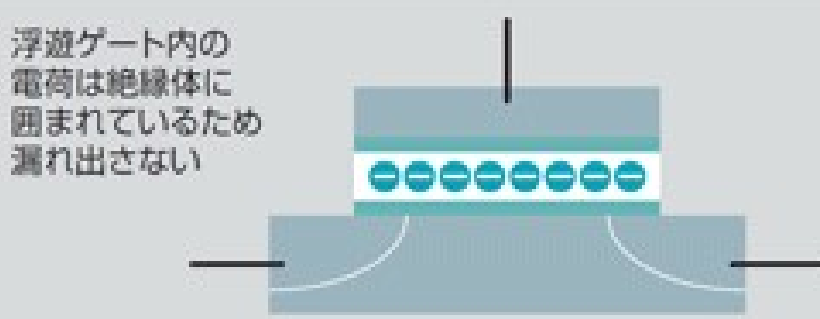
	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
0	NUL			0	@	P	`	p				ー	タ	ミ		
1			!	1	A	Q	a	q			。	ア	チ	ム		
2			"	2	B	R	b	r			「	イ	ツ	メ		
3			#	3	C	S	c	s			」	ウ	テ	モ		
4			\$	4	D	T	d	t			、	エ	ト	ヤ		
5			%	5	E	U	e	u			・	オ	ナ	ユ		
6			&	6	F	V	f	v			ヲ	カ	ニ	ヨ		
7	BEL		'	7	G	W	g	w			ア	キ	ヌ	ラ		
8	BS		(	8	H	X	h	x			イ	ク	ネ	リ		
9	HT		)	9	I	Y	i	y			ウ	ケ	ノ	ル		
A	LF		*	:	J	Z	j	z			エ	コ	ハ	レ		
B	VT	ESC	+	;	K	[	k	{			オ	サ	ヒ	ロ		
C	FF	→	,	<	L	¥	l				ヤ	シ	フ	ワ		
D	CR	←	-	=	M	]	m	}			ユ	ス	ヘ	ン		
E		↑	.	>	N	^	n	~			ヨ	セ	ホ	ゝ		
F		↓	/	?	O	_	o	DEL			ッ	ソ	マ	。°		

シフトJIS 漢字コード表

2 byte コード(全角)

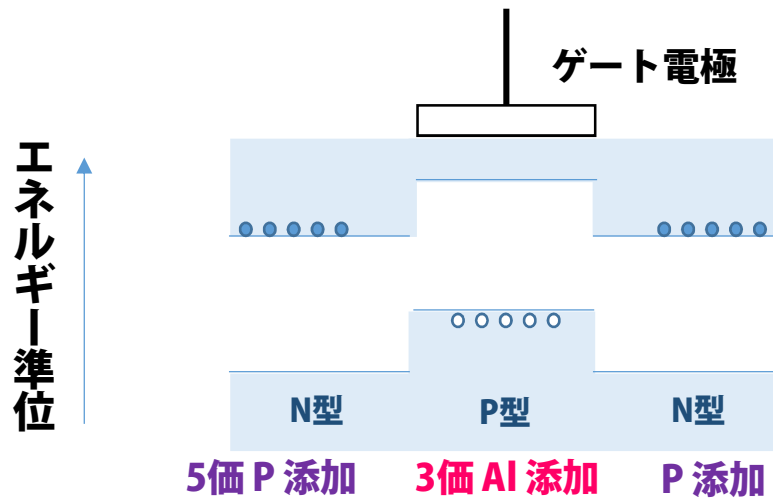
	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
8880	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	亜
88A0	啞	娃	阿	哀	愛	挨	始	逢	葵	茜	穉	惡	握	渥	旭	葦	芦	鰺	梓	圧	幹	扱	宛	姐	虻	飴	絢	綾	鮎	或	粟	裕
88C0	安	庵	按	暗	案	闇	鞍	杏	以	伊	位	依	偉	囧	夷	委	威	尉	惟	意	慰	易	椅	為	畏	異	移	維	緯	胃	萎	衣
88E0	謂	違	遺	医	井	亥	域	育	郁	磯	一	壺	溢	逸	稻	茨	芋	鰯	允	印	咽	員	因	姻	引	飲	淫	胤	蔭			

DRAM, Flash メモリーの構造

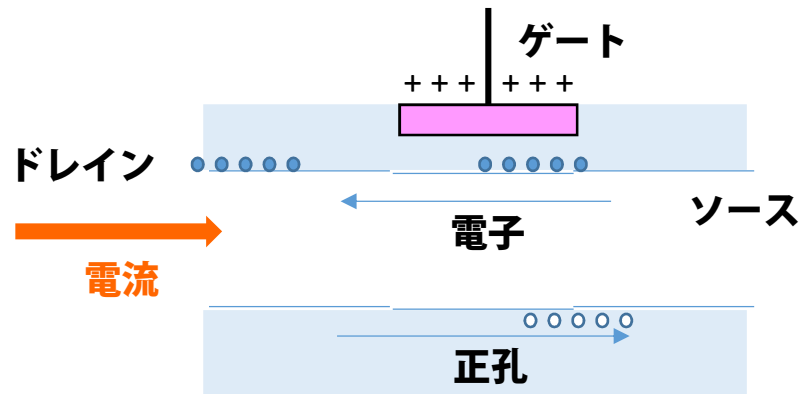
	DRAM	フラッシュメモリ
原理	 トランジスター コンデンサー	 トンネル酸化膜 浮遊ゲート トランジスター
情報を記憶させるとき	オン  コンデンサーに電荷がたまる	オン  電子がトンネル酸化膜を通過し浮遊ゲートに蓄積される
回路をオフにしたとき	 コンデンサーから徐々に電荷が漏れる	 浮遊ゲート内の電荷は絶縁体に囲まれているため漏れ出さない

トランジスタ動作原理

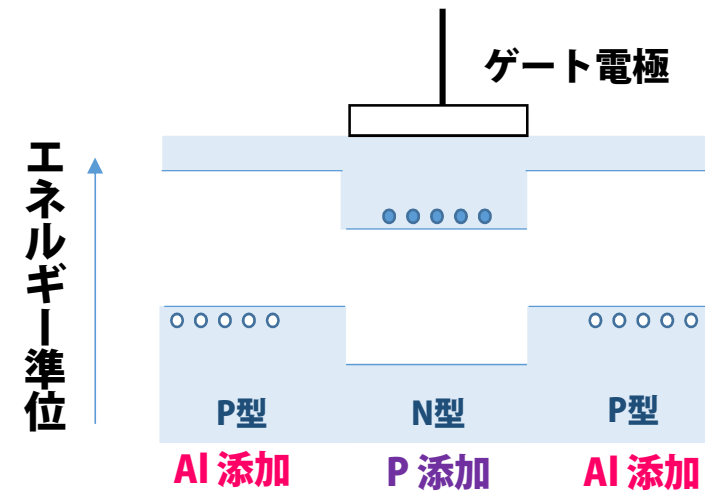
NPN型トランジスタ



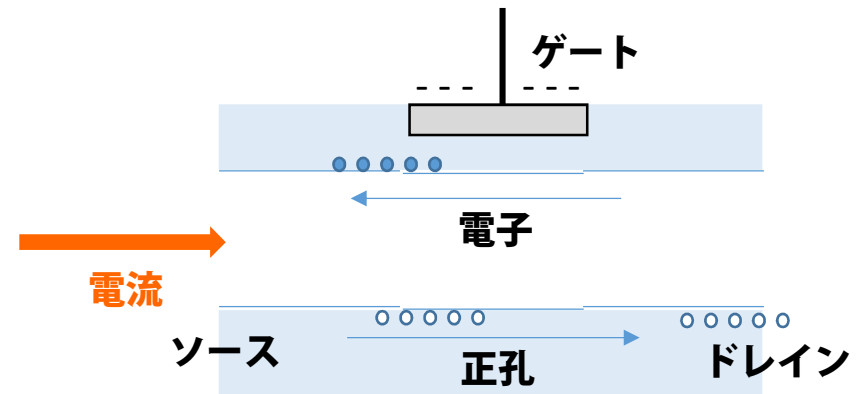
正電圧をゲートに印加 OFF → ON



PNP型トランジスタ



負電圧をゲートに印加 OFF → ON



<http://www.ktf-split.hr/periodni/en/>

(1) Pure Appl. Chem., 73, No. 4, 667-683 (2001)
Relative atomic mass is shown with five significant figures. For elements have no stable nuclides, the value enclosed in brackets indicates the mass number of the longest-lived isotope of the element.
However three such elements (Th, Pa, and U) do have a characteristic terrestrial isotopic composition, and for these an atomic weight is tabulated.

LANTHANIDE

57 138.91

58 140.12

59 140.91

60 144.24

61 (145)

62 150.36

63 151.96

64 157.25

65 158.93

66 162.50

67 164.93

68 167.26

69 168.93

70 173.04

71 174.97

La

Ce

Pr

Nd

Pm

Sm

Eu

Gd

Tb

Dy

Ho

Er

Tm

Yb

Lu

LANTHANUM

CERIUM

PRASEODYMIUM

NEODYMIUM

PROMETHIUM

SAMARIUM

EUROPIUM

GADOLINIUM

TERBIUM

DYSPROSIUM

HOLMIUM

ERBIUM

THULIUM

YTTTERBIUM

LUTETIUM

ACTINIDE

89 (227)

90 232.04

91 231.04

92 238.03

93 (237)

94 (244)

95 (243)

96 (247)

97 (247)

98 (251)

99 (252)

100 (257)

101 (258)

102 (259)

103 (262)

Ac

Th

Pa

U

Np

Pu

Am

Cm

Bk

Cf

Es

Fm

Md

No

Lr

ACTINIUM

THORIUM

PROTACTINIUM

URANIUM

NEPTUNIUM

PLUTONIUM

AMERICIUM

CURIUM

BERKELIUM

CALIFORNIUM

EINSTEINIUM

FERMIUM

MENDELEVIUM

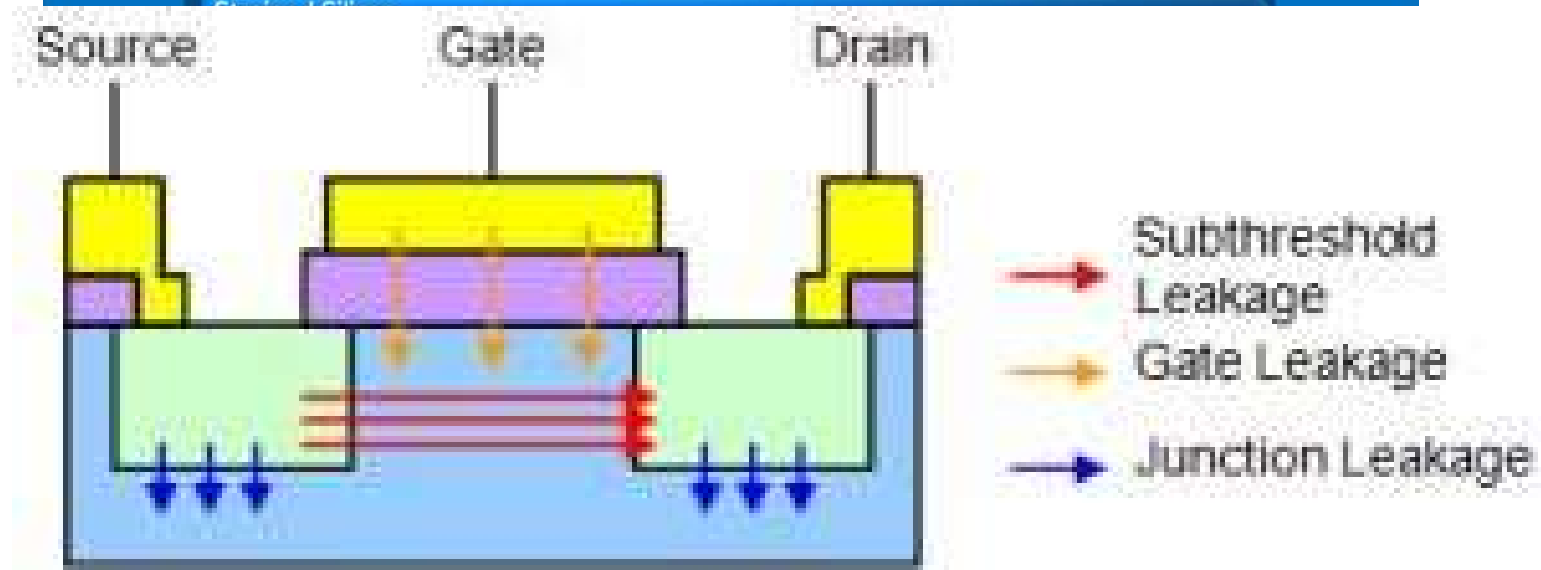
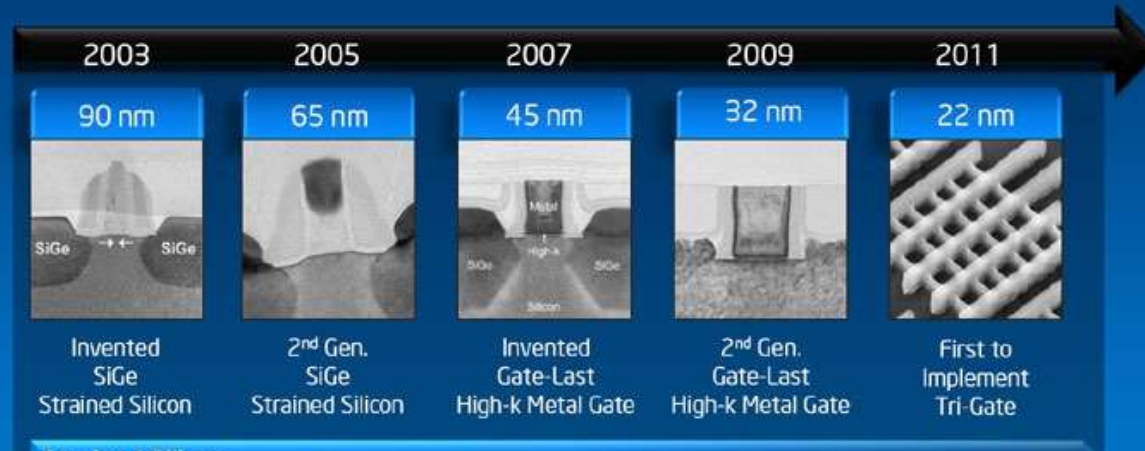
NOBELIUM

LAWRENCIUM

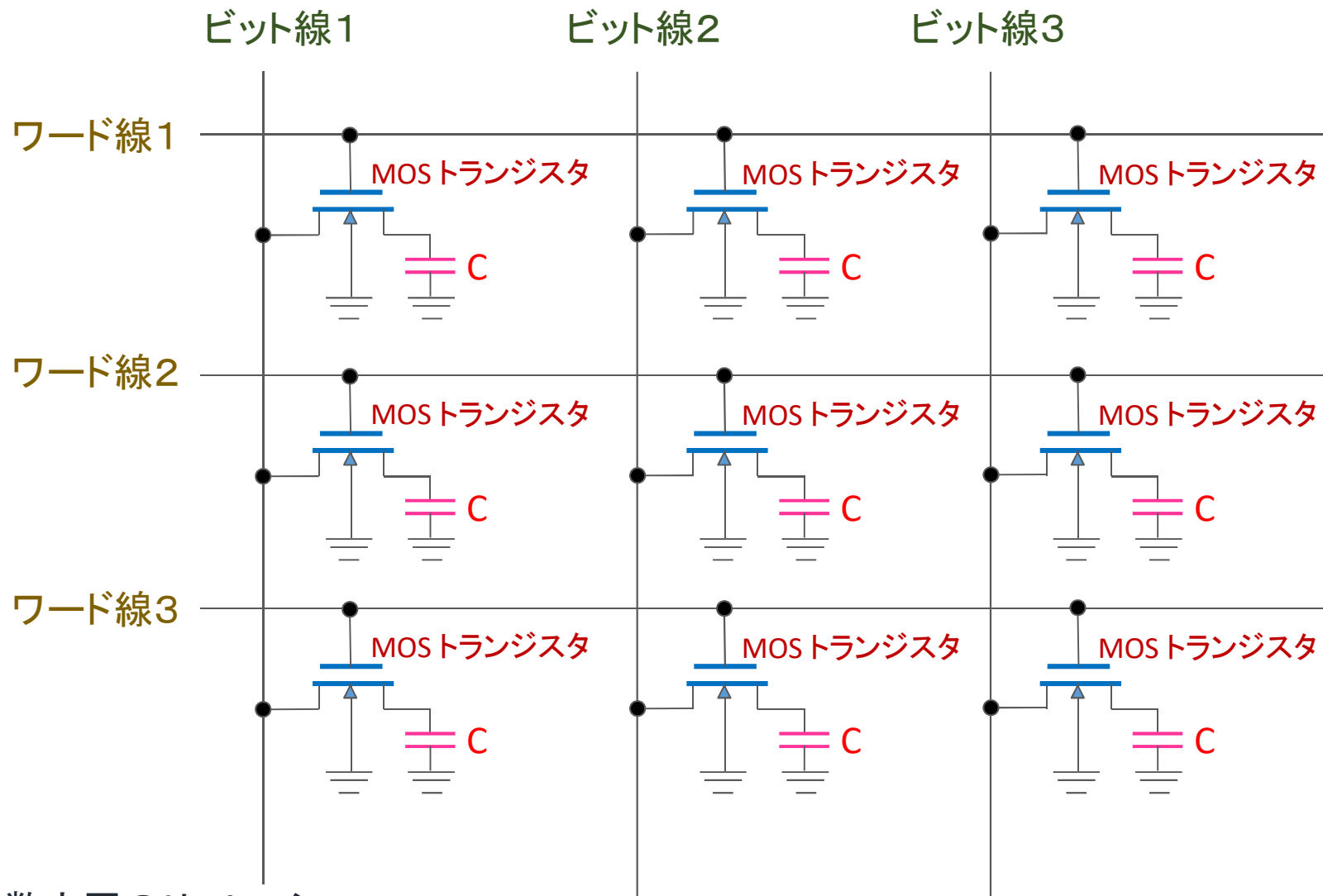
Copyright © 1998-2003 EniG. (eni@ktf-split.hr)

MOSトランジスタの構造

Transistor Innovations Enable Technology Cadence

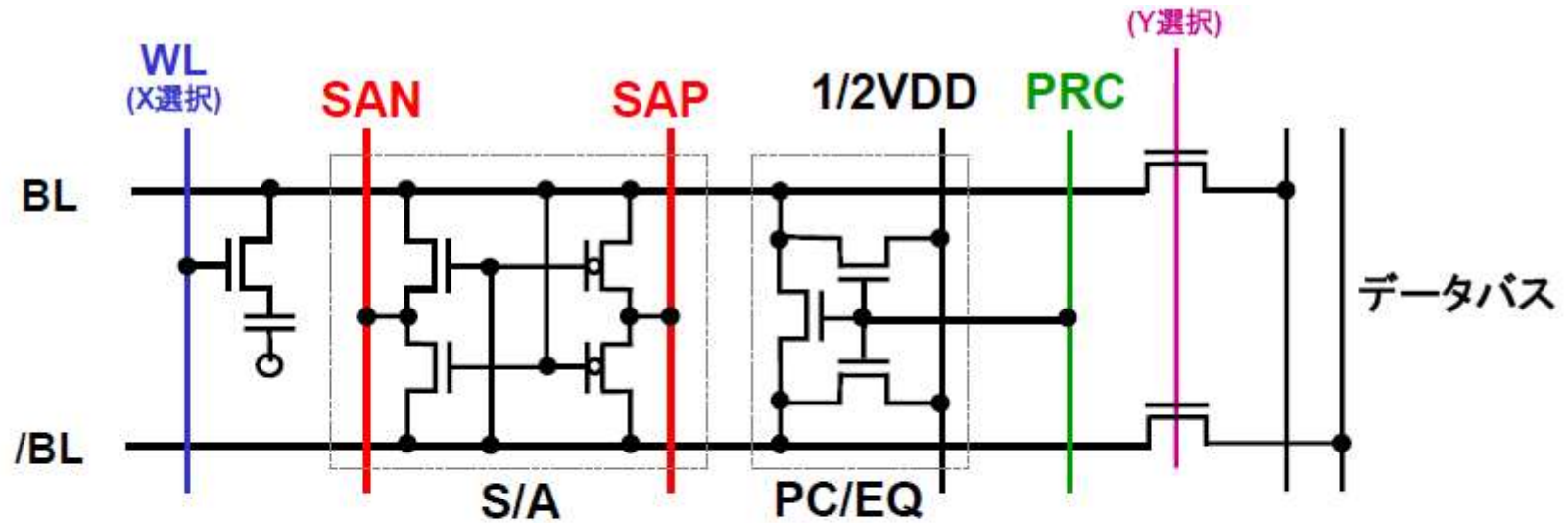


DRAM (Dynamic Random Access Memory) の構造



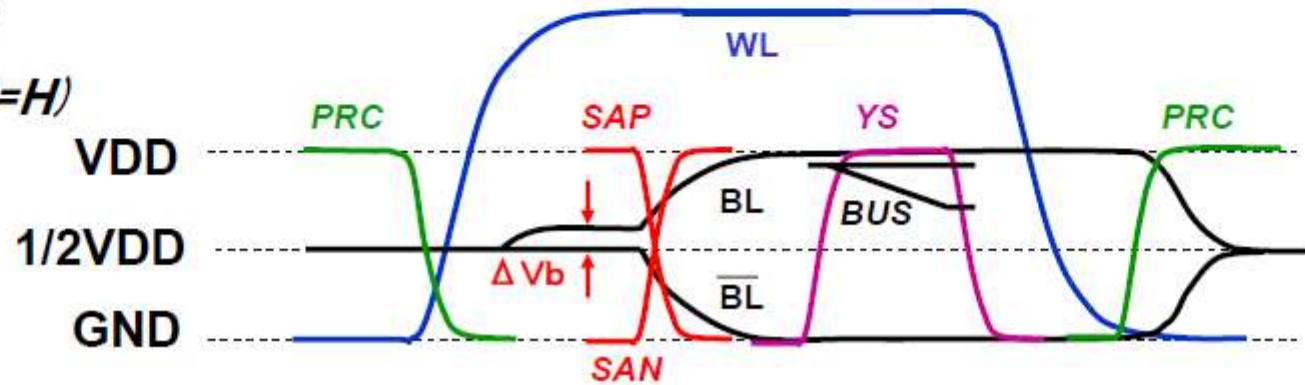
1秒間に数十回のリフレッシュ

DRAM の動作の仕組み

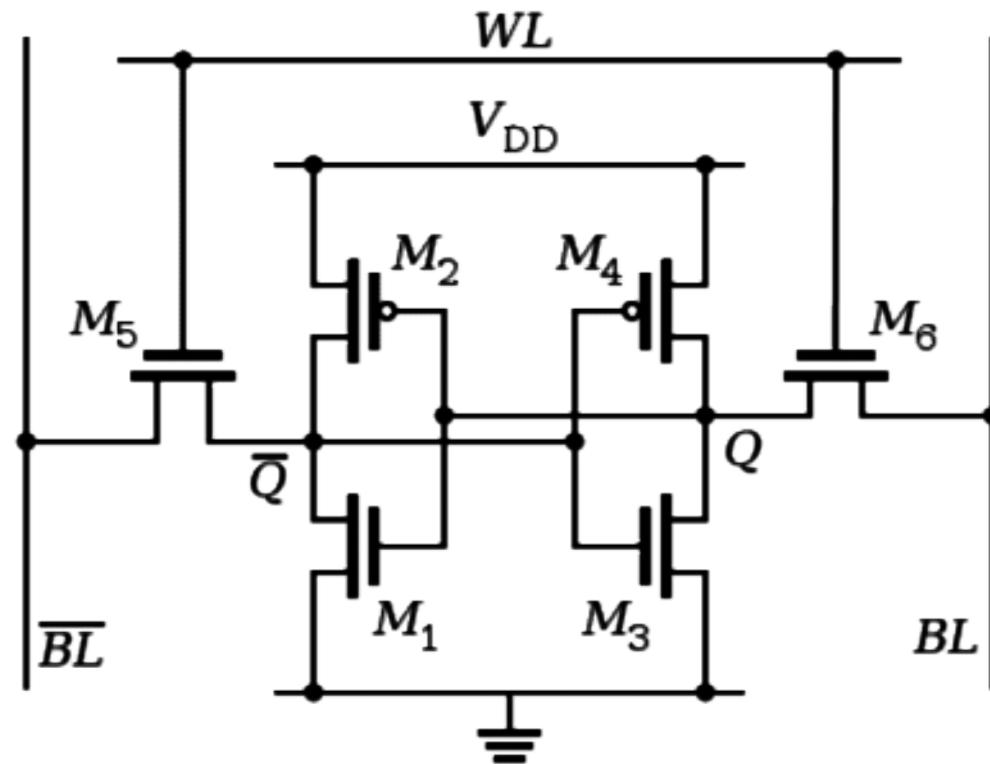


動作波形

(read, cell=H)



SRAMの構造



CPUの構造

クロック数：処理速度に比例

コアの数（thread 数）

NEC：ベクトルプロセッサ

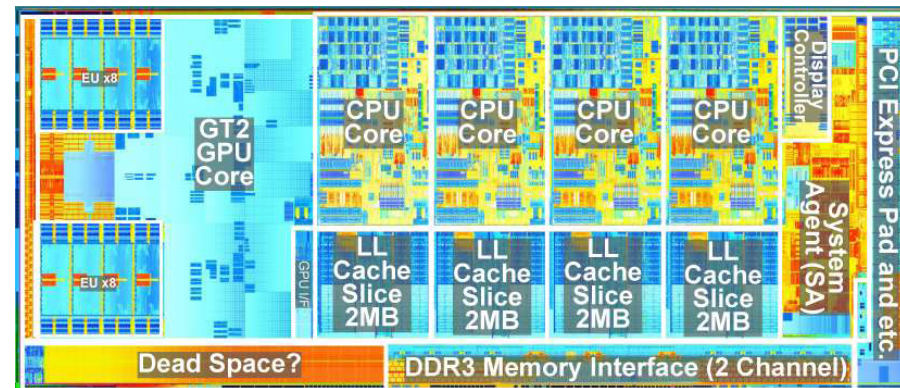
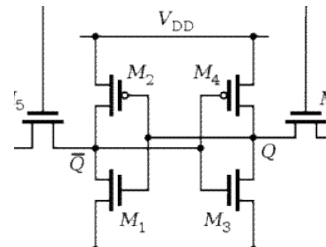
Intel，AMD：スカルプロセッサ

キャッシュメモリー
(SRAM の容量)

L1：32 KB

L2：256 KB

L3：2024 KB



14億8,000万個のトランジスタ

C言語、Fortran のソースファイル

C言語 main.c

```
#include <stdio.h>
int main(void){

    int i, j=0;

    for (i=1; i<=10; i++){
        j+=i;
    }
    printf("%5d¥n", j);

    return 0;
}
```

C++言語 main.cpp

```
#include <cstdio>
using namespace std;
int main(void){

    int j=0;

    for (int i=1; i<=10; i++){
        j+=i;
    }
    printf("%5d¥n", j);

    return 0;
}
```

Fortran 77 main.f

```
integer i, j

j=0
do 100 i=1, 10
    j=j+i
    write (*,*) j
end
```

行番号 → 100 continue

Fortran 95 main.f95

```
integer :: i, j=0

do i=1, 10
    j=j+i
end do
write (*,*) j

end
```

文の最後には;を付ける = は代入演算子(等号ではない)

i++ は i=i+1 と同じ意味 j+=i は j=j+i と同じ意味

書式指定子 %5d : 整数型5桁で表示

エスケープシーケンス ¥n : 改行コード

write (*,*)

標準出力

書式指定省略

C言語 と Fortran の構文比較

条件分岐 式が真ならその後の文を、式が偽ならelse 以下の文を実行

C言語

```
if ( 式 ) 文  
if ( 式 ) { 文 文 文 ... }  
if ( 式 ) { 文1 ... } else { 文2 ... }  
if ( 式1 ) { 文1 ... } else if ( 式2 ) { 文2 ... } else { 文3 ... }
```

C言語では、0は偽であり0以外は真になる

Fortran

```
if ( 式 ) then  
  文  
  文  
end if  
if ( 式 ) then  
  文1  
  文1  
else  
  文2  
  文2  
end if  
if ( 式1 ) then  
  文1  
  文1  
else if ( 式2 )  
  文2  
  文2  
end if
```

繰り返し 式₂が真である限り文を実行

```
for ( 式1; 式2; 式3 ) { 文 文 ... }
```

式₁ 初期式
式₂ 評価式
式₃ 評価後に実行される式

```
do 値1, 値2, 値3  
  文  
  文  
end do
```

値₁ 初期値
値₂ 最終値
値₃ 刻み幅

Fortranでは、値₃ が略された場合、初期値のパラメータを1増やす

式が真である限り、文を繰り返し実行

C言語

評価が先 while (式) { 文 文 ... }

do { 文 文 ... } while (式); **評価が後** (必ず一回は文を実行)

C 言語と Fortran の比較

型宣言文

	C言語	Fortran
0byte 無し	void	
1byte 整数型(8bit)	char	character
2byte 整数型(16bit)	int (short, short int)	integer
4byte 整数型(32bit)	long int (long)	integer*4
4byte 浮動小数点型(32bit)	float	real
8byte 浮動小数点型(64bit)	double	real*8
16byte 浮動小数点型(128bit)	(long double, __float128)	real*16

演算子

C、Fortran 共通	C言語	Fortran	論理演算子	C言語	Fortran
= 代入演算子			論理積	&&	.AND.
			論理和	 	.OR.
算術演算子	関係(比較)演算子		算術演算子		
* 積	$x < y$	$x.LT. y$	$x * = y$	$x = x * y$	
/ 商	$x \leq y$	$x.LE. y$	$x /= y$	$x = x / y$	
+ 和	$x > y$	$x.GT. y$	$x += y$	$x = x + y$	
- 差	$x \geq y$	$x.GE. y$	$x -= y$	$x = x - y$	
	$x == y$	$x.EQ. y$	べき乗	$pow(x, y)$	$x ** y$
	$x \neq y$	$x.NE. y$	剰余	$x \% y$	$mod(x, y)$