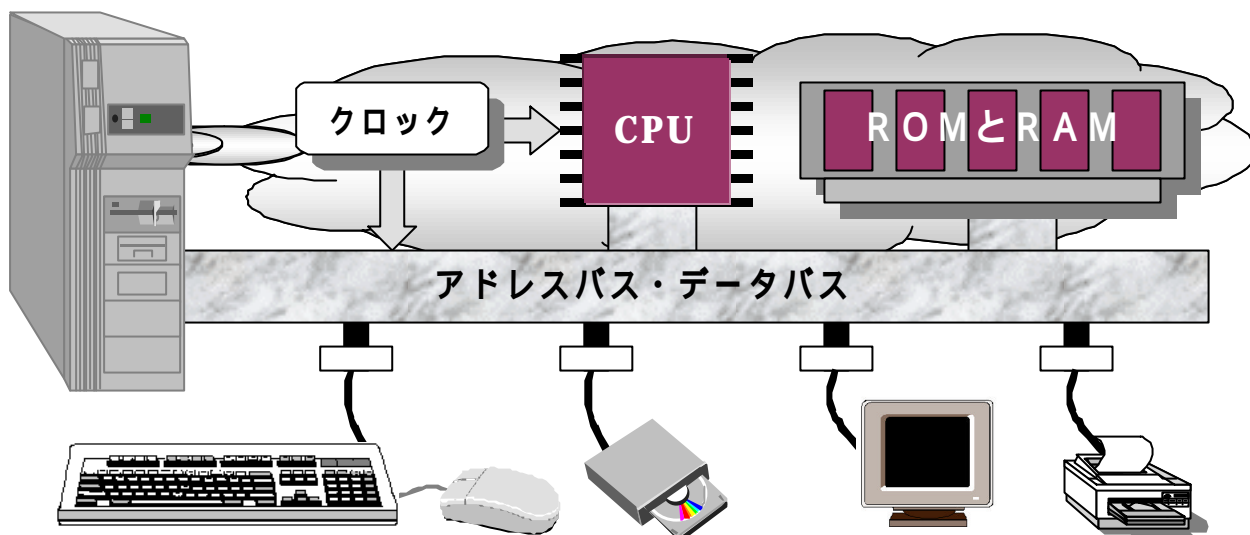


# 第二章 コンピュータの構成要素

## コンピュータ装置



制御機能	CPU	クロック	データは光のスピードでコンピュータの”神経細胞(プリント基盤)”内を飛び交うが、いつ出て、いつ着くかのタイミングをとる。100万分の1秒の単位でのタイミングになる。
演算機能		演算ユニット	+ - × ÷の計算をする。
記憶機能		ROMとRAM CDROM	磁気力でデータ記憶し計算にまわす。
入力機能		キーボード マウス	人間がコンピュータに命令するためのインターフェース。
出力機能		画面 プリンター	コンピュータからの結果を人間にわかるように表示する。

CPU : Central Processing Unit (中央処理装置)

ROM : Read Only Memory (読み込み専用メモリ)

RAM : Random Access Memory (読み書き自由メモリ)

CDROM : Compact Disk ROM (携帯用読み込み専用ディスクメモリ)

# コンピュータの性能

## 実行速度（速さ）：CPU

タイミングが短いほど速い：一秒間に計算する回数

PlayStation2 の Emotion Engine：300MHz

= 一秒間に 3 億回の計算

Hz（ヘルツ）	一秒間の振動数
K（キロ）	千倍 1km=1000m
M（メガ）	百万倍 1Mton=1,000,000ton
G（ギガ）	10億倍 1Gyear=1,000,000,000year

300MHz=300 × 百万 = 3 億

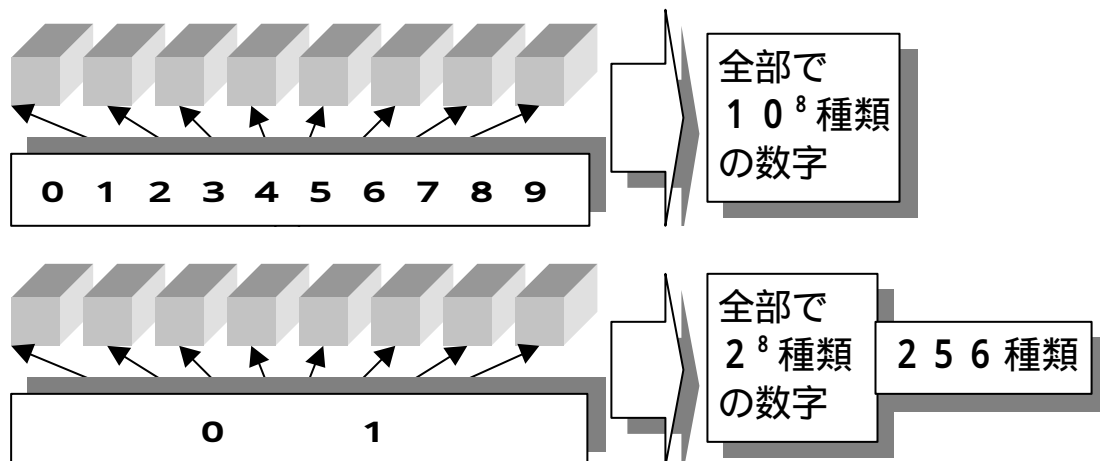
## データ記憶量（量）：ROM・RAM

記憶できる文字・数値の数(0か1)が多いほど性能がよい

PlayStation2 の Emotion Engine：32MB

= 32 × 百万倍 = 3200万の大きさの記憶量

B（バイト）	8桁の2進数：00000000～11111111
	8桁の10進数：00000000～99999999



Bit（ビット） 2進数桁数：8桁 = 8ビット = 1バイト

32MB=32 × 百万倍バイト = 3200万種類の文字

# なぜ8ビットが単位？

使いやすいコンピュータは英語が打てること

コンピュータが英語や数字や記号などを覚えていること

小文字・大文字で26文字×2	52文字
数字	10文字
演算「+ - * / = < >」	7文字
括弧「( ) { } [ ]」	6文字
単位「\$ %」	2文字
英語文章「、 . ! ? " ' & 空白」	8文字
その他記号で「# ^   @」	4文字
画面上を移動する「」	4文字
合計	93文字

約100文字程度

2進数7桁  $2^7 = 128$ 文字 7ビット？でいい！？

最初のチップ：2ビット

次のチップ：2ビットを一緒にして4ビット

その次のチップ：4ビットを一緒にして8ビット

その次のチップ：8ビットを一緒にして16ビット

その次のチップ：16ビットを一緒にして32ビット

必ず倍々になる

8ビットで英語が打てるコンピュータができる！！！！

8ビットが一塊にして、1バイト

Aは65で「01000001」：英語一文字分の容量

全部で256 (=  $2^8$ )文字分

日本語は？「かな」なら51文字 + 50文字でOK!!!

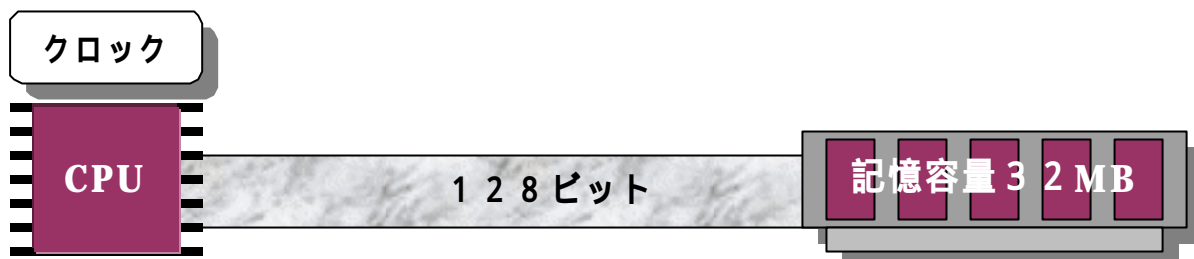
漢字はどうする？？？ 1000文字以上ある？

# データ移動量（量）：メモリバス、データバス

記憶したデータの移動用道幅、広いほど性能がよい

PlayStation2 の Emotion Engine：128ビット

128ビット = 16バイト



機種名	バス幅
ファミコン	8ビット = 1バイト
スーパーファミコン	16ビット = 2バイト
PlayStation	32ビット = 4バイト
Nintendo64	64ビット = 8バイト
DreamCast	128ビット = 16バイト
PlayStation2	

一秒間にどのくらいのデータを処理するか？

PlayStation2：300MHz × 16バイト

= 一秒間に3億回 × 一回に16バイト分

= 一秒間に48億バイト分

機種名	MHz	バイト	一秒間の転送量
ファミコン	3.1	1	300万バイト
スーパーファミコン	3.58	2	700万バイト
PlayStation	33.8	4	1億3500万バイト
Nintendo64	93.75	8	7億5000万バイト
DreamCast	200	16	32億バイト
PlayStation2	300		48億バイト

## コンピュータの機能とOS

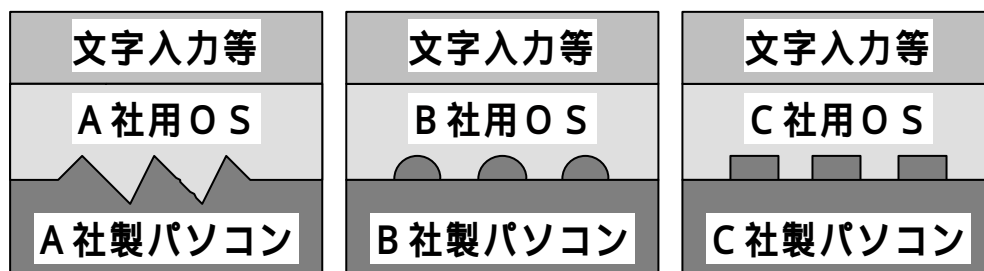
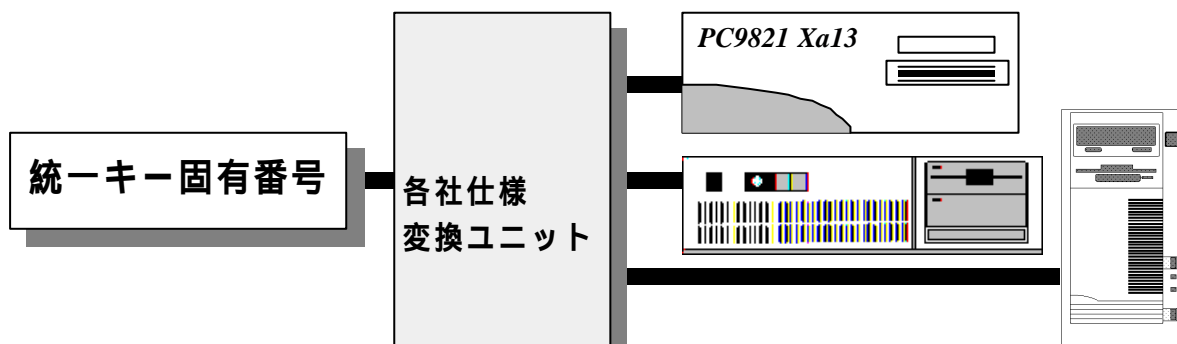
### どのように画面に文字や絵が出るの

- 1) コンピュータがキーボードやマウスやハードディスクがついていることを認識する(BIOS:基本入出力管理)。色々なところからデータが入ってくる。
- 2) キーボードから入力すると
  - キー固有番号を読み取り
  - キーボード上の文字(Aなど)を文字番号(65など)に変換
  - 対応する文字データを記憶領域から探し
  - 画面表示用の記憶領域(VRAM:ビデオ専用メモリ)に転送
  - 画面のドットをそれぞれON

### 会社によって違いがある...

- キー固有番号の割り振り方が違う
- 文字データの記憶領域が違う
- 画面表示用の記憶領域が違う

### 不便である... 違いを吸収する。



# OSの例

MSDOS ( MicroSoft Disk Operating System )

MS Windows 系列 ( Windows3.1, Windows95, Windows98 )

MS WindowsNT 系列 ( Windows New Technology )

MacOS ( Macintosh Operating System )

UNIX, Linux

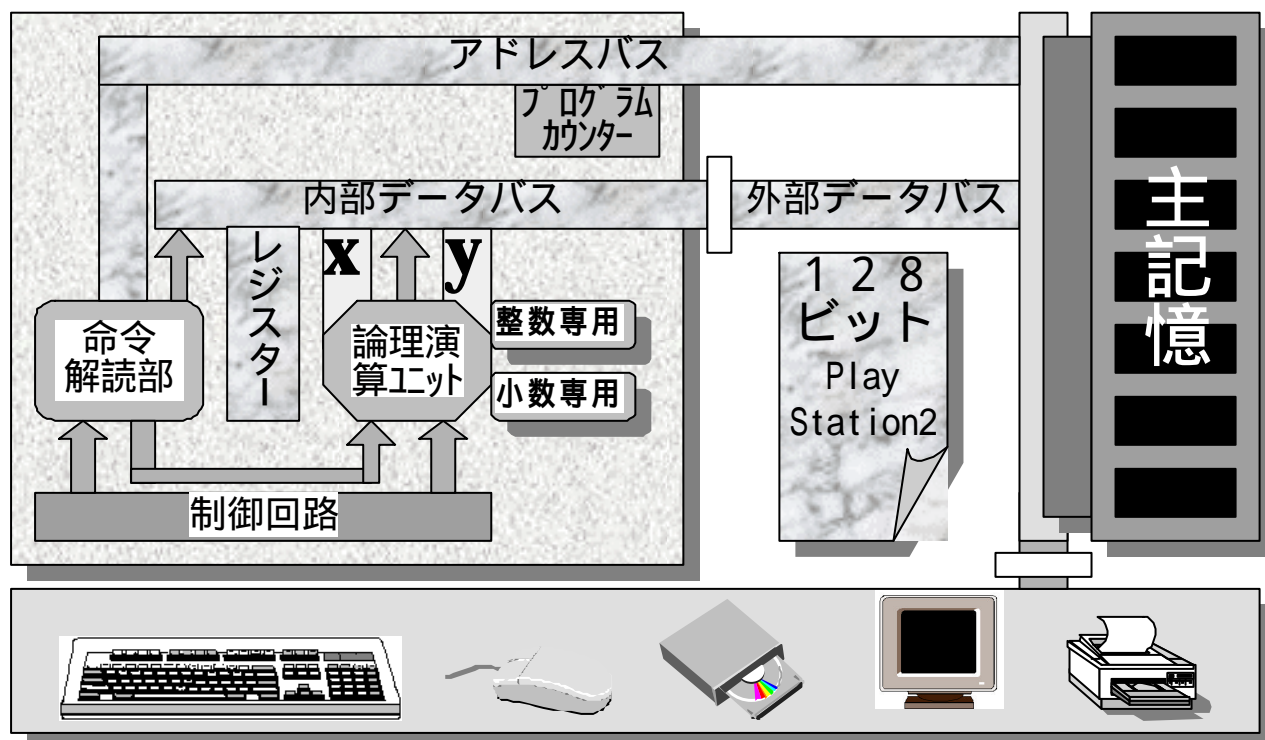
## コンピュータへの命令

### 準備

- 1) コンピュータへの命令を、英字・数字・記号などで表せるプログラム言語にする。命令を一つ一つ順番に書いていく。
- 2) 作成したプログラムをディスクへ保存する。もし必要ならデータも別に保存しておく (例えば、攻略したステージ番号や残りのキャラ数など)

### 実行

- 1) CPUが実行プログラム初期値 (プログラムカウンター) を参照、最初のメモリ無内容を読み取り解読・演算を行う。
- 2) 解読結果、どの装置を動かせばよいかを判断し、その装置に命令を送る。



アドレスバス	実行する命令の記憶されている住所を知るための電気信号の流れ
命令解読部	読み取った命令をコンピュータ用に翻訳する
データバス	命令を実行後、必要ならデータを取りこむための電気信号の流れ
レジスター	取りこんだデータを計算用に待機させておく（記憶機能）
論理演算ユニット 整数演算ユニット 浮動小数点演算ユニット	レジスターから2つの値を読み取り命令に従い演算する
プログラムカウンター	次に実行する命令の住所が入っている