

工学概論

コンピュータのハードウェアとソフトウェア

大山 健

2022/11/17

本日の講義

- コンピュータのハードウェアの仕組みと、ソフトウェアとは何なのかを学ぶ
- 資料が欲しい人は右のサイトからダウンロード



<https://indico.iist.nias.ac.jp/event/464/>

ハードウェアとは？

- ハードウェア (hardware) とは、
 - あるシステムの物理的な構成要素、および物理的構成要素の集合体のこと
 - 日本語では一般的な意味での機械であり装置や設備のことを指し、「ハード」とも呼ばれる。
- コンピュータのハードウェアは、
 - デジタル回路なども含めたコンピュータの物理的部分であり、その上で実行されるソフトウェアと区別される
 - コンピュータのハードウェアは、ソフトウェアやデータに比較するとそれほど頻繁に変更されることはない
 - 逆に言えば、ソフトウェアは頻繁に生成・変更・削除されるために soft (柔らかい) と称されている。

『Wikipedia』より

ハードウェアとは（続き）

- 多くのコンピュータ・ハードウェアは一般人の目に触れることがない。というのも、そのほとんどが自動車、電子レンジやCDプレイヤーなどの家電製品、心電図計などの各種医療機器（ME機器）などの様々な機器内に組み込みシステムとして存在しているからである。実際、パーソナルコンピュータのCPUとして使われているマイクロプロセッサは、全マイクロプロセッサ生産数のほんのわずか（0.2%以下）にすぎない。

『ウィキペディア (Wikipedia)』より

パソコンのハードウェア構造

1. 電源ユニット



4. 光学ドライブ



2. CPU



5. ハードディスク



3. メモリ



6. マザーボード



http://pasokon.main.jp/contents/architecture/pc_ba003.html

主要なパソコン (PC) のハードウェア

- **電源ユニット (Power Supply)**
 - 各部品や装置に電源を供給する装置
- **CPU (Central Processing Unit)**
 - コンピューターの計算や処理を実行する部品
- **Memory (メモリ)**
 - プログラムやデータを読み込む記憶装置
- **CD-ROM, DVD-ROM ドライブ、フロッピードライブ**
 - ディスクからプログラムやデータを読み込む記憶装置
- **マザーボード(メインボード)**
 - コンピューターの主要な部品をつないでいる基板
- **ハードディスク**
 - プログラムやデータを保存する装置
- **GPU (Graphics Processing Unit)**
 - CPUを補助し、表示関係の処理を行う

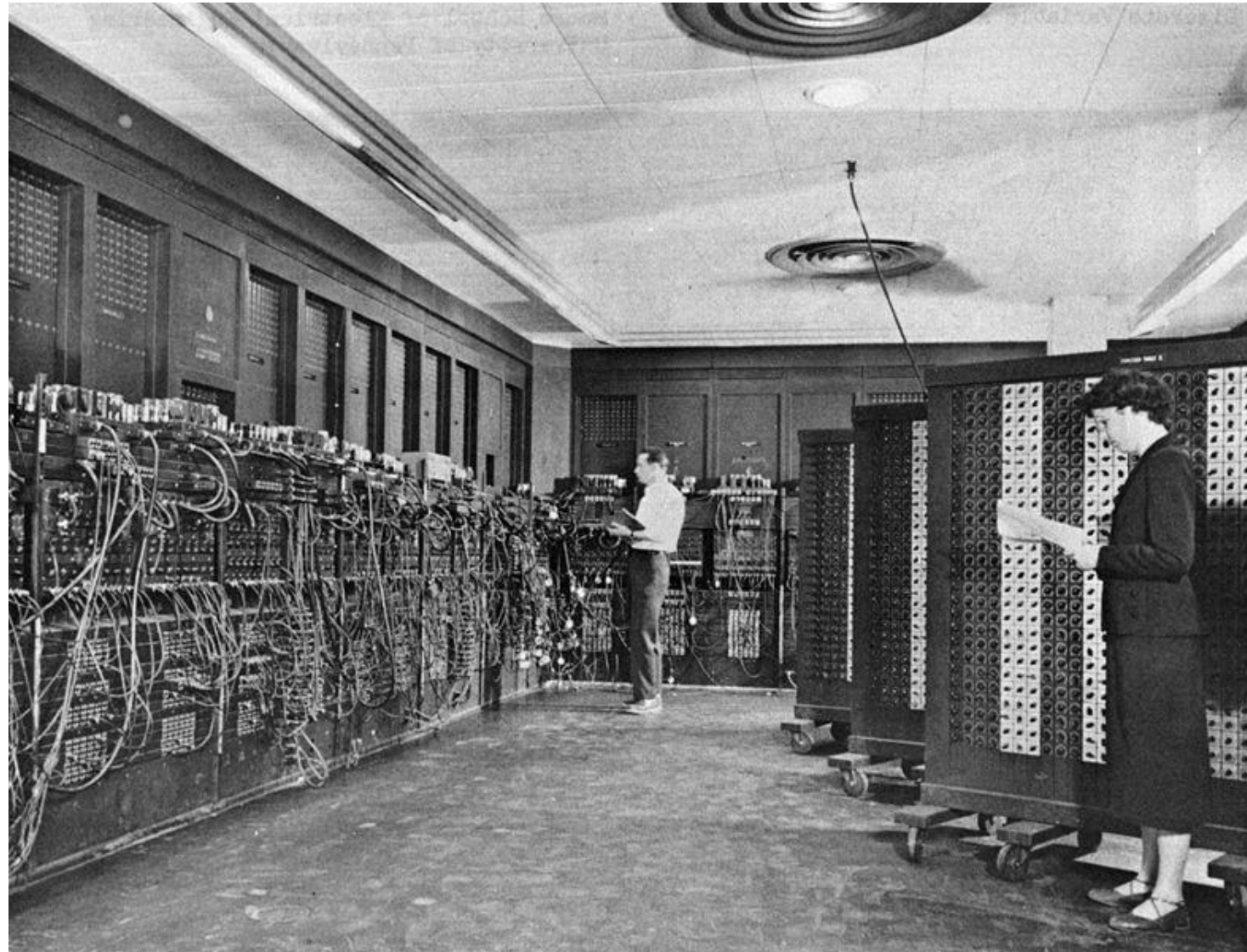
C P U (Central Processing Unit)

- コンピュータの計算や処理を実行する部品、言わばコンピュータの「頭脳」
- CPU以前の電子計算機は、計算のタスクが変わるたび、配線を変更しなくてはならなかった



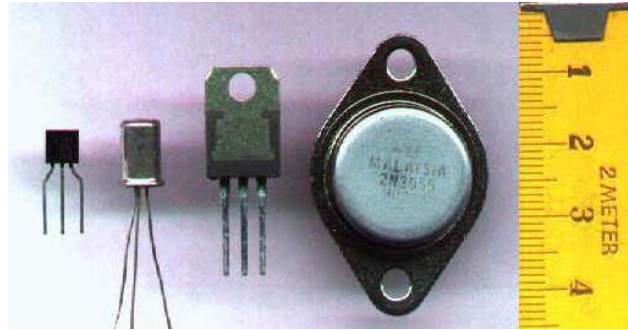
ENIAC (エニアック)

- 世界初のコンピュータと言われる
- 1946年にペンシルバニア大学で公開され、1955年まで使われた
- **真空管**で構成



C P U (Central Processing Unit)

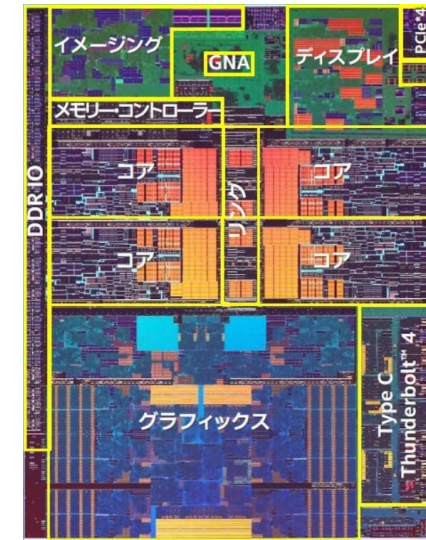
- 現在ではLSI（大規模集積回路）として、1つの「チップ」に作り上げる→厳密にはマイクロプロセッサ(MPU)と呼ばれる



30-40年前からあるトランジスタの形



写真・プリント技術

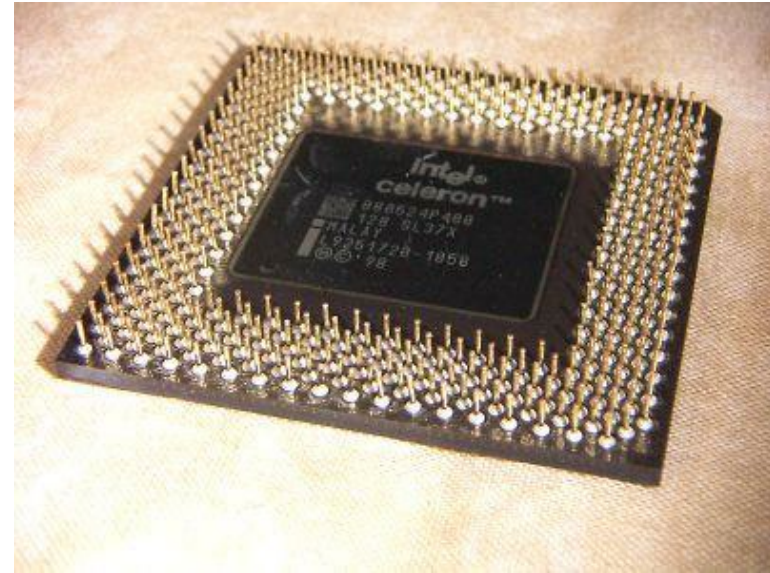


2010年頃のIntel製第11世代Coreプロセッサ
14億個ものトランジスタで構成

- CPU（中央演算処理装置）とは、「**ノイマン型計算機**」と呼ばれる仕組みにより、ソフトウェアを搭載して演算を行う装置
 1. 命令の羅列をデータとして記憶装置（メモリ）に格納
 2. それを順番に読み込んで実行

C P U (Central Processing Unit)

1980年頃のCPU (Motorola)
ゲジゲジ



2000年頃のINTEL CPU
針 (ピンがある)



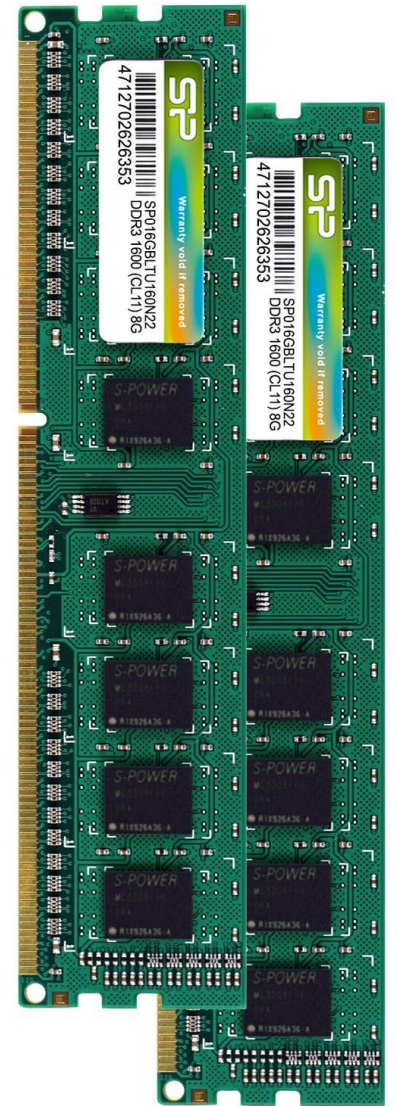
最新のINTELのCPU
(BGA: Ball Grid Array)

一般的なメモリの種類

- **不揮発性** ... 電源オフでも消えない
 - ROM (Read Only Memory) ... 普段は読み込み専用
消去、書き込みは特殊な方法で行うか、一回書き込んだら変更不可
 - **Flash Memory (Flash ROM)** ... 電氣的に消去可能だが、ブロック単位でしか消去、書き込みができい → 書き込みが遅い
 - USBメモリスティック、SDメモリなど
- **揮発性** ... 電源オフで消える!
 - RAM (Random Access Memory) ... 格納データに自由に**高速にアクセス(読み書き)**できる
 - **SRAM (Static RAM)** ... トランジスタ（フリップフロップ）を記憶に用いる → **高速**だけど高価 → より CPU に近いところに少量配置して、CPUの一時記憶場所としえ利用（**キャッシュメモリ**）
 - **DRAM (Dynamic RAM)** ... コンデンサに貯まった電荷を記憶に用いる → アクセス（繋ぎ変え）に時間がかかり、リフレッシュ（再充電）も必要 → 若干遅いが**安価で大容量**なものができる

メインメモリ (Main Memory): DRAM

- CPUが直接アクセス（読み書き）し、プログラムの実行に利用
 - プログラムそのものの配置
 - 中間データの保存
- 電源を切るとデータは消えてしまう (揮発性)
- 8個や16個のメモリICが基板に付いている (メモリーモジュール)
例: “DDR3-2666 (PC3-21333)” ... 1333 MHz のバススピードで 21 GB/s の転送速度を実現
最近では DDR4 世代
- モジュール一枚あたりの容量は 4, 8, 16, 32 GB 等
- 組み込みシステムや小型PCではマザーボード直付も



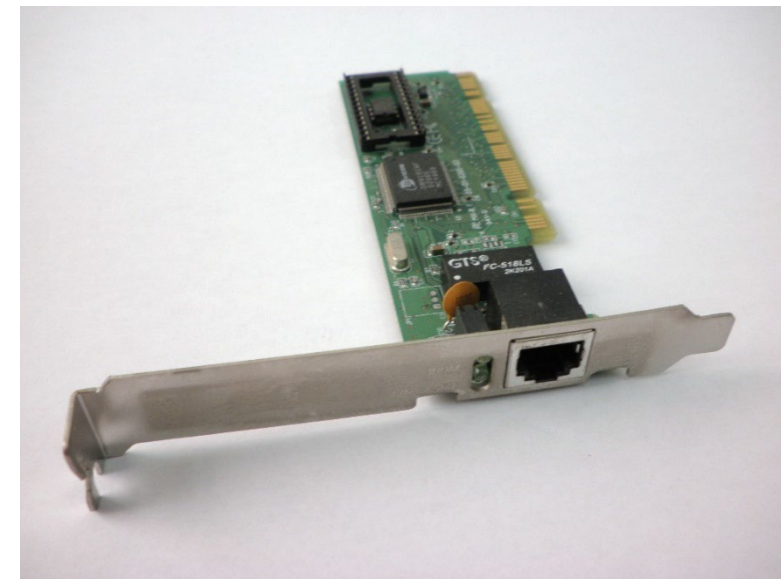
補助記憶装置

- ハードディスク
 - **HDD** (Hard Disk Drive): 磁性体を塗った円盤を回転し、記憶、読み出しを行う
 - 大容量(現在最大 16 TB)なものがあり、安価
 - **SSD** (Solid State Drive): Flash Memory を利用したもので、高価だが特に読み出しが高速
- フロッピーディスクドライブ, 薄い磁性体円盤や磁気テープを用いて、取り外し可能にしたもの
 - 容量は 1.44 MB 程度で、最近では使われない
- テープドライブ ... 磁気テープ
 - 巨大データセンター用などで、ロボットを用いてデータ保存を行ったりする
- CD, DVD, BD
 - ソフトウェア配布用などに使われる
 - 減る方向にある (ネット配信が普及)



ネットワークインターフェース

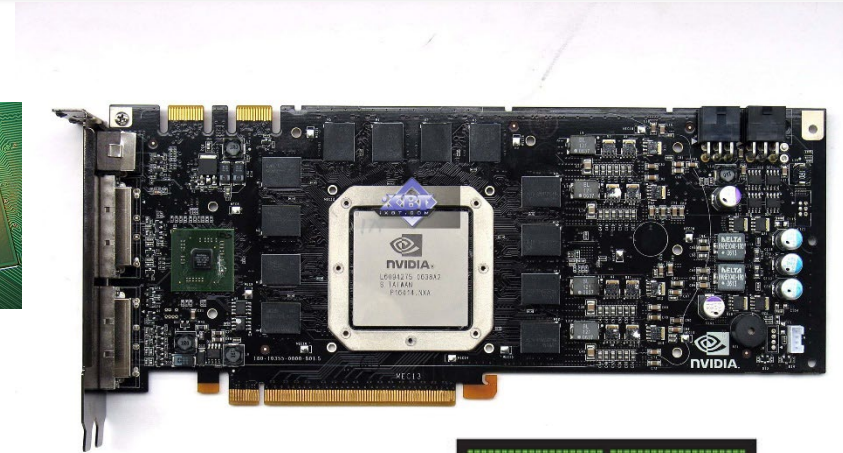
- PC間のデータ転送に不可欠
- 昔: RS-232c (シリアルデータ転送) など
 - 64 kbps 程度までの速度
 - PC-PC 接続とモデムを通してインターネットへの接続
- 今: **イーサネット (Ethernet)**
 - 100 Mbps, 1 Gbps(現在主流), 10 Gbps
 - 数台から数千台のPCをつないで**LAN (Local Area Network)**の構築
 - **インターネットの基幹技術**



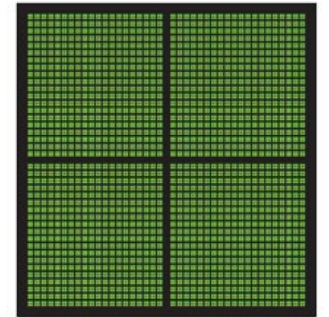
その他のPCのハードウェア

- GPU (Graphics Processing Unit)

- 高速画像処理(3Dゲーム, VR)
- 大画面化(Full HD, 4K)、アニメーション処理
 - 従来CPUで行っていた処理を専用のハードウェア(並列プロセッサ)で処理
- 特殊な科学計算などにも利用
 - 単純だが繰り返し並列で行える計算を数百～数千のコア(CPUの小型版)を用いて行う
 - 数TFLOPSレベルの計算能力
 - GPUコンピューティング



CPU
MULTIPLE CORES



GPU
THOUSANDS OF CORES

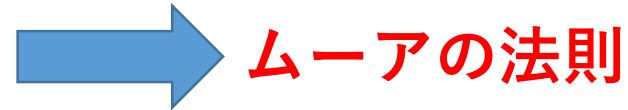
- UI(ユーザーインターフェース)

- モニタ
- キーボード
- マウス

- 冷却システム
 - 空冷ファン
 - 通常のPCのCPUには必須!
 - 空冷システム

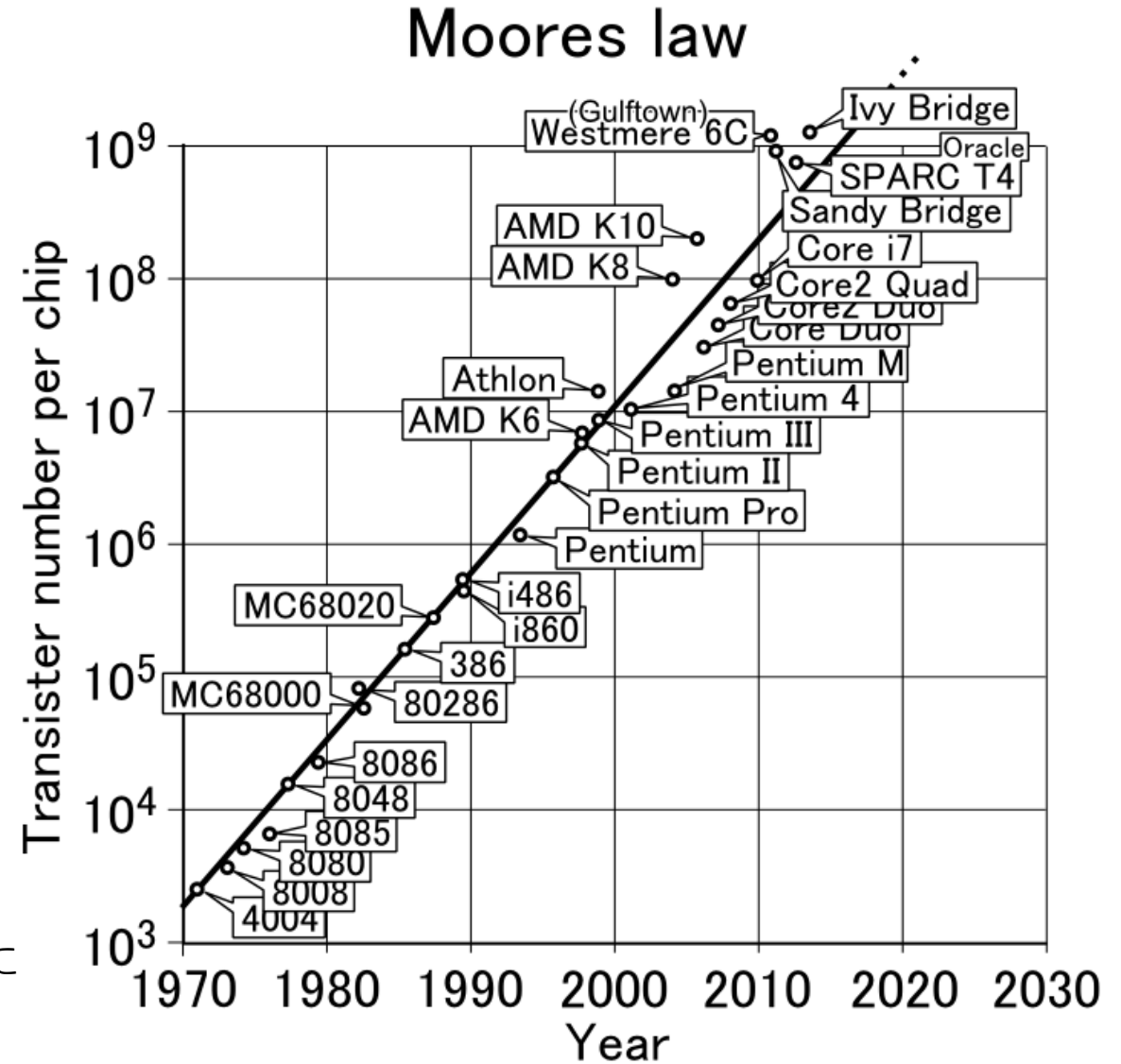
進化するコンピュータ

- 1.5年で倍(指数関数的成長)



- CPU集積度
- メモリ速度
- メモリ容量
- HDD速度
- HDD容量
- バス速度
- ネットワーク速度
- 計算速度

特に必然性も根拠もなく、単に
そうになっているだけ



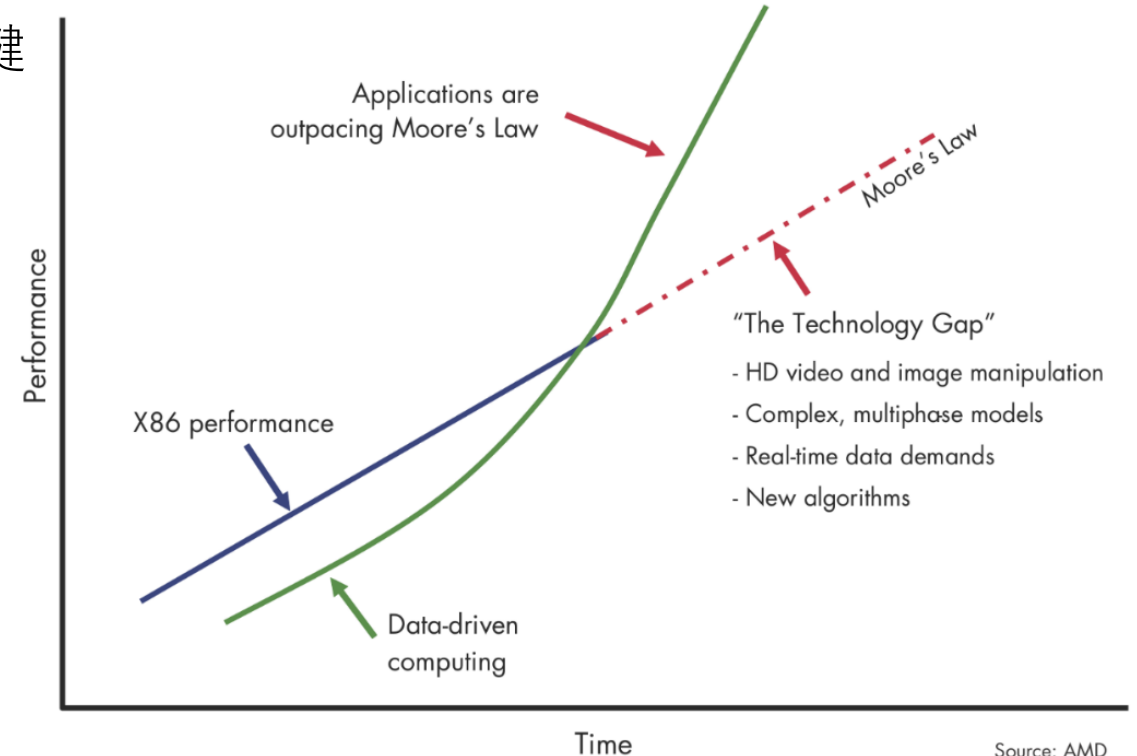
加速するが追いつかない計算機技術

- 世界のコンピュータは進化を続ける（集積度は1.5年で倍）が、ここ数年、要求される性能はそれをどんどん超えはじめている
 - 金融
 - 科学計算・大規模実験
 - 産業
- **High Performance Computing (HPC):** 高性能計算技術が鍵

しかし！

世界のデータ量、処理速度への要求はこれまで Moore の法則に従ってきたCPUの進歩を大きく上回る

= ビッグデータ



HPCの例: 理研の「富岳」

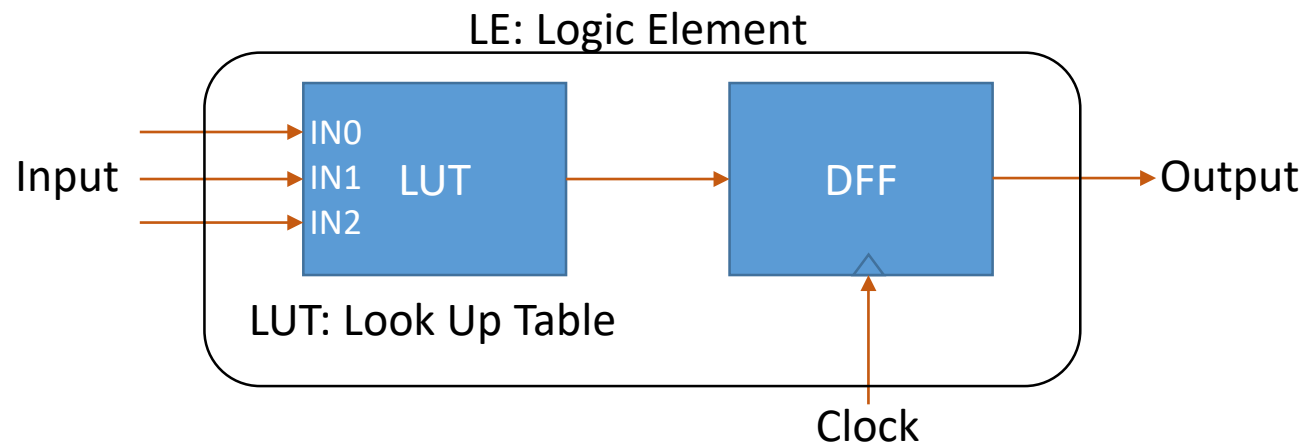
- 現在世界1位: 400 PFLOPS = 4×10^{17} FLOPS (秒間40京回の小数点計算)
- 16万個の計算機の集合体
- 800万個以上のCPU Core (ARM)
- 150 PBの共有ファイルシステム
- 建設費: 1120億円

FLOPS = Floating-point operation per second

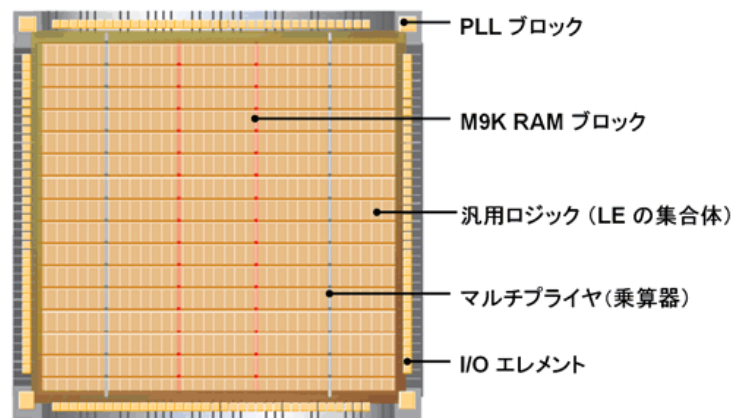


もうひとつの古くて新しい技術: FPGA

FPGA = Field Programmable Gate Array (現場で回路変更可能な論理回路配列)



このような多数入力から出力を一気に決定する回路が多数入っている



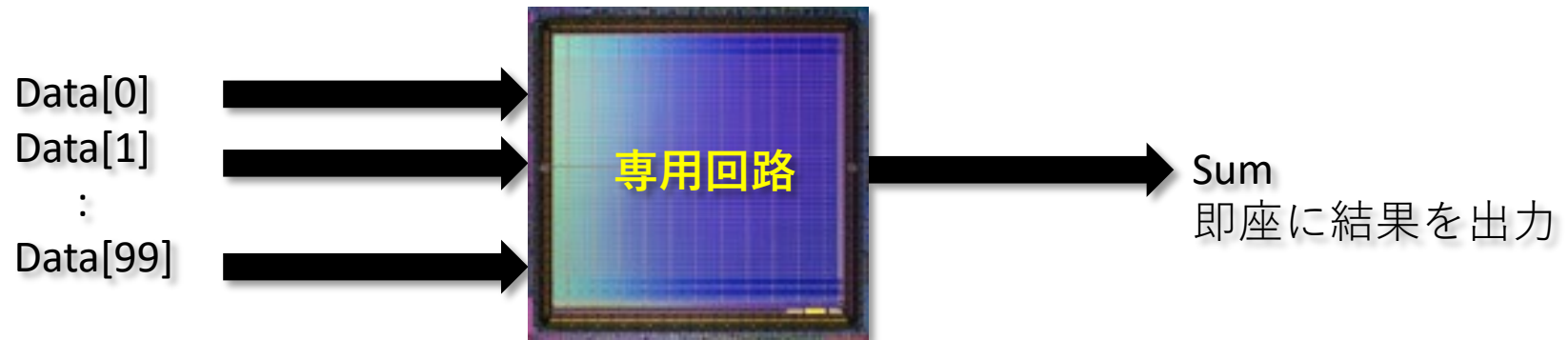
LUTの使用例

| IN0,1,2 | OUT (AND) | OUT (OR) |
|---------|-----------|----------|
| 0, 0, 0 | 0 | 0 |
| 0, 0, 1 | 0 | 1 |
| 0, 1, 0 | 0 | 1 |
| 0, 1, 1 | 0 | 1 |
| 1, 0, 0 | 0 | 1 |
| 1, 0, 1 | 0 | 1 |
| 1, 1, 0 | 0 | 1 |
| 1, 1, 1 | 1 | 1 |

FPGAを使うと超高速計算ができる

- CPU: プログラム命令を逐次実行

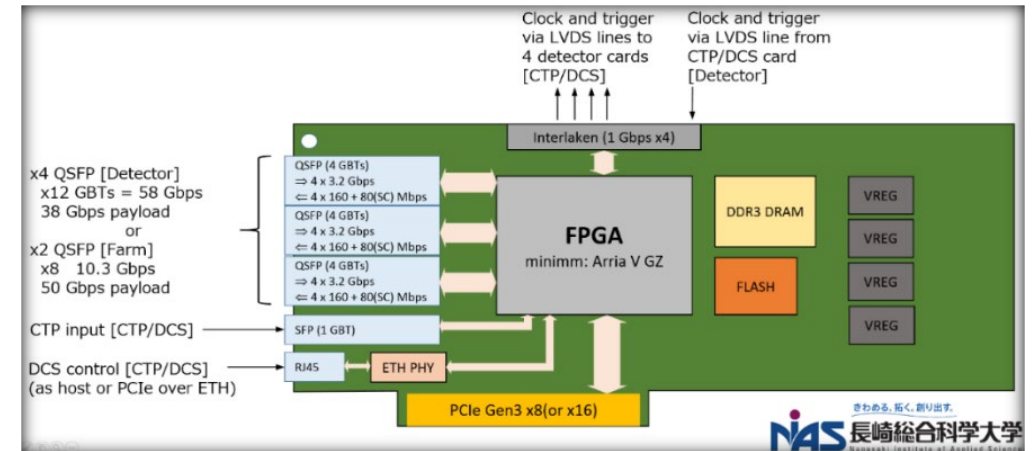
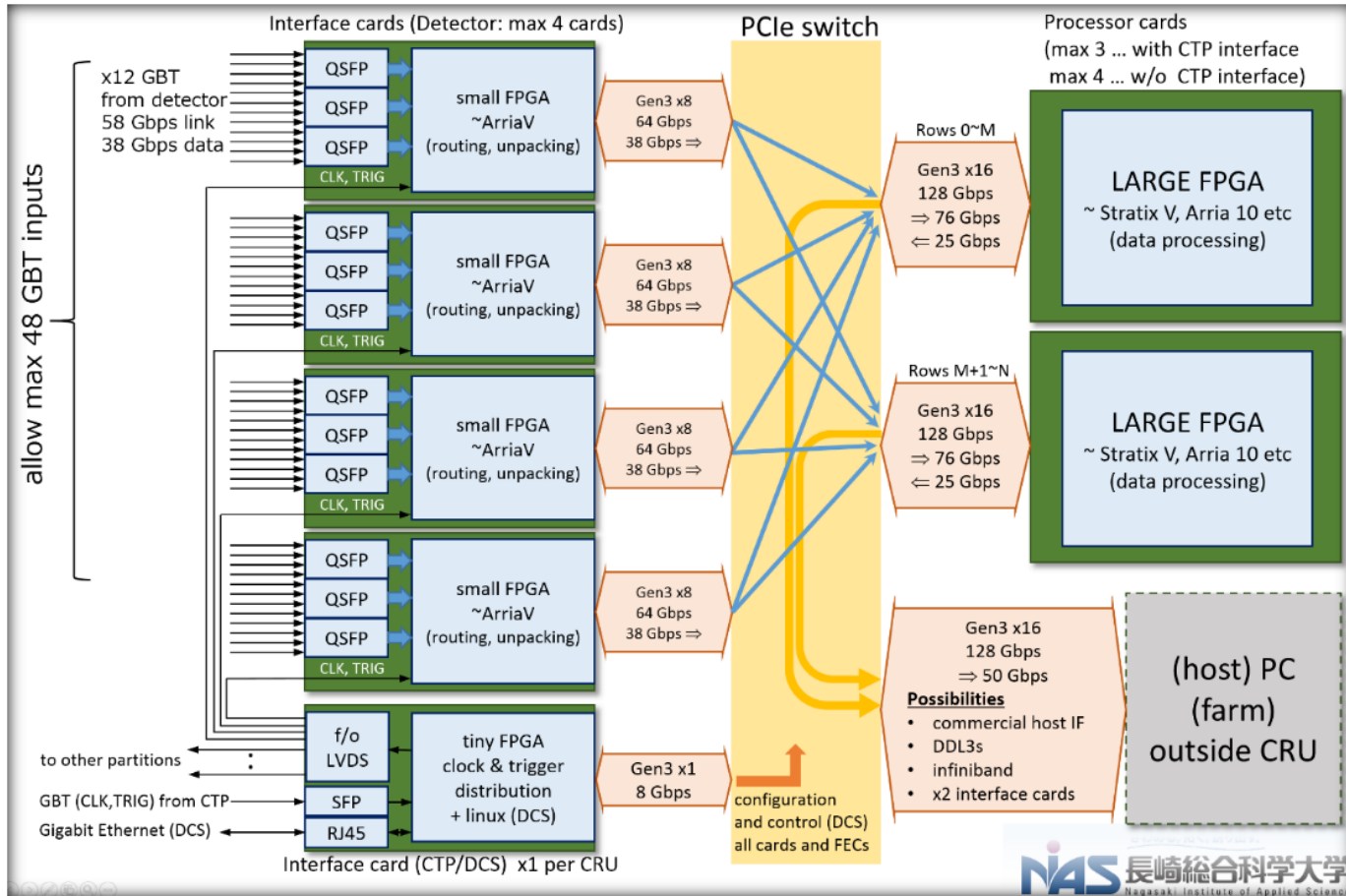
```
for( i=0; i<100; i++) { Sum += Data[i]*Data[i]; }
```
- FPGA: ハードウェアで直接即時処理



- 大量のデータ（ビッグデータ）を超大容量、低遅延で処理
- 昔: 規模も小さく、高価で、プログラミングも難しかった
- 価格の大幅低下と市場拡大 ... 2013年~3500億円(CPUの10%)
- 現在: 加速度的に大規模化、高速化 (**400万以上のLEを搭載**)
- **高級言語によるプログラミング技術が熟成**

NIASで開発中のHPCシステム

高エネルギー物理学実験用データ処理システム



ソフトウェアとは

- ソフトウェア (Software) は、
 - コンピュータシステム上で何らかの作業を行う **プログラム群、プロセス群、それらに関する文書**を指す用語
 - 物理的装置である **ハードウェアと対比**させて言うときに使う
 - プログラムとほぼ同義だが範囲は更に広い
 - 日本語では略して「ソフト」とも呼ばれる
- ソフトウェアには、
 - ワード・エクセルのように何かをする **アプリケーションソフトウェア**、
 - アプリケーションソフトウェアに必要なサービスを提供するハードウェアとのインタフェースとなる **オペレーティングシステム**のような **システムソフトウェア**、
 - 分散システムを制御・管理する **ミドルウェア**などがある

フリー百科事典『ウィキペディア (Wikipedia)』より

ソフトウェアの分類

アプリケーションソフトウェア

ビジネスソフトウェア(Officeなど)

教育ソフトウェア

ゲームソフトウェア

プログラミングツール

テキストエディタ

コンパイラ、インタプリタ

デバッガ

システム ソフトウェア

ミドルウェア(データベースなど)
デバイスドライバ
サーバ機能など

OS (Operating System)

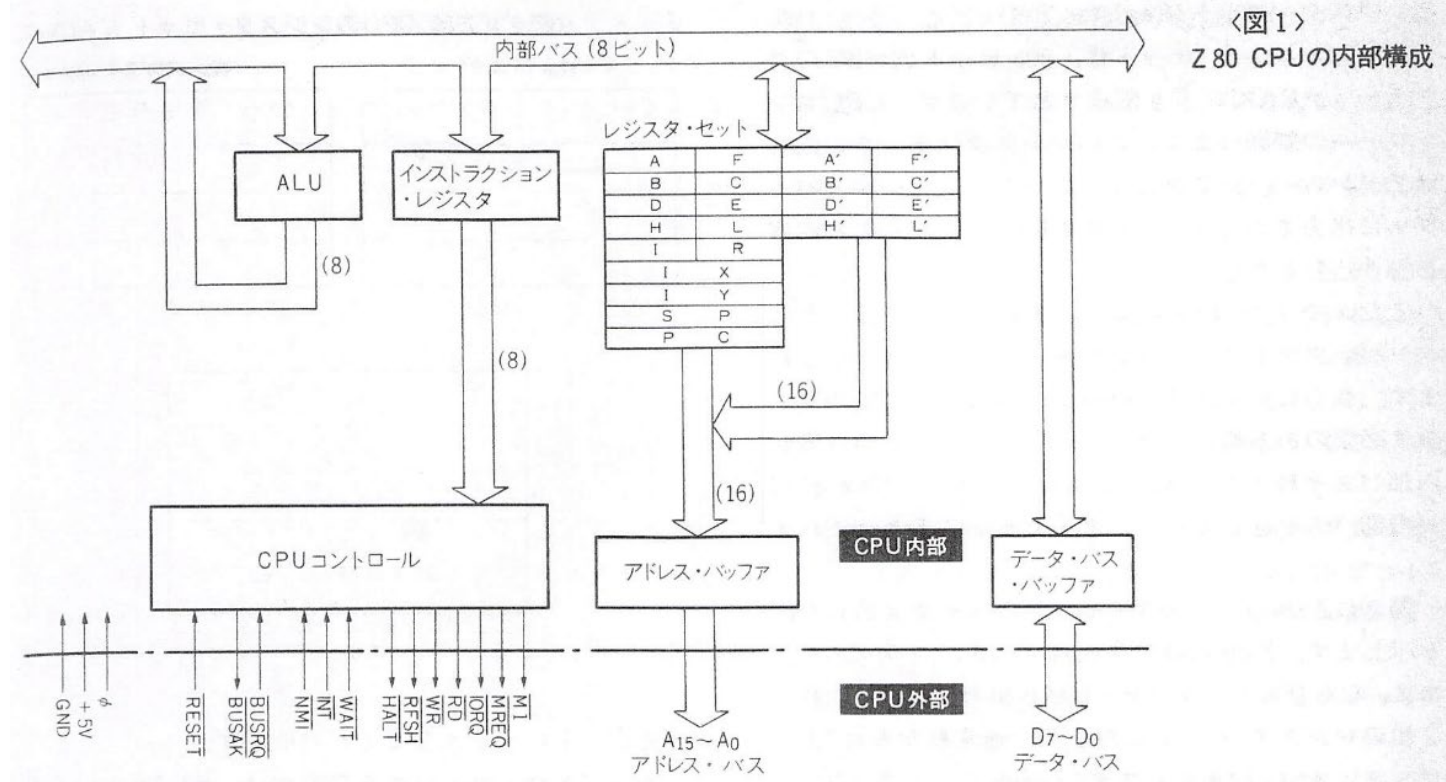
ファームウェア(BIOSなど)

ハードウェア

Software = マシン語からアプリまで

- 単純なCPUのアーキテクチャ

- 最近のものはより複雑だが、基本は同じ



- CPUができること(ノイマン型)

1. メモリ上に並んでいる命令をひとつひとつ実行するだけ

- 命令セット(機械語、マシン語)

- メモリとI/Oアクセス、計算 (和・差・論理演算)、分岐 (ジャンプ) ...

アセンブラ

- マシン語

- メモリ上に直接展開されるビット列
- 普通の人間は直接作成できない

- アセンブラ(assembly)

- マシン語をビット列ではなくアルファベットなどで表したもの(一対一対応)
- 人間も読み書きできる(要熟練)
- CPUアーキテクチャ毎に異なる表記方法

```
Terminal
0001dd0 e2eb 8966 66d0 665b 66c3 6655 6657 6656
0001de0 6653 ec83 660c c389 8966 6655 6657 6656
0001df0 0ab1 8067 3038 5075 8a67 02eb 8966 6655
0001e00 08b1 783c 4275 8a67 02eb 8966 6655 6657
0001e10 30ea 8366 09fa 0c76 8366 09fa 0c76 8366
0001e20 053c 2477 b966 0010 0000 0000 0000 0000
0001e30 1675 8067 303b 1075 8a67 0143 8366 20c8
0001e40 783c 0475 8366 02c3 6667 04c7 0024 0000
0001e50 6700 c766 2444 0004 0000 6700 0f66 2bb6
0001e60 8966 66e8 be0f 66f8 ef83 6630 ff83 7709
0001e70 6646 cf39 0b72 8566 74f6 675f 8966 eb1e
0001e80 6759 8b66 2404 6667 548b 0424 8966 66d5
0001e90 af0f 66e9 e1f7 0166 67ea 8966 243c 6667
0001ea0 44c7 0424 0000 0000 6667 0401 6724 1166
0001eb0 2454 6604 eb43 66a4 ef89 8366 dfe7 8366
0001ec0 41ef 8966 3cf8 7705 66ad cd83 6620 e889
0001ed0 0f66 f8be 8366 57ef 97eb 6667 048b 6724
0001ee0 8b66 2454 6604 c483 660c 665b 665e 665f
0001ef0 665d 66c3 c289 8067 003a 0474 4266 f6eb
0001f00 2966 66c2 d089 c366 5566 5766 5666 5366
0001f10 8366 08ec 8966 66c6 d589 8966 66d0 d0e8
0001f20 ffff 66ff c389 8966 66f0 db85 4774 e866
0001f30 ffbf ffff 8966 67c2 8d66 0604 6667 4489
0001f40 0424 6667 448b 0424 2966 67d0 8966 2404
0001f50 3966 72da 661d 674a 8b66 243c 8966 66ee
0001f60 d989 a6f3 950f 84c0 75c0 67d7 8b66 2404
0001f70 03eb 3166 66c0 c483 6608 665b 665e 665f
```

```
Terminal
00010117 78 08 js 0x00010121
00010118 08 08 outs dl, ds:[esi]
00010119 08 08 sbb al, 0x3B
0001011a 08 08 ret 0xD6E5
0001011b 08 08 jg 0x3B51EE9A
0001011c 08 08 stos es:[edi], eax
0001011d 08 53 mov ch, 0x53
0001011e 82 ED C8 sub ch, 0xC8
0001011f A6 cmps es:[edi], ds:[esi]
00010120 23 22 and esp, [edx]
00010121 54 push esp
00010122 A5 movs es:[edi], ds:[esi]
00010123 15 D7 BE 5D 85 adc eax, 0x855DBED7
00010124 B5 58 mov ch, 0x58
00010125 CF iret
00010126 FA cli
00010127 15 5E 6C EB 27 adc eax, 0x27EB6C5E
00010128 76 C3 jbe 0x00010140
00010129 DA 42 A7 fiadd [edx-0x59]
0001012a 1F pop ds
0001012b FC cld
0001012c 00 D6 add dh, dl
0001012d 8F 15 22 4E 46 C5 pop [-0x3AB9B1DE]
0001012e 0C 62 or al, 0x62
```


高級言語

- 高級言語

- 自然言語に近い形で、プログラムが書ける
- CPU依存性が少ない
- 構造化プログラミング、オブジェクト指向プログラミングが可能

- 様々な種類(数百)

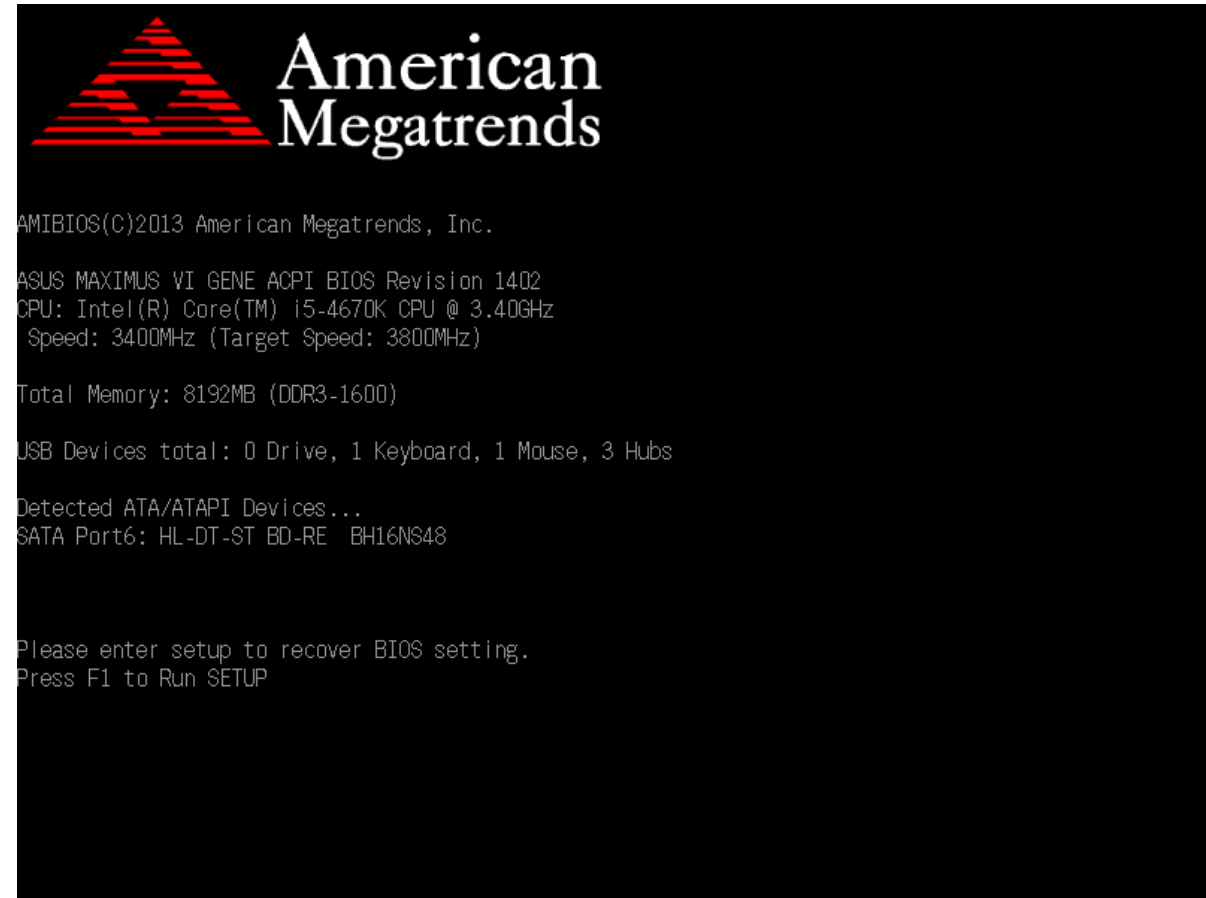
| | |
|-------------|--------|
| C | Ruby |
| C# | Python |
| C++ | Lisp |
| Objective-C | Scala |
| BASIC | PHP |
| JAVA | Perl |

```
168 static int __init obsolete_checksetup(char *line)
169 {
170     const struct obs_kernel_param *p;
171     int had_early_param = 0;
172
173     p = __setup_start;
174     do {
175         int n = strlen(p->str);
176         if (parameqn(line, p->str, n)) {
177             if (p->early) {
178                 /* Already done in parse_early_param?
179                  * (Needs exact match on param part).
180                  * Keep iterating, as we can have early
181                  * params and __setups of same names 8( */
182                 if (line[n] == '\0' || line[n] == '=')
183                     had_early_param = 1;
184             } else if (!p->setup_func) {
185                 pr_warn("Parameter %s is obsolete, ignored\n",
186                         p->str);
187                 return 1;
188             } else if (p->setup_func(line + n))
189                 return 1;
190             }
191         p++;
192     } while (p < __setup_end);
193
194     return had_early_param;
195 }
```

Cで書かれたLinux Kernelのソースコード

BIOS (Basic I/O System) メモリ

- マザーボードはプログラムが書き込まれている **不揮発性** メモリを必ず1つ備える
- 通常はフラッシュメモリーで作られており、電源オフでも消えない
- 最初CPUはこのメモリにある命令から実行する
 - そしてそこには次にHDDからOSデータをメモリに読み込んで、
 - そのOSにジャンプして実行するよう書かれている



OS (Operating System)

- システム起動後に実行される**基本ソフトウェア**

- 電源が切れるまでコンピュータを制御し続ける



- 主要なOS

- MS-DOS, Microsoft Windows (Microsoft) ... デスクトップで一番普及(2013年は3.3億台)

- UNIX ... はじまりは AT&Tベル研究所, 今はBSD, Solaris, ほかの多くの派生がある

- Linux (Linus Torvalds, **GNU**, open software) ... Linux はカーネル(kernel)のことで、これを用いたOSとしての distribution として Ubuntu, Debian, redhat, など数多くの系統がある

- UNIXの技術・アイデアを多く移植している
- サーバ用途でよく使われる

- MAC OS (Apple)

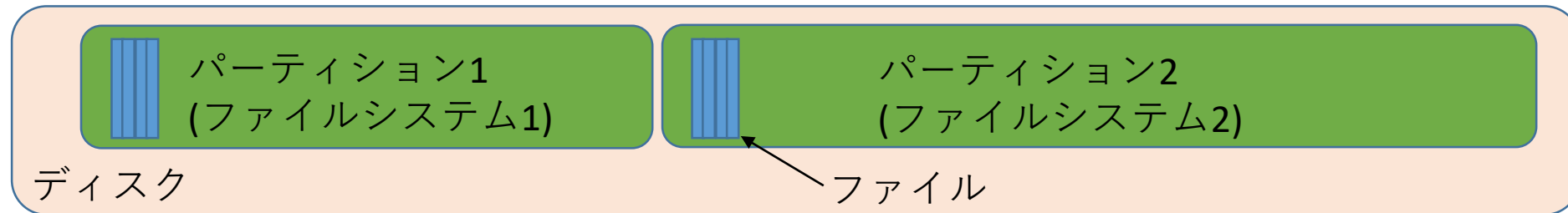
- OS X, iOS (Apple) ... BSD UNIX ベース

- Android (Google) ... Linux ベース、携帯やタブレットに使われ、現在世界で最も多く使われているOS(2013年は8.8億台)



OSの仕事

- ディスク上のファイルシステム管理
 - アプリのバイナリ、システム動作に必要なデータなどはディスクに収められている
 - それらはファイルネームをつけられ、フォルダ（ディレクトリ）に収められている



- メモリ管理
 - 異なったアプリが同一のメモリ領域を勝手に使うと？
- デバイスの管理（デバイスドライバ）
 - キーボード、ネットワークカード、マウス、サウンド、など

OSの仕事(つづき)

- アプリケーションの実行、監視(プロセス管理)
 - ディスクからアプリケーションのバイナリデータをメモリに展開し、実行
 - その後もそれを監視する (応答がないアプリは?)
- システムコール、API(Application Programming Interface)の提供
 - いわゆるライブラリ等であり、アプリなどがこれを利用する
- セキュリティ管理 (ユーザー認証、ログインなど)
- ユーザーインターフェース
 - OSによってはGUI(Graphical User Interface)の提供 ... Windows, MACなど
 - 基本的にOSの顔となる
- 基本アプリケーションの提供
 - テキストエディタやシェル(shell)
 - コンパイラ
 - 広義ではメールソフトやウェブブラウザまで

最後に

お勧めの勉強法

- コンピュータを作ってみる
 - 部品を手に入れて自分で作れば、構造がわかる
 - 修理もできるようになる
 - 古いPCの改造もお勧め(HDD→SDD, 大容量化)
- 自分でOSを選んでインストールしてみる
 - Linux は無料で、古いPCでも動作
 - OSのしくみがよくわかる

レポート課題

問1. 今日の講義で新しくわかったことを3つ解説してみよう。

問2. 今日の講義でまだよくわからなかったことを1つ以上あげよう。