

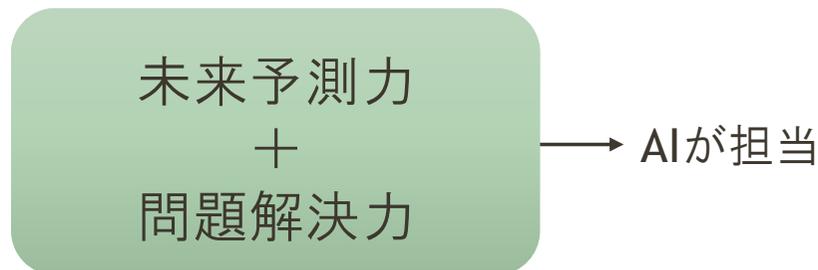
第3回 コンピュータの構成と大規模集積回路

情報通信コース 北川章夫, Microelectronics Research Lab. (MeRL)

電子情報通信技術を学ぶ前に

- 2週にわたって、現代の電子情報通信技術の基本となる、「コンピュータ」、「集積回路」、「無線ネットワーク」、「IoT」について学ぶ
- 技術は常に発展し続けているため、**一生涯学び続ける必要がある**（今の自分に満足した時点から時代遅れな人になる）
- 最先端に到達するためには、トレンドを追うのではなく、**基礎から体系的に理解することが重要**（試験の成績は重要ではなく、**なるべく広い分野を関連づけて理解**する・・・ハード/ソフトなどの分類はやめよう）

昔の技術者に求められた能力



現代の技術者に求められる能力



新しい技術者のイメージ

1. モノ（商品）を作って売る
2. モノをどうやって作るかを考える
3. 高性能化・低価格化
4. 高度な単独技術



従来の技術者

現代の技術者には起業
や事業運営の能力も求
められています

1. 方法（ソリューション）を売る
2. 課題の解決方法を考える
3. イノベーション
4. 既存技術の複合 + x



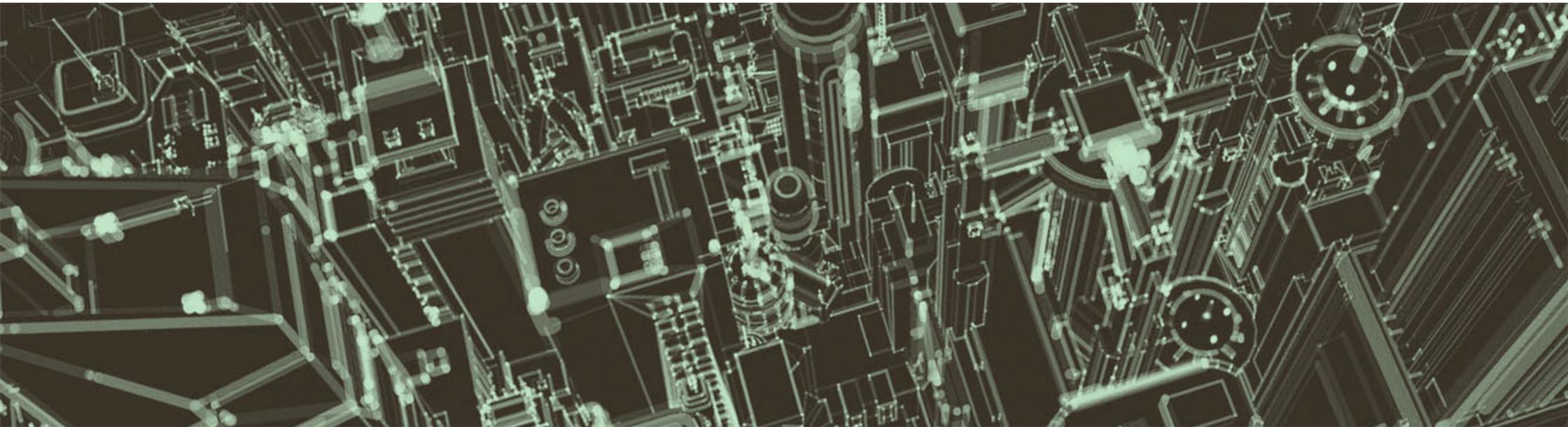
現代の技術者

はじめに

- ❖ コンピュータの歴史的発展や詳しい構造は「計算機システムA, B」の講義で学ぶことにして、ここでは、コンピュータの方式を簡単に理解しよう
- ❖ 電子情報通信技術全般の基盤技術である大規模集積回路設計技術は「集積回路工学A~D」で学ぶが、ここでは、設計ソフトを使ったVLSIの設計の手順を理解しよう
- ❖ 下記のサイトの講義資料を必要に応じて参照すること

<http://jaco.ec.t.kanazawa-u.ac.jp/kitagawa/edu/>





3-1 コンピュータの種類

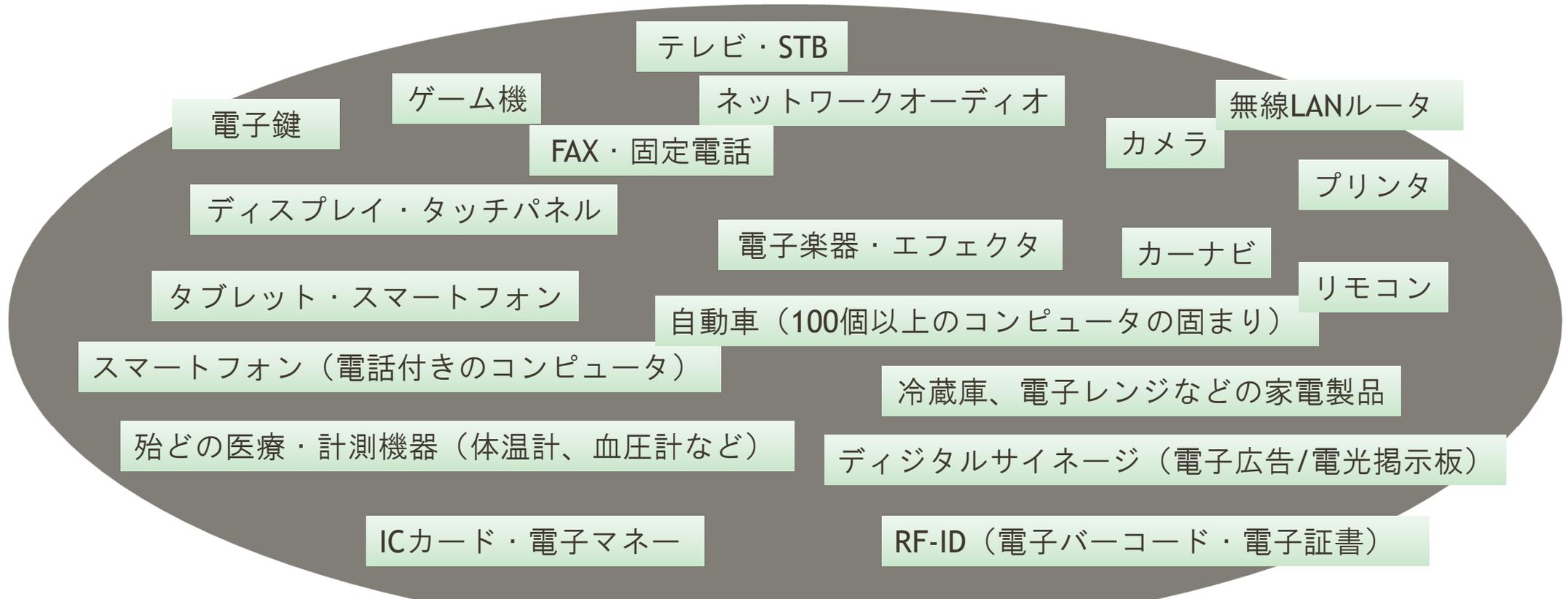
コンピュータはあらゆるところに組み込まれている



質問

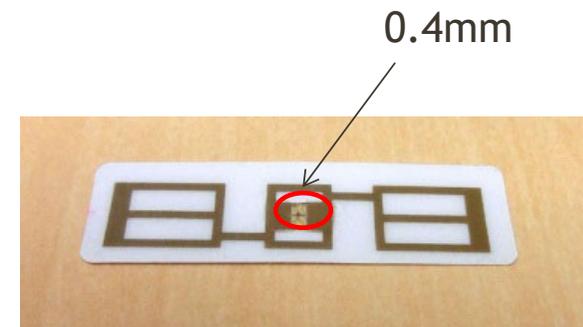
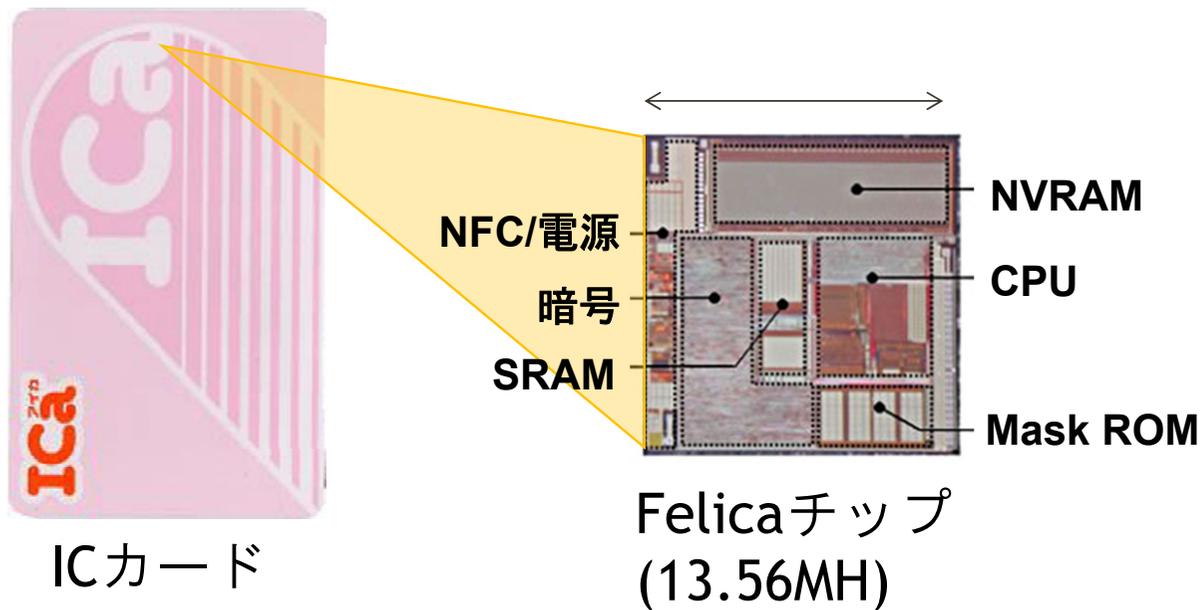
❖ 皆さんはコンピュータを何個持っていますか？

身の回りのコンピュータ



基本的に全ての電子機器は特定用途のコンピュータを搭載していると考えてよい

ICカードやICタグもれっきとしたコンピュータ



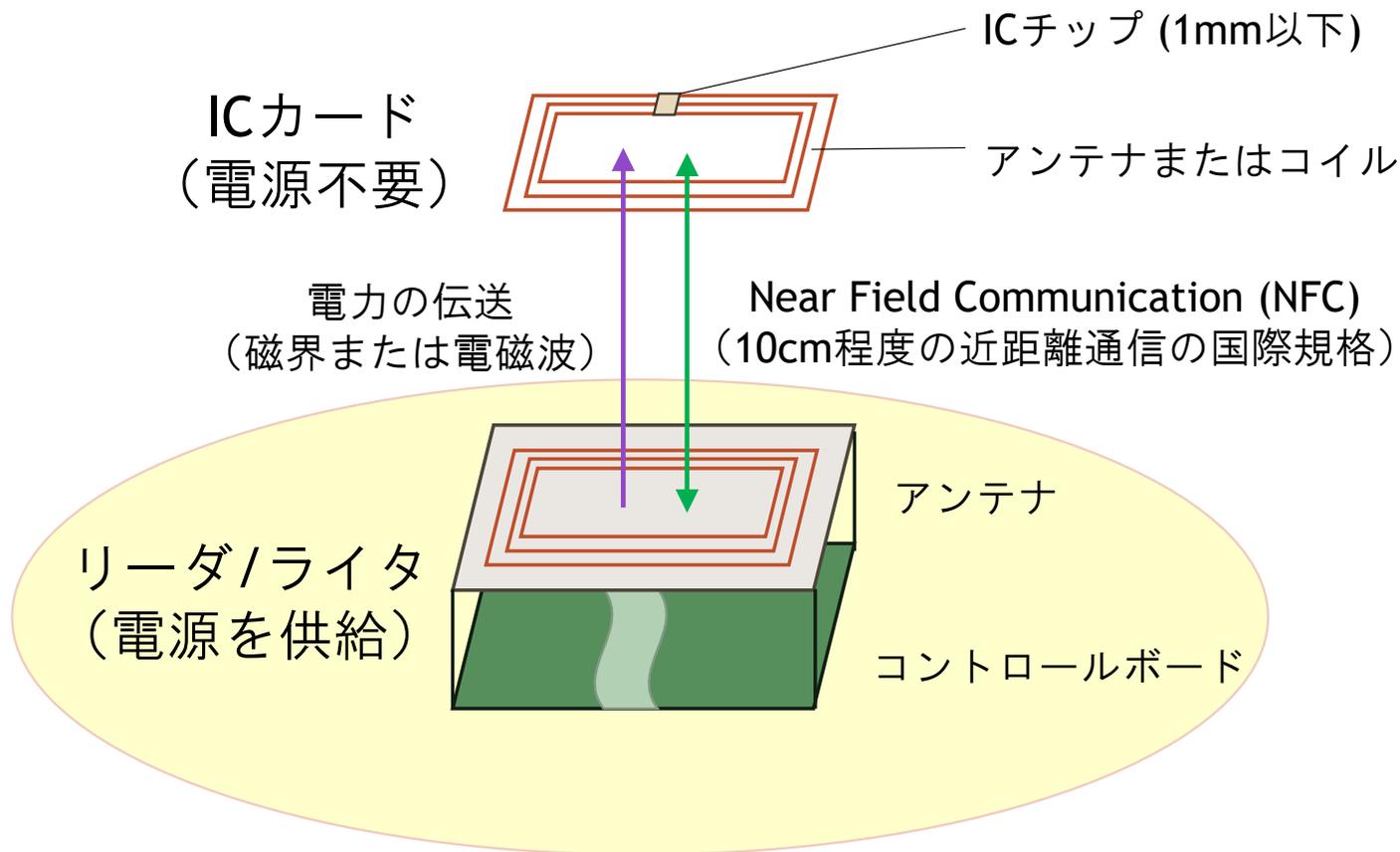
2.4GHz帯 RF-IDタグ



NFC決済

CPU = Central Processing Unit (本体)
NVRAM = Non-Volatile Random Access Memory
NFC = Near Field Communication (次頁)

ICカードの仕組み



[参考] 無電源無線通信技術
アンテナの吸収電力を変化させることによりリーダ/ライタが発する電磁界をASK変調して無線通信する技術。Backscatter方式と呼ばれる。リーダ/ライタ側もBackscatterを使う場合は、Ambient Backscatterと呼ばれ、完全な無電源となる。

ASK = Amplitude Shift Keying

使用目的によるコンピュータの分類

高速計算・技術開発に特化

汎用

特定用途向け



スパコン



GPGPUサーバー



デスクトップ



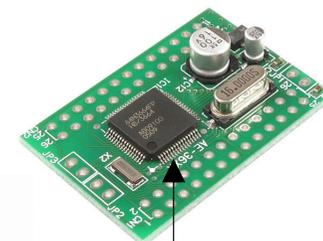
ノートブック



スマホ



1ボードPC



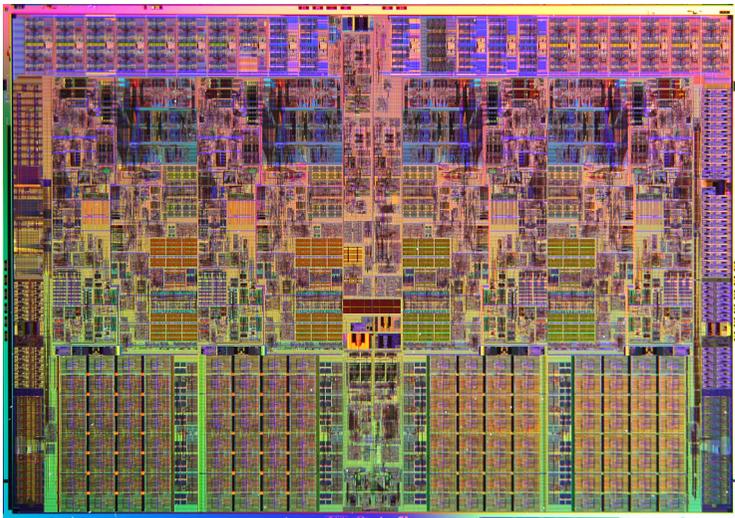
マイコン
(Microcontroller)

ソフト+ハードを含めて
「組み込みシステム」
(影で動いている)

[参考] GPGPU = General-purpose computing on graphics processing units

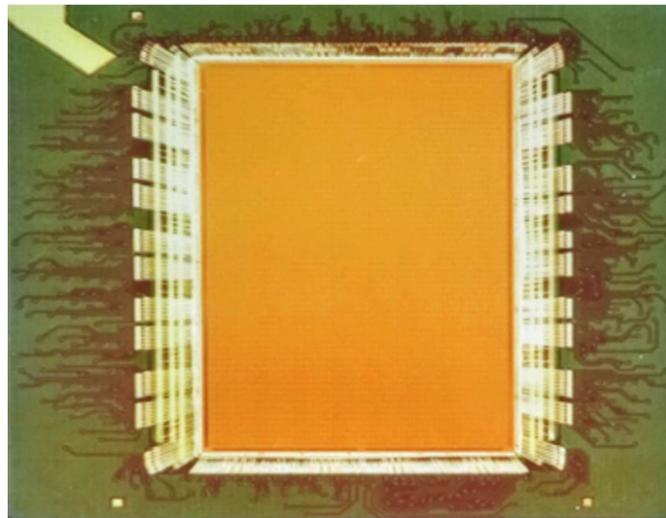
コンピュータの動作原理による分類

プログラムにより動作



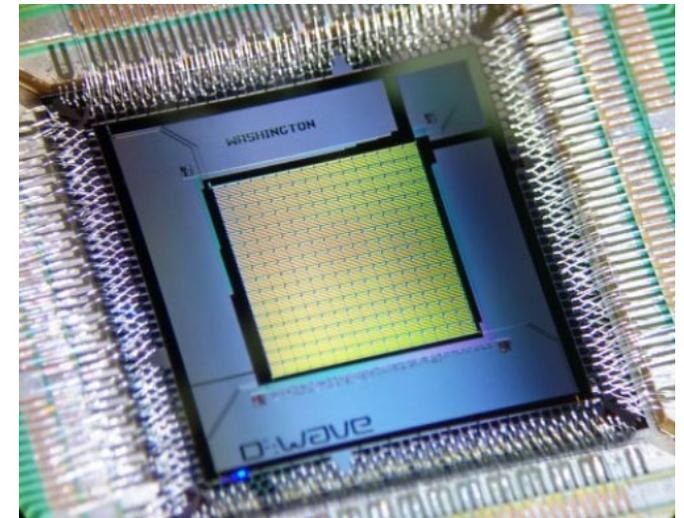
ノイマン型コンピュータ (Core i7)
出典：Intel Corporation

学習により動作



ニューラルネットワーク
(TrueNorth)
出典：IBM

プログラムにより動作



量子コンピュータ (Washington)
出典：D-Wave Systems

コンピュータの実装に使われる技術

半導体・超伝導・光技術

VLSI技術（回路とソフト）

超伝導・光デバイス技術

ノイマン型コンピュータ

ニューラルネットワーク

量子コンピュータ

ノイマン型コンピュータ

- ▶ 現在主流の汎用コンピュータはノイマン型と呼ばれる（スパコンを含む）
 - ▶ 3名の開発者の名をつけて Eckert-Mauchly-von Neumann architecture とも呼ばれる
- ▶ **ノイマン型コンピュータ**の特徴
 - ▶ メモリにプログラムとデータを内蔵し、論理記述可能な任意の処理を行う（チューリングマシンの一種）
 - ▶ プログラムを逐次実行する（**高速演算が要求される**）
- ▶ ノイマン型コンピュータでないものは「非ノイマン型コンピュータ」と呼ばれる

方式	例	特徴
条件に合致する解を求める	量子コンピュータ, DNAコンピュータ	解を並列に探索する。組み合わせ最適化問題に有効とされている。
神経回路の機能と構造を模擬	ディープニューラル ネットワーク(DNN)	所謂AI。学習により高精度な特徴抽出、パターン認識ができる。

ノイマン型コンピュータが苦手な組み合わせ最適化問題の例

ナップサック問題

遠足のおやつに何を何個持って行くか？

- 合計300円以内
- 各おやつの値段と必要度が分かっている
(N 種類のおやつの組み合わせ = 2^N 通り)

30種類のおやつから選ぶ場合、
1,073,741,824通りの組み合わせを検討する必要がある

巡回セールスマン問題

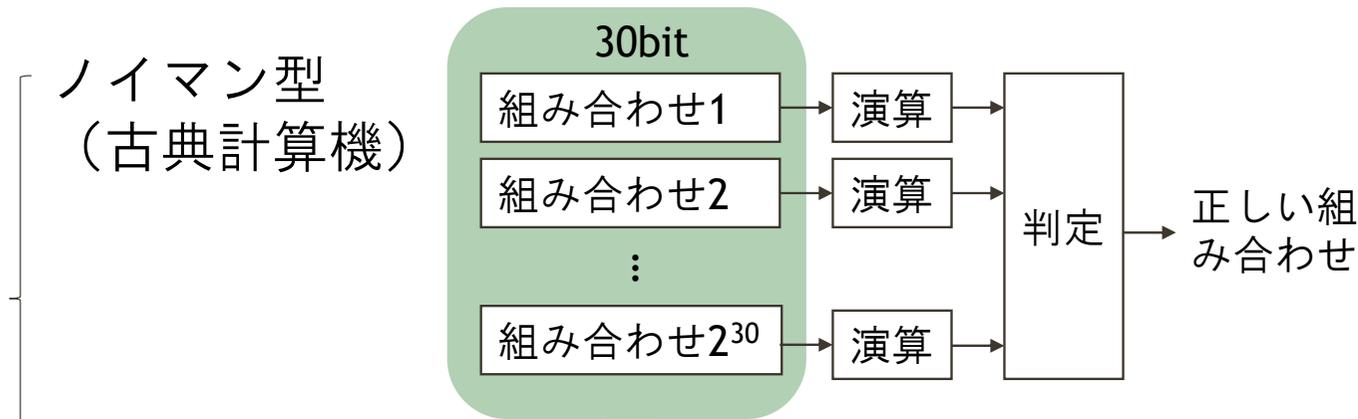
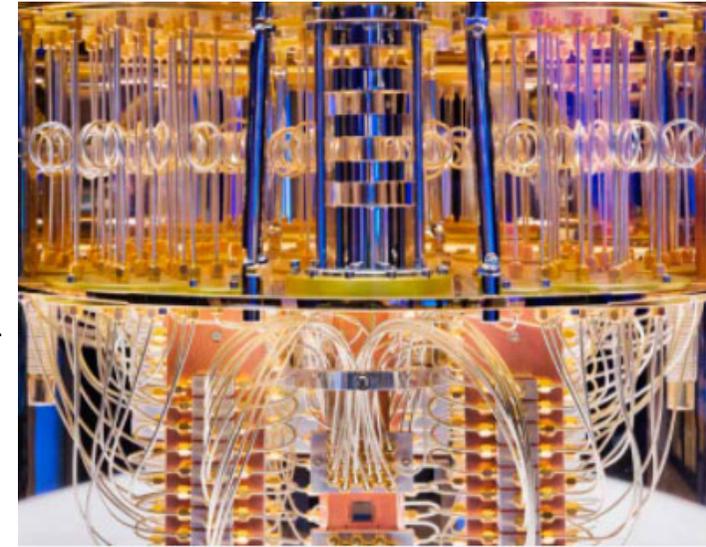
最短距離で都市を巡るにはどの順番でいくか？

- 全ての都市を1回通る
- 都市間の距離が分かっている
(N 都市を回る順序 = $N!$ 通り)

30市を回る計画の場合、
 2.653×10^{32} 通りの組み合わせ
を検討する必要がある

ノイマン型コンピュータは、全ての場合について計算を行うので現実的ではない（スパコンでも困難）

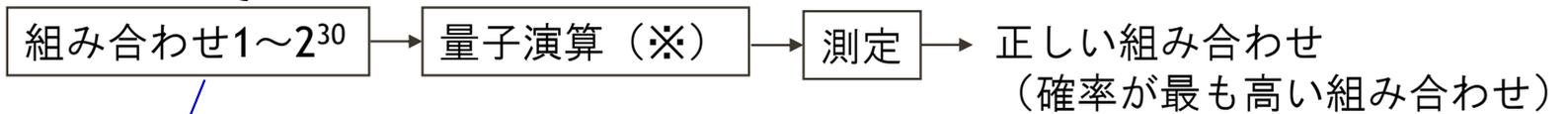
ノイマン型コンピュータと量子コンピュータ



量子コンピュータ

30Qbit

各ビットは1となる確率
で表される



データを重ね合わせるにより1回で計算を行う
→ 従来の暗号も解けてしまう

- 量子状態のベクトルで実現
- 光の位相や偏向
 - 超伝導リングの磁束など

※ 量子コンピュータによるチューリングマシン
(量子演算を何度も行える汎用コンピュータ) も
原理的には実現可能

量子コンピュータを使ってみよう

- 量子コンピュータ用プログラム開発環境
 - https://qiskit.org/documentation/locale/ja_JP/getting_started.html#
- 無料公開されている量子コンピュータ
 - <https://www.ibm.com/quantum-computing/>
- 量子コンピュータの初心者向け解説（線形代数、量子力学の知識が必要）
 - <https://dojo.qulacs.org/ja/latest/>

（参考）ビット (bit) と量子ビット (qbit)

	N bit	N qbit
表現する値	2^N 個の数値の集合	2^N 次元のベクトルの集合
N = 1 の例	$x = \{1, 0\}$	$ 1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
N = 2 の例	$x = \{00, 01, 10, 11\}$	$ 0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

実際は 2^N 次元複素空間

AIに必要な機能

一言で機械学習と呼ぶが、いろいろなタイプ（目的）のアルゴリズムが必要。

教師あり（原因と結果の関係を学習）

回帰（予測）

分類（特徴に従って分ける）

異常検知・予知（異常と診断される状態の検知）

教師なし（データ構造を学習）

クラスタリング（特徴を見つける）

異常検知・予知（いつもと違う状態の検知）

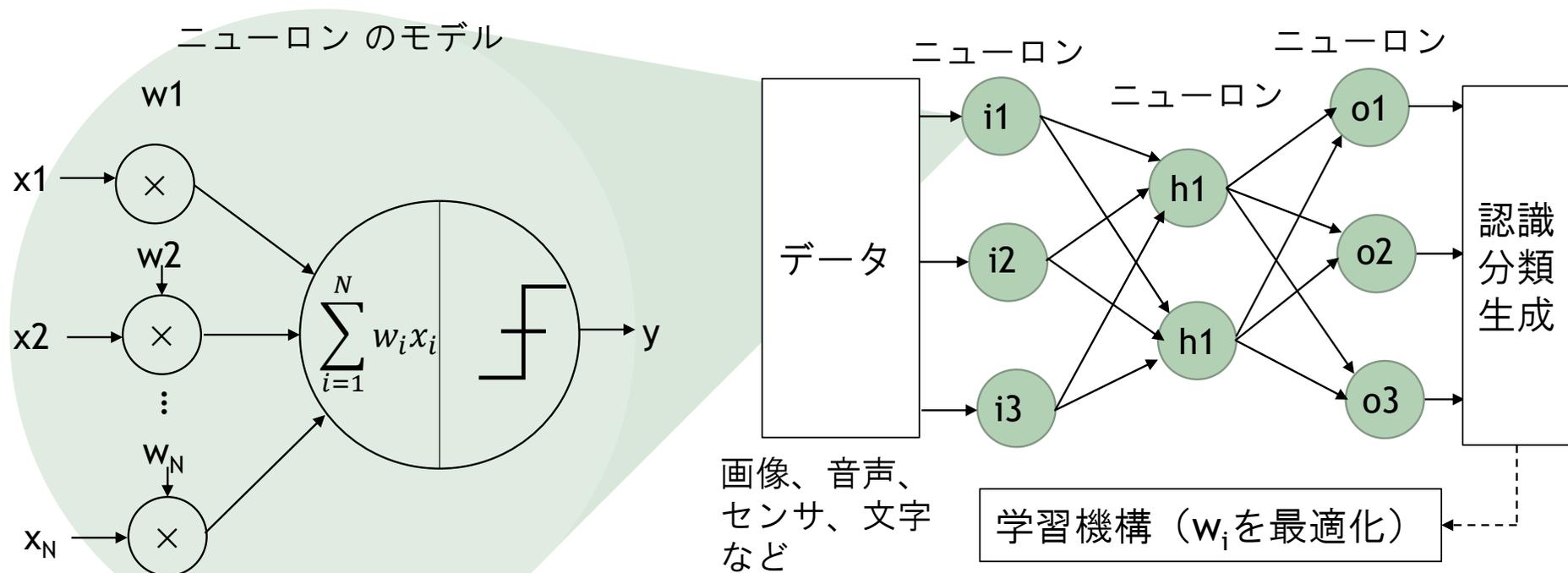
次元削減（重要な特徴を選択する）

強化学習

評価値（点数）を最大にする行動の学習

ニューラルネットワーク（AIの実装方法の一種）

ニューロンの機能と脳のネットワークを模倣（※）

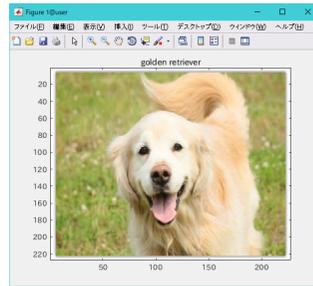


※ 現在はノイマン型コンピュータのソフトウェアで模擬したものが主

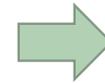
生成AI

認識などのAI

高次元情報（具体的）



符号化
(情報圧縮)



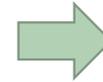
低次元情報（抽象的）
Golden Retriever

通常のAIと逆の処理を行わせる

生成AI
(生成系AI)

低次元情報
Golden Retriever
特徴A, B, C . . .

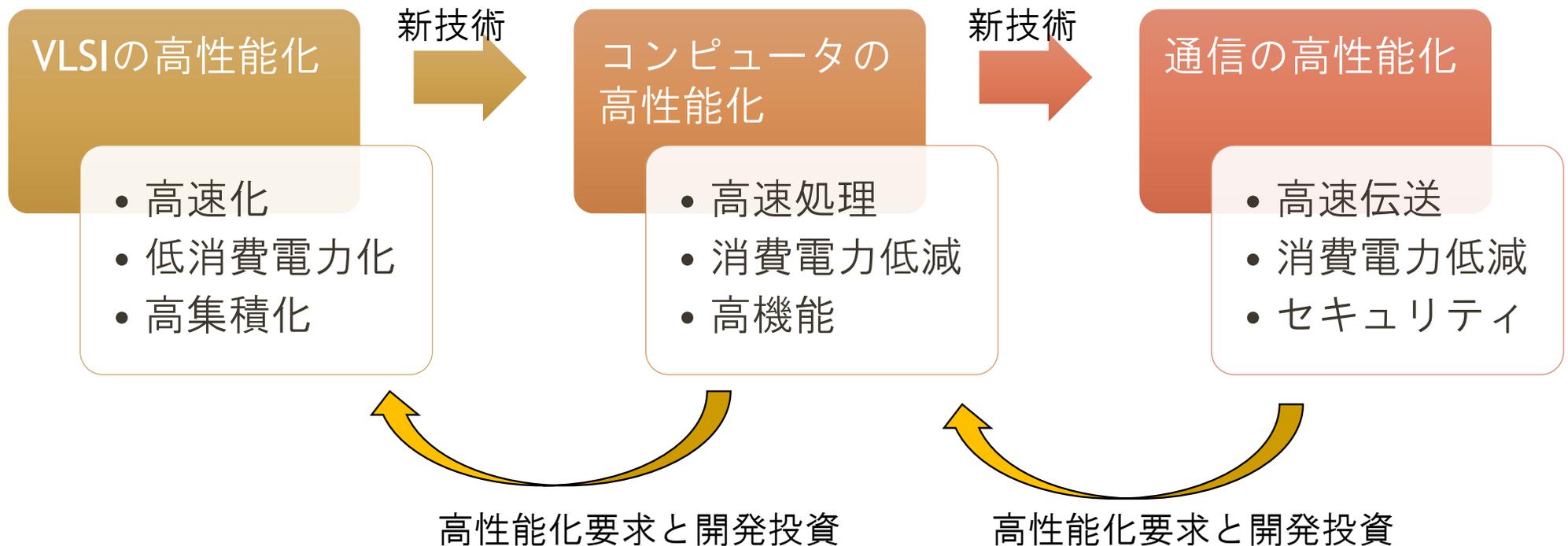
復号
(情報付加)

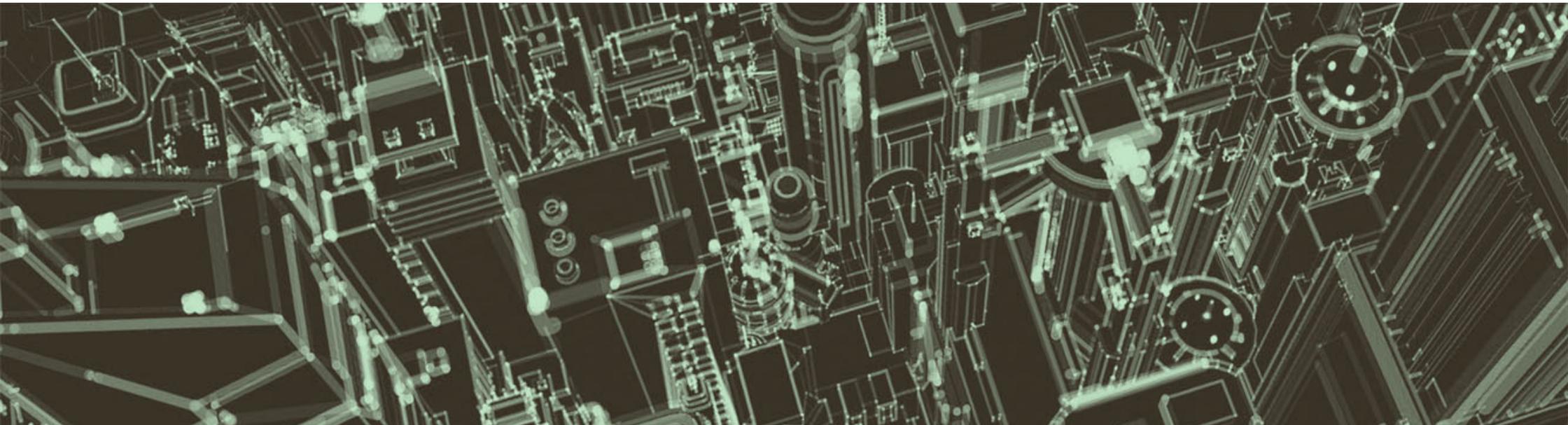


仮想モデルの自動作成（データグリッド）：<https://datagrid.co.jp/>
新聞記事の自動生成（日本経済新聞）：<https://pr.nikkei.com/qreports-ai/>

VLSI・コンピュータ技術・通信技術の発展は密接に関連

全体像を見ないと技術は理解できない（1分野の知識では足りない）





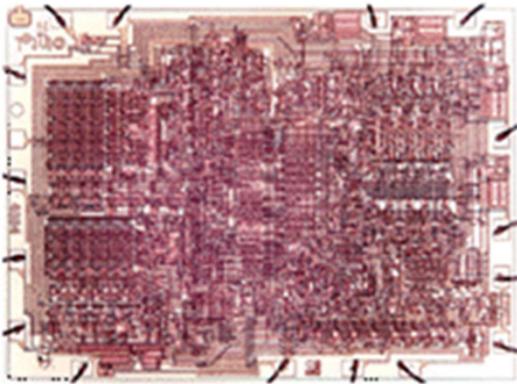
3-2 マイクロプロセッサ

コンピュータの仕組みと構造



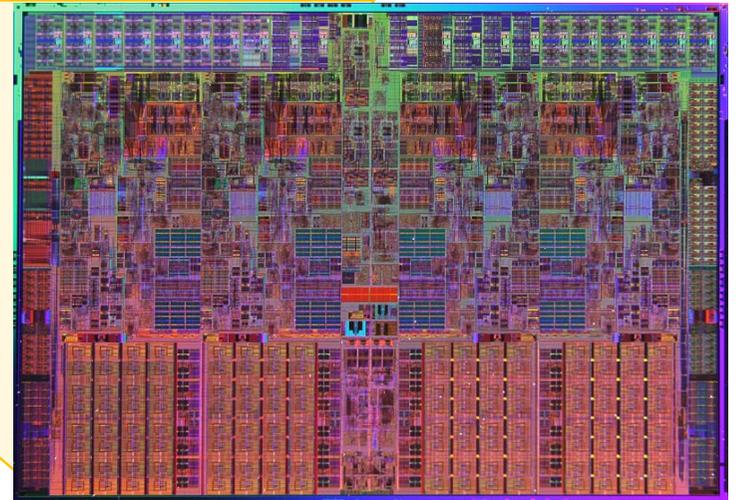
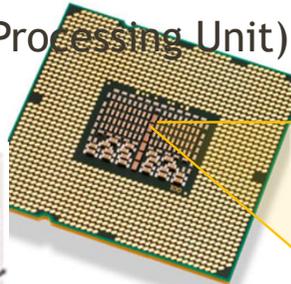
マイクロプロセッサ (Microprocessor)

- ▶ コンピュータの本体の半導体部品
 - ▶ プログラムによって与えられた、計算、判断、記憶などの命令を逐次処理する
 - ▶ 機能としてCPU (Central Processing Unit), 半導体製品としてマイクロプロセッサの名称が使われることが多い



Intel 4004 (2,300 Tr), 1971

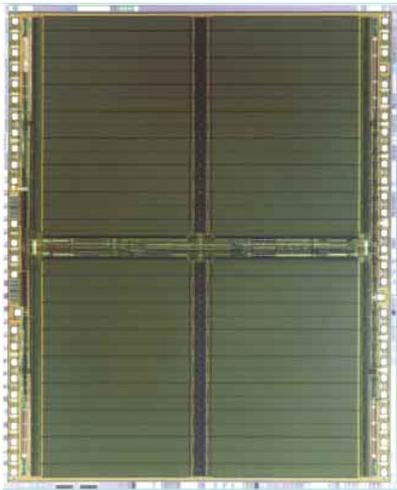
世界で最初に開発されたマイクロプロセッサ



Intel Core i7 (4 Core, 770,000,000 Tr), 2008

メモリ(RAM: Random Access Memory)

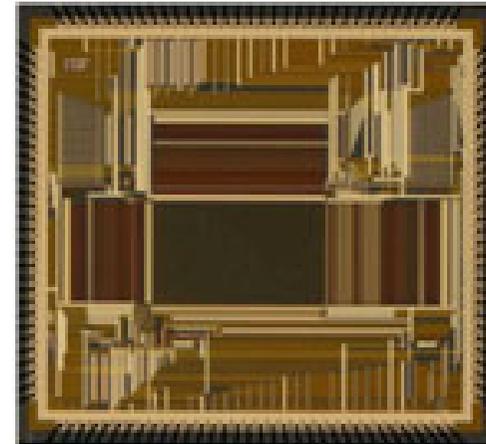
- ❖ コンピュータの記憶機能を担う半導体部品
 - ❖ アドレスを指定して、プログラムや処理対象のデータを記憶しておく
 - ❖ アクセス速度・容量・揮発/不揮発・動作原理などが異なる多くの種類がある



出典: 三菱電機 8Mbit **SRAM**
(マイクロプロセッサに内蔵)

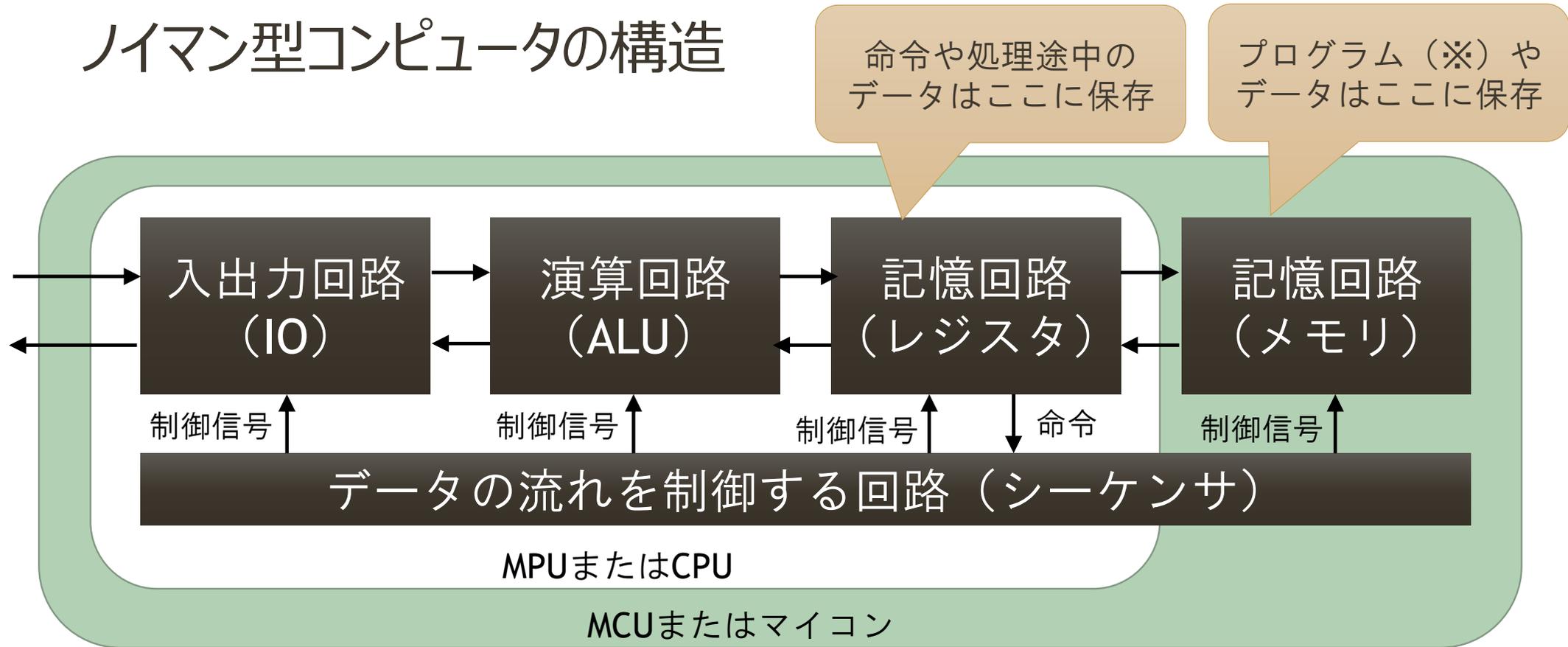


出典: NEC 64Mbit **DRAM**
(PC用の主記憶や画像処理VLSIに内蔵)



出典: シャープ・金沢大 256kbit ReRAM
(**新型不揮発性メモリ** = 主記憶と補助記憶を統合する高速不揮発性メモリ)

ノイマン型コンピュータの構造



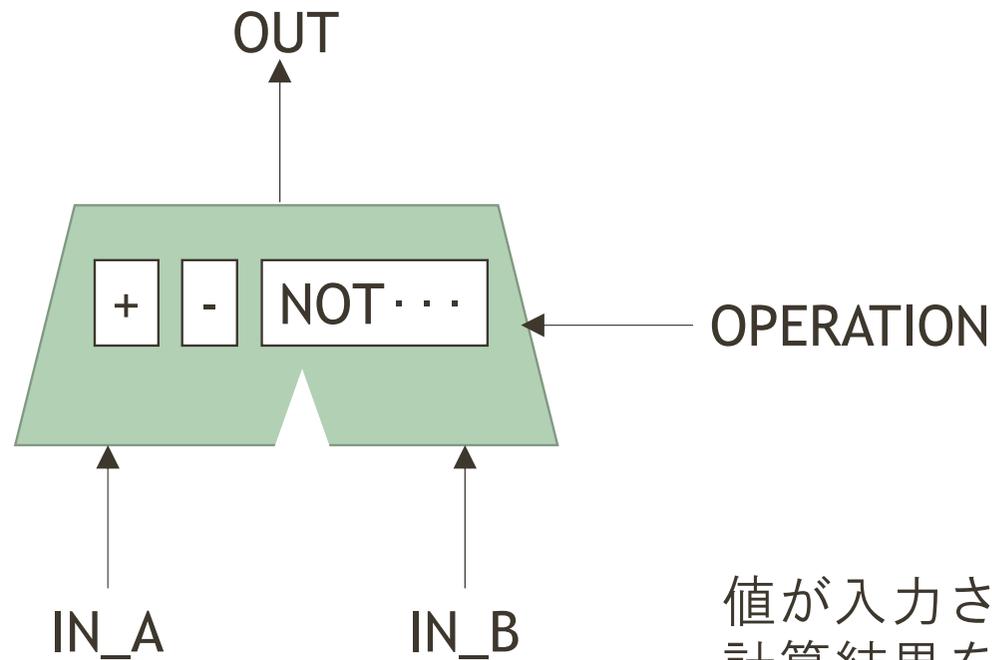
※プログラムは、コンピュータへの命令（次頁参照）の実行手順を示したもの

ハードウェアの用語（覚えておこう）

用語	意味
MPU (Microprocessor Unit)	コンピュータの主要部分（前ページ参照）を搭載したLSI (Large Scale Integration)
CPU (Central Processing Unit)	MPUと同義（ただし、物ではなく機能を指す名称）
MCU (Microcontroller Unit)	MPU + メモリ + タイマー + アナログ-デジタル変換 + 通信回路（無線の場合もある）を1チップに集積化したLSI。また、プログラムを内蔵する（書き込む）ことができる
SoC (System on a Chip)または System LSI	多数のMPUを搭載し1チップでシステム全体を構成できる特定用途向けの大規模LSI（スマートフォンLSI等）
命令セット (Instruction set)	プログラムを構成する処理要素の集合（MPUが実行する） <ul style="list-style-type: none">• 算術演算 $A + B$, $A * B$ など• 論理演算 $A \cap B$, $A \cup B$ など• 比較判定 $A > B$, $A = B$ など• データ読み出し・保存• ジャンプ（プログラムの実行順序を変更する命令）

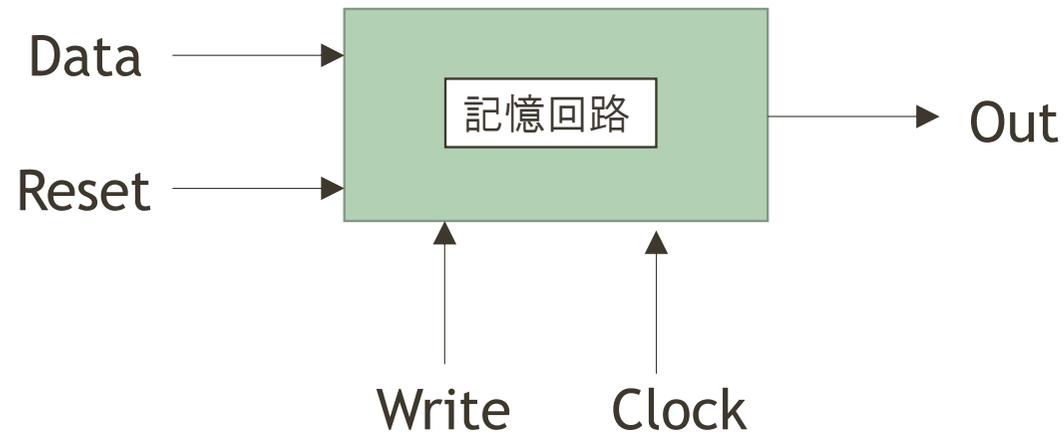
これがソフトウェアとハードウェアを接続している

ALU(算術論理演算ユニット)の動作



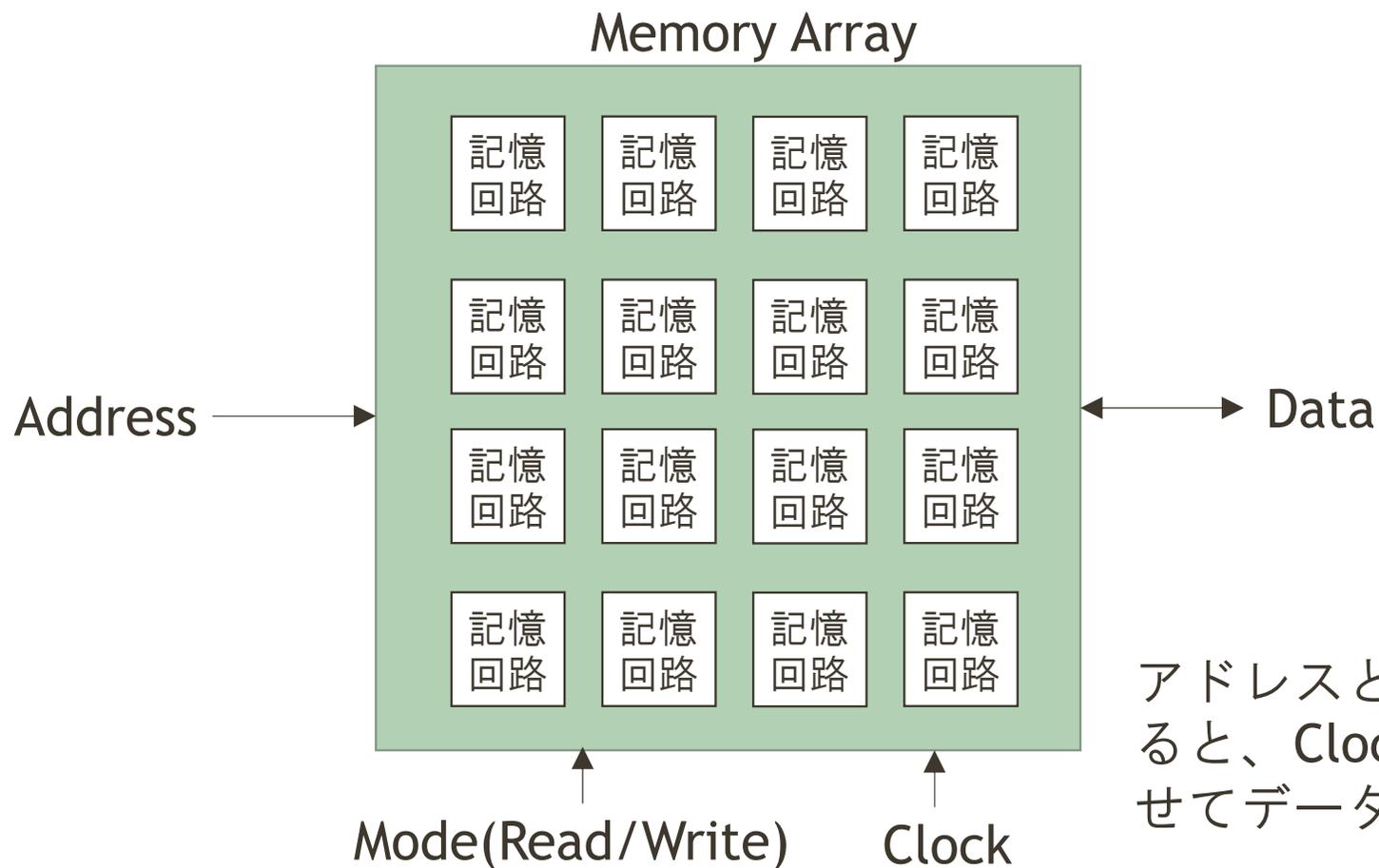
値が入力されるとすぐに
計算結果を出力する

レジスタの動作



Clockエッジに合わせてデータを入力して記憶する。
記憶したデータはOutに出力し続ける。
Reset = 1 のとき記憶データを初期化する (0にする)。

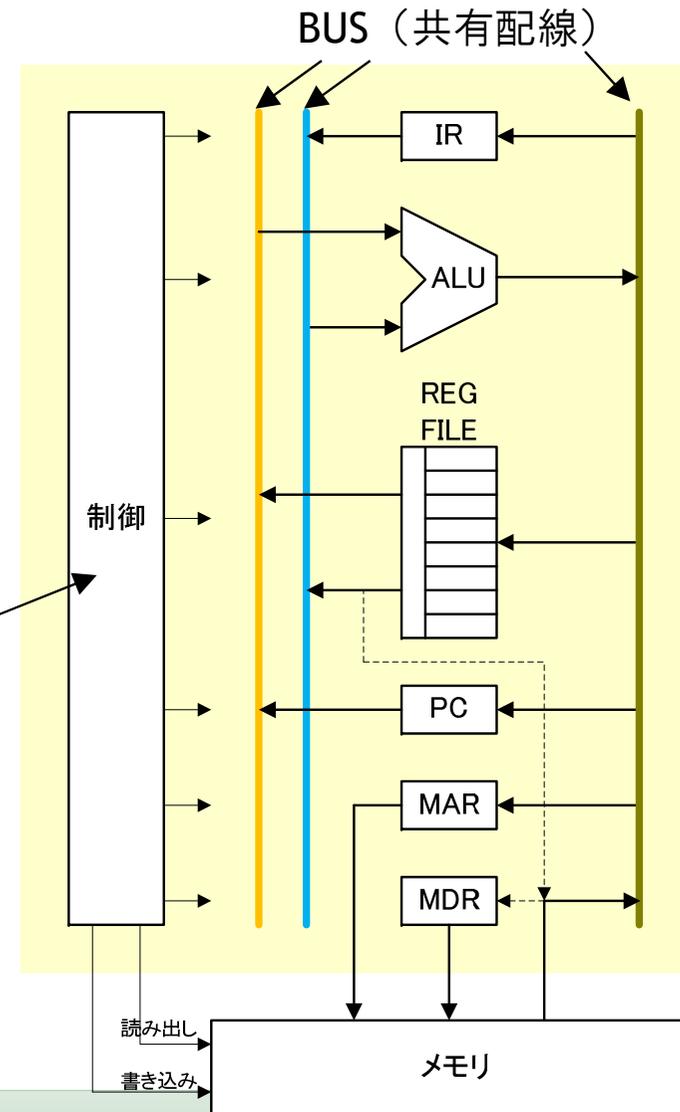
メモリの動作



マイクロプロセッサのデータパス

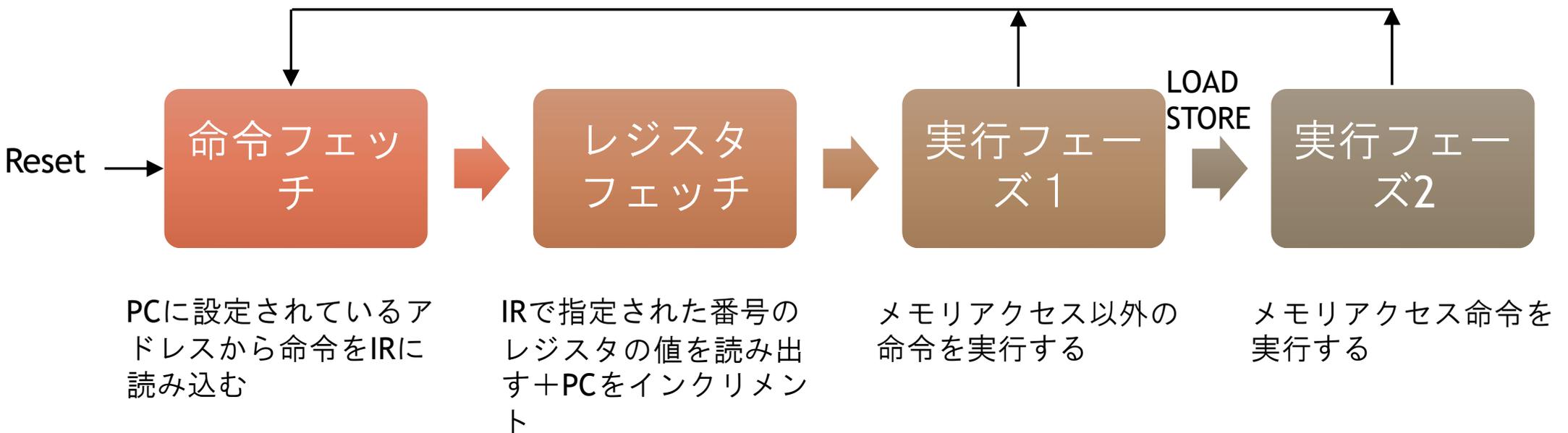
- IR: 命令レジスタ
 - メモリ読み出した命令を一時保存する
- ALU: 算術論理演算ユニット
 - 算術演算や論理演算を行う
- REG FILE: レジスタファイル
 - データやアドレスを一時保管する
- PC: プログラムカウンタ
 - 次に実行する命令のアドレスを設定する
- MAR: メモリアドレスレジスタ
 - アクセスしたいメモリのアドレスを一時保管する
- MDR: メモリデータレジスタ
 - メモリに書き込むデータを一時保管する

各ブロックの機能や配線接続を制御



マイクロプロセッサのシーケンスの例

- データパスの各ブロックを動作させる毎に時間が必要なので、命令毎に次のような状態遷移を行い、これを繰り返す



命令セット(Instruction Set Architecture)

- 命令 = オペコード + オペランド (ソフトウェアとのインタフェースを規定)
- オペコード = 命令の種類 (例: 加算、レジスタ値のコピー、メモリからのデータの読み込み、条件分岐など)
- オペランド = 操作対象の指定 (例: レジスタ名、メモリのアドレス、条件分岐先など)

ニーモニック	処理内容	説明
ADD d, s	$d = d + s$	加算
MOV d, s	$d = s$	レジスタのコピー
LD d, (s)	$MAR = s, d = MDR$	データの読み込み
BMI s, x	$\text{if } (s < 0) \text{ pc} = \text{pc} - 1 + x$	条件分岐

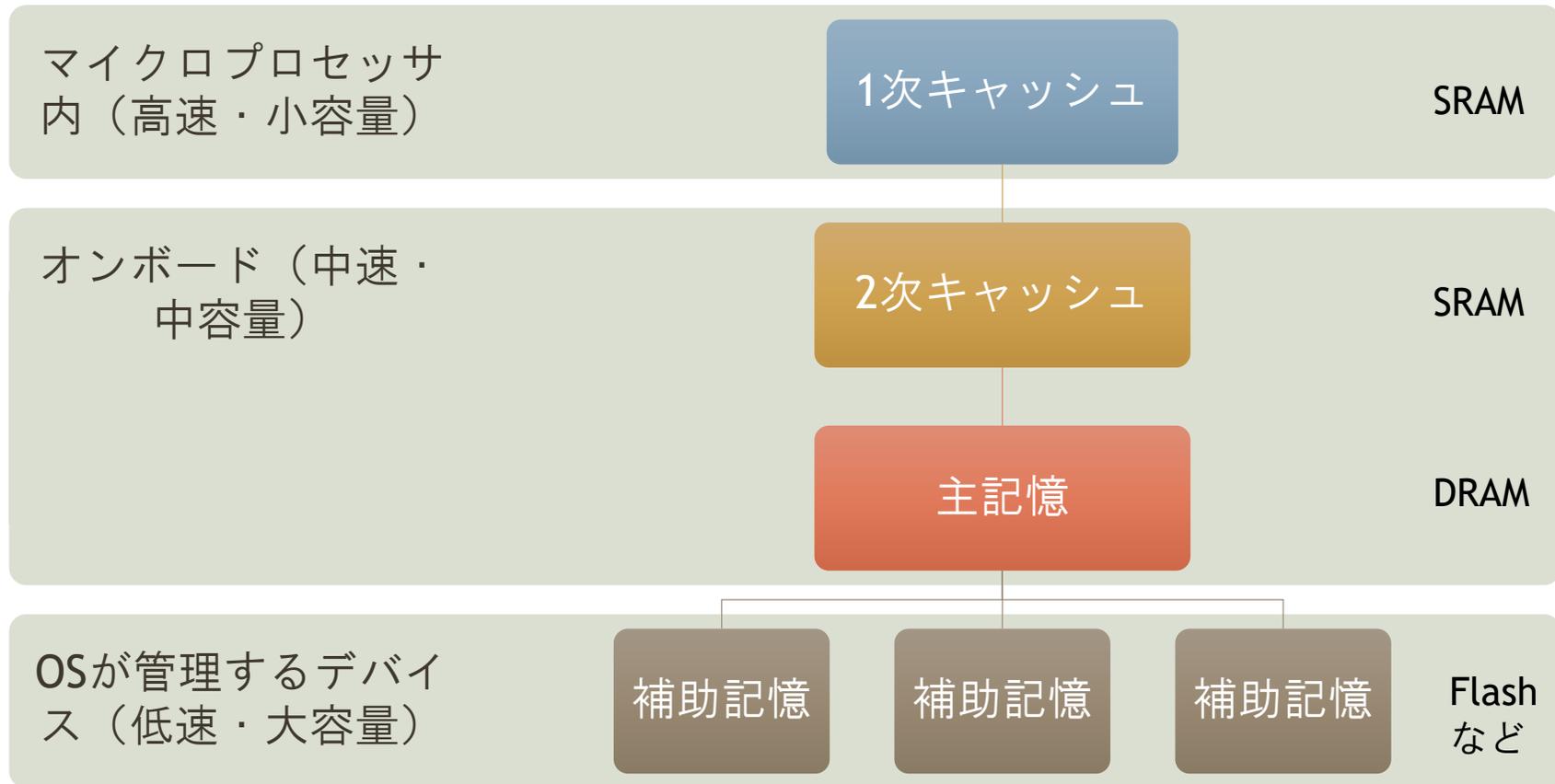
[参考] 複雑なプログラムをプロセッサの命令セットを使って書くのは大変なので、コンパイラやアセンブラで変換する。フローは、高級言語 (ソース) → [コンパイラ] → アセンブリ言語 (ニーモニックなど) → [アセンブラ] → バイナリ (実行形式)

[参考] 命令の構造によるプロセッサの分類

分類	オペランドの制限	プロセッサ例
Register-Registerマシン (RISC: Reduced Instruction Set Computer)	メモリアドレスをオペランドに直接指定できない（命令の長さが一定）。 処理 = 多くの命令数が必要 命令 = 高速化しやすい	SPARC, MIPS, POWER, ARM
Register-Memoryマシン	メモリアドレスをオペランドに1個指定できる。	Intel x86, Motorola 68
Memory-Memoryマシン (CISC: Complex Instruction Set Computer)	メモリアドレスをオペランドに2個または3個指定できる（命令の長さが可変）。 処理 = 少ない命令で実行 命令 = 回路が複雑になり高速化しにくい	VAX-11, NS32X32

[注] 現在では良いところ取りになっており、分類に当てはまらないプロセッサが多い。

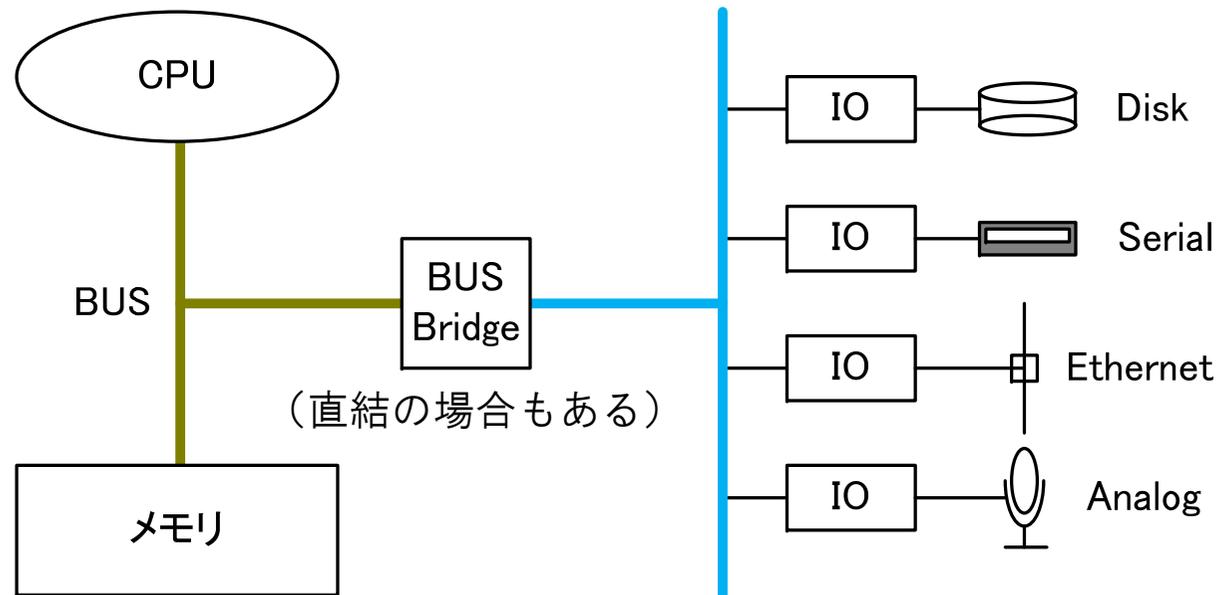
メモリの構造

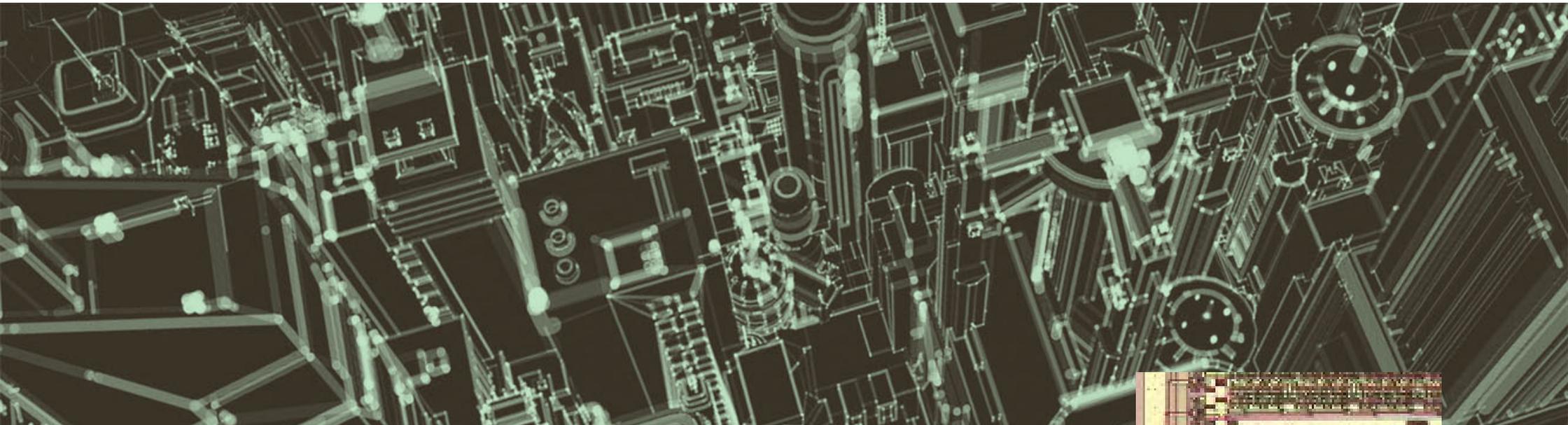


将来的には新型不揮発性メモリにより統合

IOと割り込み

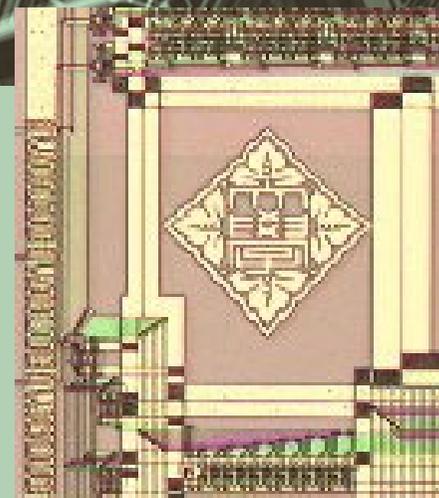
- ▶ コンピュータは、入出力(IO)を通して、制御や通信処理などの実際の仕事を行う
- ▶ CPU（プログラム）からIOをチェック = ポーリング(Polling) → CPUに負荷がかかる
- ▶ IO（他にタイマなど）からCPUに処理を要求 = 割り込み(Interrupt) → 処理ルーチンにジャンプ





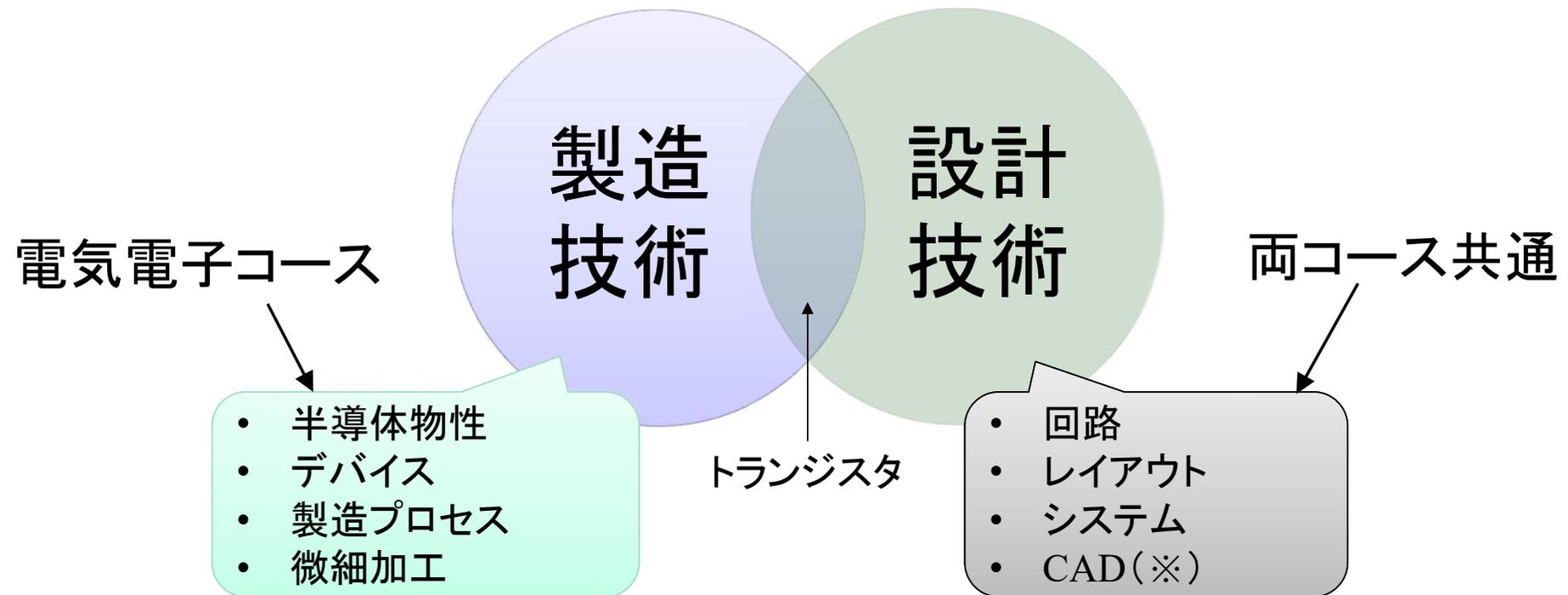
3-3 VLSI設計技術

マイクロプロセッサの情報化



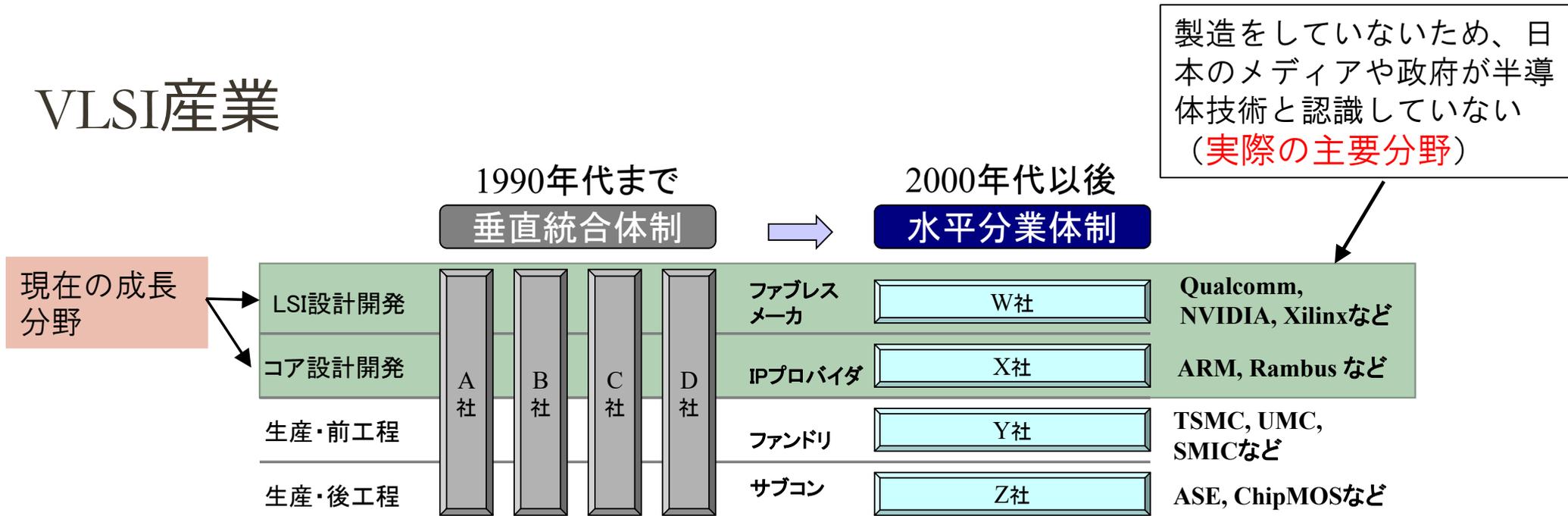
集積回路(VLSI)技術の構成

VLSI技術は2つの専門分野で成り立っている



※ Computer Aided Design (コンピュータ支援設計)

VLSI産業

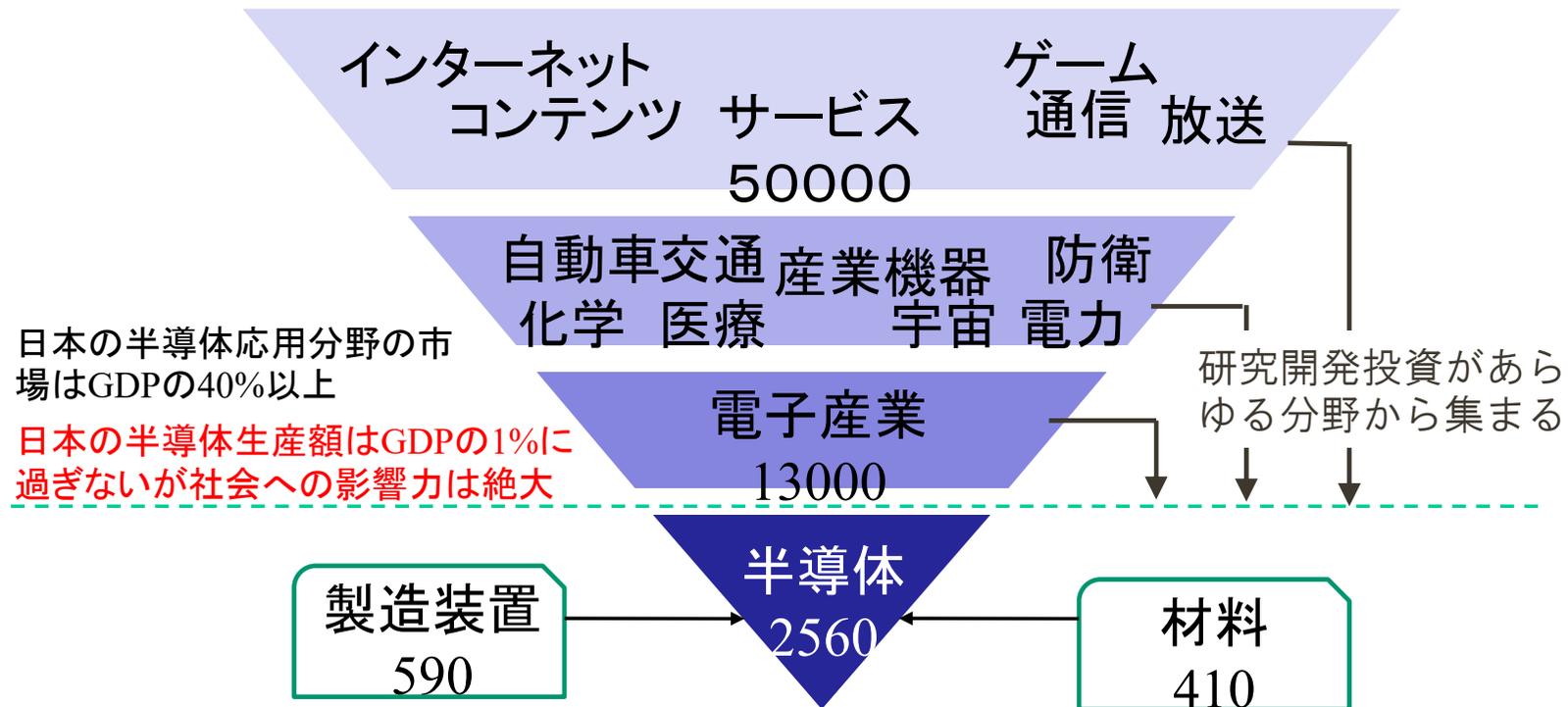


(注) LSI技術分野では、IP(Intellectual Property)は設計データなどの設計資産を指す

重要

1990年代までと大きく異なり、半導体の垂直統合企業が少数となり、半導体産業は国際分業化された。VLSI設計開発分野は、半導体に分類されていない先端技術分野（情報通信、AI、宇宙、医療、バイオ、エネルギーなど）に移行している。このため、VLSI設計技術が、先端技術分野全分野の基本知識となりつつある。

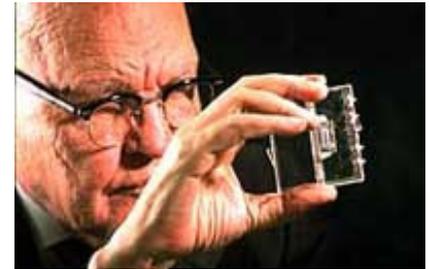
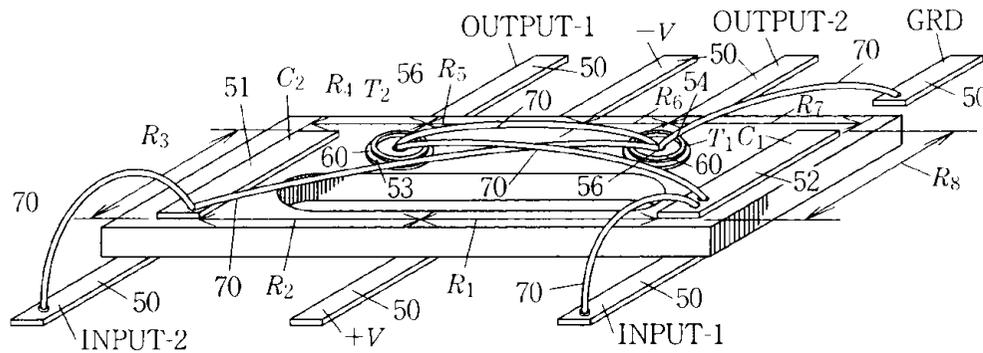
産業の米



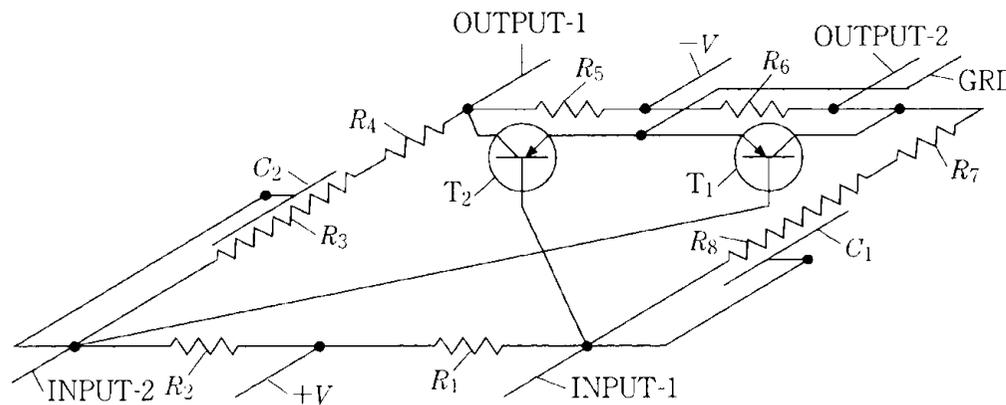
単位: 億 \$ (世界)

F. Huang, Green IT international symposium
ICハンドブック, JEITA, 2009.

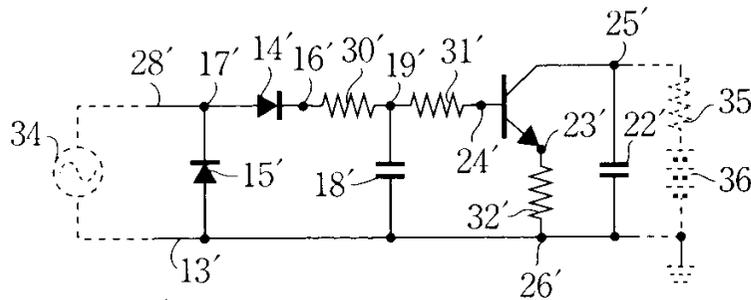
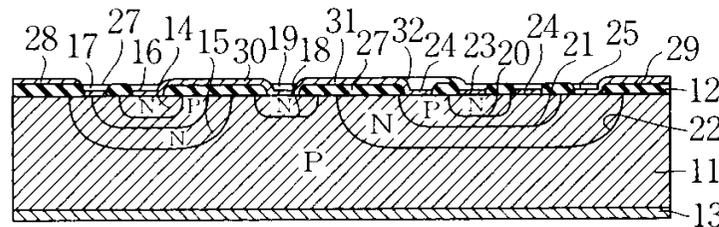
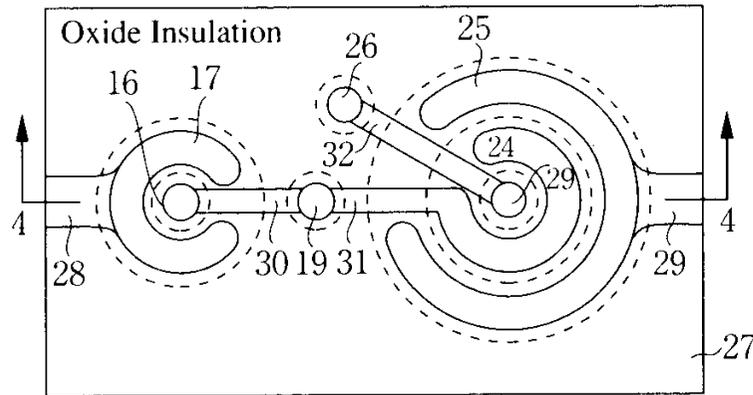
Kilbyの特許



Jack S. Kilby,
2000 ノーベル物理学賞受賞
出典: TI Inc.



US Patent
No. 2 138 743, 1964



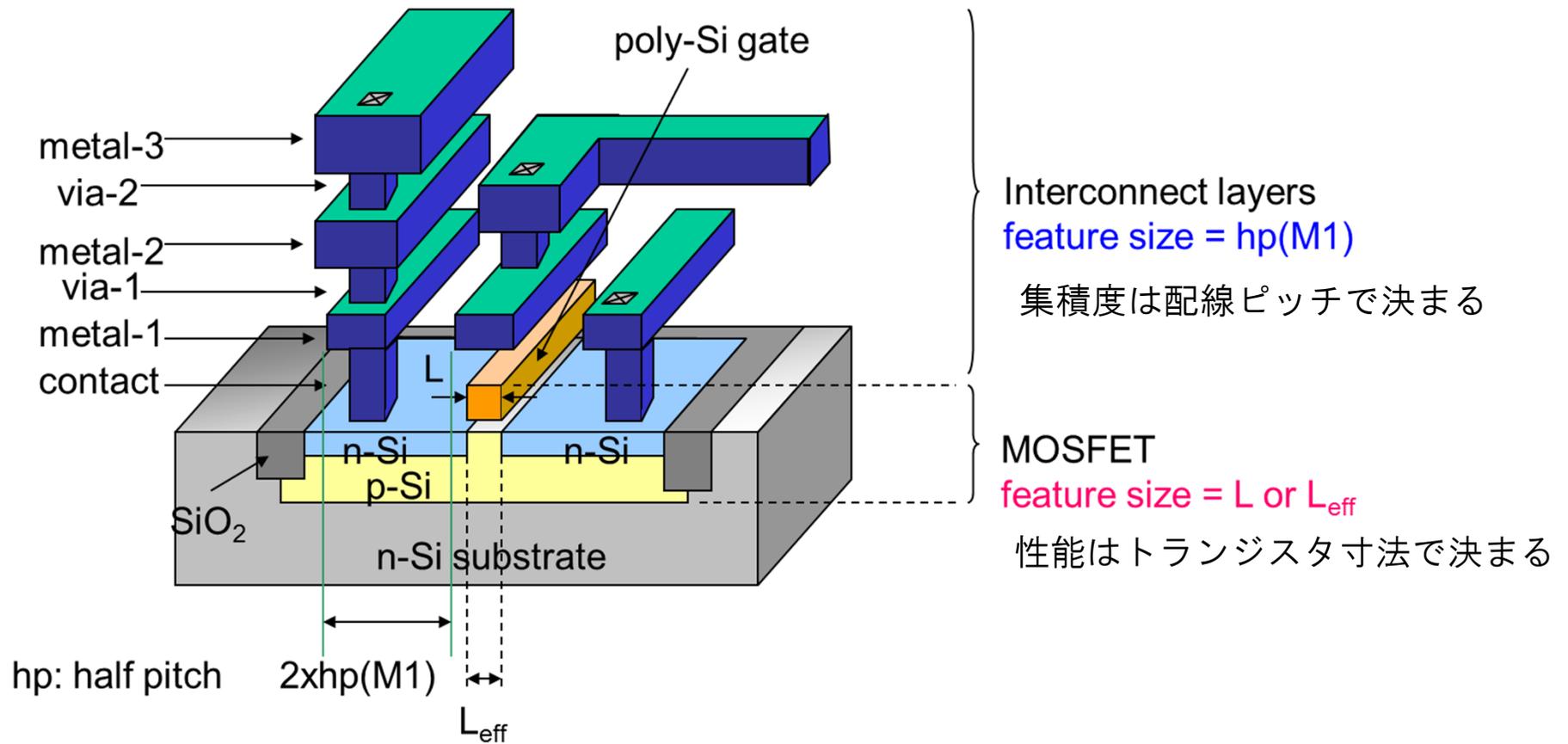
Noyceの特許



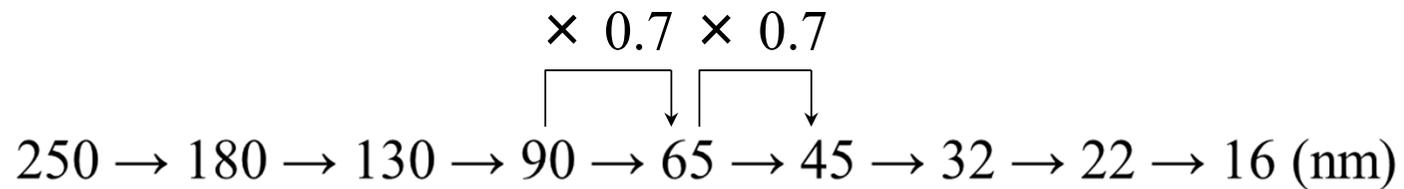
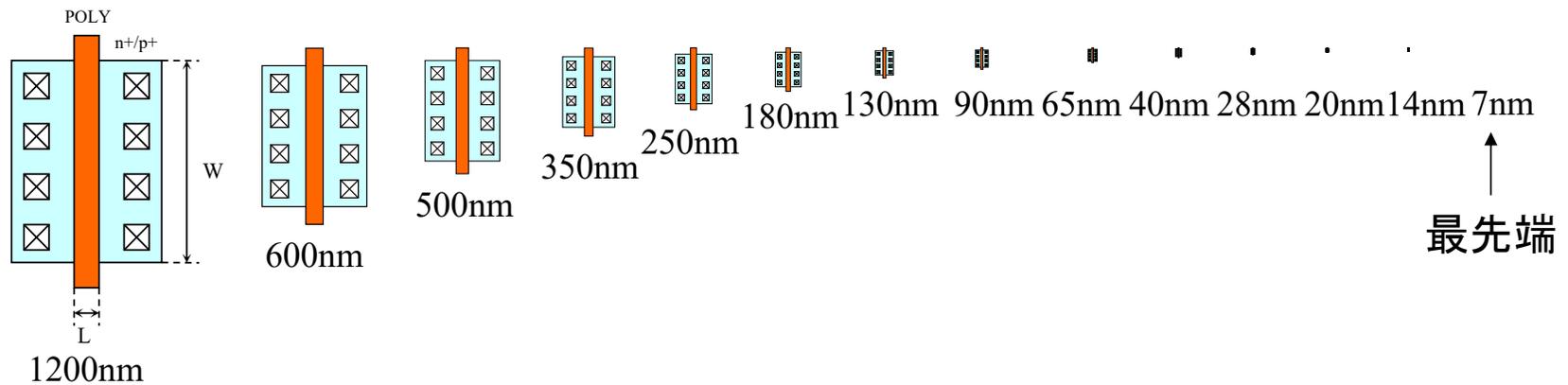
Robert Noyce,
Fairchild Semiconductor
International, Inc.
出典: Innopedia

US Patent
No. 2 981 877, 1961

VLSI内のトランジスタの構造 (MOSFETの構造)

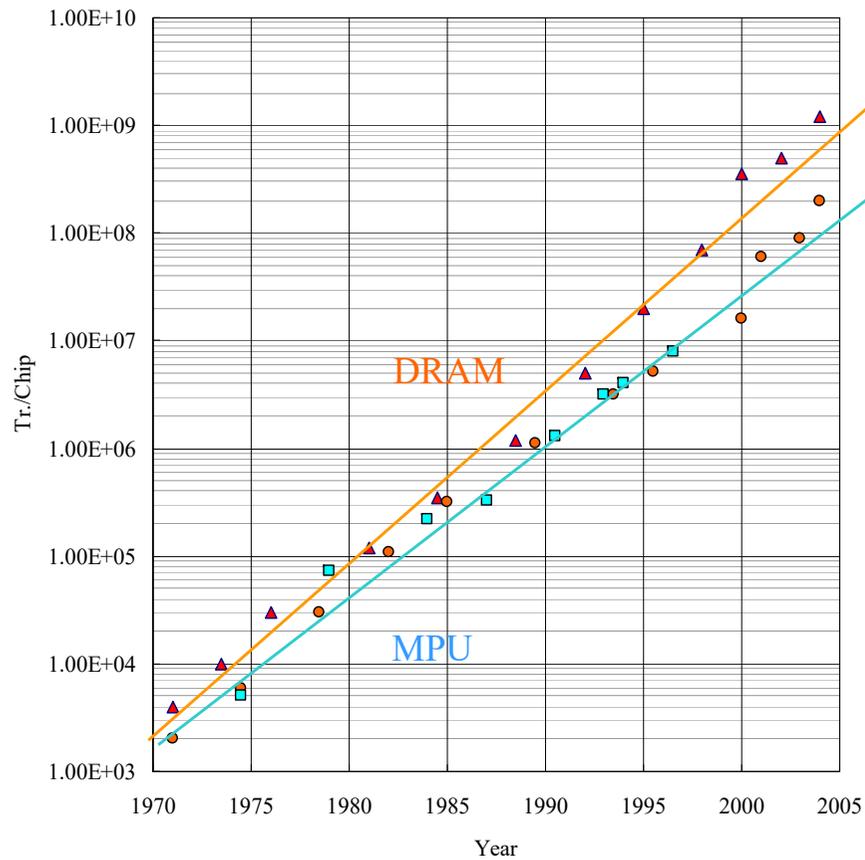


MOSFETの微細化



0.7倍/3年のペースで微細化が進められている。

ムーアの法則



Gordon Moore
Intel co-founder

出典: Intel Corporation

微細化技術の進歩による集積度向上

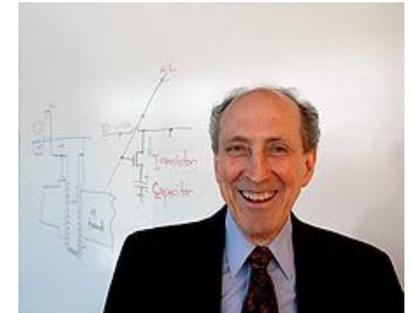
集積度 = 1チップのトランジスタ数の変化

プロセッサ
年率59%=4倍/3years

メモリ(DRAM)
年率73%=3倍/2years

デナードのスケーリング則, 1974

- トランジスタ(MOSFET)の微細化による性能と集積度の両方の向上が、微細化技術開発の原動力
- スケーリング則
 - トランジスタの寸法を、縦、横、高さを同じ縮尺 (k 倍) で微細化
 - 電圧を、寸法と同じ縮尺 (k 倍) に低減 (= 電界を一定に保つことに相当)

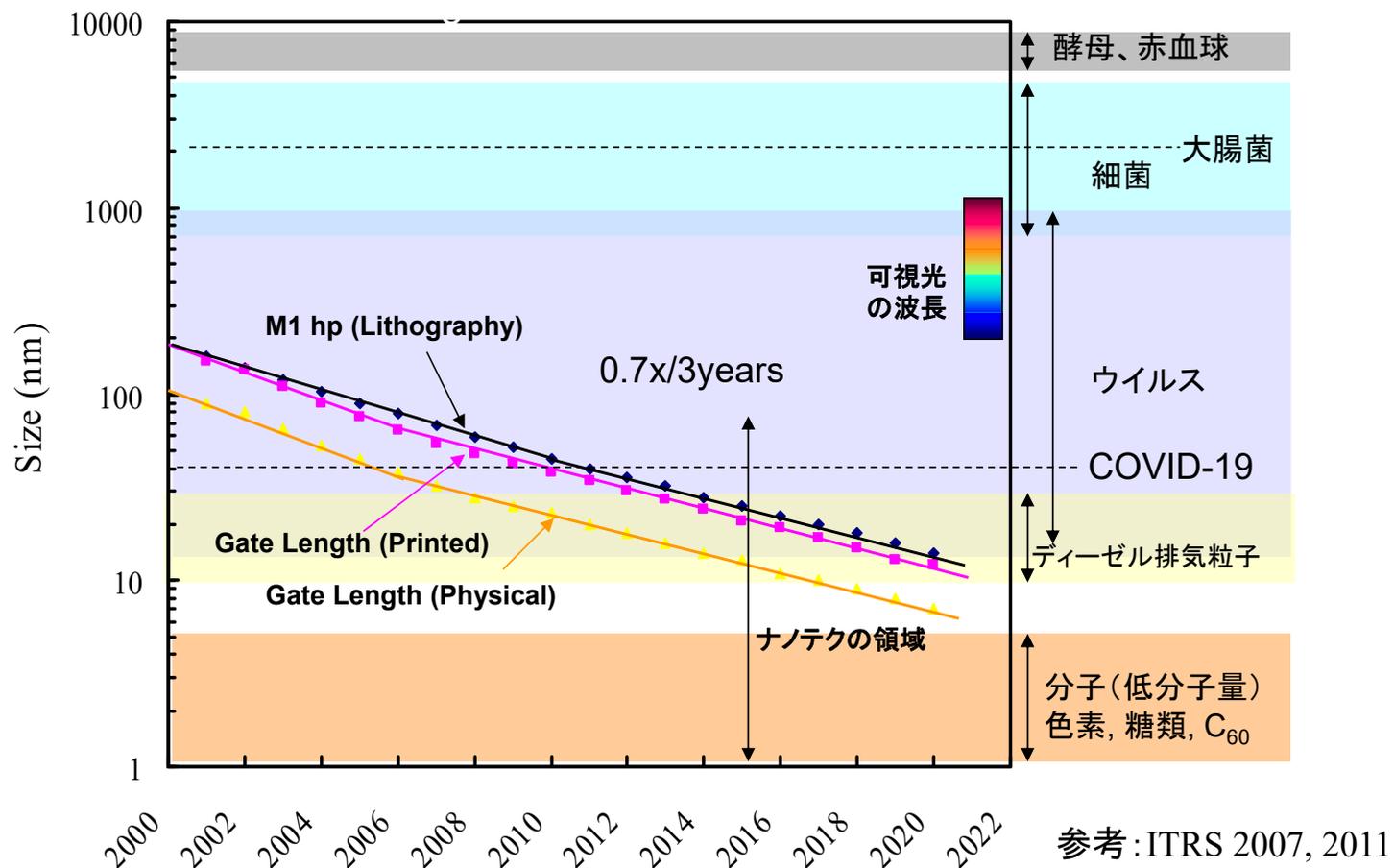


Robert H. Dennard,
1968 DRAM発明
出典:Wikipedia

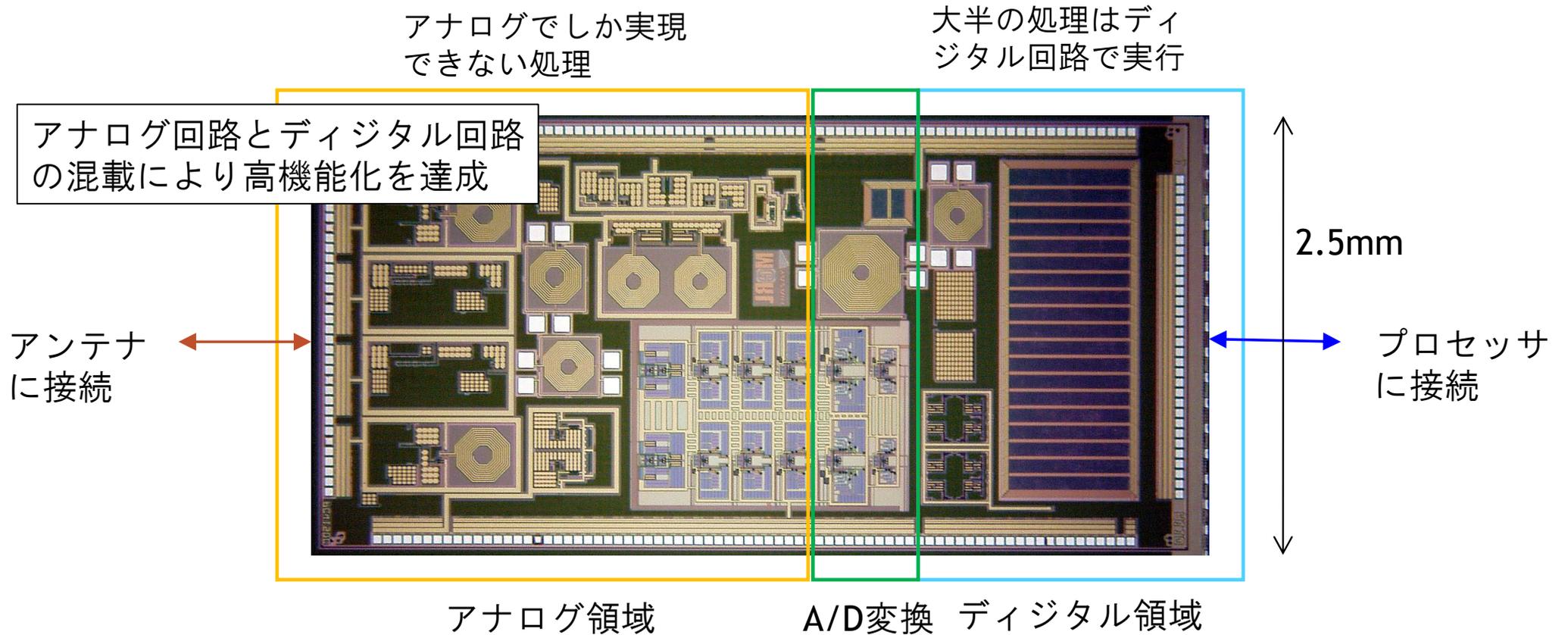
スケーリング効果 (理論的に求められる)

	集積度	動作速度	消費電力/トランジスタ	消費電力/チップ
k倍スケーリング	$1/k$	$1/k$	k^2	1
$k = 0.7$ のとき	2	1.4	0.5	1

スケーリングの動向

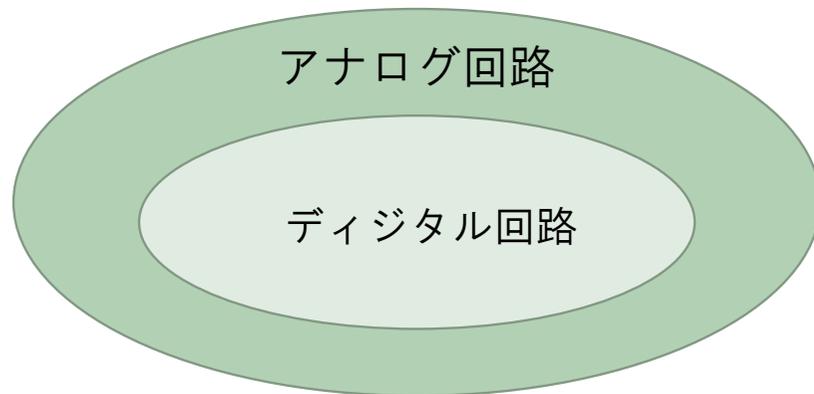


無線機能の集積化（金沢大学）



アナログ回路とデジタル回路

主な処理はデジタル回路＋外部インタフェースはアナログ回路で実現

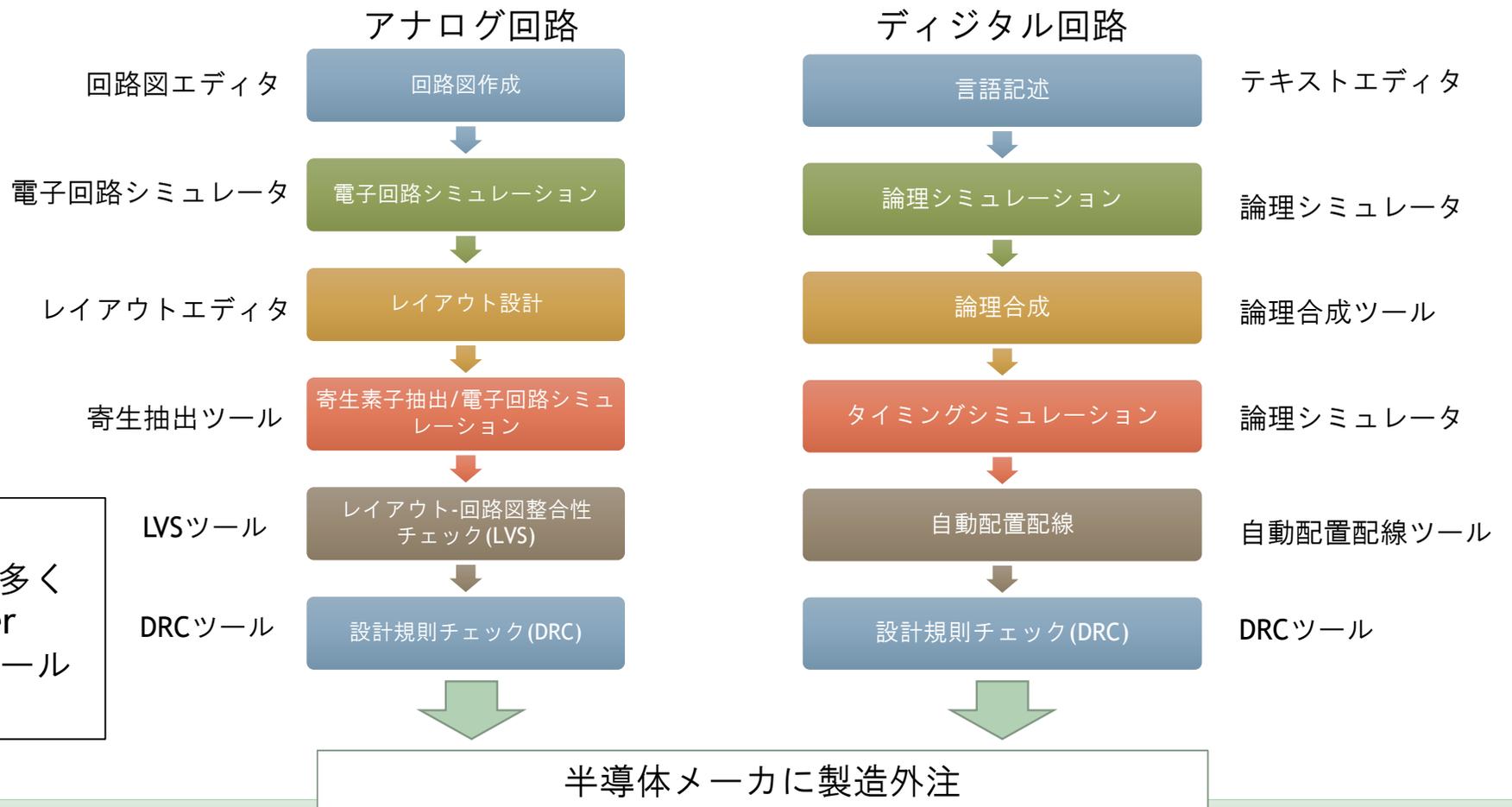


アナログ回路	デジタル回路
<ul style="list-style-type: none">• 高速• 低消費電力• 精度の保証は難しい• 遅延が小さい• 手動設計	<ul style="list-style-type: none">• 低速• 消費電力が比較的大• 高精度• 遅延が大きい• 自動設計

アナログ回路でないと実現できない機能

- データ伝送の無線化
- 電力供給の無線化
- アナログ信号とデジタル信号の相互変換
- デバイスのドライブ（液晶ディスプレイ、スピーカ、モータなど情報出力）
- センシング（イメージセンサ、バイオセンサなどの情報入力）

アナログ回路とデジタル回路の設計法の違い



ポイント！
実際の設計には多くのCAD(Computer Aided Design)ツールが必要。

IP(Intellectual Property)コア

- ✦ マイクロプロセッサのHDL設計は、「集積回路設計及び演習」で詳しく学ぶ
- ✦ マイクロプロセッサなどの設計データ（HDL）は「IPコア」と呼ばれ、ソフトウェアと同様に電子媒体として流通している（無料のものも多い）
 - ✦ IPコアがあれば、論理合成 + 自動配置配線により、半導体メーカーにVLSIを製造外注できる
 - ✦ 設計だけを行い、製造を外注するメーカーを**ファブレス**という（現在の、半導体メーカーの多くがファブレス）
 - ✦ 製造だけを行うメーカーを、**ファンドリ**という（設計と製造が分離していることを、**国際的水平分業**という）
- ✦ 代表的なオープンソースIPのコミュニティサイト
 - ✦ OpenCores <https://opencores.org/>
 - ✦ 金沢大学でも研究教育目的限定で、一部のIPを公開 <http://www.merl.jp/>

ARMプロセッサ（スマホの標準CPU）

- 代表的な32bitマイクロプロセッサ
 - 市場シェア: 組み込みシステム 75%, スマートフォン ほぼ100%
 - IPとして販売されている（ライセンス料+ロイヤリティが収入）
 - 1コア = 5円以下？
 - ARM Cortex シリーズがある
 - Cortex-A: アプリケーションプロセッサ向け
 - Cortex-R: リアルタイム制御向け
 - Cortex-M: 組み込みシステム向け
 - ARM Cortex-M0
 - IoT（次回講義）向けの低消費電力・小型のプロセッサ
 - 教育用として論理合成可能なHDL(Hardware Description Language)コードを無料で配布

HDL記述例（算術論理演算ユニットの例）

```

vlsisv04{kitagawa}25: cat alu.v
'timescale 1ns/1ns

`define AND 3'b000
`define OR 3'b001
`define NOT 3'b010
`define XOR 3'b011
`define ADD 3'b100
`define SUB 3'b101
`define ACC 3'b110
`define DAT 3'b111

module alu (alu_out, accm, data, opcode);

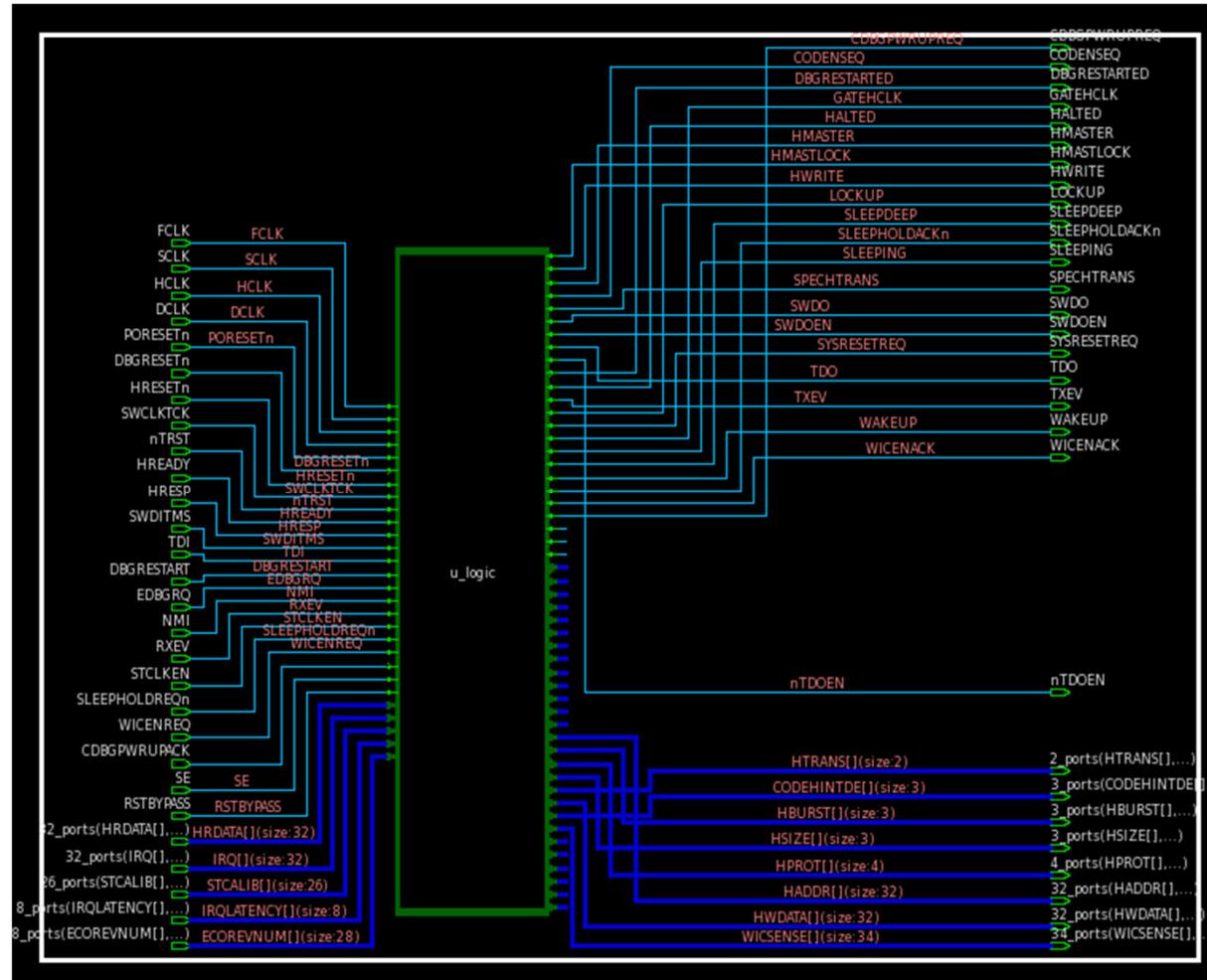
    input  [7:0] accm, data;
    input  [2:0] opcode;
    output [7:0] alu_out;
    reg    [7:0] alu_out;

    always @(accm or data or opcode) begin

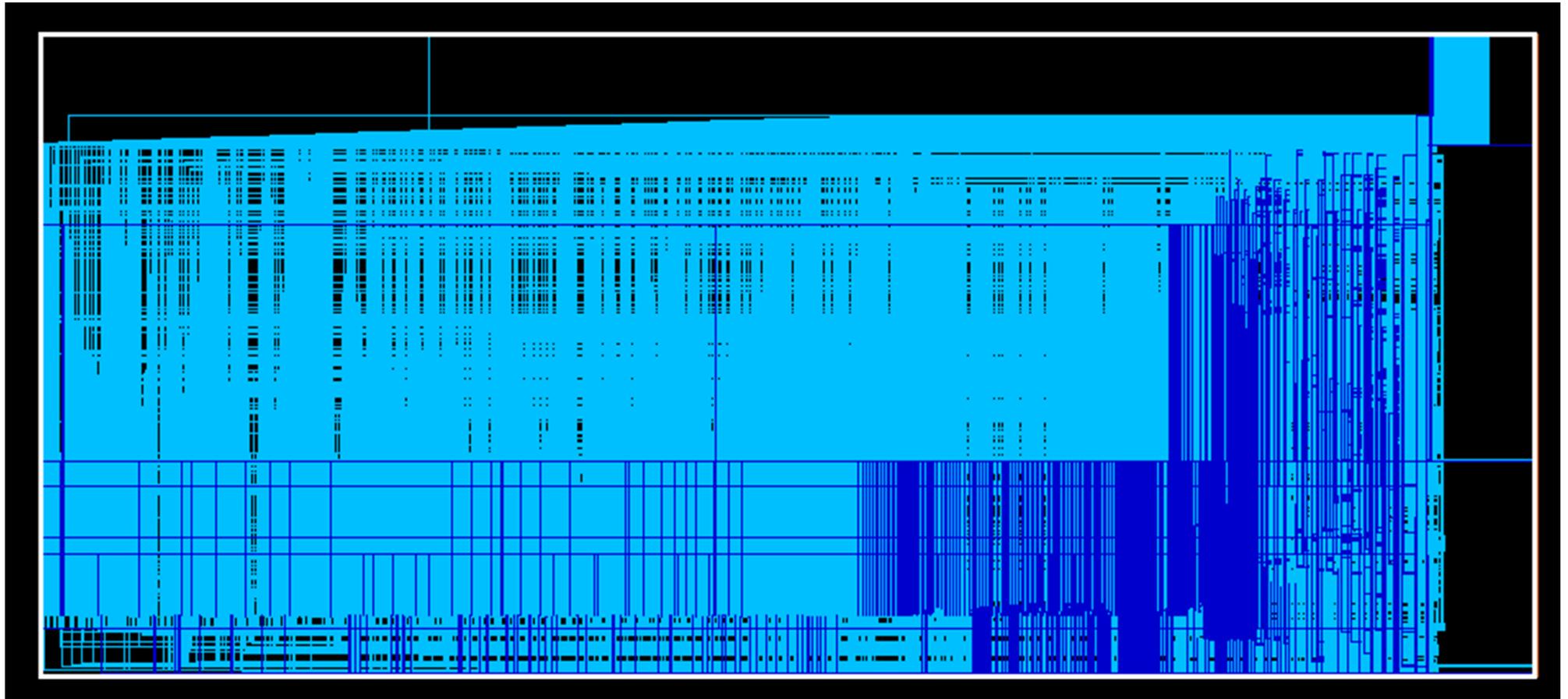
        case (opcode)
            `AND : alu_out <= accm & data;
            `OR  : alu_out <= accm | data;
            `NOT : alu_out <= ~accm;
            `XOR : alu_out <= accm ^ data;
            `ADD : alu_out <= accm + data;
            `SUB : alu_out <= accm - data;
            `ACC : alu_out <= accm;
            `DAT : alu_out <= data;
            default : alu_out <= 8'bxxxxxxxx;
        endcase

    end
endmodule
vlsisv04{kitagawa}26: █
```

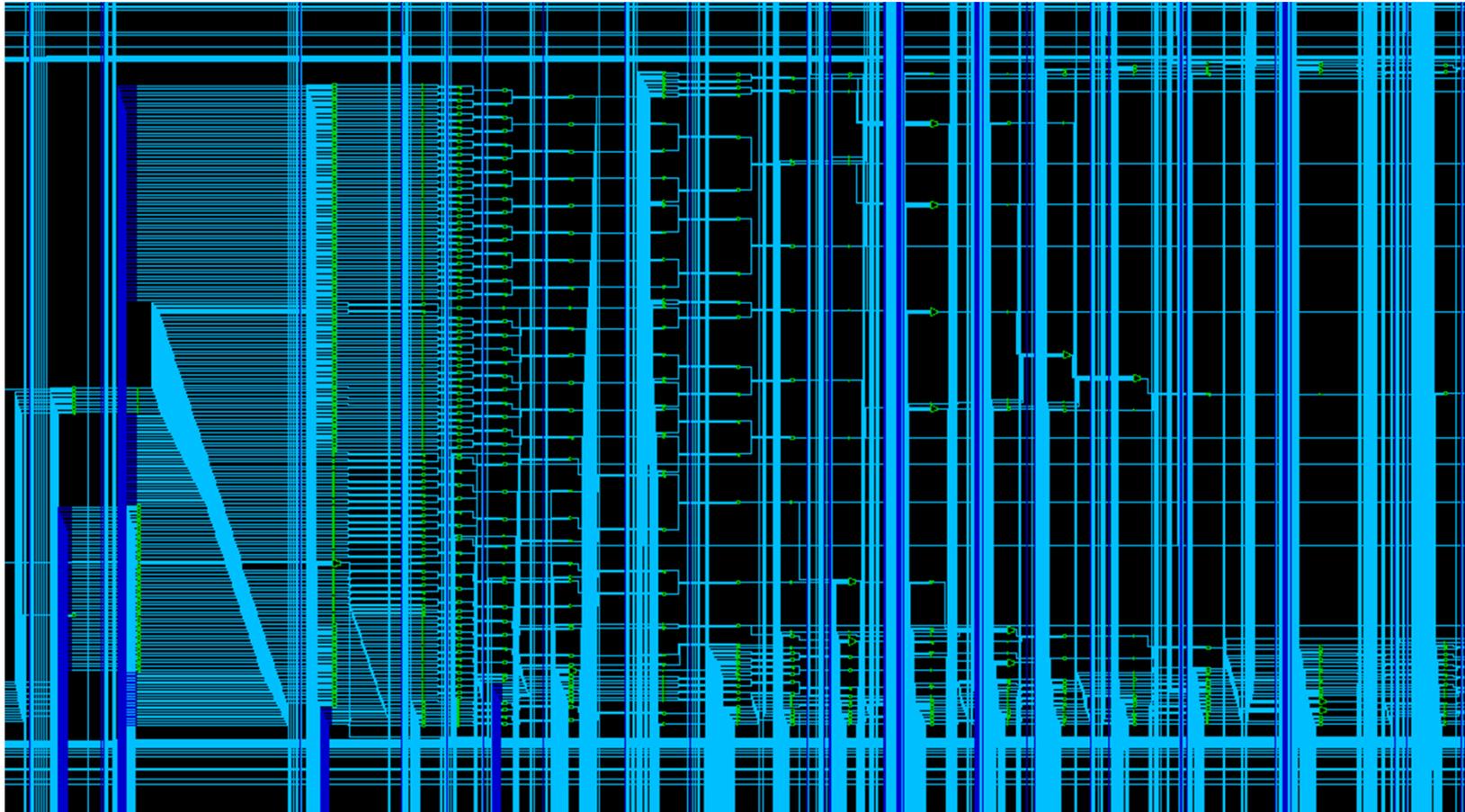
論理合成前



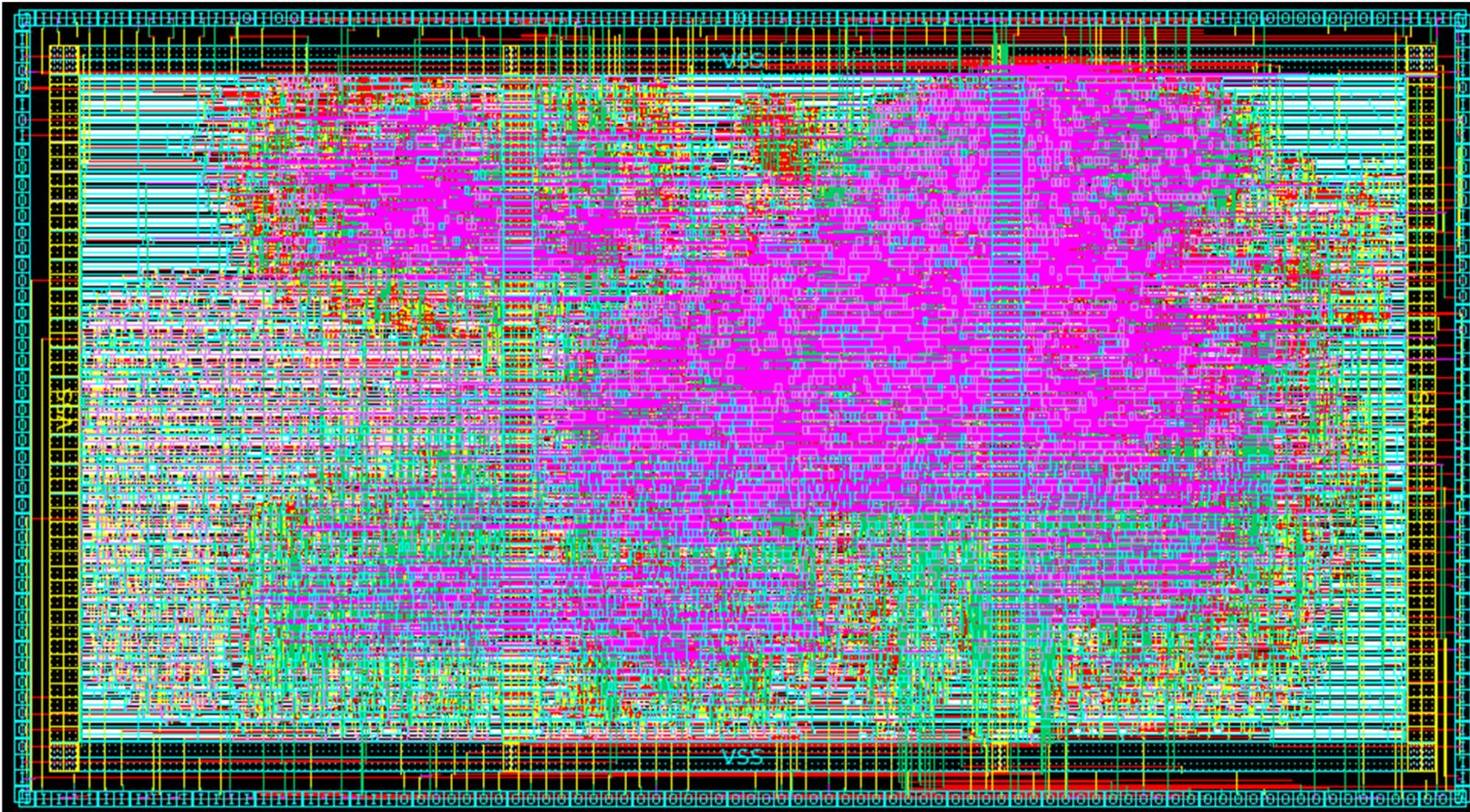
論理合成 (13202ゲート, クロック100MHz)



論理合成 (拡大)



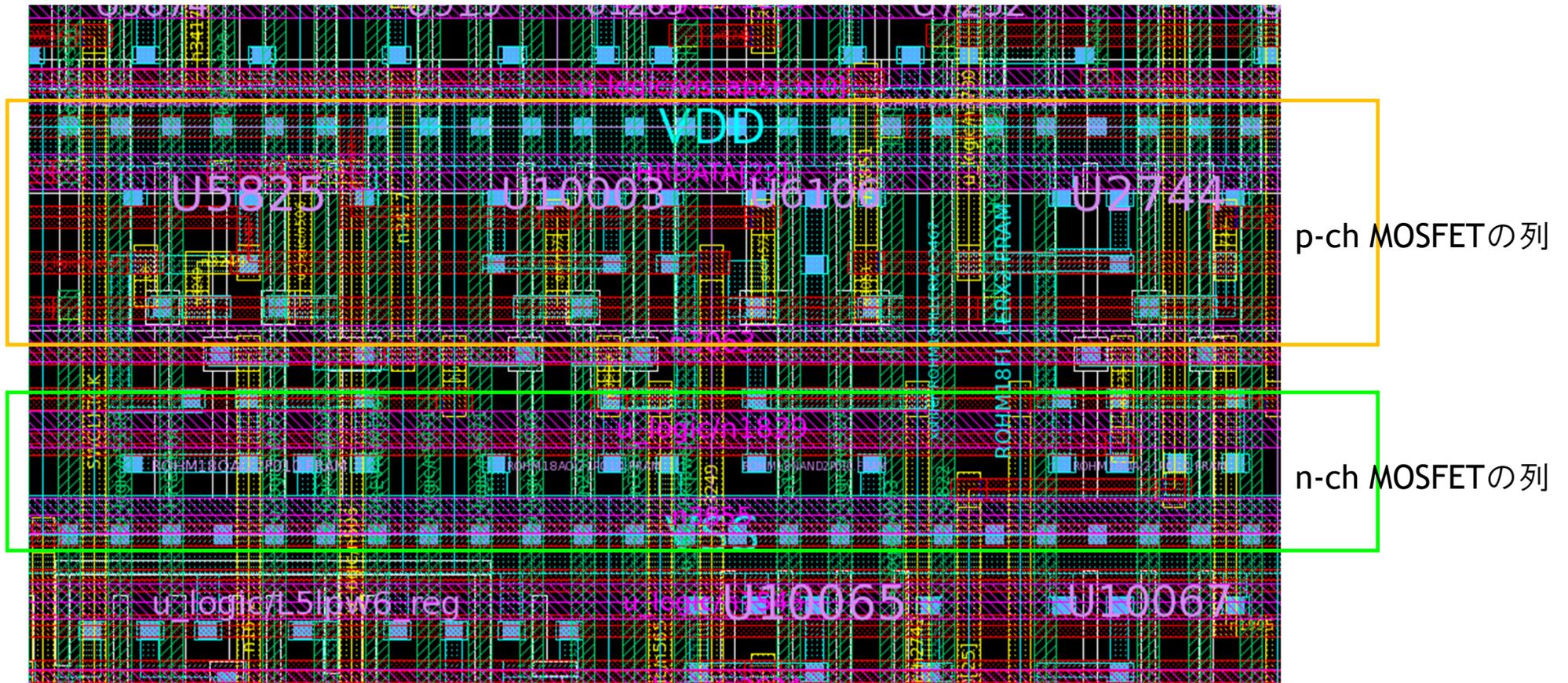
自動配置配線 (0.538mm×0.998mm)

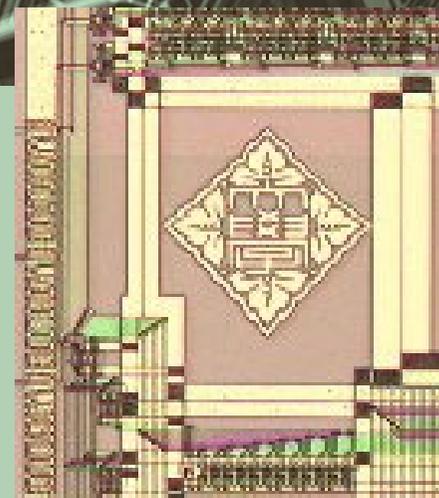
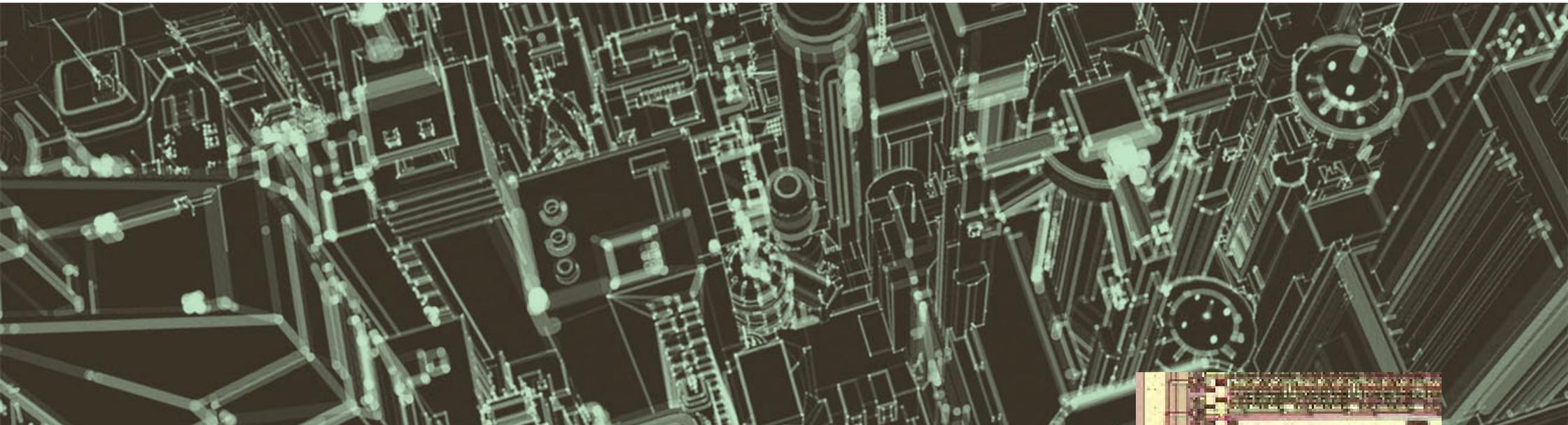


自動配置配線 (拡大)



論理ゲート (さらに拡大)





3-4 金沢大学のLSI設計教育

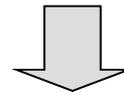
VDECの活動



VDEC

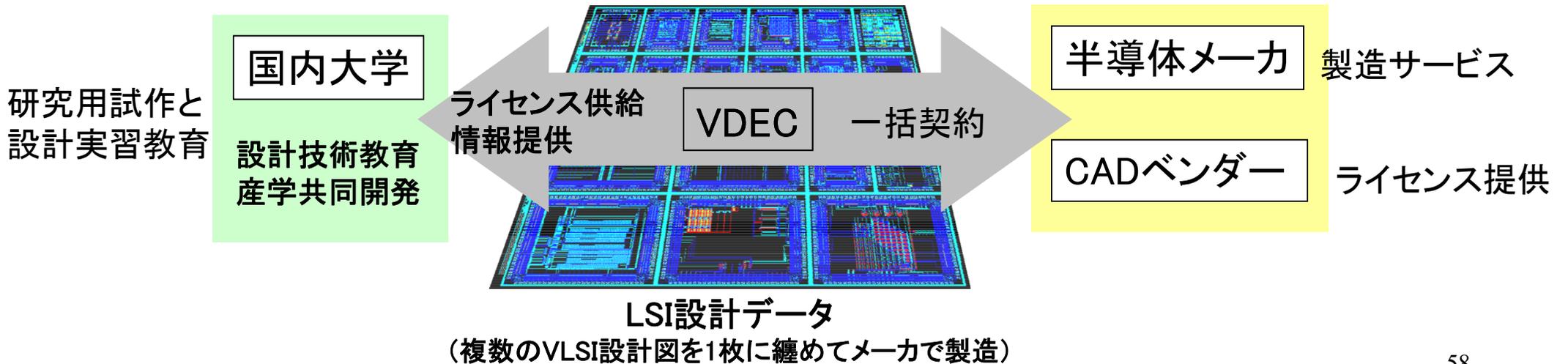
1990年代後半＝半導体製造サービスの普及とLSIのファブレス開発(工場を持たない生産)が可能に！

大学やベンチャー企業がLSI開発に参入

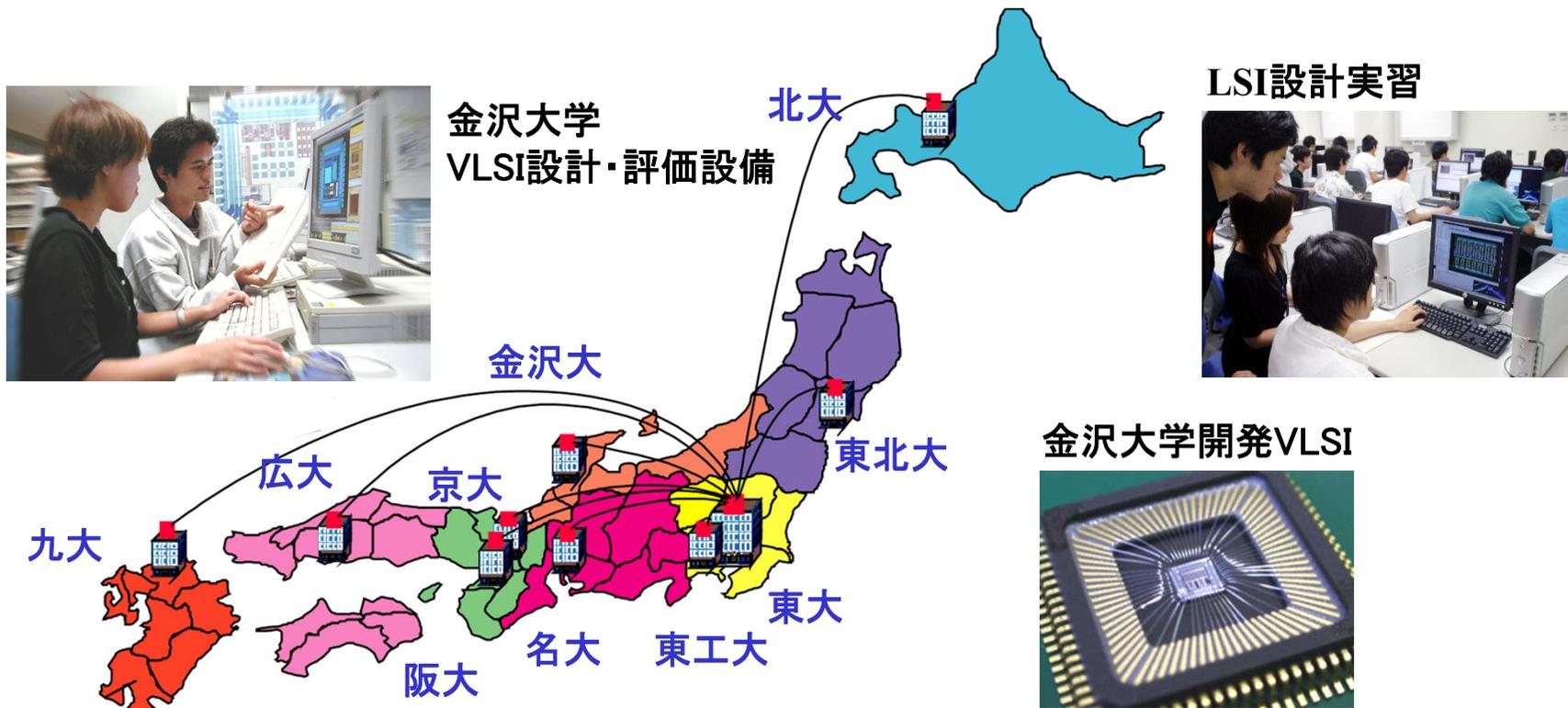


※ VLSI Design and Education Center

1996年：東京大学大規模集積システム設計教育研究センター(VDEC:通称ヴィデック※)設立(10大学ネットワークで構成)



- VDEC (現d.lab) は、全国ネットワークで組織されています。



金沢大学VDECは、北陸地域拠点として設立され、日本のVLSI 設計教育・研究の中心的役割を果たしています。

今週のまとめ

- ❖ コンピュータの殆どは、自動車、ロボット、ICカード、電子機器などの中で「組み込みシステム」として使用されている
- ❖ コンピュータは、5つの構成要素（制御回路、データパス、メモリ、入力、出力）から成る
- ❖ マイクロプロセッサの構造（アーキテクチャ）は、データパス、命令セット、メモリの構成、IOの構成により決定される
- ❖ 現在のマイクロプロセッサは、ノイマン型コンピュータの動作原理に基づいて構成されているが、他にも量子コンピュータやニューラルネットワークなどの動作原理の異なるコンピュータが実用化されている
- ❖ 回路システムは、アナログ回路とデジタル回路により構成され、それぞれ設計手順やCADツールが異なる
- ❖ マイクロプロセッサなどの集積回路のIPコア（設計データ）が流通しているので、これらを活用すれば誰でも新しいLSIが設計できる
- ❖ 金沢大学では、VDECが提供する最先端のCADソフトウェアを用いて誰でもLSI開発が可能

宿題

- ▶ 各自のパソコンにスマホアプリの開発環境(Android Studio)をインストールしておくこと
 - ▶ 既に、開発環境を持っている人は必要ない
 - ▶ インストール方法の説明サイト
 - ▶ <http://jaco.ec.t.kanazawa-u.ac.jp/edu/>
 - ▶ 「電子情報通信工学序論」をクリック
 - ▶ 「Androidアプリの開発環境インストール」をクリック
- ▶ 課題
 - ▶ WebClassの「提出課題 その3 (4/19の課題)」の掲示板に、今回の講義に関する質問または感想を記入しなさい。

