

## 13. プログラマブルデバイス

FPGAがあれば何でもできる  
テキスト9章

FPGA(Field Programmable Gate Array)は夢のデバイス  
ほとんど全てのことが実現でき、簡単に設計ができ、しかも  
安い(モノによっては高いけど、、)

プログラマブルデバイスのうちFPGA(Field Programmable Gate Array)は、最近のデジタルデバイスの中でもっとも激しく成長を遂げました。簡単に設計ができ、非常に大きなデジタル回路でも搭載して動かすことができます。モノによっては安価で使いやすくまさに夢のデバイスとっていいです。

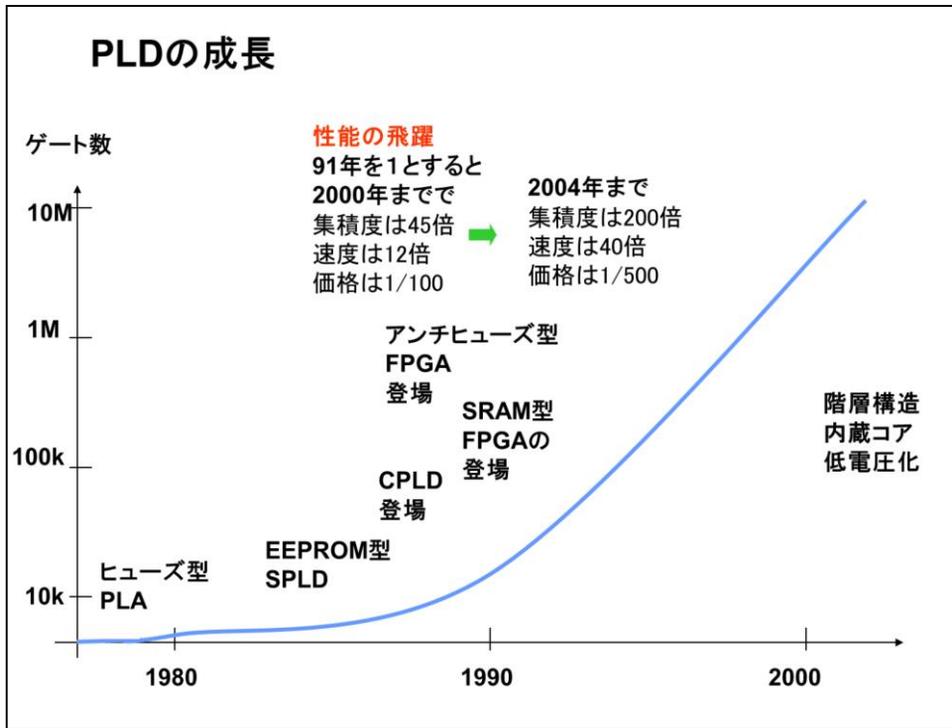
# PLD (Programmable Logic Device) とは？

- ユーザが論理機能を決めることのできるIC  
    ⇔ メモリ、CPU、ASIC (Application Specific IC)
- SPLD (Simple PLD) /  
  PLA (Programmable Logic Array )  
    – 小規模なAND-OR構造のもの
- CPLD (Complex PLD)  
    – 主としてAND-OR構造を拡張して大規模化したもの
- FPGA (Field Programmable Gate Array)  
    – LUT構造を用いた大規模なもの



用語は混乱していて、使い分けは統一されていないので注意！

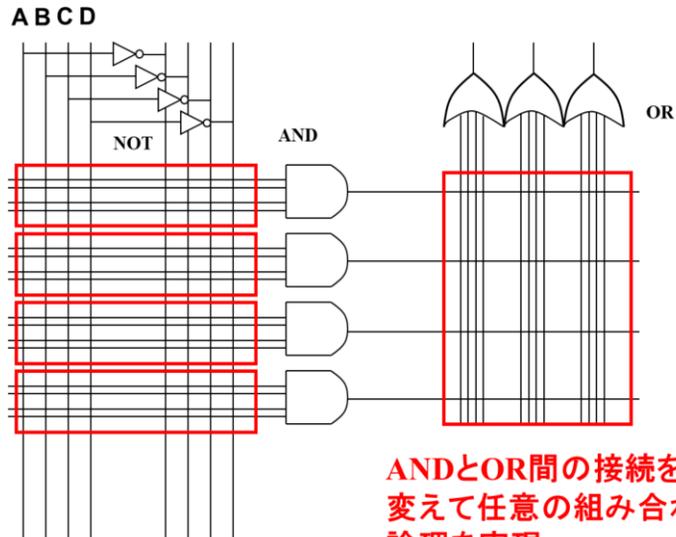
PLD (Programmable Logic Device)とは、ユーザが論理機能を決めることのできるICのことです。メモリやCPU、ASIC、昔の74シリーズのような標準デジタルICはその機能が決まっています、これらはプログラマブルデバイスとは言いません。CPUはソフトウェアで動作を変えられるので究極のプログラマブルデバイスだ、という人も居ますが、一般的には専用目的ICに分類されます。PLDには小規模なAND-OR構造でできたSPLD (Simple PLD)とこれを拡張したCPLD (Complex PLD)、LUT (LookUpTable)を用いた大規模なFPGA (Field Programmable Gate Array)に分類されます。SPLDはPLAとかPAL (これは製品名)とも呼ばれますし、用語は統一されておらず混乱しています。ただし、最近は大規模なものはほとんどFPGAになっていて、これだけ覚えておけば、まず問題ないかもしれません。



PLDの歴史は案外古く、70年代に用いられたバイナリジャンクショントランジスタを用いたヒューズ型のSPLDに遡ります。この型のSPLDは、単純なAND-OR構造の結線を外部からヒューズを切断することによってプログラムしました。小規模で再プログラムができなかったが、高速で、当時のデジタルICの主流であったTTL (Transistor Transistor Logic) の74シリーズで実現が困難な特殊な論理回路に用いられました。80年代になってCMOSを用いた再プログラム可能な素子が登場しました。Lattice社のGALシリーズがこの代表で、AND-OR構造にフリップフロップを含んだ出力ブロックを接続することにより、やや複雑な組み合わせ回路、順序回路が実現可能でした。このシリーズは再プログラミング可能であり、簡単なハードウェア記述言語からプログラミングを行う環境も整い、広く利用されるようになりました。

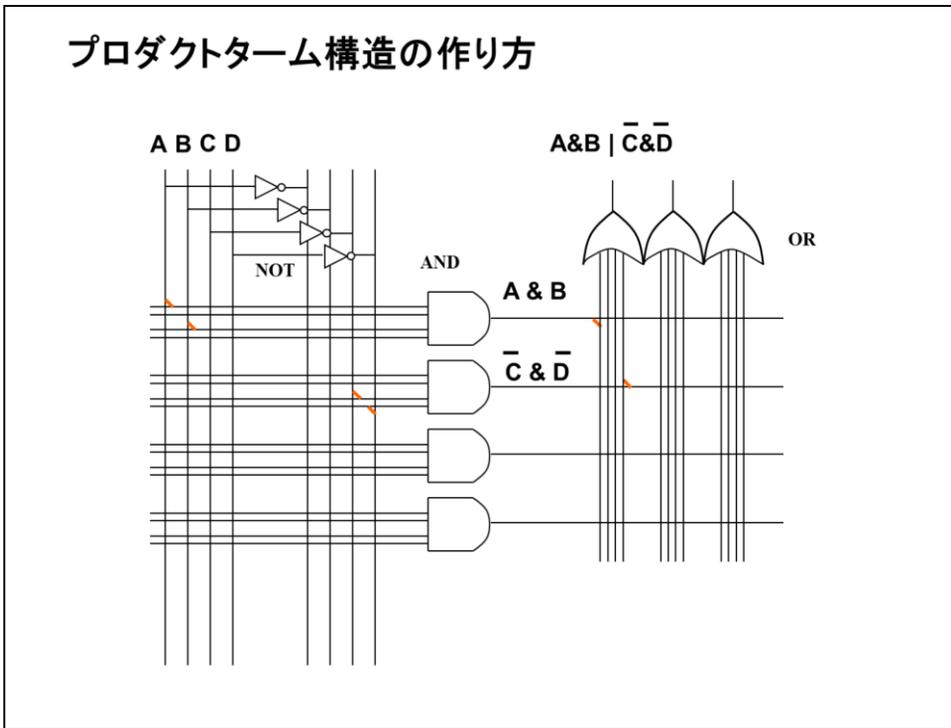
80年代の終わりに大規模なCPLD, FPGAが登場し、簡単なデジタルシステム全体がPLD上に実装可能となり、PLDは急成長時代に突入しました。アンチヒューズ型、EEPROM型、SRAM型等さまざまな特徴を持った方式が普及し、2004年現在に至るまで、集積度、速度は凄まじい勢いで進歩し、価格は急速に低下した。1991年を1とすると、2000年までの9年間で集積度は45倍、速度は12倍、価格は1/100となっている。さらに2004年には集積度は200倍、速度は40倍、価格は1/500となっています。最近では、内部構造階層化が進むと共に、メモリ、CPU、DSP、演算器、高速インタフェースを内蔵し、基板に代わってPLD上にシステムを実装するSoPD (System on Programmable Device)の考え方が登場するに至りました。また、低電圧化、低消費電力化したデバイスも登場しています。この発展速度は集積度においてメモリ素子を上回っており、もっとも急速に発展し続けるデバイスと言えます。

## SPLD (Simple PLD:プロダクトターム構造 /AND-OR構造)



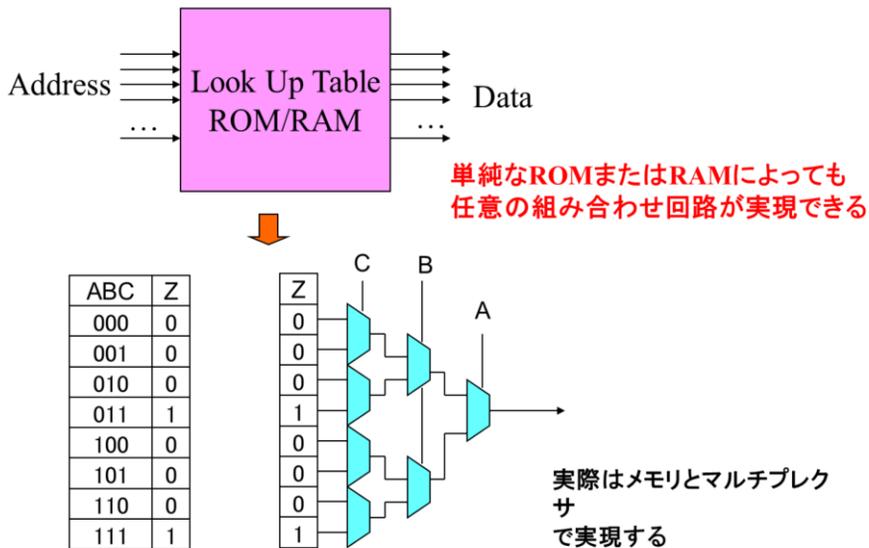
皆さんはブール代数を習ったときに、全ての論理式は加法標準形、すなわち、NOT-AND-ORの形で実現できることがわかったと思います。AND入力の選択(どの入力を繋ぐか、NOTかそのままか)と、AND-ORの結合を切ったり繋いだりすれば、任意の論理積項の組み合わせが作れ、任意の論理式を作ることができます。これがAND-ORを使ったプロダクトターム方式のSPLDの原理です。

## プロダクトターム構造の作り方



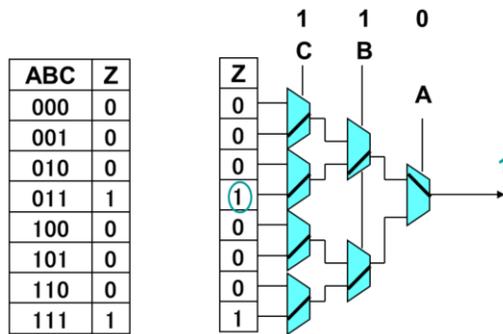
この例では $A \& B$ と $\bar{C} \& \bar{D}$ の論理積を実現する結線を示します。このように一つ一つの積項をANDゲートで作成し、これらをORゲートに入力します。

## LUT: Look Up Table方式による論理の実現



単純なメモリは、アドレスを入力、データを出力として考えると、真理値表の代わりに使えるので、任意の組み合わせ回路を実現することができます。しかし、通常のメモリは小さい面積で大きな容量を実現するのに特化した構造を持っているので、小規模の入出力で高速性が要求される論理回路に使う場合不利が大きいです。このため、実際には記憶要素にマルチプレクサのツリーを組み合わせることで表を実現します。これをLook Up Table (LUT)と呼びます。

## LUT: Look Up Tableによる論理の実現例



LUTによる論理の実現の例を示します。同じ列のマルチプレクサには同じ制御入力を繋ぎ、1ならば下から、0ならば上からの入力を出力に流します。**ABC**の順に**011**を入れると、上から**3**番目のデータが取り出せることが分かります。(表と右の図で**ABC**が逆順な点にご注意ください。)すなわちこの回路は表として働いています。マルチプレクサは以前紹介した方法で簡単に作れるので、この方法は入力が**6,7**よりも小さい場合には効率的です。

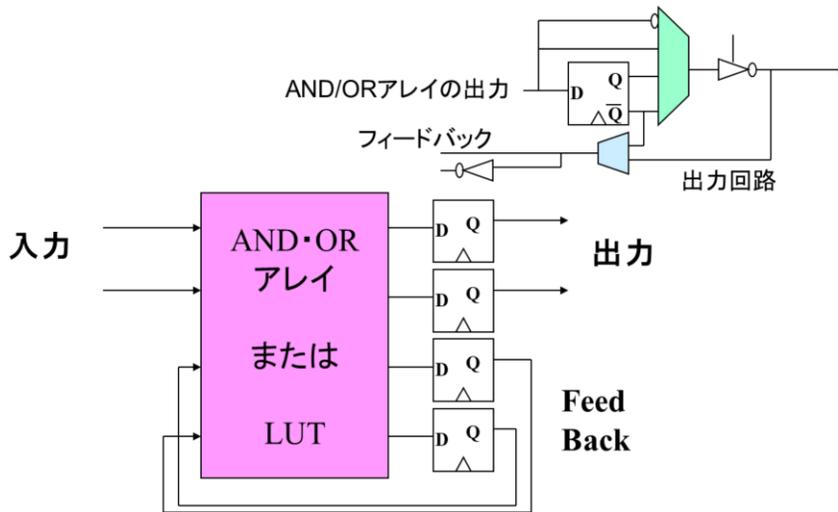
## プロダクトターム方式 vs. LUT

- プロダクトターム方式 (AND-OR構造)
  - 多入力多出力回路が効率良く実現できる
  - 場合によっては入力項数が不足する
  - EEPROM、フラッシュROMでの実現に適している
- LUT
  - 任意の論理が実現できる
  - 出力が少なく小規模な論理に有利
  - フラッシュ、アンチヒューズ、SRAM型に適している

プロダクトターム方式は、AND出力から出てくる積項を複数のORゲートで共有することができます。すなわち、場合によっては多入力、多出力回路が効率的に実現できます。しかし、ANDゲートの数、ORゲートの入力数にかなり余裕がないと、本当に任意の論理式は実現できません。先にしめした例では、ANDゲートは4つしかないので積項が4つを越えると実現できなくなってしまいます。プロダクトターム方式は結線の交点上のスイッチで実現するので、EEPROM、フラッシュROMなどによる実現に適しています。

一方で、LUTは真理値表なので本当に任意の論理式が実現できます。しかし、2の入力数乗のオーダーで表のサイズが大きくなるので、入力数は通常4-6程度に限定されます。途中結果の共有はできません。LUT方式は様々な方式で実現できますが、最近ではSRAM型が良く使われます。

## 順序回路の実現



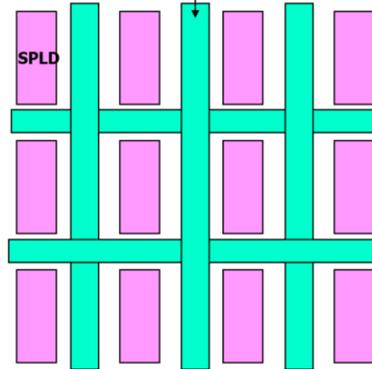
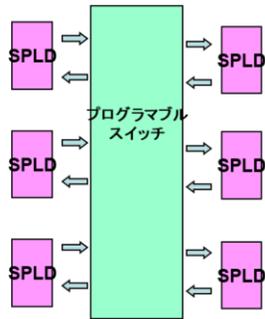
**出力にF.F.を付けて、フィードバックラインを  
装備すれば任意の順序回路が実現できる**

組み合わせ回路の出力にD-F.F.を付け、さらにフィードバックを付けることで順序回路が実現できます。このためには、F.F.の出力回路を選択可能にしてやる必要があります。図はLattice社のGALの出力回路の例です。出力はF.F.を介したものと介さないものが選択可能で、3ステート出力になっています。また、組み合わせ回路にフィードバックすることもできるようになっています。このような構成のPLDをSimple PLD (SPLD)と呼びます。

# CPLD (Complex PLD)

AND/OR ロジックブロック複数をスイッチで接続

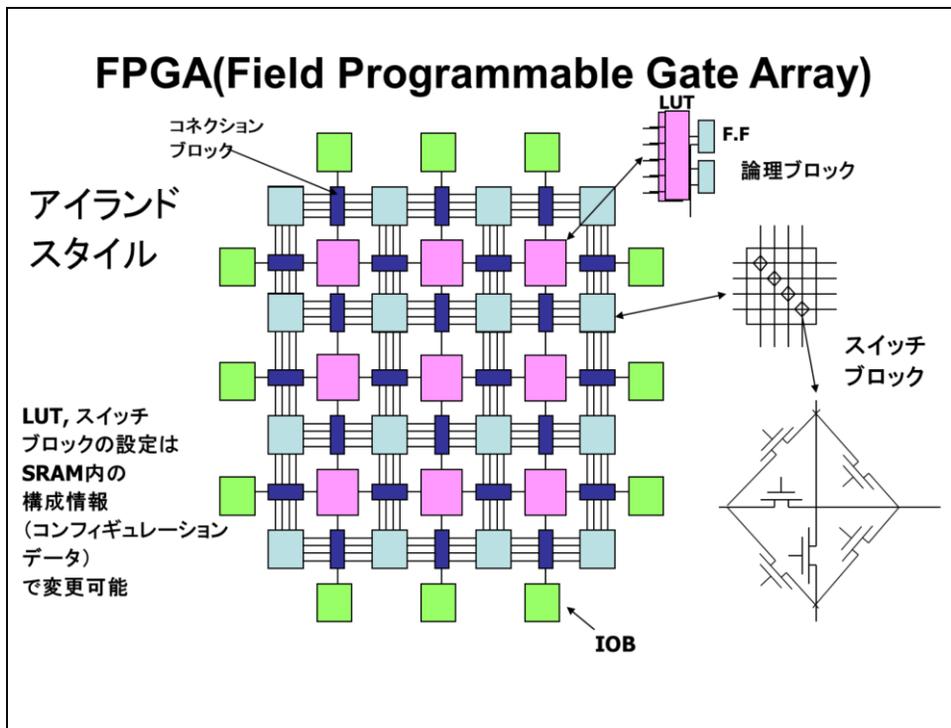
プログラマブル  
スイッチ



Altera社  
MAXシリーズなど

2次元構造で大規模化

大規模なデジタル回路は、複数の組み合わせ回路、順序回路から出来ています。これに対応するため、先に示したSPLDを複数個、スイッチで接続します。さらに大規模なシステムを作るためには2次元構造のスイッチを利用します。このようなチップをComplex PLD (CPLD)と呼びます。



FPGAは今までのAND-OR構成と違って4-6入力程度のLUT2セット程度を一つの論理要素として使います。それぞれの出力にはD-FFを備えます。この論理要素の周辺に配線を敷き詰め、交点にスイッチブロックを置きます。スイッチブロックはトランジスタのON/OFFで接続をON/OFFし、論理要素間の配線を自由に行えるようにします。また、論理要素と、配線の間にも同様のコネクションブロックを置き、論理要素との入出力配線を制御します。また、チップの入出力PINとの間の配線も行います。LUTの内容、スイッチブロックのトランジスタのON/OFFを設定することにより、様々な回路構成を実現することができます。この設定情報のことを構成情報(Configuration Data)と呼び、これをどのように蓄えるかによりFPGAの性質が決まります。このような構成を、配線の海の中に論理要素の島があるイメージからアイランドスタイルと呼ぶ場合があります。

## 柔軟性の実現

- アンチヒューズ型
  - 高圧により絶縁体を破壊し、導通させる
  - 高速だがOne-time
  - Axcelerator (Actel), pASIC (QuickLogic)
- EEPROM, フラッシュROM型
  - Floating Gateによりゲートのスイッチを実現
  - 書き換え可能
  - GAL (Lattice) MAX, MAX II (Altera)
- SRAM型
  - SRAM上のデータにより論理機能、配線を実現
  - ISP(In System Programming)が可能だがスイッチを切ると配線情報が消失する
  - LUT型FPGAに向き、最近急速に発達
  - 初期のシリーズ: XC4000(Xilinx), FLEX10K(Altera), ORCA(Lucent)
  - 現在のシリーズ: Virtex, Spartan(Xilinx), Stratix, Cyclone(Altera)
- その他
  - 磁気メモリ
  - DRAM

SPLD, CPLD, FPGAは、基本論理ブロックの実現方式や配線方式によつての分類ですが、PLDにとつて重要なのは、プロダクトターム方式の配線構造、LUTの内部論理、スイッチの設定などをいかに変えるかという柔軟性の導入方式であり、これによつてPLDの特徴が決まります。

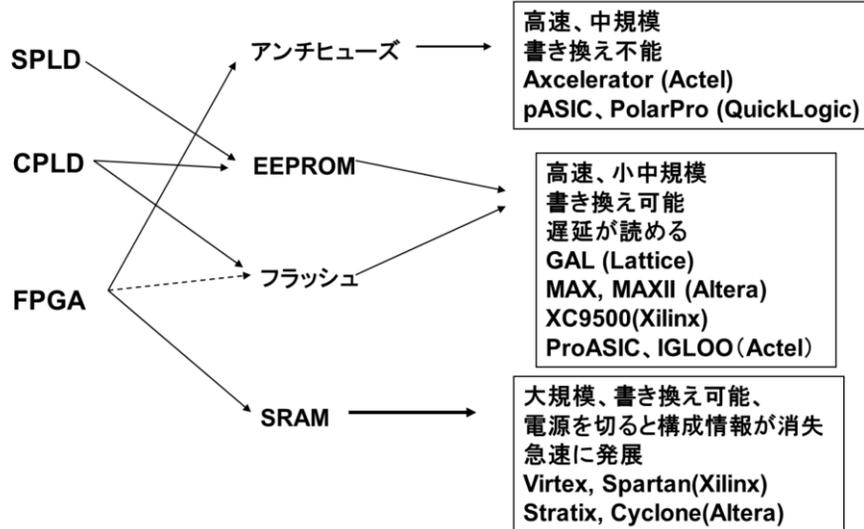
まず、70年代から伝統的に用いられているヒューズ型があります。ヒューズ型は大電流で金属を溶解させることにより、配線を切断します。一方、80年代以降、ヒューズ型に代わつて用いられるようになったアンチヒューズ型は大電圧で絶縁体を破壊することで、配線間を接続します。両者共に金属で接続が行われるため、遅延が小さい利点がある一方、一度プログラムしたら消去することができないワンタイム型です。高速動作に小規模なものから大規模なものまで広く用いられる。Actel社、QuickLogic社の製品が一般的です。

次にEEPROM同様に、特殊なゲートであるフローティングゲートをON/OFFのスイッチとして用いるEEPROM方式があります。高圧によりゲート内に電子を注入することで、電源を切ってもデータを保持することができます。書き換え可能であるが、半導体でスイッチが形成されるため、アンチヒューズ型に比べて動作速度の点では不利です。AND-OR型の基本論理ブロックに適しており、Lattice社の一連の製品、Altera社のCPLDが有名ですが、最近ではフラッシュROMを用いる大規模なものが登場しています。

最近、発展がもっとも著しいのがSRAM型である。この方式は、通常のスタティックRAMをそのままLUTに用いると共にスイッチの接続情報を保持します。この接続情報により主としてCMOSのトランスファゲートを用いて構成されたスイッチのON/OFFを制御します。このように、特にSRAM型では、チップ上の論理構成を決定するのはメモリ内の構成情報データです。この方式は、CMOSの一般的なプロセスで作れるため、大規模なチップに適しています。しかし、電源を落とすと構成情報が消えてしまいます。すなわち、通電の度に構成情報データをチップ内に送り込む必要があります。逆に、動作中に構成情報データを入れ替えることにより、機能を変更することが可能です。

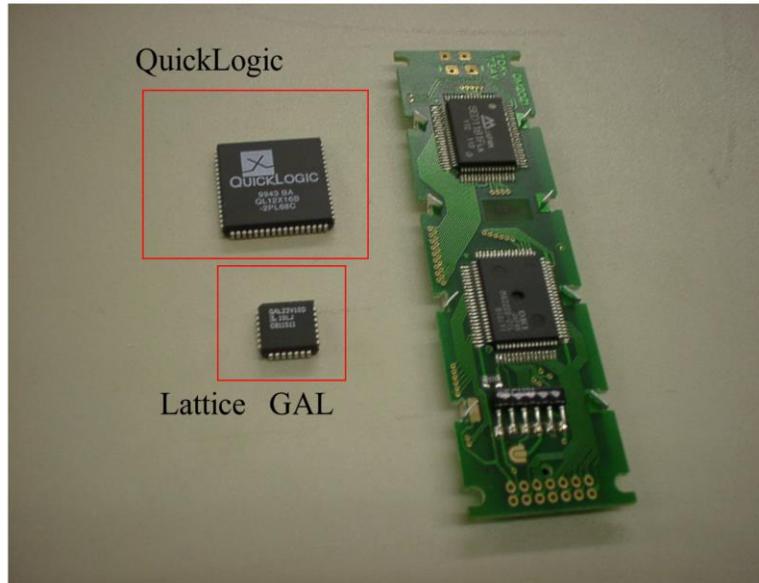
その他、MRAM, FeRAMなど新しい記憶方式を用いるもの、DRAMを用いるものも試作されているが、本格的な利用には至っていない。

## 構成方式と柔軟性実現技術



PLDをどのように構成するか、と、柔軟性をどのように実現するか、は互いに関連しています。プロダクトターム構成は、EEPROM型に向いており、FPGAは、SRAMやアンチヒューズ型に向いていますが、それ以外の型でも利用可能です。

# PLDの概観



古典的なPLDの概観を示します。Lattice社のGALは、EEPROM型のSPLDで80年代より良く使われています。QuickLogic社のFPGAは、アンチヒューズ型です。



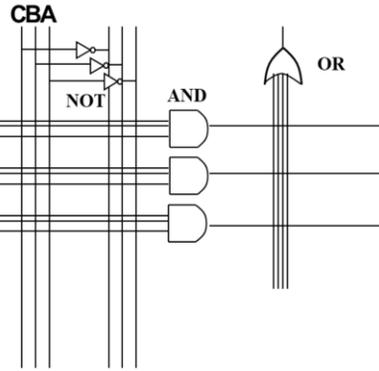
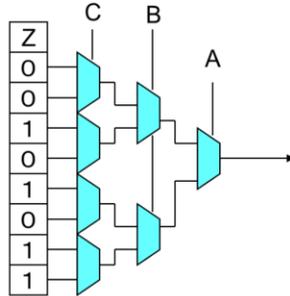
これはより大規模なXilinxのSRAM型とQuicklogic社のアンチヒューズ型です。  
BGA(Ball Grid Array)と言って、チップの裏に半田ボールが並んでいるパッケージ  
で基板の表面にくっつけます。

# 演習1

左のLUTと同等の機能を右のプロダクトタームの接続により実現せよ。

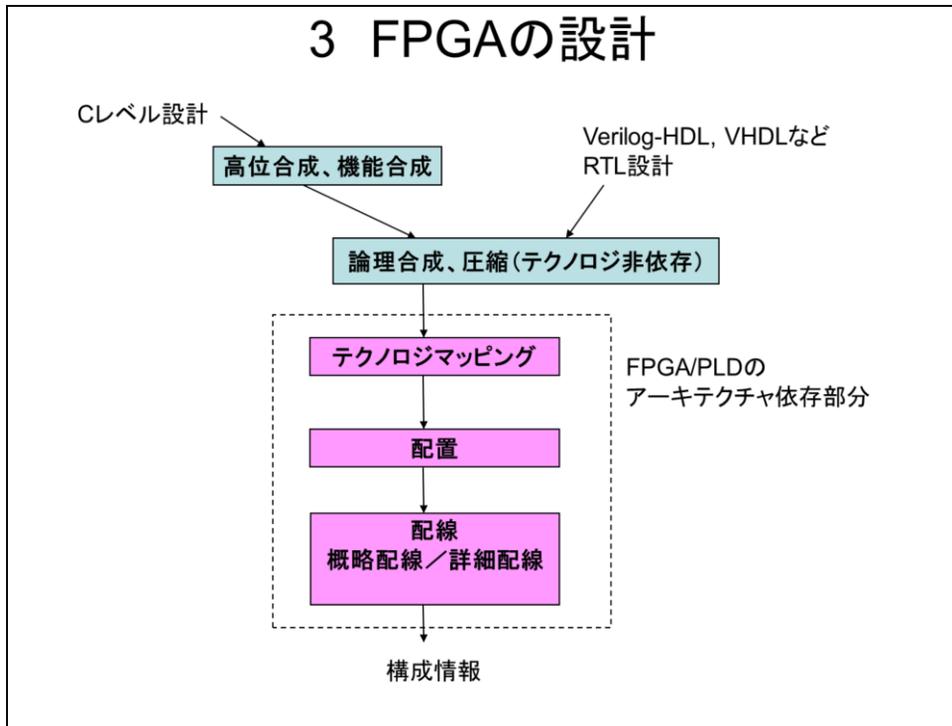
(ヒント)

LUTの論理を表わす真理値表の入力は、ABC順(Aが一番上の桁、Cが一番下の桁)であることに注意すること。



プロダクトタームの線が細かくてすいません。若干の簡単化が必要です。

### 3 FPGAの設計



FPGAの設計は、3年生の実験で学びます。設計は、Verilog-HDLやVHDLなどのハードウェア記述言語で行うことが多いのですが、最近はCレベルでの設計も多く使われるようになりました。これらの記述は論理合成、圧縮の手順を経て、対象のFPGAの種類に応じてゲートなどの割り当て(テクノロジーマッピング)が行われ、配置、配線の結果、構成情報が出力されます。これをFPGAに流し込めば、設計通りの動作を行わせることができます。

## FPGAの設計ツール

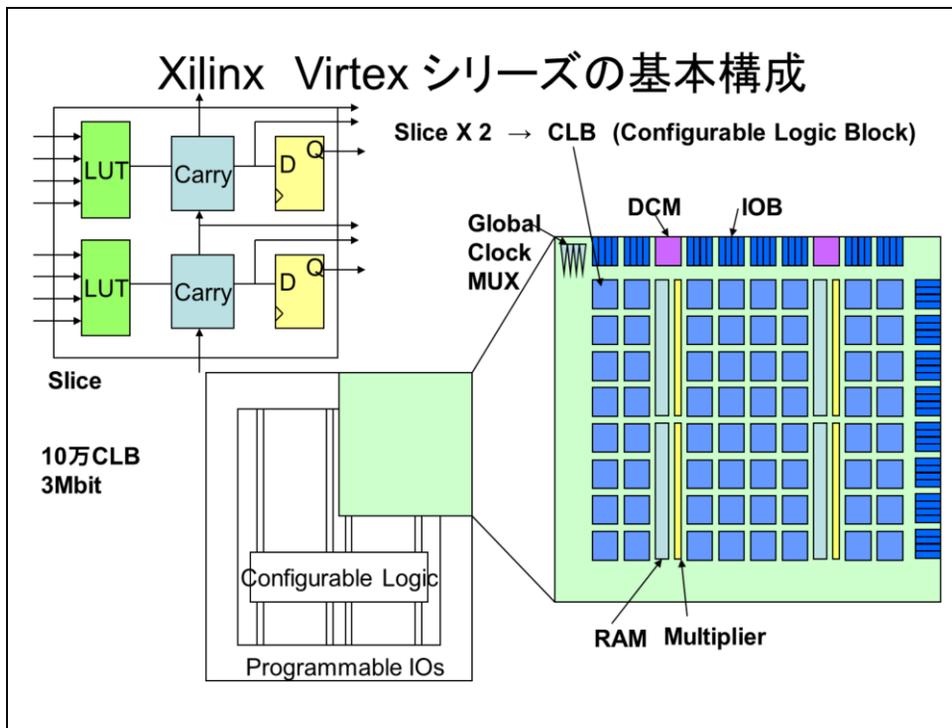
- ベンダが統合型のツールを提供
  - シミュレーション、配置、配線、構成情報の転送、デバッグツールが一式入っている
  - お試し版Web packでもかなりのことができる
- 設計入力
  - Verilog HDL、VHDLなどハードウェア記述言語が多い
  - 最近、C言語設計が発達
    - Impulse Cなどサードパーティの供給
    - Xilinx Vivado HLS
    - Intel(Altera) OpenCL

Intel (Altera) 社のQuartusなどがこの一例です。お試し版のWeb Packは無料でダウンロード可能で、かなり大きいチップまで設計ができます(実は実験ではこの無料版を使っているが、大学では大きいチップが設計できるアカデミックライセンスもある)。皆さんのPCにダウンロードして使うこともできます。

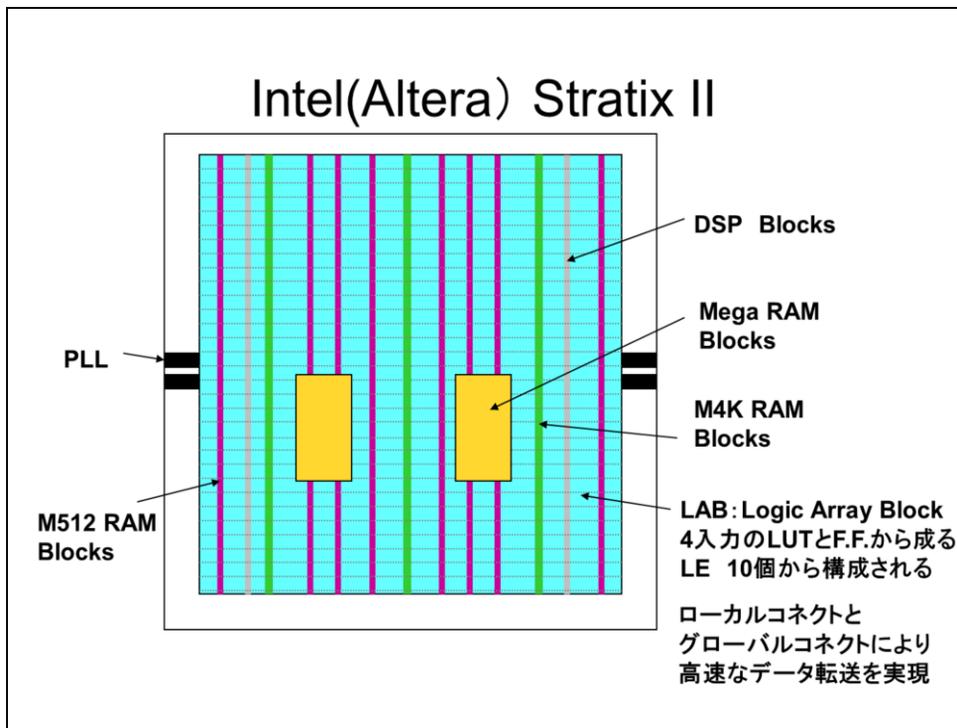
## 2. 最近のFPGA

- 最近のFPGAの情勢
  - 階層的構造による大規模化、高速化:
    - Xilinx社Virtexシリーズ、Altera社Stratixシリーズ
  - System on Programmable Device
    - DLL、CPU、DSP、メモリ、乗算器、高速リンクをハードIPとして混載
    - Xilinx社Virtex-4FX,Virtex-5FXT (Power PC)
    - PCI Express,DDR-SDRAM,Ethernetに対応
  - 量産品を目指して特化
    - 安価: Xilinx社Spartan、Altera社Cyclone
  - 低電圧化、低消費電力化
    - 不揮発性FPGAは低電力用にシフト: Actel社IGLOO、ProASIC3、SiliconBlue社iCE
    - ハイエンド・ローエンド共に低電力化が進む: Altera社Arriaシリーズ
  - 部分再構成機能の充実

最近のFPGAは、ネットワークルータ、ハブなどに用いられる大規模、高速なハイエンドの製品と組み込み用の安価なローエンド製品に分化が進んでいます。両方共に、メモリ、クロックコントローラ、乗算器、高速リンクをハードIPとして搭載する製品が増えていて、システムをまるごとプログラマブルデバイスで実現するSoPD(System on Programmable Device)が利用されています。さらに低電圧、低電力に特化した製品も登場し、一部のみを書き換える部分再構成機能が充実してきています。



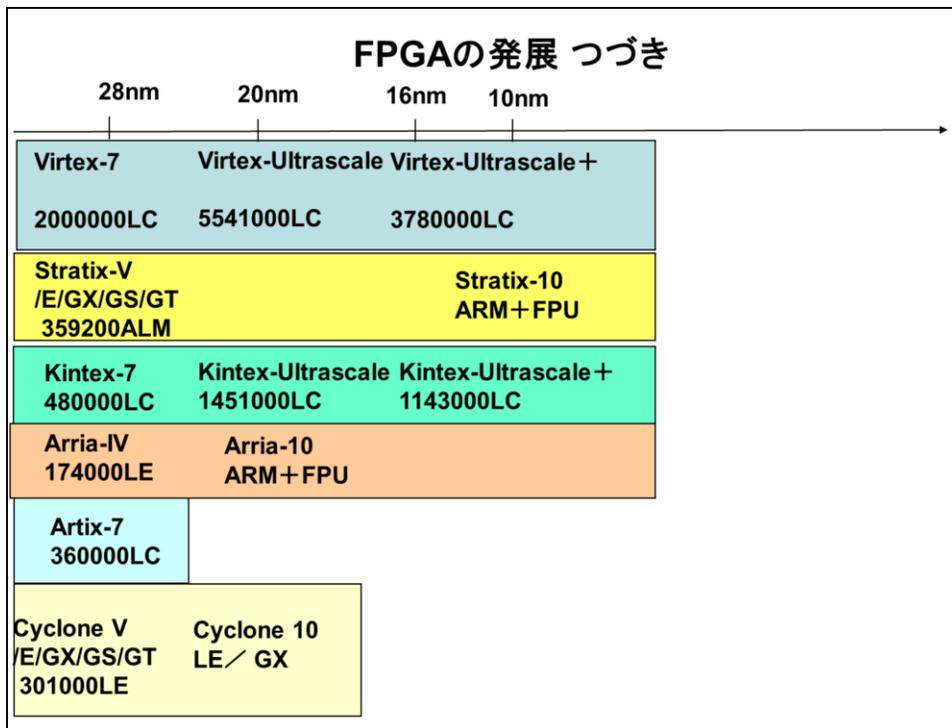
この図はXilinx社のVirtexシリーズの基本構成です。論理要素の配列の間にRAM, 乗算器などのIPが配置されており、DCM(クロックのコントローラ)、クロックバッファが周辺に配置されています。



これはIntel(昔のAltera)のStratixシリーズです。これも一定の間隔でRAMやDSPブロック(積和演算を行うハードウェアのことをこのように呼んでいます)が装備されています。演算器は論理要素を使って作ることもできるのですが、専用のハードウェアを使った方がはるかに速いです。

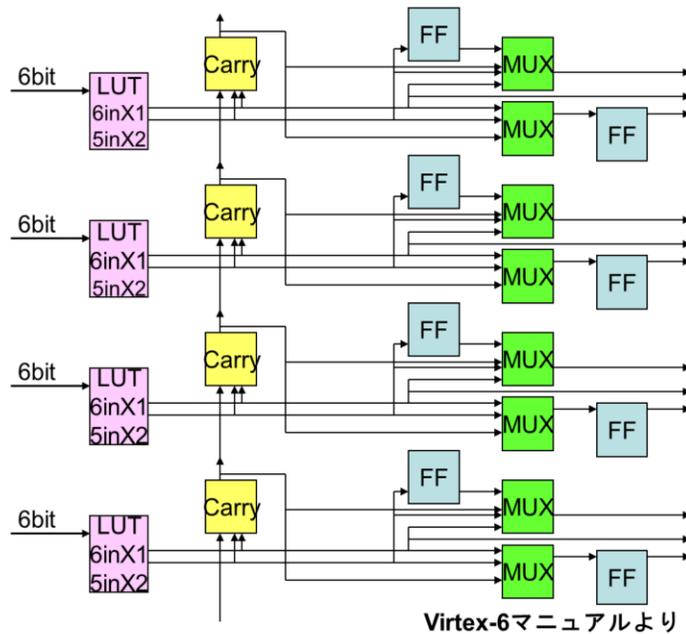
プロセステクノロジーと製品			
90nm	65nm 60nm	45nm 40nm	28nm
High-end Virtex-4LX/FX/SX 200000LC	Virtex-5LX/LXT/SXT/ FXT/TXT 330000LC	Virtex-6LXT/SXT/ HXT/CXT 760000LC	Virtex-7 T/XT/HT 2000000LC
Stratix-II/GX 179400LE	Stratix-III/L/E テクノロジー1世代でX1.5-X2.5	Stratix-IV /E/GX/GT 531200LE	Stratix-V /E/GX/GS/GT 359200ALM
Middle range			Kintex-7 480000LC
Arria		Arria-II	Arria-IV 174000LE
Low-cost Spartan-3A N/DSP 53000LC	Spartan-6LX/LXT 150000LC		Artix-7 360000LC
Cyclone II 68416LE	Cyclone III/LS 119088LE	Cyclone IV/E/GX 149760LE	Cyclone V /E/GX/GS/GT 301000LE
High-endとLow-costではX3-X5			

FPGAはその構造の単純さを利用して新しい半導体プロセスをいち早く取り入れて来ました。テクノロジーが一代違えば集積度は1.5-2.5倍になっています。低コスト用のチップも、容量はハイエンドの1/3から1/5ですが、積極的に新しいテクノロジーを取り入れていることがわかります。

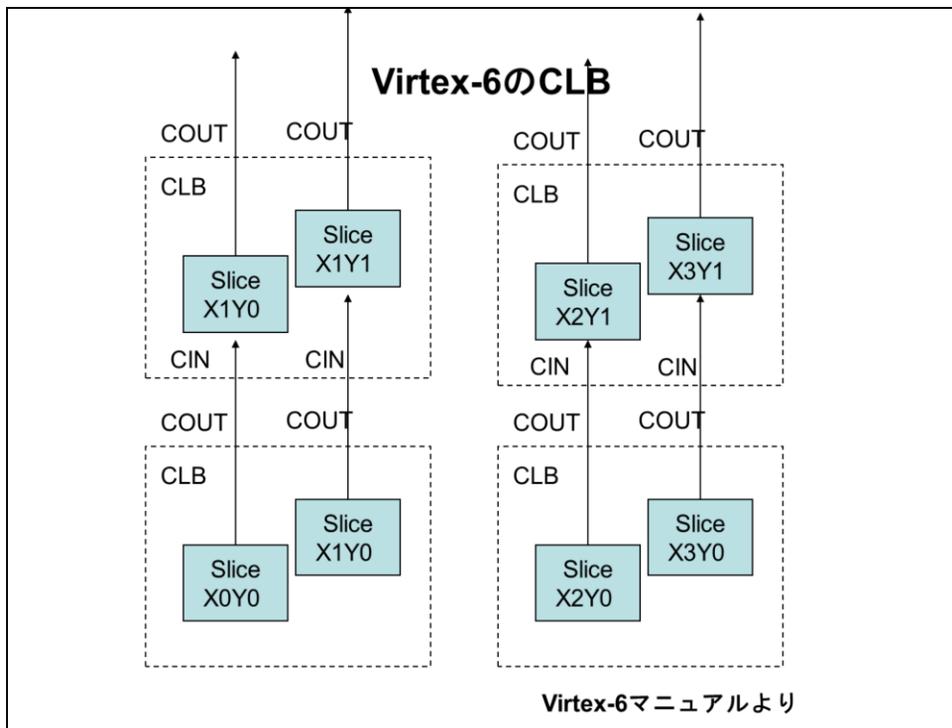


FPGAの発展はこれ以降も続いています、Low costの製品には最新のプロセスは使わないようになっています。

## Virtex-6のSlice構造

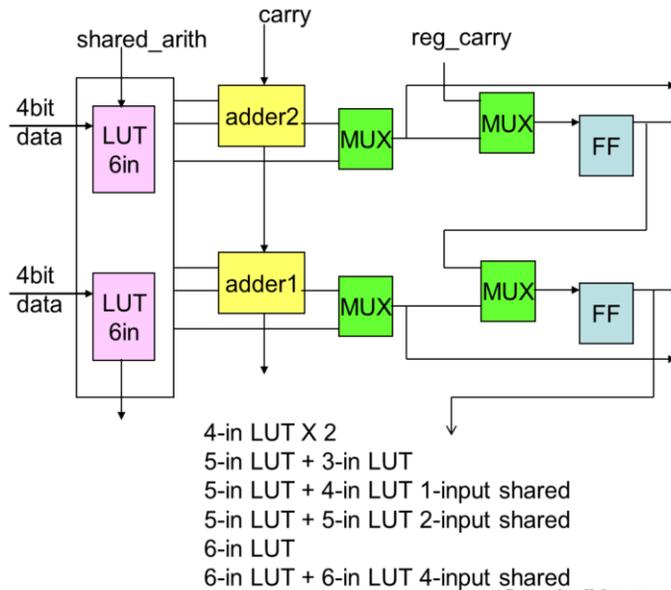


最近のFPGAの基本論理要素は、古典的なもの比べてLUTの入力が増えていま  
す。このLUTは6入力のものを1つとしても5入力のものを2つとしても使えるようになっ  
ています。論理要素間を直結するCarry(桁上げ)も用意されています。出力周辺の  
回路はマルチプレクサが入って柔軟性を増しています。

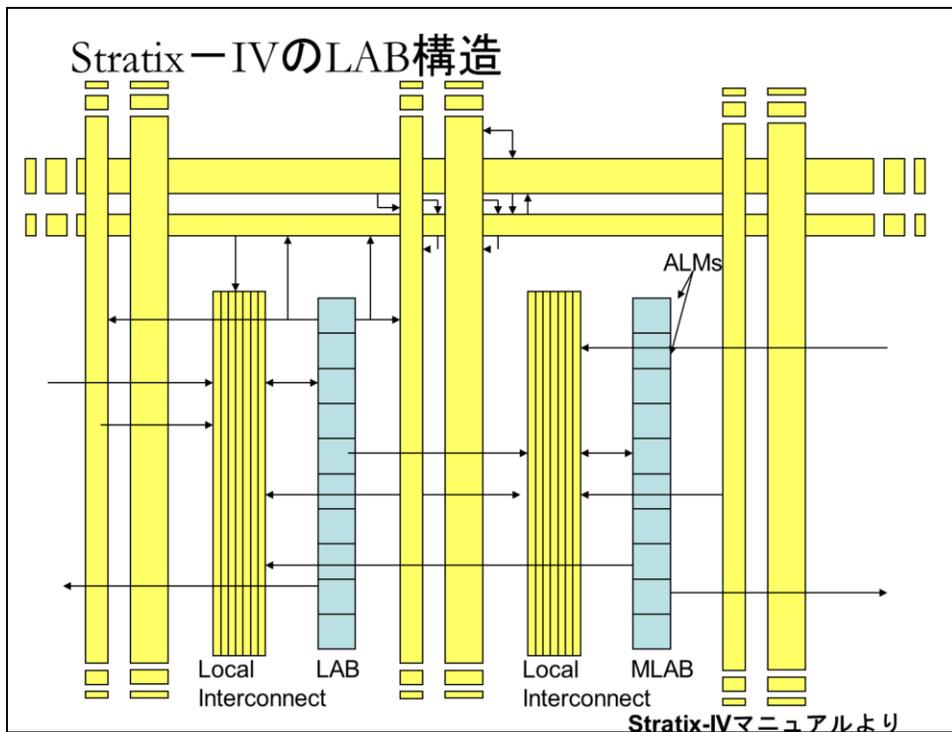


このスライス構造を二つで**CLB**と呼ぶ論理素子を構成します。スライス同士は直結線がなく、それぞれが隣の**CLB**のスライスと直結線を持っているのが特徴です。

# Stratix-IVのALM



XilinxのライバルのAltera社の基本論理構造です。Altera社は最近Intelに買収され、やや高性能製品に特化している傾向が見られます。Altera社のハイエンド製品Stratixの基本構成要素は、Xilinx同様6入力のLUTを使いますが、このLUTはさらに柔軟に様々な組み合わせを取ることができます。直結線、出力のマルチプレクサ構成など、両社の基本構成要素は似てきています。



Intel(Altera)のFPGAの特徴は、階層型の強力なネットワークを持つ点です。これにより、Intel社のStratixは非常に複雑な論理回路を搭載しても250MHzを越える周波数で動作します。

## 低コスト、量産志向製品

- Spartanシリーズ (Xilinx社)
  - Spartan-3A (90nm)
    - 5万Logic Cell以上の集積度、I/O数、内蔵RAM、乗算器、DCM等を持つ
    - 上位機種はDSPモジュールを持つ
  - Spartan-6 (45nm)
    - 14万Logic Cell以上の集積度、DSPモジュール装備
    - 上位機種はPCIExpressインタフェースを持つ
- Artix シリーズ (Xilinx社)
  - やや高価格、高性能
- Cycloneシリーズ (Altera社)
  - Cyclone-IV (60nm)
    - Eシリーズ: RAM、乗算器、PLLのみ
    - GXシリーズ: PCIExpressインタフェースを持つ
- 実際の組み込みシステムへの利用が進む

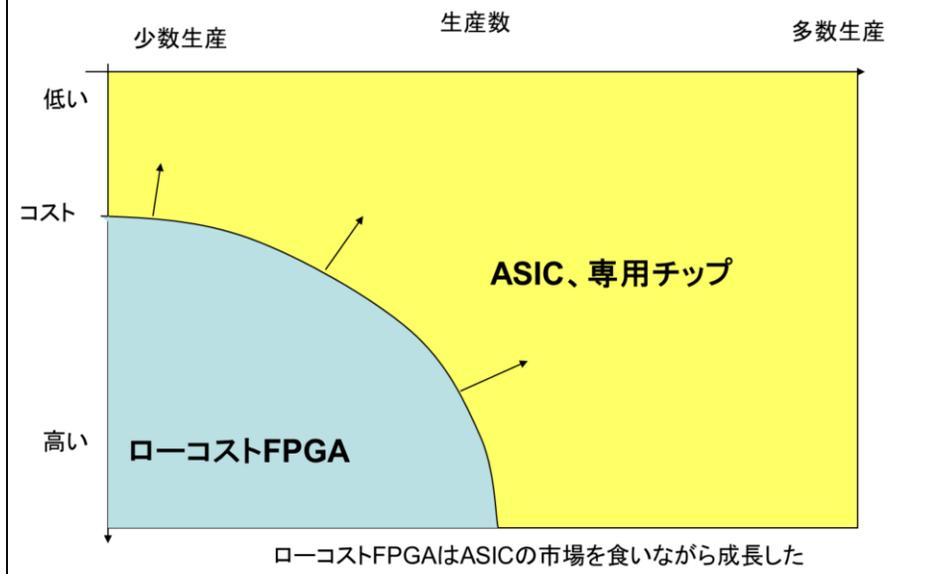
FPGAは実際の組み込み製品にも使われています。車には数十のFPGAが使われており、スマホや家電にもASIC間のつなぎのロジックとして使われています。これらは低コストで必要最低限のIPを持ちます。

# FPGA vs. ASIC

- ASIC(Application Specific IC) 特殊目的用IC
  - 製品に特化したICチップを特別に開発するもの
  - スマートフォン用、カメラ用、地デジ用、etc.
  - SoC(System on a Chip)、システムLSIと呼ばれ日本の半導体の主力だった
  - 最近のプロセスは開発コストが高騰し、非常に多くの個数を開発しない限り元が取れないことが多い
  - ビジネスとして難しい
  - 本来、FPGAよりも高速、低消費電力、低コスト
- FPGAの低価格品は、ASICの市場を食いつぶして成長
- FPGAは最新プロセスを使うため、小規模なチップでは性能的にはASICとあまり変わらない
- 消費電力、コスト(大量生産できれば)の点でASICが有利

FPGAとASICを比較すると、ASICの方が高速、低消費電力です。それなのになぜFPGAがASICの市場を取り込んで成長しているのでしょうか？それは最近のプロセスが最初の1個を作るまでのコスト(Non-Recurrent Costと呼びます)が高騰しているためです。これは、複雑なマスクパターンをいくつも使って作るため、マスク代自体と設計費用が膨大になってしまうためです。一度作ってしまうと機能の変えられないASICは、NRCに見合うだけの製造個数があらかじめ見込めないと作ることができなくなっています。一方で、FPGAは構造自体が簡単なため、いち早く最新プロセスで多くの同一製品を作り、NRCを回収することができます。しかし、本当に多数作る場合は今でもASICの方が有利です。ある製品を開発する場合、それに向けたASICを作るかどうか、FPGAでなんとかするか、経営上の決断が重要になってきます。

# ローコストFPGAの進撃

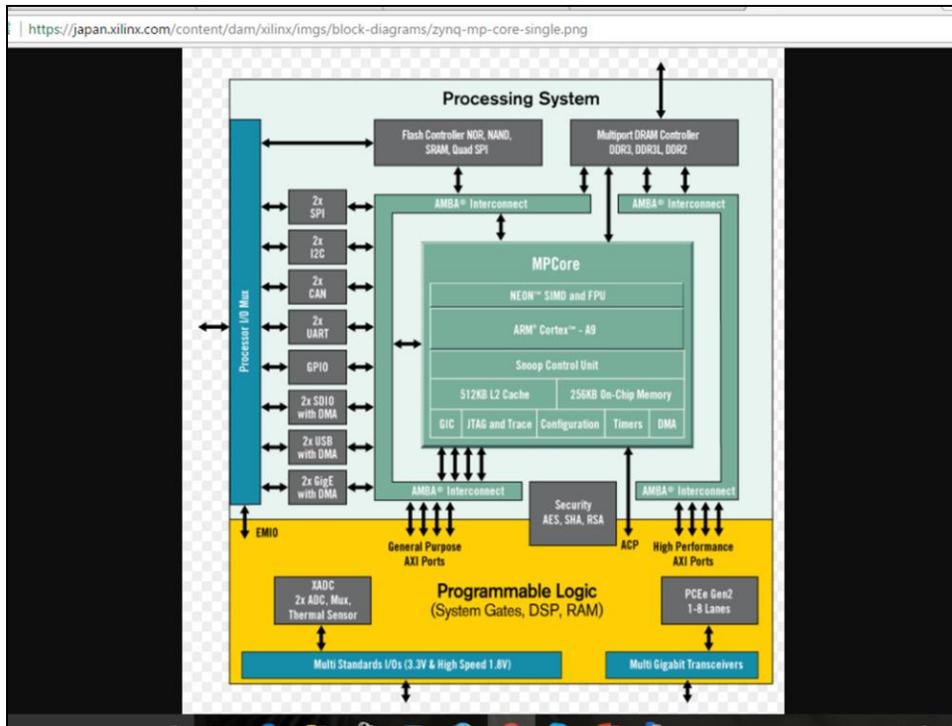


ローコストFPGAは、ASIC、専用チップの市場を食いつぶしながら成長しました。しかし、最近は一世代前のプロセスを用いたASICが、大量生産分野で踏みとどまっています。しかし、最先端プロセスはローコストFPGAには高価になりすぎたため、もうこれ以上新しいプロセスを使わなくなる可能性もあり、両者の棲み分け体制ができていくでしょう。

### 3. IP (Intellectual Property)と SoC FPGA

- 標準装備IP
  - 内蔵RAM
  - Multiplier, DSPモジュール(乗算器+ALU)
  - Clock Manager
  - 高速シリアルI/O(ハイエンド、ローエンドの一部)
- 最近のIP
  - PCI Express (Virtex-6, Stratix-IV)
  - EthernetのMACコントロール (Virtex-6)
  - DRAM用メモリコントロールブロック (Spartan-6)
  - ハードコアCPU → ARMの普及 (Arria10, Zynq)
  - 浮動小数演算器 Area 10, Stratix 10

FPGAは、標準ロジック要素だけでなく、様々なハードウェアをIP (Intellectual Property: 知的資産)として持っています。IPの中でレイアウトまで決まっているものをハードコアIP、ハードIPと呼びます。ハードIPの中にはほぼ標準装備となっているものと、チップ特有のものがあります。PCIeのインタフェースやDRAMコントローラなど大変便利です。最近ではFPGAを科学技術計算に利用するために浮動小数演算器を搭載するものも現れています。



ハードIPを持っているチップとして最近良く用いられるのが、Zynqです。ZynqはARMとFPGAの混載したチップで、SoC型FPGAと呼ばれます。このタイプはホストを必要とせず、独立して動作が可能です。Zynqは各種組み込みシステム、ロボット、ドローンの制御などに使われます。同じタイプのFPGAはIntel(Altera)のAria10, Stratix10などに使われています。

## ハードコアとソフトコア

- ハードコア
  - レイアウトレベルで組み込んである
  - 標準的な命令セットを持つ
  - 利用可能なFPGAはかなり高価となる
  - ARM (Xilinx Xynq, Intel(Altera) Aria-10, Stratix10)
- ソフトコア
  - FPGA上のロジックブロックで構成する
  - 一定のサイズがあればどの製品でも利用可能
  - MicroBlaze (Xilinx)
    - 32ビット長、32レジスタを持つ
    - 約500スライス、85MHz動作(Spartan-3)
  - PicoBlaze (Xilinx)
    - 18ビット長、16レジスタを持つ
    - 96スライス、44MHz動作(Spartan-3)
  - Nios (Altera)
    - 16ビットのNios16と32ビットのNios32
    - 状況に応じて構成をチューンすることが可能
  - Cortex-M1(Actel)
    - ARM互換32bitプロセッサ

FPGAには、先に紹介にしたハードコアのCPUを持たないものは、内部の構成要素を組み合わせるソフトコアCPUを作ります。各社共に、FPGAに適した構成のCPUを用意しており、プログラム環境毎提供しています。かなり本格的なCPUもありますが、上位構成のもの多くは有料です。もちろん、オープンソースのCPUや各自設計したCPUを使う場合も多く、この辺がFPGAの特徴です。

## 4. アクセラレータとしてのFPGA

- Stratix 10
  - 14nm Intelプロセス利用
  - HyperFlexの採用によりGHz台の動作周波数
    - 配線構造上にレジスタを置く
  - 最大10TFLOPSの浮動小数DSPモジュール
  - ARM Cortex A53 Quad Core
- Arria 10
  - 20nm TSMCプロセス
  - 最大1.5TFLOPS
  - SoCタイプはDual Core ARM Cortex

## Open-CLの利用

- GPU同様のアクセラレータとして扱う
  - ホストから入力データを転送→処理を起動→結果を収集
  - ホストの選択
    - PCIe経由でIntelのCPUを使う
    - SoCタイプでは内蔵ARMを使う
- BSP(Board Support Package)が必要
  - ボード依存性を吸収
  - 自作ボードにチップを使ってOpenCLを使うのが極めて難しい→ボード単位で利用するしかない
- HDLモジュールとの接続が難しい

# Arria10 SoCボード



SSDより簡単にLinuxがブートする  
Ethernetでネットワークに接続  
内蔵ARMをホストにOpenCLでの設計ができる

## ではGPUより速いのか？

	Stratix 10	Tesla P100
最大TFLOPS	9.2？	9.3
最大電力(W)	33-45	250
価格	177万(開発キット)	81万(Amazon)

- まだStratix 10とGPUとの比較は国際会議などでは出てきていない
- Arria10の場合は、アプリケーションとチューニングのテクニックによるがGPUには絶対性能では勝てない場合が多い。しかし電力性能では勝つ
- コストは現時点では不明(出だしなので高価すぎる)だが、Stratix Vを考える(シリーズによるが1チップ当たり130万円くらいする)と、GPUに比べて倍以上するのでは？

## アクセラレータとしてのFPGAは成功するか？

- OpenCLの導入によりソフトウェア開発者、スパコン屋が利用可能になった
  - しかしFPGAの良い所が半減したかもしれない
- 科学技術計算では、GPUに性能面では勝てそうもない
  - Stratix 10ならば良い勝負ができるかもしれない
  - しかし値段が相当違う、、、
  - そのうちスパコン屋に飽きられるのでは？
- AI分野は？
  - OpenCLを使っているとGPU
- Intellには勝算があるのだろうか???

OpenCLの無料化  
intel HLSの登場

## ソリューションの提供

- Stratix10クラスの性能を持ち、ターゲットアプリケーションが決まっている(例えばCNNの学習)場合
    - HDLを使って専門家が根性でチューニングする
    - 性能、電力でGPUに勝てる可能性がある
  - カスタマがはっきりしていれば勝機がある
- Maxelerのビジネスモデル
- Cloudで提供すれば機会が広がる

## 5.FPGA in Cloud

- Catapult project by Microsoft [ISCA14他]
  - FPGAを用いた検索エンジンの導入
- FPGA Supervessel Cloud by IBM
  - FPGAによるサービスの提供[ICFPT2016]
  - Xilinx SDACcellによるアプリケーション開発
    - OpenCLに似た開発環境
- Amazon EC2 F1インスタンス
  - Cloud上でのFPGAアプリケーション開発環境
  - Xilinx Ultrascale+を利用
  - さほど利用は広がっていない

# Microsoft's Catapult

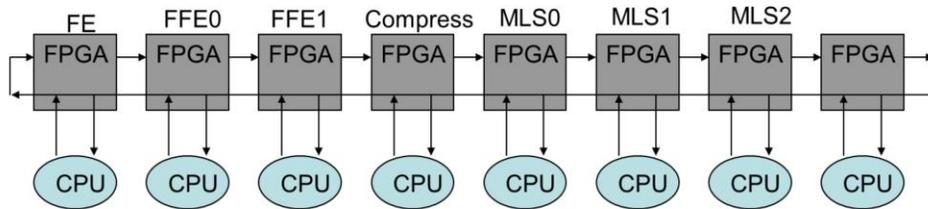
Rank computation for Web search on Bing.

Task Level Macro-Pipelining (MISD)

FE: Feature Extraction

FFE: Free Form Expression: Synthesis of feature values

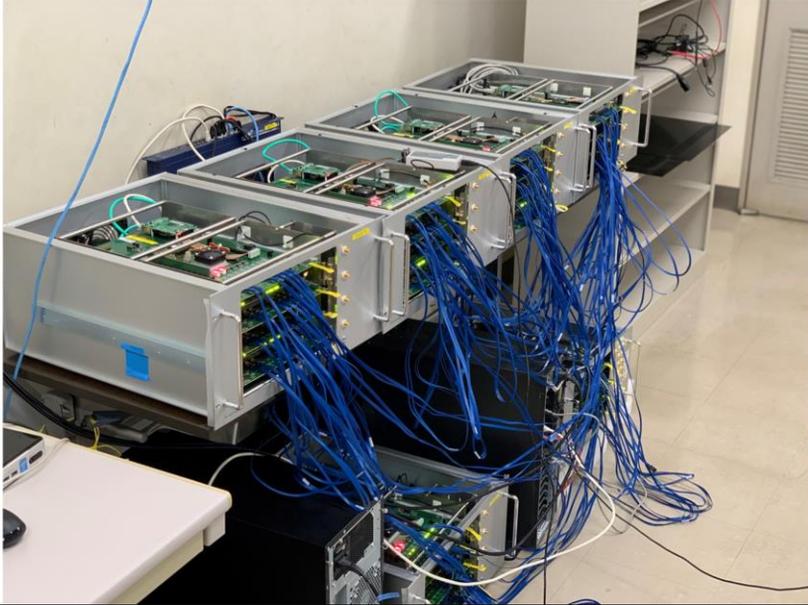
MLS: Machine Learning Scoring



FPGA: Altera's Stratix V

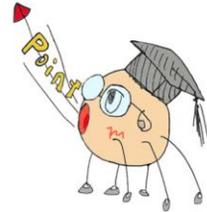
2-Dimensional Mesh is formed (8x6) for 1 cluster.

## FiC(Flow in Cloud)



## 今日のポイント

- FPGAはデジタル回路を簡単に実現するためには理想的なデバイス
- LUTを用い、SRAMに構成情報を入れておくタイプが急激に発達
  - 通常のCMOSのプロセスだけで実現できたのが原因
- ASICに比べて、性能、大量生産時のコスト、電力すべて劣るが、NRC(Non-Recurent Cost)が節約できる
- HDL設計が主流だがC言語設計も広がる
- IntelがAlteraを買収し、FPGAは演算加速(アクセラレータ)としても展開。
- FPGA in Cloudなど新しい動きが一杯



今日のポイントをインフォ丸が示します。FPGAは急激に発達しているデバイスです。世の中の動きに注意しましょう。

## 演習2

- 以下の説明に当てはまるデバイスを下の語群から選べ
1. コンピュータの主記憶に使われる記憶素子、大容量だがやや使いにくい
  2. 現在のプログラマブルデバイスの主流、Configuration dataは電源を切ると消失する
  3. 特定目的用のIC、デジタルカメラ、ビデオ、スマートフォンなどに用いられる
  4. 高速、高信頼性のプログラマブルデバイスで、一度プログラムすると、構成の変更はできない
  5. 大容量の不揮発性メモリ、最近はディスクの代わりに用いられる
  6. コンピュータのキャッシュなどに用いられる揮発性メモリ、高速で使いやすい
- SRAM型FPGA、アンチヒューズ型FPGA、ASIC、SRAM、DRAM、フラッシュメモリ

今回は珍しく語句の選択問題です。

## 次回SPICE演習

- 12階1階12-107に集合
- 毎週火曜に演習をやっている部屋
- 移動用に5分遅らせてスタート