

Technical Note

ELSENA

PLD の歴史を紐解く

文書管理番号: ELS0298_S000_10

2006年4月

株式会社エルセナ

PLD の歴史を紐解く

目次

1 はじめに.....	3
2 PLA, PAL が PLD の元祖	3
2-1 歴史は PLA, PAL から始まった	3
2-2 当時の PLA/PAL の特徴.....	4
2-3 第一世代のロジック・デバイス.....	4
2-3-1 PROM の構造	5
2-3-2 PLA の構造	6
2-3-3 PAL の構造	7
2-4 デバイス内にロジックをインプリメントする.....	7
3 PLD の変遷.....	9
3-1 初代 PLD.....	9
3-2 PLD 構造の第一次進化.....	10
3-3 PLD 構造の第二次進化(歴史を変えた第二次進化)	11
3-4 PLD 構造の第三次進化(大規模化への第一歩)	13
3-5 PLD 構造の第四次進化(FPGA の登場).....	15
3-6 PLD 構造の第五次進化.....	19
4 まとめ	20
改版履歴.....	21

1 はじめに

ミリオン・ゲート時代を迎えて CPLD/FPGA は、その規模において ASIC の領域に迫っています。これらのデバイスは、益々巨大なブラック・ボックスになってきました。一方、開発環境は、VHDL や VerilogHDL による HDL 言語開発が主流となっています。伝統的な PALASM や ABEL などの言語を使用していた時代は、デバイスのアーキテクチャを意識しないと上手く設計できませんでしたが、現在の HDL 言語と論理合成ツールを使えばデバイスの内部を知らなくても開発することができます。しかしながら、多くの洗練されたデバイスを開発することは文字(HDL)の世界だけでは解決しません。『もっと沢山回路を集積したい』、あるいは、『もっと早く動作させたい』と言った要求に駆られることが多くなって来ています。このような要求に応えるためには、デバイスの内部を良く理解して置くことが解決が早くなります。

また、今日のミリオン・ゲート時代に CPLD/FPGA を始めて使い始めた技術者は、PLD の生立ちやその変遷を勉強する機会が余りありません。そこで、この資料では、初期のロジック・デバイス (PAL/PLA/PROM) から今日の大規模デバイスまでにデバイス・アーキテクチャがどのように変遷して来たかを解説します。

FPGA とは、Field Programmable Gate Array の頭文字です。

PLD とは、Programmable Logic Device の頭文字です。

CPLD とは、Complex Programmable Logic Device の頭文字です。(大規模 PLD の意)

SPLD とは、Simple Programmable Device の頭文字です。(小規模 PLD の意)

注釈：本資料は、2003 年に出稿したものを加筆訂正したものです。従って、現在の CPLD/FPGA の状況と一部異なるものがありますのでご了承ください。

2 PLA, PAL が PLD の元祖

FPGA/PLD の歴史は、20 年以上前に氾濫していた TTL ロジック・デバイスをフル・エミュレーションし、置き換えることを目的に開発されました。最初に開発されたものが PLA: Programmable Logic Array でほぼ時期を同じくして PAL: Programmable Array Logic が発表されました。PLA と PAL の違いは、後述しますがプログラマブル領域に AND 領域と OR 領域を持つものを PLA と呼び、プログラマブル領域を AND 領域のみとし、OR 領域を固定としたものを PAL と呼びます。これらのデバイスには、各々幾つの特徴があり、PLA ではロジックの構成において柔軟性を持ち、PAL ではセル面積を小さく、しかも、高速に動作させることができます。現在は、集積度を上げ易く、しかも、高速化が容易な PAL 構造をしたものだけが使われています。しかも、PLD と呼ぶと PAL 構造を示すように代名詞化しています。

2-1 歴史は PLA, PAL から始まった

20 年前は、TTL や CMOS の汎用ロジックを使って、プリント基板上でパターンにより接続してシステムを構築していました。当時のゲート数は、SSI (Small Scale Integrated Circuit) で 1 パッケージ当たり 2 入力ゲートが 4 個、3 入力ゲートが 3 個、或いは、4 入力ゲートが 2 個入り为标准でした。また、MSI (Medium Scale Integrated Circuit) では、1 パッケージ当たり 1 個の 4 ビット・カウンタ、或いは、4 ビ

ット・アダー程度の集積度が標準でした。この時のパッケージのピン数は、14ピンと16ピンでした。

前述したように、TTL デバイス全盛のころは、AND、OR やフリップ・フロップ回路を筆頭に各種のカウンタやアダーを含めると百種類以上にも及びました。AND ゲート1つ取っても、入力本数(Fan-in)毎にパッケージが別れており、2入力 AND パッケージ、3入力 AND パッケージ、4入力 AND パッケージ、そして、8入力パッケージと4種類もありました。これが、AND だ。OR だ。はたまた、XOR だ。トータンポール出力だ。オープン・コレクタ出力だ。と言うと基本ゲートだけでも数 + 種類にも及びました。それに、フリップ・フロップを加えると D 型フリップ・フロップだ。JK 型だ。SR 型だ。T 型だ。はたまた、モノステーブル・マルチバイブレータだ。と言って、また、数 + 種類となりました。これに加えて、基本ゲートを組み合わせた AND-OR 回路や各種のカウンタ、アダーを含む MSI を含めると総勢数百種類にも及びました。当時の技術者は、7~8センチもある分厚いデータ・ブック(TI 社の黄色のデータ・ブックが最も有名)を片手に悪戦苦闘しました。しかしながら、当時の技術者は、この数百あるデバイスを殆ど暗記しました。パッケージのピン配置を含めてです。当時の技術者は“神”がっていました。

このような混沌とした状況を解決しようと登場したのが PLA/PAL です。

2-2 当時の PLA/PAL の特徴

当時の PLA/PAL の最も大きな特徴は、次の2点です。

部品の標準化(統一化)

仕様変更や設計変更に対応

部品の統一化(標準化)に関しては、数百種類もあった TTL ファミリーのデバイスの内、特殊なものを除いて、20種類程度と大幅に簡素化しました。また、仕様変更/設計変更を可能とするソリューションに対しては、伝統的な手法の場合は、この要求に応えるために PWB のパターン変更などの外科的手術を行いました。しかし、PLA/PAL を使うことにより、殆どのケースで PWB のパターンをカットするような事態を避けて仕様/設計変更を簡単に、しかも、何度でも行うことができましたので、当時の技術者はこぞって採用しました。中毒を起こした技術者も居たほどです。但し、当時のプログラミング・テクノロジーは、Bipolar Fuse を使ったワン・タイム・プログラミングであったためソケット実装により複数の変更に対応しました。(注:時を経て、UV-EPROM テクノロジーを使って、始めて、再書込み可能な PAL を発表したのが ALTERA 社で、EPLD: Erasable Programmable Logic Device と呼ばれています。)

2-3 第一世代のロジック・デバイス

PLA/PAL が登場する以前は、プログラマブル・ロジック・デバイスとして使用できるものは PROM だけでした。その用途は、コード・コンバータ、あるいは、シーケンス・コントローラ等のかかなり限られた用途だけでした。それ以外の制御回路は、全て、TTL デバイスを使用して構築していました。そこに、登場したのが PLA/PAL です。表 1 に第一世代のロジック・デバイスを示します。

ロジック・デバイス	アレイ		用途
	AND	OR	
PROM	固定	プログラマブル	コード変換等
PLA	プログラマブル	プログラマブル	汎用
PAL	プログラマブル	固定	汎用

表 1. 第一世代のロジック・デバイス

- PROM AND 領域は固定で、OR 領域がプログラマブルとなっています。AND 領域は、アドレス生成回路として知られています。
- PLA AND 領域、OR 領域ともにプログラマブルとなっています。用途としては、一番汎用的である組み合わせ回路を簡単に構築することができます。しかしながら、集積化と高速化にやや難があります。
- PAL AND 領域がプログラマブルで、OR 領域が固定となっています。汎用性が高く、殆どの組み合わせ回路が構築でき、集積化・高速化に適しています。現在の PLD と呼ばれるデバイスは 殆どのものがこの形式をしています。

2-3-1 PROM の構造

図 1 に PROM の概念図を示します。

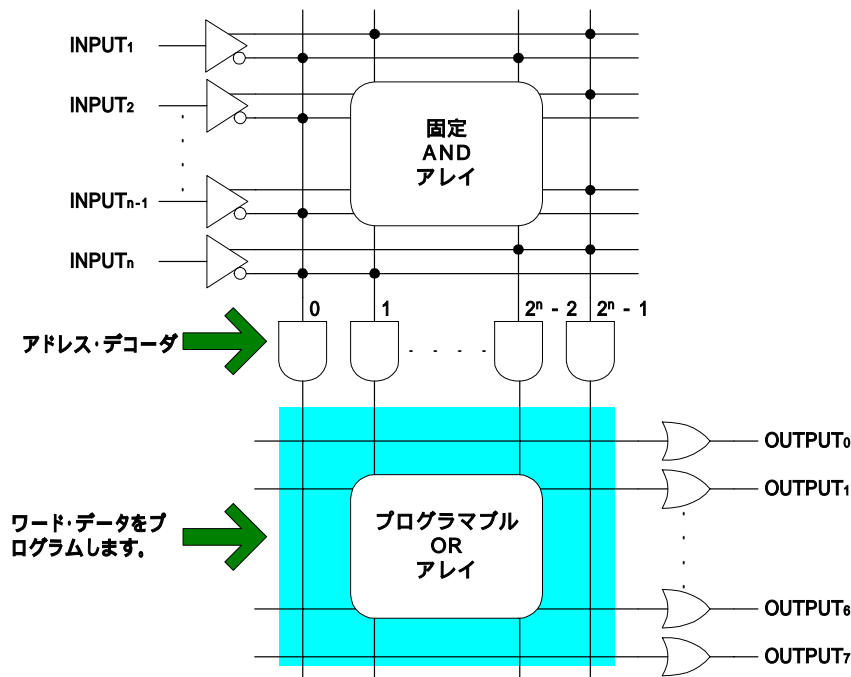


図 1. PROM の構造

先に述べた通り、AND 領域は固定になっており、入力信号に応じたアドレスを生成するアドレス・デコーダとなっています。プログラマブルOR領域は、各アドレスに対する同一ビットが全てORされて出力されています。従って、各アドレス上のメモリ位置に出力パターンがプログラムされます。このような動作機能から、コード・コンバータ、シーケンス・コントローラ、あるいは、パターン発生器等に使用されています。

2-3-2 PLA の構造

図 2 に PLA の概念図を示します。

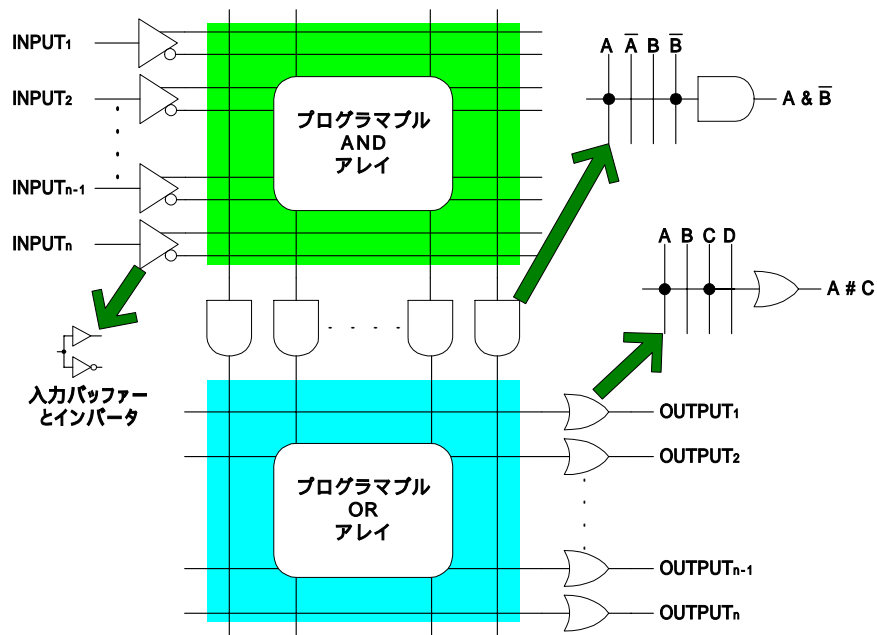


図 2. PLA の構造

図 2 で分かるように、AND 領域、OR 領域ともにプログラマブルとなっています。この時代のプログラミング・テクノロジーは、Bipolar Fuse を使っていたので、全ての入力を接続しておいて、設計後、不要な Fuse を溶断して回路から切り離し、論理を構成するものです。

全ての入力ピンには、後段のプログラマブル AND 領域で正負の極性入力を自由に使えるように遅延が均等化されたバッファとインバータが付属しています。

2-3-3 PAL の構造

図 3 に PAL の概念図を示します。

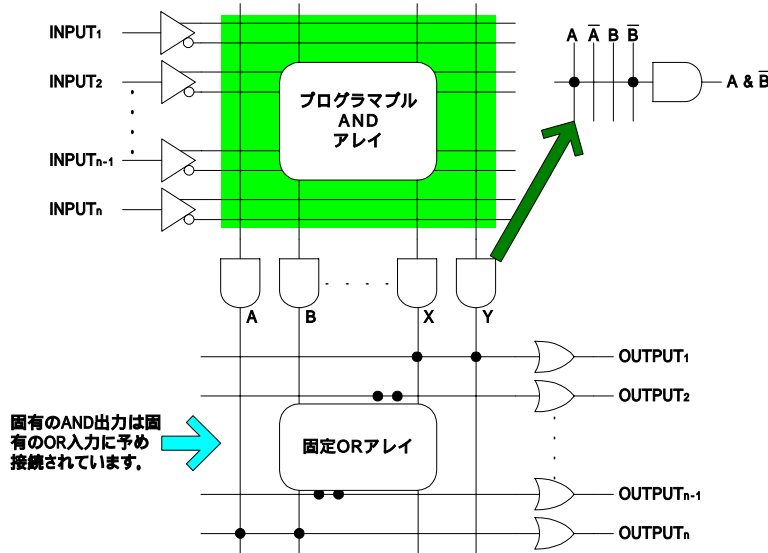


図 3. PAL の構造

AND 領域がプログラマブルで、OR 領域はプログラマブル AND のあるグループの出力が予め接続されています。一般的には、8 本の AND 出力グループを構成して OR に接続され、それが 1 つの基本 AND-OR ブロックとなっています。PAL の場合も全ての入力にバッファとインバータが付属しています。

2-4 デバイス内にロジックをインプリメントする

ここでは、PLD の中にロジックがどのようにインプリメントされるかを示します。図 4 にデバイスがプログラムされる前の回路表記とプログラムされた後の回路表記を示します。

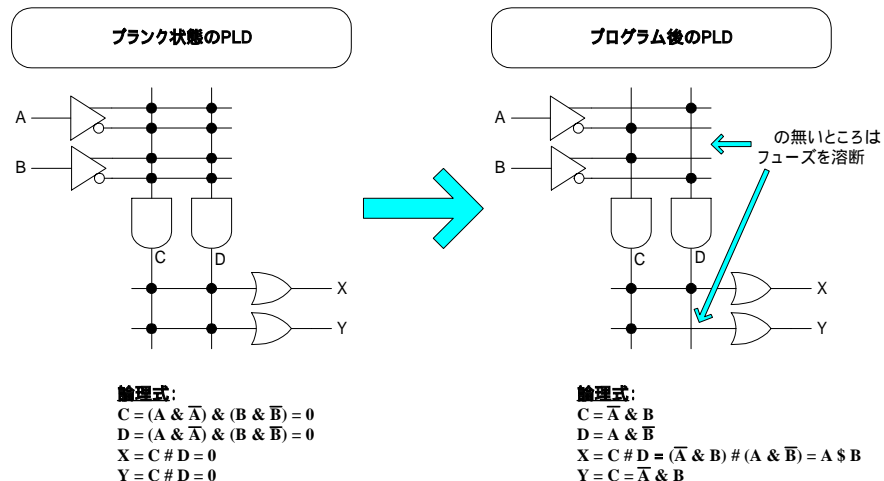


図 4. デバイスへのロジックのインプリメント

図 4 の左図がプログラム前の空白状態の PLD の内部を示しています。AND 回路には入力 of 全てが接続されています。1つの AND 回路には同じ入力の”正”と”負”が接続されていますので、結果として全ての AND 出力は論理”0”となります。OR 回路には、全ての AND 出力が接続されていますが、全ての AND 出力は論理”0”を出力していますので、結果として OR 出力も論理”0”を出力することになります。論理式は、次のようになります(入力、出力、中間パラメータは図内のものを使用)。

図 4 の左図の論理式:

$$C = (A \& /A) \& (B \& /B) = "0"$$

$$D = (A \& /A) \& (B \& /B) = "0"$$

$$X = C \# D = "0"$$

$$Y = C \# D = "0"$$

ここで、Exclusive-OR(XOR)をプログラムしてみましょう。XOR の合成式は、次のようになります。

XOR の合成式:

$$X = (A \& /B) \# (/A \& B) = A \$ B$$

図 4 の右図を見て行きましょう。左図で全て接続されていた回路において生成すべき論理に基づいて不要な Fuse を溶断して回路を構成して行きます。まず最初に、中間 AND 出力 C に対し、入力 A 側では”正”出力側を溶断し、入力 B 側では、”負”出力側を溶断することにより、 $C = (/A \& B)$ を生成します。同様にして、中間出力 D に対しては、 $/A$ 側と B 側を溶断することにより、 $D = (A \& /B)$ を生成します。その結果 OR 出力 X は、次のようになり、XOR 回路が生成されました。

図 4 の右図の X 出力:

$$X = C \# D = (/A \& B) \# (A \& /B) = A \$ B$$

図 4 から分かるように、デバイス内の構成を等価回路で表す場合は、信号線の交点が接続されている場合は、黒丸(●)を付して記述します。また、交点が接続されていない場合は、互いの線を直交させるのみで黒丸(●)は付しません。

ここで、既にお気づきかも知れませんが、PAL/PLA は多くの Fan-in を持っています。この Fan-in 線を回路上に全て記述すると回路が非常に煩雑になります。そこで、業界では、以下のように略記するのが通例となっています。図 5 に多入力 AND、図 6 に多入力 OR の略記図を示します。

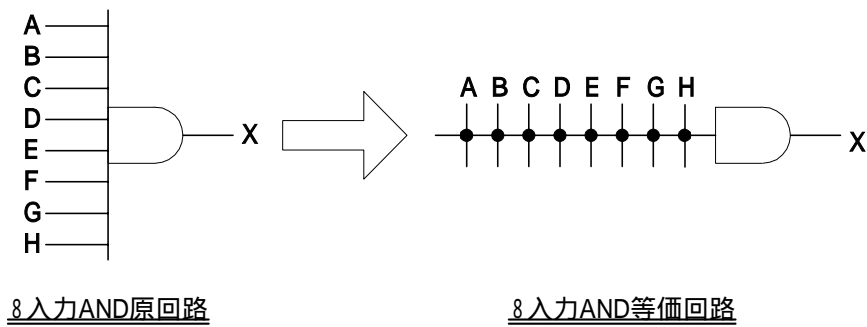


図 5. 多入力 AND の略記図

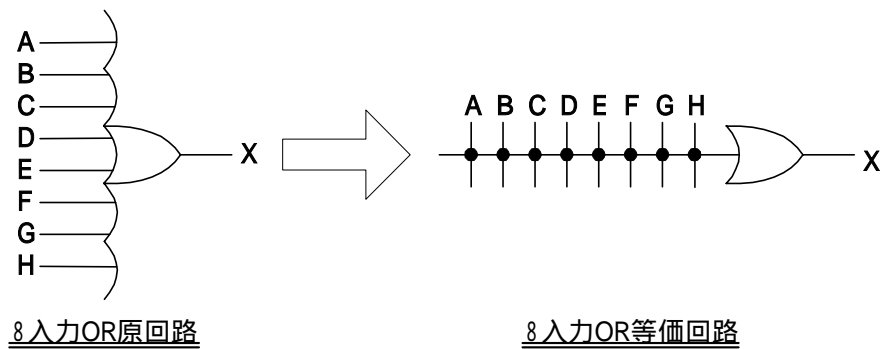


図 6. 多入力 OR の略記図

3 PLD の変遷

3-1 初代 PLD

それでは、現在主流となっている PAL 構造をした PLD に焦点を絞って説明して行きましょう。図 7 に初代 PLD の等価回路を示します。

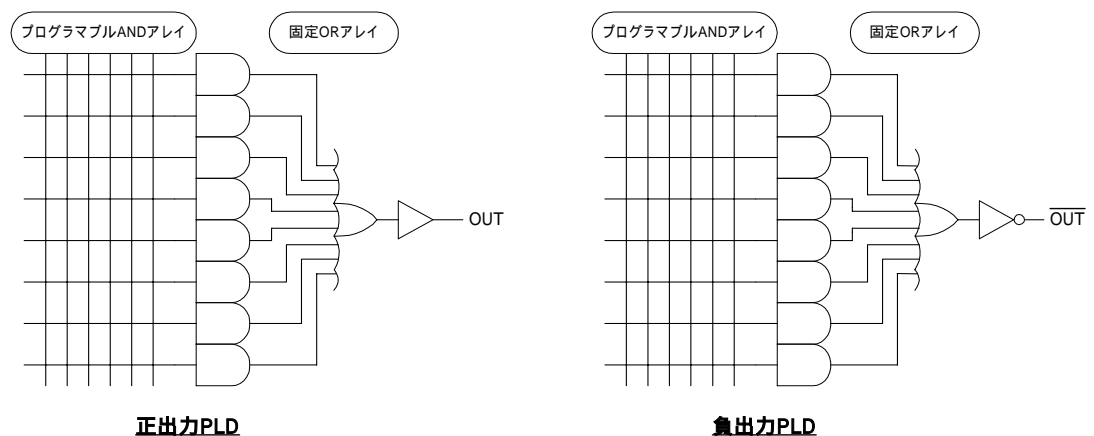


図 7. 初代 PLD 等価回路

初代の PLD は、『現存する TTL を如何に代替するか』と言う考え方を原点としていますので出力極性は、正出力、負出力が各々固定となっていました。また、大きな回路や小さな回路を自由に組めるよ

うにと Fan-in 数と出力本数とでパッケージ分けをしていました。この辺の考え方は、相変わらず TTL パッケージの思想が踏襲されています。

発表当時の PLD は、入力本数と出力極性毎にデバイスが異なっていました。例えば、代表的な 20 ピン・デバイスであれば、16H2, 14H4, 12H6, 10H8 といったものがあります。ここで、アルファベット H は正出力極性を表し、その前の 2 桁の数字は、1 つの AND 回路の Fan-in 数を表しており、H の後の数字は、出力本数を表しています。また、出力 OR に接続されている AND 出力の本数はデバイスによって異なっています。負出力デバイスは、24 ピン・デバイスとして、20L2, 18L4, 16L6, 14L8, 12L10 があり、16 ピン・デバイスとして 16L2, 14L4, 12L6, 10L8 があります。(アルファベット L は、負出力極性を表しています。)

3-2 PLD 構造の第一次進化

続いて、トリステート出力を持つものと D 型フリップ・フロップを持つものが現れました。これによって、殆どのグルー・ロジックの構成が可能となりました。図 8 にトリステート出力 PLD の等価回路とフリップ・フロップ付 PLD の等価回路を示します。この頃から AND-OR 項に対する 1 つの形が固まりつつありました。Fan-in が 16 本で出力は 8 本、かつ、OR に接続される AND 出力は 8 本となりました。図 8 の左図から分かるように、トリステート出力を持つデバイスでは、8 本の内の一本をトリステート・コントロールとして使用しています。また、右図のフリップ・フロップを持つデバイスでは、トリステート・コントロール OE とクロック CLK は、専用ピンを持ち全ての回路に共通に配線されています。

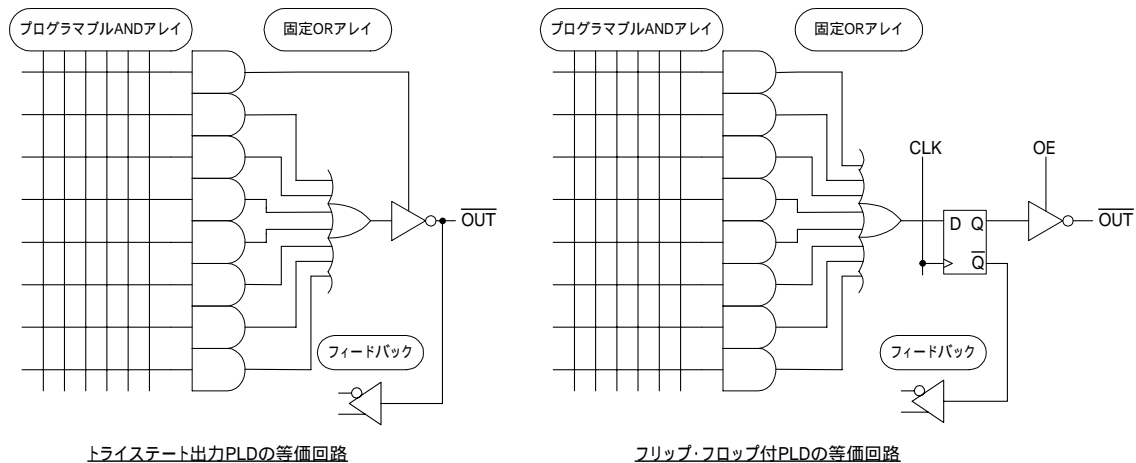


図 8 トリステート付 / フリップ・フロップ付 PLD

先に述べたように、Fan-in 数や出力本数、出力極性、トリステート付、あるいは、フリップ・フロップ付とパッケージが分かれたことにより、TTL の種類に比べれば少ないものの種類が増えて行きました。ここで登場したのが、回路内に Exclusive-OR(XOR)回路を内蔵させ、出力極性をプログラマブルにする試みです。

図 9 に XOR 回路内蔵 PLD の等価回路を示します。

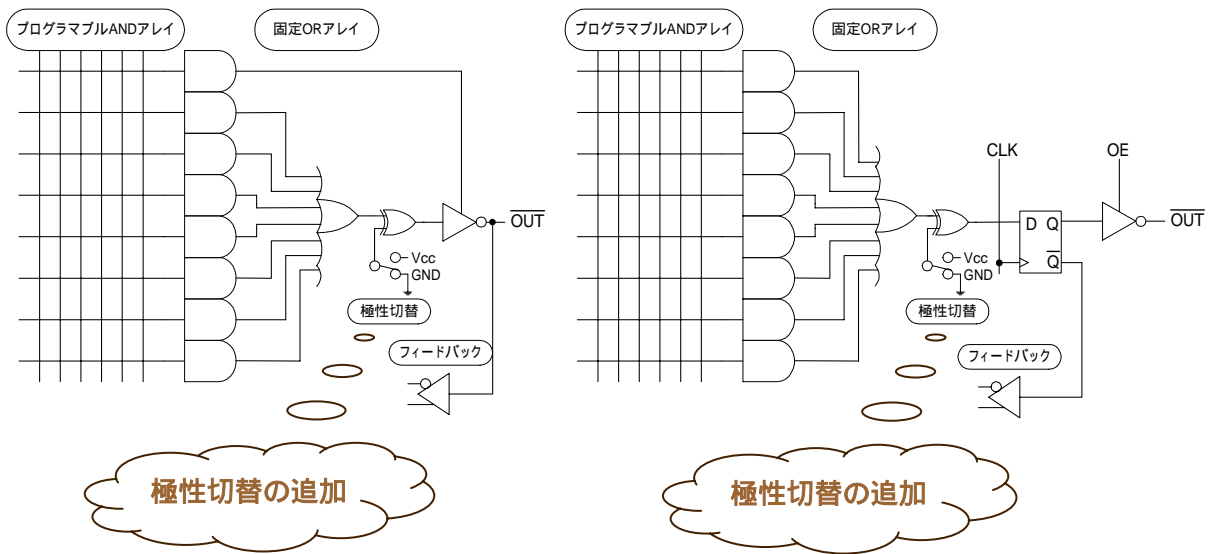


図 9. XOR による出力極性切替付 PLD

図 9 において、出力極性切替のための XOR の入力、一方が AND-OR 出力に、もう一方が極性切替のための Vcc/GND に接続されています。PLD がブランク(デフォルト)の状態では、極性切替用入力は、グラウンド(GND)に接続されています。ですから、このまま出力すると、XOR 回路は、単なるバッファとして動作します。片や、出力を正極性で出力したい場合は、極性切替用入力を電源 (Vcc)に接続し、インバータとして動作させます。

3-3 PLD 構造の第二次進化(歴史を変えた第二次進化)

過去の PLD の歴史を大きく塗り変えたのが、AMD 社から発表されたマクロセルと言う考え方です。AND-OR の後段にマクロセルが配備され、そのマクロセルは次の回路機能を持っています。

- ✓ フリップ・フロップ
- ✓ 極性切替、および、フリップ・フロップ、或いは、コンビナトリアル出力切替用マルチプレクサ
- ✓ トライステート・バッファ
- ✓ ピン・フィードバック、或いは、フリップ・フロップ出力フィードバック切替用マルチプレクサ

図 10 に AMD 社が発表したマクロセル構造の PLD(22V10)を示します。

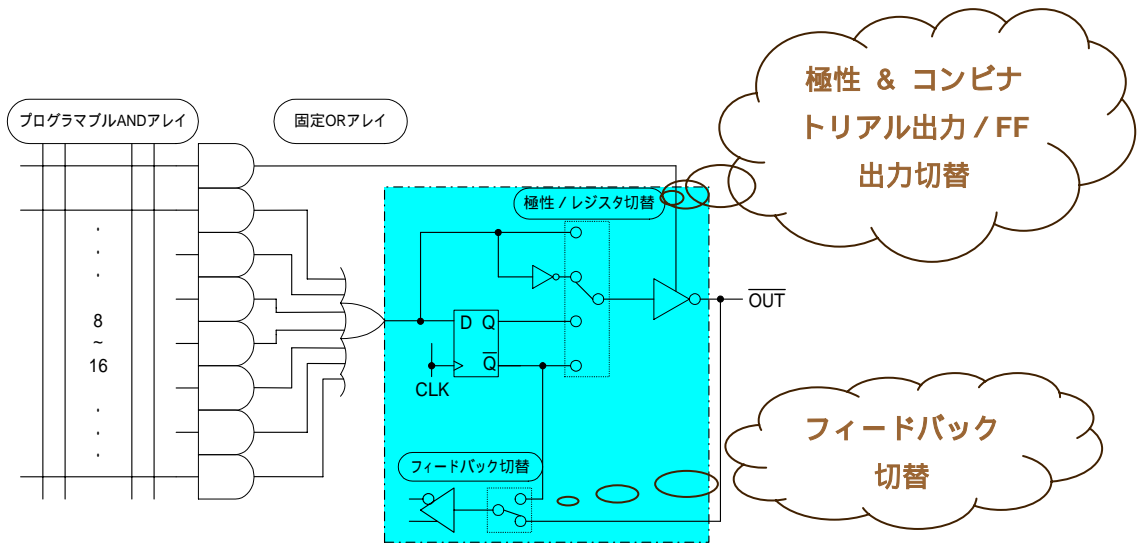


図 10. 22V10 のマクロセル構造


 本来のマクロセルには、AND-OR アレイの部分は含まれず、後段の各種のオプション機能の切替部分のみを指していました。しかしながら、最近では AND-OR アレイを含めた一体の物をマクロセルと呼んでいます。

図 11 にマクロセル内の 2 つのマルチプレクサを切替えることにより、4 つの組み合わせを構成したところを示します。

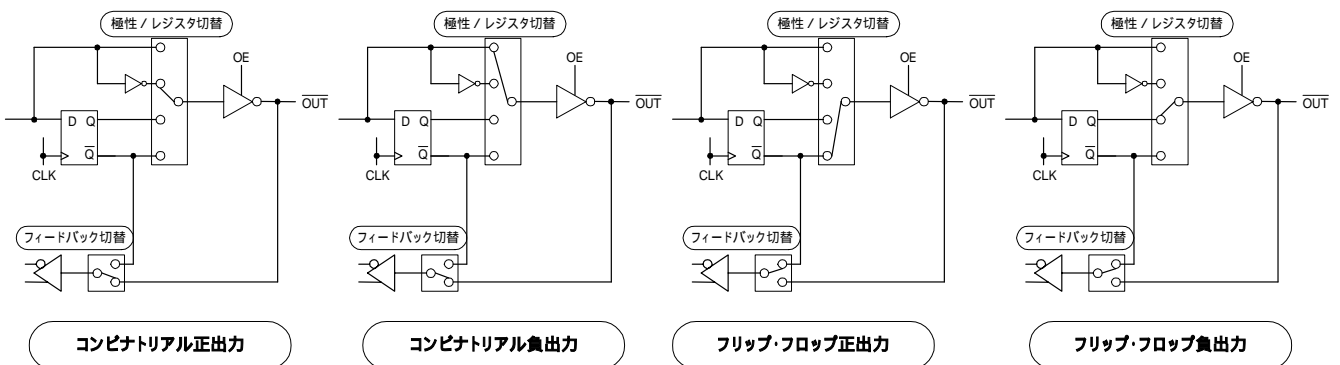


図 11. マクロセルの構成

図 11 で示す 4 種類の構成が 1 つのマクロセル内のマルチプレクサの切替えで構成できることにより、既に存在していた PLD の全てを置換できたと言っても過言ではありません。画期的な提案です。PLD 業界においては、発明と言っても良いでしょう。現在の PLD 構造は、何らかの形でこのマクロセルを原形として改良したものと言えます。

さらに、22V10 では、新たな試みがなされています。既存の PAL の AND タームの本数は最大で 8 本まででした。これより大きな回路を組む場合に、一度中間出力をピンに出した後に再度次段で拡張す

ると言った構成を取っています。この場合は、段数がかさむとその段数分の遅延が生じ、極端にパフォーマンスの低下を招きました。この問題を解決するために、出力ピンに応じて 8 本から 16 本までの OR 構成を持つマクロセルを用意し、構成するロジックの規模に応じて使い分けられるようにしました。この考え方が次のプロダクトターム (AND ターム) を可変する考え方に繋がって行きます。

3-4 PLD 構造の第三次進化(大規模化への第一歩)

PLD も第二次進化まで改良が加えられると、さらに新たな要求が生まれました。それは大規模化です。そこで、PLD 構造の第三次進化に求められた事柄を列挙します。

- ✓ 大規模化
 - 複数デバイスの統合
 - 内部デバイス間の接続用アレイ
- ✓ TTL フル・エミュレーション
- ✓ 埋め込みマクロセル
- ✓ OR アレイの柔軟性
- ✓ XOR 機能の必要性
- ✓ プロダクトタームの有効利用

大規模化のために複数の SPLD(Simple PLD)を 1 つのパッケージに封じ込める必要が生まれました。複数の SPLD を統合することにより、よりシステム化が容易になります。また、内部に統合された SPLD 間を接続するための高速でスキューの小さい内部配線用アレイが必要になりました。

一方、伝統的な TTL/CMOS デバイスを統合するために TTL 機能のフル・エミュレーションが求められていました。基本的には、ワンショット・マルチバイブレータとアステープル・マルチバイブレータを除き全ての機能の実現が求められました。

さらに、埋め込みマクロセル(ベリード・マクロセルと言う)がパッケージ・ネックを解決しました。第二世代 PLD までは、全ての出力がピンに出ていました。複数 SPLD の 1 パッケージ化により、全てのピンを出力すると、パッケージが大きくなってしまいます。そこで、マクロセルの出力がピンに配線されずに、内部の他のマクロセルにだけ配線されるような構造が生まれました。これがベリード・マクロセルです。この考え方により、大規模化の道を開きました。

また、OR アレイの柔軟性が内部リソースの有効利用度を改善しました。第二世代 PLD までは、8-Fan-in 固定の OR 回路でしたので、次の 2 つの問題を持っていました。

8-Fan-in を超える OR 回路を構成する場合、一度中間出力として外部ピンに出力しなければならないこと。 出力ピンの無駄使い。遅延の増大。

プロダクトターム (AND ターム) を余り使わない小規模回路の場合、未使用プロダクトタームが無駄になる。例えば、極端な例をあげると、1つのマクロセルをインバータとして使用した場合が最も非効率な使い方で、未使用の7本のプロダクトタームとフリップ・フロップが全く不使用となります。また、単純なシフトレジスタのような場合は、フリップ・フロップの D 入力に対し、1本のプロダクトタームだけが使われますので、他の7本のプロダクトタームが不使用となります。従って、如何に内部のリソースを有効利用するかが問題となりました。

このように、構成する論理によって柔軟に OR 回路の構成を変えられるように各 PLD ベンダは研究を重ねました。

XOR 機能の改良により、益々回路構成が柔軟になりました。当初の PAL に XOR 機能が追加された目的は、出力極性を切替える目的でした。しかしながら、キャリア・アプリケーションであるカウンタやアダプターなどを構成する場合、XOR 演算回路が内蔵されていない場合、全ての項を積和に展開して構成します。この構成の場合、大きな XOR アプリケーションでは、現存する8本のプロダクトタームでは不足する場合があります。そこで、この問題を解決すべく演算が可能な XOR 回路が追加されました。(従来の XOR 回路では、極性切替のためにグラウンド (GND) が電源 (Vcc) のみに接続できる構造でしたが、それを1本のプロダクトタームに接続し、XOR 演算が可能なように改良しました。)

図 12 に第三次 PLD のマクロセル構造を示します。

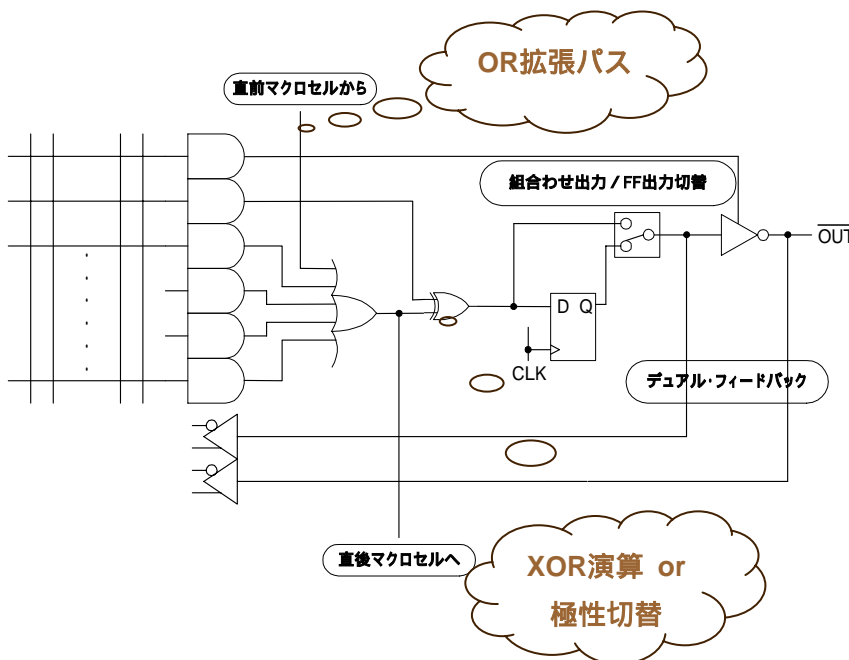


図 12. 第三次 PLD のマクロセル構造 (代表的な例)

図 12 でも分かるように、XOR 回路の入力は、積項線を持ち、XOR 演算や出力極性の切替えを行うようになっています。また、22V10 では、出力極性やフリップ・フロップ出力の切替えを1つのマルチプレクサで行っていましたが、論理の極性の切替えは先の XOR 回路で行い、後段のマルチプレクサでは、

単にフリップ・フロップを使うかどうかだけを切替えています。

新たに加わった機能として、マクロセル出力をベリード・ロジックとして使用する場合に、出力ピンを入力ピンとして使用可能なようにデュアル・フィードバック方式を採用しました。

高い Fan-in OR を構成するために、プロダクトタームの本数は標準で 4~5 本とし、小規模回路の場合に使用しないプロダクトタームが無駄になることを最小限としました。また、大規模回路を構成する場合には、他のマクロセルから未使用タームを借りたり、貸したりして対応しています。

図 13 に第三次 PLD の概念図を示します。

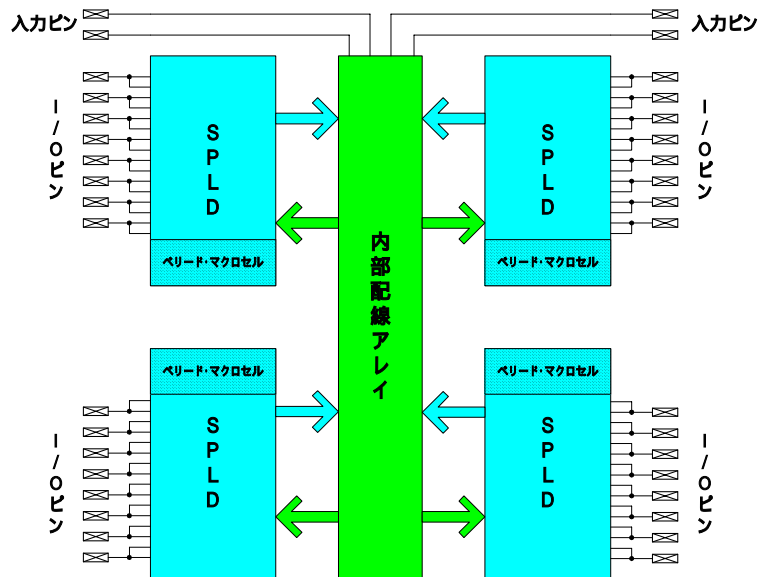


図 13. 第三次 PLD の概念図

複数の SPLD を封じ込めた CPLD(Complex PLD)は、デバイスの内部 SPLD 間を高速に接続するための内部配線アレイ(インターコネクトと言う)が追加されました。これにより、高速でスキューの無い配線が可能となりました。

3-5 PLD 構造の第四次進化(FPGA の登場)

従来の PLD は、積和形式でハイ Fan-in の AND-OR 構成を取っていたので、大規模なカウンタやステートマシン、アダー等に向いていました、しかしながら、大きな欠点は、全てのロジックを構成する上でマクロセルのリソースの利用効率をあげることが難しいことでした。デバイスのアーキテクチャとアプリケーション回路との間に密接な関連と相性があり、両者の相性がベスト・マッチした場合は、非常に力を発揮しました。先にも述べたように、使用効率の点で極端な話をしますと、マクロセルをインバータのみで使用したり、単純なシフト・レジスタとして使用した場合は、多くの内部リソース(プロダクトターム)が無駄になりました。そこで、登場して来たのが小さな論理ブロックを沢山内蔵し、それらを木目細かく

接続して論理を構成する方式です。これがFPGAの登場です。そこで、第四次PLDについて纏めて見ますと次のようになります。

- ✓ FPGA 構造の登場
 - SRAM 型、アンチ・フューズ型、フラッシュ型
- ✓ 小規模 PLD から大規模 FPGA へ
 - ASIC 分野への進出 ... 数千ゲートから数十万ゲート
- ✓ IP (Intellectual Property)の登場
 - 大規模設計への足掛かり ... 標準化
- ✓ ICR/ISP プログラミング
 - ICR: In Circuit Re-configuring
 - ISP: In System Programming

この頃からデバイス規模として数千ゲートから数十万ゲートまでが登場することとなり、小規模 ASIC マーケットの一翼を担うようになりました。また、後期になって、大規模化に伴い IP 利用による設計手法も提案されるようになってきました。一方、デバイス・プログラミングを見ますと、伝統的には固有のデバイス・プログラマを使ってプログラミングを行ってききましたが、SRAM の特徴や EEPROM、FlashPROM などの特徴を生かした ICR/ISP のプログラミングが登場してきました。これにより、試作時における利便性が益々増長されました。この機能を積極的に使用しますと、試作時の書き換えの便利さだけでなく、On-the-fly 書込みやフィールドでのバージョン・アップも可能となってきました。

図 14 に代表的な FPGA 構造の基本演算部を図示します。

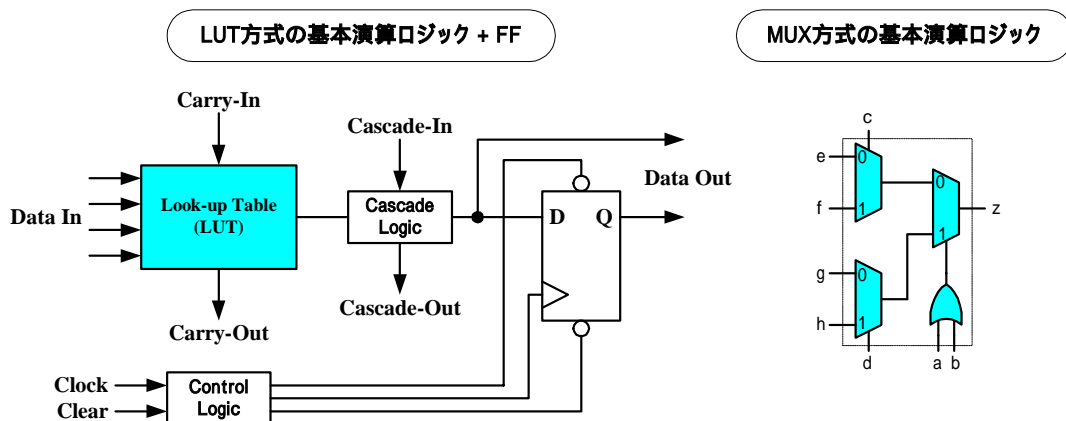


図 14. FPGA の基本演算部

図 14 の左図は、ルックアップ・テーブル(LUT)方式を使用した回路を示し、右図はマルチプレクサ(MUX)方式を使用した回路を示します。この基本演算回路を基本ブロックとし、デバイス内に無数に配置したのが FPGA です。

図 15 は、LUT 方式の FPGA で論理がどのように構成されるかを示したものです。

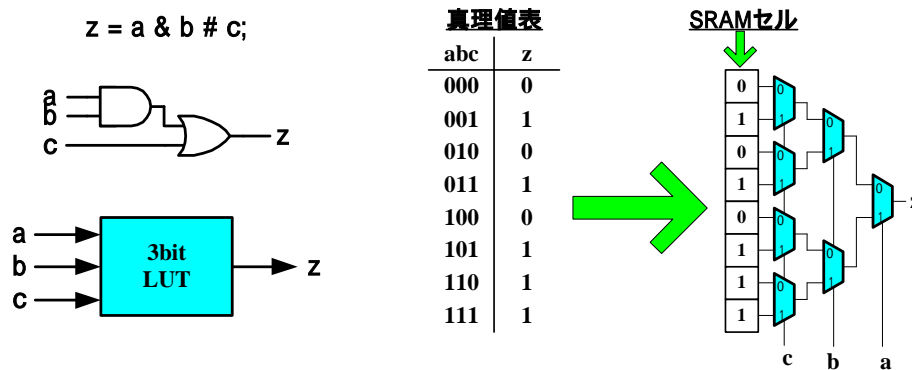


図 15. LUT 方式 FPGA へのロジックのインプリメント

図 15 の左上部の論理式 $z = a \& b \# c;$ を 3 入力 LUT を例にとって説明します。3 入力 LUT は、8 ビットのメモリとして考えることができます。そして、3 本の入力はそのアドレスです。前式を実現するための真理値表は図 15 の真中の図となります。ここで分かるように 3 つの入力(アドレス)で示されるメモリ・アドレス内にその出力 z の値を格納して置くことで演算回路が実現されます。

図 16 は、マルチプレクサ(MUX)方式の FPGA で論理がどのように構成されるかを示したものです。

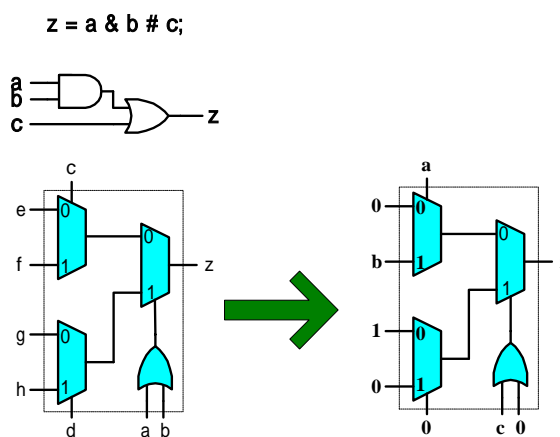


図 16. MUX 方式 FPGA へのロジックのインプリメント

図 16 の左上部の論理式 $z = a \& b \# c;$ を MUX で構成する場合は、左図の MUX の入力とその切換え信号を右図のように接続して実現します。

第四世代 CPLD/FPGA の特徴を纏めたものが表 2 になります。

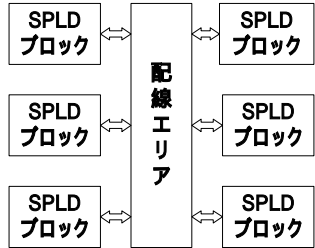

	CPLD/SPLD		FPGA		
テクノロジー	EEROM/EPROM	Flash	Flash	Anti-Fuse	SRAM
集積度	小規模	小規模	中・大規模	中・大規模	中・大規模
不揮発性					X
再書き込み				X	
速度	高速	高速	高速	高速	中速
設計自由度	低い	低い	普通	普通	普通
消費電力	大きい	大きい	普通	少ない	普通
適応回路	ステートマシン	ステートマシン	ランダム・ロジック	ランダム・ロジック	ランダム・ロジック
構成					

表 2. 第四次 CPLD/FPGA の特徴

集積度の点では、CPLD/SPLD は、1K ~ 20K ゲート相当となり、FPGA は、1K ~ 100K ゲートとなっています。また、プログラミング方式としては、使用するテクノロジーにより、次のようになります。

- ✓ UV-EPROM → 専用プログラマ
- ✓ EEPROM → 専用プログラマ、または、ISP (JTAG 或いは、専用回路)
- ✓ SRAM → ICR(JTAG 或いは、専用回路)
- ✓ Anti-Fuse → 専用プログラマ

第四次の後期から組み込み(エンベデッド)機能の考え方が登場し、更なる大規模化の研究が進められ、一部のベンダからデバイス内部に組み込み型のメモリ・ブロックを持つ物が登場しました。そのメモリは、単にメモリ機能ばかりでなく、大きな LUT して乗算器などの大きな演算回路を実現できるようになってきました。

いよいよ第五世代の幕開けです。

3-6 PLD 構造の第五次進化

それでは、現在主流の CPLD/FPGA である第五世代の話をします。

現在では、第四次 CPLD/FPGA のアーキテクチャに益々の磨きがかかり、さらには、100 万ゲートを優に超えるデバイスが実用化されています。ここまで来ますと、1 つのチップの中にかんりのシステムが入るところから SOPC(System On a Programmable Chip)と呼ばれています。これで最早手軽なグルー・ロジックの置き換え物では無くなりました。

そこで、現在の CPLD/FPGA を纏めて見ますと、次のようになります。

- ✓ 100 万ゲートを超える集積度
- ✓ On-chip RAM/ROM などの組み込み機能
- ✓ LVDS, PECL, PCML, SSTL などの高速インタフェース
- ✓ IP による開発が必死
 - 設計 TAT の短縮と設計品質の確保から IP 利用による設計は必死
- ✓ 検証が最大の関心事
- ✓ 設計環境は、ASIC と変わらない。
 - ASIC と同等の高価で複雑な EDA ツールが必要

特に、設計環境として、TAT の短縮と設計品質の確保の目的で IP の利用が成功か否かの鍵を握っています。しかしながら、現在の IP の利用環境は、種類こそ多いもののその品質と使い易さの上で問題があり、今後の改善が強く望まれています。また、大規模化したことにより、検証が非常に大事な作業となってきました。以前ですと、『簡単に変更できるから良いや』と言って検証をないがしろにしがちでしたが、昨今では、真剣に検証に取り組まないと命取りになるほど重要になってきました。同様に、設計環境も ASIC と同等の EDA ツールが必要となってきました。

図 17 に、第五次 CPLD/FPGA 内に複雑なシステムがインプリメントされているイメージを示します。

前述したように、各種の Hard-Core(エンベデッド・コア)と IP を組み合わせて 1Gbps Ethernet の MAC が複数チャンネル、ネットワーク・プロセッサとしてのマイクロ・プロセッサ、各種の FIFO、PLL、あるいは、高速 I/O インタフェースとして LVDS, SSTL, GTL 等も接続する機器によって使い分けことができます。このように、1 つのデバイス内に多くの機能と各種の高速インタフェースが組み込まれていますので、様々なシステムと直接接続することができます。

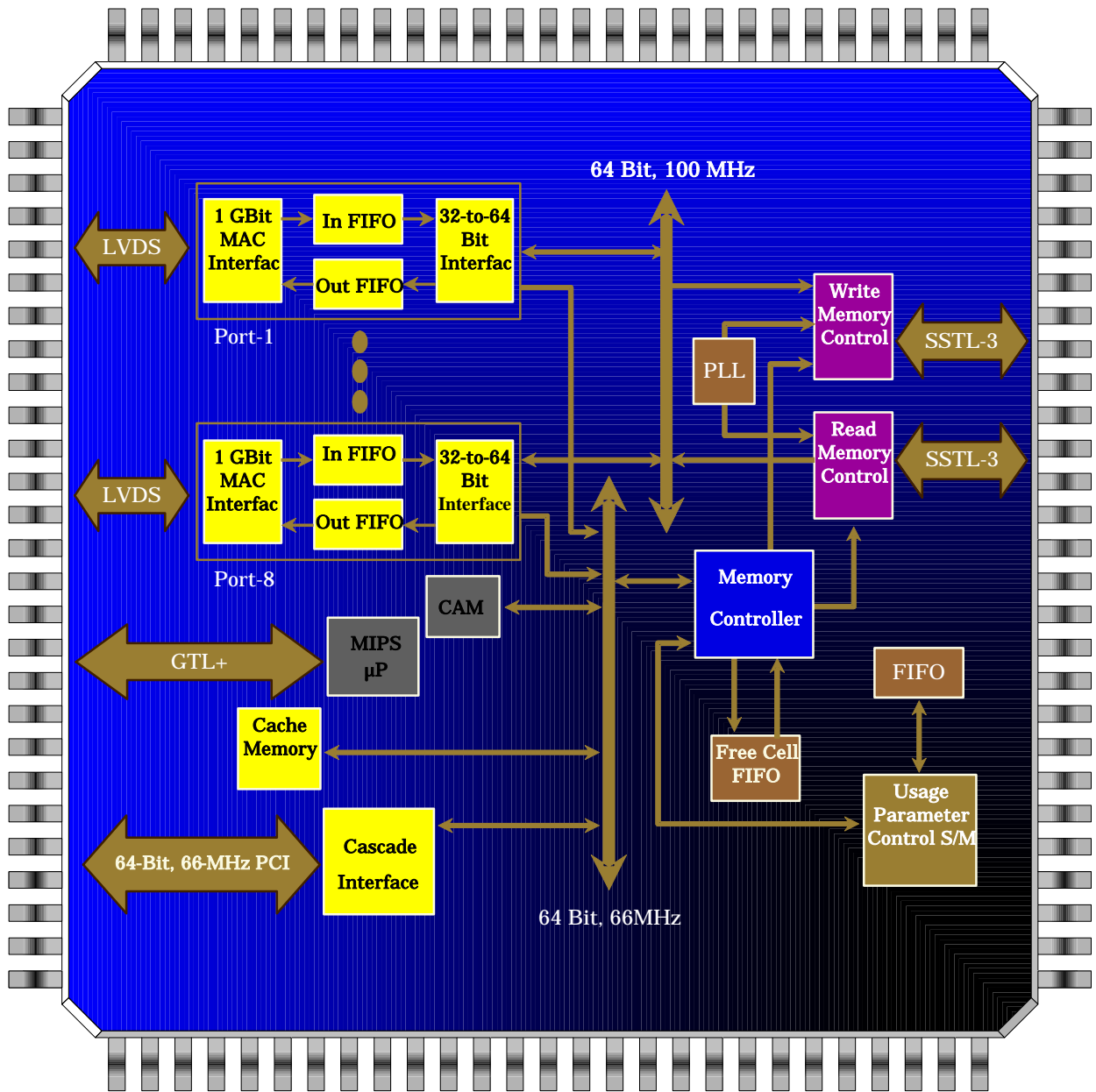


図 17. 第五世代の CPLD/FPGA イネージ図

4 まとめ

以上で、脈々と成長を遂げてきた PLD の構造の変遷について説明してきました。これを一読することで、皆様が現在の CPLD/FPGA の構造を理解し、スマートにこれらのデバイスをお使い戴くことを切に願っています。また、よりシステムへの CPLD/FPGA の採用が加速されることを願っています。

改版履歴

Version	改定日	改定内容
1.0	2006年03月	・新規作成

免責、及び、ご利用上の注意

弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

4. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。
5. 本資料は予告なく変更することがあります。
6. 本資料の作成には万全を期していますが、万一ご不審な点や誤り、記載漏れなどお気づきの点がありましたら、弊社までご一報いただければ幸いです。
7. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。
8. 本資料は製品を利用する際の補助的なものとしてかかれたものです。製品をご使用になる場合は、英語版の資料もあわせてご利用ください。

本社

〒163-0928 東京都新宿区西新宿 2 丁目 3 番 1 号 新宿モリス 28F TEL 03-3345-6205 FAX 03-3345-6209

松本営業所

〒390-0815 長野県松本市深志 1-1-15 朝日生命松本深志ビル 1F TEL 0263-39-6134 FAX 0263-39-6135

大阪営業所

〒532-0003 大阪市淀川区宮原 3 丁目 4 番 30 号 ニッセイ新大阪ビル 17F TEL06-6397-1090 FAX06-6397-1091

名古屋営業所

〒450-0002 愛知県名古屋市中村区名駅 3 丁目 11 番 22 号 IT名駅ビル 4F TEL 052-566-2513 FAX 052-566-2514