

第2章 FPGAのしくみと開発に使うソフトとハード

大規模ロジックはどうやって実現するの？

丹下 昌彦
Masahiko Tange

FPGAのしくみ

● 何にプログラムするのか…おおまかな内部構造を知る
これまで、FPGAは何回も内部の論理回路を変更できる、プログラム可能なLSIであると説明してきました。つまり、FPGAを使うためには論理回路をプログラムする必要があるのです。FPGAでは、論理回路をプログラムすることを**コンフィグレーションを行う**と言います。

図1にFPGAの内部構成を示します。FPGAは、大きく分けると次のようなブロックから成り立っています。

▶ I/O

外部とつながるピンへの入力/出力を行います。設定で、ピンの入出力特性(入力電圧、プルアップ/ダウンの有無、出力電圧、電流など)を自由に設定することができます。

▶ ロジック・セル

FPGAの中心となる部分で、この中に**LUT(ルックアップ・テーブル=論理回路を実現するためのテーブル)**、フリップフロップが入っています。

ゲートICをつなぎ合わせて組み合わせ論理回路を作る場合、AND/OR/NOTなどのゲートを組み合わせ

ますが、FPGAでは、ちょっと違った方法で実現します。

FPGAの組み合わせ論理は**LUTと呼ばれるSRAM**のような回路で作られます。

図2に、LUTとロジックICの違いのイメージを示します。

RAMに値を書き込むことで論理を実現するので、書き換え可能な論理回路を構成しやすいというメリットがあります。

また、LUTは基本的にはRAMと同じようなものなので、小容量のRAMとして使ったり、シフトレジスタとして使えるというメリットがあります。

▶ 内部配線

多数のロジック・セルやI/Oを接続するための配線です。

▶ その他

これ以外にもRAMや乗算器、クロック・マネージャ、周波数通倍/分周器、高速データ通信を行うブロックなどが内蔵されているものが増えてきています。中には、CPUや、イーサネットMAC、積和演算器を内蔵したものもあります。

● 種類によって配線情報の書き込み方が違う

FPGAにコンフィグレーションを行うということは、「これらのブロックに適切な設定を行い、目的の回路

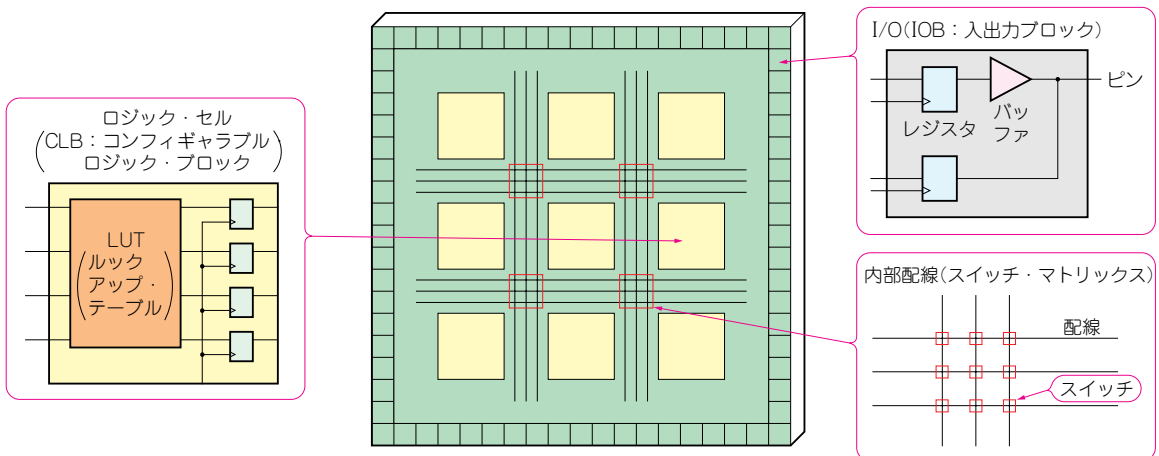


図1 FPGAのおおまかな内部構成

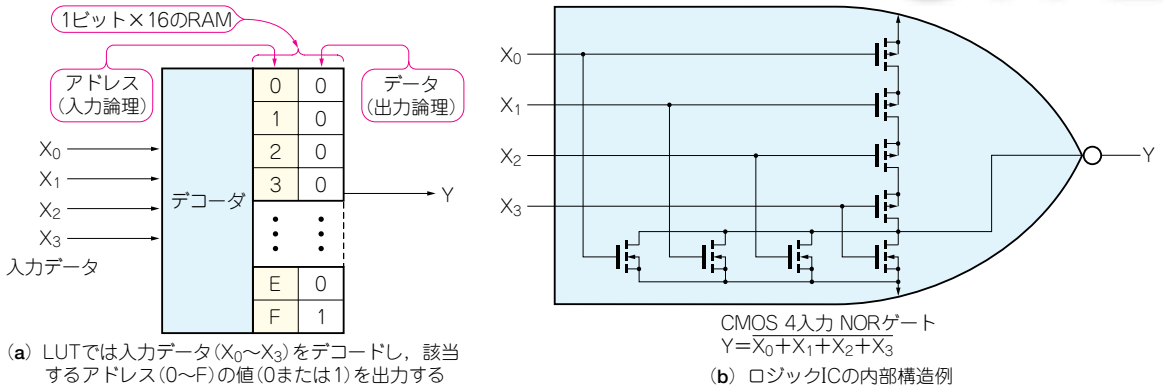


図2 4入力NORロジックのルックアップ・テーブルとロジックICの比較

を構成できるように互いに配線すること」です。

FPGAにはいくつか種類があり、それぞれコンフィグレーション・データを保持する方法が大きく異なります。FPGAには大きく分けて次のような種類があります。

▶ **SRAM型**

CPUなどで使われるSRAMと同じ技術でコンフィグレーション・データを保持するFPGAです。現在最も多く使われる方法であり、大規模化・高速化ともしやすいという大きなメリットがあります。

反面、電源を切るとコンフィグレーション・データが消えてしまいます。

▶ **ヒューズ型**

ヒューズ方式は回路の接点がアンチヒューズ(通常は絶縁状態にあり、書き込み電圧を加えることにより、接続状態になるヒューズ)素子で構成され、ヒューズを焼き切ることで回路を形成します。

アンチヒューズ素子は1回だけプログラム可能な記

憶素子で、SRAM素子よりコンパクトなため高集積が可能です。低抵抗で高速に動作できるメリットがあります。

デメリットは、ヒューズなので一度書く(切る)と書き換えられず、何度も繰り返して使うことが不可能なことです。開発上のコストを低下させにくく、あまり普及していません。

▶ **フラッシュ型**

フラッシュ・メモリ技術を使用したFPGAです。書き換え可能というSRAM方式の利点と、不揮発性というアンチヒューズFPGAのメリットを兼ね備えています。

しかし、その構造と製造プロセスの複雑さから、集積度を引き上げること、価格を下げるのが困難なため、こちらもあまり普及していません。

今回紹介するのは、SRAM型です。単にFPGAというところのタイプを示すことが多いようです。種類も非常に多くあり、価格も手頃で入手も容易です。

表1 メモリ・デバイスの種類と特徴

種類	長所	短所
SRAM	<ul style="list-style-type: none"> ● メモリの中でもっとも高速(数 ns) ● リフレッシュが不要で、駆動回路が簡単 ● 待機電力が小さい(バッテリー・バックアップが容易) 	<ul style="list-style-type: none"> ● 価格が高い ● メモリを構成する回路が複雑で大容量化が困難 (~64 Mビット程度)
	<ul style="list-style-type: none"> ● バックアップが必要な用途 ● 計測 / 画像などの超高速用途 ● 小規模な組み込み機器 	
DRAM	<ul style="list-style-type: none"> ● 大容量 / 低価格 	● リフレッシュ動作が必要で駆動回路が複雑
	<ul style="list-style-type: none"> ● パソコンなど、大容量を必要とする用途 ● 最近では組み込み用途でも主流となってきた 	
PROM (EPROM, EEPROM)	<ul style="list-style-type: none"> ● フィールドで容易に書き込み可能 	● 書き込みにはROMライター、消去にはROMイレーサという専用のツールが必要
	<ul style="list-style-type: none"> ● 従来は組み込み用のプログラム格納デバイスの主流であったが、フラッシュに取って代わり、あまり使われなくなった 	
フラッシュ・メモリ (フラッシュROM)	<ul style="list-style-type: none"> ● オンボードで書き込み / 消去が可能 ● 部分的な書き込み / 消去が可能 	● 書き換え回数は有限である
	<ul style="list-style-type: none"> ● ほとんどのデジタル機器のプログラム格納デバイス ● ストレージ・デバイス(SDメモリーカードなどもこの種類) 	