

# CPU のプログラムと FPGA のコンフィグレーション・データを一つのメモリに格納

## 第3章

# NOR 型フラッシュ ROM と CPLD を使った FPGA ブート・システムの設計事例

FPGA を搭載したシステムでは、FPGA のコンフィグレーション・データを格納する ROM が必要になる。一般にシリアル ROM が使われることが多いが、この ROM ではランダム・アクセスが行えないため、マイコンのプログラムを格納して直接実行できない。そのため、シリアル ROM と NOR 型フラッシュ ROM の両方を搭載するシステムが多い。ここでは、FPGA 起動用シリアル ROM を搭載せず、通常の NOR 型フラッシュ ROM を使いながら、FPGA とマイコンの両方のブートが可能なシステムの設計事例を紹介する。

(編集部)

山武 一郎

### 1 FPGA 起動用 ROM と マイコン起動用 ROM を一つに

第1章で解説したように、電源を切っても内容を保持するメモリを ROM と呼びます。電源 ON で起動する組み込みシステムでは、必ず ROM が必要になります。ROM にもいくつかの種類があり、適材適所で使い分けることは既に解説しました。

しかし、コストを抑えるには、できるだけ部品の個数を減らしたいところです。場合によって種類の異なる ROM を使い分ける必要性は理解できますが、なんとか兼用できないでしょうか。

#### ● FPGA 起動用シリアル ROM

FPGA を搭載したシステムでは、ROM に FPGA のコンフィグレーション・データを格納し、電源投入と同時に

FPGA が自動的に ROM の内容を読み出し、FPGA のコンフィグレーションが行われて起動します。このときの ROM は外付けになるため、FPGA との接続には何本かの I/O ピンが必要になります。このコンフィグレーション用の I/O ピンは、一般には通常の I/O ピンとして使用できません。そのため、ユーザが使える I/O ピン数をできるだけ多く確保するには、コンフィグレーション用の I/O ピンが少ないことが望まれます。また、コンフィグレーション用のデータは、先頭から順番に読み出せばよいので、アドレス・バスも不要です。以上のことから、FPGA 用のコンフィグレーション ROM として、一般にシリアル ROM が使われます〔図 1(a)〕。

FPGA 用の ROM としては、以前は 1 回だけ書き込みが可能なワンタイム ROM も使われましたが、現在では電氣的に消去/書き込みが可能なフラッシュ ROM タイプのものが一般的に使われています。

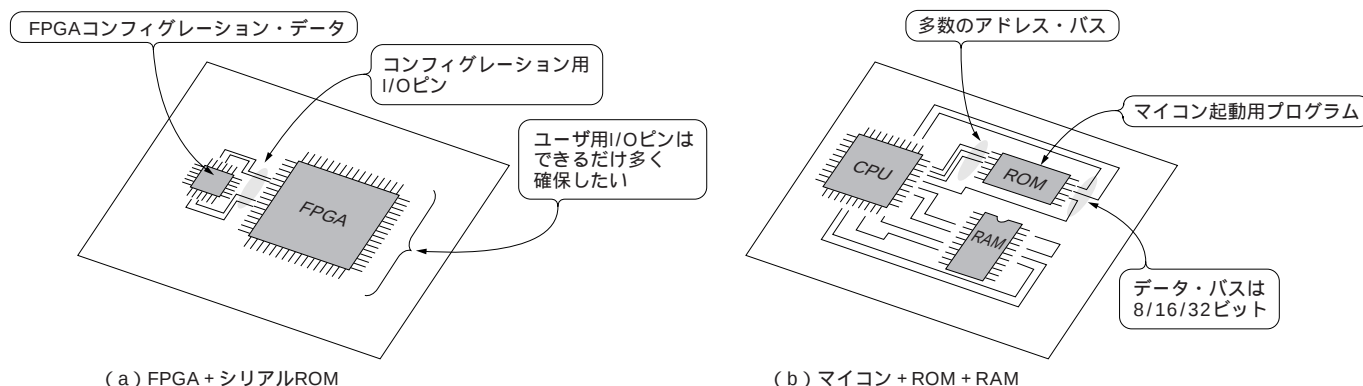


図1 FPGA およびマイコンの起動用 ROM

- マイコン起動用プログラムの格納には  
NOR 型フラッシュ ROM

マイコンが電源 ON で動作を開始するときも ROM が必要です。一般的にプログラムはアドレスを順番にカウント・アップする方向に実行されますが、途中でジャンプ命令などの分岐があるとアドレスは連続しくなくなります。また、命令だけでなくデータを保持する目的で ROM を使う場合もあります。そのため、マイコン・プログラムの格納用 ROM には、バイト単位でランダム・アクセスできることが求められます〔図 1(b)〕。

昔はマイコン用の ROM として、紫外線で消去し、専用の書き込み器で書き込む UV-EPROM が多用されていました。現在ではマイコン・ボード上に実装したままでプログラムによる書き換えも可能な、NOR 型フラッシュ ROM に置き換わっています。

この FPGA 起動用のシリアル ROM とマイコン・プログラム格納用の NOR 型フラッシュ ROM ですが、なんとか片方を省略することはできないのでしょうか。

- ランダム・アクセスできる NOR 型フラッシュに  
FPGA 用コンフィグレーション・データも格納

この二つの ROM のうち、シリアル ROM にはランダム・アクセスできないという欠点があります。ランダム・アクセスできないシリアル ROM をランダム・アクセス可能なデバイスに見せかけるには、メモリ空間サイズ分だけのバッファ・メモリを用意するか、小容量のバッファ・メモリとアドレス比較回路を駆使してキャッシュ・システムを実現する必要があります。しかし、いずれにしても回路規模が大きくなり、現実的ではありません。

逆に、ランダム・アクセスが可能な NOR 型フラッシュ ROM をシリアル ROM に見せかけることは比較的容易で、アドレス・カウンタやパラレル-シリアル変換回路を外付けするだけで済みます。

そこで、NOR 型フラッシュ ROM にマイコン・プログラムと FPGA のコンフィグレーション・データの両方を格納し、FPGA およびマイコンを起動できるシステムの構築事例を紹介します。

## 2 FPGA コンフィグレーションの各種方法

- JTAG 経由のダウンロードは比較的高度な制御が必要

FPGA をコンフィグレーションする方法として、現在では JTAG によるダウンロード方法が多用されています。JTAG によるダウンロード方法は、一般にシステムの開発時に FPGA 開発用パソコンからコンフィグレーション・データをダウンロードするときに使われる方法です。そのため、この手順をそのまま電源 ON 起動時に適用しようとすると、JTAG プロトコルにのっとったシステムの高度な制御が必要になります。

シリアル ROM を使った FPGA のコンフィグレーション方法は、JTAG による制御ではなく、次に説明する、より簡易的な方法で実現されています。

- マスタ/スレーブおよびシリアル/パラレル

表 1 に、FPGA の JTAG 以外のコンフィグレーション方法の一般的な分類を示します。クロックを FPGA が出力する場合をマスタ、外部回路で生成して FPGA に入力する場合をスレーブ、そしてコンフィグレーション・データを伝送するデータ線が 1 本の場合をシリアル、複数本(一般的には 8 本)の場合をパラレルとして分類します。表 1 の呼び方は一般的な名称で、正確には FPGA ベンダ各社でそれぞれ異なった名称があります。

- Spartan-3 の起動手順

以降では具体的な FPGA の起動手順として、Spartan-3

表 1 JTAG 以外の FPGA コンフィグレーション方法

| クロック入出力  | データ線 | 名 称       |
|----------|------|-----------|
| FPGA が出力 | パラレル | マスタ・パラレル  |
|          | シリアル | マスタ・シリアル  |
| FPGA が入力 | パラレル | スレーブ・パラレル |
|          | シリアル | スレーブ・シリアル |

表 2 Spartan-3 の FPGA コンフィグレーション・モード

| コンフィグレーション・モード | モード設定ピン |       |       | クロック    | データ幅 | DOUT |
|----------------|---------|-------|-------|---------|------|------|
|                | M0      | M1    | M2    |         |      |      |
| マスタ・シリアル       | " L "   | " L " | " L " | CCLK 出力 | 1    | 出力   |
| スレーブ・シリアル      | " H "   | " H " | " H " | CCLK 入力 | 1    | 出力   |
| マスタ・パラレル       | " H "   | " H " | " L " | CCLK 出力 | 8    | なし   |
| スレーブ・パラレル      | " L "   | " H " | " H " | CCLK 入力 | 8    | なし   |
| JTAG           | " H "   | " L " | " L " | TCK 入力  | 1    | なし   |