

特集1 動かないLSIを動かす方法

第4章

FPGA内蔵マイクロプロセッサを機能検証に使う

——UARTコアとDDRメモリ・コントローラの検証を例に

田中良平

ここでは、ソフト・マクロのマイクロプロセッサを活用する論理検証手法を紹介します。この手法は実機による検証に適しています。また、シミュレーションで検証しては時間が足りない、あるいは仕様が固まっていないけれど設計を進めなければならないときに活用できます。
(筆者)

FPGAは、手元で回路を書き換えられるという特徴を持ちます。したがってなんらかの回路を設計すれば、その場で実機による動作を確認することができます。ソフト・マクロのマイクロプロセッサ(以降、「ソフト・プロセッサ」と呼ぶ)を活用する検証は、実機と併用することで強みを発揮します。シミュレーションでは複雑であったり、時間がかかりすぎると思われる場合に有効な手法となるでしょう。

ただし、どの検証手法にも言えることですが、その長所と短所を理解し、検証結果や目的を正しく理解していないと的確な判断を下せません。ここで紹介する方法についても、すべての場面で有効な方法とはいえません。やはり得意な場面と苦手な場面が存在しますから、その判断が重要です。

本稿の前半では、マイクロプロセッサを利用した検証手法を紹介します。後半では、その具体例としてUARTとDDRメモリ・コントローラの検証について紹介します。ソフト・プロセッサとしては、ローランの「AXAS」と米国Xilinx社の「MicroBlaze」を使用します。実際に読者のみなさんが自作したプロセッサや市販品などを利用して、ソフト・プロセッサを活用した検証を行う際の参考にいただければ幸いです。

FPGA用ソフト・プロセッサを使った検証とは？

FPGA向けの設計では、ASICの場合と異なり、実機を

動作させながら機能を検証するという選択肢があります。このため、検証手法のバリエーションが増えます。

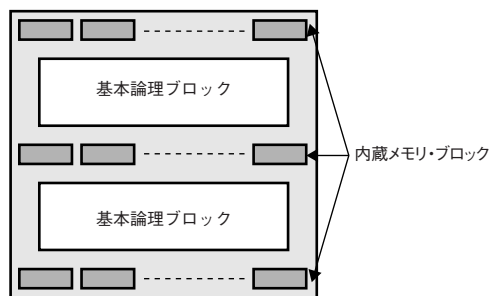
その一つに、ソフト・プロセッサを使用する検証手法があります。この手法を用いると、テスト信号の生成や出力信号の検証を、ソフト・プロセッサ上で動作するソフトウェアによって制御できるようになります。HDLシミュレータによる検証において、HDLでテストベンチを記述すると複雑になりすぎる場合でも、ソフトウェアで記述するとシンプルになることがあります。また、シミュレータでは時間のかかりすぎる機能の検証や長時間にわたる連続テストなどの際に、実機による検証が威力を発揮します(図1)。

ただし、この手法を有効活用するためには、適用可能な範囲を把握しなければなりません。ここでは、ソフト・プロセ



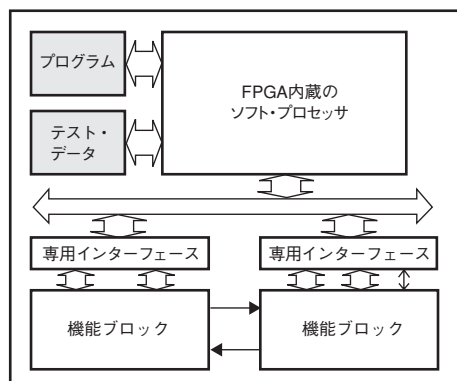
〔図1〕シミュレーションから実機テストまでをサポート

FPGAの検証では、ソフト・プロセッサはシミュレーションから実機テストまで、仕様変更を伴う現在進行形の開発に有効なツールとなる。



〔図2〕FPGA内蔵メモリ

SRAM型FPGAの多くは、基本論理ブロックとは別にメモリ・ブロックを内蔵している。



〔図4〕ソフト・プロセッサのプログラム・メモリとして活用

内蔵メモリ容量が少なくても、検証用のプログラムであれば十分に組み込むことができる。

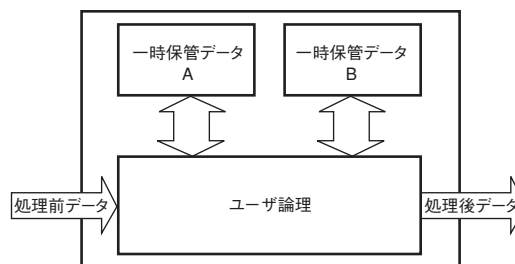
ッサを機能検証に使うための基本的な概念を解説します。

●内蔵メモリの活用

最近のFPGAでは、ユーザ論理の領域とは別に、内蔵メモリ・ブロック（エンベデッド・メモリ）が用意されていることがあたりまえになってきています（図2）。しかも、その容量は増える一方です。

単純に、外部バスとのタイミングを調整するためのバッファとして使用するのであれば、それほど多くのメモリ容量は必要となりません。しかし、デジタル・フィルタなどの演算を行う場合には、大容量の処理データを保存するメモリ領域が一時的に必要となります（図3）。最新のFPGAの仕様を見ると、乗算器などが搭載され、メモリ容量が増加しています。これはデジタル信号処理向けアプリケーションの要望にこたえた結果なのかと思うこともあります。

FPGAが内蔵するメモリの主な用途は、データを一時的に保存することだと思いますが、ソフト・プロセッサを制



〔図3〕信号処理用のデータ・バッファとして利用

FPGAを信号処理に使用する場合、処理過程において、多くのメモリ空間が要求される。外部のメモリを使用せず、内蔵メモリで処理できれば、高速処理が可能となるため、メモリ容量は増加し続けている。

御するプログラムを格納するために利用することもできます（図4）。先に述べたメモリの活用方法との大きな違いは、主体がプログラム・メモリにある点です。この使いかたでは、メモリに設計者の思想が格納されます。そして、その思想に基づいて論理回路が制御されます。処理結果を保存する、もしくはタイミング調整のために一時的にデータを保存する場合とは、メモリの利用方法がまったく異なります。

ソフト・プロセッサのプログラム格納用にメモリを使用するといった観点から見ると、現状のFPGAの内部で構成できる内蔵メモリの容量はやや小さいようです。組み込み分野でもリアルタイムOSが使われる時代です。本格的にCPUを使った制御機能を実現したければ、現在のところ、FPGAの外部に汎用CPU回路を構成したほうが安価に高機能を実現できます。

しかし、ソフト・プロセッサに担当させる機能を限定することでコンパクトなプログラムになり、必要なメモリ容量も小さく収めることができます。ソフト・プロセッサ活用のメリットを生かせるわけです。

限定された機能であれば、比較的低価格のFPGAであっても十分なプログラム容量を確保できます。FPGAベンダからもソフト・プロセッサが提供されており、今後、ソフト・プロセッサの活用はますます普及するでしょう。そしていずれは本格的なプログラムが組み込めるようになることでしょう。すでに、ハード・マクロのCPUコアが組み込まれたFPGAも登場するなど、FPGA内蔵プロセッサと関わる機会は増えてくると思います。今は「ソフト・プロセッサは使いものにならない」と思われている方も、機能検証にぜひ活用してみただけいただければと思います。