第1章 組み込みシステム開発評価キットにオープン・ソースのソフトCPUコアを実装する

MIPSソフト・マクロCPUコアPlasmaの実装事例

ここではオープン・ソースであるソフト・マクロのCPUコアとして，MIPS互換のPlasmaを実装してみる．FPGAのブロックRAMだけでなく，外部メモリのフラッシュROMやSDRAMの上に命令やデータを配置した場合でもCPUを効率良く動かせるように，キャッシュ・メモリやバースト転送対応マルチポートSDRAMコントローラなどを実装した．（編集部）

佐藤 達之

組み込みシステム開発評価キット（通称「BLANCA」）では，A/Vプロセッサに内蔵できるソフト・マクロのCPUコア（以下，ソフトCPUコア）としてMicroBlazeとM32Rをサポートしています．しかし本キットに同梱されているものには，開発環境が評価版であるなどの制限があります．

また，CPUのHDLソースは公開されておらず，ネットリストで提供されるソフトCPUコアとなります．本キットの趣旨を考えると，ソフトCPUコアもHDLで記述されたものを実装してみたくなります．

そこで今回は，既存のオープン・ソースのソフトCPUコアをBLANCAシステムへ実装してみました．

Ⅰ．Plasma基本実装編

1．設計に当たっての検討

● オープン・ソースの課題と問題点

オープン・ソースのCPUコアを利用するに際して，検討すべき事柄を幾つか挙げてみます．

1）ライセンスの確認

オープン・ソースで公開されているからといって，すべてがフリーというわけではありません．Public Domain，GPL，LGPLなど，フリーウェアとしてよく知られたライセンス方式もありますが，商用利用の禁止や断りなしの再配布を禁止するものも見かけます．

普通は添付ドキュメントかソース・リストの冒頭にライセンス方式が書いてあるので，用途によってはどのようなルールが適用されるのかをしっかり確認しておく必要があります．

2）工業所有権

ソフトCPUコアに限ったことではありませんが，フリーウェア・ライセンスをもつものが，それを使用する権利まで保障しているということではありません．例えばプロセッサにはさまざまな技術特許があります．採用したオープン・ソースのCPUコアがこれに抵触していた場合，特許使用料が利用者に請求される可能性も考えられます．

MIPSアーキテクチャでは，ワード・アラインメントの取れていないデータを二つの組み合わせ命令でロード/ストアできる特許（米国特許は2006年12月で期限切れ）について，過去にMIPS Technology社が著作権侵害の提訴を行ったことがあります．そのため，フリーのMIPSコアではこの特許に関連した命令を故意に実装しない例が見られます．

3）互換性と信頼性

オープン・ソースのCPUコアの多くは，プロセッサの学習や趣味を目的として作られていることが多く，十分な互換性テストが行われていない場合があります．特に実装例や歴史が浅いものには信頼性は期待できないので，「コアにバグがあったら自力で修正する！」といった覚悟が必要かもしれません．割り込み受け付けのタイミングなど，例外的なシーケンスには特に注意が必要です．

KEYWORD ── Plasma，MIPS，外部バス・インターフェース，シングル・アクセス・バス，バースト・アクセス・バス，SDRAMコントローラ，キャッシュ

表1　Plasmaの互換性とBLANCAシステムへ実装した際の対応方法

Plasmaコア(mlite_cpu)の仕様	Plasmaにおける対応	BLANCA版Plasmaにおける変更と対応
LWL，LWR，SWL，SWR命令が未実装	GCCの場合，通常のコードでは出力されないとして未対応	未実装命令を出力しないパッチが施されたYACC用GCCコンパイラを使用する
多くのコプロセッサ・レジスタが未実装	未対応	未対応
EPCレジスタには例外発生アドレスの次のアドレスが格納される	割り込みサービス内で，EPCレジスタから4減算することで復帰アドレスを取得する	左記に同じ
割り込み許可レジスタが，スタック構造になっていない	割り込み禁止状態でSYSCALL命令やBREAK命令は実行しないなど，割り込み禁止状態で例外割り込みを発生させない	Plasmaコアを変更して，MIPSに合わせた
リセット例外ベクタが0000_0000hである	Plasmaオリジナル仕様	コア外部から指定するようにして，MIPSと同じBFC0_0000hに変更した
一般例外ベクタが0000_003Chである	Plasmaオリジナル仕様	コア外部から指定するようにして，特殊レジスタで任意アドレスを設定可能にした
外部バスが，独自の単一クロック・サイクルである	Plasmaオリジナル仕様	CPU内部デバイスはPlasmaのバス仕様に合わせ，ブリッジ回路を経由してBLANCAシステム・バスとインターフェースした
リード・サイクルで有効なバイト・イネーブル信号がない	システムとして不要なので実装されていない	Plasmaコアを変更して，バイト・イネーブル信号を追加した
エンディアン指定はmem_ctrl.vhd内のENDIAN_MODEコンスタント値で指定する	ビッグ・エンディアン	ビッグ・エンディアン

4) 処理速度

ASIC用として設計された既存のCPUコアの動作に基づいて汎用HDLで記述されたソフトCPUコアは，設計段階からFPGAに特化したプリミティブ・セルを駆使するFPGAベンダ提供の専用ソフトCPUコアと比べると，回路規模および動作速度の両面で劣っているのが一般的です．これらの課題から，商用利用においてはFPGA上にオープン・ソースのソフトCPUコアを採用するメリットは少ないと思います．しかし，学習や実験，研究，個人利用においては大変興味深いテーマであると思います．

● CPUコアの選択

今回はバス周りの設計が主になるため，CPUコアは最小の作業で確実に動作することを第1条件としました．

参考文献(1)や(2)ではMIPS-I命令セットに基づくソフトCPUコアの回路規模や実行速度の評価がインターネット上などで公開されており，MIPSコアを選定する際には参考になります．

その中でPlasmaシステムはSteve Rhoads氏がOPENCORES.ORG(http://www.opencores.org/)で公開しているMIPSコアです．Plasmaのソース・リストやモジュール構成図，命令表などが公開ページから入手できます．今回はこれを使用させていただくことにしました．

表1にPlasmaの互換性に関する問題やBLANCAシステムへ実装する際の対応方法を示します．

選定段階では，参考文献(2)で公開されている同じMIPSコアのYACC(キャッシュ付きWishboneモデル)が最有力候補でした．キャッシュ・メモリにWishbone仕様の外部バスが採用され，筆者がなじみのあるVerilog HDLで記述されている点，そして著作者が日本人である点など，好条件でした．しかし，筆者がテストした時点では手元のFPGA上でうまく動作させることができず，今回は採用を見送りました．

なお，この記事の執筆中にOPENCORES.ORGで公開されている同じくMIPSコアであるuCoreがアップデートされたようで，MMUやキャッシュ・メモリ，Wishboneバスなどがサポートされたと記載されています．URLが間違っているのかリンクからはダウンロードはできませんでしたが，こちらも魅力的なコアになっているようです．

以下にPlasmaを選択した主なポイントを示します．

1) 外部バス・インターフェースが一本化されている

一般的なMMU/キャッシュをもつ32ビットRISCは**図1**(**a**)のようなバス構成ですが，Plasmaでは**図1**(**b**)のようになっており，MMUとキャッシュを省略した感じになります．CPUのローカル・バスは1本で，MMUとキャッシュをもつCPUとインターフェースは変わりません．

これに対して**図1**(**c**)は学習目的や命令実行ユニットの単体評価でよく見かける最小システムのバス構成です．命令バスとデータ・バスは切り離されたままなので，ローカ