第2章

# オープン・ソースのCPUコアの実力を試す

## 32ビットRISCプロセッサ「LatticeMico32」レビュー

山際伸一

**ここでは，無償で利用できるソフト・マクロのCPU「LatticeMico32」を取り上げる．HDL（Hardware Description Language）ソース・コードの形で提供され，FPGA（Field Programmable Gate Array）に限らず，ASIC（Application Specific Integrated Circuit）にも実装できる．オンチップ・バスとしてWISHBONEバス・インターフェースに対応している．ソフトウェア開発環境はGNUベースのツール群を利用する．（編集部）**

米国Lattice Semiconductor社は，オープン・ソースのCPUコア「LatticeMico32」[注1]を提供しています（p.26のコラム「LatticeMico32の入手と開発ツールのセットアップ」を参照）．LatticeMico32は，32ビット命令長/データ長のRISC（Reduced Instruction Set Computer）アーキテクチャのCPUコアです．32本の汎用レジスタを搭載し，外部割り込み機能やキャッシュ・メモリを用いることができます．さらに，ユーザが命令を定義できます．

本稿では，LatticeMico32のアーキテクチャについて解説した後，ハードウェアとソフトウェアの開発方法を説明します．例題は，UARTインターフェースとLEDを使って割り込みを確認する簡単なシステムです．

## 1．LatticeMico32のアーキテクチャ

LatticeMico32は，32ビット命令/データ長のRISCプロセッサ・コアです．アドレッシングには，ビッグ・エンディアンを採用しています．**図1**に機能ブロック図を示します[注2]．

本稿では，LatticeMico32のCPUコアとしての基本的な機能に主眼を置いて説明していきます．

### ● ハーバード・アーキテクチャとWISHBONEバス

LatticeMico32は，命令とデータのパスがそれぞれ独立しているハーバード・アーキテクチャを採ります．それぞれのバスも独立しており，WISHBONEインターフェース（p.27のコラム「WISHBONEインターフェース」を参照）で外部に接続されています．それぞれのバスに独立したメモリを接続することにより，命令フェッチとデータ・アクセスの衝突を回避できます．

命令バスから読まれた命令コードは，命令レジスタにラッチされ，デコードされます．デコード時に，その命令が用いる読み出し元データ・パスや，書き込み先データ・パス，演算器を選択し，命令を実行します．

演算器としては，加算器（減算もここで行われる）や論理演算器，シフタ，乗算・除算器が用意されています．シフタと乗算・除算器に関しては，複数サイクルで実行します．これらの演算器は，プロセッサの設定時に性能を選択できます．しかし，高速な演算器は多くのハードウェアを使う

注1：LatticeMico32のWebサイトは，http://jp.lscc.com/products/intellectualproperty/ipcores/mico32/index.html

注2：LatticeMico32開発システムのインストール先のフォルダにLatticeMico32プロセッサ・アーキテクチャに関するPDFファイルが保存されている．デフォルトのパスにインストールした場合は，C:\LatticeMico32\micosystem\components\lm32_top\document\lm32_archman.pdfがそのリファレンス・マニュアル．このドキュメントがLatticeMico32プロセッサを用いたシステム開発に必要な情報で最も詳しい資料である．

KeyWord　ソフト・マクロ，LatticeMico32，FPGA，ASIC，CPUコア，RISC，WISHBONE，ハーバード・アーキテクチャ，分岐予測，例外ハンドラ，Spartan-3E，Cyclone II

ので，性能と規模のトレードオフになります．演算結果は，それぞれの演算器の出力から一つ選択され，書き込み先に書き込まれていきます．

### ● キャッシュ・メモリを設定可能

LatticeMico32には，命令とデータに対してキャッシュ・メモリを追加できます．多くのメモリを利用する用途では，キャッシュ・メモリが効果を発揮します．しかし，ターゲットとなるFPGAが持つメモリ容量が小さかったり，キャッシュ・サイズ前後のメモリ空間しか利用しなかったりする場合は，意味がないかもしれません．性能と必要となるハードウェア・リソースのトレードオフを十分に検討する必要があります．

キャッシュ・メモリは，ライト・スルー・キャッシュで，

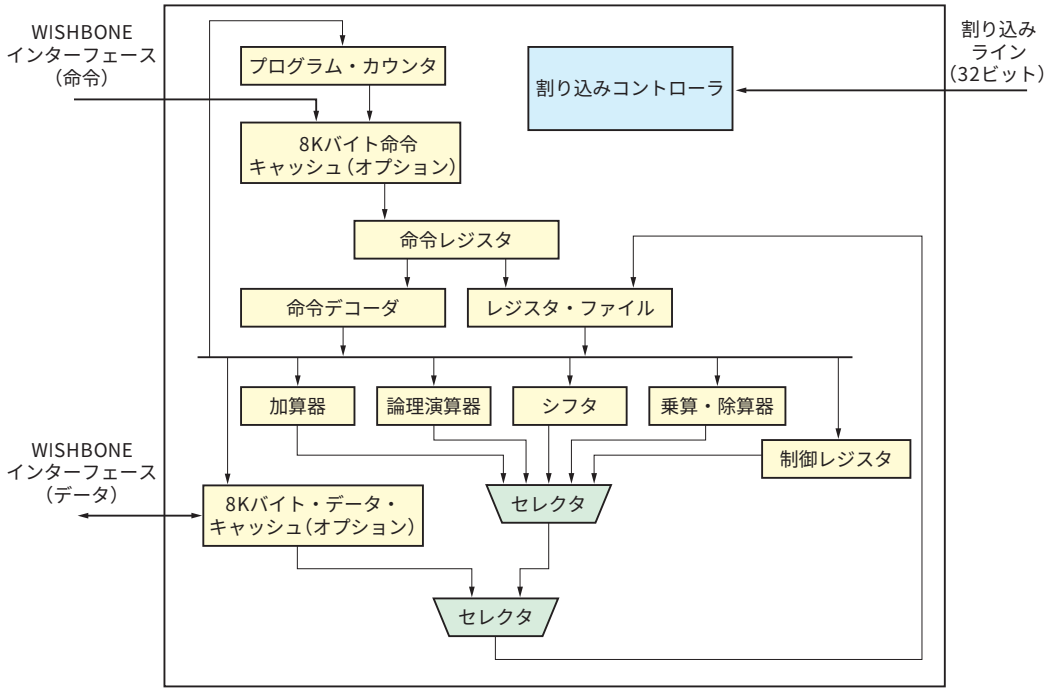


**図1**
**LatticeMico32の機能ブロック図**
32ビット命令/データ長のRISCプロセッサ・コアである．命令とデータのパスがそれぞれ独立しているハーバード・アーキテクチャを採る．オンチップ・バスはWISHBONE．

## コラム　LatticeMico32の入手と開発ツールのセットアップ

Lattice Semiconductor社のWebサイト（http://www.latticesemi.co.jp/）の「アカウント・インフォ」をクリックして開くページで，アカウントを作成しておきます．そして，「ラティスMico32」のページから「ラティスMico32システムをダウンロード」のリンクをクリックします．ライセンス条項に同意すると開発システムのインストーラをダウンロードできます．

インストールの際，GNU-based Compiler Toolsのチェックを外しておいてください．今回はCygwinで開発するのでGNUツールのインストールは必要ありません．

デフォルトのパスにLatticeMico32開発システムをインストールした場合，C:\LatticeMico32\micosystem\componentsにハードウェアのソース・コードがインストールされます．プロセッサはlm32_topフォルダにあります．関連するハードウェアはすべてVerilog HDLで記述されています．

ソフトウェア開発ツールについては，前述の「ラティスMico32」のページにある「Downloadable Software」のリンクをクリックし，表示されたツールの一覧からLatticeMico32 GNU-Tools Source Codeをダウンロードします．Mico32プロセッサのソフトウェアは，GCCで開発できます．

本稿では，バージョン6.1.1のGNUツールを用います．ソース・コードで配布されるので，Cygwin向けにコンパイル済みのものを本誌CD-ROMに掲載します．http://www.cygwin.com/を参照してCygwin環境をセットアップしておいてください．

付属CD-ROMに収録のコンパイル済みツール・チェーンは，Cygwinコンソールから以下の手順で/usr/localディレクトリに展開します．

```
$ cd /usr/local
$ tar jxvf lm32-tools.tar.bz2
```

これで，/usr/local/lm32-tools/binにgccなどのコマンドが展開されます．