

インターネット電話コーデック処理を16ビットRISCで実現

C言語プログラムから カスタム・プロセッサを生成する設計手法

大村正之/青柳栄/杉浦義英

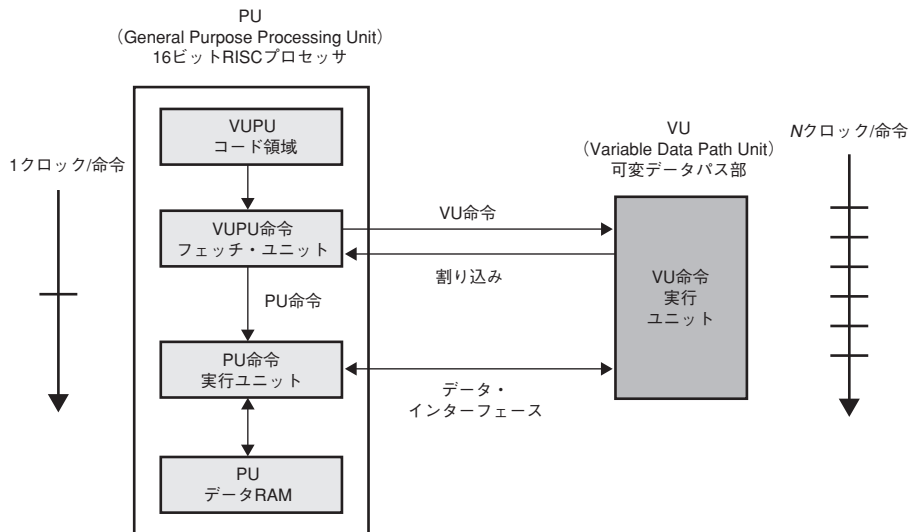
筆者ら(パシフィック・デザイン)は、任意のユーザ命令を組み込めるコンフィギャラブル・プロセッサ[VUPU]を開発している。VUPUは可変データパス部と16ビットRISCプロセッサ部に分かれている。VUPUとビヘイビア合成ツールを組み合わせることにより、ユーザはC言語プログラムによって可変データパス部を自由に書き換えられる。インターネット電話のコーデック処理G.729のC言語プログラムからカスタム・プロセッサを設計した事例を紹介する。(編集部)

筆者ら(パシフィック・デザイン)が開発したコンフィギャラブル・プロセッサ[VUPU]は、ビヘイビア合成ツールと組み合わせることにより、与えられたC言語プログラムから、そのC言語プログラムを実行する専用プロセッサを生成することができます。与えられたC言語をシリコンにマッピングするという意味で、筆者らはこの手法を「C-to-Silicon」と呼んでいます。本解説では、C-to-Siliconの概要

を簡単に説明してから、これをITU-T(International Telecommunication Union Telecommunication Standardization Sector)の規格であるG.729インターネット電話コーデックに適用した例を紹介します。

四つのキー・テクノロジー

今回のC-to-Siliconの実現には、四つのキー・テクノロジーが必要となります。まず、C言語を実装するターゲットとなるコンフィギャラブル・プロセッサ[VUPU]です。次にC言語記述中のデータパス部をRTLデータに変換するツールが必要です。ここでは米国Synopsys社のCoCentric SystemC Compiler(以下、SystemC Compiler)を使いました。3番目が命令セット・シミュレータ(ISS)です。これはRTLのHDLシミュレーションの代わりとなるサイクル・ベースのシミュレータです。以上の三つを利用してC



【図1】

VUPUアーキテクチャ

VUPUの構造を示す。PUは16ビットのRISCプロセッサである。VUはクロック制限のないデータパス部である。PU命令とVU命令はコード部に格納されていて、共通のフェッチ部によりフェッチされてPU実行ユニットとVU実行ユニットに振り分けられる。VU命令は割り込みにより戻る。両者はmove命令などのデータ転送機構を持っている。

言語プログラムを一つのVUPUにマッピングします。加えて、一つのC言語プログラムを二つのVUPUにマッピングして同時実行させる場合は、プロセッサ間通信機構IVC (Inter VUPU Communication)が必要となります。これが四つめのキー・テクノロジーとなります。

●VUPUとは？

図1にVUPUのアーキテクチャを示します。VUPUは、任意のユーザ命令を追加できるようにしたプロセッサです。ここでのユーザ命令とは、データパス部の処理(積和演算や、積和演算を含むループ処理など)を想定しています。

VUPUの「VU」は、可変データパス部 (Variable Data Path Unit)の意味です。VUは、主としてC言語プログラム中の信号処理などを受け持つハードウェア実行ユニットです。このVU部をユーザが自由に設計することにより、専用データパス回路を持つカスタム・プロセッサができていきます。

VUPUの「PU」は、16ビットRISC構成をとる汎用プロセッサ部 (General Purpose Processing Unit)の意味です。PUは、主としてC言語プログラム中の制御記述を受け持つ部分です。PU部のRTL記述は、IPコアとして提供されています。

VUPUの命令は1ワード長24ビットで、2ワード長48ビットまで拡張可能です。ユーザは、最大48ビット長の範囲でVUの命令フォーマットを定義できます。この2ワード長48ビット命令がVUPUの一つの特徴です。英国ARM社のARMコアは、1ワード長が16ビットと32ビットの2種類の命令長を持ちます。また、米国Tensilica社のXtensaは1ワード長が16ビット、24ビットとやはり2種類の命令長を持っています。それに対して、VUPUは1ワード長が24ビットで、2ワード長までの命令拡張が可能です。これは次のバージョンでLIW (並列命令)動作させることを考えているためです。すなわち、VUPUは組み込み型の小さなLIW並列プロセッサを意識して開発しています。

PUにはいくつかの品種があり、現在、P3(動作周波数140MHz、26,000ゲート)とP0(動作周波数160MHz、16,000ゲート)を出荷しています。2002年の後半には、32ビットRISC型のP4を出荷する予定です。

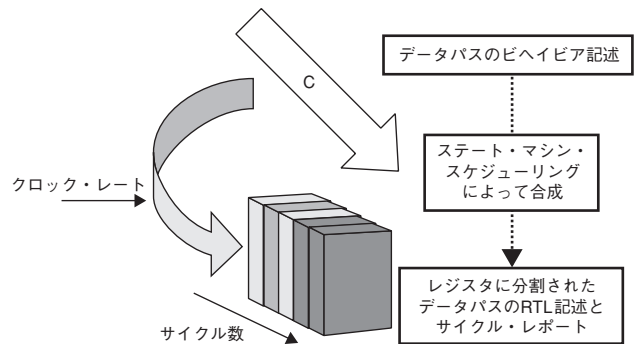
PUは、RISCなので基本的に1命令1サイクル実行です。一方、VUにはサイクル数の制限を設けていません。これはVU部にC言語のforループくらいの処理をあてはめた

いからで、それくらいの規模の専用命令を追加しないと専用プロセッサを開発する意味がないからです。このVU部のRTL記述をC言語プログラムから自動で生成するツールがSystemC Compilerです。

●ビヘイビア合成ツールでVUを生成

図2にビヘイビア合成ツール「SystemC Compiler」の処理概要を示します。ビヘイビア合成ツールとは、C言語のような高級言語を入力し、指定されたクロック・レートに応じてスケジューリングとリソース・アロケーションを行い、結果のRTLデータを出力するものです。この際、スケジューリングに使用する状態管理には、FSM (finite state machine)を使用するのが一般的です。このようなビヘイビア合成ツールは、データパス部の合成に有効です。すなわち、C言語で信号処理などのデータパス部の記述を行い、指定されたクロック・レートに応じてスケジューリングされたRTLを出力するので、クロック・レートに合わせたレジスタ段数の最適化を容易に行えます。このとき、ビヘイビア合成ツールからは、合成されたRTLのサイクル数がレポートされるのが一般的です。このサイクル数のレポートが次の命令セット・レベルのシミュレーションで重要となります。現在、この種のツールは数社から発売されていますが、筆者らはSystemC Compilerを利用して開発環境の構築を図っています。

SystemC Compilerの入力はSystemCです。筆者らのVUPUの入力はANSI Cです。したがって、ANSI CをSystemCに見立てる必要がありますが、これはラッパ



〔図2〕ビヘイビア合成ツール

ビヘイビア合成ツールの一般構成を示す。SystemC Compilerも基本的にこの構造をとっている。入力にC言語プログラムであり、パラメータとしてクロックを指定すると、指定されたクロック・レートでスケジューリングされたデータパスのRTL記述が出てくる。その際、スケジューリングされたサイクル数がレポートされ、これがきわめて重要な情報となる。