

新世代CPUコア RXファミリ誕生

～32ビットCPUの演算性能を16ビットCPUクラスのコード・サイズで実現～

組み込み機器の開発エンジニアの方々には、昨今の複雑化する制御技術への対応、開発期間の短縮、厳しいコスト要求への対応、さらに開発者の不足、省エネルギー化などの要望があります。これらの要望に答えるべく開発されたRXファミリを紹介します。

大谷 寿賀子, 藤澤 幸穂, 仲山 和彦

1. RXファミリの特徴

ルネサス テクノロジーの新しいCPUコア, 「RXファミリ」について紹介します。

RISC (Reduced Instruction Set Computer) アーキテクチャのプロセッサが多い中, RXマイコンはCISC (Complex Instruction Set Computer) とRISCの長所を融合した新しいアーキテクチャを採用し, コード・サイズの効率化と低消費電力を実現しています。フィルタリングなどの演算に対応するDSP機能命令や, 浮動小数点演算処理 (FPU) も搭載しています。これらによってシステム制御と高精度演算を1チップで実現し, コストの大幅な低減, 開発期間の短縮, システムの効率化に対応します。また, CPUコアは32ビットCPUでありながら16ビットCPUのコード・サイズで1.65DMIPS/MHzと高い演算性能を実現します。以下に, RXファミリの特徴を挙げていきます。

● CISCとRISCの長所を融合したアーキテクチャ

RXファミリはCISCの長所を活かし, 高いコード効率を実現しました。コード効率が高ければメモリ・サイズの小さなマイコンで同じ処理を実現できます。CISCのバイト可変長命令などへRISCのハード・アーキテクチャ, 5段パイプラインなどを備えています。

● ベンチマークによる徹底的な調査分析

異なるアーキテクチャを融合するに当たり, 民生機器や産業

用機器, 自動車などの各種アプリケーション・ソフトによるベンチマーク評価を行いました。そしてコード効率とパフォーマンス向上の最適解を見だし, CPUコアに適用しています (図1)。一つの最適解としてレジスタ・セットを紹介します。ベンチマーク・テストにより以下の評価が得られました。

- 演算系や制御系アプリケーション汎用レジスタを採用する方が性能的に有利
- レジスタが8本ではレジスタの退避復帰処理が頻出するためパフォーマンスが低下し, コード・サイズが増加する
- レジスタの本数が増えると, ハードウェア量および命令コード内の指定ビット数が増加する

以上を考慮し, 32ビット汎用レジスタ方式の採用と, レジスタの本数を16本と決定しました。

● 高コード効率・高性能を実現する命令セット

プログラムのコード・サイズを圧縮するために, ここでもベンチマークによる検討を行いました (図2)。高頻出の命令とアドレッシング・モードを抽出して短縮フォーマットへ割り付け, 命令数とコード・サイズの削減をしました。命令セットはバイト単位の可変長命令を採用しています。命令長は1～8バイトとなりますが, 高頻出の命令は1～4バイトの間に割り付けられています (図3)。基本演算命令 (AND, OR, ADD, SUB, MUL 命令) については2オペランドだけでなく, 3オペランド・フォーマットも用意されます。

テーブル操作で効果を発揮するインデックス付きレジスタ間

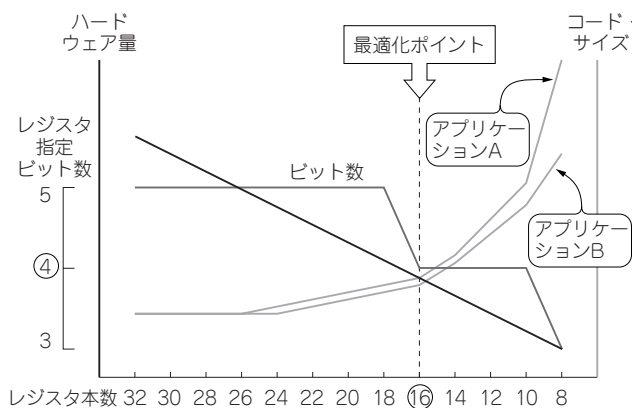


図1 アプリケーションによるベンチマーク

命令頻度解析

| | | | | | |
|-----------|-------------|-----------|------------------------|----------|-------------|
| 転送 31% | 条件分岐 15% | 比較 11% | サブ ルーチン 分岐 8% | 加算 6% | そのほか 29% |
|-----------|-------------|-----------|------------------------|----------|-------------|

命令頻度を基に強化

| | |
|---|--|
| <ul style="list-style-type: none"> ● 転送命令を2バイト短縮 (MOV: イミディエイト値⇒メモリ, メモリ⇒レジスタ) ● インデックス付きレジスタ間接アドレッシングの採用 ● ポスト・インクリメント付き転送命令, プリデクリメント付き転送命令の採用 | <ul style="list-style-type: none"> ● 比較命令を2バイト短縮 (CMP: メモリ⇄レジスタ) ● サブルーチン分岐命令を1バイト短縮 (BSR) ● 加算命令を2バイト短縮 (ADD: メモリ+レジスタ, イミディエイト値+レジスタ) ● 3オペランド・フォーマットの採用 |
| <ul style="list-style-type: none"> ● 条件付き分岐命令を1バイト短縮 (BEQ, BNE) | |

図2 命令頻度の解析