

ARMアーキテクチャの基礎知識

中森 章

特集を始めるにあたり、まずはARMプロセッサとはどのようなCPUアーキテクチャなのかを詳しく解説する。ARMプロセッサにはいくつかのアーキテクチャとCPUコアの種類がある。また、ほとんどの命令に実行する/しないを決める条件フラグが付いていたり、ロード/ストア命令に豊富なアドレッシング・モードがあるなど、ARMアーキテクチャの特徴的な命令セットについても解説する。

(編集部)

1. ARMとは

ARMは会社の名称であるとともに、ARM社が設計し、ライセンス供給を行っているプロセッサの総称でもあります。命令セット・アーキテクチャ、それを実装するCPUコア、CPUコアにキャッシュ、MMU(メモリ管理ユニット)、バス・インターフェースなどを付加したプロセッサIP、プロセッサIPと周辺回路を集積したARMアーキテクチャの半導体チップを指します。

ARM社のプロセッサは、Intel社やAMD社などのCPUメーカーとは異なり、単体のチップがそのまま発売されることはありません。ARM社からIP(設計情報)の提供を受けた半導体企業(シリコン・パートナー)が、それを使って独自の機能や技術を加えて、汎用マイクロプロセッサやASIC、ASSPなどのSoC

(System on a Chip)を開発・販売します。これが、携帯電話やデジタル家電のアプリケーション・プロセッサとして広く使用されています。

よって、読者の方々がARMプロセッサを使ってシステムを開発しようと思い、CPUを買うためにARM社に問い合わせても、デバイスは売ってもらえません。ARM社からライセンスを受けた各半導体ベンダの中から、汎用品として市販されているデバイスを購入するか、いずれかのベンダにASIC開発を依頼することになります。

本特集では前者のパターンを想定し、第2章から第6章のそれぞれで、半導体ベンダが汎用品として市販しているCPUを取り上げ、その使いかたを解説しています。

2. ARMプロセッサのアーキテクチャ概要

ARMアーキテクチャの変遷

ARMプロセッサの基本的なアーキテクチャや命令セットは、初期のものからほとんど変更されていません。しかし、ARMアーキテクチャと命令セットは、もともと1980年代中頃に開発されたため、現在や将来の組み込みシステムに必要な機能(16ビット・モードやJavaアクセラレータなど)が何世代にも渡って追加されてきました。

ARMではアーキテクチャのバージョンを示すのに、v4やv5と表現します。さらに“T”はThumb、“E”はDSP拡張(Enhanced DSP)、“J”はJava拡張(Jazelle)の機能を示し、v4Tとかv5TEJなどと表示します。また、TrustZoneを内蔵するv6ZやThumb-2を内蔵するv6T2というアーキテクチャもあります。なお現在では、v4T以降のアーキテクチャのみが有効です。

さらにARMでは、CPUコアに番号を付けて呼んでいます。ARM1~3はARM社設立以前に前身のAcorn社で設計されたコアで、ARM社設立後最初に設計されたコアがARM6です(4と5は番号が飛ばされた)。そしてARM6の後継として開発さ

C O L U M N 1

ASIC, ASSP, ASCPの違い

コラムのタイトルにある英文字の略語、これらはすべてシステムLSIの構成を表すことばです。ASIC(Application Specific Integrated Circuit)とは特定用途向けのICのことです。しかし、その定義はいまいなまま使用されることが多くあります。ASICを広義に解釈すると、特定用途向けに特化した機能を有するICのすべてを含みます。つまり、ASSP(Application Specific Standard Product)やASCP(Application Specific Customer Product)もASICの一部です。ASSPとは特定用途向けの標準品、ASCPとは特定用途向けのカスタム・チップです。ASSPとASCPの違いは一般売りをするか否かの違いでしかありません。また、狭義では、ASICは顧客の要求を受けて開発する専用ICを指すことが一般的です。ゲートアレイ製品やPLD(Programmable Logic Device)がその代表です。その意味で、ASCPとASICは同一視されることが多々あります。

表1 ARM アーキテクチャの変遷

v1	最初の命令セット．ほとんど使用されていない
v2	乗算命令とコプロセッサをサポート
v2a	キャッシュのサポートと同期命令(SWP)の追加
v3	ARM 社独立後の最初のアーキテクチャ
v3G	詳細不明．v2a と互換性なし
v3M	結果が 64 ビットの乗算
v4	システム・モードのサポート．アーキテクチャの完成版
v4T	Thumb モードの追加
v5T	BLX, CLZ, BKPT 命令の追加
v5TE	DSP 命令セットの追加
v5TEJ	Java 拡張(Jazelle)の追加
v6	マルチメディア拡張(SIMD)．同期命令の強化(LDREX/STREX)．割り込み応答の高速化
v6T2	Thumb-2 モードの追加
v6Z	TrustZone の追加
v7	最新アーキテクチャ(詳細は第 7 章を参照)

(a) アーキテクチャ

ARM1 ~ 3	Acorn 社時代に設計されたコア (現在は使われていない)
ARM4 ~ 5	存在しない
ARM6	ARM 社として最初に設計されたコア (現在は使われていない)
ARM7	現在の ARM プロセッサの基本コア
ARM8	(現在は使われていない)
ARM9	5 段パイプラインへ拡張
ARM10	6 段パイプラインへ拡張
ARM11	8 段パイプラインへ拡張

(b) コア・ファミリ

れたのが、現在の ARM の基本コアともいえる ARM7 です。以降、コアの性能向上を目指し、新しいバージョンのコアが設計されています。

コアとアーキテクチャの対応は、基本的に次のようになっています。ARM6 と前期の ARM7 が v3、後期の ARM7 ~ ARM9 が v4、ARM10 が v5、ARM11 が v6 です。ARM7 も最新の Rev.4 では Thumb を内蔵し v4T になっています。流れとしては新しいバージョンのコアほど、新しいバージョンのアーキテクチャが使われています。ただし一部には、ARM7 や ARM9 のコアでも v5 アーキテクチャを採用している CPU もあります。

表1に ARM アーキテクチャの変遷を示します。

ARM アーキテクチャの実装

最初の ARM アーキテクチャのプロセッサが開発された当時、RISC は Stanford University の MIPS と、University of California, Berkeley(UC-Berkeley)の RISC (SPARC の母体)しか例がありませんでした。ARM が Berkeley RISC を参考にして設計されたのは周知の事実です。設計目標は、CISC ライクな命令セットを、RISC に準じた単純なハードウェアで実行することに置いています。その基本はパイプライン処理です。

ARM の CPU コアでは、ARM1 ~ 7、ARM9(ARM8)、ARM10、ARM11 でパイプラインの構成に若干の差異があります。

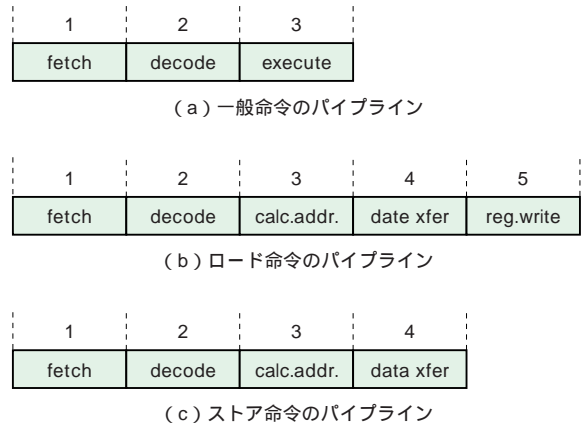


図1 ARM7 のパイプライン構成

表2 パイプライン・ステージの意味

fetch	命令フェッチ	メモリから命令をフェッチし、パイプラインに投入する
decode	命令デコード	命令をデコードし、データパスの制御信号を生成する
execute	実行	命令のデコード結果に従い、レジスタ・ファイルを読み、オペランドを(必要なら)シフトし、演算を行い、結果をレジスタにライトする
calc. addr.	アドレス計算	ロード/ストア/分岐命令の場合は演算器を用いてオペランドまたは分岐先のアドレスを計算する。ストア命令ではさらにメモリにストアするレジスタを読みする
data xfer	データ転送	ロード命令ではメモリからオペランド・データをリードする。ストア命令ではレジスタの値をメモリにライトする
reg. write	レジスタ・ライト	ロード命令でメモリからリードしたデータをレジスタにライトする

それぞれのコアの特徴とパイプラインを簡単に見ていきましょう。

▶ ARM7 コアの特徴

ARM7 は、コアの実装面積が小さいため、コストと消費電力が重視されるアプリケーション分野を対象にしています。近年では Thumb 命令セットをサポートするようになり、コード・サイズが重視される分野における採用も期待されています。

図1(a)に ARM1 ~ ARM7 のパイプラインを示します。その基本は単純な 3 ステージ構成のパイプラインです。ただし、ロード/ストア/分岐命令などはマルチサイクル命令として別のパイプライン処理を行います。それを図1(b)と図1(c)に示します。表2に各パイプライン・ステージの意味を示します。

▶ ARM9(ARM8)コアの特徴

ARM9 コアは、中～高性能レンジの CPU コア製品群です。ARM9 製品はすべて Thumb 命令セットを搭載したカスタム設計のハード・マクロセルです。論理合成可能なコアとしては ARM9E があります。ARM9E では、Thumb および ARM DSP