

ISBN978-4-7898-4549-6

C3055 ¥2800E

CQ出版社

定価：本体2,800円（税別）



9784789845496



1923055028005



vol.2

組込みプロセッサ技術

枝廣 正人 編著

CQ出版社



組込みシステム 基礎技術全集 vol.2

組込み プロセッサ技術

情報処理学会 組込みシステム研究会 監修

枝廣 正人 編著 / 黒田 一朗 共著 / 中森 章 協力

CQ出版社



組込みシステム
基礎技術全集

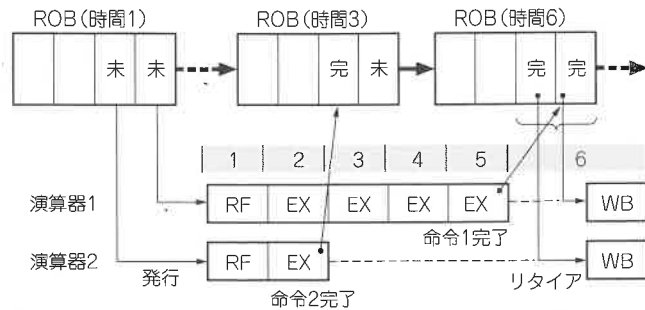


図3.13 インオーダ完了(2ウェイ)

る方法がとられる。このバッファをリオーダバッファ (ROB: Re-order Buffer) という (図3.13)。完了した命令は結果をリオーダバッファに書き込み、終了する。リオーダバッファでは、命令発行順にレジスタへの書き込みを行う。

3.3 DSP拡張

組込みシステムでは、メディア処理、通信処理などの信号処理が多く用いられる。そのために専用ハードウェアを持つことも少なくないが、小さい画像のCODEC処理のような場合、専用ハードウェアに処理を渡して実行するオーバーヘッドが、実際のCODEC処理と比べて大きくなり、効率が良くなならない。また、CPUで信号処理を行った場合、基本的なRISC命令セットが信号処理に向いているとは言えないため、十分な性能が引き出せない。

このような背景から、DSP拡張命令を持つ組込みプロセッサが増えてきた。信号処理では、画素などに代表される相当数の8ビットデータや16ビットデータに対して同じ処理を行うことが多い。そのためDSP拡張命令には、例えば32ビットのレジスタを8ビットデータ4個、もしくは16ビットデータ2個と考え、4個または2個の加減算などを同時に行うような処理が追加される。複数データに対して同じ処理を行うことから、これはSIMD (Single Instruction Stream Multiple Data Streams) 命令と呼ばれる (図3.14)。

このほかにも、複数のデータをSIMD命令のために一つのレジスタに詰め込む命令、反対に複数のレジスタに分ける命令、一つのレジスタに詰め込まれた複数データの順序を入れ替える命令などが用意される。さらに、アプリケーションに特化した命令が追加される場合もある。例えば積和演算処理や、その演算結果がオーバーフローした場合に例外や違反とせず、最大数を出力する飽和積和演算処理などである。

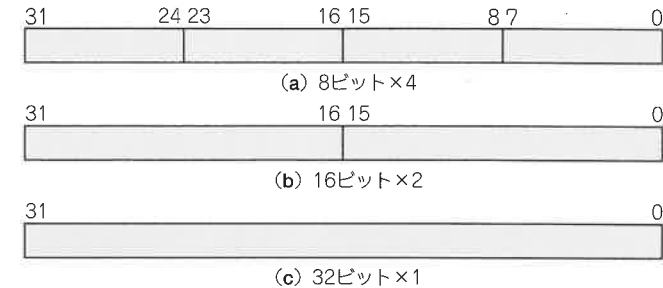


図3.14 DSP拡張命令のデータ型

3.4 FPU

組込みプロセッサにおいても、高品質のオーディオ処理や画像処理、高精度の制御処理に浮動小数点演算が使われるようになってきた。浮動小数点演算器は整数演算器と共用することが難しく、しかも整数レジスタとは別の倍精度を含む浮動小数点レジスタを持つと処理しやすいため、命令拡張ではなくコプロセッサを用意することが多い。このようなコプロセッサを、浮動小数点演算ユニット (FPU: Floating-Point Number Processing Unit) と呼ぶ。

FPUはIEEEの標準規格であるIEEE 754に準じて作られることが多いため、ここではIEEE 754に即して説明する。

3.4.1 IEEE 754とは

IEEE 754は正式には「IEEE Standard for Binary Floating-Point Arithmetic (ANSI/IEEE Std 754-1985)」という名称であり、1985年に正式に規格化された。規定されている内容は、

- 浮動小数点データの形式
- 加減乗除、平方根、剰余、比較演算
- 整数-浮動小数点データ間の形式変換
- 異なる浮動小数点データ間の変換
- 浮動小数点と10進ストリング間の変換
- 非数(NaN)を含む、浮動小数点例外とその処理

である。

データ形式(2進表現)は、単精度(32ビット)、倍精度(64ビット)、拡張単精度、拡張倍精度である。単精度データと倍精度データについて、図3.15に示す。デー

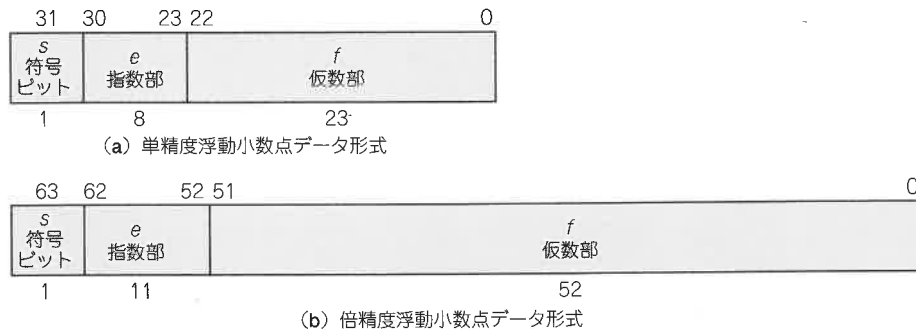


図3.15 浮動小数点データ形式

表3.1 IEEE 754における数の種類と単精度浮動小数点におけるデータ表現

数の種類	指数部 e	仮数部 f
正規化数(ノーマル数)	1 ~ 254	任意
非正規化数(デノーマル数)	0	0以外
非数(NaN)	255	0以外
無限大	255	0
ゼロ	0	0

表3.2 丸めモード

RM	記号	意味
00	RN	表現可能なもっとも近い値に結果を丸める。もっとも近い値が二つある場合は、最下位ビットが0の値のほうを選ぶ
01	RZ	ゼロの方向に丸める。いわゆる切り捨て
10	RP	$+\infty$ の方向に丸める
11	RM	$-\infty$ の方向に丸める

タは次の3領域に分けられる。

- 符号ビット: s
- 指数部: $e = E + \text{バイアス値}$ (バイアス値 = 単精度 $\rightarrow 0x7f$, 倍精度 $\rightarrow 0x3ff$)
- 仮数部: $f = .b_1b_2b_3b_4 \dots$ (小数点第1位以下の数)

これらの表現で、 $\pm 1.b_1b_2b_3b_4 \dots \times 2^E$ を表わす。

この表現では整数部は必ず1であり、これは正規化数(ノーマル数)と呼ばれている。このほかに、整数部が0である非正規化数(デノーマル数)、非数(NaN: Not a Number)、無限大、ゼロという特殊な数が定義されている(表3.1)。

またIEEE 754では4種の丸めモードがあり、RN(Toward Nearest)、RZ(Toward Zero)、RP(Toward Plus Infinity)、RM(Toward Minus Infinity)と呼ばれる(表3.2)。

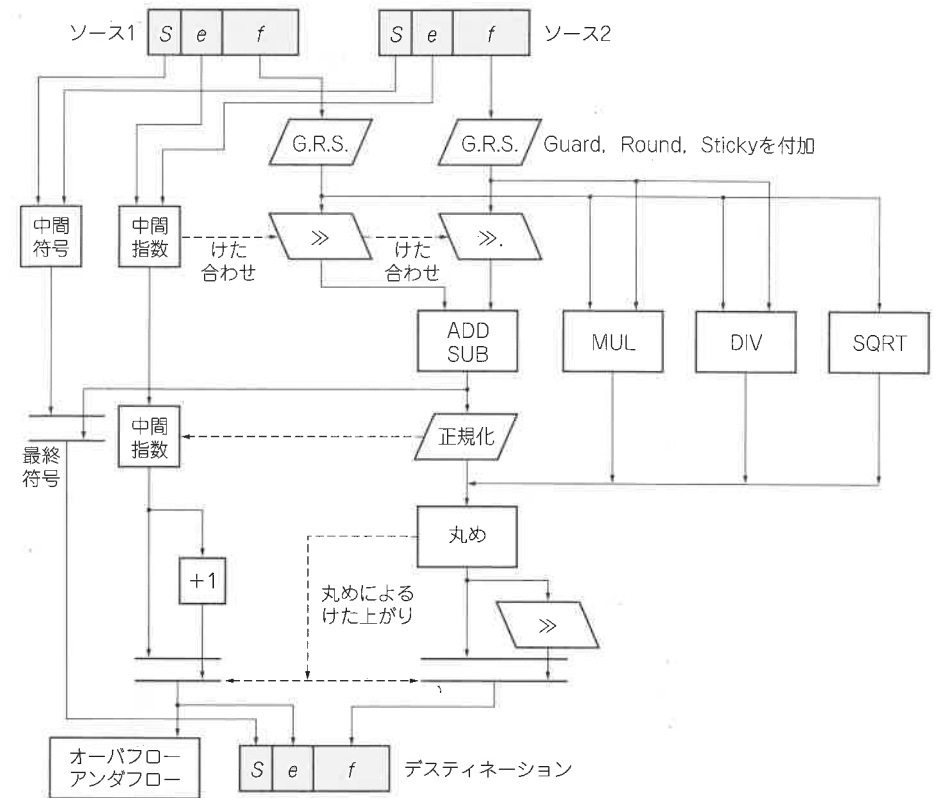


図3.16 FPUの構成例

3.4.2 浮動小数点演算の処理手順

FPUのハードウェアの概略を図3.16に示す。

(1) オペランド処理

まず、入力となる数を符号ビット、指数部、仮数部に分離する。この時点で、正規化数、非正規化数、NaN、無限大、ゼロなどが分かるので、特殊な数に対応する。

(2) 演算処理

符号ビット、指数部、仮数部のそれぞれに対して演算を実行する。図3.16の

G.R.S.は、“Guard, Round, Sticky”と呼ばれる丸め処理用のビットであり、これらを用いた演算方法についてIEEE 754に規定されている。

(3) 後処理

正規化、丸め、アンダフロー、オーバフローなどの処理を行う。また、必要に応じてトラップを発生する。

3.4.3 浮動小数点演算で発生する例外

IEEE 754では5種類の例外が定義されている。

- **無効浮動小数点演算例外 (Invalid Floating Operation Exception)**
無限大やNaNとの演算、(0/0)、負数の平方根などで発生する。
- **浮動小数点ゼロ除算例外 (Floating Zero Divide Exception)**
正規化数や非正規化数をゼロで除算した場合に発生する。
- **浮動小数点オーバフロー例外 (Floating Overflow Exception)**
演算結果の指数部が表現可能な最大数よりも大きくなった場合に発生する。
- **浮動小数点アンダフロー例外 (Floating Underflow Exception)**
演算結果の指数部が表現可能な最小数よりも小さくなった場合に発生する。
- **浮動小数点精度落ち例外 (Floating Inexact Exception)**
演算結果が精度落ちした場合に発生する。ほとんどの演算で起こる。

IEEE 754では、それぞれの例外について、発生条件、通知方法(フラグ)、トラップ発生設定方法と専用命令、結果格納方法について規定されている。特にオーバフロー、アンダフロー、精度落ちについては、例外が起こったとしても結果の精度を可能な限り保つように表現することが要求されている。

3.4.4 演算精度について

浮動小数点演算は整数演算と異なり丸め誤差が存在する。IEEE 754においてはかなり詳細に演算が規定されており、多くの場合、異なるFPUでも同じ結果が得られる。ただし、逆数計算のように複数種類のアルゴリズムが存在し、内部で多くの浮動小数点演算が行われるような演算では、途中の丸め誤差累積が異なるため、結果が異なることがある。

数値計算法では、丸め誤差の影響をできるだけ少なくするための方法(例えば、値の近い二つの数の減算はけた落ちが多くなるため避ける)が数多く知られているが、実際のソフトウェア構築の上ではアルゴリズムを工夫したうえでFPUを使うようにするべきである。

3.5 セキュリティ機能

携帯電話などの組込みシステムにおいては認証処理など、セキュリティ確保が重要なさまざまな処理がある。このような処理では多くの場合、秘密となる暗号鍵が使われる。一般ユーザは鍵の値には興味がなく、あるデータを暗号化した結果、あるいは暗号を解いて平文にした結果にのみ興味がある。鍵の値に興味があるのは悪意のあるユーザくらいのものである。このような場合のために、組込みプロセッサの中にセキュアモードと呼ばれるCPUの特別なプロセッサモードを設けたり、CPUとは別に安全性の高いプロセッサを設けてCPU処理を代行させたりすることがある。これらを総称してCPUのセキュリティ機能とよぶ。CPUのセキュリティ機能は、ARMのTrustZoneのように特権モードよりもさらに高いレベルのセキュリティを実現する。

セキュリティ機能は一般にセキュアメモリを持ち、暗号鍵などを蓄える。セキュアメモリのデータおよびセキュリティ機能向けのプログラムコードは、他のモード(ユーザのアプリケーションやOS)や他のプロセッサから参照できないようになっている。他のモードや他のプロセッサは、決められた手順やフォーマットでのみセキュリティ機能とのデータ授受とコマンド発行を行うことができる。また、セキュリティ機能はセキュアメモリ内の鍵を用いた暗号化など、決められたライブラリの実行のみを行う。このようにすることで、認証のための鍵や暗号処理などを外部か

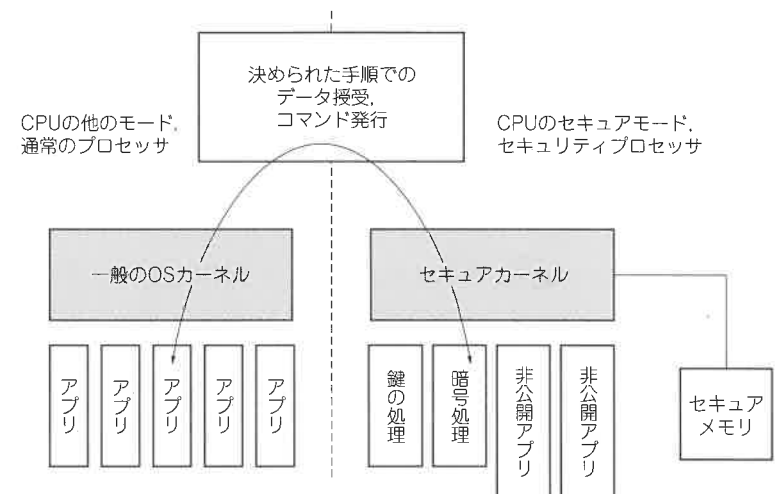


図3.17 セキュアプロセッサ