

Cortex™-A9 NEON™ メディア処理エンジン リビジョン : r2p2

テクニカルリファレンス マニュアル

ARM®

Cortex-A9 NEON メディア処理エンジン

テクニカルリファレンス マニュアル

Copyright © 2008-2010 ARM. All rights reserved.

リリース情報

本書には次の変更が加えられています。

改訂履歴

日付	変更箇所	公開の有無	変更内容
2008年4月4日	A	公開	r0p0用の最初のリリース
2008年7月10日	B	公開版、限定アクセス	r0p0用の2番目のリリース
2008年12月12日	C	公開版、限定アクセス	r1p0用の最初のリリース
2009年9月24日	D	公開版、限定アクセス	r2p0用の最初のリリース
2009年11月27日	E	公開版、アクセス制限なし	r2p0用の2番目のリリース
2010年4月27日	F	公開版、アクセス制限なし	r2p2用の最初のリリース

著作権

® または™ の付いた用語とロゴは、本著作権条項で特に明記されていない限り、EU および他諸国における ARM® の登録商標または商標です。本書に記載されている他の商標その他の名前は、対応する所有者の商標の場合があります。

本書に記載されている情報の全部または一部、ならびに本書で紹介する製品は、著作権所有者の文書による事前の許可を得ない限り、転用・複製することを禁じます。

本書に説明されている製品は、継続的に開発と改良が行われています。本書で言及されている製品とその利用方法に関する記載事項について、ARM は保証しません。したがって当社では、製品の商品性または目的への適合性を含め、黙示的・明示的に関係なく一切の保証を行いません。

本書は、本製品の利用者をサポートすることだけを目的としています。本書に記載されている情報の使用、情報の誤りまたは省略、あるいは本製品の誤使用によって発生したいかなる損失や損害についても、ARM は一切責任を負いません。

本書における ARM という用語は、「ARM、または該当する場合にはその子会社を含む」という意味で使用されています。

機密保持ステータス

本書は非機密扱いであり、本書を使用、複製、および開示する権利は、ARM および ARM が本書を提供した当事者との間で締結した契約の条項に基づいたライセンスの制限により異なります。

製品ステータス

本書の情報は最終版であり、開発済み製品に対応しています。

Web アドレス

<http://www.arm.com>

目次

Cortex-A9 NEON メディア処理エンジン テクニカルリファレンス マニュアル

	序章	
	本書について	xii
	ご意見・ご質問	xv
第 1 章	はじめに	
	1.1 Cortex-A9 NEON MPE について	1-2
	1.2 応用分野	1-4
	1.3 製品リビジョン	1-5
第 2 章	プログラマモデル	
	2.1 プログラマモデルについて	2-2
	2.2 アドバンスド SIMD と VFP の新機能	2-4
	2.3 IEEE754 規格への準拠	2-6
	2.4 サポートされている形式	2-7
	2.5 アドバンスド SIMD と VFP のレジスタへのアクセス	2-8
	2.6 レジスタの概要	2-13
	2.7 レジスタの説明	2-14

第 3 章

命令のタイミング

3.1	命令のサイクルタイミングについて	3-2
3.2	最適な VFP およびアドバンスド SIMD コードの作成	3-3
3.3	Cortex-A9 NEON MPE 命令	3-4
3.4	命令固有のスケジュール	3-10

付録 A

リビジョン

用語集

表一覧

Cortex-A9 NEON メディア処理エンジン テクニカルリファレンス マニュアル

	改訂履歴	ii
表 2-1	サポートされている数値形式	2-7
表 2-2	コプロセッサアクセス制御レジスタ	2-8
表 2-3	コプロセッサアクセス制御レジスタのビット割り当て	2-9
表 2-4	CRACR レジスタへのアクセスの結果	2-10
表 2-5	NSACR レジスタのビット割り当て	2-11
表 2-6	NSACR レジスタへのアクセスの結果	2-12
表 2-7	Cortex-A9 NEON MPE のシステムレジスタ	2-13
表 2-8	Cortex-A9 NEON MPE システムレジスタへのアクセス	2-13
表 2-9	FPSID レジスタのビット割り当て	2-14
表 2-10	FPSCR のビット割り当て	2-15
表 2-11	FPEXC レジスタのビット割り当て	2-18
表 3-1	Cortex-A9 MPE 命令	3-4
表 3-2	VFP 命令のタイミング	3-11
表 3-3	VFP ロード / ストア命令のタイミング	3-13
表 3-4	アドバンスド SIMD の整数演算命令のタイミング	3-14
表 3-5	アドバンスド SIMD の整数乗算命令	3-16
表 3-6	アドバンスド SIMD の整数シフト命令のタイミング	3-18
表 3-7	アドバンスド SIMD の置換命令のタイミング	3-19
表 3-8	アドバンスド SIMD の浮動小数点命令	3-20

表 3-9	アドバンスド SIMD のロード / ストア命令	3-22
表 A-1	A 版と B 版の相違点	A-1
表 A-2	B 版と C 版の相違点	A-1
表 A-3	C 版と D 版の相違点	A-2
表 A-4	D 版と E 版の相違点	A-2
表 A-5	E 版と F 版の相違点	A-2

図一覧

Cortex-A9 NEON メディア処理エンジン テクニカルリファレンス マニュアル

図 2-1	CPACR レジスタのビット割り当て	2-8
図 2-2	NSACR レジスタのビット割り当て	2-11
図 2-3	FPSID レジスタのビット割り当て	2-14
図 2-4	FPSCR のビット割り当て	2-15
図 2-5	FPEXC レジスタのビット割り当て	2-18

序章

本章では、*Cortex-A9 NEON™* メディア処理エンジン (MPE) テクニカルリファレンス マニュアルを紹介します。本章は次のセクションから構成されています。

- 「本書について」 (ページ xii)
- 「ご意見・ご質問」 (ページ xv)

本書について

本書は、Cortex-A9 NEON MPE のテクニカルリファレンス マニュアルです。

製品リビジョンステータス

mpn 識別子は、本書に記載されている製品のリビジョンステータスを示しています。各識別子の意味は次のとおりです。

rn 製品が大幅に修正されたことを示しています。

pn 製品に小さな修正または変更が加えられたことを示しています。

対象読者

本書は、Cortex-A9 NEON MPE を使用するシステム オンチップ (SoC) デバイスを設計するシステム設計者、システムインテグレータ、検証技術者を対象としています。本書では、Cortex-A9 MPE の外部機能について説明します。

本書の構成

本書は以下の章に分かれています。

第 1 章 はじめに

Cortex-A9 NEON MPE の高レベルなビューと、その機能について説明します。

第 2 章 プログラマモデル

Cortex-A9 NEON MPE のシステムレジスタについて説明します。

第 3 章 命令のタイミング

Cortex-A9 NEON MPE の命令のサイクルタイミングについて説明します。

付録 A リビジョン

本書の各版における技術的な変更点について説明します。

用語集

本書で使用されている用語の定義について説明します。

表記規則

本書では以下に示す表記規則が採用されています。

- 「書体の一般的な規則」 (ページ xiii)

書体の一般的な規則

本書で使用されている書体の一般的な規則は次のとおりです。

斜体	重要な注釈の強調、特別な用語の初出時、本書内での相互参照と引用に使用されます。
太字	メニュー名などのインタフェース要素を強調するために太字が使用されます。信号名を示すためにも使用されています。また、必要に応じて説明表の用語にも太字が使用されています。
<code>monospace</code>	コマンド、ファイル名、プログラム名、ソースコードなどの、キーボードから入力可能なテキストを示しています。
<u><code>monospace</code></u>	コマンドまたはオプションに使用可能な略語を示しています。コマンドやオプションの名前を全部入力する代わりに、下線部分のテキストだけを入力してこれらを指定できます。
<code>monospace italic</code>	具体的な値に置き換えられる引数を示しています。
<code>monospace bold</code>	サンプルコード以外で使用されている場合、言語のキーワードを示しています。
<および>	コードまたはコード片の中で不等号の括弧で囲まれている部分は、アセンブラ構文内で置き換え可能なことを示しています。次に例を示します。 <ul style="list-style-type: none"> • <code>MRC p15, 0 <Rd>, <CRn>, <CRm>, <Opcode_2></code>

参照資料

このセクションでは、ARM Limited やサードパーティが発行している出版物を紹介します。

ARM の出版物は Infocenter, <http://infocenter.arm.com> で参照できます。

ARM の刊行物

本書には、この製品に固有の情報が記載されています。他の関連情報については、以下の出版物を参照して下さい。

- *Cortex-A9* テクニカルリファレンス マニュアル (ARM DDI 0388)
- *Cortex-A9 MPCore* テクニカルリファレンス マニュアル (ARM DDI 0407)
- *Cortex-A9 浮動小数点ユニット* テクニカルリファレンス マニュアル (ARM DDI 0408)

- *Cortex-A9 MBIST* コントローラ テクニカルリファレンス マニュアル (ARM DDI 0414)
- *Cortex-A9* 構成およびサインオフ ガイド (ARM DII 0146)
- *CoreSight™ PTM™-A9* テクニカルリファレンス マニュアル (ARM DDI 0401)
- *CoreSight PTM-A9* 構成およびサインオフ ガイド (ARM DII 0161)
- *CoreSight PTM-A9* 統合マニュアル (ARM DII 0162)
- *CoreSight* プログラムフロートレースアーキテクチャ仕様 (ARM IHI 0035)
- *AMBA®* レベル2 キャッシュコントローラ (*L2C-310*) テクニカルリファレンス マニュアル (ARM DDI 0246)
- *L220* キャッシュコントローラ テクニカルリファレンス マニュアル (ARM DDI 0329)
- *AMBA AXI* プロトコル仕様 (ARM IHI 0022)
- *AMBA* 仕様 (ARM IHI 0011)
- *ARM* アーキテクチャ リファレンスマニュアル、*ARMv7-A* および *ARMv7-R* エディション (ARM DDI 0406)
- *RealView™* コンパイルツール デベロッパガイド (ARM DUI 0203)
- *RealView ICE* および *RealView Trace* ユーザガイド (ARM DUI 0155)
- インテリジェント電力コントローラ技術概要 (ARM DTO 0005)

その他の刊行物

このセクションでは、サードパーティが発行している関連出版物を紹介します。

- *ANSI/IEEE Std 754-1985, IEEE Standard for Binary Floating-Point Arithmetic*

ご意見・ご質問

ARM では、本製品と本書に関するご意見をお待ちしております。

製品に関するご意見

本製品に関するご意見・ご質問がございましたら、次の情報とともに製品購入元までご連絡下さい。

- 製品名
- 製品のリビジョンまたはバージョン
- できるだけ詳細な説明。該当する場合には、現象もご記載下さい。

本書の内容に関するご意見

本書の内容に関するご意見がございましたら、電子メールに次の情報をご記入の上、errata@arm.com までお寄せ下さい。

- 題名
- 資料番号、ARM DDI 0409FJ
- ご意見のあるページ番号
- ご意見についての簡潔な説明

補足または改善すべき点についての一般的なご意見もお待ちしております。

第 1 章

はじめに

本章では、ARM アドバンスド SIMD メディア処理アーキテクチャの Cortex-A9 実装を紹介します。本章は次のセクションから構成されています。

- 「Cortex-A9 NEON MPE について」 (ページ 1-2)
- 「応用分野」 (ページ 1-4)
- 「製品リビジョン」 (ページ 1-5)

1.1 Cortex-A9 NEON MPE について

Cortex-A9 NEON MPE は、Cortex-A9 の機能を拡張し、ARM v7 アドバンスト SIMD およびベクタ浮動小数点 v3 (VFPv3) 命令セットを提供します。

Cortex-A9 NEON MPE は、『ARM アーキテクチャ リファレンスマニュアル』に記載されているすべてのアドレッシングモードとデータ処理操作をサポートします。

Cortex-A9 NEON MPE には次の機能があります。

- SMID およびスカラの単精度浮動小数点演算
- スカラ倍精度浮動小数点演算
- SIMD およびスカラの半精度浮動小数点変換
- 8、16、32、64 ビットの符号付きおよび符号なし整数 SIMD 演算
- 1 ビット係数の 8 または 16 ビットの多項演算
- 構造化されたデータロード機能
- Cortex-A9 プロセッサによる ARM または Thumb 命令のデュアル発行
- VFPv3 とアドバンスト SIMD 命令用の独立したパイプライン
- 大容量の共有レジスタファイルが、次の形式でアクセス可能
 - 32 個の 32 ビット S (単精度) レジスタ
 - 32 個の 64 ビット D (倍精度) レジスタ
 - 16 個の 128 ビット Q (4 倍精度) レジスタ

MPE で各種のレジスタ形式に保持できるデータの詳細については、『ARM アーキテクチャ リファレンスマニュアル』を参照して下さい。

Cortex-A9 NEON MPE は、次のデータに対して、高パフォーマンスの SIMD ベクタ演算を提供します。

- 符号なしおよび符号付き整数
- 単一ビット係数の多項式
- 単精度浮動小数点値

操作には次のものがあります。

- 加算と減算
- 乗算、積和演算も可能
- 最大値または最小値によるレーン選択操作
- 逆平方根近似値
- 包括的なデータ構造ロード命令。レジスタバンクに存在するテーブルのルックアップも含まれます。

アドバンスト SIMD 命令の詳細については、『ARM アーキテクチャ リファレンスマニュアル』を参照して下さい。

注

アドバンスド SIMD アーキテクチャ拡張機能、それに関連する実装、およびサポートを行うソフトウェアを、一般に NEON™ テクノロジーと呼びます。

1.1.1 VFPv3 アーキテクチャのハードウェアサポート

Cortex-A9 NEON MPE のハードウェアは、ARM VFPv3 アーキテクチャに記載されている単精度および倍精度の加算、減算、乗算、除算、積和演算、平方根演算をサポートします。16 ビット、32 ビット、64 ビットの浮動小数点形式および ARM 整数ワード形式の相互変換を行い、高レベル言語をサポートするために使用される 0 への丸めモードで変換を実行するための特別な操作も用意されています。

すべての命令は、Cortex-A9 プロセッサファミリでサポートされている ARM および Thumb 命令セットの両方で使用可能です。

Cortex-A9 NEON MPE は、組み込み用途やメディア演算を大量に使用する用途に向けた、パフォーマンス、電力、実装面積において最適化されたソリューションを提供します。

ARMv7 では、VFP ベクタモードの使用は非推奨です。Cortex-A9 NEON MPE のハードウェアは、VFP ベクタ演算をサポートしません。本書では、ベクタという用語はアドバンスド SIMD の整数、多項式、単精度のベクタ操作を指します。Cortex-A9 NEON MPE は、サポートコードなしで VFP 演算を高速に実行できます。ただし、アプリケーションで VFP ベクタ演算が必要な場合、サポートコードを使用する必要があります。VFP ベクタ演算のサポートについては、『ARM アーキテクチャ リファレンスマニュアル』を参照して下さい。

注

本書では、ARM アドバンスド SIMD および VFPv3 拡張機能の、Cortex-A9 NEON MPE 実装に固有の情報について説明します。完全な命令セットと使用方法の詳細については、『ARM アーキテクチャ リファレンスマニュアル』を参照して下さい。

1.2 応用分野

Cortex-A9 NEON MPE は、データ型が混在する SIMD、および高パフォーマンスのスカラ浮動小数点演算を実行でき、次のような広範な応用分野に適しています。

- PDA やスマートフォン用のグラフィック、音声の圧縮と伸張、ユーザインタフェース、Java 解析、*Just In Time* (JIT) コンパイル
- ゲーム機で多用される 3 次元グラフィック、デジタルオーディオ、および重力などゲーム内の物理演算
- プリンタおよび複合機 (MFP) コントローラによる高解像度のカラーレンダリング
- セットトップ ボックス用の高品質なデジタルオーディオとデジタルビデオ、および対話型 3 次元ユーザインタフェース
- 自動車向けのエンジン制御と伝導機構の計算処理、および車内でのエンターテイメントやナビゲーション

1.3 製品リビジョン

ここでは、製品リビジョン間の機能の相違点について説明します。

r0p0 - r1p0 機能の変更はありませんが、Cortex-A9 リビジョン r1p0 の設計では、リビジョン r1p0 の NEON MPE を使用する必要があります。

r1p0 - r2p0 機能の変更はありませんが、Cortex-A9 リビジョン r2p0 の設計では、リビジョン r2p0 の NEON MPE を使用する必要があります。

r2p0 - r2p1 機能の変更はありませんが、Cortex-A9 リビジョン r2p1 の設計では、リビジョン r2p1 の NEON MPE を使用する必要があります。

r2p1 - r2p2 機能の変更はありません。

はじめに

第 2 章 プログラマモデル

本章では、Cortex-A9 NEON MPE のプログラマモデルについて説明します。
本章は次のセクションから構成されています。

- 「プログラマモデルについて」 (ページ 2-2)
- 「アドバンスド SIMD と VFP の新機能」 (ページ 2-4)
- 「IEEE754 規格への準拠」 (ページ 2-6)
- 「サポートされている形式」 (ページ 2-7)
- 「アドバンスド SIMD と VFP のレジスタへのアクセス」 (ページ 2-8)
- 「レジスタの概要」 (ページ 2-13)
- 「レジスタの説明」 (ページ 2-14)

2.1 プログラマモデルについて

ここでは、Cortex-A9 NEON MPE により提供される VFPv3 とアドバンスド SIMD の実装について紹介します。また、アプリケーションコードを実行する準備として Cortex-A9 NEON MPE を初期化する方法についても説明します。

アドバンスド SIMD と VFP の実装の従来の組み合わせによって提供されていた機能に加えて、Cortex-A9 NEON MPE では次の機能を使用できます。

- 半精度（16ビット）浮動小数点値の変換
- VFPv3-D32 および VFPv3-D16 のエミュレーションのサポート

詳細については、『*ARM アーキテクチャ リファレンスマニュアル*』を参照して下さい。

2.1.1 アドバンスド SIMD および VFP 機能識別レジスタ

Cortex-A9 NEON MPE は、ARMv7 のアドバンスド SIMD および VFP 拡張機能を実装しています。

ソフトウェアでは、機能識別レジスタを使用して、これらの拡張機能と、提供される機能を識別できます。拡張機能は、コプロセッサ CP10 および CP11 のコプロセッサ空間に存在します。これらのレジスタにアクセスするには、VMRS および VMSR を次の例のように使用します。

```
VMRS <Rd>, FPSID ; Read Floating-Point System ID Register
VMRS <Rd>, MVFR1 ; Read Media and VFP Feature Register 1
VMSR FPSCR, <Rt> ; Write Floating-Point System Control Register
```

レジスタの説明については、「アドバンスド SIMD と VFP のレジスタへのアクセス」（ページ 2-8）を参照して下さい。

加えて、コプロセッサアクセス制御レジスタも存在します。「非セキュアアクセス制御レジスタ」（ページ 2-10）および「コプロセッサアクセス制御レジスタ」（ページ 2-8）を参照して下さい。

2.1.2 アドバンスド SIMD と浮動小数点のサポートの稼働

リセット時には、アドバンスド SIMD および VFP 拡張機能はいずれも非稼働です。NEON または VFP 命令の実行を試みると、常に未定義命令例外が取得されます。ソフトウェアからアドバンスド SIMD および VFP 機能へのアクセスを可能にするには、次の条件が満たされている必要があります。

- 適切な特権レベルで、CP10 および CP11 へのアクセスが可能である。
「コプロセッサアクセス制御レジスタ」（ページ 2-8）を参照して下さい。

- アドバンスト SIMD 機能または VFP 機能へ非セキュアアクセスが必要な場合、NSACR のアクセスフラグが CP10 および CP11 について 1 にセットされている必要があります。「非セキュアアクセス制御レジスタ」(ページ 2-10) を参照して下さい。

さらに、ほとんどのアドバンスト SIMD および VFP 動作は、ソフトウェアで FPEXC.EN ビットを 1 にセットして稼働状態にする必要があります。「浮動小数点例外レジスタ」(ページ 2-17) を参照して下さい。

FPEXC.EN が 0 でアドバンスト SIMD および VFP 動作が非稼働のとき、アドバンスト SIMD および VFP 命令は、特権モードで次の命令を実行する場合を除いて、常に未定義命令として扱われます。

- FPEXC または FPSID レジスタへの VMRS
- FPEXC、FPSID、MVFR0、MVFR1 レジスタからの VMRS

アドバンスト SIMD と VFP を稼働する方法を、ARM 統一アセンブリ言語 (UAL) で例 2-1 に示します。このコードは、特権モードで実行する必要があります。

例 2-1 アドバンスト SIMD と VFP の稼働

```

MRC p15,0,r0,c1,c0,2 ; Read CPACR into r0
ORR r0,r0,#(3<<20) ; OR in User and Privileged access for CP10
ORR r0,r0,#(3<<22) ; OR in User and Privileged access for CP11
BIC r0, r0, #(3<<30) ; Clear ASEDIS/D32DIS if set
MCR p15,0,r0,c1,c0,2 ; Store new access permissions into CPACR
ISB ; Ensure side-effect of CPACR is visible
MOV r0,#(1<<30) ; Create value with FPEXC (bit 30) set in r0
VMRS FPEXC,r0 ; Enable VFP and SIMD extensions

```

この時点で、Cortex-A9 プロセッサはアドバンスト SIMD および VFP 命令を実行できます。

注

コプロセッサアクセス制御レジスタ (CPACR) の構成で、CP10 と CP11 に同一のアクセス許可が与えられていない場合、動作は予測不能です。

2.2 アドバンスト SIMD と VFP の新機能

Cortex-A9 NEON MPE には、ARMv7 アドバンスト SIMD および VFP アーキテクチャの次の新機能が実装されています。

- 「半精度浮動小数点値の変換」
- 「アドバンスト SIMD と VFP を別々に非稼働にする」
- 「VFP レジスタのバンクサイズを動的に構成可能」

新しい命令と制御レジスタフィールドの完全な詳細については、『ARM アーキテクチャ リファレンスマニュアル』を参照して下さい。

2.2.1 半精度浮動小数点値の変換

半精度浮動小数点値の変換は、16 ビット浮動小数点値の IEEE 表現、および代替半精度表現と呼ばれる一般的なグラフィック表現の両方をサポートしています。この形式により、大量の低精度の浮動小数点値を格納するアプリケーションでメモリ消費量が少なくなり、ソフトウェアで変換を行うオーバーヘッドも回避できます。

追加された VFP およびアドバンスト SIMD 命令により、単精度浮動小数点表現との間で、個別の値とベクタの両方の変換が可能です。その後で、他の VFP およびアドバンスト SIMD 命令を使用して、これらの値を処理できます。

IEEE または代替半精度モードを選択する方法については、「浮動小数点ステータスおよび制御レジスタ」(ページ 2-15) を参照して下さい。

2.2.2 アドバンスト SIMD と VFP を別々に非稼働にする

Cortex-A9 NEON MPE を持つ Cortex-A9 実装でアドバンスト SIMD のみを非稼働にすると、VFP 拡張機能のみが存在しているかのように動作します。これによって、Cortex-A9 NEON MPE と、浮動小数点のみのユニットとの両方を含む Cortex-A9 マルチプロセッサ クラスタ間で、最適のオペレーティングシステム タスクスケジューリングが可能になります。

Cortex-A9 プロセッサには、非セキュアアクセスでこの機能が使用されることを防止するためのサポートが含まれています。「非セキュアアクセス制御レジスタ」(ページ 2-10) および「コプロセッサアクセス制御レジスタ」(ページ 2-8) を参照して下さい。

2.2.3 VFP レジスタのバンクサイズを動的に構成可能

VFP レジスタのバンクサイズを動的に構成可能なため、VFPv3-D16 と VFPv3-D32 が混在しているマルチプロセッサ クラスタで追加サポートが可能です。Cortex-A9 NEON MPE には、32 個の 64 ビット倍精度レジスタが実装されています。VFP のみの実装では、必ずサポートされている倍精度レジスタ

は 16 個のみです。このレジスタバンク非稼働機能により、16 エントリの倍精度レジスタファイルをエミュレート可能になるため、互換性が強化されると同時に、より柔軟なタスクスケジュールが可能になります。

非セキュアアクセス制御レジスタでは、他の制御も可能です。「非セキュアアクセス制御レジスタ」(ページ 2-10) および「コプロセッサアクセス制御レジスタ」(ページ 2-8) を参照して下さい。

2.3 IEEE754 規格への準拠

IEEE754 規格では、実装で多くの選択が可能になっています。『ARM アーキテクチャ リファレンスマニュアル』では、アドバンスト SIMD と VFPv3 アーキテクチャに適用される選択について説明されています。

Cortex-A9 NEON MPE は、ARMv7 のアドバンスト SIMD および VFP 拡張機能を実装しています。次に示す IEEE754 の操作は、ハードウェアでサポートされていません。

- 剰余
- 浮動小数点数の、最も近い整数値浮動小数点数への丸め
- 2 進数から 10 進数への変換
- 10 進数から 2 進数への変換
- 単精度値と倍精度値との直接比較
- すべての拡張精度操作

2.4 サポートされている形式

Cortex-A9 NEON MPE に実装されているアドバンスド SIMD および VFPv3 命令セットのそれぞれについて、サポートされている形式を、表 2-1 に示します。すべての符号付き整数は、2 の補数形式の表現です。

表 2-1 サポートされている数値形式

形式	アドバンスド SIMD	VFPv3
8 ビット符号付き / 符号なし整数	はい	いいえ
16 ビット符号付き / 符号なし整数	はい	いいえ
32 ビット符号付き / 符号なし整数	はい	はい ^a
64 ビット符号付き / 符号なし整数	はい	いいえ
16 ビット半精度浮動小数点数	はい ^a	はい ^a
32 ビット単精度浮動小数点数	はい	はい
64 ビット倍精度浮動小数点数	いいえ	はい
8 ビット多項式	はい	いいえ
16 ビット多項式	はい	いいえ

a. 変換目的のみ

2.5 アドバンスト SIMD と VFP のレジスタへのアクセス

システム制御コプロセッサレジスタを、表 2-2 に示します。このレジスタは CP15 経由でアクセスされ、アドバンスト SIMD および VFP レジスタへのアクセスを決定します。ここで、各項目は次の意味です。

- CRn は、CP15 内部のレジスタ番号です。
- Op1 は、レジスタの Opcode_1 の値です。
- CRm は、操作対象のレジスタです。
- Op2 は、レジスタの Opcode_2 の値です。

表 2-2 コプロセッサアクセス制御レジスタ

CRn	Op1	CRm	Op2	名前	説明
c1	0	c0	2	CPACR	「コプロセッサアクセス制御レジスタ」を参照
c1	0	c1	2	NSACR	「非セキュアアクセス制御レジスタ」(ページ 2-10) を参照

2.5.1 コプロセッサアクセス制御レジスタ

CPACR レジスタは、コプロセッサ CP10 および CP11 のアクセス権を設定し、これらによって Cortex-A9 NEON MPE 機能が稼働されます。また、ソフトウェアからこのレジスタを使用して、システムに特定のコプロセッサが存在しているかどうかを判断できます。

CPACR レジスタの特徴は次のとおりです。

- 読み出し / 書き込みレジスタで、セキュア状態と非セキュア状態で共通
- 特権モードでのみアクセス可能
- リセット時の値は 0 です。

CPACR レジスタのビット割り当てを、図 2-1 に示します。

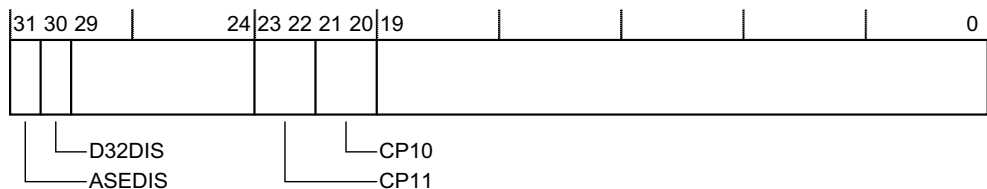


図 2-1 CPACR レジスタのビット割り当て

CPACR レジスタのビット割り当てを、表 2-3 に示します。

表 2-3 コプロセッサアクセス制御レジスタのビット割り当て

ビット	フィールド	機能
[31]	ASEDIS	アドバンスド SIMD 拡張機能の非稼働。 b1 = 『ARM アーキテクチャ リファレンスマニュアル』で、アドバンスド SIMD 拡張機能の一部であるが VFPv3 命令ではないとされている、すべての命令エンコードが不可能になります。 b0 = すべての命令が可能です。
[30]	D32DIS	VFP レジスタファイル D16 ~ D31 の使用を不可能にします。 b1 = 『ARM アーキテクチャ リファレンスマニュアル』で VFPv3 命令とされているすべての命令エンコードが、レジスタ D16 ~ D31 のいずれにもアクセス不可能になります。 b0 = すべての命令が可能です。
[29:24]	-	『Cortex-A9 テクニカルリファレンスマニュアル』を参照
[23:22]	CP11	コプロセッサのアクセス許可を定義します。リセット時の状態はアクセス拒否で、コプロセッサが存在しない場合もこの状態です。 b00 = アクセス拒否。アクセスを試みると、未定義命令例外が生成されます。 b01 = 特権モードアクセスのみ b10 = 予約 b11 = 特権モードアクセスとユーザモード アクセス
[21:20]	CP10	コプロセッサのアクセス許可を定義します。リセット時の状態はアクセス拒否で、コプロセッサが存在しない場合もこの状態です。 b00 = アクセス拒否。アクセスを試みると、未定義命令例外が生成されます。 b01 = 特権モードアクセスのみ b10 = 予約 b11 = 特権モードアクセスとユーザモード アクセス
[19:0]	-	『Cortex-A9 テクニカルリファレンスマニュアル』を参照

非セキュア状態でのコプロセッサへのアクセスは、「非セキュアアクセス制御レジスタ」(ページ 2-10) で設定されているアクセス許可によって異なります。

CPACR レジスタのアクセスビットへの読み書きを試みた結果は、「非セキュアアクセス制御レジスタ」内で各コプロセッサに対応するビットによって異なります。それぞれのモードについて、コプロセッサのアクセスビットにアクセスを試みた結果を、表 2-4 に示します。

表 2-4 CRACR レジスタへのアクセスの結果

NSACR[11:10]	セキュア特権	非セキュア特権	セキュアまたは非セキュアユーザ
b00	R/W	RAZ/WI	アクセス禁止 ^a
b01	R/W	R/W	アクセス禁止 ^a

a. ユーザ権限でアクセスを行うと、未定義命令例外が生成されます。

CPACR レジスタにアクセスするには、次の命令を使用して CP15 を読み書きします。

```
MRC p15, 0, <Rd>, c1, c0, 2 ; Read Coprocessor Access Control Register
MCR p15, 0, <Rd>, c1, c0, 2 ; Write Coprocessor Access Control Register
```

CPACR が更新されるとき、レジスタへの変更が可視となることが保証されるのは、次の命令同期バリア (ISB) 命令を実行した後でのみです。このレジスタが更新されるとき、新しいまたは古いレジスタの値に依存する命令は、ISB 命令の前に発行されないことを、ソフトウェアで保証する必要があります。

通常、ソフトウェアで CPACR を更新するときは、他のコプロセッサのアクセス設定を不必要に変更してしまうことを避けるため、読み出し - 変更 - 書き込みシーケンスを使用します。

注

アドバンスド SIMD または VFP システムレジスタにアクセスする前に、CPACR レジスタの CP10 および CP11 を可能にする必要があります。

2.5.2 非セキュアアクセス制御レジスタ

NSACR レジスタは、Cortex-A9 NEON MPE および他のシステム機能への、非セキュアアクセス権を定義します。

NSACR レジスタの特徴は次のとおりです。

- セキュア状態では読み出し / 書き込みレジスタ
- 非セキュア状態では読み出し専用レジスタ
- 特権モードでのみアクセス可能

Cortex-A9 MPE における NSACR レジスタのビット割り当てを、図 2-2 (ページ 2-11) に示します。このレジスタの他のフィールドの詳細については、『Cortex-A9 テクニカルリファレンス マニュアル』を参照して下さい。

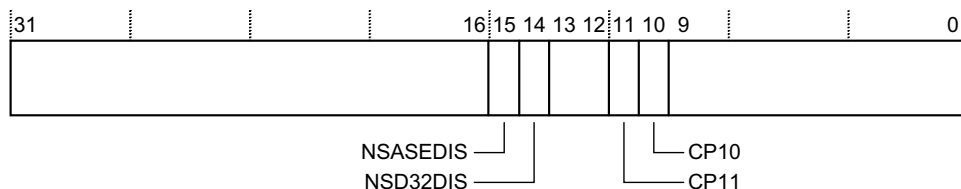


図 2-2 NSACR レジスタのビット割り当て

NSACR レジスタのビット割り当てを、表 2-5 に示します。

表 2-5 NSACR レジスタのビット割り当て

ビット	フィールド	機能
[31:16]	-	『Cortex-A9 テクニカルリファレンス マニュアル』を参照
[15]	NSASEDIS	非セキュアアドバンスト SIMD 拡張機能を非稼働にします。 b0 = CPACR.ASEDIS への完全なアクセスが提供されます。 b1 = 非セキュア状態で実行している場合は、CPACR.ASEDIS ビットの値が 1 に固定され、書き込みは無視されます。
[14]	NSD32DIS	VFP レジスタファイル D16 ~ D31 の非セキュア使用を不可能にします。 b0 = CPACR.D32DIS への完全なアクセスが提供されます。 b1 = 非セキュア状態で実行している場合は、CPACR.D32DIS ビットの値が 1 に固定され、書き込みは無視されます。
[13:12]	-	『Cortex-A9 テクニカルリファレンス マニュアル』を参照
[11]	CP11	コプロセッサ 11 へのアクセス許可。 b0 = セキュアアクセスのみ。これはリセット時の値です。 b1 = セキュアと非セキュアの両方のアクセスが可能
[10]	CP10	コプロセッサ 10 へのアクセス許可。 b0 = セキュアアクセスのみ。これはリセット時の値です。 b1 = セキュアと非セキュアの両方のアクセスが可能
[9:0]	-	『Cortex-A9 テクニカルリファレンス マニュアル』を参照

NSACR レジスタにアクセスするには、次の命令を使用して CP15 を読み書きします。

```
MRC p15, 0, <Rd>, c1, c1, 2 ; Read Non-secure Access Control Register data
MCR p15, 0, <Rd>, c1, c1, 2 ; Write Non-secure Access Control Register data
```

各モードでアクセスを試みた結果を、表 2-6 (ページ 2-12) に示します。

表 2-6 NSACR レジスタへのアクセスの結果

セキュア特権		非セキュア特権		ユーザ	
読み出し	書き込み	読み出し	書き込み	読み出し	書き込み
データ	データ	データ	未定義命令例外	未定義命令例外	未定義命令例外

2.6 レジスタの概要

Cortex-A9 NEON MPE システムレジスタを、表 2-7 に示します。すべての NEON MPE システムレジスタは 32 ビット幅です。予約レジスタのアドレスは RAZ/WI です。

表 2-7 Cortex-A9 NEON MPE のシステムレジスタ

名前	タイプ	リセット時の値	説明
FPSID	RO	0x41033092	「浮動小数点システム ID レジスタ」(ページ 2-14) を参照
FPSCR	RW	0x00000000	「浮動小数点ステータスおよび制御レジスタ」(ページ 2-15) を参照
MVFR1	RO	0x01111111	『ARM アーキテクチャ リファレンスマニュアル』を参照
MVFR0	RO	0x10110222	『ARM アーキテクチャ リファレンスマニュアル』を参照
FPEXC	RW	0x00000000	「浮動小数点例外レジスタ」(ページ 2-17) を参照

Cortex-A9 NEON MPE システムレジスタへアクセスするためのプロセッサモードを、表 2-8 に示します。

表 2-8 Cortex-A9 NEON MPE システムレジスタへのアクセス

レジスタ	特権アクセス		ユーザアクセス	
	FPEXC EN = 0	FPEXC EN = 1	FPEXC EN = 0	FPEXC EN = 1
FPSID	許可	許可	禁止	禁止
FPSCR	禁止	許可	禁止	許可
MVFR0、MVFR1	許可	許可	禁止	禁止
FPEXC	許可	許可	禁止	禁止

2.7 レジスタの説明

ここでは、Cortex-A9 NEON MPE のシステムレジスタについて説明します。
表 2-7（ページ 2-13）には、各レジスタへの相互参照がまとめられています。

2.7.1 浮動小数点システム ID レジスタ

FPSID レジスタの特徴は次のとおりです。

目的	VFP 実装についての情報を提供します。
使用制限	特権モードでのみアクセス可能です。
構成	すべての NEON MPE 構成で使用可能です。
属性	レジスタの概要については、表 2-7（ページ 2-13）を参照して下さい。

FPSID レジスタのビット割り当てを、図 2-3 に示します。

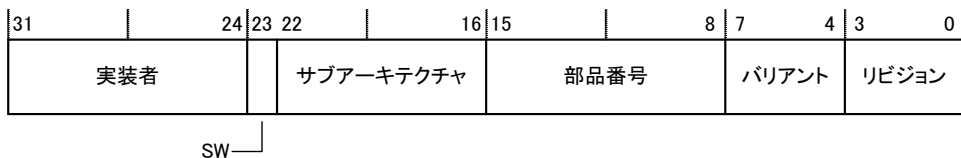


図 2-3 FPSID レジスタのビット割り当て

FPSID レジスタのビット割り当てを、表 2-9 に示します。

表 2-9 FPSID レジスタのビット割り当て

ビット	名前	機能
[31:24]	実装者	ARM を示します。
[23]	SW	ハードウェア実装、ソフトウェアエミュレーションなし
[22:16]	サブアーキテクチャ	NULL VFP サブアーキテクチャ
[15:8]	部品番号	VFPv3
[7:4]	バリエーション	Cortex-A9
[3:0]	リビジョン	リビジョン 2

FPSID レジスタにアクセスするには、次の VMRS 命令を使用します。

VMRS <Rd>, FPSID ; Read Floating-Point System ID Register

2.7.2 浮動小数点ステータスおよび制御レジスタ

FPSCR には次の特徴があります。

目的	FPU をユーザレベルで制御するために使用します。
使用制限	使用制限はありません。
構成	すべての FPU 構成で使用可能です。
属性	レジスタの概要については、表 2-7 (ページ 2-13) を参照して下さい。

FPSCR のビット割り当てを、図 2-4 に示します。

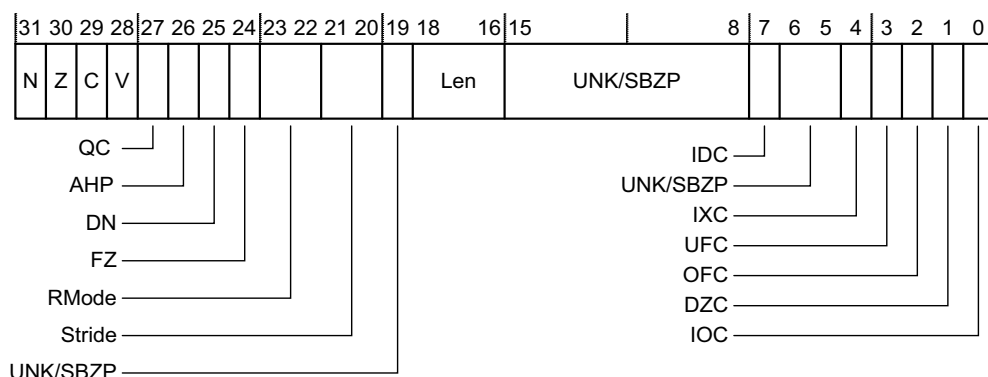


図 2-4 FPSCR のビット割り当て

FPSCR のビット割り当てを、表 2-10 に示します。

表 2-10 FPSCR のビット割り当て

ビット	フィールド	機能
[31]	N	比較操作によって「より小さい」の結果が生成された場合、1 にセットされます。
[30]	Z	比較操作によって「等しい」の結果が生成された場合、1 にセットされます。
[29]	C	比較操作によって「等しい」、「より大きい」、または「順序付けなし」の結果が生成された場合、1 にセットされます。
[28]	V	比較操作によって「順序付けなし」の結果が生成された場合、1 にセットされます。
[27]	QC	このビットに最後に 0 が書き込まれた後で、アドバンスト SIMD 整数演算の飽和が発生した場合、1 にセットされます。 ^a

表 2-10 FPSCR のビット割り当て (続き)

ビット	フィールド	機能
[26]	AHP	代替半精度の制御ビット。 b0 = IEEE 半精度形式が選択されています。 b1 = 代替半精度形式が選択されています。
[25]	DN	デフォルト NaN モードの制御ビット。 b0 = NaN オペランドは浮動小数点演算の出力まで伝播されます。 b1 = 1 つ以上の NaN を含む演算はすべて、デフォルト NaN を返します。 アドバンスド SIMD 算術演算は、DN ビットの値にかかわらず常にデフォルト NaN の設定を使用します。
[24]	FZ	Flush-to-Zero モード制御ビット。 b0 = Flush-to-Zero モードが不可能です。浮動小数点システムの動作は、IEEE 754 規格に完全に準拠しています。 b1 = Flush-to-Zero モードが可能です。 アドバンスド SIMD 算術演算は、FZ ビットの値にかかわらず常に Flush-to-Zero の設定を使用します。
[23:22]	RMode	丸めモード制御フィールド。 b00 = 近似値への丸め (RN) モード b01 = プラス無限大への丸め (RP) モード b10 = マイナス無限大への丸め (RM) モード b11 = 0 への丸め (RZ) モード アドバンスド SIMD 算術演算は、RMode ビットの値にかかわらず常に近似値への丸めの設定を使用します。
[21:20]	Stride	ショートベクタ演算との下位互換性のため使用されるストライド制御。 Cortex-A9 NEON MPE では、このフィールドの値は無視されます。 『ARM アーキテクチャ リファレンスマニュアル』を参照して下さい。
[19]	-	UNK/SBZP
[18:16]	Len	ショートベクタ演算との下位互換性のため使用されるベクタ長。 このフィールドを 0 以外の値に設定すると、VFP データ処理命令で例外が生成されます。 『ARM アーキテクチャ リファレンスマニュアル』を参照して下さい。
[15:8]	-	UNK/SBZP
[7]	IDC	入力非正規化累積例外フラグ ^a
[6:5]	-	UNK/SBZP
[4]	IXC	不正確累積例外フラグ ^a
[3]	UFC	アンダーフロー累積例外フラグ ^a

表 2-10 FPSCR のビット割り当て (続き)

ビット	フィールド	機能
[2]	OFC	オーバフロー累積例外フラグ ^a
[1]	DZC	0 による除算の累積例外フラグ ^a
[0]	IOC	無効演算の累積例外フラグ ^a

- a. 例外フラグである FPSCR のビット [27]、ビット [7]、ビット [4:0] は **DEFLAG**S 出力にエクスポートされるため、必要に応じてプロセッサの外部から監視できます。

FPSCR レジスタにアクセスするには、次の **VMSR** 命令を使用します。

```
VMRS <Rd>, FPSCR ; Read Floating-Point Status and Control Register
VMSR FPSCR, <Rt> ; Write Floating-Point Status and Control Register
```

2.7.3 浮動小数点例外レジスタ

FPEXC レジスタの特徴は次のとおりです。

目的	アドバンスト SIMD および VFP 拡張機能のグローバルな制御を行うために使用します。
使用制限	<ul style="list-style-type: none"> NSACR の CP10 ビットと CP11 ビットが 1 にセットされている場合のみ、非セキュア状態でのみアクセスできます。「非セキュアアクセス制御レジスタ」(ページ 2-10) を参照して下さい。 特権モードで、かつコプロセッサアクセス制御レジスタでコプロセッサ CP10 と CP11 へのアクセスが可能になっている場合のみ、アクセスできます。「コプロセッサアクセス制御レジスタ」(ページ 2-8) を参照して下さい。
構成	すべての NEON MPE 構成で使用可能です。
属性	レジスタの概要については、表 2-7 (ページ 2-13) を参照して下さい。

FPEXC レジスタのビット割り当てを、図 2-5 (ページ 2-18) に示します。

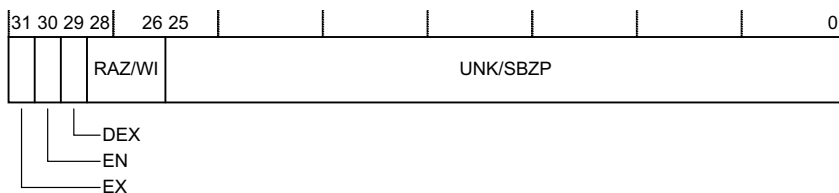


図 2-5 FPEXC レジスタのビット割り当て

FPEXC レジスタのビット割り当てを、表 2-11 に示します。

表 2-11 FPEXC レジスタのビット割り当て

ビット	名前	機能
[31]	EX	例外ビット。 このビットは 0 として読み出され、書き込みは無視されます。 Cortex-A9 NEON MPE では、非同期例外処理が必要になることはありません。
[30]	EN	イネーブルビット。 b0 = アドバンスド SIMD 拡張機能と VFP 拡張機能が非稼働です。 b1 = アドバンスド SIMD 拡張機能と VFP 拡張機能は稼働中で、正常に動作します。 EN ビットは、リセット時に 0 にクリアされます。
[29]	DEX	定義済み同期命令例外フラグ。 b0 = 例外は発生していません。 b1 = VFP ベクタ操作を実行する試みがトラップされました。 ^a DEX ビットは、リセット時に 0 にクリアされます。
[28:26]	-	RAZ/WI
[25:0]	-	UNK/SBZP

a. Cortex-A9 NEON MPE ハードウェアは、非推奨の VFP ショートベクタ機能をサポートしていません。FPSCR.LEN フィールドが 0 以外有的时候に VFP データ処理命令の実行を試みると、FPEXC.DEX ビットがセットされ、同期未定義命令例外が取得されます。必要であれば、ソフトウェアでショートベクタ機能をエミュレートできます。

FPEXC レジスタにアクセスするには、次の VMSR 命令を使用します。

VMSR <Rd>, FPEXC ; Read Floating-Point Status and Control Register
VMSR FPEXC, <Rt> ; Write Floating-Point Status and Control Register

第 3 章

命令のタイミング

本章では、Cortex-A9 NEON MPE での命令のサイクルタイミングについて説明します。本章は次のセクションから構成されています。

- 「命令のサイクルタイミングについて」 (ページ 3-2)
- 「最適な VFP およびアドバンスド SIMD コードの作成」 (ページ 3-3)
- 「Cortex-A9 NEON MPE 命令」 (ページ 3-4)
- 「命令固有のスケジュール」 (ページ 3-10)

3.1 命令のサイクルタイミングについて

本章では、特定のコードシーケンスに必要な実行時間を見積もるための情報を提供します。Cortex-A9 プロセッサの複雑さのため、手作業では正確なタイミング情報を計算できません。命令のタイミングの多くは、同時に実行される他の命令、メモリシステムの動作、および命令フロー以外のイベントの影響を受けます。

可能性のある命令の相互作用とプロセッサ内で発生する可能性のあるイベントの詳細な説明は、本書の範囲を越えています。

3.2 最適な VFP およびアドバンスト SIMD コードの作成

以下のガイドラインに従うと、VFP およびアドバンスト SIMD コードで大幅なパフォーマンス向上を実現できます。

次の操作は、できる限り避けて下さい。

- VFP 制御レジスタへの不要なアクセス
- Cortex-A9 コアレジスタと、VFP またはアドバンスト SIMD レジスタファイルとの間で値を転送する操作。コアレジスタの定義については、『*ARM アーキテクチャ リファレンスマニュアル*』を参照して下さい。
- 隣り合った命令間でのレジスタ依存関係
- アドバンスト SIMD 専用命令と、VFP 専用命令との混在

次の点に注意して下さい。

- ロードやストアを同時に行う場合を除いて、プロセッサは VFP およびアドバンスト SIMD 命令を、ARM や Thumb 命令と並列に実行できます。
- アドバンスト SIMD の値選択操作は、等価な VFP 比較と条件付き実行を使用するよりも効率的です。

3.3 Cortex-A9 NEON MPE 命令

Cortex-A9 NEON MPE でサポートされている命令と、各命令がアドバンスド SIMD と VFP のどちらの命令セットに属しているかを、表 3-1 に示します。各命令がサポートするデータ型には、次の略号が使用されています。

D	倍精度浮動小数点値
F	単精度浮動小数点値
H	半精度浮動小数点値
I	整数値
P	単一ビット係数の多項式
X	この操作はデータ表現に無関係です。

命令のエンコードと機能の詳細については、『*ARM アーキテクチャ リファレンスマニュアル*』を参照して下さい。

表 3-1 Cortex-A9 MPE 命令

名前	アドバンスド SIMD	VFP	説明
VABA	I	-	絶対値の差および累積
VABAL	I	-	ロング絶対値の差および累積
VABD	I、F	-	絶対値の差
VABDL	I	-	ロング絶対値の差
VABS	I、F	F、D	絶対値
VACGE	F	-	絶対値比較、以上
VACGT	F	-	絶対値比較、より大きい
VACLE	F	-	絶対値比較、以下
VACLT	F	-	絶対値比較、より小さい
VADD	I、F	F、D	加算
VADDHN	I	-	上位半分を返す加算および縮小
VADDL	I	-	ロング加算
VADDW	I	-	ワイド加算
VAND	X	-	ビット単位論理積
VBIC	I	-	ビット単位クリア
VBIF	X	-	FALSE 時ビット単位挿入

表 3-1 Cortex-A9 MPE 命令 (続き)

名前	アドバンスド SIMD	VFP	説明
VBIT	X	-	TRUE 時ビット単位挿入
VBSL	X	-	ビット単位の選択
VCEQ	I、F	-	比較、等しい
VCGE	I、F	-	比較、以上
VCLE	I、F	-	比較、以下
VCLS	I	-	先行符号ビットカウント
VCLT	I、F	-	比較、より小さい
VCLZ	I	-	先行ゼロカウント
VCMP	-	F、D	比較とフラグの設定
VCNT	I	-	セットされているビット数のカウント
VCVT	I、F、H	I、F、D、H	浮動小数点と 32 ビット整数型との間の変換
VDIV	-	F、D	除算
VDUP	X	-	複製
VEOR	X	-	ビット単位排他的論理和
VEXT	I	-	エレメントの抽出と連結
VHADD	I	-	加算半減
VHSUB	I	-	減算半減
VLD1	X	-	単一エレメント構造のロード
VLD2	X	-	2 エレメント構造のロード
VLD3	X	-	3 エレメント構造のロード
VLD4	X	-	4 エレメント構造のロード
VLDM	X	F、D	複数レジスタのロード
VLDR	X	F、D	単一レジスタのロード
VMAX	I、F	-	最大値
VMIN	I、F	-	最小値

表 3-1 Cortex-A9 MPE 命令 (続き)

名前	アドバンスド SIMD	VFP	説明
VMLA	I、F	F、D	積和演算
VMLS	I、F	F、D	積減算
VMLAL	I	-	ロング積和演算
VMLSL	I	-	ロング積減算
VMOV	X	F、D	レジスタまたはイミディエート移動
VMOVL	I	-	ロング移動
VMOVN	I	-	移動および縮小
VMRS	X	F、D	アドバンスド SIMD または VFP レジスタから ARM 計算エンジンへの移動
VMSR	X	F、D	ARM コアレジスタからアドバンスド SIMD または VFP への移動
VMUL	I、F、P	F、D	乗算
VMULL	I、F、P	-	ロング乗算
VMVN	X	-	ビット単位否定
VNEG	I、F	F、D	否定
VNMLA	-	F、D	否定積和演算
VNMLS	-	F、D	否定積減算
VNMUL	-	F、D	否定乗算
VORN	X	-	ビット単位否定論理和
VORR	X	-	ビット単位論理和
VPADAL	I	-	ペア単位ロング加算および累積
VPADD	I、F	-	ペア単位加算
VPADDL	I	-	ペア単位ロング加算
VPMAX	I、F	-	ペア単位最大値
VPMIN	I、F	-	ペア単位最小値
VPOP	X	F、D	スタックからポップ

表 3-1 Cortex-A9 MPE 命令 (続き)

名前	アドバンスド SIMD	VFP	説明
VPUSH	X	F、D	スタックにプッシュ
VQABS	I	-	飽和絶対値
VQADD	I	-	飽和加算
VQDMLAL	I	-	飽和ダブルロング積和演算
VQDMLSL	I	-	飽和ダブルロング積減算
VQDMULH	I	-	上位半分を返す飽和乗算 2 倍
VQDMULL	I	-	飽和ロング乗算 2 倍
VQMOVN	I	-	飽和移動および縮小
VQMOVUN	I	-	飽和移動および符号なし縮小
VQNEG	I	-	飽和否定
VQRDMULH	I	-	上位半分を返す飽和丸め付き乗算 2 倍
VQRSHL	I	-	飽和丸め付き左シフト
VQRSHRN	I	-	飽和丸め付き右シフトおよび縮小
VQRSHRUN	I	-	飽和丸め付き右シフトおよび符号なし縮小
VQSHL	I	-	飽和左シフト
VQSHLU	I	-	飽和符号なし左シフト
VQSHRN	I	-	飽和右シフトおよび縮小
VQSHRUN	I	-	飽和右シフトおよび符号なし縮小
VQSUB	I	-	飽和減算
VRADDHN	I	-	上位半分を返す丸め付き加算および縮小
VRECPE	I、F	-	逆数概算
VRECPS	F	-	逆数ステップ
VREV16	X	-	ハーフワードの反転
VREV32	X	-	ワードの反転
VREV64	X	-	ダブルワードの反転

表 3-1 Cortex-A9 MPE 命令 (続き)

名前	アドバンスド SIMD	VFP	説明
VRHADD	I	-	丸め付き加算半減
VRSHL	I	-	丸め付き左シフト
VRSHR	I	-	丸め付き右シフト
VRSHRN	I	-	丸め付き右シフトおよび縮小
VRSQRTE	I、F	-	逆数平方根概算
VRSQRTS	F	-	逆数平方根ステップ
VRSRA	I	-	丸め付き右シフトおよび累積
VRSUBHN	I	-	上位半分を返す丸め付き減算および縮小
VSHL	I	-	左シフト
VSHLL	I	-	左シフトロング
VSHR	I	-	右シフト
VSHRN	I	-	右シフトおよび縮小
VSLI	X	-	左シフトおよび挿入
VSQRT	-	F、D	平方根
VSRA	I	-	右シフトおよび累積
VSRI	X	-	右シフトおよび挿入
VST1	X	-	単一エレメント構造のストア
VST2	X	-	2 エレメント構造のストア
VST3	X	-	3 エレメント構造のストア
VST4	X	-	4 エレメント構造のストア
VSTM	X	F、D	複数レジスタのストア
VSTR	X	F、D	レジスタストア
VSUB	I、F	F、D	減算
VSUBHN	I	-	減算および縮小
VSUBL	I	-	ロング減算

表 3-1 Cortex-A9 MPE 命令 (続き)

名前	アドバンスド SIMD	VFP	説明
VSUBW	I	-	ワイド減算
VSWP	I	-	内容のスワップ
VTBL	X	-	テーブルルックアップ
VTBX	X	-	テーブル拡張
VTRN	X	-	転置
VTST	I	-	テストビット
VUZP	X	-	アンジップ
VZIP	X	-	ジップ

3.4 命令固有のスケジュール

命令の依存関係とメモリシステムの相互作用が複雑なため、すべての状況におけるすべての命令の正確なサイクルタイミングを簡潔に説明することは不可能です。以下のセクションで説明するタイミング情報は、ほとんどの場合に当てはまります。正確なタイミングが必要な場合は、システムとプロセッサ両方のサイクルに厳密なモデルを使用する必要があります。

3.4.1 命令のタイミング表

ここに示す表には、命令の各カテゴリについて有用なタイミング情報が記載されています。各列の定義は次のとおりです。

命令	命令のニーモニックは ARM UAL 形式です。等価な従来のニーモニックについては、『 <i>ARM アーキテクチャ リファレンスマニュアル</i> 』を参照して下さい。
形式	タイミング情報が記載されている命令のレジスタ形式。同じ行に複数の形式が記載されている場合、それらの形式すべてに、そのタイミング情報が適用されます。オペランドによって実行時間が異なる一部の事例では、オペランド型の情報が表に記載されています。例えば、VFP タイミング表の表 3-2（ページ 3-11）では、単精度と倍精度の演算について、それぞれ .F および .D の UAL 修飾子が記載されています。データ型指定子が省略されている場合、そのエントリに関係するすべての指定子について記載事項が適用されます。
サイクル数	特定の命令の実行に消費される発行サイクル数で、オペランドのインターロックが存在しない場合の、命令あたりの絶対最小サイクル数です。
ソース	ソースフィールドは、命令の発行が許可されるために、ソースオペランドが利用可能である必要がある実行サイクルを示します。ここに示されているカンマで区切った一覧は、形式フィールドのものと一致し、それぞれの値が適用されるレジスタを示します。 クワッド (Q) レジスタを含む単一の形式について 2 つの行がある場合、ソースレジスタの上位半分と下位半分を構成する各ダブル (D) レジスタは別々に処理されることを示しています。最初の行はクワッドレジスタの下位半分、2 番目の行は上位半分について、それぞれ同じ情報を示しています。
結果	結果フィールドは、演算の結果が利用可能になる実行サイクルを示します。この時点で、その結果を他の命令のソースオペランドとして、フォワードパスを使用して利用できます。ただし、一部の命令ペアでは、値がレジスタファイルに書き戻されるまで待つことが必要な場合があります。

ライトバック

ライトバックフィールドは、結果がレジスタファイルにコミットされる実行サイクルを示します。このサイクルから、命令の結果は他の任意の命令のソースオペランドとして利用可能です。

注

先行する命令によって特定の同じレジスタへのライトバックが発生する場合、その命令は発行されないことがあります。

ここでは、次の内容について説明します。

- 「VFP 命令のタイミング」
- 「アドバンスト SIMD の整数演算命令」 (ページ 3-14)
- 「アドバンスト SIMD の整数乗算命令」 (ページ 3-16)
- 「アドバンスト SIMD の整数シフト命令」 (ページ 3-18)
- 「アドバンスト SIMD の置換命令」 (ページ 3-19)
- 「アドバンスト SIMD の浮動小数点命令」 (ページ 3-20)
- 「アドバンスト SIMD のロード/ストア命令」 (ページ 3-22)

3.4.2 VFP 命令のタイミング

VFP 命令のタイミングを、表 3-2 に示します。

表 3-2 VFP 命令のタイミング

名前	形式	サイクル数	ソース	結果	ライトバック
VADD	.F Sd, Sn, Sm	1	-,1,1	4	4
VSUB	.D Dd, Dn, Dm				
VCVT	.F Sd, Sn .D Dd, Sn	1	-,1	4	4
VMUL	.F Sd, Sn, Sm	1	-,1,1	5	5
VNMUL	.D Dd, Dn, Dm	2	-,1,1	6	6
VMLA ⁴	.F Sd, Sn, Sm	1	-,1,1	8	8
VMLS	.D Dd, Dn, Dm	2	-,1,1	9	9
VNMLS					
VNMLA					
VABS	.F Sd, Sn	1	-,1	1	2
VNEG	.D Dd, Dn				

表 3-2 VFP 命令のタイミング (続き)

名前	形式	サイクル数	ソース	結果	ライトバック
VMOV	Rt, Sn	1	-,1	-	-
	Rt, Rt2, Dn	1	-,-,1	-	-
	Dd, Rt, Rt2	1	-,1,1	1	2
	Sd, RtSd, SnDd, Dn	1	-,1	1	2
	.F Sd, #imm .D Dd, #imm	1	-, -	1	2
VMRS	Rt, FPSCR	1	-,1	-	-
VDIV	.F Sd, Sn, Sm	10	-,1,1	15	15
	.D Dd, Dn, Dm	20	-,1,1	25	25
VSQRT	.F Sd, Sn, Sm	13	-,1,1	17	17
	.D Dd, Dn, Dm	28	-,1,1	32	32
VCMP	.F Sn, Sm	1	1,1	1	4
	.D Dn, Dm				

- a. 積和演算の後に乗算または別の積和演算が続く場合、最初の命令の結果によっては、両方の命令の間に同じ型とサイズの依存関係が存在する場合、プロセッサは特殊な乗算器と累算器のフォワードを使用することがあります。この特殊なフォワードは、最初の命令のサイクル 5 における結果が、2 番目の命令のサイクル 4 で累算器にフォワードされるため、複数の命令を連続して発行可能なことを意味します。命令のサイズと型が一致しない場合、Dd または Qd はサイクル 3 で必要となります。これは、積和演算命令 VMLA、VMLS、VQDMLA、VQDMLS、および乗算命令の VMUL と VQDMUL の組み合わせに適用されます。

3.4.3 VFP ロード / ストア命令のタイミング

VFP ロード / ストア命令のタイミングを、表 3-3 (ページ 3-13) に示します。

SPレジスタの列は、操作対象となる単精度レジスタ数を示します。複数ロード/ストア操作の場合、一覧に含まれているレジスタの数は、レジスタ数が偶数の場合は $2p$ 、奇数の場合は $2p + 1$ です。ロード/ストア操作で使用されるアドレスが64ビットアラインドでない場合、1サイクルが追加されることがあります。

表 3-3 VFP ロード/ストア命令のタイミング

名前	形式	64ビットアラインド	SPレジスタ	サイクル数	ソース ^a	結果	ライトバック
VLDR	.F Sd, []	-	1	1	-	1	1
VLDR	.D Dd, []	はい	2	1	-	1	1
		いいえ	2	2	-	1, 2	1, 2
VLDM	.F Rn, {...}	はい	$2p$	p	-	1, 2, ..., p	1, 2, ..., p
			$2p + 1$	$p + 1$	-	1, 2, ..., $p + 1$	1, 2, ..., $p + 1$
		いいえ	$2p$	$p + 1$	-	1, 2, ..., $p + 1$	1, 2, ..., $p + 1$
			$2p + 1$	$p + 1$	-	1, 2, ..., $p + 1$	1, 2, ..., $p + 1$
VLDM	.D Rn, {...}	はい	$2p$	p	-	1, 2, ..., p	1, 2, ..., p
		いいえ	$2p$	$p + 1$	-	1, 2, ..., $p + 1$	1, 2, ..., $p + 1$
VSTR	.F Sd, []	-	1	1	-	-	-
VSTR	.D Dd, []	はい	2	1	-	-	-
		いいえ	2	2	-	-	-
VSTM	.F Rn, {...}	はい	$2p$	p	-	-	-
			$2p + 1$	$p + 1$	-	-	-
		いいえ	$2p$	$p + 1$	-	-	-
			$2p + 1$	$p + 1$	-	-	-
VSTM	.D Rn, {...}	はい	$2p$	p	-	-	-
		いいえ	$2p$	$p + 1$	-	-	-

a. ストア命令は、オペランドが計算される前に発行可能です。

3.4.4 アドバンスト SIMD の整数演算命令

アドバンスト SIMD の整数演算命令のタイミングを、表 3-4 に示します。

表 3-4 アドバンスト SIMD の整数演算命令のタイミング

名前	形式	サイクル数	ソース	結果	ライトバック
VADD	Dd, Dn, Dm	1	-,2,2	3	6
VAND	Qd, Qn, Qm				
VORR					
VEOR					
VBIC					
VORN					
VSUB	Dd, Dn, Dm Qd, Qn, Qm	1	-,2,1	3	6
VADDL	Qd, Dn, Dm	1	-,1,1	3	6
VSUBL					
VADDW	Qd, Qn, Dm	1	-,2,1	3	6
VSUBW					
VHADD	Dd, Dn, Dm	1	-,2,2	4	6
VRHADD	Qd, Qn, Qm				
VQADD					
VTST					
VADH	Dd, Qn, Qm	1	-,2,2	4	6
VRADH					
VSBH	Dd, Qn, Qm	1	-,2,1	4	6
VRSBH					
VHSUB	Dd, Dn, Dm	1	-,2,1	4	6
VQSUB	Qd, Qn, Qm				
VABD					
VCEQ					
VCGE					
VCGT					
VMAX					
VMIN					
VPMAX	Dd, Dn, Dm	1	-,2,1	4	6
VPMIN					

表 3-4 アドバンスド SIMD の整数演算命令のタイミング (続き)

名前	形式	サイクル数	ソース	結果	ライトバック
VNEG	Dd, Dm Qd, Qm	1	-,1	3	6
VQNEG	Dd, Dm	1	-,1	4	6
VQABS	Qd, Qm				
VABDL	Qd, Dn, Dm	1	-,2,1	4	6
VABS	Dd, Dm Qd, Qm	1	-,2	4	6
VCEQ	Dd, Dm, #imm	1	-,2,-	4	6
VCGE	Qd, Qm, #imm				
VCGT					
VCLE					
VCLT					
VPADD	Dd, Dn, Dm	1	-,1,1	3	6
VPADDL	Dd, Dn Qd, Qn	1	-,1	3	6
VMVN	Dd, DmQd, Qm	1	-,2	3	6
VCLS	Dd, Dm	1	-,2	3	6
VCLZ	Qd, Qm	2	-,2	3	6
VCNT			-,3	4	7
VMOV	Dd, #imm	1	-, -	3	6
VMVN	Qd, #imm				
VORR	Dd, #imm	1	2, -	3	6
VBIC	Qd, #imm				
VBIT	Dd, Dn, Dm	1	2,2,2	3	6
VBIF	Qd, Qn, Qm	2	2,2,2	3	6
VBSL			3,3,3	4	7

表 3-4 アドバンスド SIMD の整数演算命令のタイミング (続き)

名前	形式	サイクル数	ソース	結果	ライトバック
VABA	Dd, Dn, Dm	1	3,2,1	6	6
	Qd, Qn, Qm	2	3,2,1 4,3,2	6	6
VABAL	Qd, Dn, Dm	1	3,2,1	6	6
VPADAL	Dd, Dm	1	3,1	6	6
	Qd, Qm				

3.4.5 アドバンスド SIMD の整数乗算命令

アドバンスド SIMD の整数乗算命令のタイミング動作を、表 3-5 に示します。

表 3-5 アドバンスド SIMD の整数乗算命令

名前	形式	サイクル数	ソース	結果	ライトバック
VMUL	.8 Dd, Dn, Dm	1	-,2,2	6	6
VQDMLH	.16 Dd, Dn, Dm				
VQRDMLH	.32 Dd, Dn, Dm	2	-,2,1	7	7
	.32 Qd, Qn, Qm	4	-,2,1 -,4,3	7 9	7 9
VMULL	.8 Qd, Dn, Dm	1	-,2,2	6	6
VQDMULL	.16 Qd, Dn, Dm				
	.32 Qd, Dn, Dm	2	-,2,1	7	7
VMLA	.8 Dd, Dn, Dm	1	3,2,2	6	6
VMLS	.16 Dd, Dn, Dm				
	.8 Qd, Qn, Qm	2	3,2,2	6	6
	.16 Qd, Qn, Qm		4,3,3	7	7
	.32 Dd, Dn, Dm	2	3,2,1	7	7
	.32 Qd, Qn, Qm	4	3,2,1 5,4,3	7 9	7 9

表 3-5 アドバンスド SIMD の整数乗算命令（続き）

名前	形式	サイクル数	ソース	結果	ライトバック
VMLAL	.8 Qd, Dn, Dm	1	3,2,2	6	6
VMLS	.16 Qd, Dn, Dm				
VQDMLAL	.32 Qd, Dn, Dm	2	3,2,1	7	7
VQDMLSL					
VMUL	.16 Dd, Dn, Dm [x]	1	-,2,1	6	6
VQDMLH	.16 Qd, Qn, Dm [x]	2	-,2,1	6	6
VQRDMLH			-,3,1	7	7
	.32 Dd, Dn, Dm [x]	2	-,2,1	7	7
	.32 Qd, Qn, Qm [x]	4	-,2,1 -,4,1	7 9	7 9
VMULL	.16 Qd, Dn, Dm [x]	1	-,2,1	6	6
VQDMULL	.32 Qd, Dn, Dm [x]	2	-,2,1	7	7
VMLA	.16 Dd, Dn, Dm [x]	1	3,2,1	6	6
VMLS	.16 Qd, Qn, Dm [x]	2	3,2,1	6	6
	.32 Dd, Dn, Dm [x]	2	3,2,1	7	7
	.32 Qd, Qn, Dm [x]	4	3,2,1 5,2,1	7 9	7 9
VMLAL	.16 Qd, Dn, Dm [x]	1	3,2,1	6	6
VMLS	.32 Qd, Dn, Dm [x]	2	3,2,1	7	7
VQDMLAL					
VQDMLSL					

3.4.6 アドバンスト SIMD の整数シフト命令

アドバンスト SIMD の整数シフト命令のタイミングを、表 3-6 に示します。

表 3-6 アドバンスト SIMD の整数シフト命令のタイミング

名前	形式	サイクル数	ソース	結果	ライトバック
VSHR	Dd, Dm, #imm	1	-,1,-	3	6
VSHL	Qd, Qm, #imm				
VQSHL	Dd, Dm, #imm	1	-,1,-	4	6
VRSHR	Qd, Qm, #imm				
VSHRN	Dd, Qm, #imm	1	-,1,-	3	6
VMOVN					
VQSHRN	Dd, Qm, #imm	1	-,1,-	4	6
VQMOVN					
VQSHRN					
VQRSHR					
VSHL	Qd, Dm, #imm	1	-,1,-	3	6
VMOVL	Qd, Dm	1	-,1	3	6
VSLI	Dd, Dm, #imm	1	1,1,-	3	6
VSRI	Qd, Qm, #imm	2	1,1,- 2,2,-	3 4	6 7
VSHL	Dd, Dm, Dn	1	-,1,1	3	6
	Qd, Qm, Qn	2	-,1,1 -,2,2	3 4	6 7
VQSHL	Dd, Dm, Dn	1	-,1,1	4	6
VRSHL	Qd, Qm, Qn	2	-,1,1 -,2,2	5	7
VQRSHL					
VSRA	Dd, Dm, #imm	1	3,1,-	4	6
VRSRA	Qd, Qm, #imm	1	3,1,-	6	6

3.4.7 アドバンスト SIMD の置換命令

アドバンスト SIMD の置換命令のタイミング動作を、表 3-7 に示します。

表 3-7 アドバンスト SIMD の置換命令のタイミング

名前	形式	サイクル数	ソース	結果	ライトバック
VDUP	Dd, Dm[x]	1	-,1	2	6
VTRN	Qd, Qm[x]				
VSWP	Dd, Dm	1	1,1	2	6
	Qd, Qm	2	1,1 2,2	2 3	6 7
VZIP	Dd, Dm	1	1,1	2	6
	Qd, Qm	3	1,1 2,2	3,4	7,8
VUZP	Dd, Dm	1	1,1	2	6
	Qd, Qm	3	1,2	3	7
VREV	Dd, DmQd, Qm	1	-,1	2	6
VEXT	Dd, Dn, Dm, #imm	1	-,1,1,-	2	6
	Qd, Qn, Qm, #imm	2	-,1,2,-	3	7
VTBL	Dd, {Dn}, Dm	2	-,2,1	3	7
	Dd, {Dn, Dn1}, Dm	2	-,2,2,1	3	7
	Dd, {Dn, Dn1, Dn2}, Dm	3	-,2,2,3,1	4	8
	Dd, {Dn, Dn1, Dn2, Dn3}, Dm	3	-,2,2,3,3,1	4	8
VTBX	Dd, {Dn}, Dm	2	1,2,1	3	7
	Dd, {Dn, Dn1}, Dm	2	1,2,2,1	3	7
	Dd, {Dn, Dn1, Dn2}, Dm	3	1,2,2,3,1	4	8
	Dd, {Dn, Dn1, Dn2, Dn3}, Dm	3	1,2,2,3,3,1	4	8

3.4.8 アドバンスト SIMD の浮動小数点命令

アドバンスト SIMD の浮動小数点命令を、表 3-8 に示します。

表 3-8 アドバンスト SIMD の浮動小数点命令

名前	形式	サイクル数	ソース	結果	ライトバック
VADD	Dd, Dn, Dm	1	-,2,2	5	6
VSUB	Qd, Qn, Qm	2	-,2,2	5	6
VABD			-,3,3	6	7
VMUL					
VCEQ					
VCGE					
VCGT					
VACGE					
VACGT					
VMAX					
VMIN					
VABS	Dd, Dm	1	-,2	5	6
VNEG	Qd, Qm	2	-,2	5	6
VRECPE			-,3	6	7
VRSQRTE					
VCVT					
VCEQ	Dd, Dm, #imm	1	-,2	5	6
VCGE	Qd, Qm, #imm	2	-,2	5	6
VCGT			-,3	6	7
VCLE					
VCLT					
VPADD	Dd, Dn, Dm	1	-,1,1	5	6
VPMAX					
VPMIN					
VMUL	Dd, Dn, Dm[x]	1	-,2,1	5	6
	Qd, Qn, Dm[x]	2	-,2,1	5	6
			-,3,1	6	7

表 3-8 アドバンスド SIMD の浮動小数点命令（続き）

名前	形式	サイクル数	ソース	結果	ライトバック
VMLA	Dd, Dn, Dm	1	3,2,2	9	10
VMLS	Qd, Qn, Qm	2	3,2,2	9	10
			4,3,3	10	11
	Dd, Dn, Dm[x]	1	3,2,1	9	10
			4,3,1	10	11
VRECPS	Dd, Dn, Dm	1	-,2,2	9	10
VRSQRTS	Qd, Qm, Qn	2	-,2,2	9	10
			-,3,3	10	11

3.4.9 アドバンスド SIMD のロード / ストア命令

アドバンスド SIMD のロード / ストア命令のタイミングを、表 3-9 に示します。

表 3-9 の値は、MPE 実行ユニット内で必要となる発行サイクル数に対応しています。Cortex-A9 整数プロセッサで必要となるサイクル数は、NEON のデータロード / ストアでオーバーラップする 64 ビットアラインド ダブルワードの数と同じです。

表 3-9 アドバンスド SIMD のロード / ストア命令

名前	形式	サイクル数	ソース	結果	ライトバック
VLD1	{Dd}, []	2	-	2	7
	{Dd}, [@]	1	-	1	6
	{Dd, Dd1}, []	2	-, -	2, 2	7, 7
	{Dd, Dd1}, [@]	1	-, -	1, 1	6, 6
	{Dd, Dd1, Dd2}, []	3	-, -, -	2, 2, 3	7, 7, 8
	{Dd, Dd1, Dd2}, [@]	2	-, -, -	1, 1, 2	6, 6, 7
	{Dd, Dd1, Dd2, Dd3}, []	3	-, -, -, -	2, 2, 3, 3	7, 7, 8, 8
	{Dd, Dd1, Dd2, Dd3}, [@]	2	-, -, -, -	1, 1, 2, 2	6, 6, 7, 7
	{Dd[x]}, []	3	1	4	8
	{Dd[x]}, [@]	2	1	3	7
	{Dd[]}, []	2	-	3	7
	{Dd[]}, [@]	1	-	2	6
	{Dd[], Dd1[]}, []	2	-, -	3, 3	7, 7
	{Dd[], Dd1[]}, [@]	1	-, -	2, 2	6, 6

表 3-9 アドバンスド SIMD のロード / ストア命令 (続き)

名前	形式	サイクル数	ソース	結果	ライトバック
VLD2	{Dd, Dd1}, []	2	-, -	3, 3	7, 7
	{Dd, Dd1}, {#}	1	-, -	2, 2	6, 6
	{Dd, Dd1, Dd2, Dd3}, []	3	-, -, -, -	3, 4, 3, 4	7, 8, 7, 8
	{Dd, Dd1, Dd2, Dd3}, {#}	2	-, -, -, -	2, 3, 2, 3	6, 7, 6, 7
	{Dd[x], Dd1[x]}, []	3	1, 1	4, 4	8, 8
	{Dd[x], Dd1[x]}, {#}	2	1, 1	3, 3	7, 7
	{Dd[], Dd1[]}, []	2	-, -	3, 3	7, 7
	{Dd[], Dd1[]}, {#}	1	-, -	2, 2	6, 6
VLD3	{Dd, Dd1, Dd2}, []	4	-, -, -	4, 4, 5	8, 8, 9
	{Dd, Dd1, Dd2}, {#}	3	-, -, -	3, 3, 4	7, 7, 8
	{Dd[x], ..., Dd2[x]}, []	5	1, 1, 2	5, 5, 6	9, 9, 10
	{Dd[], Dd1[], Dd2[]}, []	3	-, -, -	3, 3, 4	7, 7, 8
VLD4	{Dd, Dd1, Dd2, Dd3}, []	4	-, -, -, -	4, 4, 5, 5	8, 8, 9, 9
	{Dd, Dd1, Dd2, Dd3}, {#}	3	-, -, -, -	3, 3, 4, 4	7, 7, 8, 8
	{Dd[x], ..., Dd3[x]}, []	5	1, 1, 2, 2	5, 5, 6, 6	9, 9, 10, 10
	{Dd[x], ..., Dd3[x]}, {#}	4	1, 1, 2, 2	4, 4, 5, 5	8, 8, 9, 9
	{Dd[], ..., Dd3[]}, []	3	-, -, -, -	3, 3, 4, 4	7, 7, 8, 8
	{Dd[], ..., Dd3[]}, {#}	2	-, -, -, -	2, 2, 3, 3	6, 6, 7, 7
VST1	{Dd}, []	2	1	-	-
	{Dd}, {#}	1	1	-	-
	{Dd[x]}, []	2	1	-	-
	{Dd[x]}, {#}	1	1	-	-

表 3-9 アドバンスド SIMD のロード / ストア命令 (続き)

名前	形式	サイクル数	ソース	結果	ライトバック
VST2	{Dd, Dd1}, []	2	1,1	-	-
	{Dd, Dd1}, [@]	1	1,1	-	-
	{Dd[x], Dd1[x]}, []	2	1,1	-	-
	{Dd[x], Dd1[x]}, [@]	1	1,1	-	-
VST3	{Dd, Dd1, Dd2}, []	3	1,1,2	-	-
	{Dd, Dd1, Dd2}, [@]	2	1,1,2	-	-
	{Dd[x], ..., Dd2[x]}, []	3	1,1,2	-	-
VST4	{Dd, Dd1, Dd2, Dd3}, []	3	1,1,2,2	-	-
	{Dd, Dd1, Dd2, Dd3}, [@]	2	1,1,2,2	-	-
	{Dd[x], ..., Dd3[x]}, []	3	1,1,2,2	-	-
	{Dd[x], ..., Dd3[x]}, [@]	2	1,1,2,2	-	-

付録 A

リビジョン

この付録では、本書の各版の技術的な相違点について説明します。

表 A-1 A 版と B 版の相違点

変更内容	場所
セキュリティ拡張機能の実装がオプションであることを示す文章の削除	本書全体
VSWP 命令用の独立したエントリの作成	表 3-7 (ページ 3-19)
VMLA、VMLS、VRECPS、VRSQRTS 命令の値の更新	表 3-8 (ページ 3-20)

表 A-2 B 版と C 版の相違点

変更内容	場所
アドバンスド SIMD および VFP 拡張機能を稼働するためのサンプルコードを、ARM 統一アセンブリ言語に書き直し	例 2-1 (ページ 2-3)
FPSCR ビットの割り当て表の更新	表 2-10 (ページ 2-15)

表 A-3 C 版と D 版の相違点

変更内容	場所
バージョンの値を 2 に更新	表 2-7 (ページ 2-13)
FPSCR の SBZ/WI ビットを、UNK/SBZP という名前に変更	表 2-10 (ページ 2-15) および図 2-4 (ページ 2-15)
SBZ/WI ビットを、UNK/SBZ という名前に変更	表 2-11 (ページ 2-18) および図 2-5 (ページ 2-18)
2 つの VMLA 命令の特別な動作に関する脚注の追加	表 3-2 (ページ 3-11)
脚注の書き直し	表 3-3 (ページ 3-13)
タイミングに関する段落の追加	表 3-9 (ページ 3-22)
表の書式修正	表 3-3 (ページ 3-13) 表 3-4 (ページ 3-14) 表 3-5 (ページ 3-16) 表 3-6 (ページ 3-18) 表 3-7 (ページ 3-19) 表 3-8 (ページ 3-20) 表 3-9 (ページ 3-22)

表 A-4 D 版と E 版の相違点

変更内容	場所
技術的変更なし	-

表 A-5 E 版と F 版の相違点

変更内容	場所
コード例に ISB を追加	例 2-1 (ページ 2-3)
FPEXC ビットの説明を、FPU テクニカルリファレンスマニュアルの説明と一致するよう修正	表 2-11 (ページ 2-18)

用語集

この用語集では、ARM Limited の発行する技術文書で使用されている用語の一部について説明します。

- ARM 状態** ARM (32 ビット) ワードアラインド命令を実行しているプロセッサは、ARM 状態で動作しています。
- ARM 命令** ARM プロセッサが実行する操作を示すワード。ARM 命令はワードアラインしている必要があります。
- CoreSight** 完全なシステム オンチップ (SoC) のモニタ、トレース、デバッグを行うためのインフラストラクチャ。
- DP 命令** コプロセッサデータ処理命令。VFP11 コプロセッサの CDP 命令は、算術演算命令、FCPY 命令、FABS 命令、FNEG 命令です。
算術演算命令も参照。
- Flush-to-Zero モード** このモードでは、VFP11 コプロセッサは次の値を正の 0 として扱います。
- 入力精度で表現できる範囲より小さい算術演算入力。
 - 丸め前に、入力精度で表現できる範囲より小さい、0 以外の算術演算の結果。

VFP11 コプロセッサは、これらの値を非正規化値として解釈したり、非正規化値に変換したりしません。

入力精度での非正規化値の範囲は、 $-2^{E_{min}} < x < 0$ または $0 < x < 2^{E_{min}}$ です。

IEEE 754 規格

IEEE Standard for Binary Floating-Point Arithmetic, ANSI/IEEE Std. 754-1985。浮動小数点システムに関するデータ型、正常な動作、例外のタイプと処理、エラーバウンドを規定している規格です。ほとんどのプロセッサが、ハードウェア単体またはハードウェアとソフトウェアの組み合わせによって、この規格に準拠するように構築されています。

NaN

非数。浮動小数点形式でエンコードされ、指数フィールドが最大値で、仮数が 0 以外である記号エンティティ。SNaN は仮数の最上位ビットが 0 で、オペランドとして使用された場合に無効オペランド例外を引き起こします。QNaN は仮数の最上位ビットが 1 で、ほとんどすべての算術演算で例外を発生せずに伝播されます。

SBO

常に 1 参照。

SBZ

常に 0 参照。

SBZP

常に 0 または保持参照。

Thumb 状態

Thumb (16 ビット) ハーフワードアラインド命令を実行しているプロセッサは、Thumb 状態で動作しています。

Thumb 命令

ARM プロセッサが Thumb 状態で実行する動作を指定するハーフワード。Thumb 命令は、ハーフワードアラインドの必要があります。

UNP

予測不能参照。

アーキテクチャ

プロセッサとその付属コンポーネントを特徴付け、同様の特徴を持つデバイスを、ハーバードアーキテクチャ、命令セットアーキテクチャ、ARMv6 アーキテクチャなどのように、その動作を記述するときにグループ化することを可能にする、ハードウェアおよびソフトウェアの編成。

アドレッシングモード

命令で使用する値を生成するために、多くの命令で共有される機構。ARM アドレッシングモードのうち 4 つでは、値としてメモリアドレスが生成されます。これは、アドレッシングモードの伝統的な使用方法です。5 番目のアドレッシングモードでは、データ処理命令のオペランドとして使用される値が生成されます。

アポート

メモリアクセスに関連する値が無効であることをコアに通知する機構。アポートは、無効な命令またはデータメモリへのアクセスを実行した結果として、外部または内部のメモリシステムで発生する可能性があります。アポートは、プリフェッチアポートとデータアポート、内部アポートと外部アポートに分類されます。

アラインド	データサイズを定義しているバイト数で割り切れるアドレスに格納されているデータ項目を、アラインド、またはアラインしていると呼びます。アラインしているワードとハーフワードのアドレスは、それぞれ4と2で割り切れます。したがって、ワードアラインドとハーフワードアラインドという用語は、それぞれ4と2で割り切れるアドレスを意味します。
アンアラインド	データサイズを定義するバイト数で割り切れないアドレスに保存されているデータ項目を、アンアラインドまたはアラインしていないと呼びます。例えば、4で割り切れないアドレスに保存されているワードはアンアラインドです。
仮数	2進浮動小数点数の構成要素で、暗黙の2進小数点の左側にある明示的または暗黙的な先頭ビットと、右側にある小数フィールドとで構成されます。
コプロセッサ	メインプロセッサを補完するプロセッサ。メインプロセッサが実行できない付加機能を実行します。通常は、浮動小数点算術演算、信号処理、メモリ管理などに使用されます。
サポートコード	ハードウェアを補完し、IEEE 754規格との互換性を提供するために使用する必要があるソフトウェア。サポートコードには、サポートされていない入力による除算や、ハードウェアの範囲を越えた動作や例外を引き起こす可能性のある入力など、サポート済みの機能を実行するルーチンのライブラリが含まれています。サポートコードには、IEEE 754規格に従って例外状態を処理する例外ハンドラのセットが含まれています。
実装固有	動作がアーキテクチャで定義されていないが、実装ごとに文書化する必要がないことを意味します。使用可能な実装オプションが多数あり、選択したオプションによってソフトウェアの互換性に影響がない場合に使用されます。
実装定義	動作がアーキテクチャで定義されておらず、個別の実装によって定義や文書化が行われます。
条件付き実行	条件コードフラグが、命令の実行開始時に該当する条件がTRUEであることを示している場合は、命令が正常に実行されます。それ以外の場合、命令は何も実行しません。
条件フィールド	命令が実行可能な条件を指定する、命令内の4ビットのフィールド。
ショートベクタ演算	複数のデスティネーションレジスタを使用し、各デスティネーションの結果の生成に複数のソースレジスタを使用する可能性がある、VFP コプロセッサの動作。
スカラ演算	単一のソースレジスタとデスティネーションレジスタを使用する、VFP コプロセッサ演算。 ベクタ演算も参照。
ストライド	Stride フィールド (FPSCR[21:20]) は、ショートベクタ演算でレジスタアドレスに適用される増分を指定します。ストライドが00であれば、指定される増分は+1で、ショートベクタ演算の繰り返しごとに各ベクタレジスタが1ずつインクリメントされます。ストライドが11であれば、+2のインクリメントを意味します。

- 制御ビット** プログラムステータスレジスタの最下位 8 ビット。制御ビットは、例外が発生したときに変化します。プロセッサが特権モードの場合にのみ、ソフトウェアから変更できます。
- ダブルワード** 64 ビットのデータ項目。特に指定のない限り、その内容は符号なし整数とみなされます。
- ダブルワードアラインド** メモリアドレスが 8 で割り切れるデータ項目。
- 常に 0 (SBZ)** ソフトウェアで 0 (ビットフィールドの場合はすべてのビットに 0) を書き込む必要があります。1 を書き込んだ場合、結果は予測不能です。
- 常に 0 または保持 (SBZP)** ソフトウェアで 0 (ビットフィールドの場合はすべてのビットに 0) を書き込むか、同じプロセッサの同じフィールドから以前に読み出した値をそのまま書き戻して保持する必要があります。
- 常に 1 (SBO)** ソフトウェアで 1 (ビットフィールドの場合はすべてのビットに 1) を書き込む必要があります。0 を書き込んだ場合、結果は予測不能です。
- デフォルト NaN モード** 結果が NaN になった原因にかかわらず、結果が NaN になるすべての演算で、デフォルト NaN が返されるモード。このモードは IEEE 754 規格に準拠していますが、演算への入力 NaN に含まれていたすべての情報が失われることを意味します。
- トラップ** VFP コプロセッサに例外状態が発生し、FPSCR レジスタの対応する例外イネーブルビットがセットされている場合。ユーザトラップハンドラが実行されます。
- 入力例外** 指定された動作について、1 つまたは複数のオペランドがハードウェアでサポートされていない場合の VFP 例外条件。この動作は、サポートコードにバウンスされて処理されます。
- ハーフワード** 16 ビットのデータ項目。
- ハーフワードアラインド** アラインド [参照](#)。
- 倍精度数値** 2 つの 32 ビットワードで構成されます。これら 2 つのワードはメモリ上で連続していて、いずれもワードアラインしている必要があります。IEEE 754-1985 規格に従って、基本倍精度浮動小数点数として解釈されます。

バウンス	VFP コプロセッサは、ARM プロセッサに対する有効な VFP 命令の受け付けの通知に失敗したときに、命令をバウンスします。この動作によって、ARM プロセッサによる未定義命令処理が開始されます。VFP コプロセッサによって例外または未サポートとして検出された命令を完了するために、VFP サポートコードが呼び出されます。 トリガ命令、潜在的例外命令、例外状態も参照。
非正規化値	0 を除く、範囲 $(-2^{E_{min}} < x < 2^{E_{min}})$ の値。IEEE 754 規格での単精度および倍精度オペランドの形式では、非正規化値は指数が 0 で、仮数フィールドが 0 以外です。IEEE 754 規格では、非正規化値オペランドの生成と操作は、通常のオペランドと同じ精度で行う必要があります。
不正命令	アーキテクチャで未定義の命令。
プロセッサ	コンピュータ命令を使用してデータを処理するために必要な、コンピュータシステムの回路。プロセッサは、マイクロプロセッサの略称です。完全に機能する最小のコンピュータシステムを作成するには、クロックソース、電源、メインメモリも必要です。
ベースレジスタ	命令のアドレス計算の基準値を保持するために、ロード/ストア命令で指定されるレジスタ。命令とそのアドレッシングモードによっては、メモリに送られる仮想アドレスを形成するために、ベースレジスタの値にオフセットを加算または減算できます。
ベクタ演算	複数のデスティネーションレジスタを使用し、各デスティネーションの結果の生成に異なるソースレジスタを使用する可能性がある、VFP コプロセッサの動作。 スカラ演算も参照。
丸めモード	IEEE 754 規格では、すべての計算を、精度が無限であるかのように実行する必要があります。例えば、2つの単精度値の乗算では、仮数のビット数の 2 倍まで、正確に仮数を計算する必要があります。この値をデスティネーションの精度で表すために、仮数の丸めを頻繁に実行する必要があります。IEEE 754 規格では、4つの丸めモードが指定されています。 近似値への丸めモードでは、仮数の最下位ビットよりも下の値を、より近い側に丸め、値が中間である場合は最下位ビットをクリアして偶数に切り上げを行うことによって、結果が丸められます。 0 への丸めモードでは、仮数の右側のすべてのビットが必ず切り捨てられます。このモードは、C 言語、C++ 言語、Java 言語の整数変換で使用されます。 正の無限大への丸めモードと負の無限大への丸めモードは、区間演算で使用されます。
未サポート値	VFP コプロセッサのハードウェアでは処理されず、サポートコードにバウンスされて完了される特定のデータ値。このようなデータには、無限大、NaN、非正規化値、0 が含まれます。これらの値をハードウェアで完全にまたは部

分的にサポートするか、サポートコードにその処理の完了をゆだねるかは、実装で選択できます。未サポートデータの処理から発生したすべての例外は、対応する例外イネーブルビットがセットされている場合に、ユーザコードにトラップされます。

- 未定義** 未定義命令トラップを生成する命令を指します。ARM 例外の詳細については、『ARM アーキテクチャ リファレンスマニュアル』を参照して下さい。
- 無限大** 無限大を表す IEEE 754 規格の形式では、指数はその精度での最大値であり、仮数はすべて 0 です。
- メモリバンク** インターリーブされているメモリにおいて、並列にいくつかに分割されているメモリのうちの 1 つで、通常は 1 ワード幅です。これによって、一度に単一ワードではなく、複数ワードを読み書きできます。すべてのメモリバンクは同時にアドレス指定され、バンクイネーブル信号またはチップセレクト信号によって、アクセスされるバンクが転送ごとに決定されます。連続したワードアドレスへアクセスすると、連続したバンクへのアクセスが発生します。これによって、バンクアクセスに関連する遅延は隣接バンクへのアクセス中に発生するため、メモリ転送が高速化されます。
- 予測不能** 読み出しの場合は、この位置から読み出しによって返されるデータが予測不能なことを意味します。データはどのような値にもなり得ます。書き込みの場合は、この位置への書き込みによって予測不能な動作が発生するか、デバイスの構成に予測不能な変化が発生することを意味します。予測不能な命令によって、プロセッサまたはシステムのいずれかの部分に停止やハングが発生しないようにする必要があります。
- 読み出し** 読み出しは、ロードの意味を持つメモリ操作として定義されます。ARM 命令の LDM、LDRD、LDC、LDR、LDRT、LDRSH、LDRH、LDRSB、LDRB、LDRBT、LDREX、RFE、STREX、SWP、SWPB と、Thumb 命令の LDM、LDR、LDRSH、LDRH、LDRSB、LDRB、POP が該当します。
- ハードウェアで高速化される Java バイトコードの場合は、Java スタックの状態と Java ハードウェアアクセラレーションの実装によっては、大量の読み出しが発生する可能性があります。
- 予約** 制御レジスタまたは命令の形式に含まれているフィールドが実装で定義される、または 0 ではない場合に予測不能な結果が引き起こされる場合、そのフィールドは予約と記載されています。これらのフィールドは、アーキテクチャの将来の拡張に備えて予約される場合と、実装固有の場合があります。実装で使用されないすべての予約ビットは、0 として読み書きする必要があります。
- 例外** プログラムの実行に割り込む必要があるほど重大であると判断された、フォールトまたはエラーイベント。例として、無効なメモリアクセス、外部割り込み、未定義命令の実行などが挙げられます。例外が発生すると、通常のプログラムフローが中断され、対応する例外ベクタで実行が再開されます。例外ベクタには、例外を処理する割り込みハンドラの最初の命令が含まれています。

例外が可能な場合	FPSCR の例外イネーブルビットがセットされると、対応する例外が可能になります。可能である例外が発生すると、ユーザハンドラへのトラップが取得されます。例外状態を引き起こした動作は、IEEE 754 規格で定義されている結果を生成するために、サポートコードにバウンズできます。その後で、ユーザトラップハンドラに例外が報告されます。
例外が不可能な場合	FPSCR の例外イネーブルビットがクリアされている場合、そのビットに対応する例外は不可能となります。このような例外に対しては、返される結果が IEEE 754 規格で定義されています。例外状態を引き起こした動作は、IEEE 754 規格で定義されている結果を生成するために、サポートコードにバウンズできます。例外は、ユーザトラップハンドラに報告されません。
例外状態	潜在的な例外命令が発行されると、VFP11 コプロセッサは、EX ビット (FPEXC[31]) をセットし、FPINST レジスタ内の潜在的例外命令のコピーをロードします。命令がショートベクタ演算の場合は、FPINST のレジスタフィールドが、潜在的な例外が発生した繰り返しを指すように変更されます。例外状態では、VFP11 コプロセッサに対するトリガ命令の発行によって、バウンズが実行されます。 バウンズ、潜在的例外命令、トリガ命令も参照。
例外処理ルーチン	割り込みハンドラ参照。
ロード / ストアアーキテクチャ	データ処理操作が、メモリの内容に対して直接ではなく、レジスタの内容に対してのみ行われるプロセッサアーキテクチャ。
ワード	32 ビットのデータ項目。
ワードアラインド	アラインド参照。
割り込みハンドラ	割り込みが発生したときに、プロセッサの制御が渡されるプログラム。

