

## 付録1 Thumb-2命令

### ●16ビット命令 (1/2)

機能	動作	ニーモニック
四則演算命令	レジスタの値とCフラグをレジスタの値に加算	ADC <Rd>, <Rm>
	3ビットイミディエイト値をレジスタに加算	ADD <Rd>, <Rn>, #<immed_3>
	8ビットイミディエイト値をレジスタに加算	ADD <Rd>, #<immed_8>
	下位レジスタの値を下位レジスタの値に加算	ADD <Rd>, <Rn>, <Rm>
	上位レジスタの値を下位または上位レジスタの値に加算	ADD <Rd>, <Rm>
	PC 相対4 × (8ビットイミディエイト値) でレジスタに加算	ADD <Rd>, PC, #<immed_8> * 4
	SP 相対4 × (8ビットイミディエイト値) でレジスタに加算	ADD <Rd>, SP, #<immed_8> * 4
	4 × (7ビットイミディエイト値) をSPに加算	ADD SP, #<immed_7> * 4
	レジスタの値を乗算	MUL <Rd>, <Rm>
	レジスタの値を負 (ネガティブ、2の補数) にしてレジスタに保存	NEG <Rd>, <Rm>
	レジスタの値とCフラグをレジスタの値から減算	SBC <Rd>, <Rm>
	3ビットイミディエイト値をレジスタから減算	SUB <Rd>, <Rn>, #<immed_3>
	8ビットイミディエイト値をレジスタの値から減算	SUB <Rd>, #<immed_8>
	レジスタの値を減算	SUB <Rd>, <Rn>, <Rm>
4 × (7ビットイミディエイト値) をSPから減算	SUB SP, #<immed_7> * 4	
シフト・ローテート命令	イミディエイト値の数値により算術右シフト	ASR <Rd>, <Rm>, #<immed_5>
	レジスタの数値により算術右シフト	ASR <Rd>, <Rs>
	イミディエイト値により論理左シフト	LSL <Rd>, <Rm>, #<immed_5>
	レジスタの値により論理左シフト	LSL <Rd>, <Rs>
	イミディエイト値により論理右シフト	LSR <Rd>, <Rm>, #<immed_5>
	レジスタの値により論理右シフト	LSR <Rd>, <Rs>
低消費電力モード命令	イベント待ち	WFE <c>
	割り込み待ち	WFI <c>
比較命令	レジスタ値の2の補数 (負数、ネゲーション) を別のレジスタの値と比較	CMN <Rn>, <Rm>
	8ビットイミディエイト値と比較	CMP <Rn>, #<immed_8>
	レジスタと比較	CMP <Rn>, <Rm>
	上位レジスタを下位または上位レジスタと比較	CMP <Rn>, <Rm>
符号拡張命令	8ビットイミディエイト値の呼び出しコードにより、オペレーティングシステムのサービス呼び出し	SVC <immed_8>
	レジスタからバイト[7:0]を取り出し、レジスタに移動して、32ビットに符号拡張	SXTB <Rd>, <Rm>
	レジスタからハーフワード[15:0]を取り出し、レジスタに移動して、32ビットに符号拡張	SXTH <Rd>, <Rm>
	別のレジスタの値と論理積を実行して、レジスタの設定されているビットをテスト	TST <Rn>, <Rm>
	レジスタからバイト[7:0]を取り出し、レジスタに移動して、32ビットにゼロ拡張	UXTB <Rd>, <Rm>
	レジスタからハーフワード[15:0]を取り出し、レジスタに移動して、32ビットにゼロ拡張	UXTH <Rd>, <Rm>
分岐命令	条件付き分岐	B <cond> <target address>
	無条件分岐	B <target address>
	リンク付き分岐	BL <Rm>
	リンク付き分岐と状態遷移	BLX <Rm>
	比較して0でなければ分岐	CBNZ <Rn>, <label>
	比較して0で分岐	CBZ <Rn>, <label>
レジスタ転送命令	上位または下位レジスタの値を別の上位または下位レジスタにコピー	CPY <Rd> <Rm>
	8ビットイミディエイト値をレジスタに移動	MOV <Rd>, #<immed_8>
	下位レジスタの値を下位レジスタに移動	MOV <Rd>, <Rn>
	上位または下位レジスタの値を上位または下位レジスタに移動	MOV <Rd>, <Rm>
	ワード中のバイト順を反転して、レジスタへコピー	REV <Rd>, <Rn>
	2つのハーフワード中でそれぞれのバイト順を反転して、レジスタへコピー	REV16 <Rd>, <Rn>
	下位ハーフワード[15:0]でバイト順を反転、符号拡張して、レジスタへコピー	REVSH <Rd>, <Rn>

## 付録1 Thumb-2命令

### ●16ビット命令 (2/2)

機能	動作	ニーモニック
ロード・ストア命令	複数ワードを連続するメモリからロード	LDMIA <Rn>!, <registers>
	ベースレジスタのアドレス + 5ビットイミディエイト値オフセットのメモリからワードをロード	LDR <Rd>, [<Rn>, #<immed_5> * 4]
	ベースレジスタのアドレス + レジスタオフセットのメモリからワードをロード	LDR <Rd>, [<Rn>, <Rm>]
	PC アドレス + 8ビットイミディエイト値オフセットのメモリからワードをロード	LDR <Rd>, [PC, #<immed_8> * 4]
	SP アドレス + 8ビットイミディエイト値オフセットのメモリからワードをロード	LDR, <Rd>, [SP, #<immed_8> * 4]
	レジスタアドレス + 5ビットイミディエイト値オフセットのメモリバイト[7:0] ロード	LDRB <Rd>, [<Rn>, #<immed_5>]
	レジスタアドレス + レジスタオフセットのメモリからバイト[7:0] をロード	LDRB <Rd>, [<Rn>, <Rm>]
	レジスタアドレス + 5ビットイミディエイト値オフセットのメモリハーフワード[15:0]ロード	LDRH <Rd>, [<Rn>, #<immed_5> * 2]
	レジスタアドレス + レジスタオフセットのメモリからハーフワード[15:0] をロード	LDRH <Rd>, [<Rn>, <Rm>]
	レジスタアドレス + レジスタオフセットのメモリから符号付きバイト[7:0]をロード	LDRSB <Rd>, [<Rn>, <Rm>]
	レジスタアドレス + レジスタオフセットのメモリから符号付きハーフワード[15:0] をロード	LDRSH <Rd>, [<Rn>, <Rm>]
	複数のレジスタからワードを連続するメモリ位置へストア	STMIA <Rn>!, <registers>
	レジスタのアドレス + 5ビットイミディエイト値のオフセットへレジスタのワードをストア	STR <Rd>, [<Rn>, #<immed_5> * 4]
	レジスタのアドレスへレジスタのワードをストア	STR <Rd>, [<Rn>, <Rm>]
	SPアドレス + 8ビットイミディエイト値のオフセットへレジスタのワードをストア	STR <Rd>, [SP, #<immed_8> * 4]
	レジスタのアドレス + 5ビットイミディエイト値のオフセットへレジスタのバイト[7:0]をストア	STRB <Rd>, [<Rn>, #<immed_5>]
	レジスタのアドレスへレジスタのバイト[7:0] をストア	STRB <Rd>, [<Rn>, <Rm>]
	レジスタのアドレス + 5ビットイミディエイト値のオフセットへレジスタのハーフワード[15:0] をストア	STRH <Rd>, [<Rn>, #<immed_5> * 2]
レジスタのアドレス + レジスタによるオフセットへレジスタのハーフワード[15:0] をストア	STRH <Rd>, [<Rn>, <Rm>]	
論理演算命令	レジスタの値をビット単位に論理積	AND <Rd>, <Rm>
	レジスタの値をビット単位で排他的論理和	EOR <Rd>, <Rm>
	レジスタの値の否定 (1の補数、complement) をレジスタに移動	MVN <Rd>, <Rm>
	レジスタの値をビット単位で論理和	ORR <Rd>, <Rm>
IT命令		IT <cond>
	次の命令を条件付け、次の2つの命令を条件付け、次の3つの命令を条件付け、	IT<x> <cond>
	次の4つの命令を条件付け	IT<x><y> <cond>
		IT<x><y><z> <cond>
POP・PUSH	スタックから複数レジスタをポップ	POP <registers>
	スタックから複数レジスタおよびPCをポップ	POP <registers, PC>
	複数レジスタをスタックへプッシュ	PUSH <registers>
	LRおよび複数レジスタをスタックへプッシュ	PUSH <registers, LR>
その他	ビットクリア	BIC <Rd>, <Rm>
	ソフトウェアブレイクポイント	BKPT <immed_8>
	プロセッサ状態変更	CPS <effect>, <iflags>
	無操作	NOP <c>
	イベント送信	SEV <c>