1.4.2. Thumb instruction summary

The Thumb instruction set summary is shown in Table 1.12.

Table 1.12. Thumb instruction summary

Operation			Assembler
Move		Immediate	`MOV Rd, #8bit_Imm`
		High to Low	`MOV Rd, Hs`
		Low to High	`MOV Hd, Rs`
		High to High	`MOV Hd, Hs`
Arithmetic		Add	`ADD Rd, Rs, #3bit_Imm`
		Add Low and Low	`ADD Rd, Rs, Rn`
		Add High to Low	`ADD Rd, Hs`
		Add Low to High	`ADD Hd, Rs`
		Add High to High	`ADD Hd, Hs`
		Add Immediate	`ADD Rd, #8bit_Imm`
		Add Value to SP	`ADD SP, #7bit_Imm ADD SP, #-7bit_Imm`
		Add with carry	`ADC Rd, Rs`
		Subtract	`SUB Rd, Rs, Rn SUB Rd, Rs, #3bit_Imm`
		Subtract Immediate	`SUB Rd, #8bit_Imm`
		Subtract with carry	`SBC Rd, Rs`
		Negate	`NEG Rd, Rs`
		Multiply	`MUL Rd, Rs`
		Compare Low and Low	`CMP Rd, Rs`
		Compare Low and High	`CMP Rd, Hs`
		Compare High and Low	`CMP Hd, Rs`
		Compare High and High	`CMP Hd, Hs`
		Compare Negative	`CMN Rd, Rs`
		Compare Immediate	`CMP Rd, #8bit_Imm`
Logical		AND	`AND Rd, Rs`
		EOR	`EOR Rd, Rs`
		OR	`ORR Rd, Rs`

Operation			Assembler
		Bit clear	`BIC Rd, Rs`
		Move NOT	`MVN Rd, Rs`
		Test bits	`TST Rd, Rs`
Shift/Rotate		Logical shift left	`LSL Rd, Rs, #5bit_shift_imm LSL Rd, Rs`
		Logical shift right	`LSR Rd, Rs, #5bit_shift_imm LSR Rd, Rs`
		Arithmetic shift right	`ASR Rd, Rs, #5bit_shift_imm ASR Rd, Rs`
		Rotate right	`ROR Rd, Rs`
Branch	Conditional		
		If Z set	`BEQ label`
		If Z clear	`BNE label`
		If C set	`BCS label`
		If C clear	`BCC label`
		If N set	`BMI label`
		If N clear	`BPL label`
		If V set	`BVS label`
		If V clear	`BVC label`
		If C set and Z clear	`BHI label`
		If C clear and Z set	`BLS label`
		If N set and V set, or if N clear and V clear	`BGE label`
		If N set and V clear, or if N clear and V set	`BLT label`
		If Z clear and N or V set, or if Z clear, and N or V clear	`BGT label`
		If Z set, or N set and V clear, or N clear and V set	`BLE label`
		Unconditional	`B label`
		Long branch with link	`BL label`
		Optional state change	-
		To address held in Lo reg	`BX Rs`
		To address held in Hi reg	`BX Hs`
Load	With immediate offset		
		Word	`LDR Rd, [Rb, #7bit_offset]`

Operation			Assembler
		Halfword	`LDRH Rd, [Rb, #6bit_offset]`
		Byte	`LDRB Rd, [Rb, #5bit_offset]`
	With register offset		
		Word	`LDR Rd, [Rb, Ro]`
		Halfword	`LDRH Rd, [Rb, Ro]`
		Signed halfword	`LDRSH Rd, [Rb, Ro]`
		Byte	`LDRB Rd, [Rb, Ro]`
		Signed byte	`LDRSB Rd, [Rb, Ro]`
		PC-relative	`LDR Rd, [PC, #10bit_Offset]`
		SP-relative	`LDR Rd, [SP, #10bit_Offset]`
	Address		
		Using PC	`ADD Rd, PC, #10bit_Offset`
		Using SP	`ADD Rd, SP, #10bit_Offset`
		Multiple	`LDMIA Rb!, <reglist>`
Store	With immediate offset		
		Word	`STR Rd, [Rb, #7bit_offset]`
		Halfword	`STRH Rd, [Rb, #6bit_offset]`
		Byte	`STRB Rd, [Rb, #5bit_offset]`
	With register offset		
		Word	`STR Rd, [Rb, Ro]`
		Halfword	`STRH Rd, [Rb, Ro]`
		Byte	`STRB Rd, [Rb, Ro]`
	SP-relative		`STR Rd, [SP, #10bit_offset]`
	Multiple		`STMIA Rb!, <reglist>`
Push/Pop		Push registers onto stack	`PUSH <reglist>`
		Push LR and registers onto stack	`PUSH <reglist, LR>`
		Pop registers from stack	`POP <reglist>`
		Pop registers and PC from stack	`POP <reglist, PC>`
Software Interrupt			`SWI 8bit_Imm`

ARM DDI 0234B