

## Key to Tables

{cond}  
<Oprnd2>  
{field}  
S  
B  
H  
T  
<a\_mode1>  
<a\_mode2>  
<a\_mode3>  
<a\_mode4>  
<a\_mode5>  
<a\_mode6>  
#32\_Bit\_Immed

Refer to Table **Condition Field {cond}**  
Refer to Table **Oprnd2**  
Refer to Table **Field**  
Sets condition codes (optional)  
Byte operation (optional)  
Halfword operation (optional)  
Forces address translation. Cannot be used with pre-indexed addresses  
Refer to Table **Addressing Mode 1**  
Refer to Table **Addressing Mode 2**  
Refer to Table **Addressing Mode 3**  
Refer to Table **Addressing Mode 4**  
Refer to Table **Addressing Mode 5**  
Refer to Table **Addressing Mode 6**  
A 32-bit constant, formed by right-rotating an 8-bit value by an even number of bits

Operation	Assembler	S updates	Action	Notes
<b>Move</b>				
Move	MOV{cond}{S} Rd, <Oprnd2>	N Z C	Rd:= <Oprnd2>	
NOT	MVN{cond}{S} Rd, <Oprnd2>	N Z C	Rd:= 0xFFFFFFFF EOR <Oprnd2>	
SPSR to register	MRS{cond} Rd, SPSR		Rd:= SPSR	Architecture 3, 3M and 4 only
CPSR to register	MRS{cond} Rd, CPSR		Rd:= CPSR	Architecture 3, 3M and 4 only
register to SPSR	MSR{cond} SPSR{field}, Rm		SPSR:= Rm	Architecture 3, 3M and 4 only
register to CPSR	MSR{cond} CPSR{field}, Rm		SPSR:= Rm	Architecture 3, 3M and 4 only
immediate to SPSR flags	MSR{cond} SPSR_f, #32_Bit_Immed		SPSR:= #32_Bit_Immed	Architecture 3, 3M and 4 only
immediate to CPSR flags	MSR{cond} CPSR_f, #32_Bit_Immed		CPSR:= #32_Bit_Immed	Architecture 3, 3M and 4 only
<b>ALU</b>				
Arithmetic				
Add	ADD{cond}{S} Rd, Rn, <Oprnd2>	N Z C V	Rd:= Rn + <Oprnd2>	
with carry	ADC{cond}{S} Rd, Rn, <Oprnd2>	N Z C V	Rd:= Rn + <Oprnd2> + Carry	
Subtract	SUB{cond}{S} Rd, Rn, <Oprnd2>	N Z C V	Rd:= Rn - <Oprnd2>	
with carry	SBC{cond}{S} Rd, Rn, <Oprnd2>	N Z C V	Rd:= Rn - <Oprnd2> - NOT(Carry)	
reverse subtract	RSB{cond}{S} Rd, Rn, <Oprnd2>	N Z C V	Rd:= <Oprnd2> - Rn	
reverse subtract with carry	RSC{cond}{S} Rd, Rn, <Oprnd2>	N Z C V	Rd:= <Oprnd2> - Rn - NOT(Carry)	
Negate	MUL{cond}{S} Rd, Rm, Rs	N Z	Rd:= Rm * Rs	Not in Architecture 1
Multiply	MLA{cond}{S} Rd, Rm, Rs, Rn	N Z	Rd:= (Rm * Rs) + Rn	Not in Architecture 1
accumulate	UMULL{cond}{S} RdHi, RdLo, Rm, Rs	N Z	RdHi:= (Rm * Rs)[63:32] RdLo:= (Rm * Rs)[31:0]	Architecture 3M and 4 only
unsigned long	UMLAL{cond}{S} RdHi, RdLo, Rm, Rs	N Z	RdHi:= (Rm * Rs) + RdHi+ RdLo:= (Rm * Rs) + RdLo+ CarryFrom((Rm * Rs)[31:0] + RdLo)	Architecture 3M and 4 only
signed long	SMULL{cond}{S} RdHi, RdLo, Rm, Rs	N Z	RdHi:= signed(Rm * Rs)[63:32] RdLo:= signed(Rm * Rs)[31:0]	Architecture 3M and 4 only
signed accumulate long	SMLAL{cond}{S} RdHi, RdLo, Rm, Rs	N Z	RdHi:= signed(Rm * Rs) + RdHi+ CarryFrom((Rm * Rs)[31:0] + RdLo)	Architecture 3M and 4 only
Compare	CMP{cond} Rd, <Oprnd2>	N Z C V	CPSR flags:= Rn - <Oprnd2>	
negative	CMN{cond} Rd, <Oprnd2>	N Z C V	CPSR flags:= Rn + <Oprnd2>	
Logical				
Test	TST{cond} Rn, <Oprnd2>	N Z C	CPSR flags:= Rn AND <Oprnd2>	
Test equivalence	TEQ{cond} Rn, <Oprnd2>	N Z C	CPSR flags:= Rn EOR <Oprnd2>	Does not update the V flag
AND	AND{cond}{S} Rd, Rn, <Oprnd2>	N Z C	Rd:= Rn AND <Oprnd2>	
EOR	EOR{cond}{S} Rd, Rn, <Oprnd2>	N Z C	Rd:= Rn EOR <Oprnd2>	
ORR	ORR{cond}{S} Rd, Rn, <Oprnd2>	N Z C	Rd:= Rn OR <Oprnd2>	
Bit Clear	BIC{cond}{S} Rd, Rn, <Oprnd2>	N Z C	Rd:= Rn AND NOT <Oprnd2>	
Shift/Rotate				See Table <b>Oprnd2</b>

Operation	Assembler	Action	Notes
<b>Branch</b>	Branch with link and exchange instruction set B{cond} label BL{cond} label BX{cond} Rn	R15:= address R14:=R15, R15:= address R15:=Rn, T bit:= Rn[0]	Architecture 4 with Thumb only Thumb state: Rn[0] = 0 ARM state: Rn[0] = 1
<b>Load</b>	Word with user-mode privilege Byte with user-mode privilege signed Halfword signed Multiple Block data operations Increment Before Increment After Decrement Before Decrement After Stack operations and restore CPSR User registers	Rd:= [address] Rd:= [byte value from address] Loads bits 0 to 7 and sets bits 8-31 to 0 Rd:= [signed byte value from address] Loads bits 0 to 7 and sets bits 8-31 to bit 7 Rd:= [halfword value from address] Loads bits 0 to 15 and sets bits 16-31 to 0 Rd:= [signed halfword value from address] Loads bits 0 to 15 and sets bits 16-31 to bit 15 Stack manipulation (pop)	Architecture 4 only Architecture 4 only Architecture 4 only ! sets the W bit (updates the base register after the transfer) ^ sets the S bit ! sets the W bit (updates the base register after the transfer)
<b>Store</b>	Word with user-mode privilege Byte with user-mode privilege Halfword Multiple Block data operations Increment Before Increment After Decrement Before Decrement After Stack operations User registers	[address]:= Rd [address]:= byte value from Rd [address]:= halfword value from Rd Stack manipulation (push)	Architecture 4 only ! sets the W bit (updates the base register after the transfer) ^ sets the S bit
<b>Swap</b>	Word Byte	SWP{cond} Rd, Rm, [Rn] SWP{cond}B Rd, Rm, [Rn]	Not in Architecture 1 or 2 Not in Architecture 1 or 2
<b>Coprocessors</b>	Data operations Move to ARM reg from coproc Move to coproc from ARM reg Load Store	CDP{cond} p<cpnum>, <op1>, CRd, CRn, CRm, <op2> MRC{cond} p<cpnum>, <op1>, Rd, CRn, CRm, <op2> MCR{cond} p<cpnum>, <op1>, Rd, CRn, CRm, <op2> LDC{cond} p<cpnum>, CRd, <a_mode6> STC{cond} p<cpnum>, CRd, <a_mode6>	Not in Architecture 1
<b>Software Interrupt</b>	SWI #24_Bit_Value		24-bit immediate value

# ARM Addressing Modes Quick Reference Card

Addressing Mode 1	
Immediate offset	[Rn, #+/-12_Bit_Offset]
Register offset	[Rn, +/-Rm]
Scaled register offset	[Rn, +/-Rm, LSL #shift_imm] [Rn, +/-Rm, LSR #shift_imm] [Rn, +/-Rm, ASR #shift_imm] [Rn, +/-Rm, ROR #shift_imm]
Pre-indexed offset	[Rn, +/-Rm, ROR #shift_imm]
Immediate	[Rn, #+/-12_Bit_Offset]
Register	[Rn, +/-Rm]!
Scaled register	[Rn, +/-Rm, LSL #shift_imm]! [Rn, +/-Rm, LSR #shift_imm]! [Rn, +/-Rm, ASR #shift_imm]! [Rn, +/-Rm, ROR #shift_imm]!
Post-indexed offset	[Rn, #+/-12_Bit_Offset]
Immediate	[Rn], +/-Rm
Register	[Rn], +/-Rm, LSL #shift_imm
Scaled register	[Rn], +/-Rm, LSR #shift_imm [Rn], +/-Rm, ASR #shift_imm [Rn], +/-Rm, ROR #shift_imm

Addressing Mode 2	
Immediate offset	[Rn, #+/-12_Bit_Offset]
Register offset	[Rn, +/-Rm]
Scaled register offset	[Rn, +/-Rm, LSL #shift_imm] [Rn, +/-Rm, LSR #shift_imm] [Rn, +/-Rm, ASR #shift_imm] [Rn, +/-Rm, ROR #shift_imm]
Post-indexed offset	[Rn], #+/-12_Bit_Offset
Immediate	[Rn], +/-Rm
Register	[Rn], +/-Rm, LSL #shift_imm
Scaled register	[Rn], +/-Rm, LSR #shift_imm [Rn], +/-Rm, ASR #shift_imm [Rn], +/-Rm, ROR #shift_imm

Addressing Mode 3 - Signed Byte and Halfword Data Transfer	
Immediate offset	[Rn, #+/-8_Bit_Offset]
Pre-indexed	[Rn, #+/-8_Bit_Offset]!
Post-indexed	[Rn], #+/-8_Bit_Offset
Register	[Rn, +/-Rm]
Pre-indexed	[Rn, +/-Rm]!
Post-indexed	[Rn], +/-Rm

Addressing Mode 6 - Coprocessor Data Transfer	
Immediate offset	[Rn, #+/- (8_Bit_Offset*4)]
Pre-indexed	[Rn, #+/- (8_Bit_Offset*4)]!
Post-indexed	[Rn], #+/- (8_Bit_Offset*4)

Oprrnd2	
Immediate value	#32_Bit_Immed
Logical shift left	Rm LSL #5_Bit_Immed
Logical shift right	Rm LSR #5_Bit_Immed
Arithmetic shift right	Rm ASR #5_Bit_Immed
Rotate right	Rm ROR #5_Bit_Immed
Register	Rm
Logical shift left	Rm LSL Rs
Logical shift right	Rm LSR Rs
Arithmetic shift right	Rm ASR Rs
Rotate right	Rm ROR Rs
Rotate right extended	Rm RRX

Field	Sets
_c	Control field mask bit (bit 3)
_f	Flags field mask bit (bit 0)
_s	Status field mask bit (bit 1)
_x	Extension field mask bit (bit 2)

Condition Field {cond}	
Suffix	Description
EQ	Equal
NE	Not equal
CS	Unsigned higher or same
CC	Unsigned lower
MI	Negative
PL	Positive or zero
VS	Overflow
VC	No overflow
HI	Unsigned higher
LS	Unsigned lower or same
GE	Greater or equal
LT	Less than
GT	Greater than
LE	Less than or equal
AL	Always

Addressing Mode 4	
Addressing Mode	Stack Type
IA	Increment After
IB	Increment Before
DA	Decrement After
DB	Decrement Before

Addressing Mode 5	
Addressing Mode	Stack Type
IA	Increment After
IB	Increment Before
DA	Decrement After
DB	Decrement Before