Opcode	31 30 29 28	27 26 25 24	4 23 22 21	20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
AND <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 0			Rn	Rd	shift#	shift 0	Rm
AND <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 0		_	Rn	Rd	Rs	0 shift 1	Rm
MUL <cond><s> Rd, Rm, Rs STR<cond>H Rd, <address></address></cond></s></cond>	cond	0 0 0 0 0 0 0 P			Rd Rn	SBZ Rd	Rs addr mode	1 0 0 1	Rm addr mode
LDR <cond>H Rd, <address></address></cond>	cond	0 0 0 P		_	Rn	Rd	addr_mode	1 0 1 1	addr_mode
Undefined Instruction	cond	0 0 0 x		_	x x x x	x x x x	x	1 1 0 1	x x x x
LDR <cond>SB Rd, <address></address></cond>	cond	0 0 0 P		_	Rn	Rd	addr_mode	1 1 0 1	addr_mode
Undefined Instruction LDR <cond>SH Rd, <address></address></cond>	cond	0 0 0 x			X X X X	x x x x	x x x x and addr_mode	1 1 1 1	x x x x a
EOR <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 0		S	Rn	Rd	shift #	shift 0	Rm
EOR <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 0	0 0 1	S	Rn	Rd	Rs	0 shift 1	Rm
MLA <cond><s> Rd, Rm, Rs, Rn</s></cond>	cond	0 0 0 0		S	Rd	Rn	Rs	1 0 0 1	Rm
SUB <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 0		_	Rn	Rd	shift #	shift 0	Rm
SUB <cond><s> Rd, Rn, Rm OP Rs RSB<cond><s> Rd, Rn, Rm OP #</s></cond></s></cond>	cond	0 0 0 0		S	Rn Rn	Rd Rd	Rs shift #	0 shift 1 shift 0	Rm Rm
RSB <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 0		S	Rn	Rd	Rs	0 shift 1	Rm
ADD <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 0	1 0 0	S	Rn	Rd	shift #	shift 0	Rm
ADD <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 0		_	Rn	Rd	Rs	0 shift 1	Rm
UMULL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	cond	0 0 0 0		_	RdHi	RdLo	Rs	1 0 0 1	Rm
ADC <cond><s> Rd, Rn, Rm OP # ADC<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0 0		S	Rn Rn	Rd Rd	shift #	shift 0 0 shift 1	Rm Rm
UMLAL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	cond	0 0 0 0		S	RdHi	RdLo	Rs	1 0 0 1	Rm
SBC <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 0			Rn	Rd	shift #	shift 0	Rm
SBC <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 0		_	Rn	Rd	Rs	0 shift 1	Rm
SMULL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	cond	0 0 0 0		_	RdHi	RdLo	Rs	1 0 0 1	Rm
RSC <cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0 0		S	Rn Rn	Rd Rd	shift #	shift 0 shift 1	Rm Rm
SMLAL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	cond	0 0 0 0		S	RdHi	RdLo	Rs	1 0 0 1	Rm
TST <cond> Rn, Rm OP #</cond>	cond	0 0 0 1	0 0 0		Rn	SBZ	shift #	shift 0	Rm
TST <cond> Rn, Rm OP Rs</cond>	cond	0 0 0 1	0 0 0	_	Rn	SBZ	Rs	0 shift 1	Rm
MRS <cond> Rd, CPSR SWP<cond> Rd, Rm, [Rn]</cond></cond>	cond	0 0 0 1		0 BZ	SBO Rn	Rd Rd	SBZ	SBZ	Rm
TEQ <cond> Rn, Rm OP #</cond>	cond	0 0 0 1		1	Rn	SBZ	shift #	shift 0	Rm
TEQ <cond> Rn, Rm OP Rs</cond>	cond	0 0 0 1	0 0 1	1	Rn	SBZ	Rs	0 shift 1	Rm
MSR <cond> CPSR_<fields>, Rm</fields></cond>	cond	0 0 0 1		0	field_mask	SBO	SBZ	0	Rm
BX <cond> Rm</cond>	cond	0 0 0 1	0 0 1	0	SBO	SBO SB7	SBO	0 0 0 1	Rm
CMP <cond> Rn, Rm OP # CMP<cond> Rn, Rm OP Rs</cond></cond>	cond	0 0 0 1	0 1 0	_	Rn Rn	SBZ SBZ	shift #	shift 0 0 shift 1	Rm Rm
MRS <cond> Rd, SPSR</cond>	cond	0 0 0 1		_	SBO	Rd	. 10	SBZ	· Mil
SWP <cond>B Rd, Rm, [Rn]</cond>	cond	0 0 0 1		ΒZ	Rn	Rd	SBZ	1 0 0 1	Rm
CMN <cond> Rn, Rm OP #</cond>	cond	0 0 0 1	0 1 1	_	Rn	SBZ	shift #	shift 0	Rm
CMN <cond> Rn, Rm OP Rs MSR<cond> SPSR <fields>, Rm</fields></cond></cond>	cond	0 0 0 1	0 1 1		Rn field meek	SBZ SBO	Rs	0 shift 1	Rm Rm
MSR <cond> SPSR_<fields>, Rm ORR<cond><s> Rd, Rn, Rm OP #</s></cond></fields></cond>	cond	0 0 0 1	1 0 0	0 S	field_mask Rn	Rd	SBZ shift #	shift 0	Rm
ORR <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 1	1 0 0	_	Rn	Rd	Rs	0 shift 1	Rm
MOV <cond><s> Rd, Rm OP #</s></cond>	cond	0 0 0 1	1 0 1	S	SBZ	Rd	shift#	shift 0	Rm
MOV <cond><s> Rd, Rm OP Rs</s></cond>	cond	0 0 0 1	1 0 1	S	SBZ	Rd	Rs	0 shift 1	Rm
BIC <cond><s> Rd, Rn, Rm OP # BIC<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0 1	1 1 0		Rn	Rd	shift #	shift 0	Rm
BIC <cond><s> Rd, Rn, Rm OP Rs MVN<cond><s> Rd, Rm OP #</s></cond></s></cond>	cond	0 0 0 1	1 1 0	S	Rn SBZ	Rd Rd	Rs shift #	0 shift 1 shift 0	Rm Rm
MVN <cond><s> Rd, Rm OP Rs</s></cond>	cond	0 0 0 1	1 1 1		SBZ	Rd	Rs	0 shift 1	Rm
AND <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0	0 0 0	S	Rn	Rd	rotate	#	#
EOR <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0		S	Rn	Rd	rotate	#	
SUB <cond><s> Rd, Rn, # RSB<cond><s> Rd, Rn, #</s></cond></s></cond>	cond	0 0 1 0		S	Rn Rn	Rd Rd	rotate rotate	#	
ADD <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0		_	Rn	Rd	rotate	#	
ADC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0			Rn	Rd	rotate	#	
SBC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0	1 1 0	S	Rn	Rd	rotate	#	<i>‡</i>
RSC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0		S	Rn	Rd	rotate	#	
TST <cond> Rn, # TEQ<cond> Rn, #</cond></cond>	cond	0 0 1 1		_	Rn Rn	SBZ SBZ	rotate rotate	#	
MSR <cond> CPSR f, #</cond>	cond	0 0 1 1		_		SBO	rotate	#	
CMP <cond> Rn, #</cond>	cond	0 0 1 1		$\overline{}$	Rn	SBZ	rotate	#	
CMN <cond> Rn, #</cond>	cond	0 0 1 1		1	Rn	SBZ	rotate	#	‡
MSR <cond> SPSR_f, #</cond>	cond	0 0 1 1		-	field_mask	SBO	rotate	#	
ORR <cond><s> Rd, Rn, # MOV<cond><s> Rd, #</s></cond></s></cond>	cond	0 0 1 1			Rn SBZ	Rd Rd	rotate rotate	#	
MOV <cond><s> Rd, # BIC<cond><s> Rd, Rn, #</s></cond></s></cond>	cond	0 0 1 1		_	Rn SBZ	Rd	rotate	#	
MVN <cond><s> Rd, #</s></cond>	cond	0 0 1 1	1 1 1		SBZ	Rd	rotate	#	
STR <cond> Rd, Rn, #</cond>	cond	0 1 0 P			Rn	Rd		#	
LDR <cond> Rd, Rn, #</cond>	cond	0 1 0 P			Rn	Rd		#	
STR <cond>B Rd, Rn, # LDR<cond>B Rd, Rn, #</cond></cond>	cond	0 1 0 P			Rn Rn	Rd Rd		#	
		0 1 0 0		_				#	
STR <cond>T Rd, Rn, #</cond>	cond			0	Rn	Rd			
LDR <cond>T Rd, Rn, #</cond>	cond	0 1 0 0	U 0 1	1	Rn	Rd		#	
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, #</cond></cond>	cond cond	0 1 0 0	U 0 1	0	Rn Rn	Rd Rd		#	
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, #</cond></cond></cond>	cond cond	0 1 0 0 0 1 0 0 0 1 0 0	U 0 1 U 1 1 U 1 1	1 0 1	Rn Rn Rn	Rd Rd Rd	shift #	#	Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, #</cond></cond>	cond cond	0 1 0 0	U 0 1 U 1 1 U 1 1 U 0 W	1 0 1	Rn Rn	Rd Rd	shift #	#	Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, #</cond></cond></cond></cond></cond></cond>	cond cond cond	0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P	U 0 1 U 1 1 U 1 1 U 0 W U 0 W	1 0 1 1 0 0	Rn Rn Rn Rn Rn Rn	Rd Rd Rd Rd Rd Rd	shift #	# shift 0 shift 0 shift 0	Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, #</cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P	U 0 1 U 1 1 U 1 1 U 0 W U 0 W U 0 W U 1 W	1 0 1 1 0 1 1	Rn Rn Rn Rn Rn Rn Rn	Rd Rd Rd Rd Rd Rd Rd	shift # shift # shift #	# shift 0 shift 0 shift 0 shift 0	Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, #</cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0	U 0 1 U 1 1 U 1 1 U 0 W U 0 W U 0 W U 1 W U 1 W	1 0 1 0 1 0 1 0	Rn Rn Rn Rn Rn Rn Rn Rn	Rd	shift # shift # shift # shift #	# # shift 0 shift 0 shift 0 shift 0 shift 0 shift 0	Rm Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, #</cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0	U 0 1 U 1 1 U 1 1 U 0 W U 0 W U 0 W U 1 W U 0 1	1 0 1 0 1 0 1 0	Rn Rn Rn Rn Rn Rn Rn	Rd Rd Rd Rd Rd Rd Rd	shift # shift # shift #	# shift 0 shift 0 shift 0 shift 0	Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, #</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0 0 1 1 0 0 1 1 0	U 0 1 U 1 1 U 1 1 U 0 W U 0 W U 0 W U 1 W U 0 1	1 0 1 0 1 0 1 0 1	Rn Rn Rn Rn Rn Rn Rn Rn Rn	Rd R	shift # shift # shift # shift # shift #	# # shift 0	Rm Rm Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond>B Rd, Rn, # STR<cond>B Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # UDR<cond>BT Rd, Rn, # UDR<cond>BT Rd, Rn, # UDR<cond>BT Rd, Rn, # UDR<cond>BT Rd, Rn, # Undefined Instruction</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 1 1 0 0 1 1 0	U 0 1 U 1 1 U 1 1 U 0 W U 0 W U 0 W U 0 1 U 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 1 1 0 1 1 0 1 0 1 x	Rn R	Rd R	shift # x x x x	# #	Rm Rm Rm Rm Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> T Rd, R</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0 0 1 1 0	U 0 1 U 1 1 U 1 1 U 0 W U 0 W U 1 W U 0 1 U 1 1 U 0 1 U 1 1 U 0 1 U 1 1 U 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 1 1 0 1 1 0 0 1 1 x	Rn	Rd R	shift # x x x x regist	# # shift 0 x x x x 1 er list	Rm Rm Rm Rm Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond>B Rd, Rn, # STR<cond>B Rd, Rn, # LDR<cond>B Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>Cond>BT Rd, Rn, # LDR<cond>Cond>Cond>Cond STM<cond>Cond>Cond>Cond STM<cond>Cond>Cond STM<cond>Cond>Cond STM<cond>Cond>Cond STM<cond>Cond>Cond STM<cond>Cond>Cond STM<cond>Cond>Cond STM<cond>Cond>Cond STM<cond>Cond STM<cond>Cond>Cond STM<cond>Cond STM<cond>Cond STM<cond>Cond STM<cond>Cond STM<cond>Cond STM<cond>Cond STM<cond stm<cond="">Cond STM<cond stm<cond="">Cond STM<cond stm<cond="">Cond STM<cond stm<cond="" stm<cond<="" td=""><td>cond cond cond cond cond cond cond cond</td><td>0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0 0 1 1 0 0</td><td>U 0 1 U 1 1 U 1 1 U 0 W U 0 W U 1 W U 0 W U 1 W U</td><td>1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1</td><td>Rn</td><td>Rd Rd R</td><td>shift # shift # x x x x regist</td><td># # shift 0 shift 0</td><td>Rm Rm Rm Rm Rm Rm Rm</td></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0 0 1 1 0 0	U 0 1 U 1 1 U 1 1 U 0 W U 0 W U 1 W U 0 W U 1 W U	1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1	Rn	Rd R	shift # x x x x regist	# # shift 0	Rm Rm Rm Rm Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> T Rd, R</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0	U 0 1 U 1 1 U 1 1 U 0 W U 0 W U 1 W U 0 M U 1 W U 1 W U 1 W U 0 W U 1 W U	1 0 1 1 0 1 1 0 1 1 x	Rn	Rd R	shift # x x x x regist regist	# # shift 0 shift 10 shi	Rm Rm Rm Rm Rm Rm Rm x x x x
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, Rn, Rn, Rn, Rn, Rn, Rn, Rn, Rn, Rn</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0 0 1 1 0 0 0 0 0 0	U 0 1 U 1 1 U 1 1 U 0 W U 0 W U 0 1 U 0 0 U 0 0 U 0 0 U 0 1 U 0 0 U	1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 0 0 1 0 0	Rn	Rd R	shift # x x x x regist regist x x x x	# # shift 0 shift 10 shi	Rm Rm Rm Rm Rm Rm Rm X X X
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> B Rd, Rn, # STR<cond> B Rd, Rn, # LDR<cond> B Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<str<str<str<str<str<str<str<str<str<< td=""><td>cond cond cond cond cond cond cond cond</td><td>0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 1 P 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0</td><td>U 0 1 U 1 1 U 1 0 W U 0 W U 1 W 1 W U 1 W 1 W U 1 W 1 W 1 W U 1 W 1 W 1 W 1 W 1 W 1 W 1 W 1 W 1 W 1</td><td>1 0 1 1 0 1 1 0 1 0 1 1 0 0 1 1 0 0 1 1</td><td>Rn Rn R</td><td>Rd Rd R</td><td>shift # shift # shift # shift # shift # shift # shift # x x x x regist regist x x x x regist x x x x</td><td># # # # # # # # # # # # # # # # # # #</td><td>Rm Rm Rm Rm Rm Rm Rm</td></str<str<str<str<str<str<str<str<str<<></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 1 P 0 1 1 1 0 0 1 1 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0	U 0 1 U 1 1 U 1 0 W U 0 W U 1 W 1 W U 1 W 1 W U 1 W 1 W 1 W U 1 W 1 W 1 W 1 W 1 W 1 W 1 W 1 W 1 W 1	1 0 1 1 0 1 1 0 1 0 1 1 0 0 1 1 0 0 1 1	Rn R	Rd R	shift # x x x x regist regist x x x x regist x x x x	# # # # # # # # # # # # # # # # # # #	Rm Rm Rm Rm Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rm< I'>, reg list LDM<cond> caddrmode> Rm< I'>, reg list UNGREDICTABLE LDM<cond> caddrmode> Rm, reg list UNPREDICTABLE STM<cond> caddrmode> Rm, reg list UNPREDICTABLE STM<cond> caddrmode> Rm< I'>, reg list UNPREDICTABLE STM<cond> caddrmode> Rm< I'>, reg list OUNPREDICTABLE STM<cond> caddrmode> Rm< I'>, reg list</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 0 0 0 1 1 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0	U 0 1 1 U 1 1 1 U 1 1 U 1 U 1 U 1 U 1 U	1 0 1 0 1 0 1 0 1 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 0 1	Rn R	Rd R	shift # x x x x regist regist x x x x regist x x x x regist	# # # # # # # # # # # # # # # # # # #	Rm Rm Rm Rm Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rm<!-- -->, reg list LDM<cond> Rddrmode> Rm<!-- -->, reg list UNPREDICTABLE LDM<cond> Rm<!-- -->, reg list STM<cond> Rm<!-- -->, reg list LDM<cond> Rm<!-- -->, reg list</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 1 P 0 1 1 1 P 0 1 1 0 0 0 1 1 0 0 P 1 0 0 0 P	U 0 1 1 U 1 1 U 1 U 1 U 1 U 1 U 1 U 1 U	1 0 1 0 1 0 1 0 1 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 0	Rn R	Rd R	shift # x x x x regist regist x x x x regis x x x x regist regist regist regist	# # # # # # # # # # # # # # # # # # #	Rm Rm Rm Rm Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>Cond>BT Rd, Rn, # LDR<cond>Cond>Cond STM<cond>Cond>Cond STM<cond>Cond STM<cond>Cond STM<cond>Cond STM STM STM STM STM STM STM STM STM STM</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 1 P 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0	U 0 1 U 1 1 U 0 W U 0 W U 1 W U 0 W U 1 W U 1 W U 1 W U 1 W U 1 W U 0 W U	1 0 1 0 1 0 1 0 1 0 1 0 0 1 1 0 0 1 1 0 0 1 0 0 1 0	Rn R	Rd R	shift # shift	# # # # # # # # # # # # # # # # # # #	Rm Rm Rm Rm Rm Rm Rm
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rm<!-- -->, reg list LDM<cond> Rddrmode> Rm<!-- -->, reg list UNPREDICTABLE LDM<cond> Rm<!-- -->, reg list STM<cond> Rm<!-- -->, reg list LDM<cond> Rm<!-- -->, reg list</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 1 P 0 1 1 1 P 0 1 1 0 0 0 1 1 0 0 P 1 0 0 0 P	U 0 1 U 1 1 U 0 W U 0 W U 0 U 1 U 0 1 U 1 1 1 U 0 1 U 1 1 1 U 0 1 1 U 1 1 1 U 1 1 0 1 U 1 0 0 0 U 1 0 0 0 U 1 0 0 0 U 1 0 0 0 U 0 0 0 0 U	1 0 1 0 1 0 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1	Rn R	Rd R	shift # x x x x regist regist x x x x regis x x x x regist regist regist regist	# # # # # # # # # # # # # # # # # # #	Rm Rm Rm Rm Rm Rm Rm X X X X
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rm<!-- -->, reg list</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0	U 0 1 1	1 0 1 0 1 0 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 0 0 1 1 1 0 0 0 1 1	Rn R	Rd R	shift # shift	# # # # # # # # # # # # # # # # # # #	Rm Rm Rm Rm Rm Rm Rm X X X X 2
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>STR Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rm<!-- -->, reg list LDM<cond> caddrmode> Rm<!-- -->, reg list LDM<cond> caddrmode> Rm, reg list UNPREDICTABLE LDM<cond> caddrmode> Rm, reg list LDM<cond> caddrmode> Rm<!-- -->, reg list LDM<cond> caddrmode> Rm<!-- -->, reg list STM<cond> caddrmode> Rm<!-- -->, reg list LDM<cond> caddrmode> Rm<!-- -->, reg list STM<cond> caddrmode> Rm<!-- -->, reg list LDM<cond> caddrmode> Rm<!-- , reg list LDM<cond--> caddrmode> c</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 1 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 1 P 0 1 1 1 P 0 1 1 0 1 1 P 0 1 1 0 0 P 1 1 1 0 0 P	U 0 1 1 U 1 1 U 0 W U 0 W U 0 W U 1	1 0 1 1 0 1 0 1 0 1 1 0 0 1 1 1 0 0 1	Rn R	Rd R	shift # x x x x regist regist regist x x x x regict coffset coffset coffset cof_num cof_num cof_num	# # # # # # # # # # # # # # # # # # #	Rm Rm Rm Rm Rm Rm Rm X X X X X
LDR <cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # LDR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # LDR<cond> Rm<!-- -->, reg list</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 0 0 0 1 0 0 0 1 1 P 0 1 1 P 0 1 1 P 0 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 1 0	U 0 1 1 U 0 W U 0 W 0 U 0 W 0 U 1 W 0 W 0 U 0 W 0 W 0 U 0 W 0 U 0 W 0 U 0 W 0 U 0 W 0 U 0 W 0 U 0 W 0 U 0 W 0 U 0 W 0 U 0 W 0 U 0 W 0 U 0 W 0 W	1 0 1 0 1 0 1 0 0 1 1 0 0 0 1 1 0 0 1 1 0 0 0 1 1 1 0 0 0 1 1	Rn R	Rd R	shift # shift	# # # # # # # # # # # # # # # # # # #	Rm Rm Rm Rm Rm Rm X X X X X

Meaning	Mnemonic	Opcode	Status Flags
Equal	EQ	0 0 0 0	Z = 1
Not Equal	NE	0 0 0 1	Z = 0
Carry Set	CS	0 0 1 0	C = 1
Carry Clear	CC	0 0 1 1	C = 0
Unsigned Higher or Same	HS	0 0 1 0	C = 1
Unsigned Lower	LO	0 0 1 1	C = 0
Minus/Negative	MI	0 1 0 0	N = 1
Plus/Positive or Zero	PL	0 1 0 1	N = 0
Overflow	VS	0 1 1 0	V = 1
No Overflow	VC	0 1 1 1	V = 0
Unsigned Higher	HI	1 0 0 0	C = 1, Z = 0
Unsigned Lower or Same	LS	1 0 0 1	C = 0, Z = 1
Signed Greater than or Equal	GE	1 0 1 0	N = V
Signed Less than	LT	1 0 1 1	N != V
Signed Greater than	GT	1 1 0 0	Z = 0, N = V
Signed Less than or Equal	LE	1 1 0 1	Z = 1, N != V
Always	AL	1 1 1 0	-
Never	NE	1 1 1 1	-



ARM Reference	5				
Opcode	Notes	Z	С	N	٧
ADC <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-	s	s	s	s
ADD <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-	s	s	s	s
AND <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-	s	s	s	
B <cond> <target_addr></target_addr></cond>	-				
BIC <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-	s	s	s	
BL <cond> <target_addr></target_addr></cond>	-				
BX <cond> Rm</cond>	-				
CDP <cond> p<cp#>,o1,CRd,CRn,CRm,o2</cp#></cond>	-				
CMN <cond> Rn, <sh_op></sh_op></cond>	-	х	x	x	х
CMP <cond> Rn, <sh_op></sh_op></cond>	-	х	x	x	x
EOR <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-	ø	s	s	
LDC <cond> p<cp_num>, CRd, #</cp_num></cond>	-				
LDM <cond><adm> Rm, {reg list}^</adm></cond>	-				
LDM <cond><adm> Rm<!-- -->, {reg list}</adm></cond>	"!" Keeps last Rm Value				
LDM <cond><adm> Rm<!-- -->, {reg list}^</adm></cond>					
LDR <cond> Rd, Rn, #</cond>	word				
LDR <cond>B Rd, Rn, #</cond>	byte				
LDR <cond>BT Rd, Rn, #</cond>	translate byte				
LDR <cond>H Rd, <address></address></cond>	half word	Ì			
LDR <cond>SB Rd, <address></address></cond>	signed byte				
LDR <cond>SH Rd, <address></address></cond>	signed half word				
LDR <cond>T Rd, Rn, #</cond>	translate word				
MCR <cond> p<cp#>,o1,Rd,CRn,CRm,o2</cp#></cond>	-				
MLA <cond><s> Rd, Rm, Rs, Rn</s></cond>	-	s	s	s	
MOV <cond><s> Rd, <sh_op></sh_op></s></cond>	-	s	s	s	
	if Rd = r15 then flags				
MRC <cond> p<cp#>,o1,Rd,CRn,CRm,o2</cp#></cond>	affected	*	*	*	*
MRS <cond> Rd, CPSR</cond>	_				
MRS <cond> Rd, SPSR</cond>	_				
MSR <cond> CPSR <fields>, Rm</fields></cond>	_				
MSR <cond> CPSR f, #</cond>	_				
MSR <cond> SPSR_<fields>, Rm</fields></cond>	_				
MSR <cond> SPSR f, #</cond>	_				
MUL <cond><s> Rd, Rm, Rs</s></cond>	_	s	s	s	
MVN <cond><s> Rd, <sh_op></sh_op></s></cond>	_		s		
ORR <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-	s	s	s	
RSB <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-	s	s	s	s
RSC <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	_	s	s	s	s
SBC <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-	_		s	s
SMLAL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	64bit target			s	
SMULL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	64bit target			s	
STC <cond> p<cp_num>, CRd, #</cp_num></cond>	-				
STM <cond><adm> Rm, {reg list}^</adm></cond>	_				
STM <cond><adm> Rm<!-- -->, {reg list}</adm></cond>	"!" Keeps last Rm Value				
STM <cond><adm> Rm<!-- -->, {reg list}^</adm></cond>	-				
STR <cond> Rd, Rn, #</cond>	word	Ì			
STR <cond>B Rd, Rn, #</cond>	byte	Ì			
STR <cond>BT Rd, Rn, #</cond>	translate byte	Ì			
STR <cond>H Rd, <address></address></cond>	half word	Ì			
STR <cond>T Rd, Rn, #</cond>	translate word	Ì			
SUB <cond><s> Rd, Rn, <sh op=""></sh></s></cond>	-	s	s	s	s
SWI <swi_number></swi_number>	<u>.</u>	ľ	-	-	٥
SWP <cond> Rd, Rm, [Rn]</cond>	_	ŀ			
SWP <cond>Rd, Rm, [Rn]</cond>					
TEQ <cond> Rn, <sh op=""></sh></cond>		Ļ	х	Ţ	
TST <cond> Rn, <sh op=""></sh></cond>	1		x	X	
UMLAL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	64bit target	s	s	s	s
		_			_
UMULL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	64bit target	s	s	s	s

Data Processing Opcode
Load/Store Opcode
Branching Opcode
Multiplication Opcode
Other Opcodes
CoProcessor Opcodes

	Flag Settings						
s	-	if flag set					
x	-	always					
		amonial					

Oncode	Operation
Opcode	Operation
ADC <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	Rd = Rn + <s_op> + C</s_op>
ADD <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	Rd = Rn + <s_op></s_op>
AND <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	Rd = Rn & <s_op></s_op>
B <cond> <target_addr></target_addr></cond>	PC = PC + <offset></offset>
BIC <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	Rd = Rn & ! <s_op></s_op>
BL <cond> <target_addr></target_addr></cond>	LR = PC+4; PC = PC + <offset></offset>
BX <cond> Rm</cond>	PC = Rm; Mode=THUMB
CDP <cond> p<cp#>,o1,CRd,CRn,CRm,o2</cp#></cond>	execute coprocessor opcode
CMN <cond> Rn, <sh_op></sh_op></cond>	<flags> = Rn + <s_op></s_op></flags>
CMP <cond> Rn, <sh_op></sh_op></cond>	<flags> = Rn - <s_op></s_op></flags>
EOR <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	Rd = Rn ^ <s_op></s_op>
LDC <cond> p<cp_num>, CRd, #</cp_num></cond>	load coprocessor register with #
LDM <cond><adm> Rm, {reg list}^</adm></cond>	special, see doc
LDM <cond><adm> Rm<!-- -->, {reg list}</adm></cond>	for each in <reglist> = [Rn+=4]</reglist>
LDM <cond><adm> Rm<!-- -->, {reg list}^</adm></cond>	special, see doc
LDR <cond> Rd, Rn, #</cond>	Rd = [Rn + #]
LDR <cond>B Rd, Rn, #</cond>	Rd = [Rn + #]
LDR <cond>BT Rd, Rn, #</cond>	Rd = [Rn + #]
LDR <cond>H Rd, <address></address></cond>	Rd = [address]
LDR <cond>SB Rd, <address></address></cond>	Rd = [address]
LDR <cond>SH Rd, <address></address></cond>	Rd = [address]
LDR <cond>T Rd, Rn, #</cond>	Rd = [Rn + #]
MCR <cond> p<cp#>,o1,Rd,CRn,CRm,o2</cp#></cond>	move from co-cpu reg to ARM reg
MLA <cond><s> Rd, Rm, Rs, Rn</s></cond>	Rd = Rm * Rs + Rn
MOV <cond><s> Rd, <sh_op></sh_op></s></cond>	Rd = <s_op></s_op>
MRC <cond> p<cp#>,o1,Rd,CRn,CRm,o2</cp#></cond>	move from ARM reg to co-cpu reg
MRS <cond> Rd, CPSR</cond>	Rd = CPSR
MRS <cond> Rd, SPSR</cond>	Rd = SPSR
MSR <cond> CPSR_<fields>, Rm</fields></cond>	CPSR = Rm (fields pick bytes to copy)
MSR <cond> CPSR_f, #</cond>	CPSR = # (fields pick bytes to copy)
MSR <cond> SPSR_<fields>, Rm</fields></cond>	SPSR = Rm (fields pick bytes to copy)
MSR <cond> SPSR_f, #</cond>	SPSR = # (fields pick bytes to copy)
MUL <cond><s> Rd, Rm, Rs</s></cond>	Rd = Rm * Rs
MVN <cond><s> Rd, <sh_op></sh_op></s></cond>	Rd = - <s_op></s_op>
ORR <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	Rd = Rn <s_op></s_op>
RSB <cond><s> Rd, Rn, <sh op=""></sh></s></cond>	Rd = <s_op> - Rn</s_op>
RSC <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	Rd = <s_op> - Rn + C</s_op>
SBC <cond><s> Rd, Rn, <sh op=""></sh></s></cond>	Rd = Rn - <s_op> + C</s_op>
SMLAL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	RdHiRdLo = Rm*Rs+(RdHiRdLo)
SMULL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	RdHiRdLo = Rm*Rs
STC <cond> p<cp_num>, CRd, #</cp_num></cond>	Store coprocessor Reg with #
STM <cond><adm> Rm, {reg list}^</adm></cond>	special, see doc
STM <cond><adm> Rm<!-- -->, {reg list}</adm></cond>	[Rm+=4] = for each in <reglist></reglist>
STM <cond><adm> Rm<!-- -->, {reg list}</adm></cond>	special, see doc
STR <cond> Rd, Rn, #</cond>	
	[Rn + #] = Rd
STR <cond>B Rd, Rn, #</cond>	[Rn + #] = Rd
STR <cond>BT Rd, Rn, #</cond>	[Rn + #] = Rd
STR <cond>H Rd, <address></address></cond>	[address] = Rd
STR <cond>T Rd, Rn, #</cond>	[Rn + #] = Rd
SUB <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	Rd = Rn - <s_op></s_op>
SWI <swi_number></swi_number>	call software interrupt
SWP <cond> Rd, Rm, [Rn]</cond>	Rd = [Rn]; [Rn] = Rm
SWP <cond>B Rd, Rm, [Rn]</cond>	Rd = [Rn]; [Rn] = Rm
TEQ <cond> Rn, <sh_op></sh_op></cond>	<flags> = Rn ^ <s_op></s_op></flags>
TST <cond> Rn, <sh_op></sh_op></cond>	<flags> = Rn & <s_op></s_op></flags>
UMLAL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	RdHiRdLo = Rm*Rs+(RdHiRdLo)
UMULL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	RdHiRdLo = Rm*Rs

Data Processing Opcode
Load/Store Opcode
Branching Opcode
Multiplication Opcode
Other Opcodes
CoProcessor Opcodes

Flag	Description
riag	Description
Z	Zero Flag
С	Carry Flag
N	Negative Flag
V	Overflow Flag



Mnemonic	Description	Mnemonic	Description
ADC	Add with Carry	MSR	Move to Status Register
ADD	Add	MUL	Multiply
AND	Logical AND	MVN	Move Negative
В	Branch	ORR	Logical OR
BIC	Bit Clear	RSB	Reverse Subtract
BL	Branch and Link	RSC	Reverse Subtract with Carry
BX	Branch and Exchange	SBC	Subtract with Carry
CDP	Coprocessor Data Processing	SMLAL	Signed Long Multiply Accumulate
CMN	Compare Negative	SMULL	Signed Long Multiply
CMP	Compare	STC	Store Coprocessor
EOR	Logical Exclusive OR (XOR)	STM	Store Multiple
LDC	Load Coprocessor	STR	Store Register
LDM	Load Multiple	STRB	Store Register Byte
LDR	Load Register	STRBT	Store Register Byte Translate
LDRB	Load Register Byte	STRH	Store Register Half Word
LDRBT	Load Register Byte Translate	STRT	Store Register Translate
LDRH	Load Register Half Word	SUB	Subtract
LDRSB	Load Register Signed Byte	SWI	Software Interrupt
LDRSH	Load Register Signed Half Word	SWP	Swap
LDRT	Load Register Translate	SWPB	Swap Byte
MCR	Move to Coprocessor from ARM reg	TEQ	Test Equivalence
MLA	Multiply Accumulate	TST	Test
MOV	Move	UMLAL	Unsigned Long Multiply Accumulate
MRC	Move to ARM reg from Coprocessor	UMULL	Unsigned Long Multiply
MRS	Move from Status Register		

Opcode	31 30 29 28	3 27 26 25 24 23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
ADC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0 1 0 1 S	Rn	Rd	rotate #
ADC <cond><s> Rd, Rn, Rm OP # ADC<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S	Rn Rn	Rd Rd	shift # shift 0 Rm Rs 0 shift 1 Rm
ADD <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0 1 0 0 S	Rn	Rd	rotate #
ADD <cond><s> Rd, Rn, Rm OP # ADD<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S	Rn Rn	Rd Rd	shift # shift 0 Rm Rs 0 shift 1 Rm
AND <cond><s> Rd, RH, RH OF RS AND<cond><s> Rd, Rn, #</s></cond></s></cond>	cond	0 0 1 0 0 0 0 8	Rn	Rd	rotate #
AND <pre>AND</pre>	cond	0 0 0 0 0 0 0 8	Rn	Rd	shift # shift 0 Rm
AND <cond><s> Rd, Rn, Rm OP Rs B<cond> <target addr=""></target></cond></s></cond>	cond	0 0 0 0 0 0 0 S	Rn	Rd 24_bit	Rs 0 shift 1 Rm offset
BIC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 1 1 1 0 S	Rn	Rd	rotate #
BIC <cond><s> Rd, Rn, Rm OP # BIC<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0 1 1 1 0 S 0 0 0 1 1 1 0 S	Rn Rn	Rd Rd	shift # shift 0 Rm Rs 0 shift 1 Rm
BL <cond> <target addr=""></target></cond>	cond	1 0 1 1	Tai		offset
BX <cond> Rm</cond>	cond	0 0 0 1 0 0 1 0	SBO	SBO	SBO 0 0 0 1 Rm
CDP <cond> p<cp#>,<o1>,CRd,CRn,CRm,<o2> CMN<cond> Rn, #</cond></o2></o1></cp#></cond>	cond	1 1 1 0 op1 0 1 1 1	CRn Rn	CRd SBZ	cp_num op2 0 CRm rotate #
CMN <cond> Rn, Rm OP #</cond>	cond	0 0 0 1 0 1 1 1	Rn	SBZ	shift # shift 0 Rm
CMN <cond> Rn, Rm OP Rs CMP<cond> Rn, #</cond></cond>	cond	0 0 0 1 0 1 1 1	Rn Rn	SBZ SBZ	Rs 0 shift 1 Rm
CMP <cond> Rn, Rm OP #</cond>	cond	0 0 0 1 0 1 0 1	Rn	SBZ	shift # shift 0 Rm
CMP <cond> Rn, Rm OP Rs</cond>	cond	0 0 0 1 0 1 0 1	Rn	SBZ	Rs 0 shift 1 Rm
EOR <cond><s> Rd, Rn, # EOR<cond><s> Rd, Rn, Rm OP #</s></cond></s></cond>	cond	0 0 1 0 0 0 1 S 0 0 0 0 0 0 1 S	Rn Rn	Rd Rd	rotate # shift # shift 0 Rm
EOR <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 0 0 0 1 S	Rn	Rd	Rs 0 shift 1 Rm
LDC <cond> p<cp_num>, CRd, # LDM<cond><addrmode> Rm, reg list^</addrmode></cond></cp_num></cond>	cond	1 1 0 P U N W 1 1 0 0 P U 1 0 1	Rn Rn	CRd 0	cp_num # register list
LDM <cond><addrmode> Rm<!-- -->, reg list</addrmode></cond>	cond	1 0 0 P U 0 W 1	Rn		register list
LDM <cond><addrmode> Rm<!-- -->, reg list^</addrmode></cond>	cond	1 0 0 P U 1 W 1	Rn	1 Pd	register list
LDR <cond> Rd, Rn, # LDR<cond> Rd, Rn, #</cond></cond>	cond	0 1 0 P U 0 W 1 0 1 1 P U 0 W 1	Rn Rn	Rd Rd	# shift # shift 0 Rm
LDR <cond>B Rd, Rn, #</cond>	cond	0 1 0 P U 1 W 1	Rn	Rd	#
LDR <cond>B Rd, Rn, # LDR<cond>BT Rd, Rn, #</cond></cond>	cond	0 1 1 P U 1 W 1 0 1 0 0 U 1 1 1	Rn Rn	Rd Rd	shift # shift 0 Rm
LDR <cond>BT Rd, Rn, #</cond>	cond	0 1 1 0 U 1 1 1	Rn	Rd	shift # shift 0 Rm
LDR <cond>H Rd, <address> LDR<cond>SB Rd, <address></address></cond></address></cond>	cond	0 0 0 P U I W 1 0 0 0 P U I W 1	Rn Rn	Rd Rd	addr_mode 1 0 1 1 addr_mode addr_mode 1 1 0 1 addr_mode
LDR <cond>SH Rd, <address></address></cond>	cond	0 0 0 P U I W 1	Rn	Rd	addr_mode 1 1 1 1 addr_mode
LDR <cond>T Rd, Rn, #</cond>	cond	0 1 0 0 U 0 1 1	Rn	Rd	#
LDR <cond>T Rd, Rn, # MCR<cond> p<cp#>,<o1>,Rd,CRn,CRm,<o2></o2></o1></cp#></cond></cond>	cond	0 1 1 0 U 0 1 1 1 1 1 0 op1 0	Rn CRn	Rd Rd	shift # shift 0 Rm cp_num op2 1 CRm
MLA <cond><s> Rd, Rm, Rs, Rn</s></cond>	cond	0 0 0 0 0 0 1 8	Rd	Rn	Rs 1 0 0 1 Rm
MOV <cond><s> Rd, #</s></cond>	cond	0 0 1 1 1 0 1 5	SBZ	Rd	rotate #
MOV <cond><s> Rd, Rm OP # MOV<cond><s> Rd, Rm OP Rs</s></cond></s></cond>	cond	0 0 0 1 1 0 1 S 0 0 0 1 1 0 1 S	SBZ SBZ	Rd Rd	shift # shift 0 Rm Rs 0 shift 1 Rm
MRC <cond> p<cp#>,,Rd,CRn,CRm,<o2></o2></cp#></cond>	cond	1 1 1 0 op1 1	CRn	Rd	cp_num op2 1 CRm
MRS <cond> Rd, CPSR MRS<cond> Rd, SPSR</cond></cond>	cond	0 0 0 1 0 0 0 0	SBO SBO	Rd Rd	SBZ SBZ
MSR <cond> CPSR_<fields>, Rm</fields></cond>	cond	0 0 0 1 0 0 1 0	field_mask	SBO	SBZ 0 Rm
MSR <cond> CPSR_f, #</cond>	cond	0 0 1 1 0 0 1 0	field_mask	SBO	rotate #
MSR <cond> SPSR_<fields>, Rm MSR<cond> SPSR_f, #</cond></fields></cond>	cond	0 0 0 1 0 1 1 0	field_mask field_mask	SBO SBO	SBZ 0 Rm
MUL <cond><s> Rd, Rm, Rs</s></cond>	cond	0 0 0 0 0 0 0 S	Rd	SBZ	Rs 1 0 0 1 Rm
MVN <cond><s> Rd, # MVN<cond><s> Rd, Rm OP #</s></cond></s></cond>	cond	0 0 1 1 1 1 1 S 0 0 0 1 1 1 1 S	SBZ SBZ	Rd Rd	rotate # shift # shift 0 Rm
MVN <cond><s> Rd, Rm OP Rs</s></cond>	cond	0 0 0 1 1 1 1 8	SBZ	Rd	Rs 0 shift 1 Rm
ORR <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 1 1 0 0 S 0 0 0 1 1 0 0 S	Rn	Rd	rotate # shift 0 Rm
ORR <cond><s> Rd, Rn, Rm OP # ORR<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0 1 1 0 0 S 0 0 0 1 1 0 0 S	Rn Rn	Rd Rd	shift # shift 0 Rm Rs 0 shift 1 Rm
RSB <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0 0 1 1 S	Rn	Rd	rotate #
RSB <cond><s> Rd, Rn, Rm OP # RSB<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0 0 0 1 1 S 0 0 0 0 0 1 1 S	Rn Rn	Rd Rd	shift # shift 0 Rm Rs 0 shift 1 Rm
RSC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0 1 1 1 S	Rn	Rd	rotate #
RSC <cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0 0 1 1 1 S 0 0 0 0 1 1 1 S	Rn Rn	Rd Rd	shift # shift 0 Rm Rs 0 shift 1 Rm
SBC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0 1 1 0 8		Rd	rotate #
SBC <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 0 1 1 0 S	Rn	Rd	shift # shift 0 Rm
SBC <cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdHi, Rm, Rs</s></cond></s></cond>	cond	0 0 0 0 1 1 0 S 0 0 0 0 1 1 1 S	Rn RdHi	Rd RdLo	Rs 0 shift 1 Rm Rs 1 0 0 1 Rm
SMULL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	cond	0 0 0 0 1 1 0 S	RdHi	RdLo	Rs 1 0 0 1 Rm
STC <cond> p<cp_num>, CRd, #</cp_num></cond>	cond	1 1 0 P U N W 0	Rn Pn	CRd	cp_num #
STM <cond><addrmode> Rm, reg list^ STM<cond><addrmode> Rm<!-- -->, reg list</addrmode></cond></addrmode></cond>	cond	1 0 0 P U 1 0 0 1 0 0 P U 0 W 0	Rn Rn	0	register list register list
STM <cond><addrmode> Rm<!-- -->, reg list^</addrmode></cond>	cond	1 0 0 P U 1 W 0	Rn	1	register list
STR <cond> Rd, Rn, # STR<cond> Rd, Rn, #</cond></cond>	cond	0 1 0 P U 0 W 0	Rn Rn	Rd Rd	# shift # shift 0 Rm
STR <cond>B Rd, Rn, #</cond>	cond	0 1 0 P U 1 W 0	Rn	Rd	#
STR <cond>B Rd, Rn, #</cond>	cond	0 1 1 P U 1 W 0		Rd	shift # shift 0 Rm
STR <cond>BT Rd, Rn, # STR<cond>BT Rd, Rn, #</cond></cond>	cond	0 1 0 0 U 1 1 0 0 1 1 0 U 1 1 0	Rn Rn	Rd Rd	# shift # shift 0 Rm
STR <cond>H Rd, <address></address></cond>	cond	0 0 0 P U I W 0	Rn	Rd	addr_mode 1 0 1 1 addr_mode
STR <cond>T Rd, Rn, # STR<cond>T Rd, Rn, #</cond></cond>	cond	0 1 0 0 U 0 1 0 0 1 1 0 U 0 1 0	Rn Rn	Rd Rd	# shift # shift 0 Rm
STR <cond>T Rd, RN, # SUB<cond><s> Rd, Rn, #</s></cond></cond>	cond	0 0 1 0 0 1 0 8		Rd	shift # shift 0 Rm rotate #
SUB <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 0 0 1 0 S	Rn	Rd	shift # shift 0 Rm
SUB <cond><s> Rd, Rn, Rm OP Rs SWI <swi_number></swi_number></s></cond>	cond	0 0 0 0 0 1 0 S	Rn	Rd swin	Rs 0 shift 1 Rm umber
SWP <cond> Rd, Rm, [Rn]</cond>	cond	0 0 0 1 0 0 SBZ		Rd	SBZ 1 0 0 1 Rm
SWP <cond>B Rd, Rm, [Rn] TEQ<cond> Rn, #</cond></cond>	cond	0 0 0 1 0 1 SBZ	Rn Rn	Rd SBZ	SBZ 1 0 0 1 Rm
TEQ <cond> Rn, # TEQ<cond> Rn, Rm OP #</cond></cond>	cond	0 0 1 1 0 0 1 1 0 0 0 1 1	Rn Rn	SBZ	rotate # shift # shift 0 Rm
TEQ <cond> Rn, Rm OP Rs</cond>	cond	0 0 0 1 0 0 1 1	Rn	SBZ	Rs 0 shift 1 Rm
TST <cond> Rn, # TST<cond> Rn, Rm OP #</cond></cond>	cond	0 0 1 1 0 0 0 1	Rn Rn	SBZ SBZ	rotate # shift # shift 0 Rm
TST <cond> Rn, Rm OP #</cond>	cond	0 0 0 1 0 0 0 1	Rn	SBZ	Rs 0 shift 1 Rm
UMLAL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	cond	0 0 0 0 1 0 1 8	RdHi	RdLo	Rs 1 0 0 1 Rm
UMULL <cond><s> RdLo, RdHi, Rm, Rs Undefined Instruction</s></cond>	cond	0 0 0 0 1 0 0 S 0 0 0 x x x x 0	RdHi x x x x	RdLo x x x x	Rs 1 0 0 1 Rm x x x x x 1 1 0 1 x x x x x
Undefined Instruction	cond	0 0 0 x x x x 0	x x x x	x x x x	x x x x 1 1 1 1 x x x x
Undefined Instruction UNPREDICTABLE	cond				X X X X X X X X X X X X X X X X X X X
UNPREDICTABLE UNPREDICTABLE	cond	1 0 0 X X 1 1 0 1 0 0 X X 1 1 1			x x x x x x x x x x x x x x x x x x x