Opcode	31 30 29 28	27 26 25	5 24	23 1	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 0	S	Rn	Rd	shift #	shift 0	Rm
AND <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond		0		0 0	S	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 I W	S	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		_	I W	1	Rn	Rd	addr_mode	1 0 1 1	addr_mode
Undefined Instruction	cond	0 0 0	_	х	_	0	x x 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	_	_	I W	1	Rn	Rd	addr_mode	1 1 0 1	addr_mode
Undefined Instruction LDR <cond>SH Rd, <address></address></cond>	cond	0 0 0	_	U	x x	1	X X X X	x x x x	x x x x addr_mode	1 1 1 1	x x x x addr_mode
EOR <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0		_	0 1	s	Rn	Rd	shift #	shift 0	Rm
EOR <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	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 1	S	Rd	Rn	Rs	1 0 0 1	Rm
SUB <cond><s> Rd, Rn, Rm OP # SUB<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0	_	_	1 0	S	Rn Rn	Rd Rd	shift #	shift 0 shift 1	Rm Rm
RSB <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0	_		1 1	S	Rn	Rd	shift #	shift 0	Rm
RSB <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0		0	1 1	S	Rn	Rd	Rs	0 shift 1	Rm
ADD <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0			0 0	S	Rn	Rd	shift #	shift 0	Rm
ADD <cond><s> Rd, Rn, Rm OP Rs UMULL<cond><s> RdLo, RdHi, Rm, Rs</s></cond></s></cond>	cond	0 0 0			0 0	S	Rn RdHi	Rd RdLo	Rs Rs	0 shift 1	Rm Rm
ADC <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0	_		0 0	S	Rn	Rd	shift #	shift 0	Rm
ADC <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0			0 1	S	Rn	Rd	Rs	0 shift 1	Rm
UMLAL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	cond	0 0 0	_		0 1	S	RdHi	RdLo	Rs	1 0 0 1	Rm
SBC <cond><s> Rd, Rn, Rm OP # SBC<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0			1 0	S	Rn	Rd Rd	shift #	shift 0	Rm
SBC <cond><s> Rd, Rn, Rm OP Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs</s></cond></s></cond>	cond	0 0 0			1 0	S	Rn RdHi	RdLo	Rs Rs	0 shift 1	Rm Rm
RSC <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0	_		1 1	S	Rn	Rd	shift #	shift 0	Rm
RSC <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0	_	_	1 1	S	Rn	Rd	Rs	0 shift 1	Rm
SMLAL <cond><s> RdLo, RdHi, Rm, Rs</s></cond>	cond	0 0 0	_	-	1 1	S	RdHi	RdLo	Rs	1 0 0 1	Rm
TST <cond> Rn, Rm OP # TST<cond> Rn, Rm OP Rs</cond></cond>	cond	0 0 0	_	_	0 0	1	Rn Rn	SBZ SBZ	shift # Rs	shift 0 shift 1	Rm Rm
MRS <cond> Rd, CPSR</cond>	cond	0 0 0	_		0 0	0	SBO	Rd	110	SBZ	14
SWP <cond> Rd, Rm, [Rn]</cond>	cond	0 0 0	1	0	0 SI	ΒZ	Rn	Rd	SBZ	1 0 0 1	Rm
TEQ <cond> Rn, Rm OP #</cond>	cond	0 0 0	_		0 1	1	Rn	SBZ	shift #	shift 0	Rm
TEQ <cond> Rn, Rm OP Rs  MSR<cond> CPSR <fields>, Rm</fields></cond></cond>	cond	0 0 0	_	_	0 1	0	Rn field mask	SBZ SBO	Rs SBŽ	0 shift 1	Rm Rm
BX <cond> Rm</cond>	cond	0 0 0			0 1	0	SBO	SBO	SBO	0 0 0 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	Rn	SBZ	Rs	0 shift 1	Rm
MRS <cond> Rd, SPSR SWP<cond>B Rd, Rm, [Rn]</cond></cond>	cond	0 0 0		_	1 0 1 SI	0 BZ	SBO Rn	Rd Rd	SBZ	SBZ	Rm
CMN <cond> Rn, Rm OP #</cond>	cond	0 0 0	_		1 1	_	Rn	SBZ	shift #	shift 0	Rm
CMN <cond> Rn, Rm OP Rs</cond>	cond	0 0 0	_		1 1	1	Rn	SBZ	Rs	0 shift 1	Rm
MSR <cond> SPSR_<fields>, Rm</fields></cond>	cond	0 0 0	_	-	1 1	0	field_mask	SBO	SBZ		Rm
ORR <cond><s> Rd, Rn, Rm OP # ORR<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond>	cond	0 0 0	_		0 0	S	Rn	Rd Rd	shift #	shift 0	Rm
MOV <cond><s> Rd, Rm OP #</s></cond>	cond	0 0 0		_	0 1	S	Rn SBZ	Rd	Rs shift #	0 shift 1 shift 0	Rm Rm
MOV <cond><s> Rd, Rm OP Rs</s></cond>	cond	0 0 0	_		0 1	S	SBZ	Rd	Rs	0 shift 1	Rm
BIC <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0		_	1 0	S	Rn	Rd	shift #	shift 0	Rm
BIC <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0	_	_	1 0	S	Rn	Rd	Rs	0 shift 1	Rm
MVN <cond><s> Rd, Rm OP # MVN<cond><s> Rd, Rm OP Rs</s></cond></s></cond>	cond	0 0 0		_	1 1	S	SBZ SBZ	Rd Rd	shift #	shift 0	Rm Rm
AND <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1	_	0	0 0	S	Rn	Rd	rotate		#
EOR <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1	-		0 1	S	Rn	Rd	rotate		#
SUB <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1	_		1 0	S	Rn	Rd	rotate		#
RSB <cond><s> Rd, Rn, # ADD<cond><s> Rd, Rn, #</s></cond></s></cond>	cond	0 0 1	Ť	_	1 1 0 0	S	Rn Rn	Rd Rd	rotate rotate		# #
ADC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1	_		0 1	s	Rn	Rd	rotate		#
SBC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1	0	1	1 0	S	Rn	Rd	rotate		#
RSC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1	_		1 1	S	Rn	Rd	rotate		#
TST <cond> Rn, # TEQ<cond> Rn, #</cond></cond>	cond	0 0 1	_	_	0 0	1	Rn Rn	SBZ SBZ	rotate rotate		<u>#</u> #
MSR <cond> CPSR f, #</cond>	cond	0 0 1		0		0	field mask	SBO	rotate		#
CMP <cond> Rn, #</cond>	cond				1 0		Rn	SBZ	rotate		#
CMN <cond> Rn, #</cond>	cond				1 1	_	Rn	SBZ	rotate		#
MSR <cond> SPSR_f, # ORR<cond><s> Rd, Rn, #</s></cond></cond>	cond	0 0 1			1 1		field_mask Rn	SBO Rd	rotate rotate		# #
MOV <cond><s> Rd, #</s></cond>	cond				0 1		SBZ	Rd	rotate		<del>*</del>
BIC <cond><s> Rd, Rn, #</s></cond>	cond		_		1 0		Rn	Rd	rotate		#
MVN <cond><s> Rd, #</s></cond>	cond	0 0 1			1 1		SBZ	Rd	rotate		#
STR <cond> Rd, Rn, # LDR<cond> Rd, Rn, #</cond></cond>	cond				0 W		Rn	Rd		#	
LDR <cond> Rd, Rn, # STR<cond>B Rd, Rn, #</cond></cond>	cond		P		0 W 1 W		Rn Rn	Rd Rd		#	
LDR <cond>B Rd, Rn, #</cond>	cond				1 W		Rn	Rd		#	
STR <cond>T Rd, Rn, #</cond>	cond	0 1 0					Rn	Rd		#	
LDR <cond>T Rd, Rn, #</cond>	cond		0			1	Rn	Rd		#	
STR <cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, #</cond></cond>	cond	0 1 0		_	1 1 1 1	1	Rn Rn	Rd Rd		#	
STR <cond> Rd, Rn, #</cond>	cond	0 1 1	Р	U	0 W	0	Rn	Rd	shift #	shift 0	Rm
LDR <cond> Rd, Rn, #</cond>	cond	0 1 1					Rn	Rd	shift #	shift 0	Rm
I dmpd.p.pd p. "	cond	0 1 1			1 W		Rn	Rd	shift #	shift 0	Rm Pm
STR <cond>B Rd, Rn, #</cond>	cond		P 0		1 W 0 1		Rn Rn	Rd Rd	shift #	shift 0	Rm Rm
STR <cond>B Rd, Rn, # LDR<cond>B Rd, Rn, # STR<cond>T Rd, Rn, #</cond></cond></cond>	cond						Rn	Rd	shift #	shift 0	Rm
LDR <cond>B Rd, Rn, #</cond>		0 1 1	0		<u> </u>						Rm
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, #</cond></cond></cond></cond>	cond cond cond	0 1 1	0	U	1 1	0	Rn	Rd	shift #	shift 0	
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, #</cond></cond></cond></cond></cond>	cond cond cond cond	0 1 1	0	U	1 1 1 1	1	Rn	Rd	shift #	shift 0	Rm
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # Undefined Instruction</cond></cond></cond></cond></cond></cond>	cond cond cond cond cond	0 1 1 0 1 1 0 1 1	0 0 x	U U X	1 1 1 1 x x	1 x	Rn x x x x	Rd	shift#	shift 0 x x x x 1	
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, #</cond></cond></cond></cond></cond>	cond cond cond cond	0 1 1 0 1 1 0 1 1	0 0 x	U X U	1 1 1 1	1 x	Rn	Rd	shift #	shift 0 x x x x 1 er list	Rm
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # Undefined Instruction STM<cond><addrmode> Rm<!-- -->, reg list LDM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list</addrmode></cond></addrmode></cond></addrmode></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 1 0 1 1 0 1 1 1 0 0 1 0 0	0 0 x P P	U X U U U	1 1 1 1 x x 0 W 0 W	1 x 0 1	Rn x x x x Rn Rn Rn	Rd x x x x x	shift # x x x x regist regist	shift 0 x x x 1 er list er list ster list	Rm x x x x
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>STR   LDR<cond>Addrmode&gt; Rm<!-- -->, reg list   LDM<cond><addrmode> Rm<!-- -->, reg list   LDM<cond><addrmode> Rm<!-- -->, reg list   UNPREDICTABLE</addrmode></cond></addrmode></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 1 0 1 1 0 1 1 1 0 0 1 0 0 1 0 0	0 0 X P P P	U X U U U X U X U X U X U X X X U X X X U X	1 1 1 1 x x 0 W 0 W 1 0	1 x 0 1 0	Rn x x x x Rn Rn Rn x x x x	Rd x x x x x x x x x x x x x x x x x x x	shift # x x x x regist regist regist x x x x	shift 0 x x x 1 er list er list ster list x x x x x	Rm x x x x
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<stm<stm ldr<cond="">STM<cond>STM<stm ldm<cond="">STM<stm ld<="" ldm="" ldm<stm="" td=""><td>cond cond cond cond cond cond cond cond</td><td>0 1 1 0 1 1 0 1 1 1 0 0 1 0 0 1 0 0 1 0 0</td><td>0 0 P P P X</td><td>U x U U U U U U U U U U U U U U U U U U</td><td>1 1 1 1 x x 0 W 0 W 1 0 1 1</td><td>1 x 0 1 0 0</td><td>Rn x x x x Rn Rn Rn x x x x</td><td>Rd x x x x x 0</td><td>shift #  x x x x  regist  regist  regist  x x x x  regist</td><td>shift 0 x x x 1 er list er list ster list x x x x ster list</td><td>Rm   x</td></stm></stm></cond></stm<stm></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 1 0 1 1 0 1 1 1 0 0 1 0 0 1 0 0 1 0 0	0 0 P P P X	U x U U U U U U U U U U U U U U U U U U	1 1 1 1 x x 0 W 0 W 1 0 1 1	1 x 0 1 0 0	Rn x x x x Rn Rn Rn x x x x	Rd x x x x x 0	shift #  x x x x  regist  regist  regist  x x x x  regist	shift 0 x x x 1 er list er list ster list x x x x ster list	Rm   x
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>STR   LDR<cond>Addrmode&gt; Rm<!-- -->, reg list   LDM<cond><addrmode> Rm<!-- -->, reg list   LDM<cond><addrmode> Rm<!-- -->, reg list   UNPREDICTABLE</addrmode></cond></addrmode></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 1 0 1 1 0 1 1 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0	0 0 P P P X	U X U U X U X U X X U X X X X X X X X X	1 1 1 1 x x 0 W 0 W 1 0	1 x 0 1 0 0 1	Rn x x x x Rn Rn Rn x x x x	Rd x x x x x 0	shift # x x x x regist regist regist x x x x regist x x x x	shift 0 x x x x 1 er list er list x x x x x ster list x x x x x	Rm   x
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>Cond&gt;Cond STM<cond>Cond&gt;Cond&gt;Cond STM<cond>Cond&gt;Cond&gt;Cond STM<cond>Cond&gt;Cond&gt;Cond STM<cond>Cond&gt;Cond STM<cond>Cond&gt;Cond STM<cond>Cond&gt;Cond STM<cond>Cond&gt;Cond STM<cond>Cond&gt;Cond STM<cond>Cond&gt;Cond STM<cond>Cond&gt;Cond STM<cond>Cond STM 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	0 1 1 0 1 1 1 0 0 1 0 0	0 0 P P P X P X	U X U X U X U X U X U X U X U X U X U U X U X U U X U	1 1 1 1 x x 0 W 0 W 1 0 1 1 1 0	1 x 0 1 0 0 1 1	Rn	Rd	shift #  x x x x  regist  regist  x x x x  regist  x x x x  regist  regist  regist  regist  regist  regist  regist  regist  regist	shift 0 x x x 1 er list er list ster list x x x x ster list	Rm   x
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # Undefined Instruction STM<cond><addrmode> Rm<!-- -->, reg list LDM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm, reg list STM<cond><addrmode> Rm, reg list STM<cond><addrmode> Rm, reg list STM<cond><addrmode> Rm, reg list UNPREDICTABLE LDM<cond><addrmode> Rm, reg list UNPREDICTABLE STM<cond><addrmode> Rm<!-- -->, reg list LDM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list LDM<cond><addrmode> Rm<!-- -->, reg list B</addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 1 0 1 1 1 0 0 1 1 0 0 1 0 0	0 0 P P P X P X	U X U X U X U X U X U X U X U X U X U U X U X U U X U	1 1 1 1 x x 0 W 0 W 1 0 1 1 1 0 1 1	1 x 0 1 0 0 1 1	Rn	Rd	shift # x x x x x regist regist x x x x x regis x x x x x regis x regis	shift 0 x x x x 1 er list er list x x x x x ster list x x x x x ster list	Rm   x
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>STR   LDR<cond>Addrmode&gt; Rm<!-- -->, reg list   LDM<cond><addrmode> Rm<!-- -->, reg list   LDM<cond><addrmode> Rm<!-- -->, reg list   STM<cond><addrmode> Rm, reg list   UNPREDICTABLE LDM<cond><addrmode> Rm, reg list   UNPREDICTABLE STM<cond><addrmode> Rm<!-- -->, reg list   LDM<cond><addrmode> Rm<!-- -->, reg list   LDM<cond><addrmode> Rm<!-- -->, reg list   B E Cond&gt;<addrmode> Rm<!-- -->, reg list   B E STM   B Cond&gt;<addrmode> Rm Cond&gt;<addrmode> Rm   B Cond&gt;<addrmode> Cond&gt;</addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 1 0 1 1 0 1 1 1 0 0 1 0 0	0 0 x P P P X P X P P	U X U U V V V V V V V V V V V V V V V V	1 1 x x 0 W 0 W 1 0 1 1 1 0 1 1 W	1 x 0 1 0 0 1 1 1	Rn	Rd	shift # x x x x x regist regist x x x x x regis x x x x x regis x regis x regis	shift 0 x x x x 1 er list er list ster list x x x x x ster list x x x x x ster list x x x x x ster list	X X X X X X X X X X X X X X X X X X X
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>Cond&gt;E Rd, Rn, # LDR<cond>Cond&gt;Cond Cond&gt;Cond&gt;Cond Cond&gt;Cond&gt;Cond Cond&gt;Cond&gt;Cond Cond&gt;Cond&gt;Cond Cond&gt;Cond&gt;Cond Cond&gt;Cond Cond&gt;Cond Cond&gt;Cond Cond&gt;Cond Cond Cond&gt;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 1 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0	0 0 x P P P 0 x P 0 P	U X U U V V V V V V V V V V V V V V V V	1 1 1 1 x x 0 W 0 W 1 0 1 1 1 0 1 1	1 x 0 1 0 0 1 1 1	Rn	Rd	shift # x x x x x regist regist x x x x x regis x x x x x regis x regis	shift 0 x x x x 1 er list er list ster list x x x x x ster list x x x x x ster list ster list ster list	Rm   x
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<stm stc<cond="" stm="" stm<stm="">STM STM STC<cond>STM STM STC<cond>SCD STM STM STM STM STC<cond>SCD STM STM STM STM STM STM STM STM STM STM</cond></cond></cond></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></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 1 0 1 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0	0 0 x P P P x P P P P P P P P P P	U V V V V V V V V V V V V V V V V V V V	1 1 1 1	1 x 0 1 0 0 1 1 1	Rn	Rd	shift # x x x x x regist regist x x x x x x x x x x x x regist x x x x x x x regist x x x x x x x regist x x x x x x x x regist x x x x x x x x regist x x x x x x x x regist x x x x x x x x x regist x x x x x x x x x regist x x x x x x x x x regist x x x x x x x x x x regist x x x x x x x x x x x x x regist x x x x x x x x x x x x x x regist x x x x x x x x x x x x x x x x regist x x x x x x x x x x x x x x x x x regist x x x x x x x x x x x x x x x x x x x	shift 0 x x x x 1 er list er list ster list x	Rm
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<stm<stm<stm<stm<stm<stm<stm<stm<stm<< td=""><td>cond cond cond cond cond cond cond cond</td><td>0 1 1 0 1 1 1 0 0 1 1 0 0 1 0 0 0 0</td><td>0 0 P P P V V V V P O O O</td><td>U V V V V V V V V V V V V V V V V V V V</td><td>1 1 1 1</td><td>1 x 0 1 0 0 1 1 0 1</td><td>Rn</td><td>Rd</td><td>shift # x x x x regist regist regist x x x x regis x x x x regis y regis</td><td>shift 0 x x x x 1 er list er list ster list x x x x x ster list x x x x x ster list x x x x x ster list cop2 0 op2 1</td><td>  Rm                                    </td></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></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 1 0 1 1 1 0 0 1 1 0 0 1 0 0 0 0	0 0 P P P V V V V P O O O	U V V V V V V V V V V V V V V V V V V V	1 1 1 1	1 x 0 1 0 0 1 1 0 1	Rn	Rd	shift # x x x x regist regist regist x x x x regis x x x x regis y regis	shift 0 x x x x 1 er list er list ster list x x x x x ster list x x x x x ster list x x x x x ster list cop2 0 op2 1	Rm
LDR <cond>B Rd, Rn, # STR<cond>T Rd, Rn, # LDR<cond>T Rd, Rn, # STR<cond>T Rd, Rn, # STR<cond>BT Rd, Rn, # STR<cond>BT Rd, Rn, # LDR<cond>BT Rd, Rn, # LDR<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<cond>STM<stm stc<cond="" stm="" stm<stm="">STM STM STC<cond>STM STM STC<cond>SCD STM STM STM STM STC<cond>SCD STM STM STM STM STM STM STM STM STM STM</cond></cond></cond></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></cond></cond></cond></cond></cond>	cond cond cond cond cond cond cond cond	0 1 1 0 1 1 1 0 0 1 1 1 0 0 1 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0	0 0 0 P P P 0 X P 0 P 0 0	U V V V V V V V V V V V V V V V V V V V	1 1 1 1	1 x 0 1 0 0 1 1 1	Rn	Rd x x x x x 0 x x x x x x 1 1 24_bit CRd CRd CRd Rd Rd	shift # x x x x x regist regist x x x x x x x x x x x x regist x x x x x x x regist x x x x x x x regist x x x x x x x x regist x x x x x x x x regist x x x x x x x x regist x x x x x x x x x regist x x x x x x x x x regist x x x x x x x x x regist x x x x x x x x x x regist x x x x x x x x x x x x x regist x x x x x x x x x x x x x x regist x x x x x x x x x x x x x x x x regist x x x x x x x x x x x x x x x x x regist x x x x x x x x x x x x x x x x x x x	shift 0 x x x x 1 er list er list ster list x	Rm

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	<del>-</del>				
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	X
CMP <cond> Rn, <sh_op></sh_op></cond>	-	х	х	х	х
EOR <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-	s	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	Ì			
LDR <cond>B Rd, Rn, #</cond>	byte				
LDR <cond>BT Rd, Rn, #</cond>	translate byte	Ì			
LDR <cond>H Rd, <address></address></cond>	half word				
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	
MRC <cond> p<cp#>,o1,Rd,CRn,CRm,o2</cp#></cond>	if Rd = r15 then flags	*	*	*	*
Fine Condy p Cop#y, OI, Rd, CRII, CRIII, OZ	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	s	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
SBC <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-			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	s	s	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				
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	L			
SUB <cond><s> Rd, Rn, <sh_op></sh_op></s></cond>	-	s	s	s	s
SWI <swi_number></swi_number>	-	l			
SWP <cond> Rd, Rm, [Rn]</cond>	-	Ì			
SWP <cond>B Rd, Rm, [Rn]</cond>	-	L			
TEQ <cond> Rn, <sh_op></sh_op></cond>	-		X		
TST <cond> Rn, <sh_op></sh_op></cond>	-	_	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	g Settings	
s	-	if	flag	set	
×	-	alv	vays		
*	_	SDE	ecial		

Opcode	Operation
	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>
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>Rd, Rn, #</cond>	Rd = [Rn + #] Rd = [Rn + #]
LDR <cond>BT Rd, Rn, #</cond>	Rd = [Rn + #] 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></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}^ STR<cond> Rd, Rn, #</cond></adm></cond>	special, see doc
	[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 &amp; <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
Z	<b>Z</b> ero 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					
ADC <cond><s> Rd, Rn, #</s></cond>	31 30 29 28 cond	8 27 26 25 24 23 22 21 2 0 0 1 0 1 0 1 5		15 14 13 12 Rd	11 10 9 8 7 6 5 4 3 2 1 0 rotate #
ADC <cond><s> Rd, Rn, #  ADC<cond><s> Rd, Rn, Rm OP #</s></cond></s></cond>	cond	0 0 0 0 1 0 1 0 1		Rd	shift # shift 0 Rm
ADC <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 0 1 0 1 8		Rd	Rs 0 shift 1 Rm
ADD <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0 1 0 0 5	Rn	Rd	rotate #
ADD <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 0 1 0 0 5		Rd	shift # shift 0 Rm
ADD <cond><s> Rd, Rn, Rm OP Rs AND<cond><s> Rd, Rn, #</s></cond></s></cond>	cond	0 0 0 0 1 0 0 5		Rd Rd	Rs 0 shift 1 Rm
AND <cond><s> kd, kn, #  AND<cond><s> Rd, Rn, Rm OP #</s></cond></s></cond>	cond	0 0 1 0 0 0 0 5		Rd	shift # shift 0 Rm
AND <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 0 0 0 0 0		Rd	Rs 0 shift 1 Rm
B <cond> <target addr=""></target></cond>	cond	1 0 1 0		24_bit	offset
BIC <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 1 1 1 0 5		Rd	rotate #
BIC <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 1 1 1 0 5		Rd	shift # shift 0 Rm
BIC <cond><s> Rd, Rn, Rm OP Rs BL<cond> <target addr=""></target></cond></s></cond>	cond	0 0 0 1 1 1 0 5	Rn	Rd 24 bit	Rs 0 shift 1 Rm
BX <cond> Rm</cond>	cond	0 0 0 1 0 0 1 0	SBO	SBO SBO	_offset SBO001 Rm
CDP <cond> p<cp#>,<o1>,CRd,CRn,CRm,<o2></o2></o1></cp#></cond>	cond	1 1 1 0 op1	CRn	CRd	cp_num op2 0 CRm
CMN <cond> Rn, #</cond>	cond	0 0 1 1 0 1 1 1	Rn	SBZ	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</cond>	cond	0 0 0 1 0 1 1 1	Rn	SBZ	Rs 0 shift 1 Rm
CMP <cond> Rn, # CMP<cond> Rn, Rm OP #</cond></cond>	cond	0 0 1 1 0 1 0 1	Rn Rn	SBZ SBZ	rotate # 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, #</s></cond>	cond	0 0 1 0 0 0 1 5		Rd	rotate #
EOR <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 0 0 0 1 5	Rn	Rd	shift # shift 0 Rm
EOR <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 0 0 0 1 5		Rd	Rs 0 shift 1 Rm
LDC <cond> p<cp_num>, CRd, #</cp_num></cond>	cond	1 1 0 P U N W 1		CRd	cp_num #
LDM <cond><addrmode> Rm, reg list^</addrmode></cond>	cond	1 0 0 P U 1 0 1	Rn Pn	0	register list
LDM <cond><addrmode> Rm<!-- -->, reg list LDM<cond><addrmode> Rm<!-- -->, reg list^</addrmode></cond></addrmode></cond>	cond	1 0 0 P U 0 W 1	Rn Rn	1	register list register list
LDR <cond><addrmode> km<!-- -->, reg 11st LDR<cond> Rd, Rn, #</cond></addrmode></cond>	cond	0 1 0 P U 0 W 1	Rn	Rd	register list #
LDR <cond> Rd, Rn, #</cond>	cond	0 1 1 P U 0 W 1	Rn	Rd	shift # shift 0 Rm
LDR <cond>B Rd, Rn, #</cond>	cond	0 1 0 P U 1 W 1		Rd	#
LDR <cond>B Rd, Rn, #</cond>	cond	0 1 1 P U 1 W 1		Rd	shift # shift 0 Rm
LDR <cond>BT Rd, Rn, #</cond>	cond	0 1 0 0 U 1 1 1		Rd	#
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></address></cond>	cond	0 0 0 P U I W 1		Rd Rd	addr_mode 1 0 1 1 addr_mode
LDR <cond>SB Rd, <address> LDR<cond>SH Rd, <address></address></cond></address></cond>	cond	0 0 0 P U I W 1		Rd Rd	addr_mode 1 1 0 1 addr_mode addr_mode 1 1 1 1 addr_mode
LDR <cond>SH Rd, <address> LDR<cond>T Rd, Rn, #</cond></address></cond>	cond	0 1 0 0 0 0 1 1	Rn	Rd	#
LDR <cond>T Rd, Rn, #</cond>	cond	0 1 1 0 U 0 1 1	Rn	Rd	shift # shift 0 Rm
MCR <cond> p<cp#>,<o1>,Rd,CRn,CRm,<o2></o2></o1></cp#></cond>	cond	1 1 1 0 op1 0	CRn	Rd	cp_num op2 1 CRm
MLA <cond><s> Rd, Rm, Rs, Rn</s></cond>	cond	0 0 0 0 0 0 1 5		Rn	Rs 1 0 0 1 Rm
MOV <cond><s> Rd, #</s></cond>	cond	0 0 1 1 1 0 1 5		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 5		Rd Rd	shift # shift 0 Rm
MRC <cond> p<cp#>,<o1>,Rd,CRn,CRm,<o2></o2></o1></cp#></cond>	cond	0 0 0 1 1 0 1 S	SBZ CRn	Rd	Rs         0         shift         1         Rm           cp_num         op2         1         CRm
MRS <cond> Rd, CPSR</cond>	cond	0 0 0 1 0 0 0 0		Rd	SBZ
MRS <cond> Rd, SPSR</cond>	cond	0 0 0 1 0 1 0 0		Rd	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		SBO	rotate #
MSR <cond> SPSR_<fields>, Rm</fields></cond>	cond	0 0 0 1 0 1 1 0		SBO	SBZ 0 Rm
MSR <cond> SPSR_f, #</cond>	cond	0 0 1 1 0 1 1 0		SBO	rotate #
MUL <cond><s> Rd, Rm, Rs MVN<cond><s> Rd, #</s></cond></s></cond>	cond	0 0 0 0 0 0 0 0 5		SBZ Rd	Rs 1 0 0 1 Rm
MVN <cond><s> Rd, # MVN<cond><s> Rd, Rm OP #</s></cond></s></cond>	cond	0 0 0 1 1 1 1 1 5		Rd	shift # shift 0 Rm
MVN <cond><s> Rd, Rm OP Rs</s></cond>	cond	0 0 0 1 1 1 1 5		Rd	Rs 0 shift 1 Rm
ORR <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 1 1 0 0 5		Rd	rotate #
ORR <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 1 1 0 0 5		Rd	shift # shift 0 Rm
ORR <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 1 1 0 0 5		Rd	Rs 0 shift 1 Rm
RSB <cond><s> Rd, Rn, #</s></cond>	cond	0 0 1 0 0 1 1 5	Rn		
		0 0 0 0 0 1		Rd	rotate #
RSB <cond><s> Rd, Rn, Rm OP #</s></cond>	cond	0 0 0 0 0 1 1 5	Rn	Rd	shift # shift 0 Rm
RSB <cond><s> Rd, Rn, Rm OP Rs</s></cond>	cond	0 0 0 0 0 1 1 5	Rn Rn		
	cond	0 0 0 0 0 1 1 5	Rn Rn Rn	Rd	shift # shift 0 Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, #</s></cond></s></cond>	cond cond cond	0 0 0 0 0 1 1 5	Rn Rn Rn Rn	Rd Rd Rd	shift # shift 0 Rm  Rs 0 shift 1 Rm  rotate #
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP #</s></cond></s></cond></s></cond>	cond cond cond cond	0 0 0 0 0 1 1 5 0 0 1 0 1 1 1 5 0 0 0 0 1 1 1 1 5	Rn Rn Rn Rn Rn	Rd Rd Rd Rd	shift #         shift 0         Rm           Rs         0         shift 1         Rm           rotate         #           shift #         shift 0         Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond	0 0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 1 0 1 1 0 S 0 0 0 0 0 1 1 0 S	Rn Rn Rn Rn Rn Rn	Rd Rd Rd Rd Rd Rd Rd	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, #  RSC<cond><s> Rd, Rn, Rm OP #  RSC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, #  SBC<cond><s> Rd, Rn, #  SBC<cond><s> Rd, Rn, Rm OP #  SBC<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 0 1 1 5 0 0 1 0 1 1 1 5 0 0 0 0 0 1 1 1 5 0 0 0 0 0 1 1 1 5 0 0 1 0 1 1 1 0 5 0 0 0 0 1 1 0 5 0 0 0 0 1 1 0 5	Rn Rn Rn Rn Rn Rn Rn Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, #  RSC<cond><s> Rd, Rn, Rm OP #  RSC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP #  SBC<cond><s> Rd, Rn, Rm OP #  SBC<cond><s> Rd, Rn, Rm OP #  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs</s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 0 1 1 8 0 0 1 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 1 0 1 1 1 0 8 0 0 0 0 1 1 1 0 8 0 0 0 0 1 1 1 0 8	Rn Rn Rn Rn Rn Rn Rn Rn Rn Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, # SBC<cond><s> Rd, Rn, # SBC<cond><s> Rd, Rn, Rm OP # SBC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdHi, Rm, Rs SMLLL<cond><s> RdLo, RdHi, Rm, Rs</s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 0 1 1 8 0 0 1 0 1 1 1 8 0 0 0 0 0 1 1 1 8 0 0 0 0 0 1 1 1 8 0 0 0 1 0 1 1 1 0 8 0 0 0 0 0 1 1 0 8 0 0 0 0 0 1 1 0 8 0 0 0 0 0 1 1 1 1 8	Rn Rn Rn Rn Rn Rn Rn Rn Rn RdHi	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, #  RSC<cond><s> Rd, Rn, Rm OP #  RSC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, #  SBC<cond><s> Rd, Rn, Rm OP #  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs  SMULL<cond><s> Rd, Rd, Rd, Rm, Rm OP Rs  SMULL<cond><s> Rd, Rd, Rd, Rm, Rm OP Rs  STC<cond><sc rd,="" rm,="" rs="" stc<cond=""><sc rd,="" rm,="" rs="" stc<cond=""><sc rd,="" rm,="" rs="" stc<cond=""><sc rd,="" rm,="" rs="" stc<cond=""><sc rd,="" rm,="" rm<="" td=""><td>cond cond cond cond cond cond cond cond</td><td>0 0 0 0 0 0 1 1 8 0 0 1 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 0 1 8 0 0 0 0 0 1 1 0 8 0 0 0 0 0 1 1 1 0 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 1 0 8</td><td>Rn Rn Rn Rn Rn Rn Rn Rn Rn RdHi</td><td>Rd Rd R</td><td>  Shift #   Shift   0   Rm    </td></sc></sc></sc></sc></sc></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 0 1 1 8 0 0 1 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 0 1 8 0 0 0 0 0 1 1 0 8 0 0 0 0 0 1 1 1 0 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 1 0 8	Rn Rn Rn Rn Rn Rn Rn Rn Rn RdHi	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, # SBC<cond><s> Rd, Rn, # SBC<cond><s> Rd, Rn, Rm OP # SBC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdHi, Rm, Rs SMLLL<cond><s> RdLo, RdHi, Rm, Rs</s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 0 1 1 8 0 0 1 0 1 1 1 8 0 0 0 0 0 1 1 1 8 0 0 0 0 0 1 1 1 8 0 0 0 1 0 1 1 1 0 8 0 0 0 0 0 1 1 0 8 0 0 0 0 0 1 1 0 8 0 0 0 0 0 1 1 1 1 8	Rn Rn Rn Rn Rn Rn Rn Rn RdHi RdHi	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, #  RSC<cond><s> Rd, Rn, Rm OP #  RSC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP #  SBC<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s op="" rd,="" rm="" rn,="" rs="" smlal<cond=""><s op="" rd,="" rm="" rn,="" rs="" smlal="" smlal<s="" smlan="" smlan<="" td=""><td>cond cond cond cond cond cond cond cond</td><td>0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 0</td><td>Rn Rn Rn Rn Rn Rn Rn Rn RdHi RdHi Rn Rn</td><td>Rd Rd R</td><td>  Shift #   Shift   0   Rm    </td></s></s></s></s></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0 0	Rn Rn Rn Rn Rn Rn Rn Rn RdHi RdHi Rn Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP # SBC<cond><s> Rd, Rn, Rm OP # SBC<cond><s> Rd, Rn, Rm OP Rs SMLaLc<ond><s> RdLo, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs STC<cond> p<cp_num>, CRd, # STM<cond><addrmode> Rm, reg list STM<cond><addrmode> Rm, reg list</addrmode></cond></addrmode></cond></cp_num></cond></s></cond></s></cond></s></ond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 8 0 0 0 1 1 1 8 0 0 0 1 0 1	Rn RdHi RdHi Rn Rn Rn Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SCC<cond><s> Rd, Rn, Rm OP Rs SMULL<cond><s> Rd, Rn, Rm OP Rs SMULL<cond><s> RdLO, RdHi, Rm, Rs SMULL<cond><s> RdLO, RdHi, Rm, Rs STC<cond> p<cp_num>, CRd, # STM<cond><addrmode> Rm, reg list^ STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STR<cond> Rd, Rn, # STR STR STR STR COND&gt; Rd, Rn, #</cond></addrmode></cond></addrmode></cond></addrmode></cond></cp_num></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 0 0 0 1 1 1 0 0 0 1 0 1 1 1 1 5 0 0 0 0	Rn R	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, #  RSC<cond><s> Rd, Rn, Rm OP #  RSC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs  SMLLI&lt;-Cond&gt;<s> Rd, Rn, Rm OP Rs  SMLIL-Cond&gt;<s> Rd, Rn, Rm OP Rs  SMLIL-Cond&gt;<s> Rd, Rn, Rm, Rm, Rs  STC<cond><s> Rd, Rn, Rm, Rm, Rs  STC<cond><s> Rd, Rn, Rm, reg list  STM<cond><addrmode> Rm  STM<cond><addrmode> Rm  STM  STM<cond><addrmode> Rm  STM  STR<cond> Rd, Rn, #  STR  STR  STR  Rd, Rn, #  STR  STR  Rd, Rn, #  STR</cond></addrmode></cond></addrmode></cond></addrmode></cond></s></cond></s></cond></s></s></s></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 8 0 0 1 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 1 1 8 0 0 0 0 0 1 1 1 0 8 0 0 0 0 0 1 1 0 8 0 0 0 0 0 1 1 0 8 0 0 0 0 0 1 1 0 8 1 1 0 0 0 0 0 1 1 0 8 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0	Rn RdHi RdHi Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL Rd, Rn, Rm OP Rs SMLAL RdLo, RdHi, Rm, Rs SMLL RdLo, RdHi, Rm, Rs SMLL<ss> RdLo, RdHi, Rm, Rs SMCCONd&gt;<s> RdLo, RdHi, Rm, Rs STC<cond> Pccp num&gt;, CRd, # STM<stmccond><saddrmode> Rm, reg list^ STM<stmcond><addrmode> Rm, reg list STM<stmcond><addrmode> Rm, reg list STR<stmccond><saddrmode> Rm, reg list STR<str<stmcond><saddrmode> Rm, reg list STR<str<stmcond><saddrmode> Rm<stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond><stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></stmcond></saddrmode></str<stmcond></saddrmode></str<stmcond></saddrmode></stmccond></addrmode></stmcond></addrmode></stmcond></saddrmode></stmccond></cond></s></ss></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 8 0 0 0 1 1 1 8 0 0 0 1 0 1	Rn R	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, #  RSC<cond><s> Rd, Rn, Rm OP #  RSC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP #  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> RdLO, RdHi, Rm, Rs  STC<cond><s rdhi,="" rdlo,="" rm,="" rs="" stc<cond=""><s is="" rd.c,="" rm="" rm,="" stm<cond=""><s is="" rm="" stm<cond=""><s is="" rd="" str<cond=""> Rd, Rn, #  STR<cond> Rd, Rn, #</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></s></s></s></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 8 0 0 0 1 1 1 8 0 0 0 1 0 1	Rn RdHi RdHi RdHi Rn	Rd R	Shift #   Shift 0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, #  RSC<cond><s> Rd, Rn, Rm OP #  RSC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s rd,="" rm="" rm,="" s="" smlil<cond=""><s> Rd, Rn, Rm OP Rs  SMLIL<cond><s> Rd, Rn, Rm OP Rs  SMLIL<cond><s> Rd, Rn, Rm, Rm, Rs  STC<cond><s list^="" rd,="" reg="" rm,="" rn,="" stm<cond=""><s list^="" rd,="" reg="" rn,="" stm<cond=""><addrmode> Rm, reg list^  STM<cond><addrmode> Rm, reg list^  STR<cond> Rd, Rn, #  STR<cond> Rd, Rn, #  STR<cond> Rd, Rn, #  STR<cond> Bd, Rn, #  STR<cond> BT Rd, Rn, #</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></s></s></cond></s></cond></s></cond></s></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Rn R	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, #  RSC<cond><s> Rd, Rn, Rm OP #  RSC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP #  SBC<cond><s> Rd, Rn, Rm OP Rs  SBC<cond><s> Rd, Rn, Rm OP Rs  SMLAL<cond><s> RdLO, RdHi, Rm, Rs  STC<cond><s rdhi,="" rdlo,="" rm,="" rs="" stc<cond=""><s is="" rd.c,="" rm="" rm,="" stm<cond=""><s is="" rm="" stm<cond=""><s is="" rd="" str<cond=""> Rd, Rn, #  STR<cond> Rd, Rn, #</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></s></s></s></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 8 0 0 0 1 1 1 8 0 0 0 1 0 1	Rn R	Rd R	Shift #   Shift 0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs STC<cond> pccp num&gt;, CRd, # STM<cond><addrmode> Rm, reg list^ STM<cond><addrmode> Rm<!-- -->, reg list^ STM<cond><addrmode> Rm<!-- -->, reg list^ STM<cond><addrmode> Rm<!-- -->, reg list^ STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Bd, Rn, # STR<cond> Rd, Rn, # STR<cond> Bd, 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></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 1 1 0 S 1 1 0 0 0 0 1 1 0 S 1 1 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Rn RdHi RdHi Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLO, RdHi, Rm, Rs STC<cond><s> RdLO, RdHi, Rm, Rs STC<cond><s #="" crd,="" pcd,="" pumb,="" stm<cond=""><cond><cond>Rm</cond> Rm reg list STM<cond><cond><cond>Rm</cond> Rm reg list STM<cond><cond><cond>Rm</cond> Rm   STM<cond><cond><cond>Rm Rm &gt; rg list   STM<cond><cond> Rd, Rn, #   STR<cond> Rd, Rn, #   STR<cond> Bd, Rn, #   STR STR Rd, Rn, #   STR STR Rd Rn, #</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 0 0 0 0 0 1 1 0 S 1 1 0 0 P U 1 W 0 1 0 0 P U 1 W 0 0 1 1 P U 0 W 0 0 1 1 P U 1 W 0 0 1 1 P U 1 W 0 0 1 1 P U 1 W 0 0 1 1 P U 1 W 0 0 1 1 P U 1 W 0 0 1 1 0 P U 1 W 0 0 1 1 0 P U 1 W 0 0 1 1 0 P U 1 W 0 0 1 1 0 P U 1 W 0 0 1 0 0 P U 1 W 0 0 1 0 0 U 1 0 0 0 1 0 0 U 1 0 0 0 1 0 0 U 1 0 0	Rn R	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLO, RdHi, Rm, Rs STC<cond> pcp num&gt;, CRd, # STM<cond><addrmode> Rm reg list^ STM<cond><addrmode> Rm reg list^ STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond> Rd, Rn, # STR<cond> 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></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 8 0 0 0 1 1 1 8 0 0 0 1 0 1	Rn RdHi RdHi Rn	Rd RdLo CRd O  I  Rd	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdH1, Rm, Rs SMLAL<cond><s> RdLo, RdH1, Rm, Rs SMLAL<cond><s> RdLo, RdH1, Rm, Rs STC<cond> P&lt; CP NUMP. CRd, # STM<cond addrmode="" crd,="" list="" re="" reg="" rn,="" stm<cond=""> Rm Rm P RdLo, RdH2, reg list STM<cond #="" rdlo,="" rn,="" str<cond=""> Rd, Rn, # STR<cond> Rd, Rn, # SUB<cond> SRd, Rn, Rn OP # SUB<cond> SRd, Rn, Rm OP Rs</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 P U N W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 1 0 P U 1 W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 0 P U 1 W 1 0 P U 1 W	Rn RdHi RdHi Rn	Rd RdLo RdLo	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs STC<cond> pccp num&gt;, CRd, # STM<cond><addrmode> Rm. reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Bd, Rn, # STR<cond> Bd, Rn, # STR<cond> Bd, Rn, # STR<cond> TRd, Rn, # SUB<cond> SRd, Rn, # SUB<cond> SRd, Rn, Rm OP # SUB&lt; Cond&gt; SRd, Rn, Rm OP Rs SWI <swi_number></swi_number></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></cond></addrmode></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 1 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 0 1 1 0 S 0 1 0 0 0 0 0 1 1 0 S 0 0 0 0 0 0 1 1 0 S 0 0 0 0 0 0 1 1 0 S 0 0 0 0 0 0 1 1 0 S 1 1 1 1 1	Rn R	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL Rd, Rn, Rm OP Rs SMLAL Rd, Rn, Rm OP Rs SMLAL RdLo, RdHi, Rm, Rs SMULL SMCCONd&gt;<s> RdLo, RdHi, Rm, Rs SMULL STC<ss s<="" ss="" st="" stc<="" stc<ss="" td=""><td>cond cond cond cond cond cond cond cond</td><td>0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 1 0 0 0 0 1 1 C 0 1 0 0 0 0 0 1 C 0 0 0 0 0 0 1 0 S 0 0 0 0 0 0 1 0 S 0 0 0 0 0 0 1 0 S</td><td>Rn Rn Rn Rn Rn Rn Rn Rn Rn RdHi RdHi Rn Rn</td><td>Rd Rd R</td><td>  Shift #   Shift   0   Rm    </td></ss></s></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 1 0 0 0 0 1 1 C 0 1 0 0 0 0 0 1 C 0 0 0 0 0 0 1 0 S 0 0 0 0 0 0 1 0 S 0 0 0 0 0 0 1 0 S	Rn Rn Rn Rn Rn Rn Rn Rn Rn RdHi RdHi Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdH1, Rm, Rs SMLAL<cond><s> RdLo, RdH1, Rm, Rs SMLAL<cond><s> RdLo, RdH1, Rm, Rs STC<cond> P&lt; CP NUMP.  STM<cond addrod="" rdh1,="" rdlop="" rm,="" rmc!="" rs="" stm<cond="">, reg list STM<cond addrmode=""> Rmc!&gt;, reg list STM<cond addrmode=""> Rmc!&gt;, reg list STM<cond rdlop="" rmc!="">, reg list STR<cond> Rd, Rn, # STR SRd, Rn, RN OP RS SWB<swd< swd<="" swd<<="" td=""><td>cond cond cond cond cond cond cond cond</td><td>0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 S 1 1 1 0 0 0 0 1 S 0 1 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 0 1 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S</td><td>Rn Rn RdHi RdHi RdHi Rn Rn</td><td>Rd Rd R</td><td>  Shift #   Shift   0   Rm    </td></swd<></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 S 1 1 1 0 0 0 0 1 S 0 1 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 0 1 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S	Rn RdHi RdHi RdHi Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL Rd, Rn, Rm OP Rs SMLAL Rd, Rn, Rm OP Rs SMLAL RdLo, RdHi, Rm, Rs SMULL SMCCONd&gt;<s> RdLo, RdHi, Rm, Rs SMULL STC<ss s<="" ss="" st="" stc<="" stc<ss="" td=""><td>cond cond cond cond cond cond cond cond</td><td>0 0 0 0 0 1 1 S 0 0 0 1 1 S 0 0 0 0 0 1 1 S 0 0 0 0</td><td>Rn Rn R</td><td>Rd Rd R</td><td>  Shift #   Shift   0   Rm    </td></ss></s></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 0 1 1 S 0 0 0 0 0 1 1 S 0 0 0 0	Rn R	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs STC<cond> pccp num&gt;, CRd, # STM<cond><addrmode> Rm. reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STR<cond> Rd, Rn, # SUB<cond><s> Rd, Rn, # SUB<cond><s> Rd, Rn, Rn OP Rs SWI <swi_number> SWP<cond> Rd, Rm, [Rn] SWP<cond> Rd, Rm, [Rn] SWP<cond> Rd, Rm, [Rn] SWP<cond> Rd, Rm, [Rn] SWP<cond> Rd, Rm, [Rn]</cond></cond></cond></cond></cond></swi_number></s></cond></s></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 0 1 1 S 0 0 0 0 0 1 1 S 0 0 0 0	Rn RdHi RdHi Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL SMLAL SMLAL SMLAL CRD, RdHi, Rm, Rs SMLL SMLAL CRD, RdHi, Rm, Rs SMLL STC<ss s<="" ss="" st="" stc<="" stc<ss="" td=""><td>cond cond cond cond cond cond cond cond</td><td>0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 1 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 0 1 1 S 0 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 1 0 1 0 S 0 0 0 1 1 0 0 S 0 0 0 1 1 0 0 S 0 0 0 1 1 0 0 S 0 0 0 1 1 0 0 S 0 0 0 1 1 0 0 S</td><td>Rn Rn Rn Rn Rn Rn Rn Rn Rn Rn RdHi RdHi Rn Rn</td><td>Rd Rd R</td><td>  Shift #   Shift   0   Rm    </td></ss></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 1 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 0 1 1 S 0 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 1 0 1 0 S 0 0 0 1 1 0 0 S 0 0 0 1 1 0 0 S 0 0 0 1 1 0 0 S 0 0 0 1 1 0 0 S 0 0 0 1 1 0 0 S	Rn RdHi RdHi Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL Rd, Rn, Rm OP Rs SMLAL RdLo, RdHi, Rm, Rs SMLL RdLo, RdHi, Rm, Rs SMLL RGLO, RdHi, Rm, Rs SMCCOND&gt;<s> RdLo, RdHi, Rm, Rs SMCCOND&gt;<sp rdhi,="" rdlo,="" rm,="" rs="" stc<cond=""> FCP num&gt;, CRd, # STM<cond><addrmode> Rm, reg list STM<cond><addrmode> Rm, reg list STM<cond><addrmode> Rm, reg list STM<cond><addrmode> Rm, reg list STR<cond> Rd, Rn, # SUB<cond<s> Rd, Rn, Rm OP # SUB<cond<s> Rd, Rn, Rm OP RS SWI <swi_number> SWP<cond> Rd, Rn, [Rn] TEQ<cond> Rn, Rm OP R TEXT<cond> Rn, Rm OP R TEXT<cond> Rn, Rm OP R TEXT</cond></cond></cond></cond></cond></cond></cond></cond></swi_number></cond<s></cond<s></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></sp></s></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 1 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 1 1 0 S 1 1 0 0 0 0 1 1 0 S 1 1 0 0 0 0 1 1 0 S 1 1 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 0 1 1 0 S 0 0 0 0 0 0 1 S 0 0 0 0 0 1 0 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 0 1 0 1 S 0 0 0 1 0 0 1 S 0 0 0 1 0 0 1 S 0 0 0 1 0 0 1 S 0 0 0 1 0 0 1 S	Rn Rn Rn Rn Rn Rn Rn Rn Rn RdHi RdHi Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP RS SBC<cond><s> Rd, Rn, Rm OP RS SMLAL RdLO, RdH1, Rm, Rs SMLAL RdLO, RdH1, Rm, Rs SMLAL RdLO, RdH1, Rm, Rs STC<cond><s> RdLO, RdH1, Rm, Rs STC<cond><s> RdLO, RdH1, Rm, Rs STC<cond><s rdh1,="" rdlo,="" rm,="" rs="" stm<stm<scond=""><saddrmode> Rm. reg list STM<stm<scond><saddrmode> Rm.'s, reg list STM<stm<scond><saddrmode> Rm.'s, reg list STM<stm<scond><saddrmode> Rm.'s, reg list STR<cond> Rd, Rn, # STR<stcond> Rd, Rn, # STR<sond> Rd, Rn, RN OP RS SWI <swi number=""> SWP<cond> Rd, Rm, [Rn] SWP<cond> Rd, Rm, [Rn] SWP<cond> Rd, Rm, [Rn] STC<cond> Rd, Rm, [Rn] STC<cond> Rd, Rm, Rm OP RS STST<cond> Rn, Rm OP RS</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></swi></sond></stcond></stcond></stcond></stcond></stcond></stcond></stcond></stcond></stcond></cond></saddrmode></stm<scond></saddrmode></stm<scond></saddrmode></stm<scond></saddrmode></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 P U N W 0 1 0 0 P U 1 W 0 1 0 0 P U 1 W 0 1 0 0 P U 1 W 0 1 1 0 P U 1 W 0 1 1 0 P U 1 W 0 1 1 0 P U 1 W 0 1 1 1 P U 1 W 0 1 1 1 P U 1 W 0 1 1 1 P U 1 W 0 1 P U 1 W 0 1 P U 1	Rn RdHi RdHi RdHi Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdHi, Rm, Rs STC<cond> pcp num&gt;, CRd, # STM<cond><addrmode> Rd, reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STR<cond> Rd, Rn, # STR<cond> Rd, Rn, Rn STR STR<cond> Rd, Rn, Rn STR STR<cond> Rd, Rn, Rn STR STR SWC<cond> SR, Rn, Rn SWC SWC<cond> Rd, Rn, Rn SWC Rm, Rn OP RS TST STC STC STC STC STC STC STC SWC SWC</cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 1 0 S 1 1 1 1 1 0 0 1 0 S 1 1 1 1 1 0 0 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0	Rn R	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL Rd, Rn, Rm OP Rs SMLAL Rd, Rn, Rm OP Rs SMLAL RdLo, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs STC<cond> Pccp num&gt;, CRd, # STM<cond><addrmode> Rm. reg list^ STM<cond><addrmode> Rm<!-- -->, reg list^ STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Bd, Rn, # STR<cond> Bd, Rn, # STR<cond> Bd, Rn, # STR<cond> Td, Rn, # STR<cond> Td, Rn, # STR<cond> Td, Rn, # STR<cond> Td, Rn, # SUB<cond><s> Rd, Rn, Rm OP Rs SWI <awin immber=""> SWP<cond> Rd, Rn, [Rn] SWP<cond> Rd, Rn, [Rn] TEQ<cond> Rn, Rm OP Rs TST<cond> Rn, Rm OP Rs TST<cond< op="" pr="" rm="" rn,="" rs="" umlal<=""> UMLAL</cond<></cond></cond></cond></cond></cond></cond></cond></awin></s></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 0 1 0 1 1 1 S 0 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 0 0 0 0 0 1 1 0 S 1 1 0 0 P U 1 W C 0 1 0 P U 1 W C 0 1 0 P U 1 W C 0 1 0 P U 1 W C 0 1 0 P U 1 W C 0 1 0 P U 1 W C 0 1 0 P U 1 W C 0 1 0 P U 1 W C 0 1 0 P U 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 0 S 0 0 0 0 1 0 S 0 0 0 0 1 0 S 0 0 0 0 1 0 S 0 0 0 0 1 0 S 0 0 0 0 1 0 0 S 0 0 0 0 1 0 0 S 0 0 0 1 0 0 1 S 0 0 0 1 0 0 1 S 0 0 0 1 0 0 1 0 S 0 0 0 0 1 0 0 S 0 0 0 1 0 0 1 0 S 0 0 0 1 0 0 1 0 S 0 0 0 1 0 0 1 0 S	Rn R	Rd R	Shift #   Shift   0   Rm   Rs   0   Shift   1   Rm   rotate   #   Shift   0   Rm   Rs   0   Shift   1   Rm   Rs   0   Shift   1   Rm   Rs   0   Shift   1   Rm   Rs   1   0   0   1   Rm   Rs   1   0   0   1   Rm   Rs   1   0   0   1   Rm   Rs   1   1   1   1   1   1   1   1   1
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL SMLAL SMLAR SMLAR SMLAR SMLAR SMLAR RGLO, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs SMCC STC<cond> Pccp num&gt;, CRd, # STM<cond><sr rdhi,="" rdlo,="" rm,="" rs="" stc<cond=""> Rd, Rn, # STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STR<cond> Rd, Rn, # STR<cond> Rd, Rn, Rn STR<td>cond cond cond cond cond cond cond cond</td><td>0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 P U N W 0 1 0 0 P U 1 W 0 1 0 0 P U 1 W 0 1 0 0 P U 1 W 0 1 1 P U 1 W 0 1 P U 1 P U</td><td>Rn Rn Rn Rn Rn Rn Rn Rn Rn RdHi RdHi RdHi Rn Rn</td><td>Rd Rd R</td><td>  Shift #   Shift   0   Rm    </td></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></cond></addrmode></cond></sr></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 P U N W 0 1 0 0 P U 1 W 0 1 0 0 P U 1 W 0 1 0 0 P U 1 W 0 1 1 P U 1 W 0 1 P U 1 P U	Rn Rn Rn Rn Rn Rn Rn Rn Rn RdHi RdHi RdHi Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdHi, Rm, Rs SMLAL<cond><s> RdLo, RdHi, Rm, Rs SMLL<cond><s> RdLo, RdHi, Rm, Rs STC<cond> pccp num&gt;, CRd, # STM<cond><addrmode> Rm. reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Rd, Rn, # STR<cond> Bd, Rn, # STR<cond> Bd, Rn, # STR<cond> Bd, Rn, # STR<cond> Rd, Rn, Rn P STR<cond> Rd, Rn, Rn P STR<cond> Rd, Rn, Rn, P STR<cond> Rd, Rn, Rn P STR<cond> Rd, Rn, Rn P STR SWP<cond> Rd, Rn, Rn P SWP Cond&gt; Rd, Rn, Rn P STC<and> Rd, Rn, Rn) TEQ<cond> Rd, Rn, Rn) TEQ<cond> Rd, Rn, Rn) TEQ<cond> Rd, Rn, Rn P STT STT&lt;<and> RdHi, Rn, Rn STT UMLAL&lt;<and> RdHi, Rn, Rs UMULL UMGAIL UMGAIC UMGAIL Undefined Instruction Undefined Instruction</and></and></cond></cond></cond></and></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 0 0 0 0 0 1 0 S 1 1 1 0 0 0 1 0 S 1 1 1 1 1 0 0 1 0 S 1 1 1 1 1 0 0 1 0 S 1 1 1 1 1 0 0 1 0 S 1 1 1 1 1 1 0 0 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 S 1 1 1 1 1 1 0 0 1 1 0 0 1 0 S 1 1 1 1 1 1 0 0 1 1 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0	Rn R	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP # RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL SMLAL SMLAR SMLAR SMLAR SMLAR SMLAR RGLO, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs SMULL<cond><s> RdLo, RdHi, Rm, Rs SMCC STC<cond> Pccp num&gt;, CRd, # STM<cond><sr rdhi,="" rdlo,="" rm,="" rs="" stc<cond=""> Rd, Rn, # STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STR<cond> Rd, Rn, # STR<cond> Rd, Rn, Rn STR<td>cond cond cond cond cond cond cond cond</td><td>0 0 0 0 0 1 1 S 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 0 P U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 0 0 D U 1 W C 0 0 D U 1 W C 0 0 D U 1 W C 0 0 D U 1 W C 0 D U 1 W</td><td>Rn Rn Rn Rn Rn Rn Rn Rn Rn Rn RdHi RdHi Rn Rn</td><td>Rd Rd R</td><td>  Shift #   Shift   0   Rm    </td></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></cond></addrmode></cond></sr></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 0 P U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 1 0 D U 1 W C 0 0 0 D U 1 W C 0 0 D U 1 W C 0 0 D U 1 W C 0 0 D U 1 W C 0 D U 1 W	Rn RdHi RdHi Rn	Rd R	Shift #   Shift   0   Rm
RSB <cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, # RSC<cond><s> Rd, Rn, Rm OP Rs RSC<cond><s> Rd, Rn, Rm OP Rs SBC<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> Rd, Rn, Rm OP Rs SMLAL<cond><s> RdLo, RdHi, Rm, Rs SMLL<cond><s> RdLo, RdHi, Rm, Rs STC<cond> pccp num&gt;, CRd, # STM<cond><addrmode> Rm. reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STM<cond><addrmode> Rm<!-- -->, reg list STR<cond> Rd, Rn, # STR<cond> Rd, Rn, Rn STR SWD<cond><s> Rd, Rn, Rm OP Rs SWI <swi number=""> SWP<cond> Rd, Rn, [Rn] SWP<cond> Rd, Rn, [Rn] TEQ<cond> Rn, Rm OP Rs STT<cond> Rn, Rm OP Rs TST<cond> Rn, Rm OP Rs UMLAL<cond><s> RdLo, RdHi, Rm, Rs UMcdfined Instruction Undefined Instruction Undefined Instruction Undefined Instruction Undefined Instruction Undefined Instruction</s></cond></cond></cond></cond></cond></cond></cond></cond></cond></swi></s></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></cond></addrmode></cond></addrmode></cond></addrmode></cond></addrmode></cond></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond></s></cond>	cond cond cond cond cond cond cond cond	0 0 0 0 0 1 1 S 0 0 1 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 1 S 0 0 0 0 0 1 1 0 S 0 0 0 0 0 1 1 0 S 1 1 0 P U N W 0 1 0 0 P U 1 W 0 1 0 0 P U 1 W 0 1 0 0 P U 1 W 0 1 0 0 P U 1 W 0 1 1 0 P U 1 W 0 1 1 0 P U 1 W 0 1 1 0 P U 1 W 0 1 1 0 P U 1 W 0 1 1 0 P U 1 W 0 1 1 1 D U 1 W 0 1 1 1 D U 1 W 0 1 1 1 D U 1 W 0 1 1 D U 1 W 0 1 1 D U 1 W 0 1 1 D U 1 W 0 1 1 D U 1 W 0 1 1 D U 1 W 0 1 1 D U 1 W 0 1 1 D U 1 W 0 1 1 D U 1 W 0 1 1 D U 1 W 0 1 1 D U 1 W 0 1 1 D U 1 W 0 1 D U 1 D U 1 W 0 1 D U 1 D U 1 W 0 1 D U 1 D U 1 D U 1 D U 1	Rn R	Rd R	Shift #   Shift   0   Rm   Rm   rotate   #   Shift   1   Rm   Rm   rotate   #   Shift   1   Rm   Rm   Rm   Rm   Rm   Rm   Rm