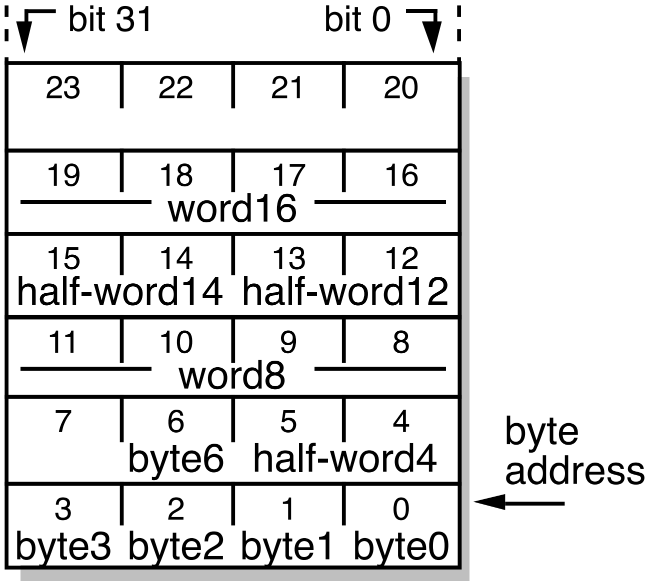
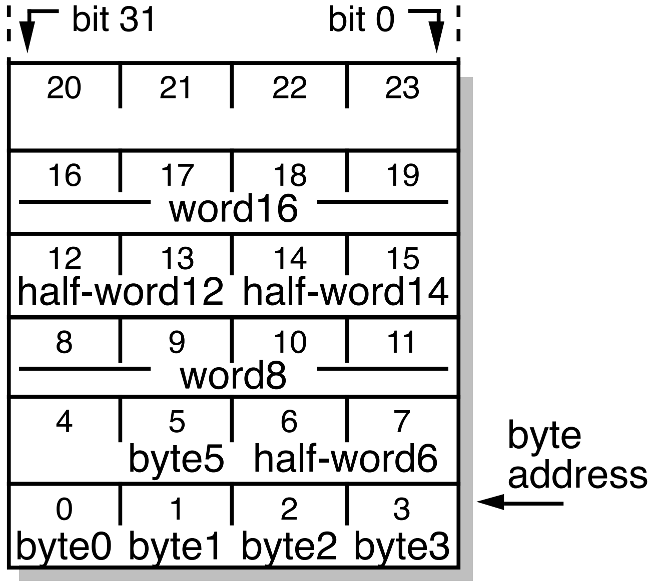
Little- and big-endian memory organizations



(a) Little-endian memory organization



(b) Big-endian memory organization

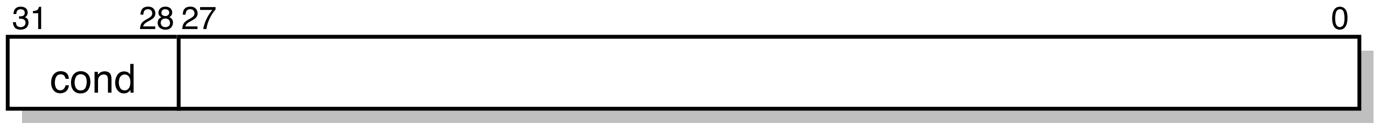
ARM operating modes and register usage.

CPSR[4:0]	Mode	Use	Registers
10000	User	Normal user code	user
10001	FIQ	Processing fast interrupts	_fiq
10010	IRQ	Processing standard interrupts	_irq
10011	SVC	Processing software interrupts (SWIs)	_svc
10111	Abort	Processing memory faults	_abt
11011	Undef	Handling undefined instruction traps	_und
11111	System	Running privileged operating system tasks	user

Exception vector addresses

Exception	Mode	Vector address
Reset	SVC	0x00000000
Undefined instruction	UND	0x00000004
Software interrupt (SWI)	SVC	0x00000008
Prefetch abort (instruction fetch memory fault)	Abort	0x0000000C
Data abort (data access memory fault)	Abort	0x00000010
IRQ (normal interrupt)	IRQ	0x00000018
FIQ (fast interrupt)	FIQ	0x0000001C

The ARM condition code field



ARM condition codes

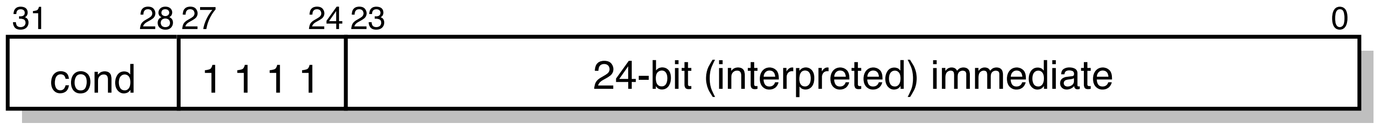
Opcode [31:28]	Mnemonic extension	Interpretation	Status flag state for execution
0000	EQ	Equal / equals zero	Z set
0001	NE	Not equal	Z clear
0010	CS/HS	Carry set / unsigned higher or same	C set
0011	CC/LO	Carry clear / unsigned lower	C clear
0100	MI	Minus / negative	N set
0101	PL	Plus / positive or zero	N clear
0110	VS	Overflow	V set
0111	VC	No overflow	V clear
1000	HI	Unsigned higher	C set and Z clear
1001	LS	Unsigned lower or same	C clear or Z set
1010	GE	Signed greater than or equal	N equals V
1011	LT	Signed less than	N is not equal to V
1100	GT	Signed greater than	Z clear and N equals V
1101	LE	Signed less than or equal	Z set or N is not equal to V
1110	AL	Always	any
1111	NV	Never (do not use!)	none

Branch and Branch with Link binary encoding

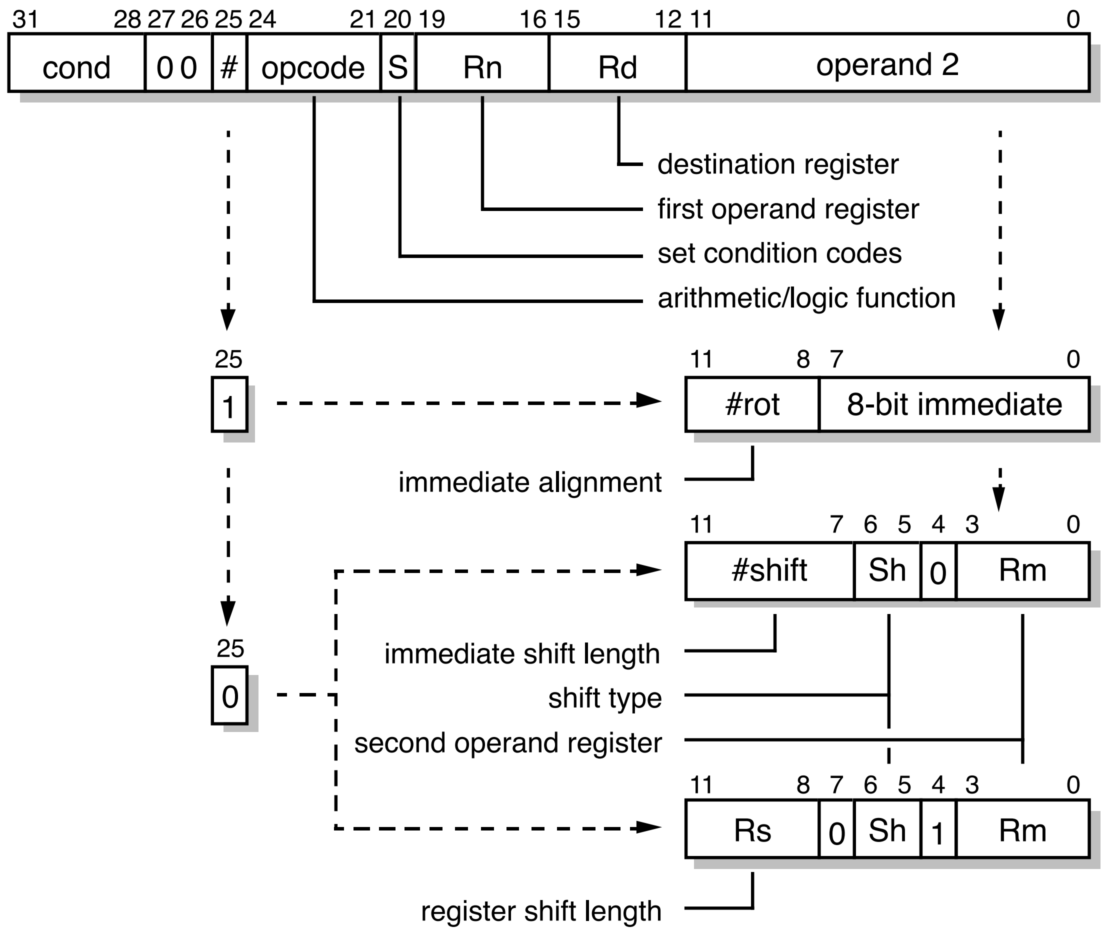
Branch and exchange instruction binary encoding

31 28	27 4	3 0
cond	0 0 0 1 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1	Rm

Software interrupt binary encoding



Data processing instruction binary encoding



ARM data processing instructions

Opcode [24:21]	Mnemonic	Meaning	Effect
0000	AND	Logical bit-wise AND	Rd := Rn AND Op2
0001	EOR	Logical bit-wise exclusive OR	Rd := Rn EOR Op2
0010	SUB	Subtract	Rd := Rn - Op2
0011	RSB	Reverse subtract	Rd := Op2 - Rn
0100	ADD	Add	Rd := Rn + Op2
0101	ADC	Add with carry	Rd := Rn + Op2 + C
0110	SBC	Subtract with carry	Rd := Rn - Op2 + C - 1
0111	RSC	Reverse subtract with carry	Rd := Op2 - Rn + C - 1
1000	TST	Test	Scc on Rn AND Op2
1001	TEQ	Test equivalence	Scc on Rn EOR Op2
1010	CMP	Compare	Scc on Rn - Op2
1011	CMN	Compare negated	Scc on Rn + Op2
1100	ORR	Logical bit-wise OR	Rd := Rn OR Op2
1101	MOV	Move	Rd := Op2
1110	BIC	Bit clear	Rd := Rn AND NOT Op2
1111	MVN	Move negated	Rd := NOT Op2

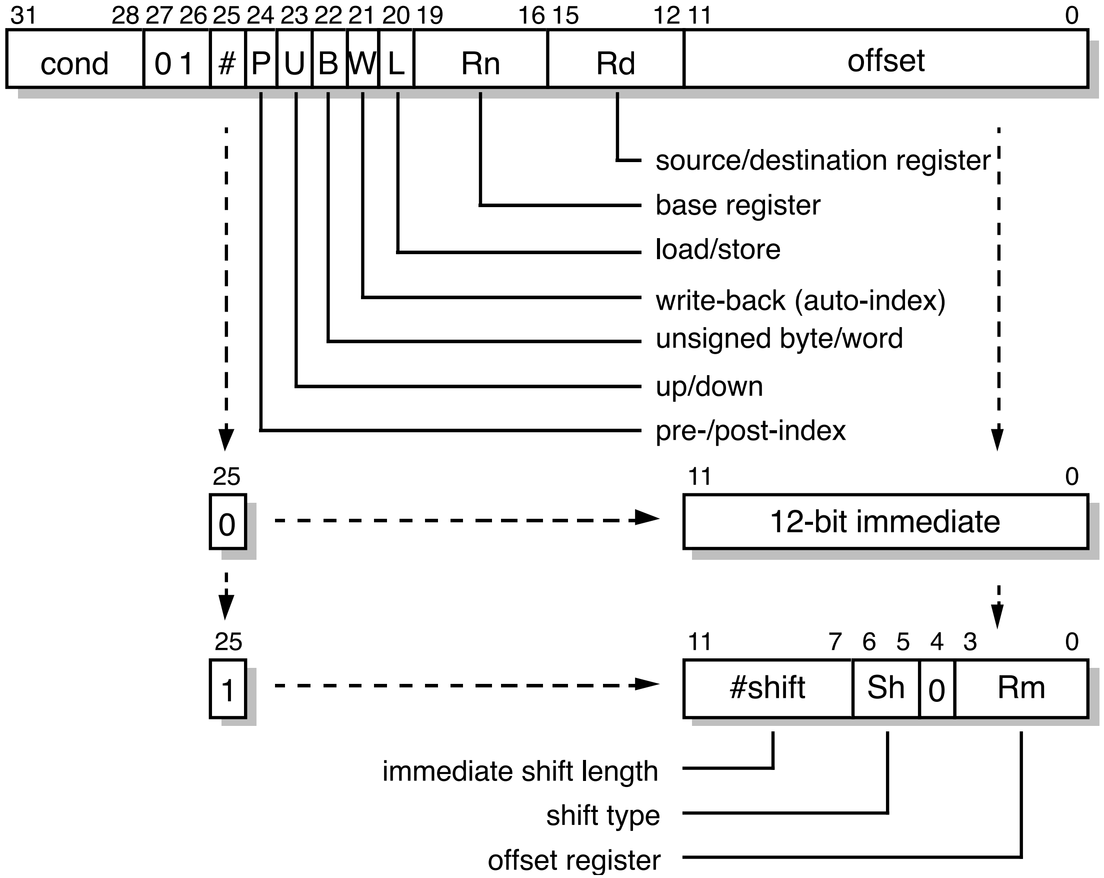
Multiply instruction binary encoding

31 28	27 24	23 21	20	19 16	15 12	11 8	7 4	3 0
cond	0 0 0 0	mul	S	Rd/RdHi	Rn/RdLo	Rs	1 0 0 1	Rm

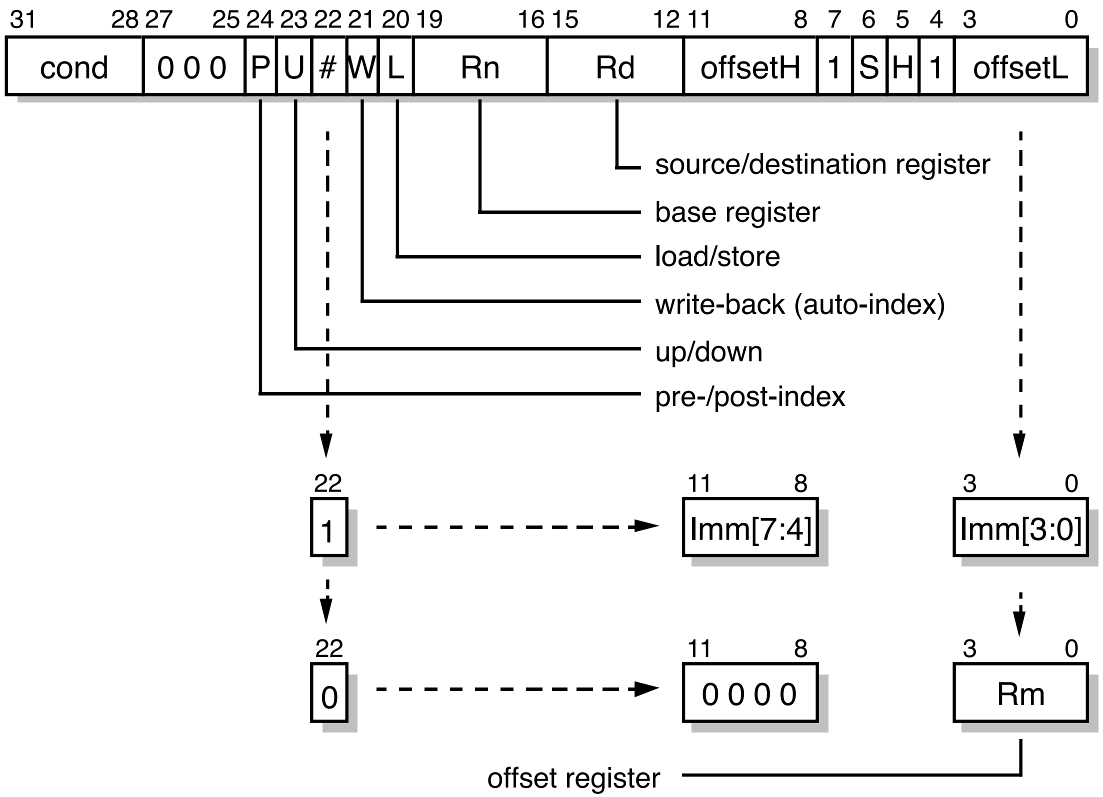
Multiply instructions

Opcode [23:21]	**Mnemonic**	**Meaning**	**Effect**
000	MUL	Multiply (32-bit result)	Rd := (Rm * Rs) [31:0]
001	MLA	Multiply-accumulate (32-bit result)	Rd := (Rm * Rs + Rn) [31:0]
100	UMULL	Unsigned multiply long	RdHi:RdLo := Rm * Rs
101	UMLAL	Unsigned multiply-accumulate long	RdHi:RdLo += Rm * Rs
110	SMULL	Signed multiply long	RdHi:RdLo := Rm * Rs
111	SMLAL	Signed multiply-accumulate long	RdHi:RdLo += Rm * Rs

Single word and unsigned byte data transfer instruction binary encoding



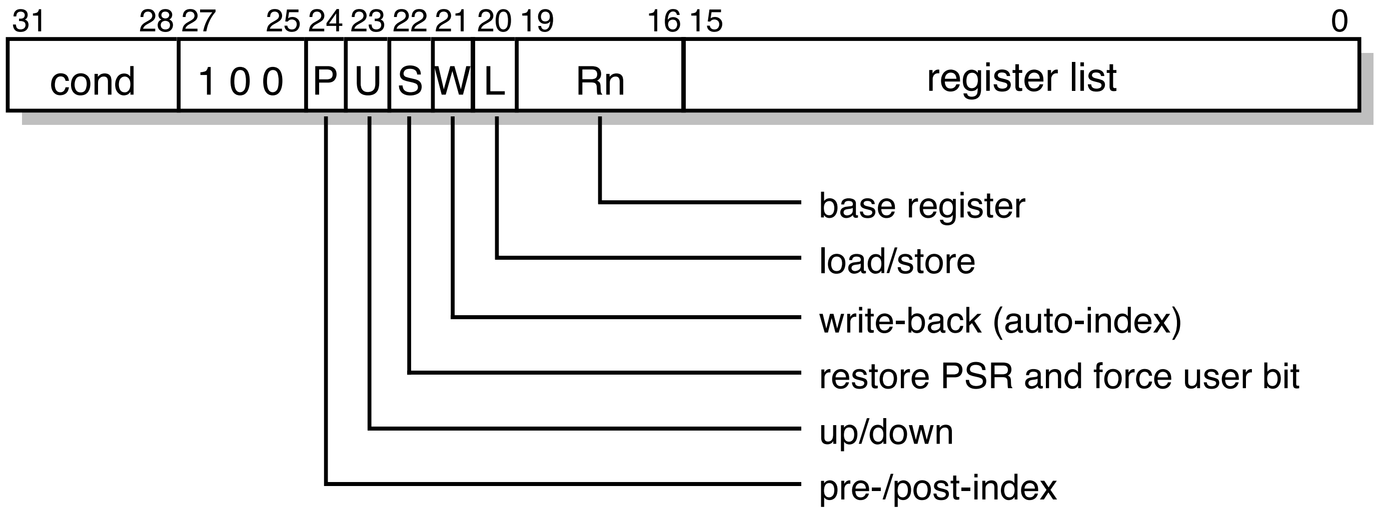
Half-word and signed byte data transfer instruction binary encoding



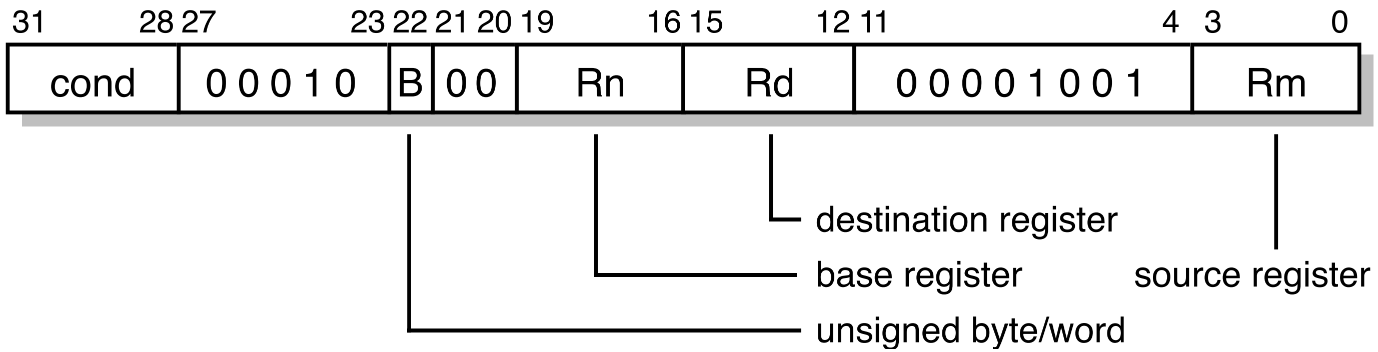
Data type encoding

S	H	Data type
1	0	Signed byte
0	1	Unsigned half-word
1	1	Signed half-word

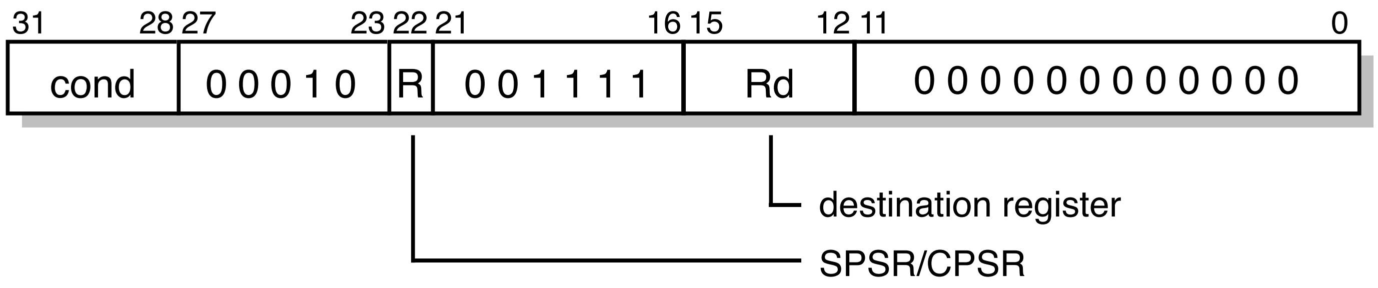
Multiple register data transfer instruction binary encoding



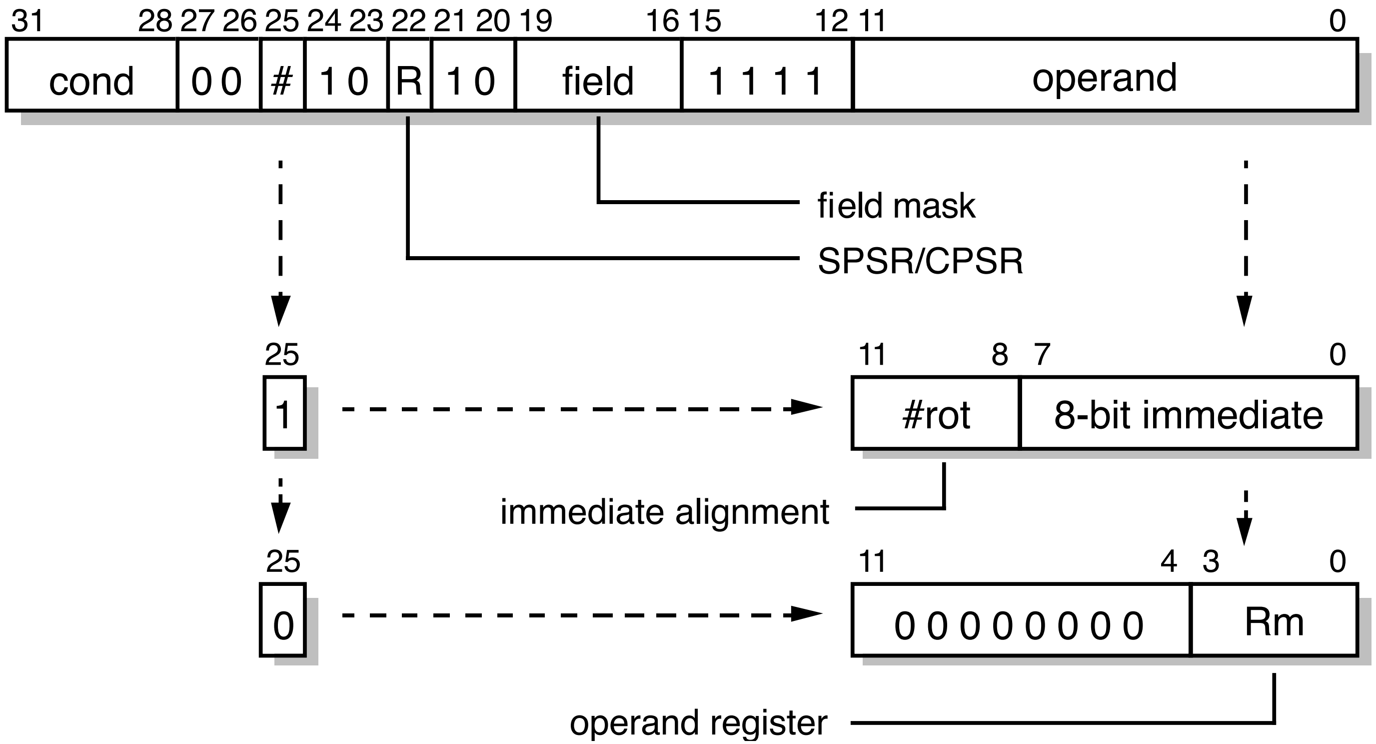
Swap memory and register instruction binary encoding



Status register to general register transfer instruction binary encoding



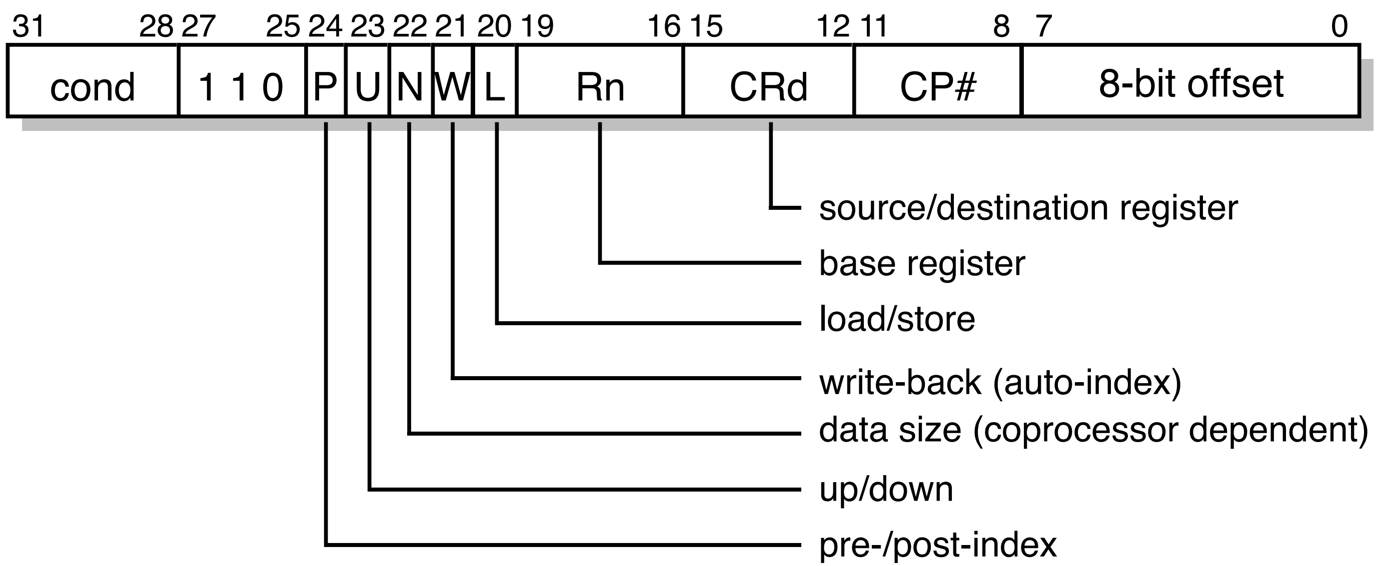
Transfer to status register instruction binary encoding



Coprocessor data processing instruction binary encoding

31 28	27 24	23 20	19 16	15 12	11 8	7 5	4	3 0
cond	1 1 1 0	Cop1	CRn	CRd	CP#	Cop2	0	CRm

Coprocessor data transfer instruction binary encoding



Coprocessor register transfer instruction binary encoding

31–28	27–24	23–21	20	19–16	15–12	11–8	7–5	4	3–0
cond	1 1 1 0	Cop1	L	CRn	Rd	CP#	Cop2	1	CRm

L: load from coprocessor/store to coprocessor

Arithmetic instruction extension space

31–28	27–22	21 20	19–16	15–12	11–8	7–4	3–0
cond	0 0 0 0 0 1	op	Rn	Rd	Rs	1 0 0 1	Rm

Control instruction extension space

31 28	27 23	22 21	20	19 16	15 12	11 8	7	6	4	3 0
cond	0 0 0 1 0	op1	0	Rn	Rd	Rs	op2		0	Rm
cond	0 0 0 1 0	op1	0	Rn	Rd	Rs	0	op2	1	Rm
cond	0 0 1 1 0	op1	0	Rn	Rd	#rot	8-bit immediate			

Data transfer instruction extension space

31 28	27 25	24	23	22	21	20	19 16	15 12	11 8	7	6 5	4	3 0
cond	0 0 0	P	U	B	W	L	Rn	Rd	Rs	1	op1	1	Rm

Coprocessor instruction extension space

31 28	27 25 24	23 22	21	20	19 16	15 12	11 8	7 0
cond	1 1 0 0	op	0	X	Rn	CRd	CP#	offset

Undefined instruction space

31 28	27 25	24 5	4	3 0
cond	0 1 1	X X X X X X X X X X X X X X X X X X X X	1	X X X X

The 26-bit architecture format of register 15