ARM Instruction Layouts, Conditions and Op-codes

Note: This handout describes only those ARM instruction layouts covered by the G51CSA module. For a full and definitive list other sources of information must be consulted.

The ARM condition 'op-codes'

Condition Opcode [31:28]	*Mnemonic extension*	*Interpretation*	*CCs for execution*
0000	EQ	Equal / Equals zero	Z
0001	NE	Not equal	!Z
0010	CS/HS	Carry set / Higher or same (unsigned)	C
0011	CC/LO	Carry clear / Lower (unsigned)	!C
0100	MI	Minus / negative	N
0101	PL	Plus / positive or zero	!N
0110	VS	Overflow set	V
0111	VC	Overflow clear	!V
1000	HI	Unsigned higher	C . !Z
1001	LS	Unsigned lower or same	!C + Z
1010	GE	Greater than or equal (signed)	N = V
1011	LT	Less than (signed)	N != V
1100	GT	Greater than (signed)	!Z . (N = V)
1101	LE	Less than or equal (signed)	Z + (N != V)
1110	AL	Always	(ignored)

Notes:

- These codes occupy bits 28–31 of *all* ARM instructions
- Every code maps to a particular setting of bits to be matched in the CPSR
- But the op-codes are **not** a direct replica of the NZVC bits in the CPSR
- The sixteenth op-code (1111) is reserved, and must not be used
- In the absence of a suffix, the op-code of most instructions is set to 1110 (AL)

The ARM Data Processing Instructions

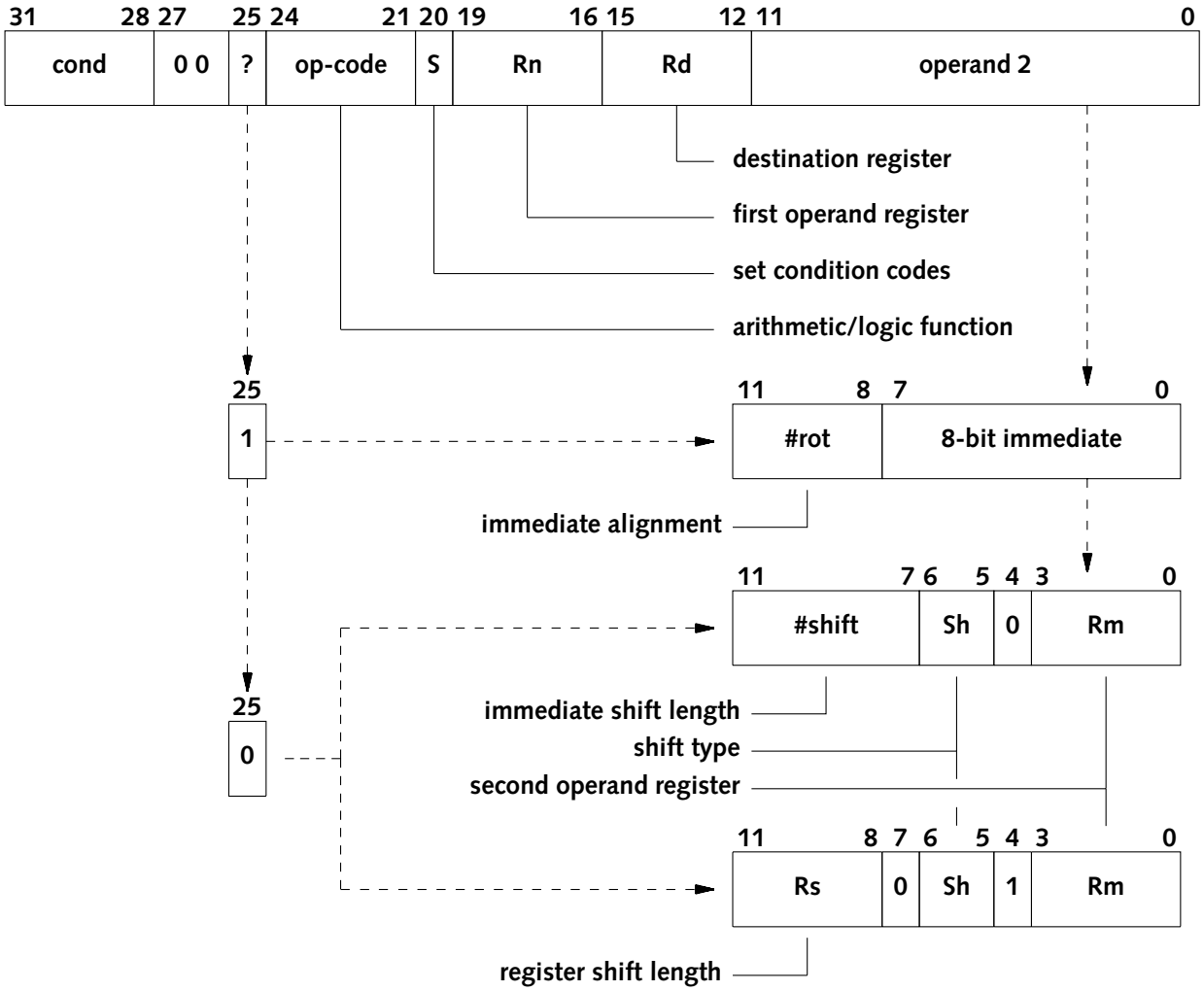
Opcode [24:21]	Mnemonic	Meaning	Effect
0000	AND	Logical bit-wise AND	Rd = Rn AND Op2
0001	EOR	Logical bit-wise exclusive OR	Rd = Rn EOR Op2
0010	SUB	Subtract	Rd = Rn – Op2
0011	RSB	Reverse Subtract	Rd = Op2 – Rn
0100	ADD	Add	Rd = Rn + Op2
0101	ADC	Add with carry	Rd + Rn + Op2 + C
0110	SBC	Subtract with carry	Rd = Rn - Op2 + C – 1
0111	RSC	Reverse subtract with carry	Rd = Op2 - Rn + C – 1
1000	TST	Test	Set cond. flags on Rn AND Op2
1001	TEQ	Test equivalence	Set cond. flags on Rn EOR Op2
1010	CMP	Compare	Set cond. flags on Rn – Op2
1011	CMN	Compare negated	Set cond. codes 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 NOT	Rd = NOT Op2

Notes:

- If the S bit (bit 20) is set in a Data Processing (DP) instruction then the CPSR's condition codes are affected by the execution of that instruction.
- At the assembler source level the setting of S is achieved by appending an S to the op-code e.g. MOVS
- DP instructions may be used to multiply a register by a small constant much more efficiently than with actual multiply intructions e.g. to multiply R0 by 5:

  ADD R0, R0, R0, LSL #2
- All DP instructions can rotate an immediate 8-bit operand so as to generate any 32-bit value whose 1 bits lie within an 8-bit span and are aligned on a 2-bit boundary.

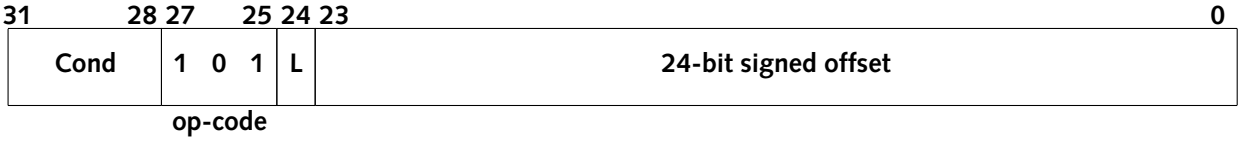
The ARM Data Processing Instruction Layout



Notes:

- The two-bit 'Sh' subfield (bits[6:5]) has the following encodings:
  00 means LSL
  01 means LSR
  10 means ASR
  11 ROR (but if shift length is zero it means RRX)
- When generating constants by rotation (bit 25 = 1) then if the [11:8] `#rot` sub-field is 0000, the gnerated small constant is just the 8-bit immediate value in [7:0].
- In all other cases where bit 25 is 1 the value in the `#rot` sub-field is *doubled* before a rotate right of that indicated amount is done.
- Thus the 8-bit immediate is rotated right through an even number of bit positions

Branch Instructions



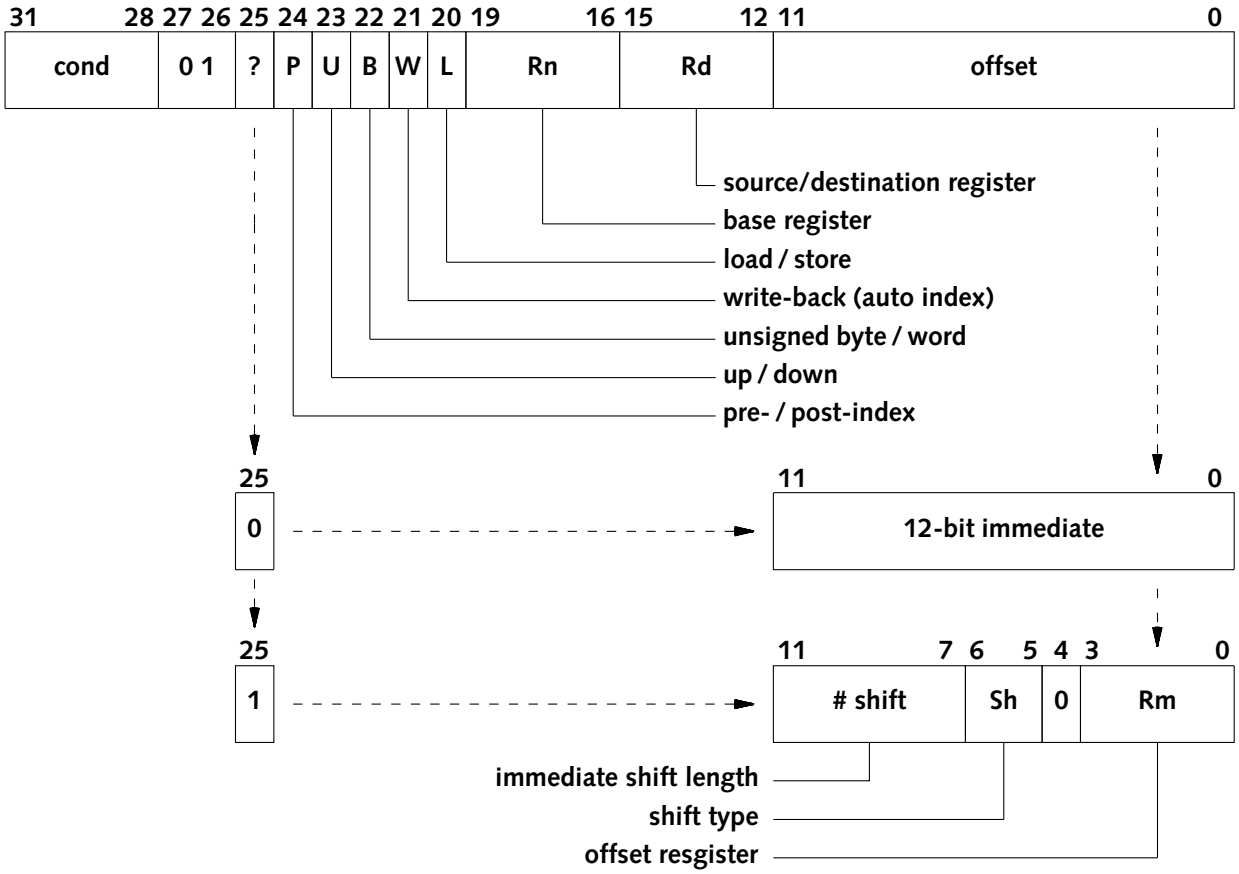
Notes:

- Normal Unconditional Branch (B) has L bit set to zero
- Unconditional branch-with-link (BL) has the L bit set to 1
- BL is used when branching to a subroutine
- BL leaves a return address in R14 (also known as the Link Register–LR)
- The 24-bit offset is always PC-relative. It is sign-extended and shifted 2 places left before being used
- Therefore, alignment on a 4-byte word boundary is assured. The offset is added to the current PC value
- This offset (effectively 26-bits), in two's complement, gives a branching range of ± 32M.
- Because of pipelining the PC value, on 32-bit ARM chips, is 8 bytes ahead of the current instruction

SWI Instructions

- SWI is a Software Interrupt Instruction
- Normally transfers control to the Operating System
- The options implemented by KoMoDo are:
  - SWI 0 outputs the LS byte of R0 to the terminal window
  - SWI 1 inputs character typed at terminal window into LS byte of R0
  - SWI 2 halts execution
  - SWI 3 prints a string, pointed to by R0
  - SWI 4 the value in R0 as a decimal integer
  - Any other value traps to an error handler
- SWI works to and from R0 *only*

The ARM Load and Store Instruction Layout



Notes:

- This format is for load (L = 1) or store (L = 0) as determined by bit 20.
- Whether it's a word (B = 0) or unsigned byte (B = 1) operation is chosen by bit 22. The corresponding op-codes are LDR, LDRB, STR, STRB,
- The base register address can be modified by a 12-bit immediate offset or by a value obtained from shifting a bit pattern within a third (offset) register.
- An immediate offset is an unsigned 12-bit value. Whether this adds (U = 1) or subtracts (U =0) to/from the base register value is determined by bit 23.
- If pre- (P = 1) or post- (P = 0) indexing is in use this is denoted by bit 24
- The pre-indexing write-back option is activated by bit 21
- The two-bit codes for the 'Sh' sub-field are the same as in the DP instructions

The ARM Multiply Instruction Layout

31 28	27 22	21	20	19 16	15 12	11 8	7 4	3 0
Cond	0 0 0 0 0 0	A	S	Rd	Rn	Rs	1 0 0 1	Rm

Notes:

- The two 32-bit instructions are MUL and MLA. Operands are *register only*.
- Bit 21 (labelled **A** above) is 0 for multiply-only (MUL) and 1 for multiply-and-accumulate (MLA).
- MUL, which performs Rd = Rm * Rs, ignores Rn. Bits 12 – 15 should accordingly be set to zero.
- MLA, which performs Rd = (Rm * Rs) + Rn, can save an explicit ADD instruction in some circumstances.
- The multiply instructions (just like the DP ones) have an option for setting the CPSR condition flags.

ARM Instruction Layout Summary

	31 30 29 28	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
Multiply	cond	0	0	0	0	0	0	A	S	Rd	Rn	Rs	1	0	0	1	Rm
Data Processing	cond	0	0	1	op-code				S	Rn	Rd	#rot	8-bit immediate				
"	cond	0	0	0	op-code				S	Rn	Rd	#shift		Sh		0	Rm
"	cond	0	0	0	op-code				S	Rn	Rd	Rs	0	Sh		1	Rm
Store/Load	cond	0	1	0	P	U	B	W	L	Rn	Rd	12-bit immediate					
"	cond	0	1	1	P	U	B	W	L	Rn	Rd	#shift		Sh		0	Rm
Branch	cond	1	0	1	L	24-bit signed offset											
SWI	cond	1	1	1	1	24-bit (interpreted) immediate											

For full information on each instruction type, please consult the relevant section of this document.