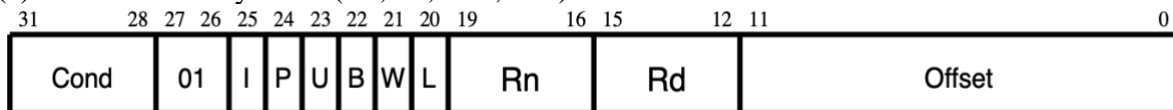# Format of Half word and Signed Data Transfer Instructions in ARM
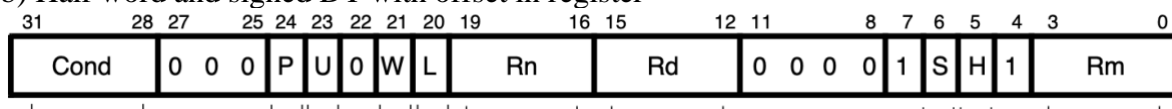
Half word and Signed DT Instructions in ARM have a different format than the full word and byte transfer instructions. For convenience, I am reproducing the relevant information from the document that had been posted on moodle earlier - "arm-instructionset.pdf".

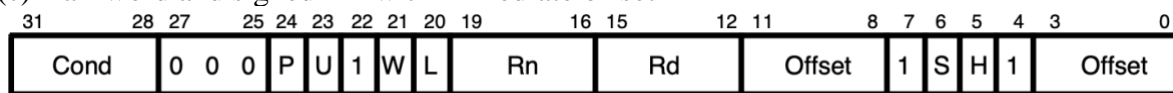First, let us compare the overall structure of the two formats.

(a)Full word and byte DT (ldr, str, ldrb, strb)

31 28	27 26	25	24	23	22	21	20	19 16	15 12	11 0
Cond	01	I	P	U	B	W	L	Rn	Rd	Offset

(b) Half word and signed DT with offset in register

31 28	27	26	25	24	23	22	21	20	19 16	15 12	11	10	9	8	7	6	5	4	3 0
Cond	0	0	0	P	U	0	W	L	Rn	Rd	0	0	0	0	1	S	H	1	Rm

(c) Half word and signed DT with immediate offset

31 28	27	26	25	24	23	22	21	20	19 16	15 12	11 8	7	6	5	4	3 0
Cond	0	0	0	P	U	1	W	L	Rn	Rd	Offset	1	S	H	1	Offset

Strangely, F field in (b) and (c) is "00" and I-bit (bit 25) is '0' which are same as that in DP instructions with operand2 in register! However, it does not clash with DP instructions which have a '0' either in bit 4 position (constant shift amount) or in bit 7 position (register specified shift amount), whereas these instructions have both these bits as '1'.

P, U, W and L bits have same meaning as in (a). Interpretation of S and H bits is as follows.

```
S H
0 0    swap (swp)    - - not to be implemented
0 1    unsigned half word load or store (ldrh or strh)
1 0    signed byte transfer load (ldrsb)
1 1    signed half word load (ldrsh)
```

Note that signed/unsigned distinction is only for loads, not stores. This is so because a byte or a half word has to be extended to make a full word while loading into register file, whereas there is no such extension in case of store.

Also note that in (a), there is a provision for shift/rotate of offset when the offset is given by a register, but there is no such provision in (b).

Figures on the next page elaborate the formats further. Full details about the instructions are available in "arm-instructionset.pdf".
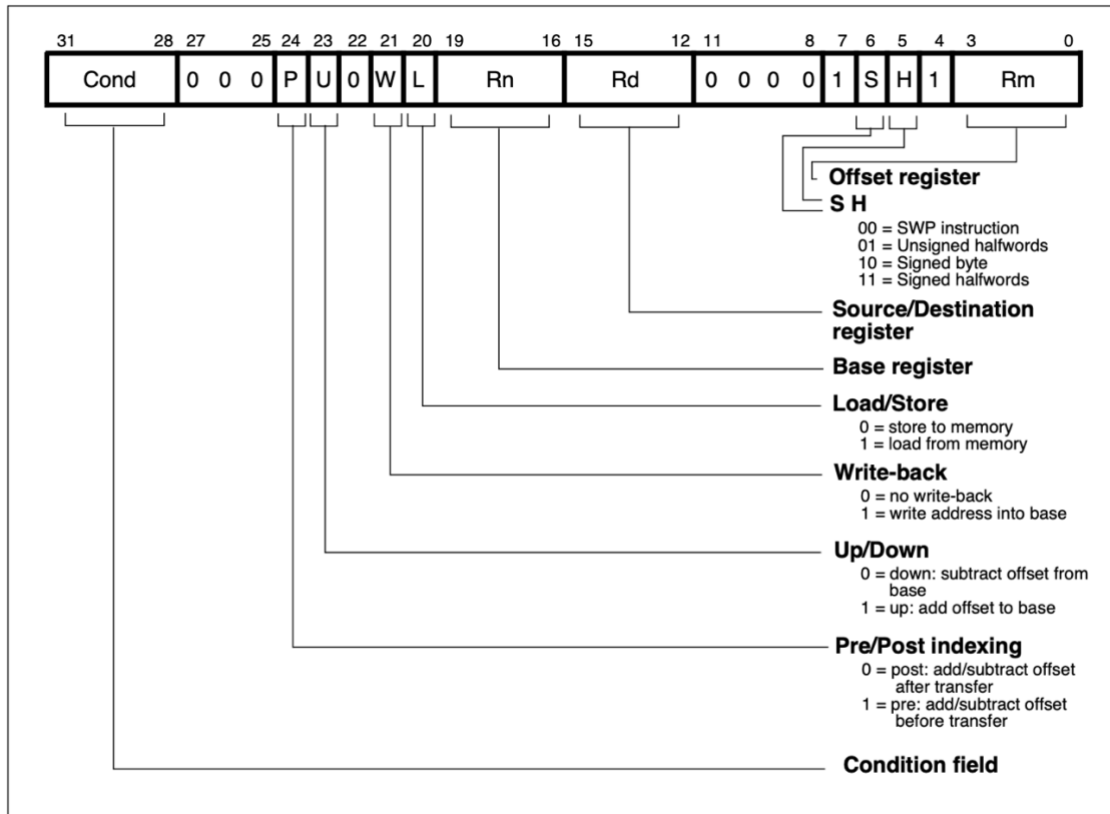
Figure 4-16

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	0	W	L	Rn		Rd		0 0 0 0		1	S	H	1	Rm	

Offset register

S H

00 = SWP instruction
01 = Unsigned halfwords
10 = Signed byte
11 = Signed halfwords

Source/Destination register

Base register

Load/Store

0 = store to memory
1 = load from memory

Write-back

0 = no write-back
1 = write address into base

Up/Down

0 = down: subtract offset from base
1 = up: add offset to base

Pre/Post indexing

0 = post: add/subtract offset after transfer
1 = pre: add/subtract offset before transfer

Condition field

Figure 4-16: Halfword and signed data transfer with register offset

Figure 4-17

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	1	W	L	Rn		Rd		Offset		1	S	H	1	Offset	

Immediate Offset (Low nibble)

S H

00 = SWP instruction
01 = Unsigned halfwords
10 = Signed byte
11 = Signed halfwords

Immediate Offset (High nibble)

Source/Destination register

Base register

Load/Store

0 = store to memory
1 = load from memory

Write-back

0 = no write-back
1 = write address into base

Up/Down

0 = down: subtract offset from base
1 = up: add offset to base

Pre/Post indexing

0 = post: add/subtract offset after transfer
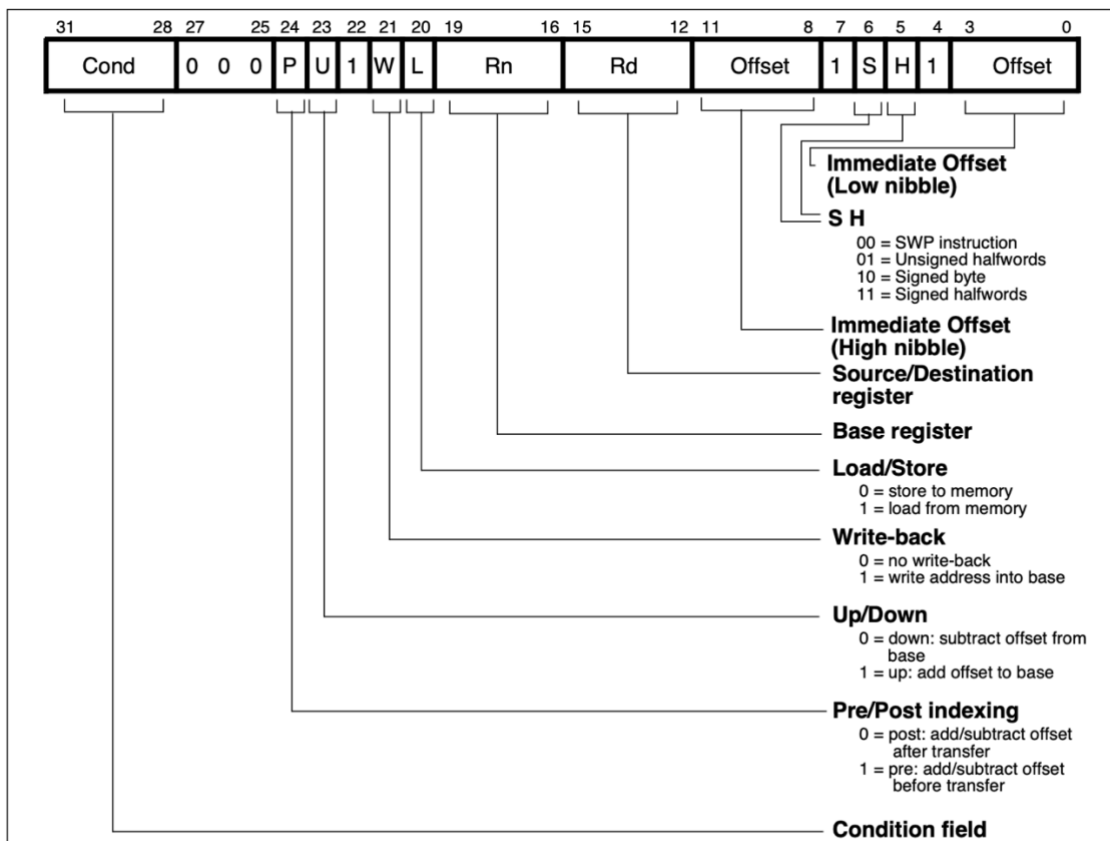1 = pre: add/subtract offset before transfer

Condition field

Figure 4-17: Halfword and signed data transfer with immediate offset