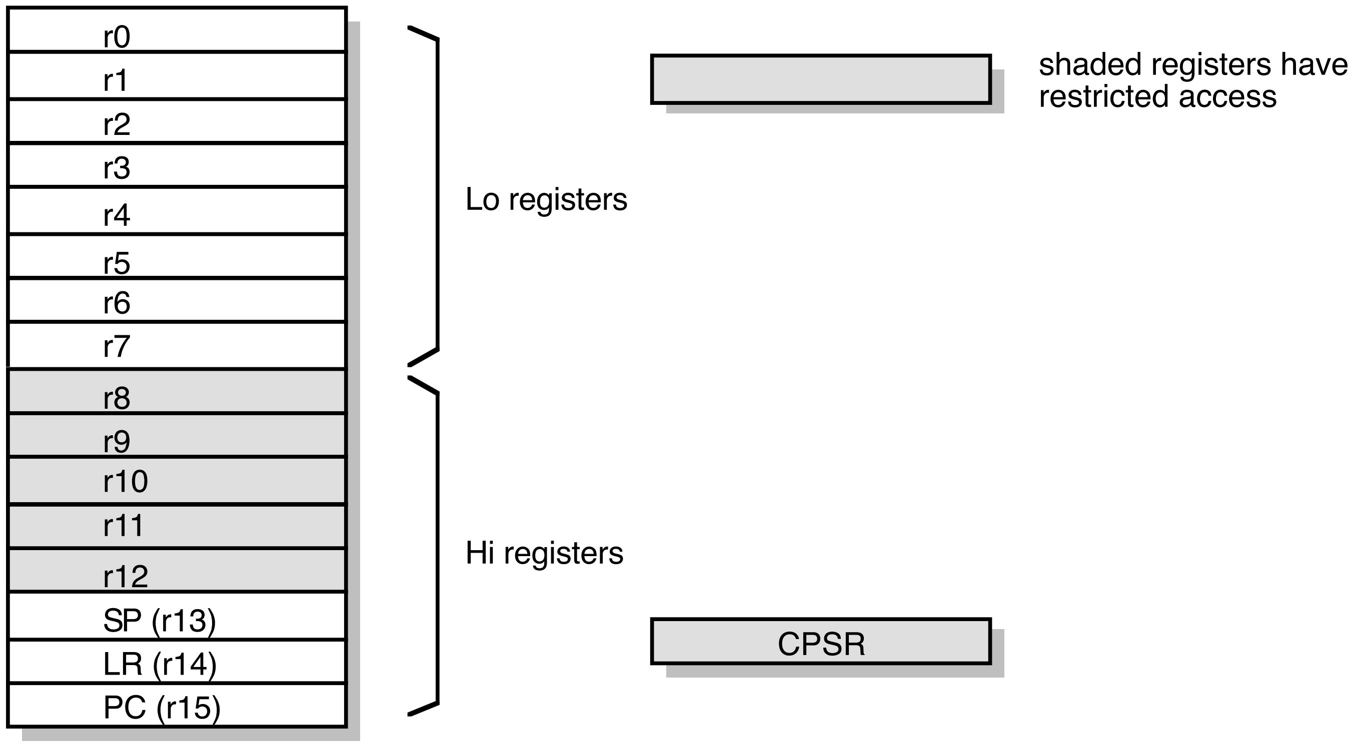
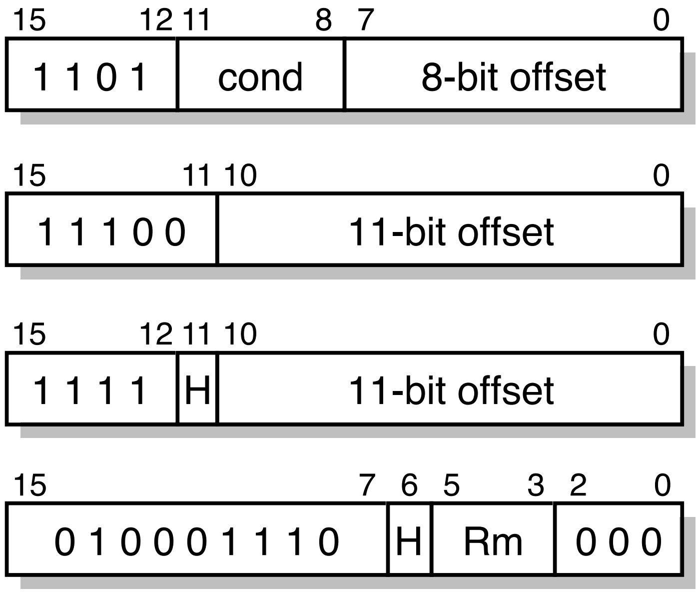
Thumb accessible registers



Thumb branch instruction binary encodings



(1) `B<cond> <label>`

(2) `B <label>`

(3) `BL <label>`

(4) `BX Rm`

Thumb software interrupt binary encoding

15 … 8	7 … 0
1 1 0 1 1 1 1 1	8-bit immediate

Thumb data processing instruction binary encodings

15 10	9	8 6	5 3	2 0
0 0 0 1 1 0	A	Rm	Rn	Rd

(1) `ADD|SUB Rd,Rn,Rm`

15 10	9	8 6	5 3	2 0
0 0 0 1 1 1	A	#imm3	Rn	Rd

(2) `ADD|SUB Rd,Rn,#imm3`

15 13	12 11	10 8	7 0
0 0 1	Op	Rd/Rn	#imm8

(3) `<Op> Rd/Rn,#imm8`

15 13	12 11	10 6	5 3	2 0
0 0 0	Op	#sh	Rn	Rd

(4) `LSL|LSR|ASR Rd,Rn,#shift`

15 10	9 6	5 3	2 0
0 1 0 0 0 0	Op	Rm/Rs	Rd/Rn

(5) `<Op> Rd/Rn,Rm/Rs`

15 10	9 8	7	6	5 3	2 0
0 1 0 0 0 1	Op	D	M	Rm	Rd/Rn

(6) `ADD|CMP|MOV Rd/Rn,Rm`

15 12	11	10 8	7 0
1 0 1 0	R	Rd	#imm8

(7) `ADD Rd,SP|PC,#imm8`

15 8	7	6 0
1 0 1 1 0 0 0 0	A	#imm7

(8) `ADD|SUB SP,SP,#imm7`

Thumb single register data transfer binary encodings

15–13	12	11	10–6	5–3	2–0
0 1 1	B	L	#off5	Rn	Rd

(1) `LDR|STR{B} Rd,[Rn,#off5]`

15–12	11	10–6	5–3	2–0
1 0 0 0	L	#off5	Rn	Rd

(2) `LDRH|STRH Rd,[Rn,#off5]`

15–12	11–9	8–6	5–3	2–0
0 1 0 1	Op	Rm	Rn	Rd

(3) `LDR|STR{S}{H|B} Rd,[Rn,Rm]`

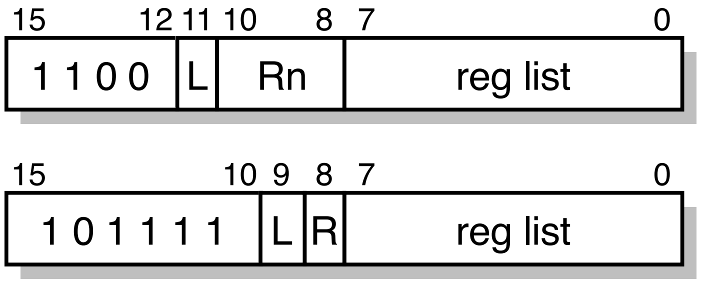
15–11	10–8	7–0
0 1 0 0 1	Rd	#off8

(4) `LDR Rd,[PC,#off8]`

15–12	11	10–8	7–0
1 0 0 1	L	Rd	#off8

(5) `LDR|STR Rd,[SP,#off8]`

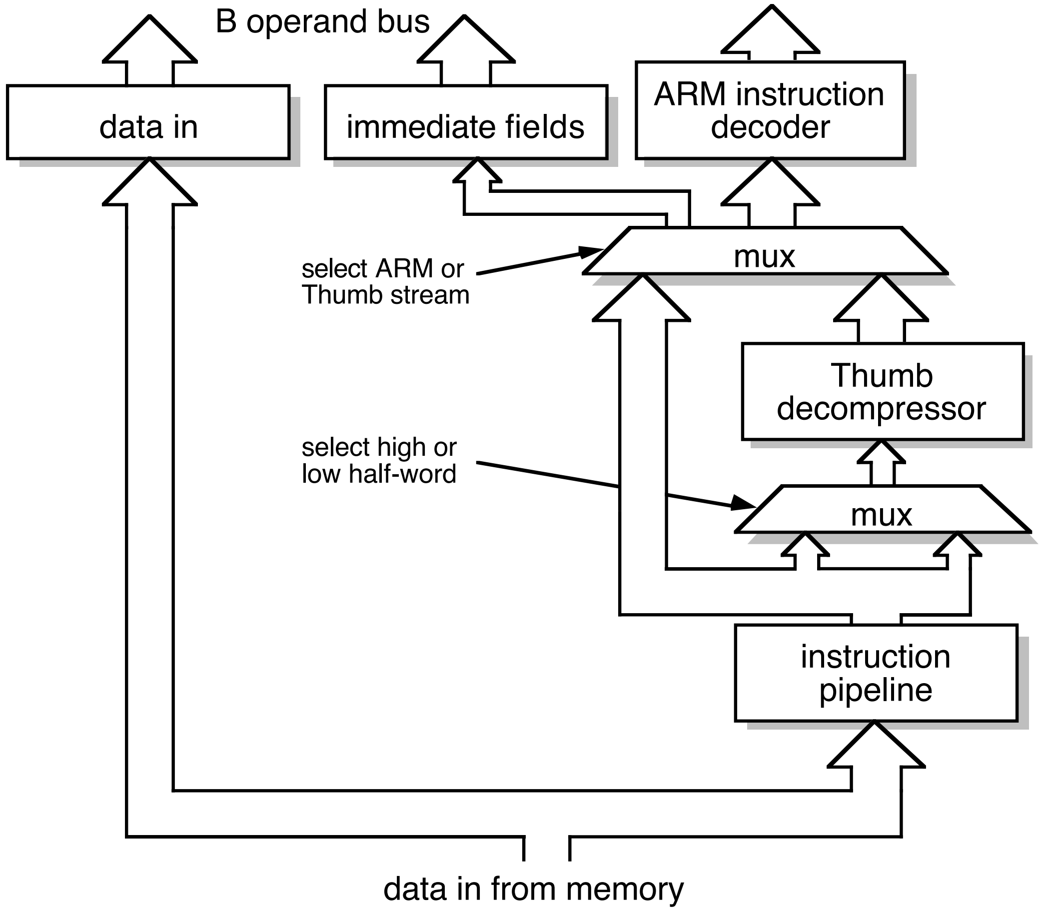
Thumb multiple register data transfer binary encodings



(1) `LDMIA|STMIA Rn!, {<reg list>}`

(2) `POP|PUSH {<reg list>{,R}}`

The Thumb instruction decompressor organization



Thumb to ARM instruction mapping