

# Thumb

Instruction set alternativo per processori ARM in cui le istruzioni sono da 16 bit. Per raggiungere l'obiettivo sono state introdotte alcune limitazioni, tra cui:

- accesso solo i primi 8 registri (-1 bit);
- spesso il registro destinazione è anche il primo operando (-4);
- immediati più corti (-4);
- esecuzione condizionale solo su salti (-4);
- tutte le operative aggiornano i flag (-1).

Inoltre caratteristiche e rappresentazione in linguaggio macchina delle istruzioni sono irregolari. Per esempio:

- **bl** è su 32 bit e supporta immediati da 22 bit, visto che spesso si usa per chiamare una funzione di libreria lontana. **b** invece è su 16 bit, con 11 di immediato;
- **bx** ha l'identificatore di registro da 4 bit per poter accedere a LR;

I processori che supportano sia Thumb che l'IS regolare possono cambiare modalità di esecuzione con **bx/blx**; la modalità corrente si trova in un bit del CPSR. In assembly si specifica una sezione di codice Thumb con **.code 16**.

La dimensione ridotta del codice (~ 65% anziché 50% perché i programmi Thumb usano più istruzioni) e delle istruzioni permette di ridurre il costo dell'hardware e i consumi.

0			
0	1	0	0
0	0	ASR LSR	funct
0	0	1	1
0	0	1	SUB
0	1	0	1
1	0	1	0
0	0	1	0
0	1	0	0
0	1	0	0
1	1	0	1
0	1	0	0
1	1	0	1
0	1	0	1
0	1	0	0
1	1	1	0
imm5			
imm3			
imm8			
Rdn [3]			
Rdn			
imm8			
Rdn [3]			
Rdn			
imm7			
Rn			
imm8			
Rd			
imm8			
Rdn [3]			
Rm			
Rdn[2:0]			
imm7			
Rn			
imm8			
Rdn			
Rm			
imm8			
B imm11			
STR(B/H)/LDR(B/H) Rd, [Rn, Rm]			
STR/LDR Rd, [Rn, #imm5]			
STR/LDR Rd, [SP, #imm8]			
LDR Rd, [PC, #imm8]			
imm22[21:11]			
1	1	1	1
imm22[10:0]			
BL imm22			