

# 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.

150															
0 1 0 0 0 0				funcnt				Rm				Rdn			
0 0 0 ASR LSR				imm5				Rm				Rd			
0 0 0 1 1 1 SUB				imm3				Rm				Rd			
0 0 1 1 SUB				Rdn				imm8				ADDs/SUBs Rdn, Rdn, #imm8			
0 1 0 0 0 1 0 0				Rdn [3]				Rm				Rdn [2:0]			
1 0 1 1 0 0 0 0				SUB				imm7				ADD / SUB SP, SP, #imm7			
0 0 1 0 1				Rn				imm8				CMP Rn, #imm8			
0 0 1 0 0				Rd				imm8				MOV Rdn, #imm8			
0 1 0 0 0 1 1 0				Rdn [3]				Rm				Rdn [2:0]			
0 1 0 0 0 1 1 1				L				Rm				0 0 0			
1 1 0 1				cond				imm8				B<cond> imm8			
1 1 1 0 0				imm8				B imm11							
0 1 0 1				L B H				Rm				Rn			
0 1 1 0				L				imm5				Rn			
1 0 0 1				L				Rd				imm8			
0 1 0 0				1				Rd				imm8			
1 1 1 1				0				imm22[21:11]				1 1 1 1			
												imm22[10:0]			

BL imm22