

## Conjunto de instrucciones ARM Thumb (1/2)

Operación	Ensamblador	Acción	Actualiza	Notas
<b>Mover</b>	Immediato 8 bits	mov Rd, #Imm8	Rd ← Imm8	N Z
	Lo a Lo	mov Rd, Rm	Rd ← Rm	N Z
	Hi a Lo, Lo a Hi, Hi a Hi	mov Rd, Rm	Rd ← Rm	
<b>Sumar</b>	Immediato 3 bits	add Rd, Rn, #Imm3	Rd ← Rn + Imm3	N Z C V
	Immediato 8 bits	add Rd, Rd, #Imm8	Rd ← Rd + Imm8	N Z C V
	Lo a Lo	add Rd, Rn, Rm	Rd ← Rn + Rm	N Z C V
	Hi a Lo, Lo a Hi, Hi a Hi	add Rd, Rd, Rm	Rd ← Rd + Rm	
	Valor a SP	add SP, SP, #Imm	SP ← SP + Imm	
	Crear dirección desde SP	add Rd, SP, #Imm	Rd ← SP + Imm	
<b>Restar</b>	Immediato 3 bits	sub Rd, Rn, #Imm3	Rd ← Rn - Imm3	N Z C V
	Immediato 8 bits	sub Rd, Rd, #Imm8	Rd ← Rd - Imm8	N Z C V
	Lo a Lo	sub Rd, Rn, Rm	Rd ← Rn - Rm	N Z C V
	Valor de SP	sub SP, SP, #Imm	SP ← SP - Imm	
	Negar	neg Rd, Rn	Rd ← -Rn	N Z C V
<b>Multiplicar</b>	Multiplifica	mul Rd, Rm, Rd	Rd ← Rm * Rd	N Z
<b>Comparar</b>	Compara	cmp Rn, Rm	Act. flags según Rn - Rm	N Z C V
	Compara con negado	cmn Rn, Rm	Act. flags según Rn + Rm	N Z C V
	Immediato	cmp Rn, #Imm8	Act. flags según Rn - Imm8	N Z C V
<b>Lógicas</b>	AND	and Rd, Rd, Rm	Rd ← Rd AND Rm	N Z
	AND NOT (borrar bits)	bic Rd, Rd, Rm	Rd ← Rd AND NOT Rm	N Z
	OR	orr Rd, Rd, Rm	Rd ← Rd OR Rm	N Z
	XOR (or exclusiva)	eor Rd, Rd, Rm	Rd ← Rd XOR Rm	N Z
	NOT	mvn Rd, Rm	Rd ← NOT Rm	N Z
	Comprueba bits	tst Rn, Rm	Act. flags según Rn AND Rm	N Z
<b>Desplazar</b>	Lógico a la izquierda	lsl Rd, Rm, #Shift	Rd ← Rm << Shift	N Z C
		lsl Rd, Rd, Rs	Rd ← Rd << [Rs] <sub>7:0</sub>	N Z C
	Lógico a la derecha	lsr Rd, Rm, #Shift	Rd ← Rm >> Shift	N Z C
		lsr Rd, Rd, Rs	Rd ← Rd >> [Rs] <sub>7:0</sub>	N Z C
	Aritmético a la derecha	asr Rd, Rm, #Shift	Rd ← Rm >> <sub>a</sub> Shift	N Z C
		asr Rd, Rd, Rs	Rd ← Rd >> <sub>a</sub> [Rs] <sub>7:0</sub>	N Z C
	Rotación a la derecha	ror Rd, Rd, Rs	Rd ← Rd ROR [Rs] <sub>7:0</sub>	N Z C

## Conjunto de instrucciones ARM Thumb (2/2)

Operación	Ensamblador	Acción	Notas
<b>Cargar</b>	Con desp. inm., palabra	<b>ldr</b> Rd, [Rn, #Imm]	Rd $\leftarrow$ [Rn + Imm]
	media palabra	<b>ldrh</b> Rd, [Rn, #Imm]	Rd $\leftarrow$ ZeroExtend([Rn + Imm] <sub>15:0</sub> )
	byte	<b>ldrb</b> Rd, [Rn, #Imm]	Rd $\leftarrow$ ZeroExtend([Rn + Imm] <sub>7:0</sub> )
	Con desp. en registro, palabra	<b>ldr</b> Rd, [Rn, Rm]	Rd $\leftarrow$ [Rn + Rm]
	media palabra	<b>ldrh</b> Rd, [Rn, Rm]	Rd $\leftarrow$ ZeroExtend([Rn + Rm] <sub>15:0</sub> )
	media palabra con signo	<b>ldrsh</b> Rd, [Rn, Rm]	Rd $\leftarrow$ SignExtend([Rn + Rm] <sub>15:0</sub> )
	byte	<b>ldrb</b> Rd, [Rn, Rm]	Rd $\leftarrow$ ZeroExtend([Rn + Rm] <sub>7:0</sub> )
	byte con signo	<b>ldrsb</b> Rd, [Rn, Rm]	Rd $\leftarrow$ SignExtend([Rn + Rm] <sub>7:0</sub> )
	Relativo al PC	<b>ldr</b> Rd, [PC, #Imm]	Rd $\leftarrow$ [PC + Imm]
	Relativo al SP	<b>ldr</b> Rd, [SP, #Imm]	Rd $\leftarrow$ [SP + Imm]
<b>Almacenar</b>	Con desp. inm., palabra	<b>str</b> Rd, [Rn, #Imm]	[Rn + Imm] $\leftarrow$ Rd
	media palabra	<b>strh</b> Rd, [Rn, #Imm]	[Rn + Imm] <sub>15:0</sub> $\leftarrow$ Rd <sub>15:0</sub>
	byte	<b>strb</b> Rd, [Rn, #Imm]	[Rn + Imm] <sub>7:0</sub> $\leftarrow$ Rd <sub>7:0</sub>
	Con desp. en registro, palabra	<b>str</b> Rd, [Rn, Rm]	[Rn + Rm] $\leftarrow$ Rd
	media palabra	<b>strh</b> Rd, [Rn, Rm]	[Rn + Rm] <sub>15:0</sub> $\leftarrow$ Rd <sub>15:0</sub>
	byte	<b>strb</b> Rd, [Rn, Rm]	[Rn + Rm] <sub>7:0</sub> $\leftarrow$ Rd <sub>7:0</sub>
	Relativo al SP	<b>str</b> Rd, [SP, #Imm]	[SP + Imm] $\leftarrow$ Rd
<b>Apilar</b>	Apilar	<b>push</b> <loreglist>	Apila registros en la pila
	Apilar y enlazar	<b>push</b> <loreglist+LR>	Apila LR y registros en la pila
<b>Desapilar</b>	Desapilar	<b>pop</b> <loreglist>	Desapila registros de la pila
	Desapilar y retorno	<b>pop</b> <loreglist+PC>	Desapila registros y salta a la dirección cargada en el PC
<b>Saltar</b>	Salto condicional	<b>b{cond}</b> <label>	Si {cond}, PC $\leftarrow$ label (rango salto: -252 a +258 bytes de la instrucción actual).
	Salto incondicional	<b>b</b> <label>	PC $\leftarrow$ label (rango salto: $\pm 2$ KiB de la instrucción actual).
	Salto largo y enlaza	<b>bl</b> <label>	LR $\leftarrow$ dirección de la siguiente instrucción, PC $\leftarrow$ label (Instrucción de 32 bits. Rango salto: $\pm 4$ MiB de la instrucción actual).
<b>Extender</b>	Con signo, media a palabra	<b>sxtb</b> Rd, Rm	Rd <sub>31:0</sub> $\leftarrow$ SignExtend(Rm <sub>15:0</sub> )
	Con signo, byte a palabra	<b>sxtb</b> Rd, Rm	Rd <sub>31:0</sub> $\leftarrow$ SignExtend(Rm <sub>7:0</sub> )
	Sin signo, media a palabra	<b>uxth</b> Rd, Rm	Rd <sub>31:0</sub> $\leftarrow$ ZeroExtend(Rm <sub>15:0</sub> )
	Sin signo, byte a palabra	<b>uxtb</b> Rd, Rm	Rd <sub>31:0</sub> $\leftarrow$ ZeroExtend(Rm <sub>7:0</sub> )
{cond}	EQ Igual	NE Distinto	MI Negativo
	HI Mayor sin signo	CS Mayor o igual sin signo	CC Menor sin signo
	GT Mayor que	GE Mayor o igual	LT Menor que
			PL Positivo
			LS Menor o igual sin signo
			LE Menor o igual que
			VS Desbordamiento
			VC No desbordamiento