

Conjunto de instrucciones ARM Thumb (1/2)

Operación		Ensamblador	Acción	Actualiza	Notas
Mover	Inmediato 8 bits	mov Rd, #Inm8	$Rd \leftarrow \text{Inm8}$	N Z	Rango Inm8: 0–255.
	Lo a Lo	mov Rd, Rm	$Rd \leftarrow Rm$	N Z	
	Hi a Lo, Lo a Hi, Hi a Hi	mov Rd, Rm	$Rd \leftarrow Rm$		
Sumar	Inmediato 3 bits	add Rd, Rn, #Inm3	$Rd \leftarrow Rn + \text{Inm3}$	N Z C V	Rango Inm3: 0–7.
	Inmediato 8 bits	add Rd, Rd, #Inm8	$Rd \leftarrow Rd + \text{Inm8}$	N Z C V	Rango Inm8: 0–255.
	Lo a Lo	add Rd, Rn, Rm	$Rd \leftarrow Rn + Rm$	N Z C V	
	Hi a Lo, Lo a Hi, Hi a Hi	add Rd, Rd, Rm	$Rd \leftarrow Rd + Rm$		
	Valor a SP	add SP, SP, #Inm	$SP \leftarrow SP + \text{Inm}$		Rango Inm: 0–508 (alineado a palabra).
	Crear dirección desde SP	add Rd, SP, #Inm	$Rd \leftarrow SP + \text{Inm}$		Rango Inm: 0–1020 (alineado a palabra).
Restar	Inmediato 3 bits	sub Rd, Rn, #Inm3	$Rd \leftarrow Rn - \text{Inm3}$	N Z C V	Rango Inm3: 0–7.
	Inmediato 8 bits	sub Rd, Rd, #Inm8	$Rd \leftarrow Rd - \text{Inm8}$	N Z C V	Rango Inm8: 0–255.
	Lo a Lo	sub Rd, Rn, Rm	$Rd \leftarrow Rn - Rm$	N Z C V	
	Valor de SP	sub SP, SP, #Inm	$SP \leftarrow SP - \text{Inm}$		
	Negar	neg Rd, Rn	$Rd \leftarrow -Rn$	N Z C V	
Multiplicar	Multiplica	mul Rd, Rm, Rd	$Rd \leftarrow Rm * Rd$	N Z	
Comparar	Compara	cmp Rn, Rm	Act. flags según $Rn - Rm$	N Z C V	Cualquier registro a cualquier registro.
	Compara con negado	cmn Rn, Rm	Act. flags según $Rn + Rm$	N Z C V	
	Inmediato	cmp Rn, #Inm8	Act. flags según $Rn - \text{Inm8}$	N Z C V	Rango Inm8: 0–255.
Lógicas	AND	and Rd, Rm	$Rd \leftarrow Rd \text{ AND } Rm$	N Z	
	AND NOT (borrar bits)	bic Rd, Rm	$Rd \leftarrow Rd \text{ AND NOT } Rm$	N Z	
	OR	orr Rd, Rm	$Rd \leftarrow Rd \text{ OR } Rm$	N Z	
	XOR (or exclusiva)	eor Rd, Rm	$Rd \leftarrow Rd \text{ XOR } Rm$	N Z	
	NOT	mvn Rd, Rm	$Rd \leftarrow \text{NOT } Rm$	N Z	
	Comprueba bits	tst Rn, Rm	Act. flags según $Rn \text{ AND } Rm$	N Z	
Desplazar	Lógico a la izquierda	lsl Rd, Rm, #Shift	$Rd \leftarrow Rm \ll \text{Shift}$	N Z C	Rango Shift: 0–31
		lsl Rd, Rd, Rs	$Rd \leftarrow Rd \ll [Rs]_{7:0}$	N Z C	
	Lógico a la derecha	lsr Rd, Rm, #Shift	$Rd \leftarrow Rm \gg \text{Shift}$	N Z C	Rango Shift: 0–31
		lsr Rd, Rd, Rs	$Rd \leftarrow Rd \gg [Rs]_{7:0}$	N Z C	
	Aritmético a la derecha	asr Rd, Rm, #Shift	$Rd \leftarrow Rm \gg_a \text{Shift}$	N Z C	Rango Shift: 0–31
		asr Rd, Rd, Rs	$Rd \leftarrow Rd \gg_a [Rs]_{7:0}$	N Z C	
	Rotación a la derecha	ror Rd, Rd, Rs	$Rd \leftarrow Rd \text{ ROR } [Rs]_{7:0}$	N Z C	

Directivas del ensamblador			
.align N	Siguiente dato empieza en dir. múltiplo de 2^N .	.equiv símbolo, expr	Como .equ , pero da error si existe símbolo.
.ascii "cadena"	Inicializa una zona de memoria con los caracteres UTF-8 de cadena.	.eqv	Equivalente a .equiv .
.asciz "cadena"	Idem que la anterior, pero termina con 0.	.hword valor	Inicializa una media palabra a valor.
.balign N	Siguiente dato empieza en dir. múltiplo de N.	.quad valor	Inicializa una doble palabra a valor.
.byte valor	Inicializa un byte a valor.	símbolo .req rd	Define símbolo como un alias para rd.
.text	Ensambla lo que sigue en la zona de código.	.set	Equivalente a .equ .
.data	Ensambla lo que sigue en la zona de datos.	.space N	Reserva N bytes de memoria a 0.
.end	No hay más instrucciones	símbolo .unreq símbolo	Cancela el alias símbolo.
.equ símbolo, expr	Asigna el valor de expr a símbolo.	.word valor	Inicializa una palabra a valor.

Conjunto de instrucciones ARM Thumb (2/2)

Operación		Ensamblador	Acción	Notas
Cargar	Con desp. inm., palabra	ldr Rd, [Rn, #Inm]	$Rd \leftarrow [Rn + Inm]$	Rango Inm: 0–124, múltiplos de 4.
	media palabra	ldrh Rd, [Rn, #Inm]	$Rd \leftarrow ZeroExtend([Rn + Inm]_{15:0})$	Bits 31:16 a 0. Rango Inm: 0–62, pares.
	byte	ldrb Rd, [Rn, #Inm]	$Rd \leftarrow ZeroExtend([Rn + Inm]_{7:0})$	Bits 31:8 a 0. Rango Inm: 0–31.
	Con desp. en registro, palabra	ldr Rd, [Rn, Rm]	$Rd \leftarrow [Rn + Rm]$	
	media palabra	ldrh Rd, [Rn, Rm]	$Rd \leftarrow ZeroExtend([Rn + Rm]_{15:0})$	Bits 31:16 a 0.
	media palabra con signo	ldrsh Rd, [Rn, Rm]	$Rd \leftarrow SignExtend([Rn + Rm]_{15:0})$	Bits 31:16 igual al bit 15.
	byte	ldrb Rd, [Rn, Rm]	$Rd \leftarrow ZeroExtend([Rn + Rm]_{7:0})$	Bits 31:8 a 0.
	byte con signo	ldrnb Rd, [Rn, Rm]	$Rd \leftarrow SignExtend([Rn + Rm]_{7:0})$	Bits 31:8 igual al bit 7.
	Relativo al PC	ldr Rd, [PC, #Inm]	$Rd \leftarrow [PC + Inm]$	Rango Inm: 0–1020, múltiplos de 4.
Almacenar	Relativo al SP	ldr Rd, [SP, #Inm]	$Rd \leftarrow [SP + Inm]$	Rango Inm: 0–1020, múltiplos de 4.
	Con desp. inm., palabra	str Rd, [Rn, #Inm]	$[Rn + Inm] \leftarrow Rd$	Rango Inm: 0–124, múltiplos de 4.
	media palabra	strh Rd, [Rn, #Inm]	$[Rn + Inm]_{15:0} \leftarrow Rd_{15:0}$	Rd _{31:16} se ignora. Rango Inm: 0–62, pares.
	byte	strb Rd, [Rn, #Inm]	$[Rn + Inm]_{7:0} \leftarrow Rd_{7:0}$	Rd _{31:8} se ignora. Rango Inm: 0–31.
	Con desp. en registro, palabra	str Rd, [Rn, Rm]	$[Rn + Rm] \leftarrow Rd$	
	media palabra	strh Rd, [Rn, Rm]	$[Rn + Rm]_{15:0} \leftarrow Rd_{15:0}$	Rd _{31:16} se ignora.
	byte	strb Rd, [Rn, Rm]	$[Rn + Rm]_{7:0} \leftarrow Rd_{7:0}$	Rd _{31:8} se ignora.
	Relativo al SP	str Rd, [SP, #Inm]	$[SP + Inm] \leftarrow Rd$	Rango Inm: 0–1020, múltiplos de 4.
Apilar	Apilar	push <loreglist>	Apila registros en la pila	
	Apilar y enlazar	push <loreglist+LR>	Apila LR y registros en la pila	
Desapilar	Desapilar	pop <loreglist>	Desapila registros de la pila	
	Desapilar y retorno	pop <loreglist+PC>	Desapila registros y salta a la dirección cargada en el PC	
Saltar	Salto condicional	b {cond} <label>	Si {cond}, $PC \leftarrow label$ (rango salto: –252 a +258 bytes de la instrucción actual).	
	Salto incondicional	b <label>	$PC \leftarrow label$ (rango salto: ± 2 KiB de la instrucción actual).	
	Salto largo y enlaza	bl <label>	LR \leftarrow dirección de la siguiente instrucción, $PC \leftarrow label$	
			(Instrucción de 32 bits. Rango salto: ± 4 MiB de la instrucción actual).	
Extender	Con signo, media a palabra	sxth Rd, Rm	$Rd_{31:0} \leftarrow SignExtend(Rm_{15:0})$	
	Con signo, byte a palabra	sxtb Rd, Rm	$Rd_{31:0} \leftarrow SignExtend(Rm_{7:0})$	
	Sin signo, media a palabra	uxth Rd, Rm	$Rd_{31:0} \leftarrow ZeroExtend(Rm_{15:0})$	
	Sin signo, byte a palabra	uxtb Rd, Rm	$Rd_{31:0} \leftarrow ZeroExtend(Rm_{7:0})$	

{cond}	EQ	Igual	z	CS	Mayor o igual sin signo	C	LT	Menor que	Nv o nV	VS	No desbordamiento	V
	HI	Mayor sin signo	Cz	GE	Mayor o igual	NV o nv	PL	Positivo o cero	n	VC	Desbordamiento	v
	GT	Mayor que	z y (NV o nv)	MI	Negativo	N	LS	Menor o igual sin signo	C o Z			
	NE	Distinto	z	CC	Menor sin signo	c	LE	Menor o igual que	Nv o nV o Z			