

Codificaciónes posibles de

(-2 147 483 648 a 2 147 483 647)

complemento a 2 y que estaremos con 32 bits y por ende los limites son:

Extra:

• Siempre 00000000

 Recordar que lo trabajaremos con

ì	as explicitamente (Instrucciones de Compilador, que traduce a set de instrucciones reales para la memoria de instruciones)
Nombre reservado BOTON DE CARGADO	Instruccion(es) que ejecuta en realidad • Modifica directamente en el excel los valores de:
DE DATOS (HighLevel)	 w2: Cantidad de bloques de 64bits disponibles en la memoria dinamica D[]: Asegura la memoria dinamica para que nuestro CPU la pueda modificar o leer
AUTHCMP	• CMPS R1,P1 • CMPS R2,P2 • CMPS R3,P3 • CMPS R4,P4 • CMPS R5,P5
	• CMPS R6,P6 • CMPS R7,P7 • CMPS R8,P8
TEA #0, #val	•MOVI W6, #val •Nota: el #val sera: #0x0, para el proceso de cifrado y 0xC6EF3720, para el proceso de descifrado •MOVI W7, #32
TEAENC #1, k?(0/1/2/3)	NOP
TEAENC #2	• ADD W9, W5, W6 • NOP • NOP • NOP • NOP • NOP • NOP
TEAENC #3, k?(0/1/2/3)	• LSRI W9, W5, #5 • NOP • NOP • NOP • NOP • NOP • NOP • ADDS W9, W9, k?.1 • NOP
TEAENC #4, k?(0/1/2/3)	• NOP
TEAENC #5	• ADD W9, W4, W6 • NOP • NOP • NOP • NOP • NOP • NOP
TEAENC #6, k?(0/1/2/3)	• LSRI W9, W4, #5 • NOP • NOP • NOP • NOP • NOP • ADDS W9, W9, k?.3 • NOP
TEAENC #7	• NOP • NOP • NOP • NOP • NOP • NOP • ADD W5, W3, W8 • SUBI W7, W7, #1
TEAD #1, k?(0/1/2/3)	• NOP • NOP • NOP • NOP • NOP • NOP • LSLI W8, W4, #4 • NOP • ADDS W8, W8, k2.2 • ADD W9, W4, W6
TEAD #2, k?(0/1/2/3)	• NOP • EOR W8, W8, W9 • LSRI W9, W4, #5 • NOP
TEAD #3	NOP
TEAD #4, k?(0/1/2/3)	• LSLI W8, W5, #4 • NOP • NOP • NOP • NOP • NOP • ADDS W8, W8, k2.0 • ADD W9, W3, W6
TEAD #5, k?(0/1/2/3)	• NOP • NOP • NOP • NOP • NOP • NOP • EOR W8, W8, W9 • LSRI W9, W5, #5 • NOP • NOP • NOP • NOP • NOP • NOP
	• NOP • NOP

• SUB W4, W4, W8

SUB W6, W6, D0SUBI W7, W7, #1

				• 1110: R14/Inf • 1111: R15/Inf	• 1110: R14/Inf • 1111: R15/Inf	• 10110: R14/Inf/Inf/k3.2 • 1111: R15/In/Inf/k3.3								Se	ñales del Contr	ol Unit segun ca	ada operación						
Acción	Ejemplo		Intrucc	ciones Codificadas ex	kplicitamente, segun c	ada tipo de operación			Re	egisterInA (1b) RegisterInB (2b) Immediate(Op (1b) BranchE (1b)	BranchOp (3b)	ALUSrc (6b) Flagst	Jpd (1b) PCSrc (1b)	Υ	Υ	WriteS (1b) RegWriteR (1b	b) PrintEn (2b)	ComS (1b)	LogOut (1b) M	emWriteG(1b) MemW	VriteD(1b) MemWrite	V(1b) MemWriteP(1b)
Rd ← Rn + Rm	ADD Rd, Rn, Rm	ADD (0000000)	Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej = (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante)		(Repende de cual registro R/W) 00/01 (Repende de cual registro R/W)	0	111	000000	0	0	11 X (Rej	pende de cual X (Repende de cu gistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn + k?.?	ADDS Rd, Rn, k?.?	ADDS (0000001)	Rd/Wd, <mark>Rn/Wn</mark> ,Rm/Wm,G/D (XXXX)	Wd (destino) Ej = (0111):W8	Wn (A) Ej = (0111):W8	k?.? (B) Ej = (0100): k1.0	Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante)		0 10 0	0	111	000000	0	0	11	1 0	11	0	0	0	0 0	0
Rd ← Rn - Rm	SUB Rd, Rn, Rm	SUB (00000010)	Rd/Wd, <mark>Rn/Wn</mark> ,Rm/Wm,G/D (0 <mark>1</mark> 0X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante)	X (I	(Repende de cual registro R/W) O0/01 (Repende de cual registro R/W)	0	111	000001	0	0	11 X (Rej	pende de cual X (Repende de cu jistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn + Rm + C	ADC Rd, Rn, Rm	ADC (00000011)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXX : No importa	Extra(No relevante)		(Repende de cual registro R/W) O0/01 (Repende de cual registro R/W)	0	111	000010	0	0	11 X (Rej	pende de cual X (Repende de cu jistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn - Rm - !C	SBC Rd, Rn, Rm	SBC (00000100)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXX : No importa	Extra(No relevante)		(Repende de cual registro R/W) 00/01 (Repende de cual registro R/W)	0	111	000011	0 0	0	11 X (Rej	pende de cual X (Repende de cu jistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn × Rm	MUL Rd, Rn, Rm	MUL (00000101)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXX : No importa	Extra(No relevante)		(Repende de cual registro R/W) O0/01 (Repende de cual registro R/W)	0	111	000100	0	0	11 X (Rej	pende de cual X (Repende de cu gistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn ÷ Rm	DIV Rd, Rn, Rm	DIV (00000110)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante)	X (I	(Repende de cual registro R/W) O0/01 (Repende de cual registro R/W)	0	111	000101	0	0	11 X (Rej	pende de cual X (Repende de cu gistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn & Rm	AND Rd, Rn, Rm	AND (00000111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXX : No importa	Extra(No relevante)	X (I	(Repende de cual registro R/W) O0/01 (Repende de cual registro R/W)	0	111	000110	0	0	11 X (Rej	pende de cual X (Repende de cu gistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn Rm	ORR Rd, Rn, Rm	ORR (00001000)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante)		(Repende de cual registro R/W) O0/01 (Repende de cual registro R/W)	0	111	000111	0	0	11 X (Rei	pende de cual X (Repende de cu gistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn ^ Rm	EOR Rd, Rn, Rm	EOR (00001001)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante)		(Repende de cual registro R/W) 00/01 (Repende de cual registro R/W)	0	111	001000	0	0		pende de cual X (Repende de cu pistro R/W) registro R/W)		0	0	0	0 0	0
Rd ← Rn & ~Rm	BIC Rd, Rn, Rm	BIC (00001010)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante)		(Repende de cual registro R/W) 00/01 (Repende de cual registro R/W)	0	111	001001	0	0		pende de cual X (Repende de cu registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn << Rm	LSL Rd, Rn, Rm	LSL (00001011)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXX : No importa	Extra(No relevante)	X (I	(Repende de cual registro R/W) 00/01 (Repende de cual registro R/W)	0	111	001010	0	0	11 X (Rep	pende de cual X (Repende de cu gistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn >> Rm	LSR Rd, Rn, Rm	LSR (00001100)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXX : No importa	Extra(No relevante)	X (I	(Repende de cual registro R/W) 00/01 (Repende de cual registro R/W)	0	111	001011	0	0		pende de cual X (Repende de cu jistro R/W) registro R/W)		0	0	0	0 0	0
Rd ← Rn >> Rm	ASR Rd, Rn, Rm	ASR (00001101)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXX : No importa	Extra(No relevante)		(Repende de cual registro R/W) 00/01 (Repende de cual registro R/W) 0	0	111	001100	0 0	0	11 X (Rep	pende de cual X (Repende de cu jistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← ROR(Rn, Rm)	ROR Rd, Rn, Rm	ROR (00001110)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (010X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante)	X (F	(Repende de cual registro R/W) 00/01 (Repende de cual registro R/W)	0	111	001101	0	0		pende de cual X (Repende de cu gistro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn + Im	ADDI Rd, Rn, #?	ADDI (00001111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)		(Repende de cual registro R/W) XX (No importa, no lo tomaremos) 1	0	111	000000	0	0	11 X (Repres	pende de cual X (Repende de cu istro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn - Im	SUBI Rd, Rn, #?	SUBI (00010000)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)		(Repende de cual registro R/W) XX (No importa, no lo tomaremos) 1	0	111	000001	0	0		ende de cual X (Repende de cual registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn + Im + C	ADCI Rd, Rn, #?	ADCI (00010001)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)	X (R	(Repende de cual registro R/W) XX (No importa, no lo tomaremos) 1	0	111	000010	0	0	11 X (Rep	ende de cual X (Repende de cual istro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn - Im - !C	SBCI Rd, Rn, #?	SBCI (00010010)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)		(Repende de cual registro R/W) XX (No importa, no lo tomaremos)	0	111	000011	0	0	11 X (Rep	ende de cual X (Repende de cual istro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn × Im	MULI Rd, Rn, #?	MULI (00010011)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)	X (R	(Repende de cual registro R/W) XX (No importa, no lo tomaremos)	0	111	000100	0	0		ende de cual X (Repende de cual registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn ÷ Im	DIVI Rd, Rn, #?	DIVI (00010100)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)		(Repende de cual registro R/W) XX (No importa, no lo tomaremos)	0	111	000101	0	0	11 X (Represent	rende de cual X (Repende de cual istro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn & Im	ANDI Rd, Rn, #?	ANDI (00010101)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC : -4852	Extra(No relevante)	X (R	(Repende de cual registro R/W) XX (No importa, no lo tomaremos)	0	111	000110	0	0	11 X (Rep	ende de cual X (Repende de cual istro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn Im	ORRI Rd, Rn, #?	ORRI (00010110)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)		(Repende de cual registro R/W) XX (No importa, no lo tomaremos)	0	111	000111	0	0	11 X (Rep	ende de cual X (Repende de cual istro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn ^ Im	EORI Rd, Rn, #?	EORI (00010111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)	X (R	(Repende de cual registro R/W) XX (No importa, no lo tomaremos) 1	0	111	001000	0	0	11 X (Rep	ende de cual X (Repende de cual registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn & ~Im	BICI Rd, Rn, #?	BICI (00011000)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)	X (R	(Repende de cual registro R/W) XX (No importa, no lo tomaremos)	0	111	001001	0	0	11 X (Rep	ende de cual X (Repende de cua registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn << Im	LSLI Rd, Rn, #?	LSLI (00011001)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC : -4852	Extra(No relevante)	X (R	(Repende de cual registro R/W) XX (No importa, no lo tomaremos)	0	111	001010	0	0	11 X (Rep	ende de cual X (Repende de cua istro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn >> Im	LSRI Rd, Rn, #?	LSRI (00011010)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)	X (R	(Repende de cual registro R/W) XX (No importa, no lo tomaremos) 1	0	111	001011	0	0		ende de cual X (Repende de cua istro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rn >> Im	ASRI Rd, Rn, #?	ASRI (00011011)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)	X (R	(Repende de cual registro R/W) XX (No importa, no lo tomaremos) 1	0	111	001100	0	0	11 X (Rep	ende de cual X (Repende de cua istro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← ROR(Rn, Im)	RORI Rd, Rn, #?	RORI (00011100)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (01XX)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)	X (R	(Repende de cual xX (No importa, no registro R/W) 1	0	111	001101	0	0	11 X (Rep	ende de cual X (Repende de cua istro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← Rm	MOV Rd, Rm	MOV (00011101)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0X0X)	Rd/Wd (destino) Ej = (0001):R1	Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xXXXXXXXX : No importa	Extra(No relevante)	X (N	(No importa, no lo tomaremos) 00/01 (Repende de cual registro R/W) 0	0	111	001110	1 0	0	11 X (Rep	ende de cual X (Repende de cual istro R/W) registro R/W)	ual 11	0	0	0	0 0	0
Rd ← ~Rm	MVN Rd, Rm	MVN	Rd/Wd,Rn/Wn,Rm/Wm,G/D	Rd/Wd (destino)	Rn/Wn (A)	Rm/Wm (B)	Imm						Υ		· · · · · · · · · · · · · · · · · · ·		<u> </u>		<u> </u>	Y		0 0	
	IVIVIVIQ, IVIII	(00011110)	(0X0X)	Fi = (0001):R1	Fi = (XXXX): No importa	Fi = (0100):R4	Fi = 0xXXXXXXXX : No importa	Extra(No relevante)	X (N	(No importa, no lo 00/01 (Repende de tomaremos) cual registro R/W)	0	111	001111	1 0	0	11 X (Rep	ende de cual X (Repende de cua istro R/W) registro R/W)	ual 11	0	0	U		0
Rd ← Im	MOVI Rd, #?	(00011110) MOVI (00011111)	(0X0X) Rd/Wd,Rn/Wn,Rm/Wm,G/D	Ej = (0001):R1 Rd/Wd (destino)	Ej = (XXXX): No importa Rn/Wn (A)	Ej = (0100):R4 Rm/Wm (B)	Ej = 0xXXXXXXXX : No importa		X (N	(No importa, no lo XX (No importa, no	0	111	<u> </u>	1 0	0	reg	registro R/W)	<u>''</u>	0	0	0	0 0	0
Rd ← Im Rd ← ~Im		MOVI (00011111) MVNI	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino)	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A)	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B)	Imm Ej = 0xFFFFED0C : -4852	Extra(No relevante)	X (N	(No importa, no lo tomaremos) XX (No importa, no lo lo tomaremos) XX (No importa, no lo XX (No importa, no l	0 0	111	001111	1 0	0 0	11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11	0	0 0	0	0 0	0
Rd ← ~Im	MOVI Rd, #? MVNI Rd, #?	MOVI (00011111) MVNI (00100000)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino)	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A)	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B)	Imm Ej = 0xFFFFED0C: -4852 Imm Ej = 0xFFFFED0C: -4852 Imm	Extra(No relevante) Extra(No relevante)	X (N	(No importa, no lo tomaremos) XX (No importa, no lo lo tomaremos) XX (No importa, no lo tomaremos) 1 XX (No importa, no lo tomaremos) 1	0 0	111	<u> </u>		0 0	11 X (Rep reg	ende de cual x (Repende de cual registro R/W)	ual 11	0 0	0 0	0 0		0 0
Rd ← ~Im Actualiza flags según (Rn - Rm)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm	MOVI (00011111) MVNI (00100000) CMP (00100001)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xFFFFED0C: -4852 Imm Ej = 0xFFFFED0C: -4852 Imm Ej = 0xXXXXXXXXX: No importa	Extra(No relevante) Extra(No relevante) Extra(No relevante)	X (N) X (N) X (N)	(No importa, no lo tomaremos) XX (No importa, no lo lo tomaremos) XX (No importa, no lo XX (No importa, no l		111 111 111	001110			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11	0 0	0 0	0 0		0 0
Rd ← ~Im	MOVI Rd, #? MVNI Rd, #?	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5	Imm Ej = 0xFFFFEDOC : -4852 Imm Ej = 0xFFFFEDOC : -4852 Imm Ej = 0xXXXXXXXXX : No importa Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante) Extra(No relevante) Extra(No relevante) Extra(No relevante)	X (N) X (N) X (N)	(No importa, no lo tomaremos) 1 (Repende de cual registro R/W) 1 1 1 0		111	<u> </u>			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11		0 0 0	0 0 0		
Rd ← ~Im Actualiza flags según (Rn - Rm)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010) CMN (00100011)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa Rd/Wd (destino) Ej = (XXXX): No importa Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xFFFFEDOC : -4852 Imm Ej = 0xFFFFEDOC : -4852 Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante) Extra(No relevante) Extra(No relevante) Extra(No relevante) Extra(No relevante)	X (N) X (N) X (N) X (R) X (R) X (R) X (R)	tomaremos) cual registro R/W) (No importa, no lo tomaremos) 1 (Repende de cual registro R/W) 1 1 1 0 (Repende de cual registro R/W) 0 (Repende de cual registro R/W) 0 0 0 0 0 0 0 0 0 0 0 0 0		111	001110			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11		0 0 0	0 0 0		
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P?	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa Rd/Wd (destino) Ej = (XXXX): No importa Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A)	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B)	Imm Ej = 0xFFFFED0C: -4852 Imm Ej = 0xFFFFED0C: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xXXXXXXXXX: No importa	Extra(No relevante) Extra(No relevante) Extra(No relevante) Extra(No relevante) Extra(No relevante)	X (N) X (N) X (N) X (R) X (R) X (R) X (R)	(No importa, no lo tomaremos) 1 (Repende de cual registro R/W) 1 1 1 0		111 111 111	001110 001111 010000			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11		0 0 0 0 0 0	0 0 0 0 0		
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010) CMN (00100011)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xFFFFEDOC : -4852 Imm Ej = 0xFFFFEDOC : -4852 Imm Ej = 0xXXXXXXXXX : No importa Imm Ej = 0xXXXXXXXXX : No importa Imm Ej = 0xXXXXXXXXX : No importa	Extra(No relevante) Extra(No relevante) Extra(No relevante) Extra(No relevante) Extra(No relevante) Extra(No relevante)	X (N) X (N) X (N) X (N) X (R)	tomaremos) cual registro R/W) (No importa, no lo tomaremos) 1 (Repende de cual registro R/W) 1 1 1 0 (Repende de cual registro R/W) 0 (Repende de cual registro R/W) 0 0 0 0 0 0 0 0 0 0 0 0 0		111 111 111	001110 001111 010000			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~lm Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010) CMN (00100011) TST (00100100)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa	Extra(No relevante)	X (N) X	tomaremos) cual registro R/W) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1 1 1 1 1 1 0 (Repende de cual registro R/W) (Repende de cual registro R/W) (Repende de cual registro R/W) 0 (Repende de cual registro R/W)		111	001110 001111 010000			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn ^ Rm)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010) TST (00100100) TEQ (00100101) CMPI	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa	Extra(No relevante)	X (N) X (N) X (N) X (N) X (R)	(No importa, no lo tomaremos) (Repende de cual registro R/W) 1		111 111 111 111 111 111	001110 001111 010000 010000 010001 010011			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn ^ Rm) Actualiza flags según (Rn - Im)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #?	MOVI (00011111) MVNI (00100000) CMP (0010001) CMPS (00100010) TST (00100100) TEQ (00100101) CMPI (00100110) CMNI (00100111) TSTI	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC : -4852 Imm Ej = 0xXXXXXXXXX : No importa Imm Ej = 0xFFFFEDOC : -4852 Imm Ej = 0xFFFFEDOC : -4852 Imm	Extra(No relevante)	X (R	(No importa, no lo tomaremos) (Repende de cual registro R/W)		111 111 111 111 111 111	001110 001111 010000 010001 010011 010000			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn ^ Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #? CMNI Rn, #?	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010) CMN (00100011) TST (00100100) CMPI (00100110) CMNI (00100111) TSTI (00100111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX : No importa Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (R	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1 (Repende de cual registro R/W) (Repende de cual registro R/W) 1 (Repende de cual registro R/W) 1 (Repende de cual XX (No importa, no lo tomaremos) 1 (Repende de cual registro R/W) 1 (Repende de cual XX (No importa, no lo tomaremos)		111 111 111 111 111 111	001110 001000 010000 010001 010001 010000 010001			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn ^ Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #? CMNI Rn, #?	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010) TST (00100100) CMPI (00100110) CMNI (00100111) TSTI (00100100) TEQI (00101000)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N) X (N) X (N) X (N) X (R)	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1		111 111 111 111 111 111 111 111 111 11	001110 010000 010000 010001 010000 010001 010010			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn & Im) PC ← etiqueta	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta	MOVI (00011111) MVNI (00100000) CMP (00100010) CMNS (00100010) TST (00100100) CMPI (00100110) CMNI (00100111) TSTI (00101000) TEQI (00101000) TEQI (00101001) BEQ	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (R	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1 1 (Repende de cual registro R/W) 1 1 (Repende de cual registro R/W) 1 1 1 1 1 1 1 1 1 1 1 1 1		111 111 111 111 111 111 111 111 111 11	001110 001011 010000 010001 010001 010001 010010			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn ^ Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta	MOVI (00011111) MVNI (00100000) CMP (00100010) CMNS (00100011) TST (00100100) CMPI (00100101) CMPI (00100111) TSTI (00100111) TSTI (00100100) TEQI (00101000) BEQ (00101011)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N) X	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1 (Repende de cual registro R/W)		111 111 111 111 111 111 111 111 111 11	001110 010000 010000 010001 010000 010001 010010			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn & Im) PC ← etiqueta	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010) TST (00100100) CMPI (00100101) CMPI (00100110) CMNI (00100111) TSTI (00101000) BEQ (00101010) BEQ (00101010)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N)	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1		111 111 111 111 111 111 111 111 111 11	001110 001011 010000 010001 010001 010001 010010			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn ^ Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010) CMN (00100011) TST (00100100) CMPI (00100110) CMNI (00100111) TSTI (00101000) TEQI (00101000) TEQI (00101001) BREQ (00101011)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N) X	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1		111 111 111 111 111 111 111 111 111 11	001110 010000 010000 010001 010001 010001 010010			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn & Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm CMPI Rn, #? CMNI Rn, #? TEQI Rn, #? Betiqueta BEQ etiqueta BNE etiqueta BLTetiqueta	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010) CMN (00100011) TST (00100100) CMPI (00100101) CMNI (00100111) TSTI (00100100) TEQI (00101000) TEQI (00101000) BEQ (00101010) BREQ (00101011)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N) X (N	(Repende de cual registro R/W) 1 (Ro importa, no lo tomaremos) 1 (Ro importa, no lo tomaremos) 1 (No importa, no lo tomaremos)		111 111 111 111 111 111 111 111 111 11	001110 010000 010000 010001 010001 010001 010010			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn * Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm CMPI Rn, #? CMNI Rn, #? TEQI Rn, #? Betiqueta BEQ etiqueta BNE etiqueta BLTetiqueta	MOVI (00011111) MVNI (00100000) CMP (00100001) CMPS (00100010) TST (00100100) CMPI (00100101) CMNI (0010011) CMNI (0010011) TSTI (00100100) BEQ (00101010) BEQ (00101011) BNE (00101100) BLT (00101101) BGT (00101110)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N) X (N	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1 (Repende de cual registro R/W) 1 (Repende de cual registro R/W) 1 (Ro importa, no lo lo tomaremos) 1 (Ro importa, no lo tomaremos) 1 (No importa, no lo tomaremos) 1		111 111 111 111 111 111 111 111 111 11	001110 010000 010000 010001 010010 010010			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn * Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BNE etiqueta BUTetiqueta SWI	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N) X (N	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1 (Repende de cual registro R/W) 1 (Repende de cual registro R/W) 1 (Ro importa, no lo lo tomaremos) 1 (Ro importa, no lo tomaremos) 1 (No importa, no lo tomaremos) 1		111 111 111 111 111 111 111 111 111 11	001110 0010000 010000 010001 010010 010001 010010			11 X (Rep reg	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	ual 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn * Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TEQ Rn, Rm CMPI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BLTetiqueta BGT etiqueta SWI NOP	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N)	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1		111 111 111 111 111 111 111 111 111 11	001110 010000 010000 010001 010010 010010			11	ende de cual istro R/W) ende de cual istro R/W) ende de cual istro R/W) o	11 11 11 11 11 11 11 11 11 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn * Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BNE etiqueta BUTetiqueta SWI	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N)	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1 (Repende de cual registro R/W) 1 (Repende de cual registro R/W) 1 (Ro importa, no lo lo tomaremos) 1 (Ro importa, no lo tomaremos) 1 (No importa, no lo tomaremos) 1		111 111 111 111 111 111 111 111 111 11	001110 0010000 010000 010001 010010 010001 010010			11	ende de cual X (Repende de cua registro R/W) ende de cual X (Repende de cua registro R/W)	11 11 11 11 11 11 11 11 11 11					
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn * Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TEQ Rn, Rm CMPI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BLTetiqueta BGT etiqueta SWI NOP	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N)	(No importa, no lo tomaremos) (No importa, no lo tomaremos) (Repende de cual registro R/W) 1		111 111 111 111 111 111 111 111 111 11	001110 0010000 010000 010001 010010 010001 010010			11	ende de cual istro R/W) O O O O O O O O O O O O O O O O O O O	11 11 11 11 11 11 11 11 11 11			O O O O O O O O O O O O O O O O O O O	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn * Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn * Im) Actualiza flags según (Rn * Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8 Rd ← Mem?[Rn + offset]	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BLTetiqueta BLTetiqueta BGT etiqueta SWI NOP LDR Rd, G/D[Rn, #offset]	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX)	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N)	(No importa, no lo tomaremos) (Repende de cual registro R/W) 1 1 1 1 1 1 1 1 1 1 1 1 1		111 111 111 111 111 111 111 111 111 11	001110 001011 010000 010001 010011 010010			11	ende de cual istro R/W) O O O O O O O O O O O O O O O O O O O	11 11 11 11 11 11 11 11 11 11			O O O O O O O O O O O O O O O O O O O	O O O O O O O O O O O O O O O O O O O	
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn - Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn & Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8 Rd ← Mem?[Rn + offset] ← Rm	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BNE etiqueta BUTetiqueta SWI NOP LDR Rd, G/D[Rn, #offset] STR Rm, G/D[Rn, #offset]	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X0): General Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X0): General Memory	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (N) X	No importa, no lo tomaremos Cual registro R/W		111 111 111 111 111 111 111 111 111 11	001110 001011 010000 010001 010010 010001 010001 010001 010001 010001 010001 000000			11	istro R/W) ende de cual istro R/W) ende de cual istro R/W) ende de cual istro R/W) 0 0 0 0 0 0 0 0 0 0 0 0 0	11 11 11 11 11 11 11 11 11 11		0 ×(O Memoria G/D) Memoria G/D) Depende de cual X (Depende de cual X (Dep	O O O O O O O O O O O O O O O O O O O	
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn - Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn & Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8 Rd ← Mem?[Rn + offset] (byte)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BNE etiqueta BNE etiqueta BUTETIQUETA BUTETIQ	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X0): General Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (R) X	tomaremos) cual registro R/W) (No importa, no lo tomaremos) (No importa, no lo tomaremos) 1 (Repende de cual registro R/W) 1 1 1 1 1 1 1 1 0 (Repende de cual registro R/W) 1 1 1 1 1 0 (Repende de cual registro R/W) 1 (Repende de cual registro R/W) 1 (Repende de cual registro R/W) (Repende de cual registro R/W) 1 (No importa, no lo tomaremos) 1 1 1 1 1 1 1 1 1 1 1 1 1		111 111 111 111 111 111 111 111 111 11	001110 0010111 010000 010001 010011 010010		1	11	istro R/W) ende de cual istro R/W) ende de cual istro R/W) ende de cual istro R/W) 0 0 0 0 0 0 0 0 0 0 0 0 0	11 11 11 11 11 11 11 11 11 11		0 ×(O memoria G/D) memo	0 0	
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn & Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8 Rd ← Mem?[Rn + offset] Mem?[Rn + offset] ← Rm Rd ← Mem?[Rn + offset] (byte) Mem?[Rn + offset] ← Rm (byte) Console ← int(Mem?[Rn + offset])	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BEQ etiqueta BUT etiqueta BUT etiqueta BUT etiqueta SWI NOP LDR Rd, G/D[Rn, #offset] STR Rm, G/D[Rn, #offset] PRINTI G/D[Rn, #offset]	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X0): General Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): General Memory	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXX: No importa Imm Ej = 0xXXXXXXXX: No importa Imm Ej = 0xXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (R)	tomaremos) cual registro R/W) (No importa, no lo tomaremos) (No importa, no lo tomaremos) 1 (Repende de cual registro R/W) 1 1 11 11 0 (Repende de cual registro R/W) 1 11 11 0 (Repende de cual registro R/W) 1 (Repende de cual registro R/W) (Repende de cual registro R/W) 1 (No importa, no lo tomaremos) 1 (No importa, no lo tomaremos) 1 1 1 1 1 1 1 1 1 1 1 1 1		111 111 111 111 111 111 111 111 111 11	001110 001011 010000 010001 010010 010001 010001 010001 010001 010001 010001 000000		1 0	11	istro R/W) ende de cual istro R/W) ende de cual istro R/W) ende de cual istro R/W) 0 0 0 0 0 0 0 0 0 0 0 0 0	11 11 11 11 11 11 11 11 11 11		0 ×(O memoria G/D) memo	0 0	
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn ^ Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn * Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn & Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8 Rd ← Mem?[Rn + offset] Mem?[Rn + offset] ← Rm Rd ← Mem?[Rn + offset] (byte) Mem?[Rn + offset] ← Rm (byte)	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BNE etiqueta BNE etiqueta BUTETIQUETA BUTETIQ	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X0): General Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (R)	tomaremos) cual registro R/W) (No importa, no lo tomaremos) (No importa, no lo tomaremos) 1 (Repende de cual registro R/W) 1 1 1 1 1 1 1 1 0 (Repende de cual registro R/W) 1 1 1 1 1 0 (Repende de cual registro R/W) 1 (Repende de cual registro R/W) 1 (Repende de cual registro R/W) (Repende de cual registro R/W) 1 (No importa, no lo tomaremos) 1 1 1 1 1 1 1 1 1 1 1 1 1		111 111 111 111 111 111 111 111 111 11	001110 0010111 010000 010001 010011 010010		1 0	11	istro R/W) ende de cual istro R/W) ende de cual istro R/W) ende de cual istro R/W) 0 0 0 0 0 0 0 0 0 0 0 0 0	11 11 11 11 11 11 11 11 11 11		0 ×(O memoria G/D) memo	0 0	
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn - Im) Actualiza flags según (Rn + Im) Actualiza flags según (Rn & Im) Actualiza flags según (Rn & Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8 Rd ← Mem?[Rn + offset] Mem?[Rn + offset] ← Rm Rd ← Mem?[Rn + offset] (byte) Mem?[Rn + offset] ← Rm (byte) Console ← int(Mem?[Rn + offset])	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BEQ etiqueta BUT etiqueta BUT etiqueta BUT etiqueta SWI NOP LDR Rd, G/D[Rn, #offset] STR Rm, G/D[Rn, #offset] PRINTI G/D[Rn, #offset]	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X0): General Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (R r x X (R r x X	tomaremos) cual registro R/W) (No importa, no lo tomaremos) (No importa, no lo tomaremos) 1 (Repende de cual registro R/W) 1 1 11 11 0 (Repende de cual registro R/W) 1 11 11 0 (Repende de cual registro R/W) 1 (Repende de cual registro R/W) (Repende de cual registro R/W) 1 (No importa, no lo tomaremos) 1 (No importa, no lo tomaremos) 1 1 1 1 1 1 1 1 1 1 1 1 1		111 111 111 111 111 111 111 111 111 11	001110 0010111 010000 010001 010011 010010			11	istro R/W) ende de cual istro R/W) ende de cual istro R/W) ende de cual istro R/W) 0 0 0 0 0 0 0 0 0 0 0 0 0	11 11 11 11 11 11 11 11 11 11		0 ×(O memoria G/D) memo	0 0	
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn ^ Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn * Im) Actualiza flags según (Rn & Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8 Rd ← Mem?[Rn + offset] Mem?[Rn + offset] ← Rm Rd ← Mem?[Rn + offset] (byte) Mem?[Rn + offset] ← Rm (byte) Console ← ASCII(Mem?[Rn + offset])	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BNE etiqueta BNE etiqueta BUTetiqueta BUTETIQUETA BUTETIQUETA BUTETIQUETA BUTETIQUETA BUTETIQUETA BUTETIQUETA BUTETIQUETA BUTETIQUETA SWI NOP LDR Rd, G/D[Rn, #offset] LDRB Rd, G/D[Rn, #offset] PRINTS G/D[Rn, #offset]	MOVI (00011111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X0): General Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante)	X (R)	Commercia Comm		111 111 111 111 111 111 111 111 111 11	001110 010000 010000 010001 010010 010001 010001 010010			11	istro R/W) ende de cual istro R/W) ende de cual istro R/W) ende de cual istro R/W) 0 0 0 0 0 0 0 0 0 0 0 0 0	11 11 11 11 11 11 11 11 11 11		0 ×(O memoria G/D) memo	0 0	
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn & Rm) Actualiza flags según (Rn - Im) Actualiza flags según (Rn & Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8 Rd ← Mem?[Rn + offset] Mem?[Rn + offset] ← Rm Rd ← Mem?[Rn + offset] (byte) Console ← int(Mem?[Rn + offset]) Console ← Binary(Mem?[Rn + offset])	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BNE etiqueta BNE etiqueta BNE etiqueta SWI NOP LDR Rd, G/D[Rn, #offset] STRB Rm, G/D[Rn, #offset] PRINTI G/D[Rn, #offset] PRINTI G/D[Rn, #offset]	MOVI (0001111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (XXXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X0): General Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X0): General Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X0): General Memory Rd/Wd,Rn/Wn,Rm/Wm,G/D Ej=(01X1): Dinamic Memory	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0101):R5 Rn/Wn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante) Extra(No relevante)	X (R) X	Itemaremos Cual registro R/N/)		111 111 111 111 111 111 111 111 111 11	001110 010000 010000 010001 010010 010001 010001 010010			11	istro R/W) ende de cual istro R/W) ende de cual istro R/W) ende de cual istro R/W) 0 0 0 0 0 0 0 0 0 0 0 0 0	11 11 11 11 11 11 11 11 11 11		0 ×(O memoria G/D) memo	0 0	
Rd ← ~Im Actualiza flags según (Rn - Rm) Actualiza flags (y extra) según (Rn - P?) Actualiza flags según (Rn + Rm) Actualiza flags según (Rn - Rm) Actualiza flags según (Rn - Im) PC ← etiqueta Si Z=1, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0, entonces PC ← etiqueta Si Z=0 y N=V, entonces PC ← etiqueta Generar interrupción software PC ← PC + 8 Rd ← Mem?[Rn + offset] Mem?[Rn + offset] ← Rm Rd ← Mem?[Rn + offset] (byte) Console ← int(Mem?[Rn + offset]) Console ← ASCII(Mem?[Rn + offset]) Console ← Binary(Mem?[Rn + offset])	MOVI Rd, #? MVNI Rd, #? CMP Rn, Rm CMPS Rn, P? CMN Rn, Rm TST Rn, Rm TEQ Rn, Rm CMPI Rn, #? CMNI Rn, #? TSTI Rn, #? TEQI Rn, #? B etiqueta BEQ etiqueta BNE etiqueta BUT etiqueta BUT etiqueta SWI NOP LDR Rd, G/D[Rn, #offset] STR Rm, G/D[Rn, #offset] PRINTI G/D[Rn, #offset] PRINTS G/D[Rn, #offset] LOGOUT	MOVI (0001111)	Rd/Wd,Rn/Wn,Rm/Wm,G/D (0XXX) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X10X) Rd/Wd,Rn/Wn,Rm/Wm,G/D (X1XX) Rd/Wd,Rn/Wn,Rm/Wm,G/D	Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (0001):R1 Rd/Wd (destino) Ej = (XXXX): No importa	Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (XXXX): No importa Rn/Wn (A) Ej = (0010):W3 Rn (A) Ej = (0010):W3 Rn/Wn (A) Ej = (XXXX): No importa	Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (0100):R4 P? (B) Ej = (0100):P5 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (0100):R4 Rm/Wm (B) Ej = (XXXX): No importa Rm/Wm (B) Ej = (XXXX): No importa	Imm Ej = 0xFFFFEDOC: -4852 Imm Ej = 0xXXXXXXXXX: No importa Imm Ej = 0xFFFFEDOC: -4852	Extra(No relevante) Extra(No relevante)	X (R) X	Itemaremos Cual registro R/N/)		111 111 111 111 111 111 111 111 111 11	001110 001011 010000 010001 010001 010001 010001 010001 010001 000000			11	istro R/W) ende de cual istro R/W) ende de cual istro R/W) ende de cual istro R/W) 0 0 0 0 0 0 0 0 0 0 0 0 0	11 11 11 11 11 11 11 11 11 11		0 ×(O memoria G/D) memo	0 0	



