

Reference Sheet

Operaciones aritméticas		
ADD Rd, Rn, Rm	Rd ← Rn + Rm	Suma entre registros
ADDS Rd, Rn, k?.?	Rd ← Rn + k?.?	Suma entre registro y bloque de la ‘Vault’
ADDI Rd, Rn, #?	Rd ← Rn + Im	Suma con inmediato
SUB Rd, Rn, Rm	Rd ← Rn - Rm	Resta entre registros
SUBI Rd, Rn, #?	Rd ← Rn - Im	Resta con inmediato
ADC Rd, Rn, Rm	Rd ← Rn + Rm + C	Suma con acarreo entre registros
ADCI Rd, Rn, #?	Rd ← Rn + Im + C	Suma con acarreo con inmediato
SBC Rd, Rn, Rm	Rd ← Rn - Rm - !C	Resta con acarreo entre registros
SBCI Rd, Rn, #?	Rd ← Rn - Im - !C	Resta con acarreo con inmediato
MUL Rd, Rn, Rm	Rd ← Rn × Rm	Multiplicación entre registros
MULI Rd, Rn, #?	Rd ← Rn × Im	Multiplicación con inmediato
DIV Rd, Rn, Rm	Rd ← Rn ÷ Rm	División entre registros
DIVI Rd, Rn, #?	Rd ← Rn ÷ Im	División con inmediato

Operaciones lógicas		
AND Rd, Rn, Rm	Rd ← Rn & Rm	AND Bit a bit entre registros
ANDI Rd, Rn, #?	Rd ← Rn & Im	AND Bit a bit con inmediato
ORR Rd, Rn, Rm	Rd ← Rn Rm	OR Bit a bit entre registros
ORRI Rd, Rn, #?	Rd ← Rn Im	OR Bit a bit con inmediato
EOR Rd, Rn, Rm	Rd ← Rn ^ Rm	XOR Bit a bit entre registros
EORI Rd, Rn, #?	Rd ← Rn ^ Im	XOR Bit a bit con inmediato
BIC Rd, Rn, Rm	Rd ← Rn & ~Rm	Bit Clear (AND con NOT) entre registros
BICI Rd, Rn, #?	Rd ← Rn & ~Im	Bit Clear (AND con NOT) con inmediato R
MVN Rd, Rm	Rd ← ~Rm	Mover NOT entre registros
MVNI Rd, #?	Rd ← ~Im	Mover NOT con inmediato

Desplazamientos y rotaciones		
LSL Rd, Rn, Rm	Rd ← Rn « Rm	Desplazamiento lógico a la izquierda con val de registro
LSLI Rd, Rn, #?	Rd ← Rn « Im	Desplazamiento lógico a la izquierda con val de inmediato
LSR Rd, Rn, Rm	Rd ← Rn » Rm	Desplazamiento lógico a la derecha con val de registro
LSRI Rd, Rn, #?	Rd ← Rn » Im	Desplazamiento lógico a la derecha con val de inmediato
ASR Rd, Rn, Rm	Rd ← signed(Rn) » (Rm & 0x1F)	Desplazamiento aritmético con signo, usando solo los 5 bits bajos de Rm
ASRI Rd, Rn, #?	Rd ← Rn » Im	Desplazamiento aritmético a la derecha con val de inmediato
ROR Rd, Rn, Rm	Rd ← ROR(Rn, Rm)	Rotación a la derecha con val de registro
RORI Rd, Rn, #?	Rd ← ROR(Rn, Src2)	Rotación a la derecha con val de inmediato

Operaciones de movimiento de datos		
MOV Rd, Rm	Rd ← Rm	Mover con registros
MOVI Rd, #?	Rd ← Im	Mover con inmediato
LDR Rd, G/D[Rn, #offset]	Rd ← MemG/D[Rn + offset]	Cargar de memoria general
STR Rd, G/D[Rn, #offset]	MemG/D[Rn + offset] ← Rd	Guardar en memoria
LDRB Rd, G/D[Rn, #offset]	Rd ← MemG/D[Rn + offset] (byte)	Cargar byte de memoria
STRB Rd, G/D[Rn, #offset]	MemG/D[Rn + offset] ← Rd (byte)	Guardar byte en memoria

Operaciones de comparación		
CMP Rn, Rm	Actualiza flags según (Rn - Rm)	Comparación entre registros
CMPI Rn, #?	Actualiza flags según (Rn - Im)	Comparación entre registro e inmediato
CMN Rn, Rm	Actualiza flags según (Rn + Rm)	Comparar negativo entre registros
CMNI Rn, #?	Actualiza flags según (Rn + Im)	Comparar negativo entre registro e inmediato
CMPS Rn, P?	Actualiza flags (y extra) según (Rn - P?)	Actualiza flags (y extra) según (Rn - P?)
TST Rn, Rm	Actualiza flags según (Rn & Rm)	Test de bits entre registros
TSTI Rn, #?		Test de bits entre registro e inmediato
TEQ Rn, Rm	Actualiza flags según (Rn ^ Rm)	Test igualdad entre registros
TEQI Rn, #?	Actualiza flags según (Rn ^ Im)	Test igualdad entre registro e inmediato

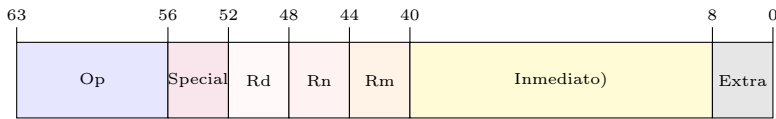
Operaciones de bifurcación		
B etiqueta	PC ← etiqueta	Salto incondicional
BEQ etiqueta	Si Z=1, entonces PC ← etiqueta	Salto si igual (Z=1)
BNE etiqueta	Si Z=0, entonces PC ← etiqueta	Salto si no igual (Z=0)
BGT etiqueta	Si Z=0 y N=V, entonces PC ← etiqueta	Salto si mayor que (Z=0 AND N=V)
BLT etiqueta	Si N!=V, entonces PC ← etiqueta	Salto si menor que (N!=V)

Operaciones Login		
LOGOUT		Comando para intentar hacer un logout del sistema de seguridad del computador.
AUTHCMP		Comparar para login con un val de memoria de login, R1-8 == P[0 - 12]

Instrucciones de compilador	
TEA #0, #val	Variables de uso para cada ronda
TEAENC #1, k?(0/1/2/3)	Realiza la suma del delta y parte del primer termino v1
TEAENC #2	Calculo (v1 + sum) y le hace XOR con el acumulado en w8.
TEAENC #3, k?(0/1/2/3)	Calcula (v1 » 5) + k1, luego XOR con lo ya presente en w8.
TEAENC #4, k?(0/1/2/3)	Calcula el nuevo valor de v0 e inicial los del v1
TEAENC #5	Calcula el nuevo valor de v0 e inicial los del v1
TEAENC #6, k?(0/1/2/3)	Calculo del XOR con (v0 » 5) + k3
TEAENC #7	Calculo final de v1 y revisión del bucle
TEAD #1, k?(0/1/2/3)	Realiza las operaciones 1, 2 y 3 del proceso de desencriptacion
TEAD #2, k?(0/1/2/3)	Realiza las operaciones 4, 5 y 6 del proceso de desencriptacion
TEAD #3	Realiza las operaciones 7, 8 y 9 del proceso de desencriptacion
TEAD #4, k?(0/1/2/3)	Realiza las operaciones 10, 11 y 12 del proceso de desencriptacion
TEAD #5, k?(0/1/2/3)	Realiza las operaciones 13, 14 y 15 del proceso de desencriptacion
TEAD #6	Realiza las operaciones 16, 17 y 18 del proceso de desencriptacion
TEAD #7	Realiza las ultimas operaciones del proceso de desencriptacion

Operaciones especiales		
SWI	Generar interrupción software	Interrupción de software
NOP	PC ← PC + 8	No operación
PRINTI G/D[Rn, #offset]	Console ← int(Mem?[Rn + offset])	Imprimir entero desde memoria general
PRINTS G/D[Rn, offset]	Console ← ASCII(Mem?[Rn + #offset])	Imprimir ASCII desde memoria (rango) general
PRINTB G/D[Rn, #offset]	Console ← Binary(Mem?[Rn + offset])	Imprimir binario desde memoria general
STRK k?.? #?	(k?.?) Vault[#? + #?] ← Im	Guarda 32bits de una llave, en una de sus 4 secciones (palabras)
STRPASS P? ?	(P?) Password[?] ← Im	Guarda 32bits de una contraseña, en una de sus 8 secciones (palabras)

Formato general de las instrucciones



Segmentos de las instrucciones		
CAMPO	BITS	DESCRIPCIÓN
Op	63-56	Tipo de operación
Special	55-52	Tipo de registro y tipo de memoria
Rd	51-48	Registro destino
Rn	47-44	Registro fuente
Inm	39-8	Inmediato (signed en complemento a 2)
Extra	7-0	Reservado para futuro

Codificación de Rd y Rn

Código	Registro
0000	R0 / W1
0001	R1 / W2
0010	R2 / W3
0011	R3 / W4
0100	R4 / W5
0101	R5 / W6
0110	R6 / W7
0111	R7 / W8
1000	R8 / W9
1001	R9 / d0
1010	R10 / Inf
1011	R11 / Inf
1100	R12 / Inf
1101	R13 / Inf
1110	R14 / Inf
1111	R15 / Inf

Codificación de Special

Código	Formato
0000	Rd / Rn / Rm / G
0001	Rd / Rn / Rm / D
0010	Rd / Rn / Wm / G
0011	Rd / Rn / Wm / D
0100	Rd / Wn / Rm / G
0101	Rd / Wn / Rm / D
0110	Rd / Wn / Wm / G
0111	Rd / Wn / Wm / D
1000	Wd / Rn / Rm / G
1001	Wd / Rn / Rm / D
1010	Wd / Rn / Wm / G
1011	Wd / Rn / Wm / D
1100	Wd / Wn / Rm / G
1101	Wd / Wn / Rm / D
1110	Wd / Wn / Wm / G
1111	Wd / Wn / Wm / D

Codificación de Rm

Código	Registro
0000	R0 / W1 / P1 / k0.0
0001	R1 / W2 / P2 / k0.1
0010	R2 / W3 / P3 / k0.2
0011	R3 / W4 / P4 / k0.3
0100	R4 / W5 / P5 / k1.0
0101	R5 / W6 / P6 / k1.1
0110	R6 / W7 / P7 / k1.2
0111	R7 / W8 / P8 / k1.3
1000	R8 / W9 / Inf / k2.0
1001	R9 / d0 / Inf / k2.1
1010	R10 / Inf / Inf / k2.2
1011	R11 / Inf / Inf / k2.3
1100	R12 / Inf / Inf / k3.0
1101	R13 / Inf / Inf / k3.1
10110	R14 / Inf / Inf / k3.2
1111	R15 / Inf / Inf / k3.3

Codificación de instrucciones (Op)

Instrucción	Código	Instrucción	Código
ADD	00000000	MOVI	00011111
ADDS	00000001	MVNI	00100000
SUB	00000010	CMP	00100001
ADC	00000011	CMPS	00100010
SBC	00000100	CMN	00100011
MUL	00000101	TST	00100100
DIV	00000110	TEQ	00100101
AND	00000111	CMPI	00100110
ORR	00001000	CMNI	00100111
EOR	00001001	TSTI	00101000
BIC	00001010	TEQI	00101001
LSL	00001011	B	00101010
LSR	00001100	BEQ	00101011
ASR	00001101	BNE	00101100
ROR	00001110	BLT	00101101
ADDI	00001111	BGT	00101110
SUBI	00010000	SWI	00101111
ADCI	00010001	NOP	00110000
SBCI	00010010	LDR	00110001
MULI	00010011	STR	00110010
DIVI	00010100	LDRE	00110011
ANDI	00010101	STRB	00110100
ORRI	00010110	PRINTI	00110101
EORI	00010111	PRINTS	00110110
BICI	00011000	PRINTB	00110111
LSLI	00011001	LOGOUT	00111000
LSRI	00011010	STRK	00111001
ASRI	00011011	STRPASS	00111010
RORI	00011100		
MOV	00011101		
MVN	00011110		

Formato de Inmediato (Imm)
Los valores inmediatos se codifican en **complemento a 2 (signed)** sobre 32 bits.
Rango permitido: **-2 147 483 648 a 2 147 483 647**

Formato de ultimos bits
Campo Extra: 00000000 (valor fijo reservado)