

Folha de consulta (não pode ser destacada do documento)

15	14	13	12	11	10	9	8	7	6	5	4	3	0		
0	0	0	[1]	0	0	Opcode		B/\bar{w}		Ad		DST		Formato 2	
0	0	1	Condition			PC offset (10 bit)									Salto
Opcode				SRC				Ad	B/\bar{w}		As		DST		Formato 1

- Formato 1 – Até 3 palavras (Instrução + Source/Dest offset + Destination offset)
- Formato 2 – Até 2 palavras (Instrução + Destination offset)

Instrução (.B/.W)	Args	Op-code	Descrição	Bits de Status			
				V	N	Z	C
Formato1							
MOV	src,dst	0x4	src → dst	-	-	-	-
ADD	src,dst	0x5	src + dst → dst	*	*	*	*
ADDC	src,dst	0x6	src + dst + C → dst	*	*	*	*
SUBC	src,dst	0x7	dst + not(src) + C → dst	*	*	*	*
SUB	src,dst	0x8	dst + not(src) + 1 → dst	*	*	*	*
CMP	src,dst	0x9	dst - src	*	*	*	*
DADD	src,dst	0xA	src + dst + C → dst (decimally)	*	*	*	*
BIT	src,dst	0xB	src .and. dst	0	*	*	\bar{Z}
BIC	src,dst	0xC	not(src) .and. dst → dst	-	-	-	-
BIS	src,dst	0xD	src .or. dst → dst	-	-	-	-
XOR	src,dst	0xE	src .xor. dst → dst	*	*	*	\bar{Z}
AND	src,dst	0xF	src .and. dst → dst	0	*	*	\bar{Z}
Formato 2							
RRC	dst	0x000 ¹	C _{n-1} → MSB →LSB → C	0	*	*	*
RRA	dst	0x010 ¹	MSB → MSB →LSB → C	0	*	*	*
PUSH	dst	0x120 ¹	SP - 2 → SP, src → SP	-	-	-	-
SWPB	dst	0x108 ¹	bit 15...bit 8 ↔ bit 7...bit 0	-	-	-	-
CALL	dst	0x128 ¹	PC+2→ TOS ; #addr →PC	-	-	-	-
RETI	dst	0x130 ¹	Return from interruption	*	*	*	*
SXT	dst	0x118 ¹	Extend sign bits (B/W/A)	0	*	*	\bar{Z}
Jumps							
JNE,JNZ	label	000b ²	Jump if zero is reset	-	-	-	-
JEQ,JZ	label	001b ²	Jump if zero/equal	-	-	-	-
JNC	label	010b ²	Jump if carry is reset	-	-	-	-
JC	label	011b ²	Jump if carry is set	-	-	-	-
JN	label	100b ²	Jump if negative set	-	-	-	-
JGE	label	101b ²	Jump if (N xor V) = 0	-	-	-	-
JL	label	110b ²	Jump if (N xor V) = 1	-	-	-	-
JMP	label	111b ²	Jump unconditionally	-	-	-	-
Instruções emuladas							
ADDC	dst	ADDC #0,dst	Add carry to dst	*	*	*	*
BR	dst	MOV dst,PC	Branch	-	-	-	-
CLR	dst	MOV #0,dst	Clear dst	-	-	-	-
CLR[C/N/Z]		BIC #[1/4/2],SR	Clear [Carry/Neg/Zero] bit	-	[0]	[0]	[0]
DADC	dst	DADD #0,dst	Add Carry to dst decimally	*	*	*	*
DEC(D)	dst	SUB #[1/2],dst	Decrement by 1 (by 2)	*	*	*	*
[D/E]INT		BI[C/S] #8,SR	[Disable/Enable] Interrupts	-	-	-	-
INC(D)	dst	ADD #[1/2],dst	Increment by 1 (by 2)	*	*	*	*
INV	dst	XOR #-1,dst	Invert DST	*	*	*	*
NOP		MOV R3,R3	No operation	-	-	-	-
POP	dst	MOV @SP+,dst	Pop operand from stack	-	-	-	-
RET		MOV @SP+,PC	Return from subroutine	-	-	-	-
RL[A/C]	dst	ADD(C) dst,dst	C ← MSB← ... ← LSB ← [0/C _{n-1}]	*	*	*	*
SBC	dst	SUBC #0,dst	Subtract carry from dst	*	*	*	*
SET[C/N/Z]		BIS #[1/4/2],SR	Set [Carry/Negative/Zero] bit	-	[1]	[1]	[1]
TST	dst	CMP(.B) #0,dst	Test dst (compare with 0)	0	*	*	1

- (1) – Os 3 bits menos significativos devem ser ignorado (correspondem a B/W e Ad no formato 2)
- (2) – No caso de instruções de salto, a coluna op-code deve ser lida como condição de salto.

As (SRC)	Ad (DST)	Modo de endereçamento	Sintaxe	R2	R3
00	0	Registro	Rn	SR	0
01	1	Indexado Simbólico Absoluto	X(Rn) ADDR &ADDR	0	+1
10	-	Indireto	@Rn	+4	+2
11	-	Indireto c/ auto-incremento Imediato	@Rn+ #value	+8	-1