

Instrução	Operandos	RTL	Flags afectadas	Comentários
ADD	Rd, Rs	$Rd \leftarrow Rd + Rs$	Z, N, C, V	
ADD	Rd, k	$Rd \leftarrow Rd + k$	Z, N, C, V	$k \in [-8 .. +7]$
ADDC	Rd, Rs	$Rd \leftarrow Rd + Rs + C$	Z, N, C, V	
SUB	Rd, Rs	$Rd \leftarrow Rd - Rs$	Z, N, C, V	
SUB	Rd, k	$Rd \leftarrow Rd - k$	Z, N, C, V	$k \in [-8 .. +7]$
SUBB	Rd, Rs	$Rd \leftarrow Rd - Rs - C$	Z, N, C, V	
CMP	Rd, Rs	$(Rd - Rs)$	Z, N, C, V	Rd não é alterado
CMP	Rd, k	$(Rd - k)$	Z, N, C, V	$K \in [-8 .. +7]$, Rd não é alterado
MUL	Rd, Rs	$Rd \leftarrow Rd * Rs$	Z, N, C, V	O registo Rs é alterado
DIV	Rd, Rs	$Rd \leftarrow \text{quociente}(Rd / Rs)$	Z, N, C, $V \leftarrow 0$	Divisão inteira
MOD	Rd, Rs	$Rd \leftarrow \text{resto}(Rd / Rs)$	Z, N, C, $V \leftarrow 0$	Resto da divisão inteira
NEG	Rd	$Rd \leftarrow -Rd$	Z, N, C, V	Complemento para 2, $V \leftarrow 1$ se Rd for 8000H
AND	Rd, Rs	$Rd \leftarrow Rd \wedge Rs$	Z, N	
OR	Rd, Rs	$Rd \leftarrow Rd \vee Rs$	Z, N	
NOT	Rd	$Rd \leftarrow Rd \oplus \text{FFFFH}$	Z, N	Complemento para 1
XOR	Rd, Rs	$Rd \leftarrow Rd \oplus Rs$	Z, N	
TEST	Rd, Rs	$Rd \wedge Rs$	Z, N	Rd não é alterado
BIT	Rd, n	$Z \leftarrow Rd(n) \oplus 1$	Z	Rd não é alterado
SET	Rd, n	$Rd(n) \leftarrow 1$	Z, N ou outra (se Rd for RE)	$n \in [0 .. 15]$, Se Rd=RE, afecta apenas RE(n)
EI		$RE(IE_index) \leftarrow 1$	EI	Enable interrupts
EI0		$RE(IE0_index) \leftarrow 1$	EI0	Enable interrupt 0
EI1		$RE(IE1_index) \leftarrow 1$	EI1	Enable interrupt 1
EI2		$RE(IE2_index) \leftarrow 1$	EI2	Enable interrupt 2
EI3		$RE(IE3_index) \leftarrow 1$	EI3	Enable interrupt 3
SETC		$RE(C_index) \leftarrow 1$	C	Set Carry flag
EDMA		$RE(DE_index) \leftarrow 1$	DE	Enable DMA
CLR	Rd, n	$Rd(n) \leftarrow 0$	Z, N ou outra (se Rd for RE)	$n \in [0 .. 15]$, Se Rd=RE, afecta apenas RE(n)
DI		$RE(IE_index) \leftarrow 0$	EI	Disable interrupts
DI0		$RE(IE0_index) \leftarrow 0$	EI0	Disable interrupt 0
DI1		$RE(IE1_index) \leftarrow 0$	EI1	Disable interrupt 1
DI2		$RE(IE2_index) \leftarrow 0$	EI2	Disable interrupt 2
DI3		$RE(IE3_index) \leftarrow 0$	EI3	Disable interrupt 3
CLRC		$RE(C_index) \leftarrow 0$	C	Clear Carry flag
DDMA		$RE(DE_index) \leftarrow 0$	DE	Disable DMA
CPL	Rd, n	$Rd(n) \leftarrow Rd(n) \oplus 1$	Z, N ou outra (se Rd for RE)	$n \in [0 .. 15]$, Se Rd=RE, afecta apenas RE(n)
CPLC		$RE(C_index) \leftarrow RE(C_index) \oplus 1$	C	Complement Carry flag
SHR	Rd, n	$n > 0 : C \leftarrow Rd(n-1), n > 0 : Rd \leftarrow 0\{n\} \parallel Rd(15..n)$	Z, N, C	$n \in [0 .. 15]$, Se n=0, actualiza Z e N (C não)
SHL	Rd, n	$n > 0 : C \leftarrow Rd(15-n+1), n > 0 : Rd \leftarrow Rd(15-n..0) \parallel 0\{n\}$	Z, N, C	$n \in [0 .. 15]$, Se n=0, actualiza Z e N (C não)
SHRA	Rd, n	$n > 0 : C \leftarrow Rd(n-1), n > 0 : Rd \leftarrow Rd(15)\{n\} \parallel Rd(15..n)$	Z, N, C	$n \in [0 .. 15]$, Se n=0, actualiza Z e N (C não)
SHLA	Rd, n	$n > 0 : C \leftarrow Rd(15-n+1), n > 0 : Rd \leftarrow Rd(15-n..0) \parallel 0\{n\}$	Z, N, C, V	$n \in [0 .. 15]$, Se n=0, actualiza Z e N (C não) $V \leftarrow 1$ se algum dos bits que sair for diferente do Rd(15) após execução
ROR	Rd, n	$n > 0 : C \leftarrow Rd(n-1), n > 0 : Rd \leftarrow Rd(n-1..0) \parallel Rd(15..n)$	Z, N, C	$n \in [0 .. 15]$, Se n=0, actualiza Z e N (C não)
ROL	Rd, n	$n > 0 : C \leftarrow Rd(15-n+1), n > 0 : Rd \leftarrow Rd(15-n..0) \parallel Rd(15..15-n+1)$	Z, N, C	$n \in [0 .. 15]$, Se n=0, actualiza Z e N (C não)
RORC	Rd, n	$n > 0 : (Rd \parallel C) \leftarrow Rd(n-2..0) \parallel C \parallel Rd(15..n-1)$	Z, N, C	$n \in [0 .. 15]$, Se n=0, actualiza Z e N (C não)
ROLC	Rd, n	$n > 0 : (C \parallel Rd) \leftarrow Rd(15-n+1..0) \parallel C \parallel Rd(15..15-n+2)$	Z, N, C	$n \in [0 .. 15]$, Se n=0, actualiza Z e N (C não)
MOV	Rd, [Rs + off]	$Rd \leftarrow Mw[Rs + off]$	Nenhuma	off $\in [-16 .. +14]$
MOV	Rd, [Rs]	$Rd \leftarrow Mw[Rs + 0000]$	Nenhuma	
MOV	Rd, [Rs + Ri]	$Rd \leftarrow Mw[Rs + Ri]$	Nenhuma	
MOV	[Rd + off], Rs	$Mw[Rd + off] \leftarrow Rs$	Nenhuma	off $\in [-16 .. +14]$
MOV	[Rd], Rs	$Mw[Rd + 0000] \leftarrow Rs$	Nenhuma	
MOV	[Rd + Ri], Rs	$Mw[Rd + Ri] \leftarrow Rs$	Nenhuma	
MOVB	Rd, [Rs]	$Rd \leftarrow 0\{8\} \parallel Mb[Rs]$	Nenhuma	
MOVB	[Rd], Rs	$Mb[Rd] \leftarrow Rs(7..0)$	Nenhuma	O byte adjacente a Mb[Rd] não é afectado
MOVBS	Rd, [Rs]	$Rd \leftarrow Mb[Rs](7)\{8\} \parallel Mb[Rs]$	Nenhuma	Idêntica a MOVB Rd, [Rs], mas em que é feita extensão de sinal
MOVVP	Rd, [Rs]	$Rd \leftarrow Mw[Rs]$	Nenhuma	Não usa memória virtual nem <i>caches</i> (para acesso aos periféricos)
MOVVP	[Rd], Rs	$Mw[Rd] \leftarrow Rs$	Nenhuma	Não usa memória virtual nem <i>caches</i> (para acesso aos periféricos)

MOVL	Rd, k	$Rd \leftarrow k(7)(8) \parallel k$	Nenhuma	$k \in [-128 \dots +127]$, k é estendido a 16 bits com sinal
MOVH	Rd, k	$Rd(15..8) \leftarrow k$	Nenhuma	$k \in [0 \dots 255]$, O byte de menor peso não é afectado
MOV	Rd, k	$Rd \leftarrow k(7)(8) \parallel k$	Nenhuma	Se $k \in [-128 \dots +127]$
MOV	Rd, k	$Rd \leftarrow k(7)(8) \parallel k(7..0), Rd(15..8) \leftarrow k(15..8)$	Nenhuma	Se $k \in [-32768 \dots -129]$ ou $k \in [+128 \dots +32767]$
MOV	Rd, Rs	$Rd \leftarrow Rs$	Nenhuma	
MOV	Ad, Rs	$Ad \leftarrow Rs$	Nenhuma	
MOV	Rd, As	$Rd \leftarrow As$	Nenhuma	
MOV	Rd, USP	$Rd \leftarrow USP$	Nenhuma	O SP lido é o de nível utilizador, independentemente do bit NP do RE
MOV	USP, Rs	$USP \leftarrow Rs$	Nenhuma	O SP escrito é o de nível utilizador, independentemente do bit NP do RE
SWAP	Rd, Rs	$TEMP \leftarrow Rd, Rd \leftarrow Rs, Rs \leftarrow TEMP$	Nenhuma	
SWAP	Rd, [Rs] ou [Rs], Rd	$TEMP \leftarrow Mw[Rs], Mw[Rs] \leftarrow Rd, Rd \leftarrow TEMP$	Nenhuma	Recomeçável sem reposição de estado mesmo que um dos acessos à memória falhe
PUSH	Rs	$Mw[SP-2] \leftarrow Rs, SP \leftarrow SP - 2$	Nenhuma	SP só é actualizado no fim para ser re-executável
POP	Rd	$Rd \leftarrow Mw[SP], SP \leftarrow SP + 2$	Nenhuma	
JZ	etiqueta	$Z=1: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JNZ	etiqueta	$Z=0: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JN	etiqueta	$N=1: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JNN	etiqueta	$N=0: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JP	etiqueta	$(N \vee Z)=0: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JNP	etiqueta	$(N \vee Z)=1: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JC	etiqueta	$C=1: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JNC	etiqueta	$C=0: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JV	etiqueta	$V=1: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JNV	etiqueta	$V=0: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JEQ	etiqueta	$Z=1: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JNE	etiqueta	$Z=0: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JLT	etiqueta	$N \oplus V = 1 : PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JLE	etiqueta	$((N \oplus V) \vee Z) = 1 : PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JGT	etiqueta	$((N \oplus V) \vee Z) = 0 : PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JGE	etiqueta	$N \oplus V = 0 : PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JA	etiqueta	$(C \vee Z) = 0 : PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JAE	etiqueta	$C = 0: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JB	etiqueta	$C = 1: PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JBE	etiqueta	$(C \vee Z) = 1 : PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS - 256 \dots EIS + 254]$
JMP	etiqueta	$PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS-4096 \dots EIS+4094]$
JMP	Rs	$PC \leftarrow Rs$	Nenhuma	
CALL	etiqueta	$Mw[SP-2] \leftarrow EIS, PC \leftarrow EIS + (2*dif), SP \leftarrow SP - 2$	Nenhuma	etiqueta $\in [EIS-4096 \dots EIS+4094]$, SP só é actualizado no fim para ser re-executável
CALL	Rs	$Mw[SP-2] \leftarrow EIS, PC \leftarrow Rs, SP \leftarrow SP - 2$	Nenhuma	SP só é actualizado no fim para ser re-executável
CALLF	etiqueta	$RL \leftarrow EIS, PC \leftarrow EIS + (2*dif)$	Nenhuma	etiqueta $\in [EIS-4096 \dots EIS+4094]$
CALLF	Rs	$RL \leftarrow EIS, PC \leftarrow Rs$	Nenhuma	
RET		$PC \leftarrow Mw[SP], SP \leftarrow SP + 2$	Nenhuma	
RETF		$PC \leftarrow RL$	Nenhuma	
SWE	k	$TEMP \leftarrow RE, RE \leftarrow 0, Mw[SP-2] \leftarrow EIS, Mw[SP-4] \leftarrow TEMP,$ $PC \leftarrow Mw[BTE+2*k], SP \leftarrow SP - 4$	Todas colocadas a zero	SP só é actualizado no fim para ser re-executável, $k \in [0 \dots 255]$ Tem de se usar a pilha de sistema
RFE		$TEMP \leftarrow Mw[SP], PC \leftarrow Mw[SP+2], SP \leftarrow SP + 4, RE \leftarrow TEMP$	Todas restauradas	SP só é actualizado no fim para ser re-executável Tem de se usar a pilha de sistema
NOP			Nenhuma	Não faz nada

Registo de Estado (bits 15 a 0)		R NP DE IE3 IE2 IE1 IE0 IE TD TV B A V C N Z
Registo de Configuração do Núcleo		bits 15 a 9: reservados
	pino de interrupção INT3	bits 7 e 6: 00 (flanco de 0 para 1) / 01 (flanco de 1 para 0) / 10 (nível 1) / 11 (nível 0)
	pino de interrupção INT2	idem nos bits 5 e 4
	pino de interrupção INT1	idem nos bits 3 e 2
	pino de interrupção INT0	idem nos bits 1 e 0