

MNEMÔNICO	EXP. GRAF.	FLAGS
INSTRUÇÕES DE TRANSFERÊNCIA		
MOV r1,r2	$(r1) \leftarrow (r2)$	NENHUM
MOV r,M	$(r) \leftarrow [(HL)]$	NENHUM
MOV M,r	$[(HL)] \leftarrow (r)$	NENHUM
MVI r,byte	$(r) \leftarrow \text{byte}$	NENHUM
MVI M,byte	$[(HL)] \leftarrow \text{byte}$	NENHUM
LXI rp,doble	$(rpl) \leftarrow 1^{\circ} \text{ byte}$ $(rph) \leftarrow 2^{\circ} \text{ byte}$	NENHUM
LDA addr	$(A) \leftarrow [\text{addr}]$	NENHUM
STA addr	$[\text{addr}] \leftarrow (A)$	NENHUM
LHLD addr	$(L) \leftarrow [\text{addr}]$ $(H) \leftarrow [\text{addr}+1]$	NENHUM
SHLD addr	$[\text{addr}] \leftarrow (L)$ $[\text{addr}+1] \leftarrow (H)$	NENHUM
LDAX rp	$(A) \leftarrow [(rp)]$	NENHUM
STAX rp	$[(rp)] \leftarrow (A)$	NENHUM
XCHG	$(H) \leftrightarrow (D) \quad (L) \leftrightarrow (E)$	NENHUM
INSTRUÇÕES ARITMÉTICAS		
ADD r	$(A) \leftarrow (A) + (r)$	TODOS
ADD M	$(A) \leftarrow (A) + [(HL)]$	TODOS
ADI byte	$(A) \leftarrow (A) + \text{byte}$	TODOS
ADC r	$(A) \leftarrow (A) + (r) + CY$	TODOS
ADC M	$(A) \leftarrow (A) + [(HL)] + CY$	TODOS
ACI byte	$(A) \leftarrow (A) + \text{byte} + CY$	TODOS
SUB r	$(A) \leftarrow (A) - (r)$	TODOS
SUB M	$(A) \leftarrow (A) - [(HL)]$	TODOS
SUI byte	$(A) \leftarrow (A) - \text{byte}$	TODOS
SBB r	$(A) \leftarrow (A) - (r) - CY$	TODOS
SBB M	$(A) \leftarrow (A) - [(HL)] - CY$	TODOS
SBI byte	$(A) \leftarrow (A) - \text{byte} - CY$	TODOS
INR r	$(r) \leftarrow (r) + 1$	Z, S, P, AC
INR M	$[(HL)] \leftarrow [(HL)] + 1$	Z, S, P, AC
DCR r	$(r) \leftarrow (r) - 1$	Z, S, P, AC
DCR M	$[(HL)] \leftarrow [(HL)] - 1$	Z, S, P, AC
INX rp	$(rp) \leftarrow (rp) + 1$	NENHUM
DCX rp	$(rp) \leftarrow (rp) - 1$	NENHUM
DAD rp	$(HL) \leftarrow (HL) + (rp)$	CY
DAA	Ajuste BCD de (A)	NENHUM
INSTRUÇÕES LÓGICAS.		
ANA r	$(A) \leftarrow (A) \text{ and } (r)$ $(CY) \leftarrow 0, (AC) \leftarrow 1$	TODOS
ANA M	$(A) \leftarrow (A) \text{ and } [(HL)]$ $(CY) \leftarrow 0, (AC) \leftarrow 1$	TODOS
ANI byte	$(A) \leftarrow (A) \text{ and byte}$ $(CY) \leftarrow 0, (AC) \leftarrow 1$	TODOS
XRA r	$(A) \leftarrow (A) \text{ xor } (r)$	TODOS
XRA M	$(A) \leftarrow (A) \text{ xor } [(HL)]$	TODOS
XRI byte	$(A) \leftarrow (A) \text{ xor byte}$	TODOS
ORA r	$(A) \leftarrow (A) \text{ or } (r)$	TODOS
ORA M	$(A) \leftarrow (A) \text{ or } [(HL)]$	TODOS
ORI byte	$(A) \leftarrow (A) \text{ or byte}$	TODOS
CMP r	$(A) - (r)$	TODOS
CMP M	$(A) - [(HL)]$	TODOS
CPI byte	$(A) - \text{byte}$	TODOS

MNEMÔNICO	EXP. GRAF.	FLAGS
INSTRUÇÕES DE ROTAÇÃO E FLAGS		
RLC	Rotação esquerda	CY
RRC	Rotação direita	CY
RAL	Rot. esqda. com CY	CY
RAR	Rot. direita. com CY	CY
CMA	Comp. A1 de (A)	NENHUM
CMC	Inverte (CY)	CY
STC	$(CY) \leftarrow 1$	CY
INSTRUÇÕES DE BIFURCAÇÃO		
ccc=NZ salto se não zero (Z=0), ccc=Z salto se zero (Z=0), ccc=NC salto se não carry (CY=0), ccc=C salto se carry (CY=1), ccc=PO salto se paridade impar (P=0), ccc=PE salto se paridade par (P=1), ccc=P salto se positivo (S=0), ccc=M salto se negativo (S=1)		
JMP addr	$(PC) \leftarrow \text{addr}$	NENHUM
Jccc addr	Si ccc=1, $(PC) \leftarrow \text{addr}$; Si ccc=0, $(PC) \leftarrow (PC) + 3$	NENHUM
CALL addr	Guarda PC na pilha $(PC) \leftarrow \text{addr}$	NENHUM
Cccc addr	Si ccc=1, guarda PC na pilha, $(PC) \leftarrow \text{addr}$; Si ccc=0, $(PC) \leftarrow (PC) + 3$	NENHUM
RET	Recupera PC da pilha	NENHUM
Rccc	Si ccc=1, recupera PC da pilha; Si ccc=0 $(PC) \leftarrow (PC) + 1$	NENHUM
RSTn	$(PC) \leftarrow n \times 8$	NENHUM
PCHL	$(PC) \leftarrow (HL)$	NENHUM
INSTRUÇÕES DE MANUSEIO DA PILHA.		
PUSH rp	$[(SP)-1] \leftarrow (rpl)$ $[(SP)-2] \leftarrow (rph)$ $(SP) \leftarrow (SP) - 2$	NENHUM
PUSH PSW	$[(SP)-1] \leftarrow (A)$ $[(SP)-2] \leftarrow (RE)$ $(SP) \leftarrow (SP) - 2$	NENHUM
POP rp	$(rph) \leftarrow [(SP)]$ $(rpl) \leftarrow [(SP)+1]$ $(SP) \leftarrow (SP) + 2$	NENHUM
POP PSW	$(RE) \leftarrow [(SP)]$ $(A) \leftarrow [(SP)+1]$ $(SP) \leftarrow (SP) + 2$	NENHUM
XLTH	$(L) \leftrightarrow [(SP)]$ $(H) \leftrightarrow [(SP)+1]$	NENHUM
SPLH	$(HL) \leftrightarrow (SP)$	NENHUM
INSTRUÇÕES DE ENTRADA E SAÍDA.		
IN porta	$(A) \leftarrow [\text{porta}]$	NENHUM
OUT porta	$[\text{porta}] \leftarrow (A)$	NENHUM
INSTRUC. DE CONTROLE DE INTERRUPTÕES		
EI	Habilita interrupções	NENHUM
DI	Desabilita interrupções	NENHUM
HLT	Para o microprocessador	NENHUM
NOP	Não faz nada	NENHUM
RIM	Lê linha serial e estado das interrupções.	NENHUM
SIM	Escreve na linha serial e Programa interrupções.	NENHUM