

S = Flag de Sinal - assume o valor 1 quando o resultado de uma operação é negativo

Z = Flag de Zero - assume o valor 1 quando o resultado de uma operação é zero

AC = Auxiliar de Carry = flag usada como auxiliar de transporte. Assume valor 1 quando há transporte do bit 3 para o bit 4. É usada em operações BCD

P = Flag de Paridade = assume valor 1 quando há um número de par de algarismos 1 no acumulador

CY = Flag de Carry (transporte) = assume valor 1 quando há transporte do bit 7.

3.4 Instruções de Transferência de Dados

Mnemônico Genérico	Simbologia	Nº de Ciclos	Nº de Estados	Modo de Endereçamento	Flags Afetadas	Comentário
MVI r, dado8	$(r) \leftarrow (\text{byte } 2)$	2	7	Imediato	nenhuma	move o dado para o registrador r indicado
MOV r1, r2	$(r1) \leftarrow (r2)$	1	4	Registrador	nenhuma	move o conteúdo do registrador r2 para o registrador r1
MOV r, M	$(r) \leftarrow ((H)(L))$	2	7	Indireto por registrador	nenhuma	move para o registrador r o dado presente no endereço de memória especificado pelo registrador par H-L
MOV M, r	$((H)(L)) \leftarrow (r)$	2	7	Indireto por registrador	nenhuma	move o conteúdo do registrador r para a posição de memória especificada pelo registrador par H-L
MVI M, dado8	$((H)(L)) \leftarrow (\text{byte } 2)$	3	10	Indireto por registrador e imediato	nenhuma	Carrega o dado na posição de memória especificada pelo registrador par H-L.
LXI rp, dado16	$(rh) \leftarrow (\text{byte } 3)$ $(rl) \leftarrow (\text{byte } 2)$	3	10	Imediato	nenhuma	Carrega o dado de 16 bits no registrador par indicado em rp . O byte 2 da instrução é colocado no registrador de menor ordem, ou byte menos significativo, rl . O byte 3 da instrução é colocado no registrador de maior ordem, ou byte mais significativo, rh
LDA addr	$(A) \leftarrow ((\text{byte } 3)(\text{byte } 2))$	4	13	direto	nenhuma	Move o conteúdo do acumulador para a posição de memória indicada pelo endereço addr . O byte 2 armazena o byte inferior do endereço. O byte 3 da instrução armazena o byte superior do endereço.
STA addr	$((\text{byte } 3)(\text{byte } 2)) \leftarrow (A)$	4	13	direto	nenhuma	Move o conteúdo do acumulador para a posição de memória indicada pelo endereço addr . O byte 2 armazena o byte inferior do endereço. O byte 3 da instrução armazena o byte superior do endereço.
LHLD addr	$(L) \leftarrow ((\text{byte } 3)(\text{byte } 2))$ $(H) \leftarrow ((\text{byte } 3)(\text{byte } 2) + 1)$	5	16	direto	nenhuma	Carrega o conteúdo da posição de memória dada por addr ((byte 3)(byte 2)) no registrador L. Carrega o conteúdo da posição subsequente ((byte 3)(byte 2) + 1) no registrador H.
SHLD addr	$((\text{byte } 3)(\text{byte } 2)) \leftarrow (L)$ $((\text{byte } 3)(\text{byte } 2) + 1) \leftarrow (H)$	5	16	direto	nenhuma	Move o conteúdo do registrador L para a posição de memória dada por addr ((byte 3)(byte 2)). Move o conteúdo do registrador H para a posição subsequente de memória ((byte 3)(byte 2) + 1).
LDAX rp	$(A) \leftarrow ((rp))$	2	7	Indireto por registrador	nenhuma	Carrega acumulador com o conteúdo da posição de memória indicada pelo registrador par rp . rp pode ser B (do registrador para BC) ou D (do registrador para DE).
STAX rp	$((rp)) \leftarrow (A)$	2	7	Indireto por registrador	nenhuma	Move o conteúdo do acumulador para a posição de memória indicada pelo registrador par rp . rp pode ser B (do registrador para BC) ou D (do registrador para DE).
XCHG	$(H) \leftrightarrow (D)$ $(L) \leftrightarrow (E)$	1	4	registrador	nenhuma	O conteúdo do registrador H é trocado com o conteúdo do registrador D. O conteúdo de L é trocado com o conteúdo de E.

3.5 Instruções Aritméticas

Obs.: A menos que seja indicado, todas as instruções desse grupo afetam todas as Flags: Zero, Sinal, Paridade, Transporte (Carry) e Auxiliar de Transporte (Auxiliar de Carry)

Mnemônico Genérico	Simbologia	Nº de Ciclos	Nº de Estados	Modo de Endereçamento	Flags Afetadas	Comentário
ADD r	$(A) \leftarrow (A) + (r)$	1	4	registrador	todas	O conteúdo do registrador r é adicionado ao conteúdo do acumulador.
ADD M	$(A) \leftarrow (A) + ((H)(L))$	2	7	registrador	todas	O conteúdo da posição de memória indicado pelo par HL é adicionado ao conteúdo do acumulador.
ADI dado8	$(A) \leftarrow (A) + (\text{byte } 2)$	2	7	imediato	todas	O valor dado em data é adicionado ao conteúdo do acumulador.
ADC r	$(A) \leftarrow (A) + (r) + (CY)$	1	4	registrador	todas	O conteúdo do registrador r é adicionado com carry ao conteúdo do acumulador.
ADC M	$(A) \leftarrow (A) + ((H)(L)) + (CY)$	2	7	imediato por registrador	todas	O conteúdo da posição indicada pelo par HL é adicionado com carry ao conteúdo do acumulador.
ACI dado8	$(A) \leftarrow (A) + (\text{byte } 2) + (CY)$	2	7	imediato	todas	O valor dado em data é adicionado com carry ao conteúdo do acumulador.
SUB r	$(A) \leftarrow (A) - (r)$	1	4	registrador	todas	O conteúdo do registrador r é subtraído do conteúdo do acumulador.
SUB M	$(A) \leftarrow (A) - ((H)(L))$	2	7	indireto por registrador	todas	O conteúdo da posição de memória indicado pelo par HL é subtraído do conteúdo do acumulador.
SUI dado8	$(A) \leftarrow (A) - (\text{byte } 2)$	2	7	imediato	todas	O valor dado em data é subtraído do conteúdo do acumulador.
SBB r	$(A) \leftarrow (A) - (r) - (CY)$	1	4	registrador	todas	O conteúdo do registrador r é subtraído com carry do conteúdo do acumulador.
SBB M	$(A) \leftarrow (A) - ((H)(L)) - (CY)$	2	7	indireto por registrador	todas	O conteúdo da posição indicada pelo par HL é subtraído com carry do conteúdo do acumulador.
SBI dado8	$(A) \leftarrow (A) - (\text{byte } 2) - (CY)$	2	7	imediato	todas	O valor dado em data é subtraído com carry do conteúdo do acumulador.
INR r	$(r) \leftarrow (r) + 1$	1	4	registrador	Z, S, P e AC	O conteúdo do registrador r é adicionado de 1. Todas as Flags são afetadas, exceto CY.
INR M	$((H)(L)) \leftarrow ((H)(L)) + 1$	3	10	indireto por registrador	Z, S, P e AC	O conteúdo da posição apontada pelo par HL é incrementado de 1.
DCR r	$(r) \leftarrow (r) - 1$	1	4	registrador	Z, S, P e AC	O conteúdo do registrador r é decrementado. Todas as Flags são afetadas, exceto CY.
DCR M	$((H)(L)) \leftarrow ((H)(L)) - 1$	3	10	indireto por registrador	Z, S, P e AC	O conteúdo da posição apontada pelo par HL é decrementado de 1.
INX rp	$(rh)(rl) \leftarrow (rh)(rl) + 1$	1	6	registrador	nenhuma	O conteúdo do registrador par rp é adicionado de 1. Nenhuma Flag é afetada.

Instruções Aritméticas - Continuação

Mnemônico Genérico	Simbologia	Nº de Ciclos	Nº de Estados	Modo de Endereçamento	Flags Afetadas	Comentário
DCX rp	$(rh)(rl) \leftarrow (rh)(rl) - 1$	1	6	registrador	nenhuma	O conteúdo do registrador par rp é decrementado de 1.
DAD rp	$((H)(L)) \leftarrow ((H)(L)) + (rh)(rl)$	3	10	registrador	CY	O conteúdo do registrador par rp é adicionado ao conteúdo do registrador par HL. Somente a Flag de Carry (CY) é afetada.
DAA		1	4	registrador	todas	Faz o ajuste decimal do número no acumulador. O número de 8 bits do acumulador é ajustado para formar dois números de 4 bits em BCD. A regra seguida é a seguinte: 1. Se o número representado pelo nibble inferior for maior do que 9, ou se a Flag AC estiver setada, o número 6 é adicionado ao conteúdo do acumulador. 2. Se o número representado pelo nibble superior for maior do que 9, ou se a Flag CY estiver setada, o número 6 é adicionado ao nibble superior.

3.6 Instruções Lógicas

Obs.: Com exceção dos casos indicados, todas as instruções lógicas afetam as Flags

Mnemônico Genérico	Simbologia	Nº de Ciclos	Nº de Estados	Modo de Endereçamento	Flags Afetadas	Comentário
ANA r	$(A) \leftarrow (A) \wedge (r)$	1	4	registrador	todas	O conteúdo do acumulador passa por uma operação lógica AND com o conteúdo do registrador indicado em r . A flag CY é zerada e a flag AC é setada.
ANA M	$(A) \leftarrow (A) \wedge ((H)(L))$	2	7	indireto por registrador	todas	O conteúdo do acumulador passa por uma operação lógica AND com o conteúdo da posição de memória apontada pelo registrador par HL. A flag CY é zerada e a flag AC é setada.
ANI dado8	$(A) \leftarrow (A) \wedge (\text{byte } 2)$	2	7	imediato	todas	O conteúdo do acumulador passa por uma operação lógica AND com o dado fornecido no byte 2 da instrução. A flag CY é zerada e a flag AC é setada.
XRA r	$(A) \leftarrow (A) \vee (r)$	1	4	registrador	todas	O conteúdo do acumulador passa por uma operação lógica XOR com o conteúdo do registrador r indicado. As flags CY e AC são zeradas.
XRA M	$(A) \leftarrow (A) \vee ((H)(L))$	2	7	indireto por registrador	todas	O conteúdo do acumulador passa por uma operação lógica XOR com o conteúdo do endereço de memória apontada pelo registrador par HL. As flags CY e AC são zeradas.
XRI dado8	$(A) \leftarrow (A) \vee (\text{byte } 2)$	2	7	imediato	todas	O conteúdo do acumulador passa por uma operação lógica XOR com o dado fornecido no byte 2 da instrução. As flags CY e AC são zeradas.
ORA r	$(A) \leftarrow (A) \vee (r)$	1	4	registrador	todas	O conteúdo do acumulador passa por uma operação lógica OR com o conteúdo do registrador indicado em r . As flags CY e AC são zeradas.
ORA M	$(A) \leftarrow (A) \vee ((H)(L))$	2	7	indireto por registrador	todas	O conteúdo do acumulador passa por uma operação lógica OR com o conteúdo da posição de memória apontada pelo registrador par HL. As flags CY e AC são zeradas.
ORI dado8	$(A) \leftarrow (A) \vee (\text{byte } 2)$	2	7	imediato	todas	O conteúdo do acumulador passa por uma operação lógica OR com o dado presente no byte 2 da instrução. As flags CY e AC são zeradas.
CMP r	$(A) - (r)$	1	4	registrador	todas	Compara o conteúdo do registrador r com o conteúdo do acumulador. Na operação o conteúdo de r é subtraído do conteúdo de A , sem que o resultado seja guardado em A . Se o resultado da subtração for zero, ou seja $(A) = (r)$, a flag de zero Z = 1. Caso contrário, Z = 0. Se $(A) < (r)$, a flag de carry CY = 1. Caso contrário, CY = 0.
CMP M	$(A) \leftarrow (A) - ((H)(L))$	2	7	indireto por registrador	todas	Compara o conteúdo da posição apontada pelo par HL com o conteúdo do acumulador. Na operação o conteúdo da posição apontada por HL é subtraído do conteúdo de A , sem que o resultado seja guardado em A . Se o resultado da subtração for zero, ou seja $(A) = ((H)(L))$, a flag de zero Z

						= 1. Caso contrário, Z = 0. Se $(A) < ((H)(L))$, a flag de carry CY = 1. Caso contrário, CY = 0.
CPI dado8	$(A) \leftarrow (A) - (\text{byte } 2)$	2	7	imediato	todas	Compara o conteúdo do byte 2 da instrução com o conteúdo do acumulador. Na operação o conteúdo do byte 2 é subtraído do conteúdo de A , sem que o resultado seja guardado em A . Se o resultado da subtração for zero, ou seja $(A) = (\text{byte } 2)$, a flag de zero Z = 1. Caso contrário, Z = 0. Se $(A) < (\text{byte } 2)$, a flag de carry CY = 1. Caso contrário, CY = 0.
RLC	$(A_{n+1}) \leftarrow (A_n)$ $(A_0) \leftarrow (A_7)$ $(CY) \leftarrow (A_7)$	1	4		CY	(Rotate Left) = (Rotacionar à Esquerda) = O conteúdo do acumulador é rotacionado uma posição à esquerda. O conteúdo do último bit (bit 7) é transferido tanto para a posição do bit 0, quanto para o Carry. O conteúdo do bit 'i' é transferido para o bit 'i + 1'. Somente a flag de carry é afetada.
RRC	$(A_n) \leftarrow (A_{n+1})$ $(A_7) \leftarrow (A_0)$ $(CY) \leftarrow (A_0)$	1	4		CY	(Rotate Right) = (Rotacionar à Direita) = O conteúdo do acumulador é rotacionado uma posição à direita. O conteúdo do bit menos significativo (bit 0) é transferido tanto para a posição do bit 7, quanto para o Carry. O conteúdo do bit 'i' é transferido para o bit 'i - 1'. Somente a flag de carry CY é afetada.
RAL	$(A_{n+1}) \leftarrow (A_n)$ $(CY) \leftarrow (A_7)$ $(A_0) \leftarrow (CY)$	1	4		CY	(Rotate Left through Carry) = (Rotacionar à Esquerda através do Carry) = O conteúdo do acumulador é rotacionado uma posição à esquerda, incluindo o bit de carry como o oitavo bit. Assim, o conteúdo do bit 7 é transferido para o bit de Carry (CY) e o conteúdo do bit de Carry é transferido para o bit menos significativo (bit 0). Somente a flag de carry é afetada.
RAR	$(A_n) \leftarrow (A_{n+1})$ $(CY) \leftarrow (A_0)$ $(A_7) \leftarrow (CY)$	1	4		CY	(Rotate Right through Carry) = (Rotacionar à Esquerda) = O conteúdo do acumulador é rotacionado uma posição à direita, incluindo o carry como oitavo bit. Assim, o conteúdo do último bit de Carry (CY) é transferido para a posição do bit 7 e conteúdo do bit 0 é transferido para o bit de Carry. Somente a flag de carry é afetada.
CMA	$(A) \leftarrow (A')$	1	4		nenhuma	(Complement Accumulator) = (Complementa Acumulador) = O conteúdo do acumulador é complementado, bit a bit. Os bits zero tornam-se 1 e os bits 1 tornam-se zero. Nenhuma flag é afetada.
CMC	$(CY) \leftarrow (CY')$	1	4		CY	(Complement Carry) = (Complementa Carry) = O conteúdo da flag de Carry complementado. Nenhuma outra flag é afetada.
STC	$(CY) \leftarrow 1$	1	4		CY	(Set Carry) = (Seta o bit de Carry) = A flag de Carry (CY) é feita igual a 1. Nenhuma outra flag é afetada.

Instruções de Desvio:

Mnemônico Genérico	Simbologia	Nº de Ciclos	Nº de Estados	Modo de Endereçamento	Flags Afetadas	Comentário
JMP addr	$(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	3	10	imediato	nenhuma	(Jump to address) = O controle é transferido incondicionalmente para a instrução cujo endereço é dado no segundo e no terceiro bytes da instrução de desvio.
J_{condição} addr	Se (CCC), então, $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	2/3	7/10	imediato	nenhuma	(Jump to address if CCC) = Se a condição indicada for verdadeira o controle é transferido para a instrução cujo endereço é dado no segundo e no terceiro bytes da instrução de desvio. Caso a condição seja falsa, o processamento continua sequencialmente.
JNZ addr	Se (CCC), então, $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	2/3	7/10	imediato	nenhuma	(Jump if Not Zero) = Desvia para o endereço indicado se o resultado da operação aritmética anterior a esta instrução não for zero, ou seja, desvia se Z = 0 .
JZ addr	Se (CCC), então, $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	2/3	7/10	imediato	nenhuma	(Jump if Zero) = Desvia para o endereço indicado se o resultado da operação aritmética anterior a esta instrução for igual a zero, ou seja, desvia se Z = 1 .
JNC addr	Se (CCC), então, $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	2/3	7/10	imediato	nenhuma	(Jump if No Carry) = Desvia para o endereço indicado se a flag de Carry estiver zerada (CY = 0).
JC addr	Se (CCC), então, $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	2/3	7/10	imediato	nenhuma	(Jump if Carry) = Desvia para o endereço indicado se a flag de Carry estiver setada (CY = 1).
JPO addr	Se (CCC), então, $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	2/3	7/10	imediato	nenhuma	(Jump if Parity Odd) = Desvia para o endereço indicado se a paridade for Ímpar, ou seja, se a flag de Paridade for zero (P = 0).
JPE addr	Se (CCC), então, $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	2/3	7/10	imediato	nenhuma	(Jump if Parity Even) = Desvia para o endereço indicado se a paridade for Par, ou seja, se a flag de Paridade estiver setada (P = 1).
JP addr	Se (CCC), então, $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	2/3	7/10	imediato	nenhuma	(Jump if Plus) = Desvia para o endereço indicado se o valor no acumulador for um número positivo, ou seja, tiver o sétimo bit zerado, ou ainda se a flag de Sinal estiver zerada (S = 0).
JM addr	Se (CCC), então, $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	2/3	7/10	imediato	nenhuma	(Jump if Minus) = Desvia para o endereço indicado se o valor no acumulador for um número negativo, ou seja, tiver o sétimo bit setado, ou ainda se a flag de Sinal estiver setada (S = 1).
CALL addr	$((SP) - 1) \leftarrow (PCH)$ $((SP) - 2) \leftarrow (PCL)$ $(SP) \leftarrow (SP) - 2$ $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	5	18	imediato e indireto por registrador	nenhuma	(Call address) = Chamada de subrotina. O processamento é desviado para o endereço indicado em addr , que é dado pelos bytes 2 e 3 da instrução. Antes do desvio para a subrotina, o endereço da próxima instrução é guardado na pilha. O byte superior do endereço da próxima instrução é guardado na posição SP - 1 da pilha e o byte inferior é guardado na posição SP - 2 da pilha. Ao final da subrotina a instrução RET faz o processamento voltar para o programa principal no endereço que foi guardado na pilha.
C_{condição} addr	$((SP) - 1) \leftarrow (PCH)$ $((SP) - 2) \leftarrow (PCL)$ $(SP) \leftarrow (SP) - 2$ $(PC) \leftarrow (\text{byte } 3)(\text{byte } 2)$	2/5	9/18	imediato e direto por registrador	nenhuma	(Call address if CCC is true) = Chamada de subrotina condicional. O processamento é desviado para o endereço indicado em addr se a condição indicada em CCC for verdadeira. O endereço é dado pelos bytes 2 e 3 da instrução. Antes do desvio para a subrotina, o endereço da próxima instrução é guardado na pilha. O byte superior do endereço da próxima instrução é guardado na posição SP - 1 da pilha e o byte inferior é guardado na posição SP - 2 da pilha. Ao final da subrotina a instrução RET faz o processamento voltar para o programa principal no endereço que foi guardado na pilha.

Casos possíveis: CNZ, CZ, CNC, CC, CPO, CPE, CP, CM					
RET	$(PCL) \leftarrow (SP)$ $(PCH) \leftarrow (SP + 1)$ $(SP) \leftarrow (SP) + 2$	3	10	indireto por registrador	nenhuma
R_{condição}	$(PCL) \leftarrow (SP)$ $(PCH) \leftarrow (SP + 1)$ $(SP) \leftarrow (SP) + 2$	1/3	6/12	indireto por registrador	nenhuma
RST n	$((SP) - 1) \leftarrow (PCH)$ $((SP) - 2) \leftarrow (PCL)$ $(SP) \leftarrow (SP) - 2$ $(PC) \leftarrow 8 * (NNN)$	3	12	indireto por registrador	nenhuma
PCHL	$(PCH) \leftarrow (H)$ $(PCL) \leftarrow (L)$	1	6	registrador	nenhuma

3.8 Instruções de Controle, Pilha e Entrada e Saída

Obs.: As instruções desse grupo não afetam as flags, a menos que seja indicado.

Mnemônico Genérico	Simbologia	Nº de Ciclos	Nº de Estados	Modo de Endereçamento	Flags Afetadas	Comentário
PUSH rp	$((SP) - 1 \leftarrow (rh))$ $((SP) - 2 \leftarrow (rl))$ $(SP) \leftarrow (SP) - 2$	3	12	indireto por registrador	nenhuma	(Push) = O conteúdo do registrador de mais alta ordem (byte superior) do par de registradores é guardado na posição de memória indicada por SP - 1 . O conteúdo do registrador que contém o byte inferior da instrução é guardado na posição de memória indicada por SP - 2 . O registrador par rp pode ser B (de BC), D (de DE) ou H (de HL). O registrador SP é decrementado de 2.
PUSH PSW	$((SP) - 1 \leftarrow (A))$ $((SP) - 2 \leftarrow (F))$ $(SP) \leftarrow (SP) - 2$	3	12	indireto por registrador	nenhuma	(Push Processor Status Word) = O conteúdo dos registradores A (acumulador) e F (flags) é guardado na pilha. O conteúdo do Acumulador é guardado na posição de memória indicada por SP - 1 . O conteúdo do registrador F é guardado na posição de memória indicada por SP - 2 .
POP rp	$(rl) \leftarrow ((SP))$ $(rh) \leftarrow ((SP) + 1)$ $(SP) \leftarrow (SP) + 2$	3	10	indireto por registrador	nenhuma	O conteúdo da posição de memória indicada por SP é movido para o registrador que representa o byte inferior (C, E ou L). O conteúdo da posição de memória indicada por SP + 1 é movido para o registrador que representa o byte superior da instrução (B, D ou H). O conteúdo do registrador SP é incrementado de 2.
POP PSW	$(F) \leftarrow ((SP))$ $(A) \leftarrow ((SP) + 1)$ $(SP) \leftarrow (SP) + 2$	3	10	indireto por registrador	todas	O conteúdo da posição de memória indicada por SP é movido para o registrador que guarda o estado das Flags (registrador F). O conteúdo da posição de memória indicada por SP + 1 é movido para o acumulador. O conteúdo do registrador SP é incrementado de 2.
XTHL	$(L) \leftarrow ((SP))$ $(H) \leftarrow ((SP) + 1)$	5	16	indireto por registrador	nenhuma	(Exchange Stack Top with H and L) = O conteúdo da posição de memória indicada por SP é trocado com o conteúdo do registrador L e o conteúdo da posição de memória indicada por SP + 1 é trocado pelo conteúdo do registrador H.
SPHL	$(SP) \leftarrow (H)(L)$	1	6	registrador	nenhuma	(Move HL to SP) = O conteúdo do registrador par HL é movido para o registrador de 16 bits SP .
IN porta	$(A) \leftarrow (\text{dado 8 bits})$	3	10	direto	nenhuma	(Input = Entrada) = O dado colocado na porta indicada é transferido para o acumulador através do barramento de dados (8 bits).
OUT porta	$(\text{dado 8 bits}) \leftarrow (A)$	3	10	direto	nenhuma	(Output = Saída) = O dado do acumulador é transferido para a porta indicada através do barramento de dados (8 bits).

EI		1	4		nenhuma	(Enable Interrupt = Habilita Interrupções) = A interrupção do sistema é habilitada após a execução da instrução seguinte. Nenhuma interrupção é reconhecida durante a execução da instrução EI.
DI		1	4		nenhuma	(Disable Interrupt = Desabilita Interrupções) = A interrupção do sistema é desabilitada imediatamente após a execução da instrução DI. Nenhuma interrupção é reconhecida durante a execução da instrução DI.
HLT		1+	5		nenhuma	A instrução HLT faz o processador parar o processamento. Os registradores e flags permanecem inalterados.
NOP		1	4		nenhuma	(No Operation) = Nenhuma operação é realizada. Os registradores e flags ficam inalterados.
RIM		1	4		nenhuma	(Read Interrupt Mask = Lê Máscara de Interrupção) = Esta instrução carrega no acumulador os dados relativos às interrupções e à entrada serial. Será melhor detalhada posteriormente.
SIM		1	4		nenhuma	(Set Interrupt Mask = Define Máscara de Interrupção) = Esta instrução usa o conteúdo do acumulador para definir as máscaras de interrupção. Será melhor detalhada posteriormente.