

Pagina_01 (BUSCA...)
BE < IP (=60)
(CPU acessa MP endereço 60)

Pagina_02 (...BUSCA)
BC < READ
BD < [60] (=CALL 00)
RI < BD (=CALL 00)
(CPU captura em RI instrução CALL 00)
(CPU finaliza BUSCA)

Pagina_03 (ATUALIZA)
IP < IP + 1 (=60 + 1 = 61)
(CPU atualiza IP sequencialmente)
(CPU finaliza ATUALIZA)

Pagina_04 (EXECUTA...)
SP < SP - 1 (=00 - 1 = FF)
BC < WRITE
[FF] < IP (=61)

Pagina_05 (...EXECUTA)
IP < 00 (operando de CALL 00 em RI)
(CPU finaliza EXECUTA)

Pagina_06 (BUSCA)
BE < IP (=00)
BC < READ
BD < [00] (=MOV A,00)
RI < BD (=MOV A,00)

Pagina_07 (ATUALIZA)
IP < IP + 1 (=00 + 1 = 01)

Pagina_08 (EXECUTA)
A < 00 (MOV A,00)
(PSW(Z) = 1 resultado igual a zero)

Pagina_09 (BUSCA)
BE < IP (=01)
BC < READ
BD < [01] (=MOV B,07)
RI < BD (=MOV B,07)

Pagina_10 (ATUALIZA)
IP < IP + 1 (=01 + 1 = 02)

Pagina_11 (EXECUTA)
B < 07 (MOV B,07)
(PSW(Z) = 0 resultado diferente de zero)

Pagina_12 (BUSCA)
BE < IP (=02)
BC < READ
BD < [02] (=MOV C,03)
RI < BD (=MOV C,03)

Pagina_13 (ATUALIZA)
IP < IP + 1 (=02 + 1 = 03)

Pagina_14 (EXECUTA)
C < 03 (MOV C,03)

Pagina_15 (...BUSCA)
BE < IP (=03)
BC < READ
BD < [03] (=CALL 05)
RI < BD (=CALL 05)

Pagina_16 (ATUALIZA)
IP < IP + 1 (=03 + 1 = 04)

Pagina_17 (EXECUTA...)
SP < SP - 1 (=FF - 1 = FE)
BC < WRITE
[FE] < IP (=04)

Pagina_18 (...EXECUTA)
IP < 05 (operando de CALL 05 em RI)

Pagina_19 (BUSCA)
BE < IP (=05)
BC < READ
BD < [05] (=ADD A,B)
RI < BD (=ADD A,B)

Pagina_20 (ATUALIZA)
IP < IP + 1 (=05 + 1 = 06)

Pagina_21 (EXECUTA...)
R1 < A (=00)
R2 < B (=07)
R3 < R1 + R2 (=07)

Pagina_22 (...EXECUTA)
A < R3 (=07)

Pagina_23 (BUSCA)
BE < IP (=06)
BC < READ
BD < [06] (=DEC C)
RI < BD (=DEC C)

Pagina_24 (ATUALIZA)
IP < IP + 1 (=06 + 1 = 07)

Pagina_25 (EXECUTA...)
R1 < C (=03)
R2 < -1 (=FF)
R3 < R1 + R2 (=02)

Pagina_26 (...EXECUTA)
C < R3 (=02)
(PSW(Z) = 0 resultado diferente de zero)

Pagina_27 (BUSCA)
BE < IP (=07)
BC < READ
BD < [07] (=JNZ 05)
RI < BD (=JNZ 05)

Pagina_28 (ATUALIZA)
IP < IP + 1 (=07 + 1 = 08)

Pagina_29 (EXECUTA)
(PSW(Z) = 0 resultado diferente de zero)
IP < 05 (JNZ 05)

Pagina_30 (BUSCA)
BE < IP (=05)
BC < READ
BD < [05] (=ADD A,B)
RI < BD (=ADD A,B)

Pagina_31 (ATUALIZA)
IP < IP + 1 (=05 + 1 = 06)

Pagina_32 (EXECUTA...)	Pagina_47 (EXECUTA...)
R1 < A (=07)	R1 < C (=01)
R2 < B (=07)	R2 < -1 (=FF)
R3 < R1 + R2 (=14)	R3 < R1 + R2 (=00)
Pagina_33 (...EXECUTA)	Pagina_48 (...EXECUTA)
A < R3 (=14)	C < R3 (=00)
Pagina_34 (BUSCA)	(PSW(Z) = 1 resultado igual a zero)
BE < IP (=06)	Pagina_49 (BUSCA)
BC < READ	BE < IP (=07)
BD < [06] (=DEC C)	BC < READ
RI < BD (=DEC C)	BD < [07] (=JNZ 05)
Pagina_35 (ATUALIZA)	RI < BD (=JNZ 05)
IP < IP + 1 (=06 + 1 = 07)	Pagina_50 (ATUALIZA)
Pagina_36 (EXECUTA...)	IP < IP + 1 (=07 + 1 = 08)
R1 < C (=02)	Pagina_51 (EXECUTA)
R2 < -1 (=FF)	(PSW(Z) = 1 resultado igual a zero)
R3 < R1 + R2 (=01)	Pagina_52 (BUSCA)
Pagina_37 (...EXECUTA)	BE < IP (=08)
C < R3 (=01)	BC < READ
(PSW(Z) = 0 resultado diferente de zero)	BD < [08] (=RET)
Pagina_38 (BUSCA)	RI < BD (=RET)
BE < IP (=07)	Pagina_53 (ATUALIZA)
BC < READ	IP < IP + 1 (=08 + 1 = 09)
BD < [07] (=JNZ 05)	Pagina_54 (EXECUTA...)
RI < BD (=JNZ 05)	BC < READ
Pagina_39 (ATUALIZA)	BE < SP (=FE)
IP < IP + 1 (=07 + 1 = 08)	IP < [FE] (=04)
Pagina_40 (EXECUTA)	Pagina_55 (...EXECUTA)
(PSW(Z) = 0 resultado diferente de zero)	SP < SP + 1 (=FE + 1 = FF)
IP < 05 (JNZ 05)	Pagina_56 (BUSCA)
Pagina_41 (BUSCA)	BE < IP (=04)
BE < IP (=05)	BC < READ
BC < READ	BD < [04] (=RET)
BD < [05] (=ADD A,B)	RI < BD (=RET)
RI < BD (=ADD A,B)	Pagina_57 (ATUALIZA)
Pagina_42 (ATUALIZA)	IP < IP + 1 (=04 + 1 = 05)
IP < IP + 1 (=05 + 1 = 06)	Pagina_58 (EXECUTA...)
Pagina_43 (EXECUTA...)	BC < READ
R1 < A (=14)	BE < SP (=FF)
R2 < B (=07)	IP < [FF] (=61)
R3 < R1 + R2 (=21)	Pagina_59 (...EXECUTA)
Pagina_44 (...EXECUTA)	SP < SP + 1 (=FF + 1 = 00)
A < R3 (=21)	Pagina_60 (BUSCA)
Pagina_45 (BUSCA)	BE < IP (=61)
BE < IP (=06)	BC < READ
BC < READ	BD < [61] (=MOV [64],A)
BD < [06] (=DEC C)	RI < BD (=MOV [64],A)
RI < BD (=DEC C)	Pagina_61 (ATUALIZA)
Pagina_46 (ATUALIZA)	IP < IP + 1 (=61 + 1 = 62)
IP < IP + 1 (=06 + 1 = 07)	Pagina_62 (EXECUTA)
	BE < 64 (endereço destino MOV [64],A)
	BC < WRITE
	BD < A
	[64] < BD (=21)