

Lista rozkazów mikroprocesora Z80

Przyjęte zostały następujące oznaczenia:

- n — argument bezpośredni 8-bitowy
- nn — argument bezpośredni 16-bitowy
- $()$ — zawartość komórki pamięci
- b — numer bitu
- d — przesunięcie w adresowaniu indeksowym
- e — przesunięcie w adresowaniu względnym
- $r, w = A, B, C, D, E, H, L$
- $s = A, B, C, D, E, H, L, (HL), (IX + d), (IY + d), n$
- $dd, ss = BC, DE, HL, SP$
- $qq = BC, DE, HL, AF$
- $pp = BC, DE, IX, SP$
- $rr = BC, DE, IY, SP$
- cc — warunek skoku
- $cc = NZ, Z, NC, C, PO, PE, P, M$

Ustawianie znaczników:

- x — wartość znacznika nieznaną
- $\bullet$ — rozkaz nie zmienia wartości znacznika
- $+$ — znacznik ustawiany zgodnie z wynikiem operacji
- 0 — znacznik zerowany
- 1 — znacznik ustawiany
- V — znacznik P/V ustawiany ze względu na nadmiar
- P — znacznik P/V ustawiany ze względu na parzystość

W kodach rozkazów:

<i>r, w</i>	Rejestr	<i>dd, ss</i>	Para rejestrów	<i>qq</i> <i>qq</i>	Para rejestrów
000	B	00	BC	00	BC
001	C	01	DE	01	DE
010	D	10	HL	10	HL
011	E	11	SP	11	AF
100	H				
101	L				
111	A				

	<i>pp</i>	Para rejestrów	<i>rr</i>	Para rejestrów
	00	BC	00	BC
	01	DE	01	DE
	10	IX	10	IY
	11	SP	11	SP

<i>cc</i>	Oznaczenie	Warunek	<i>b</i>	Nr bitu
000	NZ	$Z = 0$	000	0
001	Z	$Z = 1$	001	1
010	NC	$C = 0$	010	2
011	C	$C = 1$	011	3
100	PO	$P = 0$	100	4
101	PE	$P = 1$	101	5
110	P	$S = 0$	110	6
111	M	$S = 1$	111	7

B – liczba bajtów rozkazu

C – liczba cykli maszynowych

T – liczba taktów zegara

Mnemonic Z80	Operacja	Znaczniki S Z H P N C	Kod		B	C	T	Mnemonic 8080
			Binarny 76 543 210	Heksade- cymalny				
Rozkazy przesłań 8-bitowych								
LD A, (BC)	A ← (BC)	. . x . x . . .	00 001 010	0A	1	2	7	LDAX B
LD A, (DE)	A ← (DE)	. . x . x . . .	00 011 010	1A	1	2	7	LDAX D
LD A, (nn)	A ← (nn)	. . x . x . . .	00 111 010 ← n → ← n →	3A	3	4	13	LDA nn
LD (BC), A	(BC) ← A	. . x . x . . .	00 000 010	02	1	2	7	STAX B
LD (DE), A	(DE) ← A	. . x . x . . .	00 010 010	12	1	2	7	STAX D
LD (nn), A	(nn) ← A	. . x . x . . .	00 110 010 ← n → ← n →	32	3	4	13	STA nn
LD A, I	A ← I	+ + x 0 x IFF0 .	11 101 101 01 010 111	ED 57	2	2	9	
LD A, R	A ← R	+ + x 0 x IFF0 .	11 101 101 01 011 111	ED 5F	2	2	9	
LD I, A	I ← A	. . x . x . . .	11 101 101 01 000 111	ED 47	2	2	9	
LD R, A	R ← A	. . x . x . . .	11 101 101 01 001 111	ED 4F	2	2	9	
Rozkazy przesłań 16-bitowych								
LD dd, nn	dd ← nn	. . x . x . . .	00 dd0 001 ← n → ← n →		3	3	10	LXI dd, nn
LD IX, nn	IX ← nn	. . x . x . . .	11 011 101 00 100 001 ← n → ← n →	DD 21	4	4	14	

LD IY, nn	IY $\leftarrow$ nn	. . x . x . . .	11 111 101 00 100 001 $\leftarrow$ n $\rightarrow$ $\leftarrow$ n $\rightarrow$	FD 21	4	4	14	LHLD nn
LD HL, (nn)	H $\leftarrow$ (nn + 1) L $\leftarrow$ (nn)	. . x . x . . .	00 101 010 $\leftarrow$ n $\rightarrow$ $\leftarrow$ n $\rightarrow$	2A	3	5	16	
LD dd, (nn)	dd _H $\leftarrow$ (nn + 1) DD _L $\leftarrow$ (nn)	. . x . x . . .	11 101 101 01 dd1 011 $\leftarrow$ n $\rightarrow$ $\leftarrow$ n $\rightarrow$	ED	4	6	20	
LD IX, (nn)	IX _H $\leftarrow$ (nn + 1) IX _L $\leftarrow$ (nn)	. . x . x . . .	11 011 101 11 101 010 $\leftarrow$ n $\rightarrow$ $\leftarrow$ n $\rightarrow$	DD 2A	4	6	20	
LD IY, (nn)	IY _H $\leftarrow$ (nn + 1) IY _L $\leftarrow$ (nn)	. . x . x . . .	11 111 101 00 101 010 $\leftarrow$ n $\rightarrow$ $\leftarrow$ n $\rightarrow$	FD 2A	4	6	20	SHLD nn
LD (nn), HL	(nn + 1) $\leftarrow$ H (nn) $\leftarrow$ L	. . x . x . . .	00 100 010 $\leftarrow$ n $\rightarrow$ $\leftarrow$ n $\rightarrow$	22	3	5	16	
LD (nn), dd	(nn + 1) $\leftarrow$ dd _H (nn) $\leftarrow$ dd _L	. . x . x . . .	11 101 101 01 dd0 011 $\leftarrow$ n $\rightarrow$ $\leftarrow$ n $\rightarrow$	ED	4	6	20	
LD (nn), IX	(nn + 1) $\leftarrow$ IX _H (nn) $\leftarrow$ IX _L	. . x . x . . .	11 011 101 00 100 010 $\leftarrow$ n $\rightarrow$ $\leftarrow$ n $\rightarrow$	DD 22	4	6	20	
LD (nn), IY	(nn + 1) $\leftarrow$ IY _H (nn) $\leftarrow$ IY _L	. . x . x . . .	11 111 101 00 100 010 $\leftarrow$ n $\rightarrow$ $\leftarrow$ n $\rightarrow$	FD 22	4	6	20	SPHL
LD SP, HL	SP $\leftarrow$ HL	. . x . x . . .	11 111 001	F9	1	1	6	

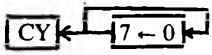
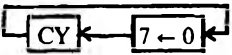
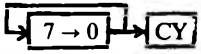
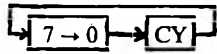
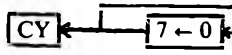
Mnemonic Z80	Operacja	Znaczniki S Z H P NC	Kod		B	C	T	Mnemonic 8080
			Binarny 76 543 210	Heksade- cymalny				
LD SP, IX	SP ← IX	. . x . x . . .	11 011 101	DD	2	2	10	PUSH qq
LD SP, IY	SP ← IY	. . x . x . . .	11 111 001	F9	2	2	10	
PUSH qq	(SP-2) ← qq _L	. . x . x . . .	11 qq0 101	F9	1	3	11	
	(SP-1) ← qq _H							
PUSH IX	(SP-2) ← IX _L	. . x . x . . .	11 011 101	DD	2	4	15	
	(SP-1) ← IX _H		11 100 101	E5				
PUSH IY	(SP-2) ← IY _L	. . x . x . . .	11 111 101	FD	2	4	15	
	(SP-1) ← IY _H		11 100 101	E5				
POP qq	qq _H ← (SP+1)	. . x . x . . .	11 qq0 001		1	3	10	POP qq
	qq _L ← (SP)							
POP IX	IX _H ← (SP+1)	. . x . x . . .	11 011 101	DD	2	4	14	
	IX _L ← (SP)		11 100 001	E1				
POP IY	IY _H ← (SP+1)	. . x . x . . .	11 111 101	FD	2	4	14	
	IY _L ← (SP)		11 100 001	E1				
Rozkazy wymian i przesyłań blokowych								
EX DE, HL	DE ↔ HL	. . x . x . . .	11 101 011	EB	1	1	4	XCHG
EX AF, AF'	AF ↔ AF'	. . x . x . . .	00 001 000	08	1	1	4	
EXX	BC ↔ BC'	. . x . x . . .	11 011 001	D9	1	1	4	
	DE ↔ DE'							
	HL ↔ HL'							
EX (SP), HL	H ↔ (SP+1)	. . x . x . . .	11 100 011	E3	1	5	19	XTHL
	L ↔ (SP)							
EX (SP), IX	IX _H ↔ (SP+1)	. . x . x . . .	11 011 101	DD	2	6	23	
	IX _L ↔ (SP)		11 100 011	E3				
EX (SP), IY	IY _H ↔ (SP+1)	. . x . x . . .	11 111 101	FD	2	6	23	
	IY _L ↔ (SP)		11 100 011	E3				

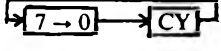
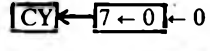
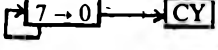
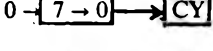
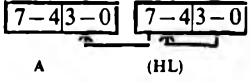
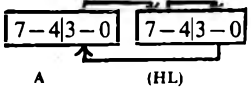
LDI	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC - 1	$\dots x 0 x + 0$ ¹⁾	11 101 101 10 100 000	ED A0	2	4	16	
LDIR	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC - 1 powtarzaj aż BC = 0	$\dots x 0 x 0 0$	11 101 101 10 110 000	ED B0	2	5 4	21 16	gdy BC ≠ 0 gdy BC = 0
LDD	(DE) ← (HL) DE ← DE - 1 HL ← HL - 1 BC ← BC - 1	$\dots x 0 x + 0$ ¹⁾	11 101 101 10 101 000	ED A8	2	4	16	
LDDR	(DE) ← (HL) DE ← DE - 1 HL ← HL - 1 BC ← BC - 1 powtarzaj aż BC = 0	$\dots x 0 x 0 0$	11 101 101 10 111 000	ED B8	2	5 4	21 16	gdy BC ≠ 0 gdy BC = 0
CPI	A ← (HL) HL ← HL + 1 BC ← BC - 1	$++ x + x + 1$ ²⁾ ¹⁾	11 101 101 10 100 001	ED A1	2	4	16	
CPIR	A ← (HL) HL ← HL + 1 BC ← BC - 1 powtarzaj aż BC = 0	$++ x + x + 1$ ²⁾ ¹⁾	11 101 101 10 110 001	ED B1	2	5 4	21 16	gdy BC ≠ 0 i A ≠ (HL) gdy BC = 0 lub A = (HL)
CPD	lub A = (HL) A ← (HL) HL ← HL - 1 BC ← BC - 1	$++ x + x + 1$ ²⁾ ¹⁾	11 101 101 10 101 001	ED A9	2	4	16	

Mnemonic Z80	Operacja	Znaczniiki S Z H P NC	Kod		B	C	T	Mnemonic 8080
			Binarny 76 543 210	Heksade- cymalny				
CPDR	A ← (HL) HL ← HL - 1 BC ← BC - 1 powtarzaj aż BC = 0 lub A = (HL)	²⁾ + + x + x + 1 . ¹⁾	11 101 101 10 111 001	ED B9	2	5 4	21 16	gdy BC ≠ 0 i A ≠ (HL) gdy BC = 0 lub A = (HL)
¹⁾ P/V = 0, gdy wynik BC - 1 równy 0, inaczej P/V = 1 ²⁾ Z = 1, jeżeli A = (HL), inaczej Z = 0								
Rozkazy arytmetyczno-logiczne. Arytmetyka 8-bitowa.								
ADD A, r	A ← A + r	+ + x + x V 0 +	10 000 r		1	1	4	ADD r
ADD A, n	A ← A + n	+ + x + x V 0 +	11 000 110 ← n →	C6	2	2	7	ADI n
ADD A, (HL)	A ← A + (HL)	+ + x + x V 0 +	10 000 110	86	1	2	7	ADD M
ADD A, (IX + d)	A ← A + (IX + d)	+ + x + x V 0 +	11 011 101 10 000 110 ← d →	DD 86	3	5	19	
ADD A, (IY + d)	A ← A + (IY + d)	+ + x + x V 0 +	11 111 101 10 000 110 ← d →	FD 86	3	5	19	
ADC A, s	A ← A + s + CY	+ + x + x V 0 +	← d → 001					ADC r ACI n ADC M

Mnemonic Z80	Operacja	Znaczniki S Z H P NC	Kod		B	C	T	Mnemonic 8080
			Binarny 76 543 210	Heksade- cymalny				
Pomocnicze rozkazy arytmetyczne i rozkazy sterujące								
DAA	korekcja dziesiętna	++ x + x P . +	00 100 111	27	1	1	4	DAA
CPL	$A \leftarrow \bar{A}$	. . x 1 x . 1 .	00 101 111	2F	1	1	4	CMA
NEG	$A \leftarrow \bar{A} + 1$	++ x + x V 1 +	11 101 101 01 000 100	ED 44	2	2	8	
CCF	$CY \leftarrow \overline{CY}$	. . x x x . 0 +	00 111 111	3F	1	1	4	CMC
SCF	$CY \leftarrow 1$	. . x 0 x . 0 1	00 110 111	37	1	1	4	STC
NOP	nic nie rób	. . x . x . . .	00 000 000	00	1	1	4	NOP
HALT		. . x . x . . .	01 110 110	76	1	1	4	HLT
DI	$IFF \leftarrow 0$	. . x . x . . .	11 110 011	F3	1	1	4	DI
EI	$IFF \leftarrow 1$	. . x . x . . .	11 111 011	FB	1	1	4	EI
IM 0	ustawienie trybu 0	. . x . x . . .	11 101 101	ED	2	2	8	
	obsługi przerwań		01 000 110	46				
IM 1	ustawienie trybu 1	. . x . x . . .	11 101 101	ED	2	2	8	
	obsługi przerwań		01 010 110	56				
IM2	ustawienie trybu 2	. . x . x . . .	11 101 101	ED	2	2	8	
	obsługi przerwań		01 011 110	5E				
Rozkazy arytmetyki 16-bitowej								
ADD HL, ss	$HL \leftarrow HL + ss$	. . x x x . 0 +	00 ss1 001		1	3	11	DAD ss
ADC HL, ss	$HL \leftarrow HL + ss + CY$	++ x x x V 0 +	11 101 101 01 ss1 010	ED.	2	4	15	
SBC HL, ss	$HL \leftarrow HL - ss - CY$	++ x x x V 1 +	11 101 101 01 ss0 010	ED	2	4	15	
ADD IX, pp	$IX \leftarrow IX + pp$	. . x x x . 0 +	11 011 101 00 pp1 001	DD	2	4	15	

ADD Y, r	$Y \leftarrow Y + r$	. . x x x . 0 +	11 111 101 00 rr 1 001	FD	2	4	15	INX ss
INC ss	$ss \leftarrow ss + 1$	. . x . x . . .	00 ss 0 011	DD	1	1	6	
INC IX	$IX \leftarrow IX + 1$	. . x . x . . .	11 011 101 00 100 011	23	2	2	10	
INC Y	$Y \leftarrow Y + 1$	. . x . x . . .	11 111 101 00 100 011	FD 23	2	2	10	
DEC ss	$ss \leftarrow ss - 1$	. . x . x . . .	00 ss 1 011	DD	1	1	6	
DEC IX	$IX \leftarrow IX - 1$	. . x . x . . .	11 011 101 00 101 011	2B	2	2	10	
DEC Y	$Y \leftarrow Y - 1$	. . x . x . . .	11 111 101 00 101 011	FD 2B	2	2	10	DCX ss
Rozkazy działań na bitach .								
BIT b, r	$Z \leftarrow \bar{r}_b$	x + x 1 x x 0 .	11 001 011 01 b r	CB	2	2	8	
BIT $b, (HL)$	$Z \leftarrow (\overline{HL})_b$	x + x 1 x x 0 .	11 001 011 01 b 110	CB	2	3	12	
BIT $b, (IX+d)$	$Z \leftarrow (\overline{IX+d})_b$	x + x 1 x x 0 .	11 011 101 11 001 011	DD CB	4	5	20	
			$\leftarrow d \rightarrow$ 01 b 110					
BIT $b, (Y+d)$	$Z \leftarrow (\overline{Y+d})_b$	x + x 1 x x 0 .	11 111 101 11 001 011	FD CB	4	5	20	
			$\leftarrow d \rightarrow$ 01 b 110					
SET b, r	$r_b \leftarrow 1$	. . x . x . . .	11 001 011 11 b r	CB	2	2	8	
SET $b, (HL)$	$(HL)_b \leftarrow 1$	. . x . x . . .	11 001 011 11 b 110	CB	2	4	15	
SET $b, (IX+d)$	$(IX+d)_b \leftarrow 1$	. . x . x . . .	11 011 101 11 001 011	DD CB	4	6	23	
			$\leftarrow d \rightarrow$ 11 b 110					

Mnemonik Z80	Operacja	Znaczni S Z H P NC	Kod		B	C	T	Mnemonik 8080
			Binarny 76 543 210	Heksade- cymalny				
SET $b, (IY+d)$	$(IY+d)_b \leftarrow 1$	. . x . x . . .	11 111 101 11 001 011 $\leftarrow d \rightarrow$	FD CB	4	6	23	
RES b, s	$s_b \leftarrow 0$	. . x . x . . .	<u>11</u> b 110 <u>10</u>					
Kod rozkazu RES otrzyma się wstawiając <u>10</u> w miejsce <u>11</u> w rozkazach SET								
Rozkazy przesunięć								
RLCA		. . x 0 x . 0 +	00 000 111	07	1	1	4	RLC
RLA		. . x 0 x . 0 +	00 010 111	17	1	1	4	RAL
RRCA		. . x 0 x . 0 +	00 001 111	0F	1	1	4	RRC
RRA		. . x 0 x . 0 +	00 011 111	1F	1	1	4	RAR
RLC r		++ x 0 x P 0 +	11 001 011 00 <u>000</u> r	CB	2	2	8	
RLC (HL)		++ x 0 x P 0 +	11 001 011 00 <u>000</u> 110	CB	2	4	15	

RLC (IX+d)		++ x 0 x P 0 +	11 011 101 11 001 011 ← d → 00 000 110	DD CB	4	6	23	
RLC (IY+d)		++ x 0 x P 0 +	11 111 101 11 001 011 ← d → 00 000 110	FD CB	4	6	23	
RL s		++ x 0 x P 0 +	010		Kody rozkazów otrzymuje się wy- mieniając pole 000 w odpowied- nim rozkazie RLC.			
RRC s		++ x 0 x P 0 +	001					
RR s		++ x 0 x P 0 +	011					
SLA s		++ x 0 x P 0 +	100					
SRA s		++ x 0 x P 0 +	101					
SRL s		++ x 0 x P 0 +	111					
RLD		++ x 0 x P 0 .	11 101 101 01 101 111	ED 6F	2	5	18	
RRD		++ x 0 x P 0 .	11 101 101 01 100 111	ED 67	2	5	18	

Mnemonic Z80	Operacja	Znaczniki S Z H P NC	Kod		B	C	T	Mnemonic 8080
			Binarny 76 543 210	Heksade- cymalny				
Rozkazy skoków								
JP <i>nn</i>	PC ← <i>nn</i>	. . x . x . . .	11 000 011 ← <i>n</i> →	C3	3	3	10	JMP <i>nn</i>
JP <i>cc, nn</i>	jeżeli warunek spełniony, PC ← <i>nn</i> , inaczej PC ← PC+3	. . x . x . . .	11 <i>cc</i> 010 ← <i>n</i> →		3	3	10	Jcc <i>nn</i>
JR <i>e</i>	PC ← PC+ <i>e</i>	. . x . x . . .	00 011 000 ← <i>e</i> -2 →	18	2	3	12	
JR C, <i>e</i>	gdy C = 1, PC ← PC+ <i>e</i> gdy C = 0, PC ← PC+2	. . x . x . . .	00 111 000 ← <i>e</i> -2 →	38	2	3	12	
JR NC, <i>e</i>	gdy C = 0, PC ← PC+ <i>e</i> gdy C = 1, PC ← PC+2	. . x . x . . .	00 110 000 ← <i>e</i> -2 →	30	2	3	12	
JR Z, <i>e</i>	gdy Z = 1 PC ← PC+ <i>e</i> gdy Z = 0 PC ← PC+2	. . x . x . . .	00 101 000 ← <i>e</i> -2 →	28	2	3	12	
JR NZ, <i>e</i>	gdy Z = 0 PC ← PC+ <i>e</i> gdy Z = 1 PC ← PC+2	. . x . x . . .	00 100 000 ← <i>e</i> -2 →	20	2	3	12	
JP (HL)	PC ← HL	. . x . x . . .	11 101 001	E9	1	1	4	PCHL
JP (IX)	PC ← IX	. . x . x . . .	11 011 101	DD	2	2	8	
			11 101 001	E9				
JP (IY)	PC ← IY	. . x . x . . .	11 111 101	FD	2	2	8	
			11 101 001	E9				

DJNZ e	$B \leftarrow B - 1$ gdy $B \neq 0$ $PC \leftarrow PC + e$ gdy $B = 0$ $PC \leftarrow PC + 2$	. . x . x . . .	00 010 000 $\leftarrow e - 2 \rightarrow$	10	2	3	13	
Rozkazy skoków do podprogramów i powrotów z podprogramów								
CALL nn	$(SP - 1) \leftarrow PC_H$ $(SP - 2) \leftarrow PC_L$ $PC \leftarrow nn$	. . x . x . . .	11 001 101 $\leftarrow n \rightarrow$	CD	3	5	17	CALL nn
CALL cc, nn	gdy warunek spełniony, to jak CALL	. . x . x . . .	11 cc 100 $\leftarrow n \rightarrow$		3	5	17	CALL cc, nn
RET	$PC_L \leftarrow (SP)$ $PC_H \leftarrow (SP + 1)$	. . x . x . . .	11 001 001 $\leftarrow n \rightarrow$	C9	1	3	10	RET
RET cc	gdy warunek spełniony, to jak RET	. . x . x . . .	11 cc 000		1	3	11	RET cc
RETI	powrót z przerwania	. . x . x . . .	11 101 101 01 001 101	ED 4D	2	4	14	
RETN	powrót z przerwania niemaskowalnego	. . x . x . . .	11 101 101 01 000 101	ED 45	2	4	14	
RST p	$(SP - 1) \leftarrow PC_H$ $(SP - 2) \leftarrow PC_L$ $PC \leftarrow 8p$	. . x . x . . .	11 p 111		1	3	11	RST p
Rozkazy wejścia-wyjścia								
IN $A, (n)$	$A \leftarrow (n)$	. . x . x . . .	11 011 011 $\leftarrow n \rightarrow$	DB	2	3	11	IN n
IN $r, (C)$	$r \leftarrow (C)$ gdy $r = 110$, tylko usta- wienie znaczników	+ + x + x P 0 .	11 101 101 01 r 000	ED	2	3	12	
INI	$(HL) \leftarrow (C)$ $B \leftarrow B - 1$ $HL \leftarrow HL + 1$	x + x x x x 1 x	11 101 101 10 100 010	ED A2	2	4	16	

Mnemonik Z80	Operacja	Znaczniki S Z H P NC	Kod		B	C	T	Mnemonik 8080
			Binarny 76 543 210	Heksade- cymalny				
INIR	(HL) ← (C) B ← B-1 HL ← HL+1 powtarzaj aż B = 0	x 1 x x x x 1 x	11 101 101 10 110 010	ED B2	2	5 4	21 16	gdy B ≠ 0 gdy B = 0
IND	(HL) ← (C) B ← B-1 HL ← HL-1	x ^p x x x x 1 x	11 101 101 10 101 010	ED AA	2	4	16	
INDR	(HL) ← (C) B ← B-1 HL ← HL-1 powtarzaj aż B = 0	x 1 x x x x 1 x	11 101 101 10 111 010	ED BA	2	5 4	21 16	gdy B ≠ 0 gdy B = 0
OUT (n), A	(n) ← A	. . x . x . . .	11 010 011 ← n →	D3	2	3	11	OUT n
OUT (C), r	(C) ← r	. . x . x . . .	11 101 101 01 r 001	ED	2	3	12	
OUTI	(C) ← (HL) B ← B-1 HL ← HL+1	x ^p x x x x 1 x	11 101 101 10 100 011	ED A3	2	4	16	
OTIR	(C) ← (HL) B ← B-1 HL ← HL+1 powtarzaj aż B = 0	x 1 x x x x 1 x	11 101 101 10 110 011	ED B3	2	5 4	21 16	gdy B ≠ 0 gdy B = 0
OUTD	(C) ← (HL) B ← B-1 HL ← HL-1	x ^p x x x x 1 x	11 101 101 10 101 011	ED AB	2	4	16	
OTDR	(C) ← (HL) B ← B-1 HL ← HL-1 powtarzaj aż B = 0	x 1 x x x x 1 x	11 101 101 10 111 011	ED BB	2	5 4	21 16	gdy B ≠ 0 gdy B = 0

¹⁾ jeżeli wynik B-1 jest zero, to Z = 1, inaczej Z = 0.

Dla rozkazów IN A, (n) i OUT (n), A jest:

n → AD₀ ~ AD₃

A → AD₀ ~ AD₃

Dla pozostałych rozkazów wejścia-wyjścia:

C → AD₀ ~ AD₃

B → AD₀ ~ AD₃