

Z80命令説明書

(有)中日電工

[はじめに]	1	52. EX AF, AF'	4
I. レジスタについて	1	53. EXX	4
II. 8ビット転送命令	1	54. EX (SP),HL	4
1. LD r, r'	1	55. EX (SP),IX	5
2. LD r,(HL)	1	56. EX (SP),IY	5
3. LD (HL),r	1	◎ フラグについての説明	5
4. LD r, n	1	57. LDI	5
5. LD (HL),n	1	58. LDIR	5
6. LD A,(BC)	2	59. LDD	5
7. LD A,(DE)	2	60. LDDR	5
8. LD A,(nn)	2	61. CPI	5
9. LD (BC),A	2	62. CPIR	5
10. LD (DE),A	2	63. CPD	6
11. LD (nn),A	2	64. CPDR	6
12. LD r,(IX+d)	2	V. 8ビット算術・論理演算命令	6
13. LD r,(IY+d)	2	65. ADD A,r	6
14. LD (IX+d),r	2	66. ADD A,n	6
15. LD (IY+d),r	2	67. ADD A,(HL)	6
16. LD (IX+d),n	2	68. ADD A,(IX+d)	6
17. LD (IY+d),n	2	69. ADD A,(IY+d)	6
III. 16ビット転送命令	2	70. ADC A,r	6
18. LD BC,nn	2	71. ADC A,n	6
19. LD DE,nn	3	72. ADC A,(HL)	6
20. LD HL,nn	3	73. ADC A,(IX+d)	6
21. LD SP,nn	3	74. ADC A,(IY+d)	6
22. LD IX,nn	3	75. SUB r	6
23. LD IY,nn	3	76. SUB n	6
24. LD HL,(nn)	3	77. SUB (HL)	6
25. LD BC,(nn)	3	78. SUB (IX+d)	7
26. LD DE,(nn)	3	79. SUB (IY+d)	7
27. LD SP,(nn)	3	80. SBC A,r	7
28. LD IX,(nn)	3	81. SBC A,n	7
29. LD IY,(nn)	3	82. SBC A,(HL)	7
30. LD (nn),HL	3	83. SBC A,(IX+d)	7
31. LD (nn),BC	3	84. SBC A,(IY+d)	7
32. LD (nn),DE	3	85. AND r	7
33. LD (nn),SP	3	86. AND n	7
34. LD (nn),IX	3	87. AND (HL)	7
35. LD (nn),IY	3	88. AND (IX+d)	7
36. LD SP, HL	3	89. AND (IY+d)	7
37. LD SP, IX	4	90. OR r	7
38. LD SP, IY	4	91. OR n	8
39. PUSH BC	4	92. OR (HL)	8
40. PUSH DE	4	93. OR (IX+d)	8
41. PUSH HL	4	94. OR (IY+d)	8
42. PUSH AF	4	95. XOR r	8
43. PUSH IX	4	96. XOR n	8
44. PUSH IY	4	97. XOR (HL)	8
45. POP BC	4	98. XOR (IX+d)	8
46. POP DE	4	99. XOR (IY+d)	8
47. POP HL	4	100. CP r	8
48. POP AF	4	101. CP n	8
49. POP IX	4	102. CP (HL)	8
50. POP IY	4	103. CP (IX+d)	8
IV. 交換、ブロック転送、検索命令	4	104. CP (IY+d)	8
51. EX DE, HL	4	105. INC r	8
		106. INC (HL)	9
		107. INC (IX+d)	9

108. INC (IY+d)	9	164. RRC r	12
109. DEC r	9	165. RRC (HL).....	12
110. DEC (HL).....	9	166. RRC (IX+d).....	12
111. DEC (IX+d).....	9	167. RRC (IY+d).....	12
112. DEC (IY+d).....	9	168. RR r	12
VI. その他の算術演算、CPU制御命令 ...	9	169. RR (HL).....	12
113. DAA	9	170. RR (IX+d).....	12
114. CPL	9	171. RR (IY+d).....	12
115. NEG	9	172. SLA r	13
116. CCF	9	173. SLA (HL).....	13
117. SCF	9	174. SLA (IX+d).....	13
118. NOP	9	175. SLA (IY+d).....	13
119. HALT	10	176. SRA r	13
VII. 16ビット算術演算命令	10	177. SRA (HL).....	13
120. ADD HL, BC	10	178. SRA (IX+d).....	13
121. ADD HL, DE	10	179. SRA (IY+d).....	13
122. ADD HL, HL	10	180. SRL r	13
123. ADD HL, SP	10	181. SRL (HL).....	13
124. ADC HL, BC	10	182. SRL (IX+d).....	13
125. ADC HL, DE	10	183. SRL (IY+d).....	13
126. ADC HL, HL	10	184. RLD	13
127. ADC HL, SP	10	185. RRD	13
128. SBC HL, BC	10	IX. ビットセット、リセット・テスト命令	14
129. SBC HL, DE	10	186. BIT b,r	14
130. SBC HL, HL	10	187. BIT b,(HL).....	14
131. SBC HL, SP	10	188. BIT b,(IX+d).....	14
132. ADD IX, BC	10	189. BIT b,(IY+d).....	14
133. ADD IX, DE	10	190. SET b,r	14
134. ADD IX, IX	10	191. SET b,(HL).....	14
135. ADD IX, SP	10	192. SET b,(IX+d).....	14
136. ADD IY, BC	10	193. SET b,(IY+d).....	15
137. ADD IY, DE	11	194. RES b,r	15
138. ADD IY, IY	11	195. RES b,(HL).....	15
139. ADD IY, SP	11	196. RES b,(IX+d).....	15
140. INC BC	11	197. RES b,(IY+d).....	15
141. INC DE	11	X. ジャンプ命令	15
142. INC HL	11	198. JP nn	15
143. INC SP	11	199. JP NZ,nn	15
144. INC IX	11	200. JP Z,nn	15
145. INC IY	11	201. JP NC,nn	15
146. DEC BC	11	202. JP C,nn	15
147. DEC DE	11	203. JP PO,nn	15
148. DEC HL	11	204. JP PE,nn	15
149. DEC SP	11	205. JP P,nn	16
150. DEC IX	11	206. JP M,nn	16
151. DEC IY	11	207. JR e	16
VIII. 回転、シフト命令	11	208. JR C,e	16
152. RLCA	11	209. JR NC,e	16
153. RLA	11	210. JR Z,e	16
154. RRCA	12	211. JR NZ,e	16
155. RRA	12	212. JP (HL)	16
156. RLC r	12	213. JP (IX).....	16
157. RLC (HL).....	12	214. JP (IY).....	16
158. RLC (IX+d).....	12	215. DJNZ e	16
159. RLC (IY+d).....	12	XI. コール、リターン命令	16
160. RL r	12	216. CALL nn	16
161. RL (HL).....	12	217. CALL NZ,nn	16
162. RL (IX+d).....	12	218. CALL Z,nn	17
163. RL (IY+d)	12		

219. CALL NC,nn	17
220. CALL C,nn	17
221. CALL PO,nn	17
222. CALL PE,nn	17
223. CALL P,nn	17
224. CALL M,nn	17
225. RET	17
226. RET NZ	17
227. RET Z	17
228. RET NC	17
229. RET C	17
230. RET PO	17
231. RET PE	17
232. RET P	17
233. RET M	17
234. RETI	17
235. RETN	17
236. RST n	17
X II. 入出力命令	18
237. IN A,(n).....	18
238. IN r,(C).....	18
239. INI	18
240. INIR	18
241. IND	18
242. INDR	18
243. OUT (n),A	18
244. OUT (C),r	18
245. OUTI	18
246. OTIR	18
247. OUTD	18
248. OTDR	18
X III. 割込制御	
249. DI	19
250. EI	19
251. IM 0	19
252. IM 1	19
253. IM 2	19

〒463-0067 名古屋市守山区守山2-8-14
パレス守山305
有限会社中日電工
TEL052-791-6254 Fax052-791-1391
E-mail thisida@alles.or.jp

2010. 3. 20 Rev. 1. 0

[はじめに]

Z80はザイログ社が開発した8ビットCPUで、8080(インテル社)の命令の全てを含み、さらにその上全く新しい多くの命令を実行することができます。8080の命令については全く同じ命令コードで8080と同じ動作をしますが、Z80と8080はメーカーが異なるため同じ命令コードであってもニーモニックは異なっています。

例) コード7Eのときザイログ社は LD A, (HL) インテル社は MOV A, M と表記します。

この説明書ではZ80ニーモニック(ザイログニーモニック)に従って説明してありますが、同時に8080ニーモニック(インテルニーモニック)も併記してあります。説明の後ろに()で記してあるのがインテルニーモニックです。インテルニーモニックが記してない命令は8080には無くて新たにZ80で追加された命令です。

I. レジスタについて

8ビットレジスタB、C、D、E、H、L、A、F 16ビットレジスタSP、PCについては8080と全く同じです。Z80にはこの他に裏レジスタとしてB'、C'、D'、E'、H'、L'、A'、F'がありEX命令で交互に使用することができますが、初心者が使用すると混乱しますので慣れるまでは使わない方が良いでしょう。この他にインデックスレジスタ IX、IY(16ビット)があります。これによって相対アドレッシングが可能になっています。この他にレジスタとRレジスタがありますが普通の処理では使いません。

II. 8ビット転送命令

[注]以下の説明中、クロックというのはその命令を実行するのに必要なサイクル数のことで、CPUクロックが6MHzのとき、1クロックは1/6μS(0.167μS)になります。したがってたとえばクロック=7の命令の実行時間は0.167×7=1.134μSになります。

1. LD r, r' コード(表1) クロック4
r、r'はB、C、D、E、H、L、Aを表します。r'の内容をrに転送します。r'の内容は変化しません。(MOV r1, r2)

2. LD r, (HL) コード(表2) クロック7
(HL)はHLレジスタで示されるメモリを表します。(HL)の内容をrに転送します。(HL)の内容は変化しません。(MOV r, M)

		r'						
		B	C	D	E	H	L	A
r	B	4 0	4 1	4 2	4 3	4 4	4 5	4 7
	C	4 8	4 9	4 A	4 B	4 C	4 D	4 F
	D	5 0	5 1	5 2	5 3	5 4	5 5	5 7
	E	5 8	5 9	5 A	5 B	5 C	5 D	5 F
	H	6 0	6 1	6 2	6 3	6 4	6 5	6 7
	L	6 8	6 9	6 A	6 B	6 C	6 D	6 F
	A	7 8	7 9	7 A	7 B	7 C	7 D	7 F

r	
B	4 6
C	4 E
D	5 6
E	5 E
H	6 6
L	6 E
A	7 E

r	
B	7 0
C	7 1
D	7 2
E	7 3
H	7 4
L	7 5
A	7 7

3. LD (HL), r コード(表3) クロック7
rの内容を(HL)に転送します。rの内容は変化しません。(MOV M, r)

4. LD r, n コード(表4) クロック7
nはこの命令コードの次のアドレスに書かれたデータを示します。8ビットデータnをrに転送します。(MVI r, n)

5. LD (HL), n コード 36×× クロック10
nを(HL)に転送します(××は任意の8ビットデータを表します)。(MVI M, n)

r	
B	0 6 × ×
C	0 E × ×
D	1 6 × ×
E	1 E × ×
H	2 6 × ×
L	2 E × ×
A	3 E × ×

r	
B	DD 4 6 × ×
C	DD 4 E × ×
D	DD 5 6 × ×
E	DD 5 E × ×
H	DD 6 6 × ×
L	DD 6 E × ×
A	DD 7 E × ×

r	
B	FD 4 6 × ×
C	FD 4 E × ×
D	FD 5 6 × ×
E	FD 5 E × ×
H	FD 6 6 × ×
L	FD 6 E × ×
A	FD 7 E × ×

表5、表6の××はインデックスレジスタに対する増分dを示します

6. LD A, (BC) コード 0A クロック7
 (BC)はBCレジスタで示されるメモリを表します。(BC)の内容をAに転送します。(BC)の内容は変化しません。(LDAX B)
7. LD A, (DE) コード 1A クロック7
 (DE)はDEレジスタで示されるメモリを表します。(DE)の内容をAに転送します。(DE)の内容は変化しません。(LDAX D)
8. LD A, (nn) コード 3A×××× クロック13
 (nn)は命令コード3Aに続く2バイトデータで示されるアドレスのメモリを表します。(nn)の内容をAに転送します。(nn)の内容は変化しません。(LDA nn)
 この命令のように2バイトのデータを扱う場合には先に来るほうが下位アドレスであとに上位アドレスがきます。たとえば1234番地の内容を転送する場合のコードは3A3412になります。
9. LD (BC), A コード 02 クロック7
 Aの内容を(BC)に転送します。Aの内容は変化しません。(STAX B)
10. LD (DE), A コード 12 クロック7
 Aの内容を(DE)に転送します。Aの内容は変化しません。(STAX D)
11. LD (nn), A コード 32×××× クロック13
 Aの内容を(nn)に転送します。Aの内容は変化しません。(STAX nn)
 nnの実際の表記については 8. LD A, (nn) の説明を参照してください。
12. LD r, (IX+d) コード (表5) クロック19
 (IX+d)の内容をrに転送します。(IX+d)はインデックスレジスタIXで示されるアドレスに+dしたアドレスのメモリを示します。dは+方向に00~7F、-方向にFF~80までの数が許されます。FFは+FF(10進の255ではなくて符号付数で-1を意味します。00~7FはIXで示すアドレス値にそのまま加算されて真のアドレスを示しますが、FF~80はIXで示すアドレス値に対してFFFF~FF80の加算が行われます。実際には1~80がIXの値から減算されてそれが真のアドレスになります。(IX+d)の内容は変化しません。
13. LD r, (IY+d) コード (表6) クロック19
 IXのかわりにIYを使うだけで動作は 12. と同じです。
14. LD (IX+d),r コード(表7) クロック19
 12. LD r, (IX+d)とは逆に、(IX+d)にrの内容を入れます。
15. LD (IY+d),r コード(表8) クロック19
 13. LD r, (IY+d)とは逆に、(IY+d)にrの内容を入れます。

r	
B	DD70xx
C	DD71xx
D	DD72xx
E	DD73xx
H	DD74xx
L	DD75xx
A	DD77xx

r	
B	FD70xx
C	FD71xx
D	FD72xx
E	FD73xx
H	FD74xx
L	FD75xx
A	FD77xx

表7、表8の××はインデックスレジスタに対する増分dを示します

16. LD (IX+d),n コード DD36××00 クロック19
 (IX+d)にデータnを転送します。(××はインデックスレジスタの増分d, 00は8ビットのデータnを示します。)
17. LD (IY+d),n コード FD36××00 クロック19
 (IY+d)にデータnを転送します。(××はインデックスレジスタの増分d, 00は8ビットのデータnを示します。)

以上の他にLD A,I LD A,R LD I,A LD R,Aの4命令がありますが一般には使いません。ここでは説明を省略します。

Ⅲ. 16ビット転送命令(この命令群もフラグに影響を与えません)

18. LD BC,nn コード 01×××× クロック10
 16ビットのデータnnをペアレジスタBCに転送します。

nnの実際の表記については 8. LD A, (nn) の説明を参照してください。(LXI B, nn)

19. LD DE,nn コード 11×××× クロック10
nnをDEに転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。(LXI D,nn)

20. LD HL,nn コード 21×××× クロック10
nnをHLに転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。(LXI H,nn)

21. LD SP,nn コード 31×××× クロック10
nnをSPに転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。(LXI SP,nn)

22. LD IX,nn コード DD21×××× クロック14
nnをIXに転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

23. LD IY,nn コード FD21×××× クロック14
nnをIYに転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

24. LD HL,(nn) コード 2A×××× クロック16
2バイトのデータnnによって示されるアドレスのメモリ内容をLに、nn+1番地のメモリ内容をHに転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。(LHLD nn)

25. LD BC,(nn) コード ED4B×××× クロック20
(nn)をCへ、(nn+1)をBへ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

26. LD DE,(nn) コード ED5B×××× クロック20
(nn)をEへ、(nn+1)をDへ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

27. LD SP,(nn) コード ED7B×××× クロック20
(nn)、(nn+1)をSPへ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

28. LD IX,(nn) コード DD2A×××× クロック20
(nn)、(nn+1)をIXへ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

29. LD IY,(nn) コード FD2A×××× クロック20
(nn)、(nn+1)をIYへ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

30. LD (nn),HL コード 22×××× クロック16
Lの内容を(nn)へ、Hの内容を(nn+1)へ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。(SHLD nn)

31. LD (nn),BC コード ED43×××× クロック20
Cの内容を(nn)へ、Bの内容を(nn+1)へ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

32. LD (nn),DE コード ED53×××× クロック20
Eの内容を(nn)へ、Dの内容を(nn+1)へ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

33. LD (nn),SP コード ED73×××× クロック20
SPの内容を(nn)、(nn+1)へ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

34. LD (nn),IX コード DD22×××× クロック20
IXの内容を(nn)、(nn+1)へ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

35. LD (nn),IY コード FD22×××× クロック20
IYの内容を(nn)、(nn+1)へ転送します。nnの実際の表記については 8. LD A, (nn) の説明を参照してください。

36. LD SP,HL コード F9 クロック6
HLの内容をSPに転送します。(SPHL)

37. LD SP,IX コード DDF9 クロック10
IXの内容をSPに転送します。
38. LD SP,IY コード FDF9 クロック10
IYの内容をSPに転送します。
39. PUSH BC コード C5 クロック11
ペアレジスタBCの内容をSPで示されるプッシュダウンスタックへ転送します。(SP-1)←B, (SP-2)←C。(PUSH B)
40. PUSH DE コード D5 クロック11
DEの内容をスタックに転送します。(SP-1)←D, (SP-2)←E。(PUSH D)
41. PUSH HL コード E5 クロック11
HLの内容をスタックに転送します。(SP-1)←H, (SP-2)←L。(PUSH H)
42. PUSH AF コード F5 クロック11
アキュムレータAとフラグレジスタFの内容をスタックに転送します。(SP-1)←A, (SP-2)←F。(PUSH PSW)
43. PUSH IX コード DDE5 クロック15
IXの内容をスタックに転送します。
44. PUSH IY コード FDE5 クロック15
IYの内容をスタックに転送します。
45. POP BC コード C1 クロック10
スタックのトップにあるデータがBCに転送されます。C←(SP)、B←(SP+1)。(POP B)
46. POP DE コード D1 クロック10
スタックのトップにあるデータがDEに転送されます。E←(SP)、D←(SP+1)。(POP D)
47. POP HL コード E1 クロック10
スタックのトップにあるデータがHLに転送されます。L←(SP)、H←(SP+1)。(POP H)
48. POP AF コード F1 クロック10
スタックのトップにあるデータがA及びFに転送されます。F←(SP)、A←(SP+1)。(POP PSW)
49. POP IX コード DDE1 クロック14
スタックのトップにあるデータがIXに転送されます。
50. POP IY コード FDE1 クロック14
スタックのトップにあるデータがIYに転送されます。

IV. 交換、ブロック転送、検索命令

51. EX DE,HL コード EB クロック4
ペアレジスタDEとHLの内容を交換します。(XCHG)
52. EX AF,AF' コード 08 クロック4
A及びFとこれに対応する裏レジスタA'、F'とを交換します。なお、プログラム実行のうえで「表レジスタ」と「裏レジスタ」とは全く同等で区別して認識することはできません。したがって不用意にこの命令を用いると混乱を招きます。この点については次のEXX命令も同じです。しかし、スタックを使わないでレジスタの退避に用いることが出来るので使い方によっては便利な命令です。その意味では次のEXX命令はBC, DE, HLの3つのペアレジスタを一度に交換できるため更にメリットが大きくなります。
53. EXX コード D9 クロック4
ペアレジスタBCとBC'、DEとDE'、HLとHL'とを交換します。(52. EX AF, AF' 参照)
54. EX (SP),HL コード E3 クロック19
スタックのトップにあるデータ(2バイト)とペアレジスタHLの内容を交換します。Lと(SP)、Hと(SP+1)の内容を交換します。(XT

63. CPD コード EDA9 クロック16
HLが-1される点を除いて他は61. CPIと同じです。

64. CPDR コード EDB9 クロック 21(BC≠0)、16(BC=0)
HLが-1されていく点を除いて他は62. CPIRと同じです。

V. 8ビット算術・論理演算命令

このグループの命令はINC, DEC命令を除いてすべてアキュムレータ(Aレジスタ)との間で演算されます。また、すべてフラグに影響を与えます。

65. ADD A,r コード (表9) クロック4
アキュムレータAとレジスタrとを加算し結果をAに入れます。この結果C、Z、S、Hがセット、リセットされN=0になります。また、結果がオーバーフローしたときP/V=1、その他のときはP/V=0になります。(ADDr)

66. ADD A,n コード C6×× クロック7
Aとnとを加算し、結果をAに入れます。フラグについては65. ADD A, r と同じです。(ADI n)

(表9)		(表10)	
r		r	
B	8 0	B	8 8
C	8 1	C	8 9
D	8 2	D	8 A
E	8 3	E	8 B
H	8 4	H	8 C
L	8 5	L	8 D
A	8 7	A	8 F

67. ADD A,(HL) コード 86 クロック7
AとHLを加算し、結果をAに入れます。フラグについては65. ADD A,r と同じです。(ADD M)

68. ADD A,(IX+d) コード DD86×× クロック19
Aと(IX+d)を加算し、結果をAに入れます。××は増分dです。フラグについては65. ADD A,r と同じです。

69. ADD A,(IY+d) コード FD86×× クロック19
Aと(IY+d)を加算し、結果をAに入れます。××は増分dです。フラグについては65. ADD A,r と同じです。

70. ADC A,r コード (表10) クロック4
AとレジスタrとさらにキャリーフラグCとを加算し結果をAに入れます。フラグについては65. ADD A, r と同じです。(ADC r)

71. ADC A,n コード CE×× クロック7
AとnとキャリーCとを加算し、Aに入れます。フラグについては65. ADD A, r と同じです。(ACI n)

72. ADC A,(HL) コード 8E クロック7
Aと(HL)とキャリーCとを加算し、Aに入れます。フラグについては65. ADDA,r と同じです。(ADC M)

73. ADC A,(IX+d) コード DD8E×× クロック19
Aと(IX+d)とキャリーCとを加算しAに入れます。××は増分dです。フラグについては65. ADDA,r と同じです。

74. ADC A,(IY+d) コード FD8E×× クロック19
Aと(IY+d)とキャリーCとを加算しAに入れます。フラグについては65. ADDA,r と同じです。

75. SUB r コード (表11) クロック4
Aからrを引いて結果をAに入れます。フラグはN=1になることを除いて65. ADD A, r と同じです。(SUB r)

76. SUB n コード D6×× クロック7
Aからnを引いて結果をAに入れます。フラグは75. SUB rと同じです。(SUI n)

77. SUB (HL) コード 96 クロック7
Aから(HL)を引いて結果をAに入れます。フラグは75. SUB rと同じです。(SUB M)

(表11)

r	
B	9 0
C	9 1
D	9 2
E	9 3
H	9 4
L	9 5
A	9 7

(表12)

r	
B	9 8
C	9 9
D	9 A
E	9 B
H	9 C
L	9 D
A	9 F

78. SUB (IX+d) コード DD96×× クロック19
Aから(IX+d)を引いて結果をAに入れます。フラグは75. SUB rと同じです。
79. SUB (IY+d) コード FD96×× クロック19
Aから(IY+d)を引いて結果をAに入れます。フラグは75. SUB rと同じです。
80. SBC A,r コード (表12) クロック4
Aからrを引き、さらにキャリーCを引いて結果をAに入れます。フラグは75. SUB rと同じです。(SBB r)
81. SBC A,n コード DE×× クロック7
Aからnを引き、さらにキャリーCを引いて結果をAに入れます。フラグは75. SUB rと同じです。(SBI n)
82. SBC A,(HL) コード 9E クロック7
Aから(HL)を引き、さらにキャリーCを引いて結果をAに入れます。フラグは75. SUB rと同じです。(SBB M)
83. SBC A,(IX+d) コード DD9E×× クロック19
Aから(IX+d)を引き、さらにキャリーCを引いて結果をAに入れます。フラグは75. SUB rと同じです。
84. SBC A,(IY+d) コード FD9E×× クロック19
Aから(IY+d)を引き、さらにキャリーCを引いて結果をAに入れます。フラグは75. SUB rと同じです。
85. AND r コード (表13) クロック4
Aとレジスタrとの論理積(AND)をとって結果をAに入れます。フラグはC=0、N=0、H=1になり、結果が0ならZ=1、また、マイナスならS=1になり、結果のうち1のビットが偶数ならばP/V=1となります。(ANA r)

(表13)

r	
B	A 0
C	A 1
D	A 2
E	A 3
H	A 4
L	A 5
A	A 7

(表14)

r	
B	B 0
C	B 1
D	B 2
E	B 3
H	B 4
L	B 5
A	B 7

86. AND n コード E6×× クロック7
AとnとのANDをとって結果をAに入れます。フラグは85. AND rと同じです。(ANI n)
87. AND (HL) コード A6 クロック7
Aと(HL)とのANDをとって結果をAに入れます。フラグは85. AND rと同じです。(ANA M)
88. AND (IX+d) コード DDA6×× クロック19
Aと(IX+d)とのANDをとって結果をAに入れます。フラグは85. AND rと同じです。
89. AND (IY+d) コード FDA6×× クロック19
Aと(IY+d)とのANDをとって結果をAに入れます。フラグは85. AND rと同じです。
90. OR r コード (表14) クロック4
Aとrとの論理和(OR)をとって結果をAに入れます。フラグは、H=0になることを除いては85. AND rと同じです。(ORA r)

91. OR n コード F6×× クロック7
AとnとのORをとって結果をAに入れます。フラグは、90. OR rと同じです。(ORI n)

92. OR (HL) コード B6 クロック7
Aと(HL)とのORをとって結果をAに入れます。フラグは、90. OR rと同じです。(ORA M)

93. OR (IX+d) コード DDB6×× クロック19
Aと(IX+d)とのORをとって結果をAに入れます。フラグは、90. OR rと同じです。

94. OR (IY+d) コード FDB6×× クロック19
Aと(IY+d)とのORをとって結果をAに入れます。フラグは、90. OR rと同じです。

95. XOR r コード (表15) クロック4
Aとrとの排他的論理和(Exclusive OR)をとって結果をAに入れます。フラグは90. OR rと同じです。(XRA r)

(表15)		(表16)	
r		r	
B	A 8	B	B 8
C	A 9	C	B 9
D	A A	D	B A
E	A B	E	B B
H	A C	H	B C
L	A D	L	B D
A	A F	A	B F

96. XOR n コード EE×× クロック7
Aとnとの排他的論理和をとって結果をAに入れます。フラグは90. OR rと同じです。(XRI n)

97. XOR (HL) コード AE クロック7
Aと(HL)との排他的論理和をとって結果をAに入れます。フラグは90. OR rと同じです。(XRA M)

98. XOR (IX+d) コード DDAE×× クロック19
Aと(IX+d)との排他的論理和をとって結果をAに入れます。フラグは90. OR rと同じです。

99. XOR (IY+d) コード FDAE×× クロック19
Aと(IY+d)との排他的論理和をとって結果をAに入れます。フラグは90. ORrと同じです。

100. CP r コード (表16) クロック4
Aとrとを比較します。具体的にはA-rを計算しその結果をフラグによって示します。しかしSUB r命令とは異なりAの内容は変化しません。N=1になります。A-rでポローが生じた時C=1、またA=rのときZ=1、A-rがマイナスの時S=1、A-rの結果、ビット4からのポローがあればH=1、結果がオーバーフローならP/V= 1になります。(CMP r)

101. CP n コード FE×× クロック7
Aとnとを比較します。フラグは100. CP rと同じです。(CPI n)

102. CP (HL) コード BE クロック7
Aと(HL)とを比較します。フラグは100. CP rと同じです。(CMP M)

103. CP (IX+d) コード DDBE×× クロック19
Aと(IX+d)とを比較します。フラグは100. CP rと同じです。

104. CP (IY+d) コード FDBE×× クロック19
Aと(IY+d)とを比較します。フラグは100. CP rと同じです。

105. INC r コード (表17) クロック4
レジスタrの内容を+1します。以下のINC, DEC命令ではキャリーCのみ変化しません。注意して下さい。N=0になります。結果が0ならZ=1、マイナスならS=1、この命令の実行後、結果が80Hならば(つまり実行前の値が7FHならば)P/V=1になります。ビット3からキャリーが出ればH=1になります。(INR r)

r	
B	0 4
C	0 C
D	1 4
E	1 C
H	2 4
L	2 C
A	3 C

r	
B	0 5
C	0 D
D	1 5
E	1 D
H	2 5
L	2 D
A	3 D

106. INC (HL) コード 34 クロック11
 (HL)の内容を+1します。フラグは105. INC rと同じです。(INR M)

107. INC (IX+d) コード DD34 × × クロック23
 (IX+d)の内容を+1します。フラグは105. INC rと同じです。

108. INC (IY+d) コード FD34 × × クロック23
 (IY+d)の内容を+1します。フラグは105. INC rと同じです。

109. DEC r コード (表18) クロック4
 rの内容を-1します。フラグはN=1になります。また、結果が7FH(つまり実行前の値が80H)ならばP/V=1、この他のフラグについては105. INC rと同じです。(DCR r)

110. DEC (HL) コード 35 クロック11
 (HL)の内容を-1します。フラグは109. DEC rと同じです。(DCR M)

111. DEC (IX+d) コード DD35 × × クロック23
 (IX+d)の内容を-1します。フラグは109. DEC rと同じです。

112. DEC (IY+d) コード FD35 × × クロック23
 (IY+d)の内容を-1します。フラグは109. DEC rと同じです。

VI. その他の算術演算、CPU制御命令

113. DAA コード 27 クロック4
 加減算後のレジスタの内容をBCD表示2桁に直します。動作はかなり複雑ですが、8ビットの数をBCD2桁とみなして加減算した後この命令を行うことによって、結果もBCD2桁で得られるものです。したがって、加減算を行う前の数値がBCD表現で誤っている場合(例えばABとか7D)はDAA命令によっても正しい結果は得られません。ここでいう加算とはADD、ADC、INC命令であり、減算とはSUB、SBC、DEC及びNEG(後退)命令です。結果が0のときZ=1、マイナスの時S=1、また、結果のビットが1であるものが偶数個あるときP/V=1、またさらに上位桁へのキャリー、ポローが出た場合はC=1になります。(DAA)

114. CPL コード 2F クロック4
 Aの内容の1、0を反転します。(各ビット毎 1→0、0→1にする)フラグはN=1、H=1他は変化しません。(CMA)

115. NEG コード ED44 クロック8
 0-Aを計算してAに入れます。(Aの符号を逆にする)ただし、80H(-128)は変化しないので注意して下さい。N=1、結果が0ならZ=1、マイナスならS=1、ビット4からポローがあればH=1、命令実行前のAの内容が0ならばC=0、また実行前の内容が80HならばP/V=1になります。

116. CCF コード 3F クロック4
 キャリーフラグCを反転します。同時にN=0になり、Hは実行前のCの内容と同じになります。他のフラグは不変です。(CMC)

117. SCF コード 37 クロック4
 キャリーフラグCをセットします。N=0、H=0になります。(STC)

118. NOP コード 00 クロック4
 何も実行しないで次の命令に進みます。No Operation(NOP)

119. HALT コード 76 クロック4
割込リセットがかかるまでCPUの実行を停止します。(HLT)

VII. 16ビット算術演算命令

120. ADD HL,BC コード 09 クロック11
HLの内容とBCの内容を加算し、結果をHLに入れます。N=0になります。ビット17からのキャリーがあればC=1になります。ビット11からのキャリーがあればH=1になります。(DAD B)

121. ADD HL,DE コード 19 クロック11
HLとDEとを加算し、結果をHLに入れます。フラグについては120. ADD HL, BCと同じです。(DAD D)

122. ADD HL,HL コード 29 クロック11
HLの内容を2倍にします。フラグは120. ADD HL,BCと同じです。(DAD H)

123. ADD HL,SP コード 39 クロック11
HLとSPとを加算し、結果をHLに入れます。フラグは120. ADD HL,BCと同じです。(DAD SP)

124. ADC HL,BC コード ED4A クロック15
HLとBCとキャリーCとを加算し、結果をHLに入れます。N=0、H=不明、ビット15からのキャリーがあればC=1、また結果が0ならばZ=1、マイナスならS=1、オーバーフローならP/V=1になります。

125. ADC HL,DE コード ED5A クロック15
HLとDEとキャリーCとを加算し、結果をHLに入れます。フラグは124. ADC HL, BCと同じです。

126. ADC HL,HL コード ED6A クロック15
HLの内容を2倍し、キャリーCを加えます。フラグは124. ADC HL,BCと同じです。

127. ADC HL,SP コード ED7A クロック15
HLとSPとキャリーCとを加算し、結果をHLに入れます。フラグは124. ADC HL, BCと同じです。

128. SBC HL,BC コード ED42 クロック15
HLからBC及びキャリーCを減算し、結果をHLに入れます。N=0になること以外は、フラグについては124. ADC HL,BCと同じです。

129. SBC HL,DE コード ED52 クロック15
HLからDE及びキャリーCを減算し、結果をHLに入れます。フラグは128. SBC HL,BCと同じです。

130. SBC HL,HL コード ED62 クロック15
HLからHL及びキャリーCを減算し、結果をHLに入れます。フラグは128. SBC HL,BCと同じです。

131. SBC HL,SP コード ED72 クロック15
HLからSP及びキャリーCを減算し、結果をHLに入れます。フラグは128. SBC HL,BCと同じです。

132. ADD IX,BC コード DD09 クロック15
インデックスレジスタIXとBCとの内容を加えIXに入れます。N=0になります。ビット17からのキャリーがあればC=1になります。ビット11からのキャリーがあればH=1になります。

133. ADD IX,DE コード DD19 クロック15
IX+DEの結果をIXに入れます。フラグは132. ADD IX,BCと同じです。

134. ADD IX,IX コード DD29 クロック15
IX+IXの結果をIXに入れます。(IXの内容を倍にする)フラグは132. ADD IX,BCと同じです。

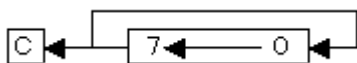
135. ADD IX,SP コード DD39 クロック15
IX+SPの結果をIXに入れます。フラグは132. ADD IX,BCと同じです。

136. ADD IY,BC コード FD09 クロック15
IY+BCの結果をIYに入れます。フラグは132. ADD IX,BCと同じです。

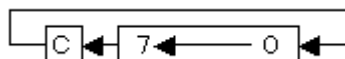
137. ADD IY,DE コード FD19 クロック15
IY+DEの結果をIYに入れます。フラグは132. ADD IX,BCと同じです。
138. ADD IY, IY コード FD29 クロック15
IYの内容を倍にします。フラグは132. ADD IX,BCと同じです。
139. ADD IY,SP コード FD39 クロック15
IY+SPを計算して結果をIYに入れます。フラグは132. ADD IX,BCと同じです。
140. INC BC コード 03 クロック6
BCの内容を+1します。フラグは変化しません。(INX B)
141. INC DE コード 13 クロック6
DEの内容を+1します。(INX D)
142. INC HL コード 23 クロック6
HLの内容を+1します。(INX H)
143. INC SP コード 33 クロック6
SPの内容を+1します。(INX SP)
144. INC IX コード DD23 クロック10
IXの内容を+1します。
145. INC IY コード FD23 クロック10
IYの内容を+1します。
146. DEC BC コード 0B クロック6
BCの内容を-1します。(DCX B)
147. DEC DE コード 1B クロック6
DEの内容を-1します。(DCX D)
148. DEC HL コード 2B クロック6
HLの内容を-1します。(DCX H)
149. DEC SP コード 3B クロック6
SPの内容を-1します。(DCX SP)
150. DEC IX コード DD2B クロック10
IXの内容を-1します。
151. DEC IY コード FD2B クロック10
IYの内容を-1します。

VIII. 回転、シフト命令

152. RLCA コード 07 クロック4
レジスタの内容を1ビット左にシフトし、ビット7のデータをビット0に入れると共にキャリーCにも入れます。N=0、H=0となり、Cはシフト前のビット7が1のときセットされます。(RLC)

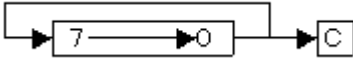


153. RLA コード 17 クロック4
キャリーを含め9ビットのデータを左に1ビット回転させます。フラグは152. RLCAと同じです。(RAL)



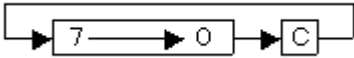
154. RRCA コード 0F クロック4

Aの内容を右に1ビットシフトさせると共にビット0のデータをキャリーCにも入れます。フラグは シフト前のビット0の内容がCに入る以外は152. RLCAと同じです。(RRC)



155. RRA コード 1F クロック4

キャリーを含め9ビットのデータを1ビット右に回転させます。フラグは154. RRCAと同じです。(RAR)



156. RLC r コード (表19) クロック8

152. RLCAと同じ動作を r について行います。

N=0, H=0, Cが変化する他に結果のレジスタの値によってZ, Sがセット、リセットされまた1のビット数によりP/Vがセット、リセットされます。

(表19)

r	
B	CB00
C	CB01
D	CB02
E	CB03
H	CB04
L	CB05
A	CB07

157. RLC (HL) コード CB06 クロック15

152. RLCAと同じ動作を(HL)について行います。フラグについては156. RLC rと同じです。

158. RLC (IX+d) コード DDCB××06 クロック23

152. RLCAと同じ動作を(IX+d)について行います。フラグについては156. RLC rと同じです。

159. RLC (IY+d) コード FDCB××06 クロック23

152. RLCAと同じ動作を(IY+d)について行います。フラグについては156. RLC rと同じです。

(表20)

r	
B	CB10
C	CB11
D	CB12
E	CB13
H	CB14
L	CB15
A	CB17

160. RL r コード (表20) クロック8

153. RLAと同じ動作をrについて行います。フラグは156. RLC rと同じです。

161. RL (HL) コード CB16 クロック15

153. RLAと同じ動作を(HL)について行います。フラグは156. RLC rと同じです。

162. RL (IX+d) コード DDCB××16 クロック23

153. RLAと同じ動作を(IX+d)について行います。フラグは156. RLC rと同じです。

163. RL (IY+d) コード FDCB××16 クロック23

153. RLAと同じ動作を(IY+d)について行います。フラグは156. RLC rと同じです。

(表21)

r	
B	CB08
C	CB09
D	CB0A
E	CB0B
H	CB0C
L	CB0D
A	CB0F

164. RRC r コード (表21) クロック8

154. RRCAと同じ動作をrについて行います。フラグは156. RLC rと同じです。

165. RRC (HL) コード CB0E クロック15

154. RRCAと同じ動作を(HL)について行います。フラグは156. RLC rと同じです。

166. RRC (IX+d) コード DDCB××0E クロック23

154. RRCAと同じ動作を(IX+d)について行います。フラグは156. RLC rと同じです。

167. RRC (IY+d) コード FDCB××0E クロック23

154. RRCAと同じ動作を(IY+d)について行います。フラグは156. RLC rと同じです。

(表22)

r	
B	CB18
C	CB19
D	CB1A
E	CB1B
H	CB1C
L	CB1D
A	CB1F

168. RR r コード (表22) クロック8

155. RRAと同じ動作をrについて行います。フラグは156. RLC rと同じです。

169. RR (HL) コード CB1E クロック15

155. RRAと同じ動作を(HL)について行います。フラグは156. RLC rと同じです。

170. RR (IX+d) コード DDCB××1E クロック23

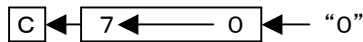
155. RRAと同じ動作を(IX+d)について行います。フラグは156. RLC rと同じです。

171. RR (IY+d) コード FDCB××1E クロック23

155. RRAと同じ動作を(IY+d)について行います。フラグは156. RLC rと同じです。

172. SLA r コード (表23) クロック8

rを左に1ビットシフトし、ビット7のデータをキャリーCに入れ、またビット0には0を入れます。フラグは156. RLC rと同じです。



(表23)

r	
B	CB20
C	CB21
D	CB22
E	CB23
H	CB24
L	CB25
A	CB27

173. SLA (HL) コード CB26 クロック15

172. SLA rと同じ動作を(HL)について行います。フラグは156. RIC rと同じです。

174. SLA (IX+d) コード DDCB××26 クロック23

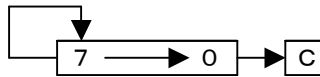
172. SLA rと同じ動作を(IX+d)について行います。フラグは156. RLC rと同じです。

175. SLA (IY+d) コード FDCB××26 クロック23

172. SLA rと同じ動作を(IY+d)について行います。フラグは156. RLC rと同じです。

176. SRA r コード (表24) クロック8

rの内容を1ビット右にシフトし、ビット0のデータをキャリーCに転送しますが、ビット7は変化しません。つまりビット7とビット6はつねに同じになります。フラグは156. RLC rと同じです。



(表24)

r	
B	CB28
C	CB29
D	CB2A
E	CB2B
H	CB2C
L	CB2D
A	CB2F

177. SRA (HL) コード CB2E クロック15

176. と同じ動作を(HL)について行います。フラグは156. RLC rと同じです。

178. SRA (IX+d) コード DDCB××2E クロック23

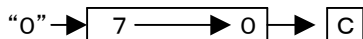
176. と同じ動作を(IX+d)について行います。フラグは156. RLC rと同じです。

179. SRA (IY+d) コード FDCB××2E クロック23

176. と同じ動作を(IY+d)について行います。フラグは156. RLC rと同じです。

180. SRL r コード (表25) クロック8

rの内容を1ビット右にシフトし、ビット0のデータをキャリーCに入れます。ビット7には0を入れます。フラグは156. RLCrと同じです。



(表25)

r	
B	CB38
C	CB39
D	CB3A
E	CB3B
H	CB3C
L	CB3D
A	CB3F

181. SRL (HL) コード CB3E クロック15

180. SRL rと同じ動作を(HL)について行います。フラグは156. RLC rと同じです。

182. SRL (IX+d) コード DDCB××3E クロック23

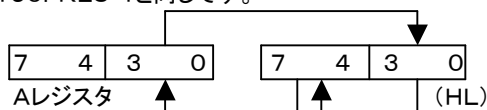
180. SRL rと同じ動作を(IX+d)について行います。フラグは156. RLC rと同じです。

183. SRL (IY+d) コード FDCB××3E クロック23

180. SRL rと同じ動作を(IY+d)について行います。フラグは156. RLC rと同じです。

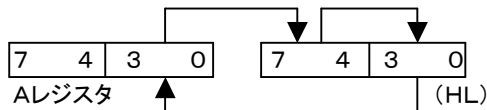
184. RLD コード ED6F クロック18

アキュムレータAとメモリ(HL)との間で下図のようなデータの交換が行われます。まず(HL)の下4ビットが上4ビットへ移され、(HL)の上4ビットはAの下4ビットへ移され、Aの下4ビットは(HL)の下4ビットへ移されます。フラグはキャリーCが変化しない点をのぞいて156. RLC rと同じです。



185. RRD コードED67 クロック18

(HL)の下位4ビットをAの下位4ビットに移し、Aの下位4ビットを(HL)の上位4ビットに移し、(HL)の上位4ビットを下位4ビットに移します。フラグは184. RLDと同じです。



IX. ビットセット・リセット、テスト命令

186. BIT b,r コード CBxx (xxは下図参照) クロック8

rレジスタのうちbで指定したビットを調べ、それが0ならZ=1、1ならZ=0にします。N=0、H=1になりますが、P/V、Sは不定です。

xx(8ビット)の内容

01 bbb rrr

bbb	ビット位置	rrr	レジスタ
000	0	000	B
001	1	001	C
010	2	010	D
011	3	011	E
100	4	100	H
101	5	101	L
110	6	111	A
111	7		

(例) Dレジスタのビット5を調べる命令コードは、bbb=101、rrr=010なので、CB6Aになります。(6A=01101010)

187. BIT b,(HL) コード CBxx (xxは下図参照) クロック12 (HL)のうちビットbの状態を調べます。動作、フラグについては186.BIT b,rと同じです。

xx(8ビット)の内容

01 bbb 110

bbbは186. の表参照

(例) ビット3のチェックならコードはCB5Eになります。

188. BIT b,(IX+d) コード DDCB××xx (xxは下図参照) クロック20

(IX+d)のうちビットbの状態を調べます。××は増分dです。動作、フラグについては186. .BIT b,rと同じです。

xx(8ビット)の内容

01 bbb 110

bbbは186. の表参照

189. BIT b,(IY+d) コード FDCB××xx (xxは下図参照) クロック20

(IY+d)のうちビットbの状態を調べます。××は増分dです。動作、フラグについては186. BIT b,rと同じです。

xx(8ビット)の内容

01 bbb 110

bbbは186. の表参照

190. SET b,r コード CBxx (xxは下図参照) クロック8

レジスタrのビットbをセットします。フラグは変化しません。

xx(8ビット)の内容

11 bbb rrr

bbb、rrrは186. の表参照

191. SET b,(HL) コード CBxx (xxは下図参照)クロック15

HLのビットbをセットします。

xx(8ビット)の内容

11 bbb 110

bbbは186. の表参照

192. SET b,(IX+d) コード DDCB××xx (xxは下図参照) クロック23

(IX+d)のビットbをセットします。××は増分dです。

xx(8ビット)の内容

11	bbb	110
----	-----	-----

 bbbは186. の表参照

193. SET b,(IY+d) コード FDCB × × xx (xxは下図参照) クロック23
(IY+d)のビットbをセットします。× ×は増分dです。

xx(8ビット)の内容

11	bbb	110
----	-----	-----

 bbbは186. の表参照

194. RES b,r コード CBxx (xxは下図参照) クロック8
レジスタrのビットbをリセットします。フラグは変化しません。

xx(8ビット)の内容

10	bbb	rrr
----	-----	-----

 bbb、rrrは186. の表参照

195. RES b,(HL) コード CBxx (xxは下図参照) クロック15
(HL)のビットbをリセットします。

xx(8ビット)の内容

10	bbb	110
----	-----	-----

 bbbは186. の表参照

196. RES b,(IX+d) コード DDCB × × xx (xxは下図参照) クロック23
(IX+d)のビットbをリセットします。× ×は増分dです。

xx(8ビット)の内容

10	bbb	110
----	-----	-----

 bbbは186. の表参照

197. RES b,(IY+d) コード FDCB × × xx (xxは下図参照) クロック23
(IY+d)のビットbをリセットします。× ×は増分dです。

xx(8ビット)の内容

10	bbb	110
----	-----	-----

 bbbは186. の表参照

X. ジャンプ命令 このグループの命令は、フラグに影響を与えません。

198. JP nn コード C3 × × × × クロック10
16ビットのデータnnで示されるアドレスにジャンプします。× × × ×については、8. LD A,(DE)参照。
ジャンプ先アドレスとしては命令コードの第一バイトが書かれているアドレスを指定します。オペランドを含めて2バイト以上の長さの命令の第一バイト以外の途中のアドレスを指定するとジャンプ命令の実行後に暴走してしまいます。(JMP nn)

199. JP NZ,nn コード C2 × × × × クロック10
Zフラグがセットされていなければnnへジャンプします。ジャンプ先アドレスについては198. JP nnを参照してください。(JNZ n n)

200. JP Z,nn コード CA × × × × クロック10
Zフラグがセットされていたらnnへジャンプします。ジャンプ先アドレスについては198. JP nnを参照してください。(JZ nn)

201. JP NC,nn コード D2 × × × × クロック10
Cフラグがセットされていなければnnへジャンプします。ジャンプ先アドレスについては198. JP nnを参照してください。(JNC n n)

202. JP C,nn コード DA × × × × クロック10
Cフラグがセットされていたらnnへジャンプします。ジャンプ先アドレスについては198. JP nnを参照してください。(JC nn)

203. JP PO,nn コード E2 × × × × クロック10
P/Vフラグがセットされていなければnnへジャンプします。ジャンプ先アドレスについては198. JP nnを参照してください。(JPO nn)

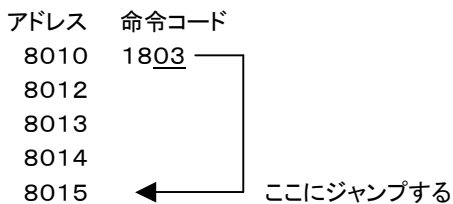
204. JP PE,nn コード EA × × × × クロック10

P/Vフラグがセットされていたらnnへジャンプします。ジャンプ先アドレスについては198. JP nnを参照してください。(JPE nn)

205. JP P,nn コード F2xxxx クロック10
Sフラグがセットされていなければnnへジャンプします。ジャンプ先アドレスについては198. JP nnを参照してください。(JP nn)

206. JP M,nn コード FAxxxx クロック10
Sフラグがセットされていたらnnへジャンプします。ジャンプ先アドレスについては198. JP nnを参照してください。(JM nn)

207. JR e コード 18xx クロック12
この命令の書かれている番地からe番地先へジャンプします。なお、命令のオペランド部(xx)にはe-2を書きます。下のように5バイト先にジャンプさせたい時オペランドは、5-2=3となります。
ジャンプ先アドレスとしては命令コードの第一バイトが書かれているアドレスを指定します。オペランドを含めて2バイト以上の長さの命令の第一バイト以外の途中のアドレスを指定するとジャンプ命令の実行後に暴走してしまいます。
+方向に進むときオペランドには00~7F(0~127)の範囲の数を使います。FE~80(-2~-128)を使うと-方向にジャンプします。FE(-2)は無限ループになります。FF(-1)は使えません(暴走します)。



208. JR C,e コード 38xx クロック7(C=0) または12(C=1)
Cフラグがセットされていればe番地先へジャンプします。e、xxについては207. JR e参照。

209. JP NC,e コード 30xx クロック7(C=0) または12(C=1)
Cフラグがセットされていなければe番地先へジャンプします。e、xxについては207. JR e参照。

210. JR Z,e コード 28xx クロック7(Z=0) または12(Z=1)
Zフラグがセットされていればe番地先へジャンプします。e、xxについては207. JR e参照。

211. JR NZ,e コード 20xx クロック7(Z=1) または12(Z=0)
Zフラグがセットされていなければe番地先へジャンプします。e、xxについては207. JR e参照。

212. JP (HL) コード E9 クロック4
ペアレジスタHLの内容をプログラムカウンタPCにロードします。つまりHLで示されるアドレスへジャンプします。(PCHL)

213. JP (IX) コード DDE9 クロック8
インデックスレジスタIXで示されるアドレスへジャンプします。

214. JP (IY) コード FDE9 クロック8
インデックスレジスタIYで示されるアドレスへジャンプします。

215. DJNZ e コード 10xx クロック 8(B=0)または13(B≠0)
少しかわった命令です。Bレジスタから1を引き、その結果が0でなければe番地先へジャンプします。(e、xxについては207. JR e参照)
例えば、ある処理を何回か繰り返し行うようなプログラムに利用します。Bレジスタに繰り返し回数を入れて使用します。

X I. コール、リターン命令 このグループの命令はフラグに影響を与えません。

216. CALL nn コード CDxxxx クロック17
nnで示されるアドレスからはじまるサブルーチンをコールします。なおこのとき、このCALL命令の次の命令のアドレスがスタックに退避されます。この結果SPは2減じられます。サブルーチンの最後にリターン命令を使うとスタックに退避されていたアドレスがPCに入れられて、その結果コール命令の次の命令のアドレスへ戻って処理が続けられます。(CALL nn)

217. CALL NZ,nn コード C4xxxx クロック 10(Z=1)または17(Z=0)

- Zフラグがセットされていなければnnからはじまるサブルーチンをコールします。(CNZ nn)
218. CALL Z,nn コード CC×××× クロック 10(Z=0)または17(Z=1)
Zフラグがセットされていたらnnからはじまるサブルーチンをコールします。(CZ nn)
219. CALL NC,nn コード D4×××× クロック 10(C=1)または17(C=0)
Cフラグがセットされていなければnnからはじまるサブルーチンをコールします。(CNC nn)
220. CALL C,nn コード DC×××× クロック 10(C=0)または17(C=1)
Cフラグがセットされていたらnnからはじまるサブルーチンをコールします。(CC nn)
221. CALL PO,nn コード E4×××× クロック 10(P/V=1)または17(P/V=0)
P/Vフラグがセットされていなければnnからはじまるサブルーチンをコールします。(CPO nn)
222. CALL PE,nn コード EC×××× クロック 10(P/V=0)または17(P/V=1)
P/Vフラグがセットされていたらnnからはじまるサブルーチンをコールします。(CPE nn)
223. CALL P,nn コード F4×××× クロック 10(S=1)または17(S=0)
Sフラグがセットされていなければnnからはじまるサブルーチンをコールします。(CP nn)
224. CALL M,nn コード FC×××× クロック 10(S=0)または17(S=1)
Sフラグがセットされていたらnnからはじまるサブルーチンをコールします。(CM nn)
225. RET コード C9 クロック10
スタックのトップに退避されていたアドレスデータをPCに入れ、そのアドレスにリターンします。この結果PCは+2されます。(RET)
226. RET NZ コード C0 クロック5(Z=1)または11(Z=0)
Zフラグがセットされていなければリターンします。(RNZ)
227. RET Z コード C8 クロック5(Z=0)または11(Z=1)
Zフラグがセットされていたらリターンします。(RZ)
228. RET NC コード D0 クロック5(C=1)または11(C=0)
Cフラグがセットされていなければリターンします。(RNC)
229. RET C コード D8 クロック5(C=0)または11(C=1)
Cフラグがセットされていたらリターンします。(RC)
230. RET PO コード E0 クロック5(P/V=1)または11(P/V=0)
P/Vフラグがセットされていなければリターンします。(RPO)
231. RET PE コード E8 クロック5(P/V=0)または11(P/V=1)
P/Vフラグがセットされていたらリターンします。(RPE)
232. RET P コード F0 クロック5(S=1)または11(S=0)
Sフラグがセットされていなければリターンします。(RP)
233. RET M コード F8 クロック5(S=0)または11(S=1)
Sフラグがセットされていたらリターンします。(RM)
234. RETI コード ED4D クロック14
割込みサービスルーチンの最後に使用します。割込み処理を終了してもとのプログラムに戻るとともに、割込み発生元(Z80ファミリ-ICまたは割込みコントローラ)に割込み処理の終了を知らせます。
235. RETN コード ED45 クロック14
ノンマスクابلインタラプト処理ルーチンの最後に使用します。
236. RST n(n=0~7) コード(下表) クロック11
1バイトで実行できるコール命令です。ただしコールできるアドレスは下表の8種に固定されています。(RST n)

ニーモニック	コード	アドレス
RST 0	C7	0000
RST 1	CF	0008
RST 2	D7	0010
RST 3	DF	0018
RST 4	E7	0020
RST 5	EF	0028
RST 6	F7	0030
RST 7	FF	0038

例えば、RST 3(コードDF)は、CALL \$0018(コードCD1800)と同じ動作になります。

X II. 入出力命令

237. IN A,(n) コード DB×× クロック11
nで指定されるアドレスのI/OデバイスからデータをアキュムレータAに入力します。フラグは変化しません。(IN n)

238. IN r,(C) コード (表26) クロック12
レジスタCの内容で指定されるアドレスのI/Oデバイスからrレジスタにデータを入力します。N=0、H=0 になりますがCフラグは変化しません。入力データが0のときZ=1、マイナスのときS=1になります。また入力データのビット1の数が偶数のときP/V=1になります

(表26)

r	
B	ED40
C	ED48
D	ED50
E	ED58
H	ED60
L	ED68
A	ED78

239. INI コード EDA2 クロック16
レジスタCで指定するI/O デバイスからデータを入力しHLレジスタで指定するメモリに転送します。この後にレジスタBの内容を-1しHLの内容を+1します。N=1。Cは変化しません。Bが0になればZ=1。他のフラグは不定です。

240. INIR コード EDB2 クロック16(B=0)または21(B≠0)
上記239. INIと同じ動作をしますがBが0になるまで繰り返し実行されます。Cは変化しません。N=1、Z=1、他は不定。

241. IND コード EDAA クロック16
239. INIと同じ動作ですが違うところは最後にHLレジスタが+1されるのではなく-1される点です。フラグは239. INIと同じです。

242. INDR コード EDDB クロック16(B=0)または21(B≠0)
240. INIRと同じ動作ですがHLレジスタは-1されて行きます。フラグは240. INIRと同じです。

(表27)

r	
B	ED41
C	ED49
D	ED51
E	ED59
H	ED61
L	ED69
A	ED79

243. OUT (n),A コード D3×× クロック11
nで指定されるアドレスのI/OデバイスにAレジスタのデータを出力します。フラグは変化しません。(OUT n)

244. OUT (C),r コード (表27) クロック12
レジスタCで指定するI/O デバイスにレジスタrの内容が出力されます。フラグは変化しません。

245. OUTI コード EDA3 クロック16
レジスタCで指定するI/O デバイスにHLレジスタで示されるメモリの内容が出力されます。このあとBレジスタの内容が-1され、HLの内容は+1されます。フラグは239. INIと同じです。

246. OTIR コード EDB3 クロック16(B=0)または21(B≠0)
上記245. OUTIと同じ動作をしますがBが0になるまで繰り返し実行されます。フラグは240. INIRと同じです。

247. OUTD コード EDAB クロック16
上記245. OUTIと同じ動作をしますがHLは+1されるのではなく-1されます。フラグは239. INIと同じです。

248. OTDR コード EDDB クロック16(B=0)または21(B≠0)
上記246. OTIRと同じ動作をしますがHLは-1されて行きます。フラグは240. INIRと同じです。

X III. 割込制御

249. DI コード F3 クロック4

割込を禁止します。DI命令の実行後はINT信号が入力されても新たな割込は発生しません。
リセット後は割込禁止状態になります。(DI)

250. EI コード FB クロック4

EI命令が実行されると、その次の命令の実行後から割込が受け付け可能状態になります。

EI命令の実行後ではなくて、その次の命令の実行後から、割込みの受け付けが可能になるようにしている理由は、割込み処理プログラムの最後は通常RET命令で終わっているために、もしEI命令の実行後に割込みの受け付けが可能になっていると、割込み処理が終わってメインプログラムに戻るためのRET命令が実行される前に次の割込みが受け付けられてしまい、二重に割込みが実行されてしまう可能性が出てくることを避けるためです。(EI)

251. IM 0 コード ED46 クロック8

Z80のマスク可能な割込みはモード0、モード1、モード2の3種類あります。IM 0の実行によってモード0の割込みが選択されます。モード0の割込みは8080の割込みと同じ動作になります。INT信号の入力とともに、データバスにRST 0～RST 7の命令コードを入力することによって、0000から8番地とびに置かれた割込み処理プログラムが実行されます。RST命令については236. RST n を参照してください。

リセット後はモード0が選択されます。

252. IM 1 コード ED56 クロック8

マスク可能な割込みをモード1に設定します。モード0はINT信号とともにデータバスにRST 0～RST 7の命令コードを入力することで8通りの割込み処理ルーチンを選択できますが、モード1はそのうちのRST 7割込み動作に特化した割込みモードです。

モード1はINT信号の入力によってつねに0038番地の割込み処理プログラムが実行されますが、そのときデータバスに命令コードを入力する必要はなく、INT信号のみで割込み処理が行われます。

253. IM 2 コード ED5E クロック8

モード2は割込みベクタレジスタ(レジスタ)を使う高度な割込み処理モードです。

あらかじめ割込み処理プログラムアドレスを配置したテーブルをメモリ内に用意しておき、そのテーブルアドレスの上位アドレスをIレジスタに入れておきます。INT割込みとともにデータバスに、テーブルアドレスの下位アドレスを入力することによって、メモリの広い範囲に数を制限されることなく必要な割込み処理プログラムを置くことができます。