

ND80ZⅢ 8080命令説明書

(有)中日電工

☆☆☆ 目次 ☆☆☆

[はじめに].....	1	58. CMA	8
[1] レジスタについて.....	1	59. CMC	8
[2] 8ビット転送命令.....	1	60. STC	8
1. MOV r1, r2	1	61. NOP	8
2. MOV r, M	2	62. HLT	8
3. MOV M, r	2	63. DI	8
4. MVI r, B2	2	64. EI	8
5. MVI M, B2.....	2	[7] 16ビット算術演算命令	8
6. LDAX B	2	65. DAD B	8
7. LDAX D	2	66. DAD D	8
8. LDA NN	2	67. DAD H	8
9. STAX B	2	68. DAD SP	8
10. STAX D	2	69. INX B	8
11. STA nn	2	70. INX D	9
[3] 16ビット転送命令	3	71. INX H	9
12. LXI B, nn	3	72. INX SP	9
13. LXI D, nn	3	73. DCX B	9
14. LXI H, nn	3	74. DCX D	9
15. LXI SP, nn	3	75. DCX H	9
16. LHLD nn	3	76. DCX SP	9
17. SHLD nn	3	[8] 回転、シフト命令	9
18. SPHL	3	77. RLC	9
19. PUSH B	3	78. RAL	9
20. PUSH D	3	79. RRC	9
21. PUSH H	3	80. RAR	9
22. PUSH PSW	3	[9] ジャンプ命令	9
23. POP B	3	81. JMP nn	9
24. POP D	3	82. JNZ nn	10
25. POP H	3	83. JZ nn	10
26. POP PSW	3	84. JNC nn	10
[4] 交換命令	3	85. JC nn	10
27. XCHG	3	86. JPO nn	10
28. XTHL	4	87. JPE nn	10
[8080のフラグ]	4	88. JP nn	10
[5] 8ビット算術・論理演算命令	4	89. JM nn	10
29. ADD r	4	90. PCHL	10
30. ADI n	5	[10] コール、リターン命令	10
31. ADD M	5	91. CALL nn	10
32. ADC r	5	92. CNZ nn	10
33. ACI n	5	93. CZ nn	10
34. ADC M	5	94. CNC nn	10
35. SUB r	5	95. CC nn	11
36. SUI n	5	96. CPO nn	11
37. SUB M	5	97. CPE nn	11
38. SBB r	5	98. CP nn	11
39. SBI n	6	99. CM nn	11
40. SBB M	6	100. RET	11
41. ANA r	6	101. RNZ	11
42. ANI n	6	102. RZ	11
43. ANA M	6	103. RNC	11
44. ORA r	6	104. RC	11
45. ORI n	6	105. RPO	11
46. ORA M	6	106. RPE	11
47. XRA r	6	107. RP	11
48. XRI n	7	108. RET M	11
49. XRA M	7	109. RST n(n=0~7)	11
50. CMP r	7	[11] 入出力命令	12
51. CPI n	7	110. IN n	12
52. CMP M	7	111. OUT n	12
53. INR r	7		
54. INR M	7		
55. DCR r	7		
56. DCR M	7		
[6] その他の算術演算、CPU制御命令	7		
57. DAA	8		

[はじめに]

ND80ZⅢのCPUはZ80です。

Z80は8080の全ての命令を8080と同じように実行することができますが、さらに8080にはなかった多くの命令が追加されています。

Z80は8080と全く同じレジスタをもっていますが、さらに8080にはなかったレジスタが追加されています。

Z80の命令とレジスタの詳細については、「Z80命令説明書」で詳しく説明をしています。

ND80ZⅢにはZ80のプログラムを作るためのツールとして当社が独自に開発したND80Zモニタプログラムが搭載されています。と同時に、かつてのマイコンブームの火付け役となった、NECのTK80のモニタプログラムも搭載していて、TK80と同じ動作をさせることもできるようになっています。

TK80に搭載されていたCPUはZ80ではなくて8080でした。当然のことながらTK80のために書かれた多くのプログラムは8080の命令コード(および8080ニーモニック)で書かれています。

Z80は8080の全ての命令コードをそのまま実行できますから、8080のプログラム、命令を参照するには、「Z80命令説明書」を利用することができます。しかし「Z80命令説明書」には8080にはない多くの命令の説明についても書かれていますから、その中から8080のみの命令を見つけるのは面倒です。

またはじめてZ80のプログラムを書こうとする方にとっては、Z80の非常に多い命令にいきなり接するというのはいかにもハードルが高いことでもあります。

その意味では、「8080には、Z80に比べて少ない命令とレジスタしかない」というのは、劣っているというよりもむしろ都合がよい長所である、とも言えます。

そのように考えた「8080の利点」を十分に活用できるように、本書はあえて、CPUが8080である、という想定で作成してあります。

命令は8080の命令に限定して、そのニーモニックもZ80ニーモニック(ザイログニーモニック)ではなくて、8080ニーモニック(インテルニーモニック)で表現してあります。

レジスタについても8080が持っているレジスタのみについて説明してあります。

Z80は一部の命令で、実行結果のフラグの状態が8080とは異なっているものがあります。また命令の実行クロック数も異なっているものがあります。

それらについては、Z80での動作を優先して表記し、あわせて8080との相違を記述することにつとめました。

[1] レジスタについて

8080には8ビットレジスタB、C、D、E、H、L、A、Fと16ビットレジスタSP、PCがあります。

ADD、SUBなどの加減算命令やORA、ANAなどの論理演算命令は、2つのレジスタ間で演算を行います。演算の結果を格納するレジスタはAレジスタに限られます。

F(フラグ)レジスタはS、Z、H、P、Cの各フラグをビット情報として格納しているレジスタですが、PUSH PSW、POP PSW命令でAレジスタとともにスタックへの格納、取出しが出来るほかは演算などの命令の対象にすることはできません。

RALなどのローテイト命令や、I/O制御命令IN、OUTでは、Aレジスタのみが使われます。

8ビットのレジスタBとC、DとE、HとLはそれぞれ結合して16ビットのレジスタBC、DE、HLとして使うこともできます。

16ビットのレジスタHLはメモリアドレスを示す間接アドレッシングレジスタとしても使われます。

8080ニーモニックでメモリを示すMは、HLレジスタで示すメモリアドレスの意味で使われます。

BC、DEレジスタも限られた命令(STAX、LDAX)においてのみHLと同じように間接アドレッシングでメモリアドレスを示すために使われます。

[2] 8ビット転送命令(この命令群はフラグに影響を与えません)

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マシクロック数のことです。

ND80ZⅢのCPUクロックは4MHzですから1クロックは $1/4 \mu\text{S}$ ($0.25 \mu\text{S}$)になります。

したがってたとえばクロック=4の命令の実行時間は $0.25 \times 4 = 1.0 \mu\text{S}$ になります。

Z80は8080の全ての命令を、同じように実行しますが、一部の命令では、その命令を実行するのに必要なクロック数が8080とは異なっています。そのような命令では、8080のクロック数を、うしろに()で示しました。

また参考までに説明のうしろにZ80ニーモニックも表記してあります。

1. MOV r1, r2 コード(表1) クロック4(5)

r1、r2はB、C、D、E、H、L、Aを表します。r2の内容をr1に転送します。r2の内容は変化しません。(LD r1, r2)

(表1)

		r2						
		B	C	D	E	H	L	A
r1	B	40	41	42	43	44	45	47
	C	48	49	4A	4B	4C	4D	4F
	D	50	51	52	53	54	55	57
	E	58	59	5A	5B	5C	5D	5F
	H	60	61	62	63	64	65	67
	L	68	69	6A	6B	6C	6D	6F
	A	78	79	7A	7B	7C	7D	7F

(表2)

r	
B	46
C	4E
D	56
E	5E
H	66
L	6E
A	7E

(表3)

r	
B	70
C	71
D	72
E	73
H	74
L	75
A	77

2. MOV r, M コード (表2) クロック7

MはHLレジスタで示されるメモリを表します。Mの内容をrに転送します。Mの内容は変化しません。(LD r, (HL))

3. MOV M, r コード (表3) クロック7

rの内容をMに転送します。rの内容は変化しません。(LD (HL), r)

4. MVI 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。(LD r, n)

(表4)

r	
B	06XX
C	0EXX
D	16XX
E	1EXX
H	26XX
L	2EXX
A	3EXX

5. MVI M, n コード 36×× クロック10

nをMに転送します(××は任意の8ビットデータを表します)。(LD (HL), n)

6. LDAX B コード 0A クロック7

BCレジスタで示されるメモリの内容をAに転送します。メモリの内容は変化しません。(LD A, (BC))

7. LDAX D コード 1A クロック7

DEレジスタで示されるメモリの内容をAに転送します。メモリの内容は変化しません。(LD A, (DE))

8. LDA nn コード 3A×××× クロック13

命令コード3Aに続く2バイトデータnnで示されるメモリの内容をAに転送します。メモリの内容は変化しません。

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。(LD A, (nn))

9. STAX B コード 02 クロック7

Aの内容をBCレジスタで示されるメモリに転送します。Aの内容は変化しません。(LD (BC), A)

10. STAX D コード 12 クロック7

Aの内容をDEレジスタで示されるメモリに転送します。Aの内容は変化しません。(LD (DE), A)

11. STA nn コード 32×××× クロック13

Aの内容をnnで示されるメモリに転送します。Aの内容は変化しません。(LD (nn), A)

nnについては 8. LDA nn の説明を参照してください。

[3] 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)

12. LXI B, nn コード 01 x x x x 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。(LD BC, nn)
nnについては 8. LDA nn の説明を参照してください。
13. LXI D, nn コード 11 x x x x クロック10
16ビットのデータnnをペアレジスタDEに転送します。(LD DE, nn)
nnについては 8. LDA nn の説明を参照してください。
14. LXI H, nn コード 21 x x x x クロック10
16ビットのデータnnをペアレジスタHLに転送します。(LD HL, nn)
nnについては 8. LDA nn の説明を参照してください。
15. LXI SP, nn コード 31 x x x x クロック10
16ビットのデータnnをSP(スタックポインタ)に転送します。(LD SP, nn)
nnについては 8. LDA nn の説明を参照してください。
16. LHLD nn コード 2A x x x x クロック16
nnで示されるメモリの内容をLに、nn+1番地のメモリの内容をHに転送します。(LD HL, (nn))
nnについては 8. LDA nn の説明を参照してください。
17. SHLD nn コード 22 x x x x クロック16
Lの内容をnnで示されるメモリへ、Hの内容をnn+1番地のメモリへ転送します。(LD (nn), HL)
nnについては 8. LDA nn の説明を参照してください。
18. SPHL コード F9 クロック6(5)
HLの内容をSPに転送します。(LD SP, HL)
19. PUSH B 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 BC)
20. PUSH D 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 DE)
21. PUSH H 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 HL)
22. PUSH PSW 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 AF)
23. POP B 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 BC)
24. POP D 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 DE)
25. POP H 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 HL)
26. POP PSW 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 AF)

[4] 交換命令(この命令群もフラグに影響を与えません)

27. XCHG 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。(EX DE, HL)

28. XTHL コード E3 クロック19(18)

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。(EX (SP), HL)

以上の命令はフラグに影響を与えませんが、以下の命令の多くはフラグに影響を与えます。

[8080のフラグ]

- C キャリーフラグ 計算の結果、上位桁へのキャリー(ポロー)が発生したときにセットされる。ローテイト命令でもセット、リセットされる。
- Z ゼロフラグ 計算結果がゼロのときセット。
- P パリティフラグ 1バイト内の1になっているビット数が偶数でセット、奇数でリセットされる。
- S サインフラグ 計算結果がマイナスのときセット、(8ビット目が1の数)がマイナスになる。FF~80がマイナス、00~7Fがプラス)
- H ハーフキャリーフラグ (CY4)
 算術演算のビット3とビット4との間のキャリー、ポローの有無によってSET、RESETされる。

◎ フラグレジスタFにおける各フラグの位置(8080)

ビット位置	7	6	5	4	3	2	1	0
フラグ	S	Z	x	H	x	P	x	C

xは使用されない

[Z80のフラグ]

- C キャリーフラグ 計算結果のオーバーフローでセットされ、ローテイト命令でもセット、リセットされる。
- Z ゼロフラグ 計算結果がゼロのときセット。
- P/V パリティ/オーバーフローフラグ(8080:P)
 1バイト内の1になっているビット数が偶数でセット、奇数でリセット、また算術演算の結果オーバーフローでセット。
- S サインフラグ 計算結果がマイナスのときセット、(8ビット目が1の数)がマイナスになる。FF~80がマイナス、00~7Fがプラス)
- N 加減算フラグ ADD命令のとき0、SUB命令のとき1(DAA命令に使用される)
- H ハーフキャリーフラグ (8080:CY4)
 算術演算のビット3とビット4との間のキャリー、ポローの有無によってSET、RESETされる。

◎ フラグレジスタFにおける各フラグの位置(Z80)

	7	6	5	4	3	2	1	0
	S	Z	x	H	x	P/V	N	C

xは使用されない

なお、以下の説明では影響を受けるフラグのみ表記します。書いてないフラグは影響を受けないことを意味します。

[5] 8ビット算術・論理演算命令

このグループの命令はINC, DCR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。

29. ADD r コード (表9) クロック4

(フラグについて)

- 計算の結果、上位桁へのキャリーが発生したときC=1、発生しないときはC=0になります。
- 計算の結果、ビット3からビット4へのキャリーが発生したときH=1、発生しないときはH=0になります。
- 結果がゼロのときZ=1、ゼロではないときはZ=0になります。
- 結果が負数のとき(80~FFのとき)S=1、正数のときはS=0になります。(ADD A, r)
- 8080では計算の結果、1のビットが偶数個あればP=1、奇数個ならP=0になります。
- Z80では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。

30. ADI 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29. ADD A, r と同じです。(ADD A, n)

(表9)

r	
B	80
C	81
D	82
E	83
H	84
L	85
A	87

(表10)

r	
B	88
C	89
D	8A
E	8B
H	8C
L	8D
A	8F

31. ADD M コード 86 クロック7

AとHLで示されるメモリの内容とを加算し、結果をAに入れます。フラグについては29. ADD A, r と同じです。(ADD A, (HL))

32. ADC 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29. ADD A, r と同じです。(ADC A, r)

33. ACI 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29. ADD A, r と同じです。(ADC A, n)

34. ADC M コード 8E クロック7

AとHLで示されるメモリの内容とキャリーCとを加算し、Aに入れます。フラグについては29. ADD A, r と同じです。(ADC A, (HL))

35. 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。(SUB r)

(フラグについて)

計算の結果、上位桁へのボローが発生したときC=1、発生しないときはC=0になります。

計算の結果、ビット3からビット4へのボローが発生したときH=1、発生しないときはH=0になります。

結果がゼロのときZ=1、ゼロではないときはZ=0になります。

結果が負数のとき(80~FFのとき)S=1、正数のときはS=0になります。

8080では計算の結果、1のビットが偶数個あればP=1、奇数個ならP=0になります。

Z80ではN=1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。

(表11)

r	
B	90
C	91
D	92
E	93
H	94
L	95
A	97

(表12)

r	
B	98
C	99
D	9A
E	9B
H	9C
L	9D
A	9F

36. SUI 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35. SUB rと同じです。(SUB n)

37. SUB M コード 96 クロック7

AからHLで示されるメモリの内容を引いて結果をAに入れます。フラグは35. SUB rと同じです。(SUB (HL))

38. SBB 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35. SUB rと同じです。(SBC A, r)

39. SBI 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35. SUB rと同じです。(SBC A, n)

40. SBB M コード 9E クロック7

AからHLで示されるメモリの内容を引き、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35. SUB rと同じです。(SBC A, (HL))

41. ANA 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。(AND r)
(フラグについて)

結果がゼロのときZ=1、ゼロではないときはZ=0になります。

結果が負数のとき(80~FFのとき)S=1、正数のときはS=0になります。

結果のうち1のビットが偶数個あればP=1、奇数個ならP=0になります。

C(キャリー)フラグがクリアされることに注意してください。

8080ではH=0になりますが、Z80ではN=0、H=1になります。

(表13)

r	
B	A0
C	A1
D	A2
E	A3
H	A4
L	A5
A	A7

(表14)

r	
B	B0
C	B1
D	B2
E	B3
H	B4
L	B5
A	B7

42. ANI 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41. ANA rと同じです。(AND n)

43. ANA M コード A6 クロック7

AとHLで示されるメモリの内容とのANDをとって結果をAに入れます。フラグは41. ANA rと同じです。(AND (HL))

44. ORA 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Z80でH=0になることを除いて41. ANA rと同じです。(OR r)

45. ORI 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、44. ORA rと同じです。(OR n)

46. ORA M コード B6 クロック7

AとHLで示されるメモリの内容とのORをとって結果をAに入れます。フラグは、44. ORA rと同じです。(OR (HL))

47. XRA 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44. ORA rと同じです。(XOR r)

(表15)

r	
B	A8
C	A9
D	AA
E	AB
H	AC
L	AD
A	AF

(表16)

r	
B	B8
C	B9
D	BA
E	BB
H	BC
L	BD
A	BF

48. XRI 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44. ORA rと同じです。(XOR n)
49. XRA M コード AE クロック7
AとHLで示されるメモリの内容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44. ORA rと同じです。(XOR (HL))
50. CMP 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。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3からビット4へのポローがあればH=1になります。(CP r)
8080では計算の結果、1のビットが偶数個あればP=1、奇数個ならP=0になります。
Z80ではN=1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。

51. CPI n コード FE×× クロック7
Aとnとを比較します。フラグは50. CP rと同じです。(CP n)

52. CMP M コード BE クロック7
AとHLで示されるメモリの内容とを比較します。フラグは50. CP rと同じです。(CP (HL))

53. INR r コード (表17) クロック4(5)
レジスタrの内容を+1します。以下のINR, DCR命令ではキャリーCのみ変化しません。注意して下さい。
ビット3からビット4へのキャリーが発生したときH=1、発生しないときはH=0になります。
結果がゼロのときZ=1、ゼロではないときはZ=0になります。
結果が負数のとき(80~FFのとき)S=1、正数のときはS=0になります。(INC r)
8080では計算の結果、1のビットが偶数個あればP=1、奇数個ならP=0になります。
Z80ではN=0になります。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。その他のときはP/V=0になります。

(表17)

r	
B	04
C	0C
D	14
E	1C
H	24
L	2C
A	3C

(表18)

r	
B	05
C	0D
D	15
E	1D
H	25
L	2D
A	3D

54. INR M コード 34 クロック11(10)
HLで示されるメモリの内容を+1します。フラグは53. INR rと同じです。(INC (HL))

55. DCR r コード (表18) クロック4(5)
rの内容を-1します。
DCR命令ではキャリーCは変化しません。注意して下さい。
ビット3からビット4へのポローが発生したときH=1、発生しないときはH=0になります。
結果がゼロのときZ=1、ゼロではないときはZ=0になります。
結果が負数のとき(80~FFのとき)S=1、正数のときはS=0になります。(DEC r)
8080では計算の結果、1のビットが偶数個あればP=1、奇数個ならP=0になります。
Z80ではN=1になります。この命令の実行後、結果が7FH(つまり実行前の値が80H)ならばP/V=1になります。その他のときはP/V=0になります。

56. DCR M コード 35 クロック11(10)
HLで示されるメモリの内容を-1します。フラグは55. DCR rと同じです。(DEC (HL))

[6] その他の算術演算、CPU制御命令

57. DAA コード 27 クロック4

加算後のA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。Z80は減算後のDAAにも対応していますが、8080は加算のみで減算には対応していません。加算にはADD、ADCのほかINC命令も含まれます。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=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)

58. CMA 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)。

8080ではフラグは変化しませんがZ80ではN=1、H=1になります。(CPL)

59. CMC コード 3F クロック4

キャリーフラグCを反転します。

8080では他のフラグは変化しませんが、Z80ではN=0になり、Hは実行前のCの内容と同じになります。(CCF)

60. STC コード 37 クロック4

キャリーフラグCをセットします。

8080では他のフラグは変化しませんが、Z80ではN=0、H=0になります。(SCF)

61. 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)

62. HLT コード 76 クロック4(7)

割込かりセットがかかるまでCPUの実行を停止します。(HALT)

63. 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)

64. 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、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)

[7] 16ビット算術演算命令

65. DAD B コード 09 クロック11(10)

HLの内容とBCの内容を加算し、結果をHLに入れます。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。(ADD HL, BC)

Z80ではN=0になります。

66. DAD D コード 19 クロック11(10)

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. DAD Bと同じです。(ADD HL, DE)

67. DAD H コード 29 クロック11(10)

HLの内容を2倍にします。フラグは120. DAD Bと同じです。(ADD HL, HL)

68. DAD SP コード 39 クロック11(10)

HLとSPとを加算し、結果をHLに入れます。フラグは120. DAD Bと同じです。(ADD HL, SP)

69. INX B コード 03 クロック6(5)

BCの内容を+1します。フラグは変化しません。(INC BC)

70. INX D コード 13 クロック6(5)
DEの内容を+1します。フラグは変化しません。(INC DE)

71. INX H コード 23 クロック6(5)
HLの内容を+1します。フラグは変化しません。(INC HL)

72. INX SP コード 33 クロック6(5)
SPの内容を+1します。フラグは変化しません。(INC SP)

73. DCX B コード 0B クロック6(5)
BCの内容を-1します。フラグは変化しません。(DEC BC)

74. DCX D コード 1B クロック6(5)
DEの内容を-1します。フラグは変化しません。(DEC BC)

75. DCX H コード 2B クロック6(5)
HLの内容を-1します。フラグは変化しません。(DEC HL)

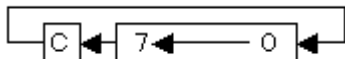
76. DCX SP コード 3B クロック6(5)
SPの内容を-1します。フラグは変化しません。(DEC SP)

[8] 回転、シフト命令

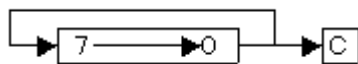
77. RLC コード 07 クロック4
Aレジスタの内容を1ビット左にシフトし、ビット7のデータをビット0に入れると共にキャリーCにも入れます。
8080ではC以外のフラグは変化しませんがZ80ではN=0、H=0になります。(RLCA)



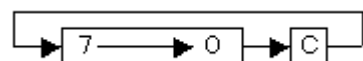
78. RAL コード 17 クロック4
キャリーを含め9ビットのデータを左に1ビット回転させます。8080ではC以外のフラグは変化しませんがZ80ではN=0、H=0になります。(RLA)



79. RRC 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。8080ではC以外のフラグは変化しませんがZ80ではN=0、H=0になります。(RRCA)



80. RAR 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。8080ではC以外のフラグは変化しませんがZ80ではN=0、H=0になります。(RRA)



[9] 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。

81. JMP nn コード C3 x x x x 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。
nnについては 8. LDA nn の説明を参照してください。(JP nn)

82. JNZ 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で示されるアドレスにジャンプします。
nnについては 8. LDA nn の説明を参照してください。(JP NZ, nn)
83. JZ 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で示されるアドレスにジャンプします。
nnについては 8. LDA nn の説明を参照してください。(JP Z, nn)
84. JNC 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で示されるアドレスにジャンプします。
nnについては 8. LDA nn の説明を参照してください。(JP NC, nn)
85. JC 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で示されるアドレスにジャンプします。
nnについては 8. LDA nn の説明を参照してください。(JP nn)
86. JPO nn コード E2×××× クロック10
Pフラグがセットされていなければnnで示されるアドレスにジャンプします。
nnについては 8. LDA nn の説明を参照してください。(JP PO, nn)
87. JPE nn コード EA×××× クロック10
Pフラグがセットされていたらnnで示されるアドレスにジャンプします。
nnについては 8. LDA nn の説明を参照してください。(JP PE, nn)
88. JP nn コード F2×××× クロック10
Sフラグがセットされていなければnnで示されるアドレスにジャンプします。
nnについては 8. LDA nn の説明を参照してください。(JP P, nn)
89. JM nn コード FA×××× クロック10
Sフラグがセットされていたらnnで示されるアドレスにジャンプします。
nnについては 8. LDA nn の説明を参照してください。(JP M, nn)
90. PCHL コード E9 クロック4(5)
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。(JP (HL))
- [10] 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。
91. CALL nn コード CD×××× 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。
nnについては 8. LDA nn の説明を参照してください。
なおこのとき、このCALL命令の次の命令のアドレスをスタックに退避します。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をPCに入れコール命令の次の命令のアドレスへ戻ります。(CALL nn)
92. CNZ nn コード C4×××× クロック17(Z=0)または10(Z=1)
Zフラグがセットされていなければ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。
nnについては 8. LDA nn の説明を参照してください。(CALL NZ, nn)
93. CZ nn コード CC×××× クロック17(Z=1)または10(Z=0)
Zフラグがセットされていたら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。
nnについては 8. LDA nn の説明を参照してください。(CALL Z, nn)
94. CNC nn コード D4×××× クロック17(C=0)または10(C=1)
Cフラグがセットされていなければ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。
nnについては 8. LDA nn の説明を参照してください。(CALL NC, nn)

95. CC nn コード DC×××× クロック17(C=1)または10(C=0)
Cフラグがセットされていたら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。
nnについては 8. LDA nn の説明を参照してください。(CALL C, nn)
96. CPO nn コード E4×××× クロック17(P=0)または10(P=1)
Pフラグがセットされていなければ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。
nnについては 8. LDA nn の説明を参照してください。(CALL PO, nn)
97. CPE nn コード EC×××× クロック17(P=1)または10(P=0)
Pフラグがセットされていたら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。
nnについては 8. LDA nn の説明を参照してください。(CALLPE, nn)
98. CP nn コード F4×××× クロック17(S=0)または10(S=1)
Sフラグがセットされていなければ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。
nnについては 8. LDA nn の説明を参照してください。(CALL P, nn)
99. CM nn コード FC×××× クロック17(S=0)または10(S=1)
Sフラグがセットされていたら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。
nnについては 8. LDA nn の説明を参照してください。(CALL M, nn)
100. 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SPは+2されます。(RET)
101. RNZ コード C0 クロック11(Z=0)または5(Z=1)
Zフラグがセットされていなければリターンします。(RET NZ)
102. RZ コード C8 クロック11(Z=1)または5(Z=0)
Zフラグがセットされていたらリターンします。(RET Z)
103. RNC コード D0 クロック11(C=0)または5(C=1)
Cフラグがセットされていなければリターンします。(RET NC)
104. RC コード D8 クロック11(C=1)または5(C=0)
Cフラグがセットされていたらリターンします。(RET C)
105. RPO コード E0 クロック11(P=0)または5(P=1)
Pフラグがセットされていなければリターンします。(RET PO)
106. RPE コード E8 クロック11(P=1)または5(P=0)
Pフラグがセットされていたらリターンします。(RET PE)
107. RP コード F0 クロック11(S=0)または5(S=1)
Sフラグがセットされていなければリターンします。(RET P)
108. RM コード F8 クロック11(S=1)または5(S=0)
Sフラグがセットされていたらリターンします。(RET M)
109. 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。
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。(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

[11] 入出力命令

110. IN n コード DB×× クロック11(10)

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 A, (n))

111. OUT n コード D3×× クロック11(10)

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), A)