

Полное описание системы команд микропроцессора Z80

**Алексей Асемов (Alex/AT)
Advanced Technologies, 2000-2005**

**При активном содействии:
Игоря Складорова (Scorpion ZS 256)**

***Ревизия 0.6
(18.06.2005)***

Оглавление

ПРЕДИСЛОВИЕ.....	7
ОСНОВЫ АРХИТЕКТУРЫ МИКРОПРОЦЕССОРА Z80.....	8
1. Двоичная система счисления.....	8
2. Шестнадцатеричная система счисления.....	8
3. Память микропроцессора Z80, порты, адресное пространство.....	9
4. Регистры микропроцессора Z80. Регистровые пары.....	9
5. Регистр флагов микропроцессора Z80. Флаги.....	11
СХЕМА ФОРМИРОВАНИЯ КОДОВ ОПЕРАЦИЙ МИКРОПРОЦЕССОРА Z80.....	13
1. Основные обозначения операций, условий и флагов.....	13
2. Операции без префикса.....	15
3. Операции с префиксом #DD.....	18
4. Операции с префиксом #FD.....	20
5. Операции с префиксом #ED.....	22
6. Операции с префиксом #CB.....	24
7. Операции с расширенным префиксом #DD #CB.....	25
8. Операции с расширенным префиксом #FD #CB.....	26
ПОДРОБНОЕ ОПИСАНИЕ КОМАНД МИКРОПРОЦЕССОРА Z80.....	27
1. Команды загрузки 8-битного регистра непосредственным 8-битным значением.....	27
2. Команды загрузки 16-битного регистра непосредственным 16-битным значением.....	27
3. Команды загрузки 8-битного регистра значением 8-битного регистра.....	28
4. Команды загрузки, использующие служебный 8-битный регистр в качестве одного из операндов.....	30
5. Команды загрузки 16-битного регистра значением 16-битного регистра.....	30
6. Команды загрузки 8-битного регистра значением в памяти по абсолютному адресу.....	30
7. Команды загрузки 8-битного регистра значением в памяти по адресу в 16-битной регистровой паре.....	31
8. Команды загрузки 8-битного регистра значением в памяти по адресу в индексном регистре (со смещением).....	31
9. Команды помещения значения регистра в память по абсолютному адресу.....	31
10. Команды помещения значения 8-битного регистра в память по адресу в 16-битной регистровой паре.....	32
11. Команда помещения непосредственного 8-битного значения в память по адресу в регистре HL.....	32
12. Команды помещения значения 8-битного регистра в память по адресу в индексном регистре (со смещением).....	32
13. Команды помещения непосредственного 8-битного значения в память по адресу в индексном регистре (со смещением).....	33
14. Команды обмена значений 16-битных регистровых пар.....	33
15. Команды обмена значений 16-битных регистровых пар и памяти.....	33
16. Команды сложения значения аккумулятора со значением 8-битного регистра.....	33
17. Команда сложения значения аккумулятора с непосредственным 8-битным значением.....	34
18. Команда сложения значения аккумулятора с 8-битным значением в памяти по адресу в регистре HL.....	34
19. Команды сложения значения аккумулятора с 8-битным значением в памяти по адресу в индексном регистре (со смещением).....	34
20. Команды сложения значений 16-битных регистровых пар.....	34
21. Команды инкремента значения 8-битного регистра.....	35
22. Команды инкремента 8-битного значения в памяти по адресу в регистре HL.....	35
23. Команды инкремента 8-битного значения в памяти по адресу в индексном регистре (со смещением).....	35
24. Команды инкремента значения 16-битного регистра.....	35

25. Команды сложения значения аккумулятора со значением 8-битного регистра с учетом флага переноса	36
26. Команда сложения значения аккумулятора с непосредственным 8-битным значением с учетом флага переноса.....	36
27. Команда сложения значения аккумулятора с 8-битным значением в памяти по адресу в регистре HL с учетом флага переноса	36
28. Команды сложения значения аккумулятора с 8-битным значением в памяти по адресу в индексном регистре (со смещением) с учетом флага переноса.....	36
29. Команды сложения значений 16-битных регистровых пар с учетом флага переноса	37
30. Команды вычитания значения 8-битного регистра из значения аккумулятора	37
31. Команда вычитания непосредственного 8-битного значения из значения аккумулятора.....	37
32. Команда вычитания 8-битного значения в памяти по адресу в регистре HL из значения аккумулятора.....	37
33. Команды вычитания 8-битного значения в памяти по адресу в индексном регистре (со смещением) из значения аккумулятора.....	37
34. Команды декремента значения 8-битного регистра	38
35. Команды декремента 8-битного значения в памяти по адресу в регистре HL	38
36. Команды декремента 8-битного значения в памяти по адресу в индексном регистре (со смещением)	38
37. Команды декремента значения 16-битного регистра	38
38. Команды вычитания значения 8-битного регистра из значения аккумулятора с учетом флага переноса	39
39. Команда вычитания непосредственного 8-битного значения из значения аккумулятора с учетом флага переноса.....	39
40. Команда вычитания 8-битного значения в памяти по адресу в регистре HL из значения аккумулятора с учетом флага переноса	39
41. Команды вычитания 8-битного значения в памяти по адресу в индексном регистре (со смещением) из значения аккумулятора с учетом флага переноса	39
42. Команды вычитания значений 16-битных регистровых пар с учетом флага переноса	40
43. Команды сравнения значения 8-битного регистра со значением аккумулятора.....	40
44. Команда сравнения непосредственного 8-битного значения со значением аккумулятора.....	40
45. Команда сравнения 8-битного значения в памяти по адресу в регистре HL со значением аккумулятора	41
46. Команды сравнения 8-битного значения в памяти по адресу в индексном регистре (со смещением) со значением аккумулятора	41
47. Команды логического «И» над значением 8-битного регистра и значением аккумулятора.....	41
48. Команда логического «И» над непосредственным 8-битным значением и значением аккумулятора.....	41
49. Команда логического «И» над 8-битным значением в памяти по адресу в регистре HL и значением аккумулятора	42
50. Команды логического «И» над 8-битным значением в памяти по адресу в индексном регистре (со смещением) и значением аккумулятора	42
51. Команды логического «ИЛИ» над значением 8-битного регистра и значением аккумулятора.....	42
52. Команда логического «ИЛИ» над непосредственным 8-битным значением и значением аккумулятора	42
53. Команда логического «ИЛИ» над 8-битным значением в памяти по адресу в регистре HL и значением аккумулятора	43
54. Команды логического «ИЛИ» над 8-битным значением в памяти по адресу в индексном регистре (со смещением) и значением аккумулятора	43
55. Команды логического «исключающего ИЛИ» над значением 8-битного регистра и значением аккумулятора	43

56. Команда логического «исключающего ИЛИ» над непосредственным 8-битным значением и значением аккумулятора	43
57. Команда логического «исключающего ИЛИ» над 8-битным значением в памяти по адресу в регистре HL и значением аккумулятора	44
58. Команды логического «исключающего ИЛИ» над 8-битным значением в памяти по адресу в индексном регистре (со смещением) и значением аккумулятора	44
59. Команда безусловного перехода по непосредственному адресу	44
60. Команды безусловного перехода по адресу в 16-битном регистре	44
61. Команды условного перехода по непосредственному адресу	44
62. Команда безусловного относительного перехода	45
63. Команды условного относительного перехода	45
64. Команда условного относительного перехода с организацией цикла по регистру B	45
65. Команды помещения значения 16-битной регистровой пары в стек	45
66. Команды снятия значения 16-битной регистровой пары со стека	46
67. Команды системного вызова	46
68. Команда безусловного вызова по непосредственному адресу	46
69. Команды условного вызова по непосредственному адресу	46
70. Команда безусловного возврата	47
71. Команды условного возврата	47
72. Команды логического сдвига вправо значения 8-битного регистра	47
73. Команда логического сдвига вправо 8-битного значения в памяти по адресу в регистре HL	48
74. Команды логического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением)	48
75. Сложные команды логического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре	48
76. Команды арифметического сдвига вправо значения 8-битного регистра	49
77. Команда арифметического сдвига вправо 8-битного значения в памяти по адресу в регистре HL	49
78. Команды арифметического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением)	49
79. Сложные команды арифметического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре	49
80. Команды логического сдвига влево значения 8-битного регистра	50
81. Команда логического сдвига влево 8-битного значения в памяти по адресу в регистре HL	50
82. Команды логического сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением)	50
83. Сложные команды логического сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре	51
84. Команды логического сдвига влево с увеличением значения 8-битного регистра	51
85. Команда логического сдвига влево с увеличением 8-битного значения в памяти по адресу в регистре HL	51
86. Команды логического сдвига влево с увеличением 8-битного значения в памяти по адресу в индексном регистре (со смещением)	52
87. Сложные команды логического сдвига влево с увеличением 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре	52
88. Команды расширенного сдвига влево значения 8-битного регистра	52
89. Команда расширенного сдвига влево 8-битного значения в памяти по адресу в регистре HL	53
90. Команды расширенного сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением)	53

91. Сложные команды расширенного сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре	53
92. Команды расширенного сдвига вправо значения 8-битного регистра.....	54
93. Команда расширенного сдвига вправо 8-битного значения в памяти по адресу в регистре HL	54
94. Команды расширенного сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением).....	54
95. Сложные команды расширенного сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре	55
96. Команды циклического сдвига влево значения 8-битного регистра.....	55
97. Команда циклического сдвига влево 8-битного значения в памяти по адресу в регистре HL	55
98. Команды циклического сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением).....	56
99. Сложные команды циклического сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре	56
100. Команды циклического сдвига вправо значения 8-битного регистра.....	56
101. Команда циклического сдвига вправо 8-битного значения в памяти по адресу в регистре HL	57
102. Команды циклического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением).....	57
103. Сложные команды циклического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре	57
104. Команды расширенного сдвига аккумулятора	58
105. Команды циклического сдвига аккумулятора	58
106. Команды установки бита внутри 8-битного регистра.....	58
107. Команда установки бита внутри 8-битного значения в памяти по адресу в регистре HL.....	59
108. Команды установки бита внутри 8-битного значения в памяти по адресу в индексном регистре (со смещением).....	60
109. Сложные команды установки бита внутри 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре.....	60
110. Команды сброса бита внутри 8-битного регистра.....	62
111. Команда сброса бита внутри 8-битного значения в памяти по адресу в регистре HL.....	63
112. Команды сброса бита внутри 8-битного значения в памяти по адресу в индексном регистре (со смещением).....	64
113. Сложные команды сброса бита внутри 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре.....	64
114. Команды проверки бита внутри 8-битного регистра	66
115. Команда проверки бита внутри 8-битного значения в памяти по адресу в регистре HL.....	68
116. Команды проверки бита внутри 8-битного значения в памяти по адресу в индексном регистре (со смещением).....	68
117. Команды блочной пересылки данных	69
118. Команды блочного поиска (сравнения).....	70
119. Команды ввода из порта ввода/вывода	70
120. Команды блочного ввода из порта ввода/вывода	71
121. Команды вывода в порт ввода/вывода	71
122. Команды блочного вывода в порт ввода/вывода	72
123. Команды управления прерываниями.....	72
124. Прочие команды.....	73

Предисловие

Данный текст является описанием системы команд процессора Z80, используемого в компьютерах, совместимых с ZX-Spectrum. Мной была предпринята попытка составить если и не исчерпывающее, то по крайней мере наиболее подробное описание системы команд данного процессора. Хочется сказать огромное спасибо всем, кто помогал найти необходимую информацию, а также всем тем, кто каким-либо образом принимал участие в создании данного описания.

Задачей ставилось помочь всем (а в первую очередь – себе ;) разобраться в тонкостях системы команд Z80 – простого, но чрезвычайно мощного 8-битного микропроцессора. Этот текст послужит любому пишущему под Z80 ценным справочником – здесь можно найти ту информацию о командах, которая очень часто остается «за кадром», и искать ее приходится в различного рода форумах, разрозненных описаниях «недокументированных особенностей», а иногда даже просто спрашивая у друзей и знакомых. Здесь содержится не только список *всех известных мне на данный момент* команд Z80, но и подробная информация – раскладка кода операции, информация о времени выполнения, о воздействии на флаги, а также о «недокументированных» особенностях каждой конкретной команды. Думаю, что польза от этого описания будет и для тех, кто разрабатывает языки программирования, трансляторы с языка ассемблера, отладчики и дизассемблеры для платформ на базе Z80. Если общественность решит, что пора обобщить всю информацию в одном описании – значит так оно и будет. Милости прошу присылать всю программную и аппаратную информацию, которая только есть. Очень хотелось бы дополнить описание «железными» параметрами – назначением выводов, «расщипкой» шины и т.д.

Может показаться, что Z80 уже отошел в прошлое, все о нем уже известно, а задача не нова и не актуальна. По моему личному опыту это не так. Z80 до сих пор используется как в ZX-Spectrum-совместимых микрокомпьютерах (количество поклонников которых – нас – думаю, до сих пор насчитывает по всему миру не менее миллиона человек), так и в различных бытовых приложениях (яркий пример тому – до сих пор популярные телефоны «Русь», большинство модификаций которых выполнены именно на базе Z80). Кроме того, на базе Z80 очень просто научиться тонкому и эффективному программированию на ассемблере, отчасти из-за простоты архитектуры, а отчасти – из-за необходимости писать быстросействующие программы. Опыт в написании подобного рода программ является незаменимым, и может быть легко перенесен на более производительные платформы.

Конечно, это описание требует серьезной начальной подготовки. Необходимо быть знакомым с азами языка ассемблера, четко представлять себе – что такое команды, регистры и флаги процессора, легко работать с двоичной и шестнадцатеричной системами счисления. Здесь содержится краткий обзор основных «элементов» микропроцессора Z80, однако, этого мало. Думаю, со временем удастся включить сюда большинство теории, а пока – стоит почитать вот это:

1. Как написать игру на ассемблере для ZX Spectrum. – СПб.: Питер, 1995.
2. ZX Spectrum для пользователей и программистов. 2-е изд. – СПб.: Питер, 1993.
3. Журнал ZX-Ревю (ныне доступен в электронном виде на Web-сайте <http://trd.speccy.cz>).

Как видно из заголовка, описание имеет ревизию 0.6. Это значит, что текст еще полностью не готов. Если удалось найти ошибку или просто есть желание что-то дополнить или уточнить, можно прислать свои комментарии по адресу alex@atcg.info. Это поможет как можно быстрее исправить описание с учетом Ваших комментариев. По указанному адресу также можно присылать все отзывы и предложения. Постараюсь ответить на все письма, включить все существенные дополнения и учесть большинство пожеланий.

Алексей Асёмов (Alex/AT)

Основы архитектуры микропроцессора Z80

Этот раздел посвящен основам функционирования микропроцессора Z80. Он содержит теоретическую информацию, которая пригодится при просмотре данного описания. Частично эта информация была почерпнута (но не скопирована целиком) из книги «Как написать игру на ассемблере» издательства «Питер». Кроме того, данная часть поможет новичкам разобраться с основными принципами работы микропроцессора.

1. Двоичная система счисления

Микропроцессор Z80 воспринимает команды и данные с 8-разрядной шины данных, а проще говоря – с 8 контактов. У каждого из этих восьми контактов может быть лишь два состояния – есть заряд или нет заряда. Наличие заряда можно представить как 1, отсутствие – как 0. Последовательности из единиц и нулей дают числа в двоичном представлении, каждый знак которых (0 или 1) называется битом. Эти числа можно легко перевести в десятичный вид. Допустим, у нас есть число 00111100. Самый младший бит (крайнюю правую цифру) умножаем на 1, второй – на 2, третий – на 4, следующий – на 8 и так далее. Тут следует заметить, что биты обычно нумеруются, начиная с 0, поэтому «второй» бит обычно имеет номер 1, «третий» – 2, и т.д. Иными словами, значение каждого бита умножается на 2 в степени n , где n – номер бита. Говоря научным языком, в данном случае 2 – это основание системы счисления. Если Вам такое определение не очень понятно, взгляните на следующую формулу для перевода нашего двоичного числа в десятичное:

$$00111100 = 0*128 + 0*64 + 1*32 + 1*16 + 1*8 + 1*4 + 0*2 + 0*1 = 60.$$

Точно так же можно перевести любое другое двоичное число в десятичное. Отметим, что последовательности из 8 битов называют байтами. Байт – это единица данных, с которой чаще всего и работает микропроцессор. Кроме того, группы из двух байт обычно называют словами, а из четырех – двойными словами.

2. Шестнадцатеричная система счисления

Записывать числа в двоичном представлении бывает крайне неудобно – они занимают много места на экране и плохо читаются. Использовать десятичное представление чисел тоже бывает весьма неудобно, особенно при работе со словами, поскольку максимальное значение байта – число 255, а старший байт слова для перевода в десятичную систему надо умножать на 256 – число крайне «неудобное». Т.е., чтобы задать старший байт слова равным, допустим, 18, а младший – 37, нам придется умножить 18 на 256 (получая при этом 4608), затем прибавлять к этому числу значение младшего байта слова (получая 4645), и записывать полученное число, из которого старший байт слова мы сможем выделить, только снова разделив это число на 256. Поскольку «половинки» одного и того же слова очень часто используются в программах для совершенно различных целей, возникла необходимость перейти от десятичной записи к некоторой другой, более краткой биту. В качестве такой записи была выбрана шестнадцатеричная система счисления. В ней к 10 привычным нам цифрам добавлены еще 6 «цифр», представляющих из себя первые 6 букв латинского алфавита – «A», «B», «C», «D», «E» и «F». Логично, что «A» – это 10, «B» – 11, «C» – 12, «D» – 13, «E» – 14, а «F» – это 15. В такой системе удобно то, что одна ее цифра (принимаящая 16 значений) может быть записана строго четырьмя двоичными цифрами ($16 = 2$ в степени 4). Это облегчает перевод из двоичной системы в шестнадцатеричную и назад. А один байт (2 в степени 8 значений) записывается всего лишь двумя знаками шестнадцатеричной системы. Перевод шестнадцатеричных чисел в десятичную систему выполняется аналогично двоичной системе, только каждая цифра умножается не на ее номер в степени 2, а на ее номер в степени 16 – на 16, 256, 4096, 65536 и так далее. Основанием системы счисления в данном случае является число 16.

В языке ассемблера для Z80 шестнадцатеричные числа принято записывать, предвзяя их знаком #. Таким образом, наше число выглядело бы в шестнадцатеричной системе так: старший байт слова – 18 = #12 ($16*1 + 2$), младший байт слова – 37 = #25 ($16*2 + 5$). Комбинируя эти значения в слово, получаем число #1225, из которого легко можно визуально выделить старший

и младший байты (#12 и #25, т.е. 18 и 37 соответственно). Если мы попробуем перевести это число в десятичное по формуле

$$1*4096 + 2*256 + 2*16 + 5 = 4096 + 512 + 32 + 5 = 4645,$$

то мы легко можем заметить, что перевод двух байт слова по отдельности в шестнадцатеричный вид целого слова нами был выполнен верно.

Еще более простым является перевод из шестнадцатеричной системы в двоичную и наоборот. Его можно выполнять по одному знаку, достаточно лишь помнить, что

$$0 = 0000; 1 = 0001; 2 = 0010; 3 = 0011; 4 = 0100; 5 = 0101; 6 = 0110; 7 = 0111;$$

$$8 = 1000; 9 = 1001; A = 1010; B = 1011; C = 1100; D = 1101; E = 1110; F = 1111.$$

Таким образом, наше число #1225 может быть записано как 0001 0010 0010 0101 в двоичной системе счисления. Это представление можно перевести назад в шестнадцатеричную систему аналогичным образом без особых затруднений. А наше предыдущее число (60 = 00111100) в шестнадцатеричной системе запишется (0011 = 3, 1100 = C) как #3C (3*16 + 12).

Это лишний раз доказывает удобство использования шестнадцатеричной системы счисления для задания машинных слов по сравнению с десятичной. В дальнейшем мы будем использовать для записи команд и значений байт в основном шестнадцатеричную систему, обращаясь к десятичной только в случае необходимости.

3. Память микропроцессора Z80, порты, адресное пространство

Микропроцессор Z80 использует память (постоянную или оперативную), каждая ячейка которой хранит ровно один байт. У каждой ячейки памяти (у каждого байта памяти), доступной микропроцессору, есть свой строго определенный адрес. Все множество доступных процессору адресов памяти называется адресным пространством. Адрес ячейки памяти для микропроцессора Z80 – это слово, принимающее значения от #0000 до #FFFF (65535). Таким образом, адресное пространство Z80 включает в себя 65536 байт. В этом адресном пространстве может любым произвольным образом располагаться как постоянная, так и оперативная память.

Кроме адресного пространства памяти, микропроцессору Z80 доступно второе аналогичное 16-битное адресное пространство, называемое пространством портов ввода-вывода. Каждый порт ввода-вывода можно рассматривать как независимую ячейку памяти, однако обращение к такой ячейке (порту) задействует внешние, обменивающиеся с микропроцессором через данный порт ввода-вывода устройства. Для работы с портами ввода-вывода у микропроцессора Z80 имеются специальные команды ввода из порта (IN) и вывода в порт (OUT).

4. Регистры микропроцессора Z80. Регистровые пары

Регистры – это особые внутренние ячейки памяти микропроцессора, которые необходимы ему для выполнения различных операций. Практически ни одна операция не обходится без участия регистров, а различные арифметические и логические действия без них невозможны в принципе.

Особенностью регистров, в отличие от обычных ячеек памяти, является то, что для обращения к ним используются не адреса, а собственные имена, состоящие из букв латинского алфавита. Эти удобные для человека имена ассоциируются в машинном коде с числовыми индексами, входящими прямо в структуру команды процессора. Еще одна особенность регистров – это их неравноценность. Это означает, что действия, которые возможны с одними регистрами, могут быть невозможны с другими и наоборот. А кроме того, некоторые регистры процессора изменяются автоматически, и работать с ними можно лишь косвенно, с помощью специальных команд. Яркий пример такого регистра – регистр флагов F.

Большинство регистров микропроцессора Z80 имеют размер в 8 бит (байт). Кроме того, еще одной интересной их особенностью является способность объединяться в 16-битовые (двухбайтовые) регистровые пары, и с таким объединенным регистром можно работать, как с целым словом.

В остальном регистры микропроцессора весьма схожи с ячейками памяти. Их значение можно изменять и читать (за исключением системных регистров), информация в них (за некоторым исключением) сохраняется до тех пор, пока не будет изменена программой.

Все регистры могут быть разделены на несколько функциональных групп, учитывая характер функций, которые они выполняют. Основной и наиболее важной группой является группа регистров общего назначения или регистров данных. В нее входят регистры A, B, C, D, E, H и L. Эти регистры могут использоваться для произвольных целей, однако многие из них имеют еще и свое строго определенное назначение. Регистр A называется аккумулятором, и участвует в большинстве арифметических и логических операций в качестве первого операнда, причем результат таких операций обычно сохраняется именно в этом же регистре. Регистр B часто используется в качестве счетчика цикла (у Z80 даже есть специальная команда цикла DJNZ, использующая B в качестве счетчика). Другие регистры этой группы тоже имеют свои особенности, к примеру, регистровая пара HL (H и L) предназначена для использования в качестве адресного регистра для обращения к ячейкам памяти.

Среди регистров общего назначения возможны регистровые пары BC, DE и HL. В них регистры B, D, H являются старшими, а C, E и L – младшими. Существует масса команд, работающих с регистровыми парами, как с целыми словами. Основная часть этих команд использует регистровые пары в качестве адресов памяти. Регистр A тоже имеет «пару», появляющуюся только в командах помещения регистровой пары в стек и снятия регистровой пары со стека. Это регистр F – особый регистр состояния микропроцессора, называемый регистром флагов.

Работая с регистровыми парами, приходится представлять, как двухбайтовые величины размещаются в регистрах и в памяти. Старшие регистры пар (B, D, H) хранят старшие байты (биты 15-8) величин, младшие (C, E, L) – младшие байты (биты 7-0). Шестнадцатеричная запись этих величин аналогична записи регистровых пар – сначала старший байт, потом – младший. В памяти же значения двухбайтовых величин хранятся в обратном порядке, т.е. сначала сохраняется младший байт, а в следующей по счету ячейке памяти – старший.

Следует отметить, что Z80 имеет две группы регистров общего назначения – основную и дополнительную («теневою»). Дополнительная группа регистров для простоты именуется AF', BC', DE' и HL' (регистры со знаками апострофа). Все операции всегда выполняются с основной группой (т.е. операции с дополнительной группой невозможны), однако значения основной и дополнительной групп регистров могут быть в любой момент обменены между собой командами EX AF,AF' (регистр AF с AF') и EXX (регистры BC, DE, HL с BC', DE', HL' соответственно).

Следующая группа – группа индексных регистров – включает в себя две регистровые пары IX и IY. Эти регистры используются для адресации небольших массивов памяти (до 256 байт), а индексными названы потому, что команды адресации памяти, работающие с этими парами, включают в себя так называемый «индекс» – смещение от -128 до 127 относительно адреса, заданного в регистровой паре. В остальном работа с этими регистрами практически аналогична работе с регистром HL, а с половинками индексных регистров можно работать точно так же, как и с половинками регистровой пары HL. Такая возможность долго считалась «недокументированной» особенностью микропроцессора Z80, однако сам набор команд предполагает такую возможность, поэтому ничего «недокументированного» здесь нет. В языке ассемблера для половинок IX и IY обычно используются мнемоники XH/YH (старшие половинки) и XL/YL (младшие половинки).

К системным регистрам микропроцессора относятся неделимые регистровые пары PC (счетчик команд) и SP (указатель стека), а также регистры I и R. Кроме того, у Z80 есть скрытый временный регистр, значение битов 5 и 3 которого можно считывать с использованием регистра флагов. Подробнее об этом регистре можно прочитать в разделе «Особенности функционирования микропроцессора Z80».

Регистр PC называется счетчиком команд. Он хранит в себе адрес текущей выполняемой процессором команды, и увеличивается каждый раз после ее выполнения. Неявно изменяют этот регистр команды перехода, вызова и возврата из процедур (JP, JR, DJNZ, CALL, RST, RET, RETI, RETN), а также команда HALT. Этот регистр также изменяется перед обработкой аппаратных

прерываний на значение адреса обработчика прерывания. Сохранить значение регистра PC стандартными средствами невозможно. Команды CALL и RST помещают значение PC+3 в стек, поэтому оно может быть оттуда прочитано в случае необходимости. Для загрузки регистра PC можно использовать команды JP, JR, CALL, *JP (HL)* (LD PC,HL), *JP (IX)* (LD PC,IX) и *JP (IY)* (LD PC,IY), а также комбинации PUSH/RET. Работа с половинками регистра PC невозможна.

Регистр SP называется указателем стека. Для его загрузки и сохранения существуют специальные команды процессора. Работа с половинками регистра SP невозможна, однако в этом нет большой необходимости, ибо регистр SP, так же, как и регистр PC, имеет служебное назначение. Он указывает адрес вершины области стека, и изменять его нужно лишь совершенно точно представляя, что при этом происходит. Этот регистр неявно изменяется командами PUSH и POP.

Регистр I называется регистром вектора прерываний и содержит указатель на обработчик прерываний в режиме обработки 2. Подробнее об этом регистре можно прочитать в секции «Аппаратные прерывания» раздела «Особенности функционирования микропроцессора Z80».

Регистр R называется регистром регенерации и служит для обеспечения регенерации динамической памяти в микропроцессорной системе. 7 младших бит регистра R изменяются после выполнения каждой команды микропроцессора, а старший бит может быть использован для любых целей. Загрузка регистра R влияет на регенерацию памяти, поэтому использовать данный регистр крайне не рекомендуется. Подробнее об этом регистре можно прочитать в секции «Регистр регенерации» раздела «Особенности функционирования микропроцессора Z80».

Регистры I и R имеют еще одно скрытое назначение – используя команды чтения этих регистров, можно прочитать текущее состояние флага IFF2 разрешения аппаратных прерываний. Подробнее об этой функции можно прочитать в секции «Аппаратные прерывания» раздела «Особенности функционирования микропроцессора Z80».

5. Регистр флагов микропроцессора Z80. Флаги.

Флаги – это отдельные биты, определяющие состояние некоторых условий. Значение бита 1 означает, что флаг (состояние) присутствует – «взведен», «установлен». Значение 0 означает, что флаг (состояние) отсутствует – «опущен», «сброшен». Флаги являются дополнением к результату выполнения многих операций процессора, и позволяют определить состояние результата выполнения этих операций.

В микропроцессоре Z80 под флаги состояния различных условий выделен специальный регистр F, образующий пару с регистром аккумулятора А. Раскладка данного регистра приведена ниже:

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

Рис.1.5.1. Регистр флагов микропроцессора Z80

Флаг C (Carry) – флаг переноса. Этот флаг устанавливается в том случае, если в результате выполнения арифметической операции или операции сдвига произошел перенос. К примеру, сложение байт 254 и 1 дает нам значение 255, помещающееся в байт. В этом случае флаг переноса будет сброшен в 0. Однако сложение байт 254 и 3 даст нам число 257, не помещающееся в байт. В этом случае результатом будет число 1 (257 по модулю 256), а флаг переноса будет установлен в 1, что позволит нам прибавить единицу к старшему байту результата или обработать переполнение. Флаг переноса устанавливается при необходимости заема при вычитании (к примеру, 2-3), кроме того, в него вытесняется «лишний» бит в операциях сдвига.

Флаг N (Negative) – флаг декремента. Это системный флаг, не имеющий ассоциированных операций проверки его значения. Он устанавливается командами уменьшения/вычитания и сбрасывается командами увеличения/сложения. Этот флаг проверяется и используется командой DAA.

Флаг P/V (**Parity/oVerflow**) – флаг четности/переполнения. На этот флаг возложена индикация сразу двух различных состояний, однако эти состояния никогда не пересекаются в одной операции, поэтому такой подход разумен. Флаг четности устанавливается логическими операциями, если число установленных (или сброшенных, что аналогично) бит результата чётно, и сбрасывается в противном случае. Флаг переноса устанавливается арифметическими операциями в случае изменения в результате их выполнения знака операнда.

Флаг H (**Half-carry**) – флаг полупереноса. Он устанавливается, если во время выполнения арифметической операции произошел перенос из одной 4-битовой половинки двоично-десятичного (BCD) числа в другую. Это системный флаг, не имеющий ассоциированных операций проверки его значения, он проверяется и используется командой DAA.

Флаг Z (**Zero**) – флаг нуля. Этот флаг устанавливается только в том случае, если в результате выполнения арифметической операции получился нулевой результат. Кроме того, этот флаг устанавливается операцией проверки бита, если проверяемый бит установлен (и сбрасывается, если бит сброшен).

Флаг S (**Sign**) – флаг знака. Он обычно соответствует значению самого старшего бита результата арифметической операции, т.е. установлен, когда число отрицательно и сброшен, когда число положительно.

Флаги 5 и 3 – системные флаги. Во многих операциях они отражают значения битов 5 и 3 внутреннего временного регистра микропроцессора Z80. Подробнее об этом регистре можно прочитать в разделе «Особенности функционирования микропроцессора Z80».

Далеко не все команды изменяют содержимое регистра флагов. Некоторые команды влияют только на отдельные биты данного регистра, а некоторые – не влияют на регистр флагов вообще. Это удобно, поскольку операция проверки определенного флага может отстоять от операции, изменившей этот флаг на несколько операций, данный флаг не модифицирующих.

Для проверки состояния регистра флагов используются специальные операции условных команд перехода/вызова – C, NC, Z, NZ, PO, PE, P, M. Условие C выполняет переход, если флаг C установлен (равен 1), NC – если флаг C сброшен (равен 0), Z и NZ действуют аналогично для флага Z, PO и PE – для флага P/V, а P и M – для флага S.

Схема формирования кодов операций микропроцессора Z80

Данный раздел посвящен описанию схем формирования кодов операций процессора Z80 в зависимости от типа операции, регистров и других операндов. Этот раздел будет крайне полезен тем, кто создает языки программирования и трансляторы, поскольку он описывает принципы перевода мнемоник команд ассемблера в машинный код процессора.

1. Основные обозначения операций, условий и флагов

Каждая команда микропроцессора, будучи всего лишь числом – машинным кодом, в языке ассемблера имеет свое название, называемое мнемоникой. Мнемоники всегда так или иначе связаны с действием, выполняемым командой. Для тех, кто плохо знает английский язык, понять смысл той или иной мнемоники бывает достаточно трудно, поэтому не лишним будет привести здесь расшифровку всех мнемоник ассемблера Z80, а также некоторых условий. Это облегчит понимание следующих частей.

<u>Мнемоника</u>	<u>Расшифровка</u>
NOP	Нет операции (No OPeration).
LD	Загрузка (LoaD).
INC	Увеличение на 1 (INCrement).
DEC	Уменьшение на 1 (DECrement).
ADD	Сложение (ADD).
ADC	Сложение с заемом (ADD with Carry).
SUB	Вычитание (SUBtract).
SBC	Вычитание с заемом (SuBtract with Carry).
AND	Логическое «И» (AND).
XOR	Логическое «исключающее ИЛИ» (eXclusive OR).
OR	Логическое «ИЛИ» (OR).
CP	Сравнение (ComPare).
CPL	Инверсия аккумулятора (ComPLement).
NEG	Отрицательное значение аккумулятора (NEGative).
DAA	Корректировка BCD-числа после сложения (Decimal Adjust register A).
RLA	Поворот аккумулятора влево с заемом (Rotate Left register A).
RRA	Поворот аккумулятора вправо с заемом (Rotate Right register A).
RLCA	Циклический поворот аккумулятора влево (Rotate Left Cyclic register A).
RRCA	Циклический поворот аккумулятора вправо (Rotate Right Cyclic register A).
RLD	Поворот BCD влево (Rotate Left Decimal).
RRD	Поворот BCD вправо (Rotate Right Decimal).
RLC	Циклический поворот влево (Rotate Left Cyclic).
RRC	Циклический поворот вправо (Rotate Right Cyclic).
RL	Поворот влево с заемом (Rotate Left).
RR	Поворот вправо с заемом (Rotate Right).
SLA	Арифметический сдвиг влево с заполнением нулями (Shift Left Arithmetic).
SRA	Арифметический сдвиг вправо с заполнением знаком (Shift Right Arithmetic).
SLI	Логический сдвиг влево с заполнением единицами (Shift Left Increment).
SRL	Логический сдвиг вправо с заполнением нулями (Shift Right Logical).
BIT	Проверка состояния бита (BIt Test).
RES	Сброс бита (RESet bit).
SET	Установка бита (SET bit).
EX	Обмен регистров (EXchange).
EXX	Обмен наборов регистров (у Z80 два набора регистров BC/DE/HL – основной и дополнительный, все команды используют основной набор).
POP	Снятие со стека (POP).

PUSH	Помещение на стек (PUSH).
JP	Переход (JumP).
JR	Относительный переход (JumP Relative).
DJNZ	Уменьшить В на единицу (DEC) и перейти (JR), если не 0 (NZ).
CALL	Вызов подпрограммы (CALL).
RET	Возврат из подпрограммы (RETurn).
RST	Системный вызов – рестарт (ReStArt).
CCF	Сброс флага переноса (Clear Carry Flag).
SCF	Установка флага переноса (Set Carry Flag).
LDI	Пересылка и увеличение адреса (LoaD and Increment).
LDD	Пересылка и уменьшение адреса (LoaD and Decrement).
LDIR	Пересылка и увеличение адреса с повторением (LoaD and Increment, Repeat).
LDDR	Пересылка и уменьшение адреса с повторением (LoaD and Decrement, Repeat).
CPI	Сравнение и увеличение адреса (ComPare and Increment).
CPD	Сравнение и уменьшение адреса (ComPare and Decrement).
CPIR	Сравнение и увеличение адреса с повторением (ComPare and Increment, Repeat).
CPDR	Сравнение и уменьшение адреса с повторением (ComPare and Decrement, Repeat).
INI	Ввод и увеличение адреса (INput and Increment).
IND	Ввод и уменьшение адреса (INput and Decrement).
INIR	Ввод и увеличение адреса с повторением (INput and Increment, Repeat).
INDR	Ввод и уменьшение адреса с повторением (INput and Decrement, Repeat).
OUTI	Вывод и увеличение адреса (OUTput and Increment).
OUTD	Вывод и уменьшение адреса (OUTput and Decrement).
OUTIR	Вывод и увеличение адреса с повторением (OUTput and Increment, Repeat).
OUTDR	Вывод и уменьшение адреса с повторением (OUTput and Decrement, Repeat).
IN	Ввод из порта ввода/вывода (INput).
OUT	Вывод в порт ввода/вывода (OUTput).
INF	«Пустой» ввод из порта ввода/вывода (INput Fake).
OUTF	«Пустой» вывод в порт ввода/вывода (OUTput Fake).
DI	Запрещение прерываний INT (Disable Interrupt).
EI	Разрешение прерываний INT (Enable Interrupt).
IM	Режим обработки прерываний INT (Interrupt handling Mode).
RETI	Возврат из прерывания INT (RETurn from Interrupt).
RETN	Возврат из немаскируемого прерывания NMI (RETurn from Non-maskable interrupt).
HALT	Останов процессора до прихода прерывания (HALT).

Условие

NZ	Не ноль (Not Zero , Z=0).
Z	Ноль (Zero , Z=1).
NC	Не перенос (Not Carry , C=0).
C	Перенос (Carry , C=1).
PO	Нечет (Parity Odd , P/V=0) или отсутствие переполнения.
PE	Чет (Parity Even , P/V=1) или переполнение.
P	Плюс (Plus , S=0).
M	Минус (Minus , S=1).

Расшифровка

NZ	Не ноль (Not Zero , Z=0).
Z	Ноль (Zero , Z=1).
NC	Не перенос (Not Carry , C=0).
C	Перенос (Carry , C=1).
PO	Нечет (Parity Odd , P/V=0) или отсутствие переполнения.
PE	Чет (Parity Even , P/V=1) или переполнение.
P	Плюс (Plus , S=0).
M	Минус (Minus , S=1).

2. Операции без префикса

В этот список операций входят операции, исполняемые непосредственно. Им не предшествуют никакие префиксы, а сами базовые префиксы входят в данный список операций.

Код операции	Мнемоника	Примечание
00 000 000	NOP	
00 001 000	EX AF,AF'	
00 010 000 <u>ssssssss</u>	DJNZ <u>s</u>	
00 011 000 <u>ssssssss</u>	JR <u>s</u>	
00 1cc 000 <u>ssssssss</u>	JR <u>cc,s</u>	
00 <u>RR</u> 0 001 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD <u>RR</u> , <u>NN</u>	
00 <u>RR</u> 1 001	ADD HL, <u>RR</u>	
00 000 010	LD (BC),A	
00 001 010	LD A,(BC)	
00 010 010	LD (DE),A	
00 011 010	LD A,(DE)	
00 100 010 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD (<u>NN</u>),HL	
00 101 010 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD HL,(<u>NN</u>)	
00 110 010 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD (<u>NN</u>),A	
00 111 010 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD A,(<u>NN</u>)	
00 <u>RR</u> 0 011	INC <u>RR</u>	
00 <u>RR</u> 1 011	DEC <u>RR</u>	
00 <u>rrr</u> 100	INC <u>r</u>	
00 <u>rrr</u> 101	DEC <u>r</u>	
00 <u>rrr</u> 110 <u>NNNNNNNN</u>	LD <u>r</u> , <u>N</u>	
00 000 111	RLCA	
00 001 111	RRCA	
00 010 111	RLA	
00 011 111	RRA	
00 100 111	DAA	
00 101 111	CPL	
00 110 111	SCF	
00 111 111	CCF	
01 <u>ddd</u> <u>rrr</u>	LD <u>d</u> , <u>r</u>	*1
01 110 110	HALT	*2
10 000 <u>rrr</u>	ADD A, <u>r</u>	
10 001 <u>rrr</u>	ADC A, <u>r</u>	
10 010 <u>rrr</u>	SUB <u>r</u>	*3
10 011 <u>rrr</u>	SBC A, <u>r</u>	
10 100 <u>rrr</u>	AND <u>r</u>	*3
10 101 <u>rrr</u>	XOR <u>r</u>	*3
10 110 <u>rrr</u>	OR <u>r</u>	*3
10 111 <u>rrr</u>	CP <u>r</u>	*3
11 <u>ccc</u> 000	RET <u>c</u>	
11 <u>RS</u> 0 001	POP <u>RS</u>	
11 001 001	RET	
11 011 001	EXX	
11 101 001	JP (HL)	*4
11 111 001	LD SP,HL	
11 <u>ccc</u> 010 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	JP <u>c</u> , <u>NN</u>	
11 000 011 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	JP <u>NN</u>	
11 001 011	Префикс #CB	*5
11 010 011 <u>NNNNNNNN</u>	OUT (<u>N</u>),A	*6
11 011 011 <u>NNNNNNNN</u>	IN A,(<u>N</u>)	*6
11 100 011	EX (SP),HL	

Код операции	Мнемоника	Примечание
11 101 011	EX DE,HL	
11 110 011	DI	
11 111 011	EI	
11 <u>ccc</u> 100 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	CALL <u>c</u> , <u>NN</u>	
11 <u>RS0</u> 101	PUSH <u>RS</u>	
11 001 101 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	CALL <u>NN</u>	
11 011 101	Префикс #DD	*5
11 101 101	Префикс #ED	*5
11 111 101	Префикс #FD	*5
11 000 110 <u>NNNNNNNN</u>	ADD A, <u>N</u>	
11 001 110 <u>NNNNNNNN</u>	ADC A, <u>N</u>	
11 010 110 <u>NNNNNNNN</u>	SUB <u>N</u>	*3
11 011 110 <u>NNNNNNNN</u>	SBC A, <u>N</u>	
11 100 110 <u>NNNNNNNN</u>	AND <u>N</u>	*3
11 101 110 <u>NNNNNNNN</u>	XOR <u>N</u>	*3
11 110 110 <u>NNNNNNNN</u>	OR <u>N</u>	*3
11 111 110 <u>NNNNNNNN</u>	CP <u>N</u>	*3
11 <u>ppp</u> 111	RST <u>p</u>	*7

Примечание	Комментарий
*1	В данной команде <u>d</u> (<u>ddd</u>) – регистр-приемник, <u>r</u> (<u>rrr</u>) – регистр-источник.
*2	Команды LD (HL),(HL) не существует. Вместо нее введена команда HALT – останов процессора до прихода прерывания (INT/NMI).
*3	Мнемоники некоторых команд содержат только регистр-источник. Регистром-приемником в данном случае выступает регистр A. Интересно отличие команды SUB от команд ADD/ADC/SBC. Данная команда присваивает регистру A значение A- <u>x</u> . В «стандартном» виде ее мнемоника выглядела бы так: SUB A, <u>x</u> . Это верно и для команд AND <u>x</u> (AND A, <u>x</u> – присвоить A значение A AND <u>x</u>), OR <u>x</u> , XOR <u>x</u> , CP <u>x</u> .
*4	JP (HL) на самом деле – команда JP HL (или LD PC,HL). Почему устоялась данная ошибочная мнемоника команды, автору не известно.
*5	Префиксы выбирают другие наборы инструкций (эти наборы описаны в соответствующих подразделах данного раздела).
*6	Команды OUT (<u>N</u>),A и IN A,(<u>N</u>) используют регистр A, как старшую часть номера порта. Поэтому их «стандартный» вид – OUT (A*256+ <u>N</u>),A и IN A,(A*256+ <u>N</u>).
*7	Команды RST <u>p</u> аналогичны командам CALL <u>p</u> *8, но занимают всего 1 байт.

Мнемоника	Код	Расшифровка
<u>s</u>	<u>ssssssss</u>	Относительное смещение от -128 до +127 от адреса текущей команды (8 бит).
<u>cc</u>	<u>cc</u>	Малый код условия (2 бита): 00 – NZ; 01 – Z; 10 – NC; 11 – C.
<u>c</u>	<u>ccc</u>	Полный код условия (3 бита): 000 – NZ; 001 – Z; 010 – NC; 011 – C; 100 – PO; 101 – PE; 110 – P; 111 – M.
<u>RR</u>	<u>RR</u>	Код 16-битного регистра (2 бита): 00 – BC; 01 – DE; 10 – HL; 11 – SP.
<u>RS</u>	<u>RS</u>	Код 16-битного регистра стековой операции (2 бита): 00 – BC; 01 – DE; 10 – HL; 11 – AF.
<u>r/d</u>	<u>rrr/ddd</u>	Код 8-битного операнда (3 бита): 000 – B; 001 – C; 010 – D; 011 – E; 100 – H; 101 – L; 110 – (HL); 111 – A.
<u>p</u>	<u>ppp</u>	Адрес для команды RST (3 бита, умножается на 8): 000 – #0; 001 – #8; 010 – #10; 011 – #18; 100 – #20; 101 – #28; 110 – #30; 111 – #38.
<u>N</u>	<u>NNNNNNNN</u>	8-битный непосредственный операнд.
<u>NN</u>	<u>NNNNNNNN</u> <u>NNNNNNNN</u>	16-битный непосредственный операнд.

3. Операции с префиксом #DD

В этот список операций входят операции, изменяющиеся после появления префикса #DD – префикса замены операнда HL на IX. Остальные операции после данных префиксов выполняются аналогично беспрефиксным. Половинки регистра IX обычно обозначаются как XH/XL соответственно.

Код операции	Мнемоника	Примечание
00 <u>RR</u> 0 001 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD <u>RR</u> , <u>NN</u>	
00 <u>RR</u> 1 001	ADD IX, <u>RR</u>	
00 100 010 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD (NN),IX	
00 101 010 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD IX,(NN)	
00 <u>RR</u> 0 011	INC <u>RR</u>	
00 <u>RR</u> 1 011	DEC <u>RR</u>	
00 <u>rrr</u> 100 [<u>ssssssss</u>]	INC <u>r</u>	*1
00 <u>rrr</u> 101 [<u>ssssssss</u>]	DEC <u>r</u>	*1
00 <u>rrr</u> 110 [<u>ssssssss</u>] <u>NNNNNNNN</u>	LD <u>r</u> , <u>N</u>	*1
01 <u>ddd</u> <u>rrr</u> [<u>ssssssss</u>]	LD <u>d</u> , <u>r</u>	*1 *2 *3
01 110 110	HALT	*4
10 000 <u>rrr</u> [<u>ssssssss</u>]	ADD A, <u>r</u>	
10 001 <u>rrr</u> [<u>ssssssss</u>]	ADC A, <u>r</u>	
10 010 <u>rrr</u> [<u>ssssssss</u>]	SUB <u>r</u>	*5
10 011 <u>rrr</u> [<u>ssssssss</u>]	SBC A, <u>r</u>	
10 100 <u>rrr</u> [<u>ssssssss</u>]	AND <u>r</u>	*5
10 101 <u>rrr</u> [<u>ssssssss</u>]	XOR <u>r</u>	*5
10 110 <u>rrr</u> [<u>ssssssss</u>]	OR <u>r</u>	*5
10 111 <u>rrr</u> [<u>ssssssss</u>]	CP <u>r</u>	*5
11 <u>RS</u> 0 001	POP <u>RS</u>	
11 011 001	EXX	*6
11 101 001	JP (IX)	*7
11 111 001	LD SP,IX	
11 001 011	Префикс #CB (#DD #CB)	*8
11 100 011	EX (SP),IX	
11 101 011	EX DE,HL	*9
11 <u>RS</u> 0 101	PUSH <u>RS</u>	
11 011 101	Префикс #DD	*8
11 101 101	Префикс #ED	*8
11 111 101	Префикс #FD	*8

Примечание	Комментарий	
*1	В случае использования операнда (IX) в данных командах он принимает вид (IX+s), где s – относительное смещение от адреса в IX. В этом случае 8-битное значение смещения sssssss указывается сразу после кода команды.	
*2	В данной команде d (ddd) – регистр-приемник, r (rrr) – регистр-источник.	
*3	В случае использования в качестве одного из операндов (IX+s), замены регистра второго операнда H/L на XH/XL не происходит, к примеру, команда LD L,(HL) превращается в LD L,(IX+s), а не в LD XL,(IX+s).	
*4	Команды LD (IX+s),(IX+s) не существует. Вместо нее введена команда HALT – останов процессора до прихода прерывания (INT/NMI).	
*5	Мнемоники некоторых команд содержат только регистр-источник. Регистром-приемником в данном случае выступает регистр A. Интересно отличие команды SUB от команд ADD/ADC/SBC. Данная команда присваивает регистру A значение A-x. В «стандартном» виде ее мнемоника выглядела бы так: SUB A,x. Это верно и для команд AND x (AND A,x – присвоить A значение A AND x), OR x, XOR x, CP x.	
*6	Команда EXX обменивает наборы регистров BC/DE/HL аналогично беспрефиксной команде. Регистр IX в обмене не участвует.	
*7	JP (IX) на самом деле – команда JP IX (или LD PC,IX). Почему устоялась данная ошибочная мнемоника команды, автору не известно.	
*8	Префиксы выбирают другие наборы инструкций (эти наборы описаны в соответствующих подразделах данного раздела). Префиксы #DD/#ED/#FD выбирают свои наборы инструкций, префикс #CB в данном случае выбирает расширенный набор инструкций #DD #CB.	
*9	Команда EX DE,HL обменивает содержимое регистров DE и HL аналогично беспрефиксной команде. Регистр IX в обмене не участвует.	
Мнемоника	Код	Расшифровка
Нет (IX+s)	[ssssssss]	Относительное смещение от -128 до +127 от адреса в регистре IX при использовании операнда (IX+s) (8 бит).
RR	RR	Код 16-битного регистра (2 бита): 00 – BC; 01 – DE; 10 – IX; 11 – SP.
RS	RS	Код 16-битного регистра стековой операции (2 бита): 00 – BC; 01 – DE; 10 – IX; 11 – AF.
r/d	rrr/ddd	Код 8-битного операнда (3 бита): 000 – B; 001 – C; 010 – D; 011 – E; 100 – XH или H (см. *3); 101 – XL или L (см. *3); 110 – (IX+s); 111 – A.
N	NNNNNNNN	8-битный непосредственный операнд.
NN	NNNNNNNN	16-битный непосредственный операнд.
	NNNNNNNN	

4. Операции с префиксом #FD

В этот список операций входят операции, изменяющиеся после появления префикса #FD – префикса замены операнда HL на IX. Остальные операции после данных префиксов выполняются аналогично беспрефиксным. Половинки регистра IX обычно обозначаются как YH/YL соответственно.

Код операции				Мнемоника	Примечание
00	<u>RR</u> 0	001	<u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD <u>RR</u> , <u>NN</u>	
00	<u>RR</u> 1	001		ADD IX, <u>RR</u>	
00	100	010	<u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD (NN),IX	
00	101	010	<u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD IX,(NN)	
00	<u>RR</u> 0	011		INC <u>RR</u>	
00	<u>RR</u> 1	011		DEC <u>RR</u>	
00	<u>rrr</u>	100	[<u>ssssssss</u>]	INC <u>r</u>	*1
00	<u>rrr</u>	101	[<u>ssssssss</u>]	DEC <u>r</u>	*1
00	<u>rrr</u>	110	[<u>ssssssss</u>] <u>NNNNNNNN</u>	LD <u>r</u> , <u>N</u>	*1
01	<u>ddd</u>	<u>rrr</u>	[<u>ssssssss</u>]	LD <u>d</u> , <u>r</u>	*1 *2 *3
01	110	110		HALT	*4
10	000	<u>rrr</u>	[<u>ssssssss</u>]	ADD A, <u>r</u>	
10	001	<u>rrr</u>	[<u>ssssssss</u>]	ADC A, <u>r</u>	
10	010	<u>rrr</u>	[<u>ssssssss</u>]	SUB <u>r</u>	*5
10	011	<u>rrr</u>	[<u>ssssssss</u>]	SBC A, <u>r</u>	
10	100	<u>rrr</u>	[<u>ssssssss</u>]	AND <u>r</u>	*5
10	101	<u>rrr</u>	[<u>ssssssss</u>]	XOR <u>r</u>	*5
10	110	<u>rrr</u>	[<u>ssssssss</u>]	OR <u>r</u>	*5
10	111	<u>rrr</u>	[<u>ssssssss</u>]	CP <u>r</u>	*5
11	<u>RS</u> 0	001		POP <u>RS</u>	
11	011	001		EXX	*6
11	101	001		JP (IX)	*7
11	111	001		LD SP,IX	
11	001	011		Префикс #CB (#FD #CB)	*8
11	100	011		EX (SP),IX	
11	101	011		EX DE,HL	*9
11	<u>RS</u> 0	101		PUSH <u>RS</u>	
11	011	101		Префикс #DD	*8
11	101	101		Префикс #ED	*8
11	111	101		Префикс #FD	*8

Примечание	Комментарий	
*1	В случае использования операнда (IY) в данных командах он принимает вид (IY+s), где s – относительное смещение от адреса в IY. В этом случае 8-битное значение смещения sssssss указывается сразу после кода команды.	
*2	В данной команде d (ddd) – регистр-приемник, r (rrr) – регистр-источник.	
*3	В случае использования в качестве одного из операндов (IY+s), замены регистра второго операнда H/L на YH/YL не происходит, к примеру, команда LD L,(HL) превращается в LD L,(IY+s), а не в LD YL,(IY+s).	
*4	Команды LD (IY+s),(IY+s) не существует. Вместо нее введена команда HALT – останов процессора до прихода прерывания (INT/NMI).	
*5	Мнемоники некоторых команд содержат только регистр-источник. Регистром-приемником в данном случае выступает регистр A. Интересно отличие команды SUB от команд ADD/ADC/SBC. Данная команда присваивает регистру A значение A-x. В «стандартном» виде ее мнемоника выглядела бы так: SUB A,x. Это верно и для команд AND x (AND A,x – присвоить A значение A AND x), OR x, XOR x, CP x.	
*6	Команда EXX обменивает наборы регистров BC/DE/HL аналогично беспрефиксной команде. Регистр IY в обмене не участвует.	
*7	JP (IY) на самом деле – команда JP IY (или LD PC,IY). Почему устоялась данная ошибочная мнемоника команды, автору не известно.	
*8	Префиксы выбирают другие наборы инструкций (эти наборы описаны в соответствующих подразделах данного раздела). Префиксы #DD/#ED/#FD выбирают свои наборы инструкций, префикс #CB в данном случае выбирает расширенный набор инструкций #FD #CB.	
*9	Команда EX DE,HL обменивает содержимое регистров DE и HL аналогично беспрефиксной команде. Регистр IY в обмене не участвует.	
Мнемоника	Код	Расшифровка
Нет (IX+s)	[ssssssss]	Относительное смещение от -128 до +127 от адреса в регистре IY при использовании операнда (IY+s) (8 бит).
RR	RR	Код 16-битного регистра (2 бита): 00 – BC; 01 – DE; 10 – IY; 11 – SP.
RS	RS	Код 16-битного регистра стековой операции (2 бита): 00 – BC; 01 – DE; 10 – IY; 11 – AF.
r/d	rrr/ddd	Код 8-битного операнда (3 бита): 000 – B; 001 – C; 010 – D; 011 – E; 100 – YH или H (см. *3); 101 – YL или L (см. *3); 110 – (IY+s); 111 – A.
N	NNNNNNNN	8-битный непосредственный операнд.
NN	NNNNNNNN	16-битный непосредственный операнд.
	NNNNNNNN	

5. Операции с префиксом #ED

В этот список операций входят операции, выполняющиеся после появления префикса #ED – модификатора расширенного набора общих команд.

Код операции	Мнемоника	Примечание
00 <u>???</u> <u>???</u>	NOP	*1
01 <u>rrr</u> 000	IN <u>r</u> ,(C)	*2
01 110 000	INF	*2
01 <u>rrr</u> 001	OUT (C), <u>r</u>	*3
01 110 001	OUTF	*3
01 <u>RR</u> 0 010	SBC HL, <u>RR</u>	
01 <u>RR</u> 1 010	ADC HL, <u>RR</u>	
01 <u>RR</u> 0 011 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD (<u>NN</u>), <u>RR</u>	*4
01 <u>RR</u> 1 011 <u>NNNNNNNN</u> <u>NNNNNNNN</u>	LD <u>RR</u> ,(<u>NN</u>)	*4
01 <u>???</u> 100	NEG	*1
01 <u>???</u> 0 101	RETN	*1
01 <u>???</u> 1 101	RETI	*1
01 <u>?ii</u> 110	IM <u>i</u>	*1
01 000 111	LD I,A	
01 001 111	LD R,A	
01 010 111	LD A,I	
01 011 111	LD A,R	
01 100 111	RLD	
01 101 111	RRD	
01 11? 111	NOP	*1
10 0?? <u>???</u>	NOP	*1
10 100 000	LDI	
10 101 000	LDD	
10 110 000	LDIR	
10 111 000	LDDR	
10 100 001	CPI	
10 101 001	CPD	
10 110 001	CPIR	
10 111 001	CPDR	
10 100 010	INI	
10 101 010	IND	
10 110 010	INIR	
10 111 010	INDR	
10 100 011	OUTI	
10 101 011	OUTD	
10 110 011	OUTIR	
10 111 011	OUTDR	
10 1?? 1??	NOP	*1
11 <u>???</u> <u>???</u>	NOP	*1

Примечание	Комментарий
*1	Работа данных групп команд до конца не изучена. Предположительно (это подтверждается практикой) они выполняются аналогично первой команде группы (команде, где все неопределенные биты равны на 0). NOP означает, что данная группа команд не выполняет никаких действий, что подтверждается практикой, однако полностью не проверено.
*2	Команда IN (HL),(C) вводит значение из порта ввода/вывода, выполняя цикл ввода и модифицирует флаги, однако введенное значение не сохраняет. Распространенные мнемоники: INF; INZ; IN (C); IN 0,(C).
*3	Команда OUT (C),HL выводит в порт ввода/вывода значение из внутреннего регистра процессора. Процессор Z80 всегда выводит в порт значение 0, Z85 – значение #FF. Другие совместимые процессоры могут выводить и другие значения, поэтому использование данной команды крайне не рекомендуется. Распространенные мнемоники: OUTF; OUTZ; OUT (C).
*4	Для операнда HL у процессора Z80 есть беспрефиксные аналоги данных команд. Они выполняются быстрее команд с префиксом #ED.

Мнемоника	Код	Расшифровка
Нет	?	Неопределенный бит (может принимать любое значение).
<u>RR</u>	<u>RR</u>	Код 16-битного регистра (2 бита): 00 – BC; 01 – DE; 10 – HL; 11 – SP.
<u>r</u>	<u>rr</u>	Код 8-битного операнда (3 бита): 000 – B; 001 – C; 010 – D; 011 – E; 100 – H; 101 – L; 110 – (HL); 111 – A.
<u>i</u>	<u>ii</u>	Код режима обработки прерываний (2 бита): 00 – 0; 01 – 0; 10 – 1; 11 – 2.
<u>NN</u>	<u>NNNNNNNN</u> <u>NNNNNNNN</u>	16-битный непосредственный операнд.

6. Операции с префиксом #CB

В этот список операций входят операции, выполняющиеся после появления префикса #CB – модификатора расширенного набора битовых и сдвиговых команд.

Код операции	Мнемоника	Примечание
00 000 <u>rrr</u>	RLC <u>r</u>	
00 001 <u>rrr</u>	RRC <u>r</u>	
00 010 <u>rrr</u>	RL <u>r</u>	
00 011 <u>rrr</u>	RR <u>r</u>	
00 100 <u>rrr</u>	SLA <u>r</u>	
00 101 <u>rrr</u>	SRA <u>r</u>	
00 110 <u>rrr</u>	SLI <u>r</u>	
00 111 <u>rrr</u>	SRL <u>r</u>	
01 <u>bbb</u> <u>rrr</u>	BIT <u>b</u> , <u>r</u>	
10 <u>bbb</u> <u>rrr</u>	RES <u>b</u> , <u>r</u>	
11 <u>bbb</u> <u>rrr</u>	SET <u>b</u> , <u>r</u>	

Мнемоника	Код	Расшифровка
<u>r</u>	<u>rrr</u>	Код 8-битного операнда (3 бита): 000 – В; 001 – С; 010 – D; 011 – E; 100 – H; 101 – L; 110 – (HL); 111 – A.
<u>b</u>	<u>bbb</u>	Номер бита для проверки/сброса/установки (3 бита). Значение 0 – младший бит операнда. Значение 7 – старший бит операнда.

7. Операции с расширенным префиксом #DD #CB

В этот список операций входят операции, которым предшествуют два префикса – сначала #DD (префикс замены операнда HL на IX), а затем – #CB (модификатор расширенного набора битовых и сдвиговых команд).

Код операции	Мнемоника	Примечание
<u>ssssssss</u> 00 000 <u>rrr</u>	RLC <u>r</u> ,(IX+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 001 <u>rrr</u>	RRC <u>r</u> ,(IX+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 010 <u>rrr</u>	RL <u>r</u> ,(IX+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 011 <u>rrr</u>	RR <u>r</u> ,(IX+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 100 <u>rrr</u>	SLA <u>r</u> ,(IX+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 101 <u>rrr</u>	SRA <u>r</u> ,(IX+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 110 <u>rrr</u>	SLI <u>r</u> ,(IX+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 111 <u>rrr</u>	SRL <u>r</u> ,(IX+ <u>s</u>)	*1 *2
<u>ssssssss</u> 01 <u>bbb</u> <u>???</u>	BIT <u>b</u> ,(IX+ <u>s</u>)	*1 *3
<u>ssssssss</u> 10 <u>bbb</u> <u>rrr</u>	RES <u>b</u> , <u>r</u> ,(IX+ <u>s</u>)	*1 *2
<u>ssssssss</u> 11 <u>bbb</u> <u>rrr</u>	SET <u>b</u> , <u>r</u> ,(IX+ <u>s</u>)	*1 *2

Примечание Комментарий

- *1** Всем командам с расширенным префиксом #DD #CB предшествует 8-битное значение смещения ssssssss. Это обусловлено тем, что префикс #DD обязывает к появлению этого значения после кода команды, а кодом команды в данном случае является префикс #CB.
- *2** Если основной операнд команды r – (IX+s), то операция выполняется только с этим операндом, в противном случае – и с регистром, и с (IX+s). Особенности функционирования двухоперандных команд можно найти в разделе «Подробное описание команд микропроцессора Z80».
- *3** Все команды данной группы команд выполняются как BIT b,(IX+s).

Мнемоника	Код	Расшифровка
Нет	<u>?</u>	Неопределенный бит (может принимать любое значение).
<u>s</u>	<u>ssssssss</u>	Относительное смещение от -128 до +127 от адреса в регистре IX (8 бит).
<u>r</u>	<u>rrr</u>	Код 8-битного операнда (3 бита): 000 – В; 001 – С; 010 – D; 011 – E; 100 – H; 101 – L; 110 – (IX+ <u>s</u>); 111 – A.
<u>b</u>	<u>bbb</u>	Номер бита для проверки/сброса/установки (3 бита). Значение 0 – младший бит операнда. Значение 7 – старший бит операнда.

8. Операции с расширенным префиксом #FD #CB

В этот список операций входят операции, которым предшествуют два префикса – сначала #FD (префикс замены операнда HL на IY), а затем – #CB (модификатор расширенного набора битовых и сдвиговых команд).

Код операции	Мнемоника		Примечание
<u>ssssssss</u> 00 000 <u>rrr</u>	RLC	<u>r</u> ,(IY+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 001 <u>rrr</u>	RRC	<u>r</u> ,(IY+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 010 <u>rrr</u>	RL	<u>r</u> ,(IY+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 011 <u>rrr</u>	RR	<u>r</u> ,(IY+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 100 <u>rrr</u>	SLA	<u>r</u> ,(IY+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 101 <u>rrr</u>	SRA	<u>r</u> ,(IY+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 110 <u>rrr</u>	SLI	<u>r</u> ,(IY+ <u>s</u>)	*1 *2
<u>ssssssss</u> 00 111 <u>rrr</u>	SRL	<u>r</u> ,(IY+ <u>s</u>)	*1 *2
<u>ssssssss</u> 01 <u>bbb</u> <u>???</u>	BIT	<u>b</u> ,(IY+ <u>s</u>)	*1 *3
<u>ssssssss</u> 10 <u>bbb</u> <u>rrr</u>	RES	<u>b</u> , <u>r</u> ,(IY+ <u>s</u>)	*1 *2
<u>ssssssss</u> 11 <u>bbb</u> <u>rrr</u>	SET	<u>b</u> , <u>r</u> ,(IY+ <u>s</u>)	*1 *2

Примечание	Комментарий
*1	Всем командам с расширенным префиксом #FD #CB предшествует 8-битное значение смещения <u>ssssssss</u> . Это обусловлено тем, что префикс #FD обязывает к появлению этого значения после кода команды, а кодом команды в данном случае является префикс #CB.
*2	Если основной операнд команды <u>r</u> – (IY+ <u>s</u>), то операция выполняется только с этим операндом, в противном случае – и с регистром, и с (IY+ <u>s</u>). Особенности функционирования двухоперандных команд можно найти в разделе «Подробное описание команд микропроцессора Z80».
*3	Все команды данной группы команд выполняются как BIT <u>b</u> ,(IY+ <u>s</u>).

Мнемоника	Код	Расшифровка
Нет	<u>?</u>	Неопределенный бит (может принимать любое значение).
<u>s</u>	<u>ssssssss</u>	Относительное смещение от -128 до +127 от адреса в регистре IY (8 бит).
<u>r</u>	<u>rrr</u>	Код 8-битного операнда (3 бита): 000 – В; 001 – С; 010 – D; 011 – E; 100 – H; 101 – L; 110 – (IY+ <u>s</u>); 111 – А.
<u>b</u>	<u>bbb</u>	Номер бита для проверки/сброса/установки (3 бита). Значение 0 – младший бит операнда. Значение 7 – старший бит операнда.

Подробное описание команд микропроцессора Z80

Данный раздел содержит подробное описание всех имеющихся команд микропроцессора Z80. Все команды в данном разделе сгруппированы по выполняемой ими функции. Приводимые таблицы содержат информацию о мнемонике команды, коде операции, воздействии команды на флаги и времени выполнения команды в тактах. Кроме сводной таблицы, каждая группа команд содержит ее функциональное описание. Особенности некоторых команд отдельно вынесены в раздел «Особенности функционирования микропроцессора Z80», поскольку их описание весьма обширно.

В таблицах активно используются различные обозначения. Поле мнемоники содержит мнемоническую запись команды. В этом поле могут появляться хорошо знакомые аргументы N – 8-битный операнд, NN – 16-битный операнд и s – 8-битное относительное смещение. Поле кода операции содержит побайтную шестнадцатеричную (без символов #) запись команды. Здесь могут встречаться значения NN – байт операнда и ss – байт смещения. Два NN подряд в поле кода операции соответствуют 16-битному операнду NN в поле мнемоники.

Поля воздействия на флаги используют следующие обозначения: «-» – команда не влияет на данный флаг. «+» – команда изменяет значение данного флага. «0» – команда сбрасывает флаг в 0. «1» – команда устанавливает флаг в 1. «P» – команда изменяет флаг P/V в зависимости от четности результата. «V» – команда изменяет флаг P/V в зависимости от состояния переполнения. «3», «5», «7» – команда выставляет флаг в соответствии с указанным битом внутреннего регистра. «*» – воздействие команды на данный флаг сложное и описано в функциональном описании.

Поле тактов содержит количество тактов, необходимое Z80 для исполнения данной команды (включая все префиксы).

1. Команды загрузки 8-битного регистра непосредственным 8-битным значением

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD A,N	3E NN	-	-	-	-	-	-	-	-	7
LD B,N	06 NN	-	-	-	-	-	-	-	-	7
LD C,N	0E NN	-	-	-	-	-	-	-	-	7
LD D,N	16 NN	-	-	-	-	-	-	-	-	7
LD E,N	1E NN	-	-	-	-	-	-	-	-	7
LD H,N	26 NN	-	-	-	-	-	-	-	-	7
LD L,N	2E NN	-	-	-	-	-	-	-	-	7
LD XH,N	DD 26 NN	-	-	-	-	-	-	-	-	11
LD XL,N	DD 2E NN	-	-	-	-	-	-	-	-	11
LD YH,N	FD 26 NN	-	-	-	-	-	-	-	-	11
LD YL,N	FD 2E NN	-	-	-	-	-	-	-	-	11

Команды данной группы загружают в указанный 8-битный регистр значение, следующее непосредственно за командой. Влияния на флаги данные команды не оказывают.

2. Команды загрузки 16-битного регистра непосредственным 16-битным значением

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD BC,NN	01 NN NN	-	-	-	-	-	-	-	-	10
LD DE,NN	11 NN NN	-	-	-	-	-	-	-	-	10
LD HL,NN	21 NN NN	-	-	-	-	-	-	-	-	10
LD SP,NN	31 NN NN	-	-	-	-	-	-	-	-	10
LD IX,NN	DD 21 NN NN	-	-	-	-	-	-	-	-	10
LD IY,NN	FD 21 NN NN	-	-	-	-	-	-	-	-	10

Команды данной группы загружают в указанный 16-битный регистр значение, следующее непосредственно за командой. Влияния на флаги данные команды не оказывают.

3. Команды загрузки 8-битного регистра значением 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD A,A	7F	-	-	-	-	-	-	-	-	4
LD A,B	78	-	-	-	-	-	-	-	-	4
LD A,C	79	-	-	-	-	-	-	-	-	4
LD A,D	7A	-	-	-	-	-	-	-	-	4
LD A,E	7B	-	-	-	-	-	-	-	-	4
LD A,H	7C	-	-	-	-	-	-	-	-	4
LD A,L	7D	-	-	-	-	-	-	-	-	4
LD A,XH	DD 7C	-	-	-	-	-	-	-	-	8
LD A,XL	DD 7D	-	-	-	-	-	-	-	-	8
LD A,YH	FD 7C	-	-	-	-	-	-	-	-	8
LD A,YL	FD 7D	-	-	-	-	-	-	-	-	8
LD B,A	47	-	-	-	-	-	-	-	-	4
LD B,B	40	-	-	-	-	-	-	-	-	4
LD B,C	41	-	-	-	-	-	-	-	-	4
LD B,D	42	-	-	-	-	-	-	-	-	4
LD B,E	43	-	-	-	-	-	-	-	-	4
LD B,H	44	-	-	-	-	-	-	-	-	4
LD B,L	45	-	-	-	-	-	-	-	-	4
LD B,XH	DD 44	-	-	-	-	-	-	-	-	8
LD B,XL	DD 45	-	-	-	-	-	-	-	-	8
LD B,YH	FD 44	-	-	-	-	-	-	-	-	8
LD B,YL	FD 45	-	-	-	-	-	-	-	-	8
LD C,A	4F	-	-	-	-	-	-	-	-	4
LD C,B	48	-	-	-	-	-	-	-	-	4
LD C,C	49	-	-	-	-	-	-	-	-	4
LD C,D	4A	-	-	-	-	-	-	-	-	4
LD C,E	4B	-	-	-	-	-	-	-	-	4
LD C,H	4C	-	-	-	-	-	-	-	-	4
LD C,L	4D	-	-	-	-	-	-	-	-	4
LD C,XH	DD 4C	-	-	-	-	-	-	-	-	8
LD C,XL	DD 4D	-	-	-	-	-	-	-	-	8
LD C,YH	FD 4C	-	-	-	-	-	-	-	-	8
LD C,YL	FD 4D	-	-	-	-	-	-	-	-	8
LD D,A	57	-	-	-	-	-	-	-	-	4
LD D,B	50	-	-	-	-	-	-	-	-	4
LD D,C	51	-	-	-	-	-	-	-	-	4
LD D,D	52	-	-	-	-	-	-	-	-	4
LD D,E	53	-	-	-	-	-	-	-	-	4
LD D,H	54	-	-	-	-	-	-	-	-	4
LD D,L	55	-	-	-	-	-	-	-	-	4
LD D,XH	DD 54	-	-	-	-	-	-	-	-	8
LD D,XL	DD 55	-	-	-	-	-	-	-	-	8
LD D,YH	FD 54	-	-	-	-	-	-	-	-	8
LD D,YL	FD 55	-	-	-	-	-	-	-	-	8
LD E,A	5F	-	-	-	-	-	-	-	-	4
LD E,B	58	-	-	-	-	-	-	-	-	4
LD E,C	59	-	-	-	-	-	-	-	-	4
LD E,D	5A	-	-	-	-	-	-	-	-	4

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD E,E	5B	-	-	-	-	-	-	-	-	4
LD E,H	5C	-	-	-	-	-	-	-	-	4
LD E,L	5D	-	-	-	-	-	-	-	-	4
LD E,XH	DD 5C	-	-	-	-	-	-	-	-	8
LD E,XL	DD 5D	-	-	-	-	-	-	-	-	8
LD E,YH	FD 5C	-	-	-	-	-	-	-	-	8
LD E,YL	FD 5D	-	-	-	-	-	-	-	-	8
LD H,A	67	-	-	-	-	-	-	-	-	4
LD H,B	60	-	-	-	-	-	-	-	-	4
LD H,C	61	-	-	-	-	-	-	-	-	4
LD H,D	62	-	-	-	-	-	-	-	-	4
LD H,E	63	-	-	-	-	-	-	-	-	4
LD H,H	64	-	-	-	-	-	-	-	-	4
LD H,L	65	-	-	-	-	-	-	-	-	4
LD L,A	6F	-	-	-	-	-	-	-	-	4
LD L,B	68	-	-	-	-	-	-	-	-	4
LD L,C	69	-	-	-	-	-	-	-	-	4
LD L,D	6A	-	-	-	-	-	-	-	-	4
LD L,E	6B	-	-	-	-	-	-	-	-	4
LD L,H	6C	-	-	-	-	-	-	-	-	4
LD L,L	6D	-	-	-	-	-	-	-	-	4
LD XH,A	DD 67	-	-	-	-	-	-	-	-	8
LD XH,B	DD 60	-	-	-	-	-	-	-	-	8
LD XH,C	DD 61	-	-	-	-	-	-	-	-	8
LD XH,D	DD 62	-	-	-	-	-	-	-	-	8
LD XH,E	DD 63	-	-	-	-	-	-	-	-	8
LD XH,XH	DD 64	-	-	-	-	-	-	-	-	8
LD XH,XL	DD 65	-	-	-	-	-	-	-	-	8
LD XL,A	DD 6F	-	-	-	-	-	-	-	-	8
LD XL,B	DD 68	-	-	-	-	-	-	-	-	8
LD XL,C	DD 69	-	-	-	-	-	-	-	-	8
LD XL,D	DD 6A	-	-	-	-	-	-	-	-	8
LD XL,E	DD 6B	-	-	-	-	-	-	-	-	8
LD XL,XH	DD 6C	-	-	-	-	-	-	-	-	8
LD XL,XL	DD 6D	-	-	-	-	-	-	-	-	8
LD YH,A	FD 67	-	-	-	-	-	-	-	-	8
LD YH,B	FD 60	-	-	-	-	-	-	-	-	8
LD YH,C	FD 61	-	-	-	-	-	-	-	-	8
LD YH,D	FD 62	-	-	-	-	-	-	-	-	8
LD YH,E	FD 63	-	-	-	-	-	-	-	-	8
LD YH,YH	FD 64	-	-	-	-	-	-	-	-	8
LD YH,YL	FD 65	-	-	-	-	-	-	-	-	8
LD YL,A	FD 6F	-	-	-	-	-	-	-	-	8
LD YL,B	FD 68	-	-	-	-	-	-	-	-	8
LD YL,C	FD 69	-	-	-	-	-	-	-	-	8
LD YL,D	FD 6A	-	-	-	-	-	-	-	-	8
LD YL,E	FD 6B	-	-	-	-	-	-	-	-	8
LD YL,YH	FD 6C	-	-	-	-	-	-	-	-	8
LD YL,YL	FD 6D	-	-	-	-	-	-	-	-	8

Команды данной группы загружают в указанный 8-битный регистр-приемник (первый указанный регистр) значение 8-битного регистра-источника (второй указанный регистр). Влияния на флаги данные команды не оказывают.

4. Команды загрузки, использующие служебный 8-битный регистр в качестве одного из операндов

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD I,A	ED 47	-	-	-	-	-	-	-	-	9
LD A,I	ED 57	+	+	-	0	-	*1	0	-	9
LD R,A	ED 4F	-	-	-	-	-	-	-	-	9
LD A,R	ED 5F	+	+	-	0	-	*1	0	-	9

Команды данной группы используются для загрузки аккумулятора значением служебных 8-битных регистров I и R, или для загрузки этих регистров значением аккумулятора. Команды LD A,I и LD A,R выставляют флаги S и Z в соответствии со значением, загруженным в A.

***1** Флаг P/V после выполнения этих команд содержит значение флага IFF2 разрешения аппаратных прерываний. Подробнее об этой функции можно прочитать в секции «Аппаратные прерывания» раздела «Особенности функционирования микропроцессора Z80».

5. Команды загрузки 16-битного регистра значением 16-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD SP,HL	F9	-	-	-	-	-	-	-	-	6
LD SP,IX	DD F9	-	-	-	-	-	-	-	-	10
LD SP,IY	FD F9	-	-	-	-	-	-	-	-	10

Команды данной группы загружают в указанную 16-битную регистровую пару-приемник (первый указанный регистр) значение 16-битной регистровой пары-источника (второй указанный регистр). Влияния на флаги данные команды не оказывают.

6. Команды загрузки 8-битного регистра значением в памяти по абсолютному адресу

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD A,(NN)	3A NN NN	-	-	-	-	-	-	-	-	13
LD BC,(NN)	ED 4B NN NN	-	-	-	-	-	-	-	-	20
LD DE,(NN)	ED 5B NN NN	-	-	-	-	-	-	-	-	20
LD HL,(NN) *1	2A NN NN	-	-	-	-	-	-	-	-	16
LD HL,(NN) *1	ED 6B NN NN	-	-	-	-	-	-	-	-	20
LD SP,(NN)	ED 7B NN NN	-	-	-	-	-	-	-	-	20
LD IX,(NN)	DD 2A NN NN	-	-	-	-	-	-	-	-	20
LD IY,(NN)	FD 2A NN NN	-	-	-	-	-	-	-	-	20

Команды данной группы загружают в указанный 8-битный регистр или 16-битную регистровую пару значение, находящееся по указанному адресу памяти. Адрес следует непосредственно за командой. Влияния на флаги данные команды не оказывают.

***1** Команда LD HL,(NN) может быть записана в двух вариантах – кодом операции 2A или кодом операции ED 6B. Запись ED 6B занимает на 1 байт больше и выполняется на 4 такта дольше за счет префикса, поэтому ее использование не рекомендуется.

7. Команды загрузки 8-битного регистра значением в памяти по адресу в 16-битной регистровой паре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD A,(BC)	0A	-	-	-	-	-	-	-	-	7
LD A,(DE)	1A	-	-	-	-	-	-	-	-	7
LD A,(HL)	7E	-	-	-	-	-	-	-	-	7
LD B,(HL)	46	-	-	-	-	-	-	-	-	7
LD C,(HL)	4E	-	-	-	-	-	-	-	-	7
LD D,(HL)	56	-	-	-	-	-	-	-	-	7
LD E,(HL)	5E	-	-	-	-	-	-	-	-	7
LD H,(HL)	66	-	-	-	-	-	-	-	-	7
LD L,(HL)	6E	-	-	-	-	-	-	-	-	7

Команды данной группы загружают в указанный 8-битный регистр значение, находящееся по указанному адресу памяти. Адрес хранится в одной из 16-битных регистровых пар, искомая регистровая пара указывается в команде. Влияния на флаги данные команды не оказывают.

8. Команды загрузки 8-битного регистра значением в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD A,(IX+s)	DD 7E ss	-	-	-	-	-	-	-	-	19
LD B,(IX+s)	DD 46 ss	-	-	-	-	-	-	-	-	19
LD C,(IX+s)	DD 4E ss	-	-	-	-	-	-	-	-	19
LD D,(IX+s)	DD 56 ss	-	-	-	-	-	-	-	-	19
LD E,(IX+s)	DD 5E ss	-	-	-	-	-	-	-	-	19
LD H,(IX+s)	DD 66 ss	-	-	-	-	-	-	-	-	19
LD L,(IX+s)	DD 6E ss	-	-	-	-	-	-	-	-	19
LD A,(IY+s)	FD 7E ss	-	-	-	-	-	-	-	-	19
LD B,(IY+s)	FD 46 ss	-	-	-	-	-	-	-	-	19
LD C,(IY+s)	FD 4E ss	-	-	-	-	-	-	-	-	19
LD D,(IY+s)	FD 56 ss	-	-	-	-	-	-	-	-	19
LD E,(IY+s)	FD 5E ss	-	-	-	-	-	-	-	-	19
LD H,(IY+s)	FD 66 ss	-	-	-	-	-	-	-	-	19
LD L,(IY+s)	FD 6E ss	-	-	-	-	-	-	-	-	19

Команды данной группы загружают в указанный 8-битный регистр значение, находящееся по указанному адресу памяти. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется. Влияния на флаги данные команды не оказывают.

9. Команды помещения значения регистра в память по абсолютному адресу

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD (NN),A	32 NN NN	-	-	-	-	-	-	-	-	13
LD (NN),BC	ED 43 NN NN	-	-	-	-	-	-	-	-	20
LD (NN),DE	ED 53 NN NN	-	-	-	-	-	-	-	-	20
LD (NN),HL *1	22 NN NN	-	-	-	-	-	-	-	-	16
LD (NN),HL *1	ED 63 NN NN	-	-	-	-	-	-	-	-	20
LD (NN),SP	ED 73 NN NN	-	-	-	-	-	-	-	-	20
LD (NN),IX	DD 22 NN NN	-	-	-	-	-	-	-	-	20
LD (NN),IY	FD 22 NN NN	-	-	-	-	-	-	-	-	20

Команды данной группы помещают значение указанного 8-битного регистра или 16-битной регистровой пары по указанному адресу памяти. Адрес следует непосредственно за командой. Влияния на флаги данные команды не оказывают.

- *1 Команда LD (NN),HL может быть записана в двух вариантах – кодом операции 22 или кодом операции ED 63. Запись ED 63 занимает на 1 байт больше и выполняется на 4 такта дольше за счет префикса, поэтому ее использование не рекомендуется.

10. Команды помещения значения 8-битного регистра в память по адресу в 16-битной регистровой паре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD (BC),A	02	-	-	-	-	-	-	-	-	7
LD (DE),A	12	-	-	-	-	-	-	-	-	7
LD (HL),A	77	-	-	-	-	-	-	-	-	7
LD (HL),B	70	-	-	-	-	-	-	-	-	7
LD (HL),C	71	-	-	-	-	-	-	-	-	7
LD (HL),D	72	-	-	-	-	-	-	-	-	7
LD (HL),E	73	-	-	-	-	-	-	-	-	7
LD (HL),H	74	-	-	-	-	-	-	-	-	7
LD (HL),L	75	-	-	-	-	-	-	-	-	7

Команды данной группы помещают значение указанного 8-битного регистра по указанному адресу памяти. Адрес хранится в одной из 16-битных регистровых пар, искомая регистровая пара указывается в команде. Влияния на флаги данные команды не оказывают.

11. Команда помещения непосредственного 8-битного значения в память по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD (HL),N	36 NN	-	-	-	-	-	-	-	-	10

Данная команда помещает в память по адресу в регистровой паре HL 8-битовое значение, следующее непосредственно за командой. Влияния на флаги данная команда не оказывает.

12. Команды помещения значения 8-битного регистра в память по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD (IX+s),A	DD 77 ss	-	-	-	-	-	-	-	-	19
LD (IX+s),B	DD 70 ss	-	-	-	-	-	-	-	-	19
LD (IX+s),C	DD 71 ss	-	-	-	-	-	-	-	-	19
LD (IX+s),D	DD 72 ss	-	-	-	-	-	-	-	-	19
LD (IX+s),E	DD 73 ss	-	-	-	-	-	-	-	-	19
LD (IX+s),H	DD 74 ss	-	-	-	-	-	-	-	-	19
LD (IX+s),L	DD 75 ss	-	-	-	-	-	-	-	-	19
LD (IY+s),A	FD 77 ss	-	-	-	-	-	-	-	-	19
LD (IY+s),B	FD 70 ss	-	-	-	-	-	-	-	-	19
LD (IY+s),C	FD 71 ss	-	-	-	-	-	-	-	-	19
LD (IY+s),D	FD 72 ss	-	-	-	-	-	-	-	-	19
LD (IY+s),E	FD 73 ss	-	-	-	-	-	-	-	-	19
LD (IY+s),H	FD 74 ss	-	-	-	-	-	-	-	-	19
LD (IY+s),L	FD 75 ss	-	-	-	-	-	-	-	-	19

Команды данной группы помещают значение указанного 8-битного регистра по указанному адресу памяти. Адрес хранится в одной из 16-битных индексных регистровых пар

IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется. Влияния на флаги данные команды не оказывают.

13. Команды помещения непосредственного 8-битного значения в память по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LD (IX+s),N	DD 36 ss NN	-	-	-	-	-	-	-	-	22
LD (IY+s),N	FD 36 ss NN	-	-	-	-	-	-	-	-	22

Команды данной группы помещают 8-битовое значение, следующее непосредственно за командой, по указанному адресу памяти. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется. Влияния на флаги данные команды не оказывают.

14. Команды обмена значений 16-битных регистровых пар

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
EXX	D9	-	-	-	-	-	-	-	-	4
EX AF,AF'	08	*	*	*	*	*	*	*	*	4
EX DE,HL	EB	-	-	-	-	-	-	-	-	4

Команда EXX выполняет обмен значений основных регистров BC, DE и HL со значениями их дополнительных «теневых» пар – BC', DE' и HL'. Команда EXX не воздействует на флаги.

Команда EX AF,AF' выполняет обмен значения пары аккумулятор-регистр флагов AF со значением ее дополнительной «теневой» копии AF'. Состояния флагов после выполнения команды не определены и зависят от состояния «теневой» копии регистра флагов F.

Команда EX DE,HL выполняет обмен значений регистров DE и HL. Данная команда не воздействует на флаги.

15. Команды обмена значений 16-битных регистровых пар и памяти

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
EX (SP),HL	E3	-	-	-	-	-	-	-	-	19
EX (SP),IX	DD E3	-	-	-	-	-	-	-	-	23
EX (SP),IY	FD E3	-	-	-	-	-	-	-	-	23

Команды данной группы выполняют обмен значения регистровой пары HL, IX или IY (искомая регистровая пара указывается в команде) с 16-битным значением, находящимся в памяти по адресу, определяемому указателем стека SP. Влияния на флаги данные команды не оказывают.

16. Команды сложения значения аккумулятора со значением 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
ADD A,A	87	+	+	5	+	3	V	0	+	4
ADD A,B	80	+	+	5	+	3	V	0	+	4
ADD A,C	81	+	+	5	+	3	V	0	+	4
ADD A,D	82	+	+	5	+	3	V	0	+	4
ADD A,E	83	+	+	5	+	3	V	0	+	4
ADD A,H	84	+	+	5	+	3	V	0	+	4
ADD A,L	85	+	+	5	+	3	V	0	+	4
ADD A,XH	DD 84	+	+	5	+	3	V	0	+	8

ADD A,XL	DD 85	+	+	5	+	3	V	0	+	8
ADD A,YH	FD 84	+	+	5	+	3	V	0	+	8
ADD A,YL	FD 85	+	+	5	+	3	V	0	+	8

Команды данной группы складывают значение аккумулятора со значением в указанном 8-битном регистре. Результат сложения помещается в аккумулятор.

17. Команда сложения значения аккумулятора с непосредственным 8-битным значением

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
ADD A,N	C6 NN	+	+	5	+	3	V	0	+	7

Данная команда складывает значение аккумулятора с 8-битным значением, следующим непосредственно за командой. Результат сложения помещается в аккумулятор.

18. Команда сложения значения аккумулятора с 8-битным значением в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
ADD A,(HL)	86	+	+	5	+	3	V	0	+	7

Данная команда складывает значение аккумулятора с 8-битным значением в памяти по адресу в регистровой паре HL. Результат сложения помещается в аккумулятор.

19. Команды сложения значения аккумулятора с 8-битным значением в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
ADD A,(IX+s)	DD 86 ss	+	+	5	+	3	V	0	+	19
ADD A,(IY+s)	FD 86 ss	+	+	5	+	3	V	0	+	19

Команды данной группы складывают значение аккумулятора с 8-битным значением в памяти по указанному адресу. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержащееся самой регистровой пары не изменяется. Результат сложения помещается в аккумулятор.

20. Команды сложения значений 16-битных регистровых пар

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
ADD HL,BC	09	-	-	*	*	*	-	0	+	11
ADD HL,DE	19	-	-	*	*	*	-	0	+	11
ADD HL,HL	29	-	-	*	*	*	-	0	+	11
ADD HL,SP	39	-	-	*	*	*	-	0	+	11
ADD IX,BC	DD 09	-	-	*	*	*	-	0	+	15
ADD IX,DE	DD 19	-	-	*	*	*	-	0	+	15
ADD IX,IX	DD 29	-	-	*	*	*	-	0	+	15
ADD IX,SP	DD 39	-	-	*	*	*	-	0	+	15
ADD IY,BC	FD 09	-	-	*	*	*	-	0	+	15
ADD IY,DE	FD 19	-	-	*	*	*	-	0	+	15
ADD IY,IY	FD 29	-	-	*	*	*	-	0	+	15
ADD IY,SP	FD 39	-	-	*	*	*	-	0	+	15

Команды данной группы складывают значение регистровой пары HL, IX или IY со значением 16-битной регистровой пары общего назначения. Результат сложения помещается в

первую указанную регистровую пару. Искомые регистровые пары указываются в команде. В результате выполнения данных команд флаг C выставляется согласно результату операции.

Флаги 5 и 3 выставляются в соответствии с битами 5 и 3 старшего байта результата. Флаг H выставляется аналогично команде сложения с учетом переноса для старшего байта результата.

21. Команды инкремента значения 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
INC A	3C	+	+	5	+	3	V	0	-	4
INC B	04	+	+	5	+	3	V	0	-	4
INC C	0C	+	+	5	+	3	V	0	-	4
INC D	14	+	+	5	+	3	V	0	-	4
INC E	1C	+	+	5	+	3	V	0	-	4
INC H	24	+	+	5	+	3	V	0	-	4
INC L	2C	+	+	5	+	3	V	0	-	4
INC XH	DD 24	+	+	5	+	3	V	0	-	8
INC XL	DD 2C	+	+	5	+	3	V	0	-	8
INC YH	FD 24	+	+	5	+	3	V	0	-	8
INC YL	FD 2C	+	+	5	+	3	V	0	-	8

Команды данной группы увеличивают значение указанного 8-битного регистра на единицу. Данные команды не влияют на флаг переноса.

22. Команды инкремента 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
INC (HL)	34	+	+	5	+	3	V	0	-	11

Данная команда увеличивает 8-битное значение в памяти по адресу в регистре HL на единицу. Команда не влияет на флаг переноса.

23. Команды инкремента 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
INC (IX+s)	DD 34 ss	+	+	5	+	3	V	0	-	23
INC (IY+s)	FD 34 ss	+	+	5	+	3	V	0	-	23

Команды данной группы увеличивают 8-битное значение в памяти по указанному адресу на единицу. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержащее самой регистровой пары не изменяется. Команды не влияют на флаг переноса.

24. Команды инкремента значения 16-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
INC BC	03	-	-	-	-	-	-	-	-	6
INC DE	13	-	-	-	-	-	-	-	-	6
INC HL	23	-	-	-	-	-	-	-	-	6
INC SP	33	-	-	-	-	-	-	-	-	6
INC IX	DD 23	-	-	-	-	-	-	-	-	10
INC IY	FD 23	-	-	-	-	-	-	-	-	10

Команды данной группы увеличивают значение указанного 16-битного регистра на единицу. Влияния на флаги данные команды не оказывают.

25. Команды сложения значения аккумулятора со значением 8-битного регистра с учетом флага переноса

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
ADC A,A	8F	+	+	5	+	3	V	0	+	4
ADC A,B	88	+	+	5	+	3	V	0	+	4
ADC A,C	89	+	+	5	+	3	V	0	+	4
ADC A,D	8A	+	+	5	+	3	V	0	+	4
ADC A,E	8B	+	+	5	+	3	V	0	+	4
ADC A,H	8C	+	+	5	+	3	V	0	+	4
ADC A,L	8D	+	+	5	+	3	V	0	+	4
ADC A,XH	DD 8C	+	+	5	+	3	V	0	+	8
ADC A,XL	DD 8D	+	+	5	+	3	V	0	+	8
ADC A,YH	FD 8C	+	+	5	+	3	V	0	+	8
ADC A,YL	FD 8D	+	+	5	+	3	V	0	+	8

Команды данной группы складывают значение аккумулятора со значением в указанном 8-битном регистре. Если флаг переноса был установлен, то дополнительно прибавляется единица. Результат сложения помещается в аккумулятор.

26. Команда сложения значения аккумулятора с непосредственным 8-битным значением с учетом флага переноса

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
ADC A,N	CE NN	+	+	5	+	3	V	0	+	7

Данная команда складывает значение аккумулятора с 8-битным значением, следующим непосредственно за командой. Если флаг переноса был установлен, то дополнительно прибавляется единица. Результат сложения помещается в аккумулятор.

27. Команда сложения значения аккумулятора с 8-битным значением в памяти по адресу в регистре HL с учетом флага переноса

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
ADC A,(HL)	8E	+	+	5	+	3	V	0	+	7

Данная команда складывает значение аккумулятора с 8-битным значением в памяти по адресу в регистровой паре HL. Если флаг переноса был установлен, то дополнительно прибавляется единица. Результат сложения помещается в аккумулятор.

28. Команды сложения значения аккумулятора с 8-битным значением в памяти по адресу в индексном регистре (со смещением) с учетом флага переноса

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
ADC A,(IX+s)	DD 8E ss	+	+	5	+	3	V	0	+	19
ADC A,(IY+s)	FD 8E ss	+	+	5	+	3	V	0	+	19

Команды данной группы складывают значение аккумулятора с 8-битным значением в памяти по указанному адресу. Если флаг переноса был установлен, то дополнительно прибавляется единица. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержащее самой регистровой пары не изменяется. Результат сложения помещается в аккумулятор.

29. Команды сложения значений 16-битных регистровых пар с учетом флага переноса

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
ADC HL,BC	ED 4A	+	+	*	*	*	V	0	+	15
ADC HL,DE	ED 5A	+	+	*	*	*	V	0	+	15
ADC HL,HL	ED 6A	+	+	*	*	*	V	0	+	15
ADC HL,SP	ED 7A	+	+	*	*	*	V	0	+	15

Команды данной группы складывают значение регистровой пары HL со значением 16-битной регистровой пары общего назначения. Если флаг переноса был установлен, то дополнительно прибавляется единица. Результат сложения помещается в регистровую пару HL.

Флаги 5 и 3 выставляются в соответствии с битами 5 и 3 старшего байта результата. Флаг H выставляется аналогично команде сложения с учетом переноса для старшего байта результата.

30. Команды вычитания значения 8-битного регистра из значения аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SUB A	97	+	+	5	+	3	V	1	+	4
SUB B	90	+	+	5	+	3	V	1	+	4
SUB C	91	+	+	5	+	3	V	1	+	4
SUB D	92	+	+	5	+	3	V	1	+	4
SUB E	93	+	+	5	+	3	V	1	+	4
SUB H	94	+	+	5	+	3	V	1	+	4
SUB L	95	+	+	5	+	3	V	1	+	4
SUB XH	DD 94	+	+	5	+	3	V	1	+	8
SUB XL	DD 95	+	+	5	+	3	V	1	+	8
SUB YH	FD 94	+	+	5	+	3	V	1	+	8
SUB YL	FD 95	+	+	5	+	3	V	1	+	8

Команды данной группы вычитают значение в указанном 8-битном регистре из значения аккумулятора. Результат вычитания помещается в аккумулятор.

31. Команда вычитания непосредственного 8-битного значения из значения аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SUB N	D6 NN	+	+	5	+	3	V	1	+	7

Данная команда вычитает 8-битное значение, следующее непосредственно за командой, из значения аккумулятора. Результат вычитания помещается в аккумулятор.

32. Команда вычитания 8-битного значения в памяти по адресу в регистре HL из значения аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SUB (HL)	96	+	+	5	+	3	V	1	+	7

Данная команда вычитает 8-битное значение в памяти по адресу в регистровой паре HL из значения аккумулятора. Результат вычитания помещается в аккумулятор.

33. Команды вычитания 8-битного значения в памяти по адресу в индексном регистре (со смещением) из значения аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
-----------	--------------	---	---	---	---	---	-----	---	---	-------

SUB (IX+s)	DD 96 ss	+	+	5	+	3	V	1	+	19
SUB (IY+s)	FD 96 ss	+	+	5	+	3	V	1	+	19

Команды данной группы вычитают 8-битное значением в памяти по указанному адресу из значения аккумулятора. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется. Результат вычитания помещается в аккумулятор.

34. Команды декремента значения 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
DEC A	3D	+	+	5	+	3	V	1	-	4
DEC B	05	+	+	5	+	3	V	1	-	4
DEC C	0D	+	+	5	+	3	V	1	-	4
DEC D	15	+	+	5	+	3	V	1	-	4
DEC E	1D	+	+	5	+	3	V	1	-	4
DEC H	25	+	+	5	+	3	V	1	-	4
DEC L	2D	+	+	5	+	3	V	1	-	4
DEC XH	DD 25	+	+	5	+	3	V	1	-	8
DEC XL	DD 2D	+	+	5	+	3	V	1	-	8
DEC YH	FD 25	+	+	5	+	3	V	1	-	8
DEC YL	FD 2D	+	+	5	+	3	V	1	-	8

Команды данной группы уменьшают значение указанного 8-битного регистра на единицу. Данные команды не влияют на флаг переноса.

35. Команды декремента 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
DEC (HL)	35	+	+	5	+	3	V	1	-	11

Данная команда уменьшает 8-битное значение в памяти по адресу в регистре HL на единицу. Команда не влияет на флаг переноса.

36. Команды декремента 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
DEC (IX+s)	DD 35 ss	+	+	5	+	3	V	1	-	23
DEC (IY+s)	FD 35 ss	+	+	5	+	3	V	1	-	23

Команды данной группы уменьшают 8-битное значение в памяти по указанному адресу на единицу. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется. Команды не влияют на флаг переноса.

37. Команды декремента значения 16-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
DEC BC	0B	-	-	-	-	-	-	-	-	6
DEC DE	1B	-	-	-	-	-	-	-	-	6
DEC HL	2B	-	-	-	-	-	-	-	-	6
DEC SP	3B	-	-	-	-	-	-	-	-	6
DEC IX	DD 2B	-	-	-	-	-	-	-	-	10

DEC IY FD 2B - - - - - - - - - 10

Команды данной группы уменьшают значение указанного 16-битного регистра на единицу. Влияния на флаги данные команды не оказывают.

38. Команды вычитания значения 8-битного регистра из значения аккумулятора с учетом флага переноса

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SBC A,A	9F	+	+	5	+	3	V	1	+	4
SBC A,B	98	+	+	5	+	3	V	1	+	4
SBC A,C	99	+	+	5	+	3	V	1	+	4
SBC A,D	9A	+	+	5	+	3	V	1	+	4
SBC A,E	9B	+	+	5	+	3	V	1	+	4
SBC A,H	9C	+	+	5	+	3	V	1	+	4
SBC A,L	9D	+	+	5	+	3	V	1	+	4
SBC A,XH	DD 9C	+	+	5	+	3	V	1	+	8
SBC A,XL	DD 9D	+	+	5	+	3	V	1	+	8
SBC A,YH	FD 9C	+	+	5	+	3	V	1	+	8
SBC A,YL	FD 9D	+	+	5	+	3	V	1	+	8

Команды данной группы вычитают значение в указанном 8-битном регистре из значения аккумулятора. Если флаг переноса был установлен, то дополнительно вычитается единица. Результат вычитания помещается в аккумулятор.

39. Команда вычитания непосредственного 8-битного значения из значения аккумулятора с учетом флага переноса

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SBC A,N	DE NN	+	+	5	+	3	V	1	+	7

Данная команда вычитает 8-битное значение, следующее непосредственно за командой, из значения аккумулятора. Если флаг переноса был установлен, то дополнительно вычитается единица. Результат вычитания помещается в аккумулятор.

40. Команда вычитания 8-битного значения в памяти по адресу в регистре HL из значения аккумулятора с учетом флага переноса

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SBC A,(HL)	9E	+	+	5	+	3	V	1	+	7

Данная команда вычитает 8-битное значение в памяти по адресу в регистровой паре HL из значения аккумулятора. Если флаг переноса был установлен, то дополнительно вычитается единица. Результат вычитания помещается в аккумулятор.

41. Команды вычитания 8-битного значения в памяти по адресу в индексном регистре (со смещением) из значения аккумулятора с учетом флага переноса

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SBC A,(IX+s)	DD 9E ss	+	+	5	+	3	V	1	+	19
SBC A,(IY+s)	FD 9E ss	+	+	5	+	3	V	1	+	19

Команды данной группы вычитают 8-битное значение в памяти по указанному адресу из значения аккумулятора. Если флаг переноса был установлен, то дополнительно вычитается единица. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется

относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется. Результат вычитания помещается в аккумулятор.

42. Команды вычитания значений 16-битных регистровых пар с учетом флага переноса

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SBC HL,BC	ED 42	+	+	*	*	*	V	1	+	15
SBC HL,DE	ED 52	+	+	*	*	*	V	1	+	15
SBC HL,HL	ED 62	+	+	*	*	*	V	1	+	15
SBC HL,SP	ED 72	+	+	*	*	*	V	1	+	15

Команды данной группы вычитают значение 16-битной регистровой пары общего назначения из значения регистровой пары HL. Если флаг переноса был установлен, то дополнительно вычитается единица. Результат вычитания помещается в регистровую пару HL.

Флаги 5 и 3 выставляются в соответствии с битами 5 и 3 старшего байта результата. Флаг H выставляется аналогично команде вычитания с учетом переноса для старшего байта результата.

43. Команды сравнения значения 8-битного регистра со значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
CP A	BF	+	+	*1	+	*1	V	1	+	4
CP B	B8	+	+	5	+	3	V	1	+	4
CP C	B9	+	+	5	+	3	V	1	+	4
CP D	BA	+	+	5	+	3	V	1	+	4
CP E	BB	+	+	5	+	3	V	1	+	4
CP H	BC	+	+	5	+	3	V	1	+	4
CP L	BD	+	+	5	+	3	V	1	+	4
CP XH	DD BC	+	+	5	+	3	V	1	+	8
CP XL	DD BD	+	+	5	+	3	V	1	+	8
CP YH	FD BC	+	+	5	+	3	V	1	+	8
CP YL	FD BD	+	+	5	+	3	V	1	+	8

Команды данной группы вычитают значение в указанном 8-битном регистре из значения аккумулятора и устанавливают флаги согласно результату. Сам результат не сохраняется.

*1 Флаги 5 и 3 в данной команде копируются из операнда, т.е. из регистра A, а не из результата выполнения операции.

44. Команда сравнения непосредственного 8-битного значения со значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
CP N	FE NN	+	+	5	+	3	V	1	+	7

Данная команда вычитает 8-битное значение, следующее непосредственно за командой, из значения аккумулятора и устанавливает флаги согласно результату. Сам результат не сохраняется.

45. Команда сравнения 8-битного значения в памяти по адресу в регистре HL со значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
CP (HL)	BE	+	+	5	+	3	V	1	+	7

Данная команда вычитает 8-битное значение в памяти по адресу в регистровой паре HL из значения аккумулятора и устанавливает флаги согласно результату. Сам результат не сохраняется.

46. Команды сравнения 8-битного значения в памяти по адресу в индексном регистре (со смещением) со значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
CP (IX+s)	DD BE ss	+	+	5	+	3	V	1	+	19
CP (IY+s)	FD BE ss	+	+	5	+	3	V	1	+	19

Команды данной группы вычитают 8-битное значение в памяти по указанному адресу из значения аккумулятора и устанавливают флаги согласно результату. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется. Результат выполнения операции не сохраняется.

47. Команды логического «И» над значением 8-битного регистра и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
AND A	A7	+	+	5	1	3	P	0	0	4
AND B	A0	+	+	5	1	3	P	0	0	4
AND C	A1	+	+	5	1	3	P	0	0	4
AND D	A2	+	+	5	1	3	P	0	0	4
AND E	A3	+	+	5	1	3	P	0	0	4
AND H	A4	+	+	5	1	3	P	0	0	4
AND L	A5	+	+	5	1	3	P	0	0	4
AND XH	DD A4	+	+	5	1	3	P	0	0	8
AND XL	DD A5	+	+	5	1	3	P	0	0	8
AND YH	FD A4	+	+	5	1	3	P	0	0	8
AND YL	FD A5	+	+	5	1	3	P	0	0	8

Команды данной группы производят операцию логического «И» над значением в указанном 8-битном регистре и значением аккумулятора. Результат операции сохраняется в аккумуляторе.

48. Команда логического «И» над непосредственным 8-битным значением и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
AND N	E6 NN	+	+	5	1	3	P	0	0	7

Данная команда производит операцию логического «И» над 8-битным значением, следующим непосредственно за командой, и значением аккумулятора. Результат операции сохраняется в аккумуляторе.

49. Команда логического «И» над 8-битным значением в памяти по адресу в регистре HL и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
AND (HL)	A6	+	+	5	1	3	P	0	0	7

Данная команда производит операцию логического «И» над 8-битным значением в памяти по адресу в регистровой паре HL и значением аккумулятора. Результат операции сохраняется в аккумуляторе.

50. Команды логического «И» над 8-битным значением в памяти по адресу в индексном регистре (со смещением) и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
AND (IX+s)	DD A6 ss	+	+	5	1	3	P	0	0	19
AND (IY+s)	FD A6 ss	+	+	5	1	3	P	0	0	19

Команды данной группы производят операцию логического «И» над 8-битным значением в памяти по указанному адресу и значением аккумулятора. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется. Результат операции сохраняется в аккумуляторе.

51. Команды логического «ИЛИ» над значением 8-битного регистра и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
OR A	B7	+	+	5	0	3	P	0	0	4
OR B	B0	+	+	5	0	3	P	0	0	4
OR C	B1	+	+	5	0	3	P	0	0	4
OR D	B2	+	+	5	0	3	P	0	0	4
OR E	B3	+	+	5	0	3	P	0	0	4
OR H	B4	+	+	5	0	3	P	0	0	4
OR L	B5	+	+	5	0	3	P	0	0	4
OR XH	DD B4	+	+	5	0	3	P	0	0	8
OR XL	DD B5	+	+	5	0	3	P	0	0	8
OR YH	FD B4	+	+	5	0	3	P	0	0	8
OR YL	FD B5	+	+	5	0	3	P	0	0	8

Команды данной группы производят операцию логического «ИЛИ» над значением в указанном 8-битном регистре и значением аккумулятора. Результат операции сохраняется в аккумуляторе.

52. Команда логического «ИЛИ» над непосредственным 8-битным значением и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
OR N	F6 NN	+	+	5	0	3	P	0	0	7

Данная команда производит операцию логического «ИЛИ» над 8-битным значением, следующим непосредственно за командой, и значением аккумулятора. Результат операции сохраняется в аккумуляторе.

53. Команда логического «ИЛИ» над 8-битным значением в памяти по адресу в регистре HL и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
OR (HL)	B6	+	+	5	0	3	P	0	0	7

Данная команда производит операцию логического «ИЛИ» над 8-битным значением в памяти по адресу в регистровой паре HL и значением аккумулятора. Результат операции сохраняется в аккумуляторе.

54. Команды логического «ИЛИ» над 8-битным значением в памяти по адресу в индексном регистре (со смещением) и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
OR (IX+s)	DD B6 ss	+	+	5	0	3	P	0	0	19
OR (IY+s)	FD B6 ss	+	+	5	0	3	P	0	0	19

Команды данной группы производят операцию логического «ИЛИ» над 8-битным значением в памяти по указанному адресу и значением аккумулятора. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержащее самой регистровой пары не изменяется. Результат операции сохраняется в аккумуляторе.

55. Команды логического «исключающего ИЛИ» над значением 8-битного регистра и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
XOR A	AF	+	+	5	0	3	P	0	0	4
XOR B	A8	+	+	5	0	3	P	0	0	4
XOR C	A9	+	+	5	0	3	P	0	0	4
XOR D	AA	+	+	5	0	3	P	0	0	4
XOR E	AB	+	+	5	0	3	P	0	0	4
XOR H	AC	+	+	5	0	3	P	0	0	4
XOR L	AD	+	+	5	0	3	P	0	0	4
XOR XH	DD AC	+	+	5	0	3	P	0	0	8
XOR XL	DD AD	+	+	5	0	3	P	0	0	8
XOR YH	FD AC	+	+	5	0	3	P	0	0	8
XOR YL	FD AD	+	+	5	0	3	P	0	0	8

Команды данной группы производят операцию логического «исключающего ИЛИ» над значением в указанном 8-битном регистре и значением аккумулятора. Результат операции сохраняется в аккумуляторе.

56. Команда логического «исключающего ИЛИ» над непосредственным 8-битным значением и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
XOR N	EE NN	+	+	5	0	3	P	0	0	7

Данная команда производит операцию логического «исключающего ИЛИ» над 8-битным значением, следующим непосредственно за командой, и значением аккумулятора. Результат операции сохраняется в аккумуляторе.

57. Команда логического «исключающего ИЛИ» над 8-битным значением в памяти по адресу в регистре HL и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
XOR (HL)	AE	+	+	5	0	3	P	0	0	7

Данная команда производит операцию логического «исключающего ИЛИ» над 8-битным значением в памяти по адресу в регистровой паре HL и значением аккумулятора. Результат операции сохраняется в аккумуляторе.

58. Команды логического «исключающего ИЛИ» над 8-битным значением в памяти по адресу в индексном регистре (со смещением) и значением аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
XOR (IX+s)	DD AE ss	+	+	5	0	3	P	0	0	19
XOR (IY+s)	FD AE ss	+	+	5	0	3	P	0	0	19

Команды данной группы производят операцию логического «исключающего ИЛИ» над 8-битным значением в памяти по указанному адресу и значением аккумулятора. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержащее самой регистровой пары не изменяется. Результат операции сохраняется в аккумуляторе.

59. Команда безусловного перехода по непосредственному адресу

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
JP NN	C3 NN NN	-	-	-	-	-	-	-	-	10

Данная команда выполняет безусловный переход по непосредственному 16-битному адресу, следующему прямо за командой.

60. Команды безусловного перехода по адресу в 16-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
JP (HL)	E9	-	-	-	-	-	-	-	-	10
JP (IX)	DD E9	-	-	-	-	-	-	-	-	10
JP (IY)	FD E9	-	-	-	-	-	-	-	-	10

Команды данной группы выполняют безусловный переход по 16-битному адресу, находящемуся в указанном регистре.

61. Команды условного перехода по непосредственному адресу

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
JP NZ,NN	C2 NN NN	-	-	-	-	-	-	-	-	10
JP Z,NN	CA NN NN	-	-	-	-	-	-	-	-	10
JP NC,NN	D2 NN NN	-	-	-	-	-	-	-	-	10
JP C,NN	DA NN NN	-	-	-	-	-	-	-	-	10
JP PO,NN	E2 NN NN	-	-	-	-	-	-	-	-	10
JP PE,NN	EA NN NN	-	-	-	-	-	-	-	-	10
JP P,NN	F2 NN NN	-	-	-	-	-	-	-	-	10
JP M,NN	FA NN NN	-	-	-	-	-	-	-	-	10

Команды данной группы выполняют условный переход по непосредственному 16-битному адресу, следующему прямо за командой, согласно заданному условию. Если условие не выполняется, исполнение продолжается со следующей команды.

62. Команда безусловного относительного перехода

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
JR s	18 ss	-	-	-	-	-	-	-	-	12

Данная команда выполняет безусловный переход по непосредственному 8-битному смещению, следующему прямо за командой. Смещение рассматривается как число со знаком и прибавляется к текущему значению регистра PC, которое указывает на следующий за командой JR адрес (адрес команды JR + 2).

63. Команды условного относительного перехода

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
JR NZ,s	20 ss	-	-	-	-	-	-	-	-	7/12
JR Z,s	28 ss	-	-	-	-	-	-	-	-	7/12
JR NC,s	30 ss	-	-	-	-	-	-	-	-	7/12
JR C,s	38 ss	-	-	-	-	-	-	-	-	7/12

Команды данной группы выполняют условный переход по непосредственному 8-битному смещению, следующему прямо за командой, согласно заданному условию. Смещение рассматривается как число со знаком и прибавляется к текущему значению регистра PC, которое указывает на следующий за командой JR адрес (адрес команды JR + 2). Если условие не выполняется, исполнение продолжается со следующей команды, при этом команда JR исполняется всего 7 тактов, а не 12.

64. Команда условного относительного перехода с организацией цикла по регистру B

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
DJNZ s	10 ss	-	-	-	-	-	-	-	-	8/13

Данная команда уменьшает значение регистра B на единицу и выполняет переход по непосредственному 8-битному смещению, следующему прямо за командой, если значение регистра B после уменьшения не равно 0. Смещение рассматривается как число со знаком и прибавляется к текущему значению регистра PC, которое указывает на следующий за командой JR адрес (адрес команды JR + 2). Если значение регистра B равно 0, исполнение продолжается со следующей команды, при этом команда DJNZ исполняется 8 тактов, а не 13.

65. Команды помещения значения 16-битной регистровой пары в стек

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
PUSH AF	F5	-	-	-	-	-	-	-	-	11
PUSH BC	C5	-	-	-	-	-	-	-	-	11
PUSH DE	D5	-	-	-	-	-	-	-	-	11
PUSH HL	E5	-	-	-	-	-	-	-	-	11
PUSH IX	DD E5	-	-	-	-	-	-	-	-	15
PUSH IY	FD E5	-	-	-	-	-	-	-	-	15

Команды данной группы уменьшают значение регистра стека SP на 2, а затем помещают значение указанной 16-битной регистровой пары в память по адресу в регистре SP.

66. Команды снятия значения 16-битной регистровой пары со стека

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
POP AF	F1	*	*	*	*	*	*	*	*	10
POP BC	C1	-	-	-	-	-	-	-	-	10
POP DE	D1	-	-	-	-	-	-	-	-	10
POP HL	E1	-	-	-	-	-	-	-	-	10
POP IX	DD E1	-	-	-	-	-	-	-	-	14
POP IY	FD E1	-	-	-	-	-	-	-	-	14

Команды данной группы загружают указанную 16-битную регистровую пару из памяти по адресу в регистре SP, а затем увеличивают регистр стека SP на 2. Если загружается регистровая пара AF, то значения флагов определяются содержимым первого байта в памяти по адресу в регистре SP.

67. Команды системного вызова

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RST #0	C7	-	-	-	-	-	-	-	-	11
RST #8	CF	-	-	-	-	-	-	-	-	11
RST #10	D7	-	-	-	-	-	-	-	-	11
RST #18	DF	-	-	-	-	-	-	-	-	11
RST #20	E7	-	-	-	-	-	-	-	-	11
RST #28	EF	-	-	-	-	-	-	-	-	11
RST #30	F7	-	-	-	-	-	-	-	-	11
RST #38	FF	-	-	-	-	-	-	-	-	11

Команды данной группы помещают в стек адрес возврата (уменьшают значение регистра стека SP на 2, а затем помещают в память по адресу в регистре SP следующий за командой RST адрес – адрес команды RST + 1), после чего выполняют безусловный переход по указанному в команде фиксированному адресу памяти, который может быть записан в любом (шестнадцатеричном, десятичном и т.д.) приемлемом виде. В таблице приведены мнемоники с адресом в шестнадцатеричном виде.

68. Команда безусловного вызова по непосредственному адресу

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
CALL NN	CD NN NN	-	-	-	-	-	-	-	-	17

Данная команда помещает в стек адрес возврата (уменьшает значение регистра стека SP на 2, а затем помещает в память по адресу в регистре SP следующий за командой CALL адрес – адрес команды CALL + 3), после чего выполняет безусловный переход по непосредственному 16-битному адресу, следующему прямо за командой. Команды вызова CALL обычно используются в паре с командами возврата RET.

69. Команды условного вызова по непосредственному адресу

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
CALL NZ,NN	C4 NN NN	-	-	-	-	-	-	-	-	10/17
CALL Z,NN	CC NN NN	-	-	-	-	-	-	-	-	10/17
CALL NC,NN	D4 NN NN	-	-	-	-	-	-	-	-	10/17
CALL C,NN	DC NN NN	-	-	-	-	-	-	-	-	10/17
CALL PO,NN	E4 NN NN	-	-	-	-	-	-	-	-	10/17
CALL PE,NN	EC NN NN	-	-	-	-	-	-	-	-	10/17
CALL P,NN	F4 NN NN	-	-	-	-	-	-	-	-	10/17
CALL M,NN	FC NN NN	-	-	-	-	-	-	-	-	10/17

Команды данной группы помещают в стек адрес возврата (уменьшают значение регистра стека SP на 2, а затем помещают в память по адресу в регистре SP следующий за командой CALL адрес – адрес команды CALL + 3), после чего выполняют переход по непосредственному 16-битному адресу, следующему прямо за командой, согласно заданному условию. Если условие не выполняется, адрес возврата в стек не помещается, а исполнение продолжается со следующей команды, при этом команда CALL исполняется 10 тактов, а не 17. Команды вызова CALL обычно используются в паре с командами возврата RET.

70. Команда безусловного возврата

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RET	C9	-	-	-	-	-	-	-	-	10

Данная команда снимает со стека 16-битный адрес возврата (снимает значение из памяти по адресу в регистре SP, а затем увеличивает значение регистра стека SP на 2), после чего выполняет безусловный переход по данному адресу. Команды возврата RET обычно используются в паре с командами вызова CALL.

71. Команды условного возврата

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RET NZ	C0	-	-	-	-	-	-	-	-	5/11
RET Z	C8	-	-	-	-	-	-	-	-	5/11
RET NC	D0	-	-	-	-	-	-	-	-	5/11
RET C	D8	-	-	-	-	-	-	-	-	5/11
RET PO	E0	-	-	-	-	-	-	-	-	5/11
RET PE	E8	-	-	-	-	-	-	-	-	5/11
RET P	F0	-	-	-	-	-	-	-	-	5/11
RET M	F8	-	-	-	-	-	-	-	-	5/11

Команда данной группы снимают со стека 16-битный адрес возврата (снимают значение из памяти по адресу в регистре SP, а затем увеличивают значение регистра стека SP на 2), после чего выполняют переход по данному адресу, согласно заданному условию. Если условие не выполняется, адрес возврата со стека не снимается, а исполнение продолжается со следующей команды, при этом команда RET исполняется не 11 тактов, а 5. Команды возврата RET обычно используются в паре с командами вызова CALL.

72. Команды логического сдвига вправо значения 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SRL A	CB 3F	+	+	5	0	3	P	0	+	8
SRL B	CB 38	+	+	5	0	3	P	0	+	8
SRL C	CB 39	+	+	5	0	3	P	0	+	8
SRL D	CB 3A	+	+	5	0	3	P	0	+	8
SRL E	CB 3B	+	+	5	0	3	P	0	+	8
SRL H	CB 3C	+	+	5	0	3	P	0	+	8
SRL L	CB 3D	+	+	5	0	3	P	0	+	8

Команды данной группы производят операцию логического сдвига вправо над значением указанного 8-битного регистра. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется нулем, «вытесненный» бит 0 (младший) переходит во флаг переноса C.

73. Команда логического сдвига вправо 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SRL (HL)	CB 3E	+	+	5	0	3	P	0	+	15

Команды данной группы производят операцию логического сдвига вправо над 8-битным значением в памяти по адресу в регистровой паре HL. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется нулем, «вытесненный» бит 0 (младший) переходит во флаг переноса C.

74. Команды логического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SRL (IX+s)	DD CB ss 3E	+	+	5	0	3	P	0	+	23
SRL (IY+s)	FD CB ss 3E	+	+	5	0	3	P	0	+	23

Команды данной группы производят операцию логического сдвига вправо над 8-битным значением в памяти по указанному адресу. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется нулем, «вытесненный» бит 0 (младший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

75. Сложные команды логического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SRL A,(IX+s)	DD CB ss 3F	+	+	5	0	3	P	0	+	23
SRL B,(IX+s)	DD CB ss 38	+	+	5	0	3	P	0	+	23
SRL C,(IX+s)	DD CB ss 39	+	+	5	0	3	P	0	+	23
SRL D,(IX+s)	DD CB ss 3A	+	+	5	0	3	P	0	+	23
SRL E,(IX+s)	DD CB ss 3B	+	+	5	0	3	P	0	+	23
SRL H,(IX+s)	DD CB ss 3C	+	+	5	0	3	P	0	+	23
SRL L,(IX+s)	DD CB ss 3D	+	+	5	0	3	P	0	+	23
SRL A,(IY+s)	FD CB ss 3F	+	+	5	0	3	P	0	+	23
SRL B,(IY+s)	FD CB ss 38	+	+	5	0	3	P	0	+	23
SRL C,(IY+s)	FD CB ss 39	+	+	5	0	3	P	0	+	23
SRL D,(IY+s)	FD CB ss 3A	+	+	5	0	3	P	0	+	23
SRL E,(IY+s)	FD CB ss 3B	+	+	5	0	3	P	0	+	23
SRL H,(IY+s)	FD CB ss 3C	+	+	5	0	3	P	0	+	23
SRL L,(IY+s)	FD CB ss 3D	+	+	5	0	3	P	0	+	23

Команды данной группы представляют из себя сложные команды сдвига. Они загружают в указанный 8-битный регистр значение из памяти по указанному адресу, производят операцию логического сдвига вправо над указанным 8-битным регистром, а затем сохраняют значение указанного 8-битного регистра в память по указанному адресу. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется нулем, «вытесненный» бит 0 (младший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

76. Команды арифметического сдвига вправо значения 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SRA A	CB 2F	+	+	5	0	3	P	0	+	8
SRA B	CB 28	+	+	5	0	3	P	0	+	8
SRA C	CB 29	+	+	5	0	3	P	0	+	8
SRA D	CB 2A	+	+	5	0	3	P	0	+	8
SRA E	CB 2B	+	+	5	0	3	P	0	+	8
SRA H	CB 2C	+	+	5	0	3	P	0	+	8
SRA L	CB 2D	+	+	5	0	3	P	0	+	8

Команды данной группы производят операцию арифметического сдвига вправо над значением указанного 8-битного регистра. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Исходный» бит 7 не изменяется, «вытесненный» бит 0 (младший) переходит во флаг переноса C.

77. Команда арифметического сдвига вправо 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SRA (HL)	CB 2E	+	+	5	0	3	P	0	+	15

Команды данной группы производят операцию арифметического сдвига вправо над 8-битным значением в памяти по адресу в регистровой паре HL. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Исходный» бит 7 не изменяется, «вытесненный» бит 0 (младший) переходит во флаг переноса C.

78. Команды арифметического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SRA (IX+s)	DD CB ss 2E	+	+	5	0	3	P	0	+	23
SRA (IY+s)	FD CB ss 2E	+	+	5	0	3	P	0	+	23

Команды данной группы производят операцию арифметического сдвига вправо над 8-битным значением в памяти по указанному адресу. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Исходный» бит 7 не изменяется, «вытесненный» бит 0 (младший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержащее самой регистровой пары не изменяется.

79. Сложные команды арифметического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SRA A,(IX+s)	DD CB ss 2F	+	+	5	0	3	P	0	+	23
SRA B,(IX+s)	DD CB ss 28	+	+	5	0	3	P	0	+	23
SRA C,(IX+s)	DD CB ss 29	+	+	5	0	3	P	0	+	23
SRA D,(IX+s)	DD CB ss 2A	+	+	5	0	3	P	0	+	23
SRA E,(IX+s)	DD CB ss 2B	+	+	5	0	3	P	0	+	23
SRA H,(IX+s)	DD CB ss 2C	+	+	5	0	3	P	0	+	23
SRA L,(IX+s)	DD CB ss 2D	+	+	5	0	3	P	0	+	23
SRA A,(IY+s)	FD CB ss 2F	+	+	5	0	3	P	0	+	23
SRA B,(IY+s)	FD CB ss 28	+	+	5	0	3	P	0	+	23

SRA C,(IY+s)	FD CB ss 29	+	+	5	0	3	P	0	+	23
SRA D,(IY+s)	FD CB ss 2A	+	+	5	0	3	P	0	+	23
SRA E,(IY+s)	FD CB ss 2B	+	+	5	0	3	P	0	+	23
SRA H,(IY+s)	FD CB ss 2C	+	+	5	0	3	P	0	+	23
SRA L,(IY+s)	FD CB ss 2D	+	+	5	0	3	P	0	+	23

Команды данной группы представляют из себя сложные команды сдвига. Они загружают в указанный 8-битный регистр значение из памяти по указанному адресу, производят операцию арифметического сдвига вправо над указанным 8-битным регистром, а затем сохраняют значение указанного 8-битного регистра в память по указанному адресу. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Исходный» бит 7 не изменяется, «вытесненный» бит 0 (младший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

80. Команды логического сдвига влево значения 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SLA A	CB 27	+	+	5	0	3	P	0	+	8
SLA B	CB 20	+	+	5	0	3	P	0	+	8
SLA C	CB 21	+	+	5	0	3	P	0	+	8
SLA D	CB 22	+	+	5	0	3	P	0	+	8
SLA E	CB 23	+	+	5	0	3	P	0	+	8
SLA H	CB 24	+	+	5	0	3	P	0	+	8
SLA L	CB 25	+	+	5	0	3	P	0	+	8

Команды данной группы производят операцию логического сдвига влево над значением указанного 8-битного регистра. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется нулем, «вытесненный» бит 7 (старший) переходит во флаг переноса C.

81. Команда логического сдвига влево 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SLA (HL)	CB 26	+	+	5	0	3	P	0	+	15

Команды данной группы производят операцию логического сдвига влево над 8-битным значением в памяти по адресу в регистровой паре HL. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется нулем, «вытесненный» бит 7 (старший) переходит во флаг переноса C.

82. Команды логического сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SLA (IX+s)	DD CB ss 26	+	+	5	0	3	P	0	+	23
SLA (IY+s)	FD CB ss 26	+	+	5	0	3	P	0	+	23

Команды данной группы производят операцию логического сдвига влево над 8-битным значением в памяти по указанному адресу. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется нулем, «вытесненный» бит 7 (старший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

83. Сложные команды логического сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SLA A,(IX+s)	DD CB ss 27	+	+	5	0	3	P	0	+	23
SLA B,(IX+s)	DD CB ss 20	+	+	5	0	3	P	0	+	23
SLA C,(IX+s)	DD CB ss 21	+	+	5	0	3	P	0	+	23
SLA D,(IX+s)	DD CB ss 22	+	+	5	0	3	P	0	+	23
SLA E,(IX+s)	DD CB ss 23	+	+	5	0	3	P	0	+	23
SLA H,(IX+s)	DD CB ss 24	+	+	5	0	3	P	0	+	23
SLA L,(IX+s)	DD CB ss 25	+	+	5	0	3	P	0	+	23
SLA A,(IY+s)	FD CB ss 27	+	+	5	0	3	P	0	+	23
SLA B,(IY+s)	FD CB ss 20	+	+	5	0	3	P	0	+	23
SLA C,(IY+s)	FD CB ss 21	+	+	5	0	3	P	0	+	23
SLA D,(IY+s)	FD CB ss 22	+	+	5	0	3	P	0	+	23
SLA E,(IY+s)	FD CB ss 23	+	+	5	0	3	P	0	+	23
SLA H,(IY+s)	FD CB ss 24	+	+	5	0	3	P	0	+	23
SLA L,(IY+s)	FD CB ss 25	+	+	5	0	3	P	0	+	23

Команды данной группы представляют из себя сложные команды сдвига. Они загружают в указанный 8-битный регистр значение из памяти по указанному адресу, производят операцию логического сдвига влево над указанным 8-битным регистром, а затем сохраняют значение указанного 8-битного регистра в память по указанному адресу. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется нулем, «вытесненный» бит 7 (старший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

84. Команды логического сдвига влево с увеличением значения 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SLI A	CB 37	+	+	5	0	3	P	0	+	8
SLI B	CB 30	+	+	5	0	3	P	0	+	8
SLI C	CB 31	+	+	5	0	3	P	0	+	8
SLI D	CB 32	+	+	5	0	3	P	0	+	8
SLI E	CB 33	+	+	5	0	3	P	0	+	8
SLI H	CB 34	+	+	5	0	3	P	0	+	8
SLI L	CB 35	+	+	5	0	3	P	0	+	8

Команды данной группы производят операцию логического сдвига влево над значением указанного 8-битного регистра. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется единицей, «вытесненный» бит 7 (старший) переходит во флаг переноса C.

85. Команда логического сдвига влево с увеличением 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SLI (HL)	CB 36	+	+	5	0	3	P	0	+	15

Команды данной группы производят операцию логического сдвига влево над 8-битным значением в памяти по адресу в регистровой паре HL. Бит 0 (младший) переходит в бит 1, бит 1

– в бит 2, и так далее. «Освободившийся» бит 0 заполняется единицей, «вытесненный» бит 7 (старший) переходит во флаг переноса C.

86. Команды логического сдвига влево с увеличением 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SLI (IX+s)	DD CB ss 36	+	+	5	0	3	P	0	+	23
SLI (IY+s)	FD CB ss 36	+	+	5	0	3	P	0	+	23

Команды данной группы производят операцию логического сдвига влево над 8-битным значением в памяти по указанному адресу. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется единицей, «вытесненный» бит 7 (старший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

87. Сложные команды логического сдвига влево с увеличением 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SLI A,(IX+s)	DD CB ss 37	+	+	5	0	3	P	0	+	23
SLI B,(IX+s)	DD CB ss 30	+	+	5	0	3	P	0	+	23
SLI C,(IX+s)	DD CB ss 31	+	+	5	0	3	P	0	+	23
SLI D,(IX+s)	DD CB ss 32	+	+	5	0	3	P	0	+	23
SLI E,(IX+s)	DD CB ss 33	+	+	5	0	3	P	0	+	23
SLI H,(IX+s)	DD CB ss 34	+	+	5	0	3	P	0	+	23
SLI L,(IX+s)	DD CB ss 35	+	+	5	0	3	P	0	+	23
SLI A,(IY+s)	FD CB ss 37	+	+	5	0	3	P	0	+	23
SLI B,(IY+s)	FD CB ss 30	+	+	5	0	3	P	0	+	23
SLI C,(IY+s)	FD CB ss 31	+	+	5	0	3	P	0	+	23
SLI D,(IY+s)	FD CB ss 32	+	+	5	0	3	P	0	+	23
SLI E,(IY+s)	FD CB ss 33	+	+	5	0	3	P	0	+	23
SLI H,(IY+s)	FD CB ss 34	+	+	5	0	3	P	0	+	23
SLI L,(IY+s)	FD CB ss 35	+	+	5	0	3	P	0	+	23

Команды данной группы представляют из себя сложные команды сдвига. Они загружают в указанный 8-битный регистр значение из памяти по указанному адресу, производят операцию логического сдвига влево над указанным 8-битным регистром, а затем сохраняют значение указанного 8-битного регистра в память по указанному адресу. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется единицей, «вытесненный» бит 7 (старший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

88. Команды расширенного сдвига влево значения 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RL A	CB 17	+	+	5	0	3	P	0	+	8
RL B	CB 10	+	+	5	0	3	P	0	+	8
RL C	CB 11	+	+	5	0	3	P	0	+	8
RL D	CB 12	+	+	5	0	3	P	0	+	8

RL E	CB 13	+	+	5	0	3	P	0	+	8
RL H	CB 14	+	+	5	0	3	P	0	+	8
RL L	CB 15	+	+	5	0	3	P	0	+	8

Команды данной группы производят операцию расширенного сдвига влево над значением указанного 8-битного регистра и флагом переноса C. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется значением флага переноса C, «вытесненный» бит 7 (старший) переходит во флаг переноса C.

89. Команда расширенного сдвига влево 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RL (HL)	CB 16	+	+	5	0	3	P	0	+	15

Команды данной группы производят операцию расширенного сдвига влево над 8-битным значением в памяти по адресу в регистровой паре HL и флагом переноса C. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется значением флага переноса C, «вытесненный» бит 7 (старший) переходит во флаг переноса C.

90. Команды расширенного сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RL (IX+s)	DD CB ss 16	+	+	5	0	3	P	0	+	23
RL (IY+s)	FD CB ss 16	+	+	5	0	3	P	0	+	23

Команды данной группы производят операцию расширенного сдвига влево над 8-битным значением в памяти по указанному адресу и флагом переноса C. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется значением флага переноса C, «вытесненный» бит 7 (старший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, исходная регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

91. Сложные команды расширенного сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RL A,(IX+s)	DD CB ss 17	+	+	5	0	3	P	0	+	23
RL B,(IX+s)	DD CB ss 10	+	+	5	0	3	P	0	+	23
RL C,(IX+s)	DD CB ss 11	+	+	5	0	3	P	0	+	23
RL D,(IX+s)	DD CB ss 12	+	+	5	0	3	P	0	+	23
RL E,(IX+s)	DD CB ss 13	+	+	5	0	3	P	0	+	23
RL H,(IX+s)	DD CB ss 14	+	+	5	0	3	P	0	+	23
RL L,(IX+s)	DD CB ss 15	+	+	5	0	3	P	0	+	23
RL A,(IY+s)	FD CB ss 17	+	+	5	0	3	P	0	+	23
RL B,(IY+s)	FD CB ss 10	+	+	5	0	3	P	0	+	23
RL C,(IY+s)	FD CB ss 11	+	+	5	0	3	P	0	+	23
RL D,(IY+s)	FD CB ss 12	+	+	5	0	3	P	0	+	23
RL E,(IY+s)	FD CB ss 13	+	+	5	0	3	P	0	+	23
RL H,(IY+s)	FD CB ss 14	+	+	5	0	3	P	0	+	23
RL L,(IY+s)	FD CB ss 15	+	+	5	0	3	P	0	+	23

Команды данной группы представляют из себя сложные команды сдвига. Они загружают в указанный 8-битный регистр значение из памяти по указанному адресу, производят операцию расширенного сдвига влево над указанным 8-битным регистром и флагом переноса C, а затем сохраняют значение указанного 8-битного регистра в память по указанному адресу. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется значением флага переноса C, «вытесненный» бит 7 (старший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

92. Команды расширенного сдвига вправо значения 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RR A	CB 1F	+	+	5	0	3	P	0	+	8
RR B	CB 18	+	+	5	0	3	P	0	+	8
RR C	CB 19	+	+	5	0	3	P	0	+	8
RR D	CB 1A	+	+	5	0	3	P	0	+	8
RR E	CB 1B	+	+	5	0	3	P	0	+	8
RR H	CB 1C	+	+	5	0	3	P	0	+	8
RR L	CB 1D	+	+	5	0	3	P	0	+	8

Команды данной группы производят операцию расширенного сдвига вправо над значением указанного 8-битного регистра и флагом переноса C. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется значением флага переноса C, «вытесненный» бит 0 (младший) переходит во флаг переноса C.

93. Команда расширенного сдвига вправо 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RR (HL)	CB 1E	+	+	5	0	3	P	0	+	15

Команды данной группы производят операцию расширенного сдвига вправо над 8-битным значением в памяти по адресу в регистровой паре HL и флагом переноса C. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется значением флага переноса C, «вытесненный» бит 0 (младший) переходит во флаг переноса C.

94. Команды расширенного сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RR (IX+s)	DD CB ss 1E	+	+	5	0	3	P	0	+	23
RR (IY+s)	FD CB ss 1E	+	+	5	0	3	P	0	+	23

Команды данной группы производят операцию расширенного сдвига вправо над 8-битным значением в памяти по указанному адресу и флагом переноса C. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется значением флага переноса C, «вытесненный» бит 0 (младший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

95. Сложные команды расширенного сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RR A,(IX+s)	DD CB ss 1F	+	+	5	0	3	P	0	+	23
RR B,(IX+s)	DD CB ss 18	+	+	5	0	3	P	0	+	23
RR C,(IX+s)	DD CB ss 19	+	+	5	0	3	P	0	+	23
RR D,(IX+s)	DD CB ss 1A	+	+	5	0	3	P	0	+	23
RR E,(IX+s)	DD CB ss 1B	+	+	5	0	3	P	0	+	23
RR H,(IX+s)	DD CB ss 1C	+	+	5	0	3	P	0	+	23
RR L,(IX+s)	DD CB ss 1D	+	+	5	0	3	P	0	+	23
RR A,(IY+s)	FD CB ss 1F	+	+	5	0	3	P	0	+	23
RR B,(IY+s)	FD CB ss 18	+	+	5	0	3	P	0	+	23
RR C,(IY+s)	FD CB ss 19	+	+	5	0	3	P	0	+	23
RR D,(IY+s)	FD CB ss 1A	+	+	5	0	3	P	0	+	23
RR E,(IY+s)	FD CB ss 1B	+	+	5	0	3	P	0	+	23
RR H,(IY+s)	FD CB ss 1C	+	+	5	0	3	P	0	+	23
RR L,(IY+s)	FD CB ss 1D	+	+	5	0	3	P	0	+	23

Команды данной группы представляют из себя сложные команды сдвига. Они загружают в указанный 8-битный регистр значение из памяти по указанному адресу, производят операцию расширенного сдвига вправо над указанным 8-битным регистром и флагом переноса C, а затем сохраняют значение указанного 8-битного регистра в память по указанному адресу. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется значением флага переноса C, «вытесненный» бит 0 (младший) переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержащее самой регистровой пары не изменяется.

96. Команды циклического сдвига влево значения 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RLC A	CB 07	+	+	5	0	3	P	0	+	8
RLC B	CB 00	+	+	5	0	3	P	0	+	8
RLC C	CB 01	+	+	5	0	3	P	0	+	8
RLC D	CB 02	+	+	5	0	3	P	0	+	8
RLC E	CB 03	+	+	5	0	3	P	0	+	8
RLC H	CB 04	+	+	5	0	3	P	0	+	8
RLC L	CB 05	+	+	5	0	3	P	0	+	8

Команды данной группы производят операцию циклического сдвига влево над значением указанного 8-битного регистра. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется значением «вытесненного» бита 7 (старшего), кроме того бит 7 переходит во флаг переноса C.

97. Команда циклического сдвига влево 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RLC (HL)	CB 06	+	+	5	0	3	P	0	+	15

Команды данной группы производят операцию циклического сдвига влево над 8-битным значением в памяти по адресу в регистровой паре HL. Бит 0 (младший) переходит в бит 1, бит 1

– в бит 2, и так далее. «Освободившийся» бит 0 заполняется значением «вытесненного» бита 7 (старшего), кроме того бит 7 переходит во флаг переноса C.

98. Команды циклического сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RLC (IX+s)	DD CB ss 06	+	+	5	0	3	P	0	+	23
RLC (IY+s)	FD CB ss 06	+	+	5	0	3	P	0	+	23

Команды данной группы производят операцию циклического сдвига влево над 8-битным значением в памяти по указанному адресу. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется значением «вытесненного» бита 7 (старшего), кроме того бит 7 переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

99. Сложные команды циклического сдвига влево 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RLC A,(IX+s)	DD CB ss 07	+	+	5	0	3	P	0	+	23
RLC B,(IX+s)	DD CB ss 00	+	+	5	0	3	P	0	+	23
RLC C,(IX+s)	DD CB ss 01	+	+	5	0	3	P	0	+	23
RLC D,(IX+s)	DD CB ss 02	+	+	5	0	3	P	0	+	23
RLC E,(IX+s)	DD CB ss 03	+	+	5	0	3	P	0	+	23
RLC H,(IX+s)	DD CB ss 04	+	+	5	0	3	P	0	+	23
RLC L,(IX+s)	DD CB ss 05	+	+	5	0	3	P	0	+	23
RLC A,(IY+s)	FD CB ss 07	+	+	5	0	3	P	0	+	23
RLC B,(IY+s)	FD CB ss 00	+	+	5	0	3	P	0	+	23
RLC C,(IY+s)	FD CB ss 01	+	+	5	0	3	P	0	+	23
RLC D,(IY+s)	FD CB ss 02	+	+	5	0	3	P	0	+	23
RLC E,(IY+s)	FD CB ss 03	+	+	5	0	3	P	0	+	23
RLC H,(IY+s)	FD CB ss 04	+	+	5	0	3	P	0	+	23
RLC L,(IY+s)	FD CB ss 05	+	+	5	0	3	P	0	+	23

Команды данной группы представляют из себя сложные команды сдвига. Они загружают в указанный 8-битный регистр значение из памяти по указанному адресу, производят операцию циклического сдвига влево над указанным 8-битным регистром, а затем сохраняют значение указанного 8-битного регистра в память по указанному адресу. Бит 0 (младший) переходит в бит 1, бит 1 – в бит 2, и так далее. «Освободившийся» бит 0 заполняется значением «вытесненного» бита 7 (старшего), кроме того бит 7 переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

100. Команды циклического сдвига вправо значения 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RRC A	CB 0F	+	+	5	0	3	P	0	+	8
RRC B	CB 08	+	+	5	0	3	P	0	+	8
RRC C	CB 09	+	+	5	0	3	P	0	+	8
RRC D	CB 0A	+	+	5	0	3	P	0	+	8

RRC E	CB 0B	+	+	5	0	3	P	0	+	8
RRC H	CB 0C	+	+	5	0	3	P	0	+	8
RRC L	CB 0D	+	+	5	0	3	P	0	+	8

Команды данной группы производят операцию циклического сдвига вправо над значением указанного 8-битного регистра. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется значением «вытесненного» бита 0 (младшего), кроме того бит 0 переходит во флаг переноса C.

101. Команда циклического сдвига вправо 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RRC (HL)	CB 0E	+	+	5	0	3	P	0	+	15

Команды данной группы производят операцию циклического сдвига вправо над 8-битным значением в памяти по адресу в регистровой паре HL. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется значением «вытесненного» бита 0 (младшего), кроме того бит 0 переходит во флаг переноса C.

102. Команды циклического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RRC (IX+s)	DD CB ss 0E	+	+	5	0	3	P	0	+	23
RRC (IY+s)	FD CB ss 0E	+	+	5	0	3	P	0	+	23

Команды данной группы производят операцию циклического сдвига вправо над 8-битным значением в памяти по указанному адресу. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется значением «вытесненного» бита 0 (младшего), кроме того бит 0 переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

103. Сложные команды циклического сдвига вправо 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RRC A,(IX+s)	DD CB ss 0F	+	+	5	0	3	P	0	+	23
RRC B,(IX+s)	DD CB ss 08	+	+	5	0	3	P	0	+	23
RRC C,(IX+s)	DD CB ss 09	+	+	5	0	3	P	0	+	23
RRC D,(IX+s)	DD CB ss 0A	+	+	5	0	3	P	0	+	23
RRC E,(IX+s)	DD CB ss 0B	+	+	5	0	3	P	0	+	23
RRC H,(IX+s)	DD CB ss 0C	+	+	5	0	3	P	0	+	23
RRC L,(IX+s)	DD CB ss 0D	+	+	5	0	3	P	0	+	23
RRC A,(IY+s)	FD CB ss 0F	+	+	5	0	3	P	0	+	23
RRC B,(IY+s)	FD CB ss 08	+	+	5	0	3	P	0	+	23
RRC C,(IY+s)	FD CB ss 09	+	+	5	0	3	P	0	+	23
RRC D,(IY+s)	FD CB ss 0A	+	+	5	0	3	P	0	+	23
RRC E,(IY+s)	FD CB ss 0B	+	+	5	0	3	P	0	+	23
RRC H,(IY+s)	FD CB ss 0C	+	+	5	0	3	P	0	+	23
RRC L,(IY+s)	FD CB ss 0D	+	+	5	0	3	P	0	+	23

Команды данной группы представляют из себя сложные команды сдвига. Они загружают в указанный 8-битный регистр значение из памяти по указанному адресу, производят операцию циклического сдвига вправо над указанным 8-битным регистром, а затем сохраняют значение указанного 8-битного регистра в память по указанному адресу. Бит 7 (старший) переходит в бит 6, бит 6 – в бит 5, и так далее. «Освободившийся» бит 7 заполняется значением «вытесненного» бита 0 (младшего), кроме того бит 0 переходит во флаг переноса C. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержащее самой регистровой пары не изменяется.

104. Команды расширенного сдвига аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RLA	17	-	-	5	0	3	-	0	+	4
RRA	1F	-	-	5	0	3	-	0	+	4

Команды данной группы производят операции расширенного сдвига над значением аккумулятора A и флагом переноса C. Они являются более короткими и быстродействующими функциональными аналогами команд RL A и RR A, однако, в отличие от этих команд, не воздействуют на флаги S, Z и P/V.

105. Команды циклического сдвига аккумулятора

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RLCA	07	-	-	5	0	3	-	0	+	4
RRC A	0F	-	-	5	0	3	-	0	+	4

Команды данной группы производят операции циклического сдвига над значением аккумулятора A. Они являются более короткими и быстродействующими функциональными аналогами команд RLC A и RRC A, однако, в отличие от этих команд, не воздействуют на флаги S, Z и P/V.

106. Команды установки бита внутри 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SET 0,A	CB C7	-	-	-	-	-	-	-	-	8
SET 1,A	CB CF	-	-	-	-	-	-	-	-	8
SET 2,A	CB D7	-	-	-	-	-	-	-	-	8
SET 3,A	CB DF	-	-	-	-	-	-	-	-	8
SET 4,A	CB E7	-	-	-	-	-	-	-	-	8
SET 5,A	CB EF	-	-	-	-	-	-	-	-	8
SET 6,A	CB F7	-	-	-	-	-	-	-	-	8
SET 7,A	CB FF	-	-	-	-	-	-	-	-	8
SET 0,B	CB C0	-	-	-	-	-	-	-	-	8
SET 1,B	CB C8	-	-	-	-	-	-	-	-	8
SET 2,B	CB D0	-	-	-	-	-	-	-	-	8
SET 3,B	CB D8	-	-	-	-	-	-	-	-	8
SET 4,B	CB E0	-	-	-	-	-	-	-	-	8
SET 5,B	CB E8	-	-	-	-	-	-	-	-	8
SET 6,B	CB F0	-	-	-	-	-	-	-	-	8
SET 7,B	CB F8	-	-	-	-	-	-	-	-	8
SET 0,C	CB C1	-	-	-	-	-	-	-	-	8
SET 1,C	CB C9	-	-	-	-	-	-	-	-	8
SET 2,C	CB D1	-	-	-	-	-	-	-	-	8
SET 3,C	CB D9	-	-	-	-	-	-	-	-	8
SET 4,C	CB E1	-	-	-	-	-	-	-	-	8
SET 5,C	CB E9	-	-	-	-	-	-	-	-	8

SET 6,C	CB F1	-	-	-	-	-	-	-	-	8
SET 7,C	CB F9	-	-	-	-	-	-	-	-	8
SET 0,D	CB C2	-	-	-	-	-	-	-	-	8
SET 1,D	CB CA	-	-	-	-	-	-	-	-	8
SET 2,D	CB D2	-	-	-	-	-	-	-	-	8
SET 3,D	CB DA	-	-	-	-	-	-	-	-	8
SET 4,D	CB E2	-	-	-	-	-	-	-	-	8
SET 5,D	CB EA	-	-	-	-	-	-	-	-	8
SET 6,D	CB F2	-	-	-	-	-	-	-	-	8
SET 7,D	CB FA	-	-	-	-	-	-	-	-	8
SET 0,E	CB C3	-	-	-	-	-	-	-	-	8
SET 1,E	CB CB	-	-	-	-	-	-	-	-	8
SET 2,E	CB D3	-	-	-	-	-	-	-	-	8
SET 3,E	CB DB	-	-	-	-	-	-	-	-	8
SET 4,E	CB E3	-	-	-	-	-	-	-	-	8
SET 5,E	CB EB	-	-	-	-	-	-	-	-	8
SET 6,E	CB F3	-	-	-	-	-	-	-	-	8
SET 7,E	CB FB	-	-	-	-	-	-	-	-	8
SET 0,H	CB C4	-	-	-	-	-	-	-	-	8
SET 1,H	CB CC	-	-	-	-	-	-	-	-	8
SET 2,H	CB D4	-	-	-	-	-	-	-	-	8
SET 3,H	CB DC	-	-	-	-	-	-	-	-	8
SET 4,H	CB E4	-	-	-	-	-	-	-	-	8
SET 5,H	CB EC	-	-	-	-	-	-	-	-	8
SET 6,H	CB F4	-	-	-	-	-	-	-	-	8
SET 7,H	CB FC	-	-	-	-	-	-	-	-	8
SET 0,L	CB C5	-	-	-	-	-	-	-	-	8
SET 1,L	CB CD	-	-	-	-	-	-	-	-	8
SET 2,L	CB D5	-	-	-	-	-	-	-	-	8
SET 3,L	CB DD	-	-	-	-	-	-	-	-	8
SET 4,L	CB E5	-	-	-	-	-	-	-	-	8
SET 5,L	CB ED	-	-	-	-	-	-	-	-	8
SET 6,L	CB F5	-	-	-	-	-	-	-	-	8
SET 7,L	CB FD	-	-	-	-	-	-	-	-	8

Команды данной группы устанавливают указанный бит в указанном 8-битном регистре в единицу.

107. Команда установки бита внутри 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SET 0,(HL)	CB C6	-	-	-	-	-	-	-	-	15
SET 1,(HL)	CB CE	-	-	-	-	-	-	-	-	15
SET 2,(HL)	CB D6	-	-	-	-	-	-	-	-	15
SET 3,(HL)	CB DE	-	-	-	-	-	-	-	-	15
SET 4,(HL)	CB E6	-	-	-	-	-	-	-	-	15
SET 5,(HL)	CB EE	-	-	-	-	-	-	-	-	15
SET 6,(HL)	CB F6	-	-	-	-	-	-	-	-	15
SET 7,(HL)	CB FE	-	-	-	-	-	-	-	-	15

Команды данной группы устанавливают указанный бит в 8-битной ячейке памяти по адресу в регистровой паре HL в единицу.

108. Команды установки бита внутри 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SET 0,(IX+s)	DD CB ss C6	-	-	-	-	-	-	-	-	23
SET 1,(IX+s)	DD CB ss CE	-	-	-	-	-	-	-	-	23
SET 2,(IX+s)	DD CB ss D6	-	-	-	-	-	-	-	-	23
SET 3,(IX+s)	DD CB ss DE	-	-	-	-	-	-	-	-	23
SET 4,(IX+s)	DD CB ss E6	-	-	-	-	-	-	-	-	23
SET 5,(IX+s)	DD CB ss EE	-	-	-	-	-	-	-	-	23
SET 6,(IX+s)	DD CB ss F6	-	-	-	-	-	-	-	-	23
SET 7,(IX+s)	DD CB ss FE	-	-	-	-	-	-	-	-	23
SET 0,(IY+s)	FD CB ss C6	-	-	-	-	-	-	-	-	23
SET 1,(IY+s)	FD CB ss CE	-	-	-	-	-	-	-	-	23
SET 2,(IY+s)	FD CB ss D6	-	-	-	-	-	-	-	-	23
SET 3,(IY+s)	FD CB ss DE	-	-	-	-	-	-	-	-	23
SET 4,(IY+s)	FD CB ss E6	-	-	-	-	-	-	-	-	23
SET 5,(IY+s)	FD CB ss EE	-	-	-	-	-	-	-	-	23
SET 6,(IY+s)	FD CB ss F6	-	-	-	-	-	-	-	-	23
SET 7,(IY+s)	FD CB ss FE	-	-	-	-	-	-	-	-	23

Команды данной группы устанавливают указанный бит в 8-битной ячейке памяти по указанному адресу в единицу. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

109. Сложные команды установки бита внутри 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
SET 0,A,(IX+s)	DD CB ss C7	-	-	-	-	-	-	-	-	23
SET 1,A,(IX+s)	DD CB ss CF	-	-	-	-	-	-	-	-	23
SET 2,A,(IX+s)	DD CB ss D7	-	-	-	-	-	-	-	-	23
SET 3,A,(IX+s)	DD CB ss DF	-	-	-	-	-	-	-	-	23
SET 4,A,(IX+s)	DD CB ss E7	-	-	-	-	-	-	-	-	23
SET 5,A,(IX+s)	DD CB ss EF	-	-	-	-	-	-	-	-	23
SET 6,A,(IX+s)	DD CB ss F7	-	-	-	-	-	-	-	-	23
SET 7,A,(IX+s)	DD CB ss FF	-	-	-	-	-	-	-	-	23
SET 0,B,(IX+s)	DD CB ss C0	-	-	-	-	-	-	-	-	23
SET 1,B,(IX+s)	DD CB ss C8	-	-	-	-	-	-	-	-	23
SET 2,B,(IX+s)	DD CB ss D0	-	-	-	-	-	-	-	-	23
SET 3,B,(IX+s)	DD CB ss D8	-	-	-	-	-	-	-	-	23
SET 4,B,(IX+s)	DD CB ss E0	-	-	-	-	-	-	-	-	23
SET 5,B,(IX+s)	DD CB ss E8	-	-	-	-	-	-	-	-	23
SET 6,B,(IX+s)	DD CB ss F0	-	-	-	-	-	-	-	-	23
SET 7,B,(IX+s)	DD CB ss F8	-	-	-	-	-	-	-	-	23
SET 0,C,(IX+s)	DD CB ss C1	-	-	-	-	-	-	-	-	23
SET 1,C,(IX+s)	DD CB ss C9	-	-	-	-	-	-	-	-	23
SET 2,C,(IX+s)	DD CB ss D1	-	-	-	-	-	-	-	-	23
SET 3,C,(IX+s)	DD CB ss D9	-	-	-	-	-	-	-	-	23
SET 4,C,(IX+s)	DD CB ss E1	-	-	-	-	-	-	-	-	23
SET 5,C,(IX+s)	DD CB ss E9	-	-	-	-	-	-	-	-	23
SET 6,C,(IX+s)	DD CB ss F1	-	-	-	-	-	-	-	-	23

SET 7,C,(IX+s)	DD CB ss F9	-	-	-	-	-	-	-	-	23
SET 0,D,(IX+s)	DD CB ss C2	-	-	-	-	-	-	-	-	23
SET 1,D,(IX+s)	DD CB ss CA	-	-	-	-	-	-	-	-	23
SET 2,D,(IX+s)	DD CB ss D2	-	-	-	-	-	-	-	-	23
SET 3,D,(IX+s)	DD CB ss DA	-	-	-	-	-	-	-	-	23
SET 4,D,(IX+s)	DD CB ss E2	-	-	-	-	-	-	-	-	23
SET 5,D,(IX+s)	DD CB ss EA	-	-	-	-	-	-	-	-	23
SET 6,D,(IX+s)	DD CB ss F2	-	-	-	-	-	-	-	-	23
SET 7,D,(IX+s)	DD CB ss FA	-	-	-	-	-	-	-	-	23
SET 0,E,(IX+s)	DD CB ss C3	-	-	-	-	-	-	-	-	23
SET 1,E,(IX+s)	DD CB ss CB	-	-	-	-	-	-	-	-	23
SET 2,E,(IX+s)	DD CB ss D3	-	-	-	-	-	-	-	-	23
SET 3,E,(IX+s)	DD CB ss DB	-	-	-	-	-	-	-	-	23
SET 4,E,(IX+s)	DD CB ss E3	-	-	-	-	-	-	-	-	23
SET 5,E,(IX+s)	DD CB ss EB	-	-	-	-	-	-	-	-	23
SET 6,E,(IX+s)	DD CB ss F3	-	-	-	-	-	-	-	-	23
SET 7,E,(IX+s)	DD CB ss FB	-	-	-	-	-	-	-	-	23
SET 0,H,(IX+s)	DD CB ss C4	-	-	-	-	-	-	-	-	23
SET 1,H,(IX+s)	DD CB ss CC	-	-	-	-	-	-	-	-	23
SET 2,H,(IX+s)	DD CB ss D4	-	-	-	-	-	-	-	-	23
SET 3,H,(IX+s)	DD CB ss DC	-	-	-	-	-	-	-	-	23
SET 4,H,(IX+s)	DD CB ss E4	-	-	-	-	-	-	-	-	23
SET 5,H,(IX+s)	DD CB ss EC	-	-	-	-	-	-	-	-	23
SET 6,H,(IX+s)	DD CB ss F4	-	-	-	-	-	-	-	-	23
SET 7,H,(IX+s)	DD CB ss FC	-	-	-	-	-	-	-	-	23
SET 0,L,(IX+s)	DD CB ss C5	-	-	-	-	-	-	-	-	23
SET 1,L,(IX+s)	DD CB ss CD	-	-	-	-	-	-	-	-	23
SET 2,L,(IX+s)	DD CB ss D5	-	-	-	-	-	-	-	-	23
SET 3,L,(IX+s)	DD CB ss DD	-	-	-	-	-	-	-	-	23
SET 4,L,(IX+s)	DD CB ss E5	-	-	-	-	-	-	-	-	23
SET 5,L,(IX+s)	DD CB ss ED	-	-	-	-	-	-	-	-	23
SET 6,L,(IX+s)	DD CB ss F5	-	-	-	-	-	-	-	-	23
SET 7,L,(IX+s)	DD CB ss FD	-	-	-	-	-	-	-	-	23
SET 0,A,(IY+s)	FD CB ss C7	-	-	-	-	-	-	-	-	23
SET 1,A,(IY+s)	FD CB ss CF	-	-	-	-	-	-	-	-	23
SET 2,A,(IY+s)	FD CB ss D7	-	-	-	-	-	-	-	-	23
SET 3,A,(IY+s)	FD CB ss DF	-	-	-	-	-	-	-	-	23
SET 4,A,(IY+s)	FD CB ss E7	-	-	-	-	-	-	-	-	23
SET 5,A,(IY+s)	FD CB ss EF	-	-	-	-	-	-	-	-	23
SET 6,A,(IY+s)	FD CB ss F7	-	-	-	-	-	-	-	-	23
SET 7,A,(IY+s)	FD CB ss FF	-	-	-	-	-	-	-	-	23
SET 0,B,(IY+s)	FD CB ss C0	-	-	-	-	-	-	-	-	23
SET 1,B,(IY+s)	FD CB ss C8	-	-	-	-	-	-	-	-	23
SET 2,B,(IY+s)	FD CB ss D0	-	-	-	-	-	-	-	-	23
SET 3,B,(IY+s)	FD CB ss D8	-	-	-	-	-	-	-	-	23
SET 4,B,(IY+s)	FD CB ss E0	-	-	-	-	-	-	-	-	23
SET 5,B,(IY+s)	FD CB ss E8	-	-	-	-	-	-	-	-	23
SET 6,B,(IY+s)	FD CB ss F0	-	-	-	-	-	-	-	-	23
SET 7,B,(IY+s)	FD CB ss F8	-	-	-	-	-	-	-	-	23
SET 0,C,(IY+s)	FD CB ss C1	-	-	-	-	-	-	-	-	23
SET 1,C,(IY+s)	FD CB ss C9	-	-	-	-	-	-	-	-	23
SET 2,C,(IY+s)	FD CB ss D1	-	-	-	-	-	-	-	-	23
SET 3,C,(IY+s)	FD CB ss D9	-	-	-	-	-	-	-	-	23
SET 4,C,(IY+s)	FD CB ss E1	-	-	-	-	-	-	-	-	23
SET 5,C,(IY+s)	FD CB ss E9	-	-	-	-	-	-	-	-	23

SET 6,C,(IY+s)	FD CB ss F1	-	-	-	-	-	-	-	-	23
SET 7,C,(IY+s)	FD CB ss F9	-	-	-	-	-	-	-	-	23
SET 0,D,(IY+s)	FD CB ss C2	-	-	-	-	-	-	-	-	23
SET 1,D,(IY+s)	FD CB ss CA	-	-	-	-	-	-	-	-	23
SET 2,D,(IY+s)	FD CB ss D2	-	-	-	-	-	-	-	-	23
SET 3,D,(IY+s)	FD CB ss DA	-	-	-	-	-	-	-	-	23
SET 4,D,(IY+s)	FD CB ss E2	-	-	-	-	-	-	-	-	23
SET 5,D,(IY+s)	FD CB ss EA	-	-	-	-	-	-	-	-	23
SET 6,D,(IY+s)	FD CB ss F2	-	-	-	-	-	-	-	-	23
SET 7,D,(IY+s)	FD CB ss FA	-	-	-	-	-	-	-	-	23
SET 0,E,(IY+s)	FD CB ss C3	-	-	-	-	-	-	-	-	23
SET 1,E,(IY+s)	FD CB ss CB	-	-	-	-	-	-	-	-	23
SET 2,E,(IY+s)	FD CB ss D3	-	-	-	-	-	-	-	-	23
SET 3,E,(IY+s)	FD CB ss DB	-	-	-	-	-	-	-	-	23
SET 4,E,(IY+s)	FD CB ss E3	-	-	-	-	-	-	-	-	23
SET 5,E,(IY+s)	FD CB ss EB	-	-	-	-	-	-	-	-	23
SET 6,E,(IY+s)	FD CB ss F3	-	-	-	-	-	-	-	-	23
SET 7,E,(IY+s)	FD CB ss FB	-	-	-	-	-	-	-	-	23
SET 0,H,(IY+s)	FD CB ss C4	-	-	-	-	-	-	-	-	23
SET 1,H,(IY+s)	FD CB ss CC	-	-	-	-	-	-	-	-	23
SET 2,H,(IY+s)	FD CB ss D4	-	-	-	-	-	-	-	-	23
SET 3,H,(IY+s)	FD CB ss DC	-	-	-	-	-	-	-	-	23
SET 4,H,(IY+s)	FD CB ss E4	-	-	-	-	-	-	-	-	23
SET 5,H,(IY+s)	FD CB ss EC	-	-	-	-	-	-	-	-	23
SET 6,H,(IY+s)	FD CB ss F4	-	-	-	-	-	-	-	-	23
SET 7,H,(IY+s)	FD CB ss FC	-	-	-	-	-	-	-	-	23
SET 0,L,(IY+s)	FD CB ss C5	-	-	-	-	-	-	-	-	23
SET 1,L,(IY+s)	FD CB ss CD	-	-	-	-	-	-	-	-	23
SET 2,L,(IY+s)	FD CB ss D5	-	-	-	-	-	-	-	-	23
SET 3,L,(IY+s)	FD CB ss DD	-	-	-	-	-	-	-	-	23
SET 4,L,(IY+s)	FD CB ss E5	-	-	-	-	-	-	-	-	23
SET 5,L,(IY+s)	FD CB ss ED	-	-	-	-	-	-	-	-	23
SET 6,L,(IY+s)	FD CB ss F5	-	-	-	-	-	-	-	-	23
SET 7,L,(IY+s)	FD CB ss FD	-	-	-	-	-	-	-	-	23

Команды данной группы представляют из себя сложные команды установки бита. Они загружают в указанный 8-битный регистр значение из памяти по указанному адресу, производят операцию установки указанного бита в указанном 8-битном регистре в единицу, а затем сохраняют значение указанного 8-битного регистра в память по указанному адресу. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

110. Команды сброса бита внутри 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RES 0,A	CB 87	-	-	-	-	-	-	-	-	8
RES 1,A	CB 8F	-	-	-	-	-	-	-	-	8
RES 2,A	CB 97	-	-	-	-	-	-	-	-	8
RES 3,A	CB 9F	-	-	-	-	-	-	-	-	8
RES 4,A	CB A7	-	-	-	-	-	-	-	-	8
RES 5,A	CB AF	-	-	-	-	-	-	-	-	8
RES 6,A	CB B7	-	-	-	-	-	-	-	-	8
RES 7,A	CB BF	-	-	-	-	-	-	-	-	8
RES 0,B	CB 80	-	-	-	-	-	-	-	-	8
RES 1,B	CB 88	-	-	-	-	-	-	-	-	8

RES 2,B	CB 90	-	-	-	-	-	-	-	-	8
RES 3,B	CB 98	-	-	-	-	-	-	-	-	8
RES 4,B	CB A0	-	-	-	-	-	-	-	-	8
RES 5,B	CB A8	-	-	-	-	-	-	-	-	8
RES 6,B	CB B0	-	-	-	-	-	-	-	-	8
RES 7,B	CB B8	-	-	-	-	-	-	-	-	8
RES 0,C	CB 81	-	-	-	-	-	-	-	-	8
RES 1,C	CB 89	-	-	-	-	-	-	-	-	8
RES 2,C	CB 91	-	-	-	-	-	-	-	-	8
RES 3,C	CB 99	-	-	-	-	-	-	-	-	8
RES 4,C	CB A1	-	-	-	-	-	-	-	-	8
RES 5,C	CB A9	-	-	-	-	-	-	-	-	8
RES 6,C	CB B1	-	-	-	-	-	-	-	-	8
RES 7,C	CB B9	-	-	-	-	-	-	-	-	8
RES 0,D	CB 82	-	-	-	-	-	-	-	-	8
RES 1,D	CB 8A	-	-	-	-	-	-	-	-	8
RES 2,D	CB 92	-	-	-	-	-	-	-	-	8
RES 3,D	CB 9A	-	-	-	-	-	-	-	-	8
RES 4,D	CB A2	-	-	-	-	-	-	-	-	8
RES 5,D	CB AA	-	-	-	-	-	-	-	-	8
RES 6,D	CB B2	-	-	-	-	-	-	-	-	8
RES 7,D	CB BA	-	-	-	-	-	-	-	-	8
RES 0,E	CB 83	-	-	-	-	-	-	-	-	8
RES 1,E	CB 8B	-	-	-	-	-	-	-	-	8
RES 2,E	CB 93	-	-	-	-	-	-	-	-	8
RES 3,E	CB 9B	-	-	-	-	-	-	-	-	8
RES 4,E	CB A3	-	-	-	-	-	-	-	-	8
RES 5,E	CB AB	-	-	-	-	-	-	-	-	8
RES 6,E	CB B3	-	-	-	-	-	-	-	-	8
RES 7,E	CB BB	-	-	-	-	-	-	-	-	8
RES 0,H	CB 84	-	-	-	-	-	-	-	-	8
RES 1,H	CB 8C	-	-	-	-	-	-	-	-	8
RES 2,H	CB 94	-	-	-	-	-	-	-	-	8
RES 3,H	CB 9C	-	-	-	-	-	-	-	-	8
RES 4,H	CB A4	-	-	-	-	-	-	-	-	8
RES 5,H	CB AC	-	-	-	-	-	-	-	-	8
RES 6,H	CB B4	-	-	-	-	-	-	-	-	8
RES 7,H	CB BC	-	-	-	-	-	-	-	-	8
RES 0,L	CB 85	-	-	-	-	-	-	-	-	8
RES 1,L	CB 8D	-	-	-	-	-	-	-	-	8
RES 2,L	CB 95	-	-	-	-	-	-	-	-	8
RES 3,L	CB 9D	-	-	-	-	-	-	-	-	8
RES 4,L	CB A5	-	-	-	-	-	-	-	-	8
RES 5,L	CB AD	-	-	-	-	-	-	-	-	8
RES 6,L	CB B5	-	-	-	-	-	-	-	-	8
RES 7,L	CB BD	-	-	-	-	-	-	-	-	8

Команды данной группы сбрасывают указанный бит в указанном 8-битном регистре в ноль.

111. Команда сброса бита внутри 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RES 0,(HL)	CB 86	-	-	-	-	-	-	-	-	15
RES 1,(HL)	CB 8E	-	-	-	-	-	-	-	-	15

RES 2,(HL)	CB 96	-	-	-	-	-	-	-	-	15
RES 3,(HL)	CB 9E	-	-	-	-	-	-	-	-	15
RES 4,(HL)	CB A6	-	-	-	-	-	-	-	-	15
RES 5,(HL)	CB AE	-	-	-	-	-	-	-	-	15
RES 6,(HL)	CB B6	-	-	-	-	-	-	-	-	15
RES 7,(HL)	CB BE	-	-	-	-	-	-	-	-	15

Команды данной группы сбрасывают указанный бит в 8-битной ячейке памяти по адресу в регистровой паре HL в ноль.

112. Команды сброса бита внутри 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RES 0,(IX+ss)	DD CB ss 86	-	-	-	-	-	-	-	-	23
RES 1,(IX+ss)	DD CB ss 8E	-	-	-	-	-	-	-	-	23
RES 2,(IX+ss)	DD CB ss 96	-	-	-	-	-	-	-	-	23
RES 3,(IX+ss)	DD CB ss 9E	-	-	-	-	-	-	-	-	23
RES 4,(IX+ss)	DD CB ss A6	-	-	-	-	-	-	-	-	23
RES 5,(IX+ss)	DD CB ss AE	-	-	-	-	-	-	-	-	23
RES 6,(IX+ss)	DD CB ss B6	-	-	-	-	-	-	-	-	23
RES 7,(IX+ss)	DD CB ss BE	-	-	-	-	-	-	-	-	23
RES 0,(IY+ss)	FD CB ss 86	-	-	-	-	-	-	-	-	23
RES 1,(IY+ss)	FD CB ss 8E	-	-	-	-	-	-	-	-	23
RES 2,(IY+ss)	FD CB ss 96	-	-	-	-	-	-	-	-	23
RES 3,(IY+ss)	FD CB ss 9E	-	-	-	-	-	-	-	-	23
RES 4,(IY+ss)	FD CB ss A6	-	-	-	-	-	-	-	-	23
RES 5,(IY+ss)	FD CB ss AE	-	-	-	-	-	-	-	-	23
RES 6,(IY+ss)	FD CB ss B6	-	-	-	-	-	-	-	-	23
RES 7,(IY+ss)	FD CB ss BE	-	-	-	-	-	-	-	-	23

Команды данной группы сбрасывают указанный бит в 8-битной ячейке памяти по указанному адресу в ноль. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

113. Сложные команды сброса бита внутри 8-битного значения в памяти по адресу в индексном регистре (со смещением) с сохранением результата в 8-битном регистре

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
RES 0,A,(IX+ss)	DD CB ss 87	-	-	-	-	-	-	-	-	23
RES 1,A,(IX+ss)	DD CB ss 8F	-	-	-	-	-	-	-	-	23
RES 2,A,(IX+ss)	DD CB ss 97	-	-	-	-	-	-	-	-	23
RES 3,A,(IX+ss)	DD CB ss 9F	-	-	-	-	-	-	-	-	23
RES 4,A,(IX+ss)	DD CB ss A7	-	-	-	-	-	-	-	-	23
RES 5,A,(IX+ss)	DD CB ss AF	-	-	-	-	-	-	-	-	23
RES 6,A,(IX+ss)	DD CB ss B7	-	-	-	-	-	-	-	-	23
RES 7,A,(IX+ss)	DD CB ss BF	-	-	-	-	-	-	-	-	23
RES 0,B,(IX+ss)	DD CB ss 80	-	-	-	-	-	-	-	-	23
RES 1,B,(IX+ss)	DD CB ss 88	-	-	-	-	-	-	-	-	23
RES 2,B,(IX+ss)	DD CB ss 90	-	-	-	-	-	-	-	-	23
RES 3,B,(IX+ss)	DD CB ss 98	-	-	-	-	-	-	-	-	23
RES 4,B,(IX+ss)	DD CB ss A0	-	-	-	-	-	-	-	-	23

RES 5,B,(IX+s)	DD CB ss A8	-	-	-	-	-	-	-	-	23
RES 6,B,(IX+s)	DD CB ss B0	-	-	-	-	-	-	-	-	23
RES 7,B,(IX+s)	DD CB ss B8	-	-	-	-	-	-	-	-	23
RES 0,C,(IX+s)	DD CB ss 81	-	-	-	-	-	-	-	-	23
RES 1,C,(IX+s)	DD CB ss 89	-	-	-	-	-	-	-	-	23
RES 2,C,(IX+s)	DD CB ss 91	-	-	-	-	-	-	-	-	23
RES 3,C,(IX+s)	DD CB ss 99	-	-	-	-	-	-	-	-	23
RES 4,C,(IX+s)	DD CB ss A1	-	-	-	-	-	-	-	-	23
RES 5,C,(IX+s)	DD CB ss A9	-	-	-	-	-	-	-	-	23
RES 6,C,(IX+s)	DD CB ss B1	-	-	-	-	-	-	-	-	23
RES 7,C,(IX+s)	DD CB ss B9	-	-	-	-	-	-	-	-	23
RES 0,D,(IX+s)	DD CB ss 82	-	-	-	-	-	-	-	-	23
RES 1,D,(IX+s)	DD CB ss 8A	-	-	-	-	-	-	-	-	23
RES 2,D,(IX+s)	DD CB ss 92	-	-	-	-	-	-	-	-	23
RES 3,D,(IX+s)	DD CB ss 9A	-	-	-	-	-	-	-	-	23
RES 4,D,(IX+s)	DD CB ss A2	-	-	-	-	-	-	-	-	23
RES 5,D,(IX+s)	DD CB ss AA	-	-	-	-	-	-	-	-	23
RES 6,D,(IX+s)	DD CB ss B2	-	-	-	-	-	-	-	-	23
RES 7,D,(IX+s)	DD CB ss BA	-	-	-	-	-	-	-	-	23
RES 0,E,(IX+s)	DD CB ss 83	-	-	-	-	-	-	-	-	23
RES 1,E,(IX+s)	DD CB ss 8B	-	-	-	-	-	-	-	-	23
RES 2,E,(IX+s)	DD CB ss 93	-	-	-	-	-	-	-	-	23
RES 3,E,(IX+s)	DD CB ss 9B	-	-	-	-	-	-	-	-	23
RES 4,E,(IX+s)	DD CB ss A3	-	-	-	-	-	-	-	-	23
RES 5,E,(IX+s)	DD CB ss AB	-	-	-	-	-	-	-	-	23
RES 6,E,(IX+s)	DD CB ss B3	-	-	-	-	-	-	-	-	23
RES 7,E,(IX+s)	DD CB ss BB	-	-	-	-	-	-	-	-	23
RES 0,H,(IX+s)	DD CB ss 84	-	-	-	-	-	-	-	-	23
RES 1,H,(IX+s)	DD CB ss 8C	-	-	-	-	-	-	-	-	23
RES 2,H,(IX+s)	DD CB ss 94	-	-	-	-	-	-	-	-	23
RES 3,H,(IX+s)	DD CB ss 9C	-	-	-	-	-	-	-	-	23
RES 4,H,(IX+s)	DD CB ss A4	-	-	-	-	-	-	-	-	23
RES 5,H,(IX+s)	DD CB ss AC	-	-	-	-	-	-	-	-	23
RES 6,H,(IX+s)	DD CB ss B4	-	-	-	-	-	-	-	-	23
RES 7,H,(IX+s)	DD CB ss BC	-	-	-	-	-	-	-	-	23
RES 0,L,(IX+s)	DD CB ss 85	-	-	-	-	-	-	-	-	23
RES 1,L,(IX+s)	DD CB ss 8D	-	-	-	-	-	-	-	-	23
RES 2,L,(IX+s)	DD CB ss 95	-	-	-	-	-	-	-	-	23
RES 3,L,(IX+s)	DD CB ss 9D	-	-	-	-	-	-	-	-	23
RES 4,L,(IX+s)	DD CB ss A5	-	-	-	-	-	-	-	-	23
RES 5,L,(IX+s)	DD CB ss AD	-	-	-	-	-	-	-	-	23
RES 6,L,(IX+s)	DD CB ss B5	-	-	-	-	-	-	-	-	23
RES 7,L,(IX+s)	DD CB ss BD	-	-	-	-	-	-	-	-	23
RES 0,A,(IY+s)	FD CB ss 87	-	-	-	-	-	-	-	-	23
RES 1,A,(IY+s)	FD CB ss 8F	-	-	-	-	-	-	-	-	23
RES 2,A,(IY+s)	FD CB ss 97	-	-	-	-	-	-	-	-	23
RES 3,A,(IY+s)	FD CB ss 9F	-	-	-	-	-	-	-	-	23
RES 4,A,(IY+s)	FD CB ss A7	-	-	-	-	-	-	-	-	23
RES 5,A,(IY+s)	FD CB ss AF	-	-	-	-	-	-	-	-	23
RES 6,A,(IY+s)	FD CB ss B7	-	-	-	-	-	-	-	-	23
RES 7,A,(IY+s)	FD CB ss BF	-	-	-	-	-	-	-	-	23
RES 0,B,(IY+s)	FD CB ss 80	-	-	-	-	-	-	-	-	23
RES 1,B,(IY+s)	FD CB ss 88	-	-	-	-	-	-	-	-	23
RES 2,B,(IY+s)	FD CB ss 90	-	-	-	-	-	-	-	-	23
RES 3,B,(IY+s)	FD CB ss 98	-	-	-	-	-	-	-	-	23

RES 4,B,(IY+s)	FD CB ss A0	-	-	-	-	-	-	-	-	-	23
RES 5,B,(IY+s)	FD CB ss A8	-	-	-	-	-	-	-	-	-	23
RES 6,B,(IY+s)	FD CB ss B0	-	-	-	-	-	-	-	-	-	23
RES 7,B,(IY+s)	FD CB ss B8	-	-	-	-	-	-	-	-	-	23
RES 0,C,(IY+s)	FD CB ss 81	-	-	-	-	-	-	-	-	-	23
RES 1,C,(IY+s)	FD CB ss 89	-	-	-	-	-	-	-	-	-	23
RES 2,C,(IY+s)	FD CB ss 91	-	-	-	-	-	-	-	-	-	23
RES 3,C,(IY+s)	FD CB ss 99	-	-	-	-	-	-	-	-	-	23
RES 4,C,(IY+s)	FD CB ss A1	-	-	-	-	-	-	-	-	-	23
RES 5,C,(IY+s)	FD CB ss A9	-	-	-	-	-	-	-	-	-	23
RES 6,C,(IY+s)	FD CB ss B1	-	-	-	-	-	-	-	-	-	23
RES 7,C,(IY+s)	FD CB ss B9	-	-	-	-	-	-	-	-	-	23
RES 0,D,(IY+s)	FD CB ss 82	-	-	-	-	-	-	-	-	-	23
RES 1,D,(IY+s)	FD CB ss 8A	-	-	-	-	-	-	-	-	-	23
RES 2,D,(IY+s)	FD CB ss 92	-	-	-	-	-	-	-	-	-	23
RES 3,D,(IY+s)	FD CB ss 9A	-	-	-	-	-	-	-	-	-	23
RES 4,D,(IY+s)	FD CB ss A2	-	-	-	-	-	-	-	-	-	23
RES 5,D,(IY+s)	FD CB ss AA	-	-	-	-	-	-	-	-	-	23
RES 6,D,(IY+s)	FD CB ss B2	-	-	-	-	-	-	-	-	-	23
RES 7,D,(IY+s)	FD CB ss BA	-	-	-	-	-	-	-	-	-	23
RES 0,E,(IY+s)	FD CB ss 83	-	-	-	-	-	-	-	-	-	23
RES 1,E,(IY+s)	FD CB ss 8B	-	-	-	-	-	-	-	-	-	23
RES 2,E,(IY+s)	FD CB ss 93	-	-	-	-	-	-	-	-	-	23
RES 3,E,(IY+s)	FD CB ss 9B	-	-	-	-	-	-	-	-	-	23
RES 4,E,(IY+s)	FD CB ss A3	-	-	-	-	-	-	-	-	-	23
RES 5,E,(IY+s)	FD CB ss AB	-	-	-	-	-	-	-	-	-	23
RES 6,E,(IY+s)	FD CB ss B3	-	-	-	-	-	-	-	-	-	23
RES 7,E,(IY+s)	FD CB ss BB	-	-	-	-	-	-	-	-	-	23
RES 0,H,(IY+s)	FD CB ss 84	-	-	-	-	-	-	-	-	-	23
RES 1,H,(IY+s)	FD CB ss 8C	-	-	-	-	-	-	-	-	-	23
RES 2,H,(IY+s)	FD CB ss 94	-	-	-	-	-	-	-	-	-	23
RES 3,H,(IY+s)	FD CB ss 9C	-	-	-	-	-	-	-	-	-	23
RES 4,H,(IY+s)	FD CB ss A4	-	-	-	-	-	-	-	-	-	23
RES 5,H,(IY+s)	FD CB ss AC	-	-	-	-	-	-	-	-	-	23
RES 6,H,(IY+s)	FD CB ss B4	-	-	-	-	-	-	-	-	-	23
RES 7,H,(IY+s)	FD CB ss BC	-	-	-	-	-	-	-	-	-	23
RES 0,L,(IY+s)	FD CB ss 85	-	-	-	-	-	-	-	-	-	23
RES 1,L,(IY+s)	FD CB ss 8D	-	-	-	-	-	-	-	-	-	23
RES 2,L,(IY+s)	FD CB ss 95	-	-	-	-	-	-	-	-	-	23
RES 3,L,(IY+s)	FD CB ss 9D	-	-	-	-	-	-	-	-	-	23
RES 4,L,(IY+s)	FD CB ss A5	-	-	-	-	-	-	-	-	-	23
RES 5,L,(IY+s)	FD CB ss AD	-	-	-	-	-	-	-	-	-	23
RES 6,L,(IY+s)	FD CB ss B5	-	-	-	-	-	-	-	-	-	23
RES 7,L,(IY+s)	FD CB ss BD	-	-	-	-	-	-	-	-	-	23

Команды данной группы представляют из себя сложные команды сброса бита. Они загружают в указанный 8-битный регистр значение из памяти по указанному адресу, производят операцию сброса указанного бита в указанном 8-битном регистре в ноль, а затем сохраняют значение указанного 8-битного регистра в память по указанному адресу. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется.

114. Команды проверки бита внутри 8-битного регистра

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
-----------	--------------	---	---	---	---	---	-----	---	---	-------

BIT 0,A	CB 47	*	+	*	1	*	*	0	-	8
BIT 1,A	CB 4F	*	+	*	1	*	*	0	-	8
BIT 2,A	CB 57	*	+	*	1	*	*	0	-	8
BIT 3,A	CB 5F	*	+	*	1	*	*	0	-	8
BIT 4,A	CB 67	*	+	*	1	*	*	0	-	8
BIT 5,A	CB 6F	*	+	*	1	*	*	0	-	8
BIT 6,A	CB 77	*	+	*	1	*	*	0	-	8
BIT 7,A	CB 7F	*	+	*	1	*	*	0	-	8
BIT 0,B	CB 40	*	+	*	1	*	*	0	-	8
BIT 1,B	CB 48	*	+	*	1	*	*	0	-	8
BIT 2,B	CB 50	*	+	*	1	*	*	0	-	8
BIT 3,B	CB 58	*	+	*	1	*	*	0	-	8
BIT 4,B	CB 60	*	+	*	1	*	*	0	-	8
BIT 5,B	CB 68	*	+	*	1	*	*	0	-	8
BIT 6,B	CB 70	*	+	*	1	*	*	0	-	8
BIT 7,B	CB 78	*	+	*	1	*	*	0	-	8
BIT 0,C	CB 41	*	+	*	1	*	*	0	-	8
BIT 1,C	CB 49	*	+	*	1	*	*	0	-	8
BIT 2,C	CB 51	*	+	*	1	*	*	0	-	8
BIT 3,C	CB 59	*	+	*	1	*	*	0	-	8
BIT 4,C	CB 61	*	+	*	1	*	*	0	-	8
BIT 5,C	CB 69	*	+	*	1	*	*	0	-	8
BIT 6,C	CB 71	*	+	*	1	*	*	0	-	8
BIT 7,C	CB 79	*	+	*	1	*	*	0	-	8
BIT 0,D	CB 42	*	+	*	1	*	*	0	-	8
BIT 1,D	CB 4A	*	+	*	1	*	*	0	-	8
BIT 2,D	CB 52	*	+	*	1	*	*	0	-	8
BIT 3,D	CB 5A	*	+	*	1	*	*	0	-	8
BIT 4,D	CB 62	*	+	*	1	*	*	0	-	8
BIT 5,D	CB 6A	*	+	*	1	*	*	0	-	8
BIT 6,D	CB 72	*	+	*	1	*	*	0	-	8
BIT 7,D	CB 7A	*	+	*	1	*	*	0	-	8
BIT 0,E	CB 43	*	+	*	1	*	*	0	-	8
BIT 1,E	CB 4B	*	+	*	1	*	*	0	-	8
BIT 2,E	CB 53	*	+	*	1	*	*	0	-	8
BIT 3,E	CB 5B	*	+	*	1	*	*	0	-	8
BIT 4,E	CB 63	*	+	*	1	*	*	0	-	8
BIT 5,E	CB 6B	*	+	*	1	*	*	0	-	8
BIT 6,E	CB 73	*	+	*	1	*	*	0	-	8
BIT 7,E	CB 7B	*	+	*	1	*	*	0	-	8
BIT 0,H	CB 44	*	+	*	1	*	*	0	-	8
BIT 1,H	CB 4C	*	+	*	1	*	*	0	-	8
BIT 2,H	CB 54	*	+	*	1	*	*	0	-	8
BIT 3,H	CB 5C	*	+	*	1	*	*	0	-	8
BIT 4,H	CB 64	*	+	*	1	*	*	0	-	8
BIT 5,H	CB 6C	*	+	*	1	*	*	0	-	8
BIT 6,H	CB 74	*	+	*	1	*	*	0	-	8
BIT 7,H	CB 7C	*	+	*	1	*	*	0	-	8
BIT 0,L	CB 45	*	+	*	1	*	*	0	-	8
BIT 1,L	CB 4D	*	+	*	1	*	*	0	-	8
BIT 2,L	CB 55	*	+	*	1	*	*	0	-	8
BIT 3,L	CB 5D	*	+	*	1	*	*	0	-	8
BIT 4,L	CB 65	*	+	*	1	*	*	0	-	8
BIT 5,L	CB 6D	*	+	*	1	*	*	0	-	8
BIT 6,L	CB 75	*	+	*	1	*	*	0	-	8

BIT 7,L

CB 7D

*

+

*

1

*

*

0

-

8

Команды данной группы проверяют указанный бит в указанном 8-битном регистре в ноль. Они имеют достаточно сложное воздействие на флаги:

Флаг Z устанавливается в единицу, если значение проверяемого бита в указанном регистре равно 0, в противном случае флаг Z сбрасывается в ноль.

Флаг P/V в точности повторяет значение флага Z.

Флаг S устанавливается в единицу, если проверяемый бит – бит 7, и значение бита 7 в указанном регистре равно 1. Во всех остальных случаях флаг S сбрасывается в 0.

Флаг 5 устанавливается в единицу, если проверяемый бит – бит 5, и значение бита 5 в указанном регистре равно 1. Во всех остальных случаях флаг 5 сбрасывается в 0.

Флаг 3 устанавливается в единицу, если проверяемый бит – бит 3, и значение бита 3 в указанном регистре равно 1. Во всех остальных случаях флаг 3 сбрасывается в 0.

115. Команда проверки бита внутри 8-битного значения в памяти по адресу в регистре HL

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
BIT 0,(HL)	CB 46	*	+	*	1	*	*	0	-	12
BIT 1,(HL)	CB 4E	*	+	*	1	*	*	0	-	12
BIT 2,(HL)	CB 56	*	+	*	1	*	*	0	-	12
BIT 3,(HL)	CB 5E	*	+	*	1	*	*	0	-	12
BIT 4,(HL)	CB 66	*	+	*	1	*	*	0	-	12
BIT 5,(HL)	CB 6E	*	+	*	1	*	*	0	-	12
BIT 6,(HL)	CB 76	*	+	*	1	*	*	0	-	12
BIT 7,(HL)	CB 7E	*	+	*	1	*	*	0	-	12

Команды данной группы проверяют указанный бит в 8-битной ячейке памяти по адресу в регистровой паре HL. Они имеют достаточно сложное воздействие на флаги:

Флаг Z устанавливается в единицу, если значение проверяемого бита в ячейке памяти равно 0, в противном случае флаг Z сбрасывается в ноль.

Флаг P/V в точности повторяет значение флага Z.

Флаг S устанавливается в единицу, если проверяемый бит – бит 7, и значение бита 7 в ячейке памяти равно 1. Во всех остальных случаях флаг S сбрасывается в 0.

Флаги 5 и 3 копируются из битов 5 и 3 *скрытого временного регистра* микропроцессора. Подробнее об этом регистре можно прочитать в разделе «Особенности функционирования микропроцессора Z80».

116. Команды проверки бита внутри 8-битного значения в памяти по адресу в индексном регистре (со смещением)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
BIT 0,(IX+s)	*1 DD CB ss 46	*	+	*	1	*	*	0	-	20
BIT 1,(IX+s)	*1 DD CB ss 4E	*	+	*	1	*	*	0	-	20
BIT 2,(IX+s)	*1 DD CB ss 56	*	+	*	1	*	*	0	-	20
BIT 3,(IX+s)	*1 DD CB ss 5E	*	+	*	1	*	*	0	-	20
BIT 4,(IX+s)	*1 DD CB ss 66	*	+	*	1	*	*	0	-	20
BIT 5,(IX+s)	*1 DD CB ss 6E	*	+	*	1	*	*	0	-	20
BIT 6,(IX+s)	*1 DD CB ss 76	*	+	*	1	*	*	0	-	20
BIT 7,(IX+s)	*1 DD CB ss 7E	*	+	*	1	*	*	0	-	20
BIT 0,(IY+s)	*1 FD CB ss 46	*	+	*	1	*	*	0	-	20
BIT 1,(IY+s)	*1 FD CB ss 4E	*	+	*	1	*	*	0	-	20
BIT 2,(IY+s)	*1 FD CB ss 56	*	+	*	1	*	*	0	-	20
BIT 3,(IY+s)	*1 FD CB ss 5E	*	+	*	1	*	*	0	-	20

BIT 4,(IY+s)	*1	FD	CB	ss	66	*	+	*	1	*	*	0	-	20
BIT 5,(IY+s)	*1	FD	CB	ss	6E	*	+	*	1	*	*	0	-	20
BIT 6,(IY+s)	*1	FD	CB	ss	76	*	+	*	1	*	*	0	-	20
BIT 7,(IY+s)	*1	FD	CB	ss	7E	*	+	*	1	*	*	0	-	20

Команды данной группы проверяют указанный бит в 8-битной ячейке памяти по указанному адресу. Адрес хранится в одной из 16-битных индексных регистровых пар IX или IY, искомая регистровая пара указывается в команде. К адресу в регистровой паре прибавляется относительное смещение ss (от -128 до 127), содержимое самой регистровой пары не изменяется. Они имеют достаточно сложное воздействие на флаги:

Флаг Z устанавливается в единицу, если значение проверяемого бита в ячейке памяти равно 0, в противном случае флаг Z сбрасывается в ноль.

Флаг P/V в точности повторяет значение флага Z.

Флаг S устанавливается в единицу, если проверяемый бит – бит 7, и значение бита 7 в ячейке памяти равно 1. Во всех остальных случаях флаг S сбрасывается в 0.

Флаги 5 и 3 копируются из битов 5 и 3 *скрытого временного регистра* микропроцессора. Подробнее об этом регистре можно прочитать в разделе «Особенности функционирования микропроцессора Z80». Поскольку данные команды затрагивают этот регистр, можно считать, что флаги 5 и 3 в данном случае копируются из битов 5 и 3 старшего байта адреса искомой ячейки памяти.

- *1 Для данных команд приведен только основной код операции. На самом деле каждая из команд имеет еще 7 различных «недокументированных» кодов операции, выполняющих одну и ту же функцию. Все коды операции отличаются последним байтом. Для команд BIT 0 все коды операций – это 40, 41, 42, 43, 44, 45, 46, 47. Для команд BIT 1 – 48-4F, для BIT 2 – 50-5F, для BIT 3 – 58-5F, для BIT 4 – 60-6F, для BIT 5 – 68-6F, для BIT 6 – 70-7F и для BIT 7 – 78-7F.

117. Команды блочной пересылки данных

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
LDI	ED A0	-	-	*	0	*	*	0	-	16
LDD	ED A8	-	-	*	0	*	*	0	-	16
LDIR	ED B0	-	-	*	0	*	*	0	-	16/21
LDDR	ED B8	-	-	*	0	*	*	0	-	16/21

Команда LDI пересылает 1 байт из ячейки памяти по адресу в регистре HL в ячейку памяти по адресу в регистре DE. После пересылки значения регистров HL и DE увеличиваются на 1, значение регистра BC уменьшается на 1.

Команда LDD аналогична команде LDI, однако значение регистров HL и DE не увеличиваются, а уменьшаются на 1.

Команда LDIR представляет из себя команду LDI с зацикливанием. Если после однократного выполнения LDIR (аналогично LDI) регистр BC не равен 0, то исполнение продолжается с самой команды LDIR (т.е., происходит переход на адрес команды LDIR). При этом команда LDIR исполняется 21 такт. В противном случае (регистр BC равен 0) исполнение продолжается со следующей команды, а исполнение команды LDIR занимает 16 тактов.

Команда LDDR аналогична LDIR, однако представляет из себя зацикленную команду LDD, а не LDI.

Специальное воздействие команд на флаги следующее:

Флаг P/V устанавливается в 1, если регистр BC после выполнения команды не равен 0 и сбрасывается в 0 в противном случае.

Флаг 5 представляет собой бит 1 суммы пересылаемого командой значения и значения аккумулятора A.

Флаг 3 представляет собой бит 3 суммы пересылаемого командой значения и значения аккумулятора A.

118. Команды блочного поиска (сравнения)

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
CPI	ED A1	+	+	*	+	*	*	1	-	16
CPD	ED A9	+	+	*	+	*	*	1	-	16
CPIR	ED B1	+	+	*	+	*	*	1	-	16/21
CPDR	ED B9	+	+	*	+	*	*	1	-	16/21

Команда CPI сравнивает значение аккумулятора со значением из ячейки памяти по адресу в регистре HL и устанавливает флаги S, Z и H согласно результату операции сравнения. Флаг C при этом не изменяется. После сравнения значение регистра HL увеличивается на 1, значение регистра BC уменьшается на 1.

Команда CPD аналогична команде CPI, однако значение регистра HL не увеличивается, а уменьшается на 1.

Команда CPIR представляет из себя команду CPI с зацикливанием. Если после однократного выполнения CPIR (аналогично CPI) регистр BC не равен 0 и флаг Z не установлен, то исполнение продолжается с самой команды CPIR (т.е., происходит переход на адрес команды CPIR). При этом команда CPIR исполняется 21 такт. В противном случае (регистр BC равен 0 или установлен флаг Z) выполнение продолжается со следующей команды, а исполнение команды CPIR занимает 16 тактов.

Команда CPDR аналогична CPIR, однако представляет из себя зацикленную команду CPD, а не CPI.

Специальное воздействие команд на флаги следующее:

Флаг P/V устанавливается в 1, если регистр BC после выполнения команды не равен 0 и сбрасывается в 0 в противном случае.

Флаг 5 представляет собой бит 1 результата выражения, равного $A - x - H$, где A – значение аккумулятора A, x – значение последней сравниваемой ячейки памяти, H – значение флага N.

Флаг 3 представляет собой бит 3 результата того же выражения.

119. Команды ввода из порта ввода/вывода

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
IN A,(N)	DB NN	-	-	-	-	-	-	-	-	11
IN A,(C)	ED 78	+	+	5	0	3	P	0	-	12
IN B,(C)	ED 40	+	+	5	0	3	P	0	-	12
IN C,(C)	ED 48	+	+	5	0	3	P	0	-	12
IN D,(C)	ED 50	+	+	5	0	3	P	0	-	12
IN E,(C)	ED 58	+	+	5	0	3	P	0	-	12
IN H,(C)	ED 60	+	+	5	0	3	P	0	-	12
IN L,(C)	ED 68	+	+	5	0	3	P	0	-	12
INF	*1 ED 70	+	+	5	0	3	P	0	-	12

Команда IN A,(N) вводит значение из указанного порта ввода/вывода и сохраняет его в аккумуляторе. Данная команда не воздействует на флаги. Младший байт адреса порта ввода/вывода следует непосредственно за командой, старший байт адреса порта ввода/вывода берется из аккумулятора A.

Команды группы IN r,(C) вводят значение из указанного порта ввода/вывода и сохраняют его в указанном регистре. Адрес порта ввода/вывода берется из регистра BC. Эти команды устанавливают флаги согласно введенному из порта ввода/вывода значению.

***1** Команда INF представляет из себя команду IN (HL),(C). Альтернативные мнемоники – INZ; IN (C); IN 0,(C). Поскольку два одновременных цикла шины (ввод из порта и запись в память) невозможны, данная команда вводит значение из порта и устанавливает флаги, однако само введенное значение никуда не сохраняет. Функциональность данной команды на новых Z80-совместимых процессорах может отличаться, поэтому ее использование не рекомендуется.

120. Команды блочного ввода из порта ввода/вывода

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
INI	ED A2	*	*	*	*	*	*	*	*	16
IND	ED AA	*	*	*	*	*	*	*	*	16
INIR	ED B2	*	*	*	*	*	*	*	*	16/21
INDR	ED BA	*	*	*	*	*	*	*	*	16/21

Команда INI вводит значение из порта ввода/вывода с адресом в регистре BC и сохраняет введенное значение в ячейке памяти по адресу в регистре HL. После этого значение регистра HL увеличивается на 1, значение регистра B уменьшается на 1. Регистр C командой не изменяется.

Команда IND аналогична команде INI, однако значение регистра HL не увеличивается, а уменьшается на 1.

Команда INIR представляет из себя команду INI с закикливанием. Если после однократного выполнения INIR (аналогично INI) регистр B не равен 0, то исполнение продолжается с самой команды INIR (т.е., происходит переход на адрес команды INIR). При этом команда INIR исполняется 21 такт. В противном случае (регистр BC равен 0) выполнение продолжается со следующей команды, а исполнение команды INIR занимает 16 тактов.

Команда INDR аналогична INIR, однако представляет из себя закикленную команду IND, а не INI.

Команды блочного ввода/вывода имеют крайне сложное влияние на флаги микропроцессора. Более подробное влияние данных команд на флаги описано в разделе «Особенности функционирования микропроцессора Z80».

121. Команды вывода в порт ввода/вывода

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
OUT (N),A	D3 NN	-	-	-	-	-	-	-	-	11
OUT (C),A	ED 79	-	-	-	-	-	-	-	-	12
OUT (C),B	ED 41	-	-	-	-	-	-	-	-	12
OUT (C),C	ED 49	-	-	-	-	-	-	-	-	12
OUT (C),D	ED 51	-	-	-	-	-	-	-	-	12
OUT (C),E	ED 59	-	-	-	-	-	-	-	-	12
OUT (C),H	ED 61	-	-	-	-	-	-	-	-	12
OUT (C),L	ED 69	-	-	-	-	-	-	-	-	12
OUTF	*1 ED 71	-	-	-	-	-	-	-	-	12

Команда OUT (N),A выводит значение аккумулятора A в указанный порт ввода/вывода. Младший байт адреса порта ввода/вывода следует непосредственно за командой, старший байт адреса порта ввода/вывода берется из аккумулятора A.

Команды группы OUT (C),г выводят значение указанного регистра в указанный порт ввода/вывода. Адрес порта ввода/вывода берется из регистра BC.

***1** Команда OUTF представляет из себя команду OUT (C),(HL). Альтернативные мнемоники – OUTZ; OUT (C); OUT (C),0. Также известна мнемоника OUT (C),#FF (только для процессора Z84). Поскольку два одновременных цикла шины (чтение из памяти и вывод в порт) невозможны, данная команда выводит в порт значение 0. Функциональность данной команды на новых Z80-совместимых процессорах может отличаться, поэтому ее использование не

рекомендуется. К примеру, на микропроцессорах Z84 данная команда выводит в указанный порт значение #FF (255), а не 0.

122. Команды блочного вывода в порт ввода/вывода

Мнемоника		Код операции	S	Z	5	H	3	P/V	N	C	Такты
OTI	*1	ED A3	*	*	*	*	*	*	1	*	16
OTD	*1	ED AB	*	*	*	*	*	*	1	*	16
OTIR	*1	ED B3	*	*	*	*	*	*	1	*	16/21
OTDR	*1	ED BB	*	*	*	*	*	*	1	*	16/21

Команда OTI выводит значение из ячейки памяти по адресу в регистре HL в порт ввода/вывода с адресом в регистре BC. После этого значение регистра HL увеличивается на 1, значение регистра B уменьшается на 1. Регистр C командой не изменяется.

Команда OTD аналогична команде OTI, однако значение регистра HL не увеличивается, а уменьшается на 1.

Команда OTIR представляет из себя команду OTI с зацикливанием. Если после однократного выполнения OTIR (аналогично OTI) регистр B не равен 0, то исполнение продолжается с самой команды OTIR (т.е., происходит переход на адрес команды OTIR). При этом команда OTIR исполняется 21 такт. В противном случае (регистр BC равен 0) выполнение продолжается со следующей команды, а исполнение команды OTIR занимает 16 тактов.

Команда OTDR аналогична OTIR, однако представляет из себя зацикленную команду OTD, а не OTI.

Команды блочного ввода/вывода имеют крайне сложное влияние на флаги микропроцессора. Подробнее влияние данных команд на флаги описано в разделе «Особенности функционирования микропроцессора Z80».

*1 Данные команды имеют альтернативные мнемоники – OUTI, OUTD, OUTIR и OUTDR.

123. Команды управления прерываниями

Мнемоника		Код операции	S	Z	5	H	3	P/V	N	C	Такты
DI		F3	-	-	-	-	-	-	-	-	4
EI		FB	-	-	-	-	-	-	-	-	4
IM 0	*1	ED 46	-	-	-	-	-	-	-	-	8
IM 1	*1	ED 56	-	-	-	-	-	-	-	-	8
IM 2	*1	ED 5E	-	-	-	-	-	-	-	-	8
RETN	*1	ED 45	-	-	-	-	-	-	-	-	14
RETI	*1	ED 4D	-	-	-	-	-	-	-	-	14

Команда DI запрещает маскируемое прерывание INT (устанавливает внутренние флаги IFF1 и IFF2 в 0).

Команда EI разрешает маскируемое прерывание INT (устанавливает внутренние флаги IFF1 и IFF2 в 0).

Команды IM 0, IM 1 и IM 2 переключают режим обработки маскируемого прерывания INT.

Команды RETN и RETI выполняются аналогично команде безусловного возврата RET. Отличие заключается в цикле шины. Кроме того, команда RETN устанавливает значение флага IFF1 равным значению флага IFF2.

Подробнее о данной группе команд, а также флагах IFF1, IFF2 и обработке прерываний можно прочитать в разделе «Особенности функционирования микропроцессора Z80».

*1 Для команд IM 0, IM 1, IM 2, RETN и RETI указан только один из кодов операции. Дополнительные коды операции для данных команд отличаются последним байтом. Для IM 0

все коды операций – это 46, 4E, 66 и 6E, для IM 1 – 56 и 76, для IM 2 – 5E и 7E, для RETI – 4D, 5D, 6D и 7D, для RETN – 45, 55, 65 и 75. Все дополнительные коды операций для каждой команды выполняют аналогичную основному функцию.

124. Прочие команды

Мнемоника	Код операции	S	Z	5	H	3	P/V	N	C	Такты
NOP	*1 00	-	-	-	-	-	-	-	-	4
CPL	2F	-	-	5	1	3	-	1	-	4
NEG	ED 44	+	+	5	+	3	V	1	+	8
SCF	37	-	-	*	0	*	-	0	1	4
CCF	3F	-	-	*	*	*	-	0	*	4
HALT	76	-	-	-	-	-	-	-	-	*
DAA	27	*	*	*	*	*	*	-	*	4

Команда NOP не выполняет никаких операций. Она может использоваться для организации задержек или замещения ненужных команд.

Команда CPL инвертирует значения битов аккумулятора A. Флаги 5 и 3 устанавливаются согласно результату.

Команда NEG изменяет знак значения аккумулятора A на обратный (вычитает значение аккумулятора A из 0) и устанавливает флаги согласно результату.

Команда SCF устанавливает флаг переноса C в единицу, а также обнуляет флаги N и H. Флаги 5 и 3 устанавливаются согласно значению аккумулятора A.

Команда CCF инвертирует значение флага переноса C, а также обнуляет флаг N. Флаг N устанавливается в значение флага C до инвертирования. Флаги 5 и 3 устанавливаются согласно значению аккумулятора A.

Команда HALT эквивалентна выполнению команд NOP до момента появления сигнала прерывания INT или NMI. После каждого выполнения HALT (эквивалентно NOP), если аппаратного прерывания не было, то исполнение продолжается с самой команды HALT. В противном случае исполнение продолжается со следующей команды. Каждый шаг исполнения команды HALT занимает 4 такта.

Команда DAA производит коррекцию значения аккумулятора A после сложения или вычитания для двоично-десятичных чисел. Функционирование и воздействие данной команды на флаги достаточно сложное и подробно описывается в разделе «Особенности функционирования микропроцессора Z80».

*1 Все не вошедшие в данное описание команды не имеют никакой функции, т.е. исполняются аналогично команде NOP. Префиксы DD/FD/ED/CB перед командами исполняются 4 такта, например, не имеющая функции команда NOP с префиксом DD будет исполняться 8 тактов.