**ZDM/ZDMZ/ZDMH**

**Отладчик Z-80 и монитор**

**Руководство пользователя**

**Авторское право © 1985**

**RD SOFTWARE**

**1290 Monument Street**

**Pacific Palisades,CA 90272**

**Содержание**

1 Введение 2

2 Команды ZDM 3

1. Команда D (Просмотреть) 3

2. Команда DI (Запретить прерывания) 3

3. Команда EI (Включить прерывания) 3

4. Команда F (Заполнить) 3

5. Команда G (Выполнить) 3

6. Команда H (Шестнадцатеричная математика) 4

7. Команда I (Ввод) 4

8. Команда L (Перечислить) 4

9. Команда M (Переместить) 5

10. Команда QI (Запрос ввода) 5

11. Команда QO (Запрос вывода) 5

12. Команда R (Чтение) 5

13. Команда S (Установка) 6

14. Команда T (Трассировка) 6

15. Команда U (Отмена трассировки) 6

16. Команда X (Исследование) 6

17. Команда & (Альтернативные регистры) 7

18. Команда B (Поиск блока) 7

19. Команда V (Сравнение) 8

20. Команда P (Печать) 8

21. Команда J (Переход) 8

3 Процедура инсталляции 9

Приложение 1 Специальные замечания для ZDM/ZDMZ версии 3.2 10

Приложение 2 Мнемоники ZDM 11

# Введение

ZDM, ZDMZ и ZDMH - отладчики машинного языка Z-80 и мониторы, разработанные для использования в среде операционной системы CP/M[[1]](#footnote-1). Они распознают и отлаживают код процессоров 8080, Z80 и Z180 (ZDMH) несмотря на то, что выполняются только на машинах с процессорами Z80/Z180. Тип и структура команд почти идентичны утилите DDT, поставляемой Digital Research в составе их операционной системы CP/M. Основное различие заключается в том, что можно отлаживать код Z-80 и были добавлены 10 новых команд с изменениями. ZDM/ZDMZ/ZDMH поддерживает все команды DDT, кроме команды A, встроенного ассемблерного кода. Кроме того, реализована только одна контрольная точка.

ZDMZ выводит на экран мнемоники Zilog Z-80. ZDMH также включает добавленные мнемоники Z180/HD64180. ZDM использует расширенные мнемоники Intel 8080, подобные TDL (Xitan), которые сохраняют все стандартные мнемоники 8080. См. [Приложение 2](#Приложение_2) для ознакомления с мнемониками, используемых ZDM. Обратитесь также к [Разделу 3](#_Процедура_инсталляции) *Процедура установки* прежде, чем попытаться выполнить ZDM или ZDMZ.

В оставшейся части данного руководства все ссылки на ZDM применяются в равной мере к ZDMZ и ZDMH если, не оговорено иначе.

ZDM вызвается, вводом в консоли одной из следующих трех форм:

**ZDM**

**ZDM filename**

**ZDM filename.filetype**

где "filename" - имя загружаемого файла. Затем ZDM оповестит о начале работы и переместит себя на место CCP и будет располагаться непосредственно под BDOS. Переход к BDOS в расположении 5 изменяется на начальный адрес размещения ZDM, который, в свою очередь, содержит переход к BDOS. Обратите внимание, что ZDM обеспечивает дополнительную страницу доступной переходной памяти по сравнению с DDT для данного размера системы CP/M. Как и DDT, транзитные программы, загруженные для отладки, могут перезаписать модуль дизассемблера. В этом случае команда L отключена и поле инструкций для команд T и X заменяется на соответствующие шестнадцатеричные байты инструкций.

У ZDM есть дополнительная функция, предотвращающая накладывание самого ядра ZDM. Если это должно произойет, то ZDM прервет загрузку и распечатает сообщение "OUT OF MEMORY" (недостаточно памяти). Затем он вернется в командный уровень, для возможности исследования загруженных частей.

После сообщения о начале работы и загрузки программы, если она определена в командной строке, ZDM выведет символ приглащения "-" и будет ожидать входных команд. Каждая команда состоит из одного или двух символов, согласно опописанию в [Разделе 2](#_Команды_ZDM), которые определяют тип команды. Эти символы могут также сопровождаться дополнительными параметрами. Разделитель не должен использоваться между символами типа команды и первым параметром за исключением описанного для команды G. Последующие параметры разделяются запятой или одинарным пробелом. Во всех случаях, если команда ожидает заключительный параметр и этот параметр опущен, ZDM предполагает, что он равен нулю.

Все командные строки завершаются возвратом каретки. Весь ввод с клавиатуры в ZDM и вывод от ZDM находятся в шестнадцатеричном формате. ZDM примет прописные или строчные буквы. Будет распечатан единственный символ (?), если произошла ошибка. Чтобы выйти из ZDM и возвратиться к уровню команд CP/M могут быть выполнены Сtrl-C или G0 (переход к 0).

# Команды ZDM

Детали каждой команды приведены ниже.

## 1. Команда D (Просмотреть)

Команда D разрешает оператору просматривать содержание памяти в ASCII и шестнадцатеричном виде. Формы:

**D**

**Ds**

**Ds,f**

В первом случае память вывводится на экран начиная с текущего адреса отображения (первоначально 100H) и продолжается для числа строк, определенных при инициализации. Последующие адреса дисплея инициализируются значением счетчика команд после команд X, T, U или G.

Вторая форма команды D похожа на первую, за исключением того, что первому отображаемому адресу присваивается адрес **s**. Третья форма осуществляет вывод на экран начиная с адреса **s** до **f**. Во всех случаях последующий вывод первой формы начнется с адреса отображения после последнего выведенного на экран адреса, приводя к продолжению отбражения. Длинные выводы могут быть прерваны с помощью клавиши стирания.

## 2. Команда DI (Запретить прерывания)

Эта команда принимает единственную форму:

**DI**

По умолчанию, при вводе команд G, T или U, прерывания разрешены. (Прерывания всегда запрещены при возврате в ZDM). Команда DI переопределяет это условие по умолчанию. Команда DI останется в силе, до задания последующий команды EI. (Прерывания всегда включены при возврате в ZDM).

## 3. Команда EI (Включить прерывания)

Эта команда восстанавливает условие прерывания по умолчанию. См. описание команды DI. Она принимает единственную форму:

**EI**

## 4. Команда F (Заполнить)

Команда F принимает форму:

**Fs,f,c**

где **s** - начальный адрес, **f** - заключительный адрес и **c** - байт шестнадцатеричной константы. Если c опущен то, он предполагается равным нулю. Эта команда заполняет блок памяти начиная с **s** до **f** включительно константой **c**. Если **f** будет меньше, чем **s**, то произойдет ошибка (?).

## 5. Команда G (Выполнить)

Запуск выполнения программы, используя команду G с одним дополнительным адресом контрольной точки. Команда G принимает четыре возможные формы:

**G**

**Gs**

**Gs,b**

**G,b**

Первая форма запускает выполнение целевой программы с текущего адреса счетчика команд и в текущем состоянии машины без установки контрольных точек. Единственный путь ZDM, для восстановления управление является выполнение RST 7. Вторая форма подобна первой за исключением того, что счетчику команд сначала присваивается значение **s**, до начала выполнение. Третья форма совпадает со второй за исключением того, что в адресе **b** установливается контрольная точка. Выполнение программы остановливается и управление возвращается к ZDM. При достижении контрольной точки инструкции по адресу **b** не выполняется. Четвертая форма запускает выполнение с текущего счетчика команд и состояния машины и устанавливает контрольную точку в адресе **b**.

После ввода контрольной точки ZDM выводит \*d, где d - адрес остановки. В этой точке может быть исследовано состояние машины, используя команду X или &X.

## 6. Команда H (Шестнадцатеричная математика)

Команда H принимает форму:

**Ha,b**

где **a** и **b** - шестнадцатеричные константы от 1 до 4 цифр. Сумма a+b и разница a-b выводятся на экран в шестнадцатеричном в форме:

**a+b a-b**

## 7. Команда I (Ввод)

Команда I позволяет оператору вставлять имя файла в блок управления файлом CP/M по умолчанию, расположенный начиная с адреса 5CH. FCB по умолчанию может использоваться программой при тестировании, как будто он было передан процессором консольных команд CP/M. Эта команда должна также использоваться до команды R при чтении дополнительных файлов HEX или COM. Формат команды I следующий:

**Ifilename**

**Ifilename.filetype**

Если расширение файла будет чем-нибудь кроме HEX, то ZDM предположит, что это файл COM, и команда R считает его в память, начиная с адреса 100H. (См. команду R для получения более подробной информации).

## 8. Команда L (Перечислить)

Команда L используется, для перечисления мнемоник ассемблера. Существует три формы этой команды:

**L**

**Ls**

**Ls,f**

Первая форма отображает количество строк, указанных в списке инициализации, начиная с текущего адреса счетчика команд. Вторая форма перечисляет то же число строк, но начинается с адреса **s**. Третья форма начинается с адреса **s** и продолжается для **f** строк. Все три формы могут продолжиться последующей командой L, аналогичной команде D. Кроме того, как и в команде D, если не определен начальный адрес, всегда инициализируется значением счетчика команд после команд X, T, U или G. Длинные выводы могут быть прерваны с помощью клавиши стирания.

## 9. Команда M (Переместить)

Команда M переместит блок памяти от одного расположения до другого. Форма команды M следущая:

**Ms,f,d**

где **s** - начальный адрес перемещения, **f** - конечный адрес, и **d** - адрес назначения. Если **f** будет меньше, чем **s**, то произойдет ошибка (?).

## 10. Команда QI (Запрос ввода)

Команда QI позволяет оператору читать входной порт и отображать значение порта по адресу. Форма этой команды:

**QIa**

где **a** однобайтовый адрес порта в шестнадцатеричном виде. Сразу после выполнения этой команды распечатывается значение. Обратите внимание, если **a** отсутствует, его значение равняется нулю.

## 11. Команда QO (Запрос вывода)

Команда QО позволяет оператору выводить указанный байт на указанный адрес порта. Форма этой команды:

**QOa,b**

где **a** является адресом порта, а **b** это байт для вывода. Если **a** или **b** опущен, он принимается равным нулю.

## 12. Команда R (Чтение)

Команда R используется совместно с командой I для чтения COM и HEX файлов с диска в память. Существуют две формы этой команды:

**R**

**Rb**

где **b** - дополнительное смещение адреса, которое добавляется к каждой программе или адресу данных, по мере загруки. Если **b** не указан, то он принимается равным нулю. Обратите внимание, если имя файла в FCB от предыдущей команды I имеет расширение отличное от HEX то, ZDM предполагает, что оно - COM и загрузит его начиная с адреса 100H или 100H+b, если параметр **b** включен. Если файл не может быть открыт, или происходит ошибка при чтении, ZDM отвечает индикатором ошибки (?). В противном случае при завершении загрузки выводится сообщение:

**NEXT PC**

**nnnn pppp**

где **nnnn** - следующий адрес после только что загруженной программы, и **pppp** - первый адрес только что загруженной программы. Для HEX файлов pppp берется из последней записи HEX файла и будет равен нулю, если оператор END, сопровождаемый начальным адресом, не был включен в исходную программу до ассемблирования.

## 13. Команда S (Установка)

Команда S позволяет проанализировать ячейки памяти и при необходимости изменить. Форма команды следующая:

**Ss**

где **s** - шестнадцатеричный начальный адрес для исследования или изменения памяти. ZDM распечатает адрес, сопровождаемый байтом, сохраненным по этому адресу. Возврат каретки будет осуществлять переход к следующему адресу, выводя на экран следующий адрес и следующий байт. Если будет введено новое значение байта сопровождаемое возвратом каретки, то это новое значение будет сохранено по этому адресу, и ZDM автоматически перейдет к следующему адресу. Для завершения команды, вводится точка вместо значения байта. Команда также завершится, если будет введено недопустимое шестнадцатеричное значение.

## 14. Команда T (Трассировка)

Команда T разрешает пошаговое выполнение трассировки инструкций программы для 1 - 65535 шагов. Формы этой команды следующие:

**T**

**Tn**

где **n** - необязательное число шагов. Первая форма предполагает, что n равняется еденице. Состояние процессора выводится на экран, и выполняется следующий шаг программы. Адрес завершения выводится на экран в виде \*hhhh, где hhhh - следующий адрес, который будет выполнен. Формат отображения состояния процессора в остальном идентичен выводу по команде X.

Трассировка программы прекращеется в интерфейсе для CP/M и возобновляется снова после возврата из CP/M к целевой программе. Длинная трассировка команды Tn может быть остановлена клавишей удаления. ZDM будет продолжать трассировку от этой остановки, если будет задана другая команда T или Tn.

## 15. Команда U (Отмена трассировки)

Команда U идентична команде T за исключением того, что состояние процессора не выводится на экран. Формы команды следующие:

**U**

**Un**

Все условия команды T применяются к команде U. Последнее состояние процессора выводится на экран после выполнения команды U или Un.

## 16. Команда X (Исследование)

(См. также команду &) Команда X позволяет выборочное отображение и изменение текущего состояния процессора в любое время. Формы:

**X**

**Xr**

где **r** - любой из регистров или флагов Z-80.

**C** Флаг переноса (0/1)

**Z** Флаг нуля (0/1)

**M** Флаг минус (0/1)

**E** Флаг четности (0/1)

**I** Внутренний перенос (0/1)

**A** Аккумулятор (0-FF)

**B** Регистроаая пара BC (0-FFFF)

**D** Регистроаая пара DE (0-FFFF)

**H** Регистроаая пара HL (0-FFFF)

**S** Указатель стека (0-FFFF)

**P** программный счетчик (0-FFFF)

**X** Индексный регистр X (0-FFFF)

**Y** Индексный регистр Y (0-FFFF)

В первом случае состояние регистра процессора выводится на экран в формате

**CfZfMfEfIf A=bb B=dddd D=dddd H=dddd P=dddd S=dddd**

**X=dddd Y=dddd instruction**

где **f** - 0 или 1 значение флага, **bb** - значение байта, и **dddd** - двойной байт, соответствующий регистровой паре. Поле инструкции содержит дизассемблированную инструкцию в расположении, адресуемом счетчиком команд.

Вторая форма команды X позволяет отображать и при необходимости изменять значение регистра или флага определеные r. Если после команды Xr введен возврат каретки, то команда завершается без сохраненя изменений. В противном случае ZDM принимает ввод для изменения флага или регистра. Если введено шестнадцатеричное число в надлежащем диапазоне то, что флаг или регистр принимает соответствующее знначение.

## 17. Команда & (Альтернативные регистры)

Команда & принимает одну из трех форм:

**&**

**&X**

**&Xr**

Первая форма безусловно обменивает содержимое всех регистров процессора и флагов с набором альтернативных регистров Z-80.

Вторые и третьи формы этой команды идентичны командам X и Xr за исключением того, что операции имеют место с альтернативным набором регистров или флагов. После завершения этих последних двух форм процессор востанавливается в состояние до выполнения команды. Отображеник, связанное с командой &X, заменяет регистры X и Y на значение регистра вектора прерываний V. Это значение регистра может также быть изменено командой &XV, за которой располагается значение байта, которое будет сохранено в регистре вектора прерываний. Регистры A, B, D и H и регистровые пары каждый раз помечаются символом (') , когда альтернативный набор выводится на экран или изменяется. Обратите внимание, что символ (') не используются для регистра флагов за исключением момента изменения.

## 18. Команда B (Поиск блока)

Команда B позволяет пользователю искать в памяти все вхождения строки байтов. Строки ограничены десятью байтами. Вторая форма этой команды использует BT, а не B. Эта вторая форма принимает ASCII или текстовую строку. Форма этой команды:

**Bs,f or BTs,f**

где **s** - начальный адрес, и **f** - конечный адрес блока памяти для поиска. После ввода возврата каретки вам будет предложено ввести строку. Форма B ожидает, что эта строка представлена в виде последовательность шестнадцатиричных байтов. Разделитель может быть пробелом или запятой. Ворма BT ожидает единственную строку ASCII. Вводимая строка завершается возвратом каретки после последнего ввода. Начальный адрес каждого вхождения строки от **s** до **f** будет выведен на экран.

## 19. Команда V (Сравнение)

Команд V проверят, идентичны ли два блока памяти. Форма этой команды:

**Vs,f,b**

где **s** - начальный адрес, **f** - конечный адрес первого блока и **b** - начальный адрес второго блока. Если совпадение не найдено, то печатается адрес с следующим за ним байтом соответствующего адреса во втором блоке.

## 20. Команда P (Печать)

Команда P является переключателем, не требующим параметров. Эффект состоит в том, чтобы отправить весь вывод на устройство печати, а также на консоль. Это дейчтвие отключается последующей командой P. Каждый раз, когда переключатель P находится в состоянии on, символ 'P' будет выведен на экран как часть отображения T или X.

## 21. Команда J (Переход)

Команда J является переключателем, который не ожидает параметров. Она влияет только на последующие команды T или U. Если была выполнена команда J то, команда T выведет на экран только условное выражение и безусловные вызовы, переходы, возвраты, рестарты, PCHL (IX или IY) и относительные переходы. Форма Tn команды T обычно используется, когда n представляет фактическое число отслеживаемых инструкций. Как обычно, эта команда может быть прервана клавишей удаления. Каждый раз, когда переключатель J в состоянии on, на экран будет выведен символ 'J' как часть отображения T или X.

# Процедура инсталляции

Пользователь должен сначала сделать копию ZDM и сохранить исходный оригинал в качестве резервной копии. Не пишите на оригинале. Используя копию введите "ZDM". ZDM ответит вопросом, "What is your terminal width (in hex)" (Что является вашей терминальной шириной). Возврат каретки в ответ на этот вопрос примет значение ширины по умолчанию равное 80(50H). Иначе, сначала введите ширину символов вашего терминала в шестнадцатеричном виде, сопровождаемое возвратом каретки. Затем, запрашивается желаемое число строк, для отображения команд D и L. Возврат каретки в этой точке присваивает значение по умолчанию равным 21(15H) строк. (Это - рекомендуемый размер для 24-строчных терминалов.) Иначе, введите желаемое число строк, в шестнадцатеричном виде, сопровождаемой возвратом каретки.

Наконец, ZDM задаст вопрос, "Is this correct? (Y or N)" (Это корректно? Y или N). В этот момент не отвечайте "Y". После ответа "N" ZDM сигнализирует о начале работы, распечатает символ приглашения "-" и будет ждать команду. Теперь вы должны протестировать команды D и L, чтобы определить, правильно ли определены эти значения для вашего терминала. В противном случае вернитесь на уровень команд CP/M, введя G0 или Сtrl-C и повторите вышеупомянутую процедуру. Когда вы удостоверились, что значения определены корректно, на вопрос, "Is this correct? (Y or N)" ответьте "Y". После ответа "Y" ZDM автоматически создаст пользовательский файл установки. Если вы будете использовать ZDM, то установленным именем файла будет ZDI.COM. Если вы используете ZDMZ, то имя установленного файла будет ZDI.COM. Если вы используете ZDMZ то устанавливается имя файла ZDIZ.COM. Вы можете переименовать их как угодно при желании.

ZDM теперь корректно настроен для вашего терминала. Последующие вызовы ZDM дудет начнаться непосредственно с сообщению входа в программу. Если в последующем вы захотите изменить эти параметры экрана, вы должны будете еще раз запустить его с копии основного диска.

Приложение 1 Специальные замечания для ZDM/ZDMZ версии 3.2

Некоторые наши клиенты испытали трудности с ZDM/ZDMZ при использовании систем управляемых прерываниями. Следующие патчи рекомендуются при выполнении ZDM/ZDMZ в таких системах.

Используя DDT или эквивалентную программу, загрузите образ ZDM/ZDMZ в память, начиная с 100H, и измените следующие байты:

|  |  |  |  |
| --- | --- | --- | --- |
| **Адрес** | **Адрес** |  |  |
| **ZDM** | **ZDMZ** | **с** | **на** |
| 0C61H | 0C31H | 0F3H | 00H |
| 0C7AH | 0C4AH | 0F3H | 00H |
| 0DC1H | 0D91H | 0F3H | 00H |

Вернитесь в CP/M не нарушая образ памяти и сохраните 22 страницы с помощью команды CP/M SAVE.

Другая часть клиента просит изменить ZDM/ZDMZ адрес рестарта, который в настоящее время RST 7 по адресу 38H, на другой адрес рестарта. Следующие патчи будут учитывать это изменение. (Помните, что RST 0 не может использоваться с CP/M).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Адрес** | **Адрес** | **Адрес** |  |  |
| **ZDM** | **ZDMZ** | **ZDMH** | **с** | **на** |
| 0D9DH | 0D6DH | 0DEBH | 38H | Новый адрес RST |
| 0DA5H | 0D75H | 0DF3H | 39H | Новый адрес+1 RST |
| 1007H | 0FD7H | 1054H | 0FFH | Новый код RST |

Приложение 2 Мнемоники ZDM

Модуль дизассемблера ZDM использует список мнемоник, который подобен мнемоникам Technical Design Laboratories (TDL). Все мнемоники Intel 8080 сохранены. Специфические инструкции Z-80 отличаются от мнемоники Zilog как показано в ниже следующей таблице. Перечень мнемоник ZDM практически идентичен реализованным Digital Research в Z80.LIB, которая используется с их CP/M макроассемблером "MAC". В таблице используются следующие соглашения.

**r** – Любые регистры или память

**rr** – Любые регистровые пары или указатель стека

**nn** - Непосредственные 8 разрядные данные (от 0 до 255)

**d** - 8 разрядные смещения со знаком (от -128 до 127)

**nnnn** - 16 разрядные адреса или непосредственные данные (от 0 до 65535)

**b** – Номер бита (от 0 до 7, 7 является старшим битом)

**addr** - 16 разрядные адрес от PC+127 до PC-128

Обратите внимание, что во всех случаях, включающих смещение **d**, этот параметр всегда является последним в поле операнда.

**ZDM ZILOG ZDM ZILOG**

LDX r,d LD r,(IX+d) LDIR LDIR

LDY r,d LD r,(IY+d) LDD LDD

STX r,d LD (IX+d),r LDDR LDDR

STY r,d LD (IY+d),r CCI CPI

MVIX nn,d LD (IX+d),nn CCIR CPIR

MVIY nn,d LD (IY+d),nn CCD CPD

LDAI LD A,I CCDR CPDR

LDAR LD A,R ADDX d ADD (IX+d)

STAI LD I,A ADDY d ADD (IY+d)

STAR LD R,A ADCX d ADC (IX+d)

LXIX nnnn LD IX,nnnn ADCY d ADC (IY+d)

LXIY nnnn LD IY,nnnn SUBX d SUB (IX+d)

LBCD nnnn LD BC,(nnnn) SUBY d SUB (IY+d)

LDED nnnn LD DE,(nnnn) SBBX d SBC (IX+d)

LSPD nnnn LD SP,(nnnn) SBBY d SBC (IY+d)

LIXD nnnn LD IX,(nnnn) ANAX d AND (IX+d)

LIYD nnnn LD IY,(nnnn) ANAY d AND (IY+d)

SBCD nnnn LD (nnnn),BC XRAX d XOR (IX+d)

SDED nnnn LD (nnnn),DE XRAY d XOR (IY+d)

SSPD nnnn LD (nnnn),SP ORAX d OR (IX+d)

SIXD nnnn LD (nnnn),IX ORAY d OR (IY+d)

SIYD nnnn LD (nnnn),IY CMPX d CP (IX+d)

SPIX LD SP,IX CMPY d CP (IY+d)

SPIY LD SP,IY INRX d INC (IX+d)

PUSHIX PUSH IX INRY d INC (IY+d)

PUSHIY PUSH IY DCRX d DEC (IX+d)

POPIX POP IX DCRY d DEC (IY+d)

POPIY POP IY NEG NEG

EXAF EX AF,AF' IM0 IM 0

EXX EXX IM1 IM 1

**ZDM ZILOG ZDM ZILOG**

XTIX EX (SP),IX IM2 IM 2

XTIY EX (SP),IY DADC rr ADC HL,rr

LDI LDI DSBC rr SBC HL,rr

DADX rr ADD IX,rr OUTI OUTI

DADY rr ADD IY,rr OUTIR OTIR

INXIX INC IX IND IND

INXIY INC IY INDR INDR

DCXIX DEC IX OUTD OUTD

DCXIY DEC IY OUTDR OTDR

BIT b,r BIT b,r RLCR r RLC r

SET b,r SET b,r RLCX d RLC (IX+d)

RES b,r RES b,r RLCY d RLC (IY+d)

BITX b,d BIT b,(IX+d) RALR r RL r

BITY b,d BIT b,(IY+d) RALX d RL (IX+d)

SETX b,d SET b,(IX+d) RALY d RL (IY+d)

SETY b,d SET b,(IY+d) RRCR r RRC r

RESX b,d RES b,(IX+d) RRCX d RRC (IX+d)

RESY b,d RES b,(IY+d) RRCY d RRC (IY+d)

JR addr JR addr RARR r RR r

JRC addr JR C, addr RARX d RR (IX+d)

JRNC addr JR NC,addr RARY d RR (IY+d)

JRZ addr JR Z, addr SLAR r SLA r

JRNZ addr JR NZ,addr SLAX d SLA (IX+d)

DJNZ addr DJNZ, addr SLAY d SLA (IY+d)

PCIX JP (IX) SRAR r SRA r

PCIY JP (IY) SRAX d SRA (IX+d)

RETI RETI SRAY d SRA (IY+d)

RETN RETN SRLR r SRL r

INP r IN r,(C) SRLX d SRL (IX+d)

OUTP r OUT (C),r SRLY d SRL (IY+d)

INI INI RLD RLD

INIR INIR RRD RRD

1. CP/M - зарегистрированная торговая марка Digital Research [↑](#footnote-ref-1)