**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ПЭВМ**

**Динамический отладчик DDT и символический отладчик ZSID**

**(руководство оператора)**

**Введение**

Программа DDT представляет возможность динамического интерактивного контроля и отладки программ в среде CP/M. Отладчик инициируется введением следующих команд:

DDT

DDT имя файла,HEX

DDT имя файла,COM

Где имя файла - имя программы, которая должна быть загружена и проверена. Во всех случаях программа DDT загружается в оперативную память на место CCP и располагается непосредственно над BDOS.

Вторая и третья формы команды DDT, показанные выше, выполняют те же действия, что и первая, но дополнительно автоматически загружают определенные COM или HEX файлы. Их действие идентично последовательности команд:

DDT

I - имя файла .HEX или имя файла .COM

Где I и R определяют имя файла и загружают его в память (более подробно эти команды описаны ниже).

После инициализации DDT выводит на консоль сообщение:

NNK DDT-S VER M.M.

Где:

NN - размер памяти;

S - тип механических средств;

D - DIGITAL RESEARCH стандартная версия;

I - IMSA версии;

O - OMRON - система;

S - DIGITAL SYSTEM стандартная версия

M.M - номер версии.

После вывода сообщения DDT выводит символ "-" и ожидает ввода команды с консоли. Для выполнения команды оператор должен вести и закончить возвратом каретки (<CR>). Каждая введенная команда может быть отредактирована стандартными управляющими клавишами:

RUBOUT - удалить последний введенный символ;

CTL-U - удалить введенное предложение;

CTL-C - перезагрузка системы.

Любая команда может иметь не более 32 символов (вместо 33-го символа автоматически вводится возврат каретки). Первый символ определяет тип команды:

A - запись памяти с использованием мнемокода;

D - вывод на консоль содержимого памяти в 16-пичном коде и ASCII,

F - заполнение памяти константой,

G - начать выполнение и определить контрольные точки,

I - задать имя файла для загрузки,

L - вывести содержимое памяти в мнемокоде,

M - пересылка блоков из одной области памяти в другую,

R - считать программу с диска для последующего выполнения,

S - изменить содержимое памяти,

T - трассировка программы,

U - выполнение N команд без трассировки,

X - выполнение и изменение состояния ЦП.

За символом команды может следовать до трех шестнадцатеричных параметров, определяемых запятыми или пробелами. Все значения DDT выводит в шестнадцатеричной форме. Во всех случаях команда не начинает выполняться, пока не будет введен возврат каретки.

В любое время процесса отладки оператор может остановить выполнение DDT, нажав CTL-C или введя команду GO (переход в ячейку 0000H) и записать текущее состояние памяти на диск, используя команду SAVE:

SAVE N имя файла COM

где N количество 256 байтных блоков, которые будут записаны на диск.

Количество блоков может быть определено из старшего байта последовательного адреса программы, после его преобразования из 16-ричного в десятичный вид. Например, если старший адрес программы 1234H, то старший байт 12H или 18 в десятичной форме и после нажатия CTRL-C оператор может записать программу на диск, задавая

SAVE 18 X.COM

Таким образом содержимое памяти, начиная с 100H, запишется на диск и может быть загружено в память при вводе имени X. Если после загрузки будет осуществляться проверка и отладка, тогда следует загружать, используя команду

DDT X.COM

которая загружает ранее записанную на диск программу, начиная с адреса 100H до 12F (18 страниц). Состояние машины этой программой не изменяется и такая программа может быть запущена сначала в целях проверки.

**2. Команды DDT**

Ниже детально описаны команды DDT. В каждом случае оператор перед вводом команды должен ожидать вывода на консоль символа "-". Если команда выполняется в режиме тестирования, но не достигает контрольной точки, управление можно возвратить DDT нажатием клавиши RS17 с лицевой панели (клавиша возврат на шаг может использоваться при выполнении программ по командам U и T). В описаниях каждой команды после ее обозначения стоят строчные символы, которые разделяются запятыми. Эти символы представляют 16-ричное число не более 4 знаков в длину (из более длинных выбираются 4 правых знака).

Несколько команд управляет состоянием ЦП во время тестирования программы. Перед началом отладки все регистры устанавливаются в начальный адрес, приведенный в последней записи файла .HEX, если загружается файл этой формы (см. команды I и R).

**1) Команда А (ASSEMBLE)**

DDT позволяет заносить в память данные с использованием мнемокода с помощью команды A, имеющей форму:

AS,

где S - начальный адрес занесения в память.

DDT - выводит на консоль адрес следующей команды и считывает с консоли мнемокод, преобразует его в двоичную форму и вводит в память. После этого на консоль выводится следующий адрес для записи. Команда А завершается при вводе первого пустого предложения с консоли.

После завершения ввода, оператор может проверить содержимое памяти, используя команду L (дизассемблер DDT).

Отметим, что сегмент АССЕМБЛЕРА (дизассемблера DDT) может быть испорчен загружаемой для отладки программой. В этом случае при введении команд A и L выдаются сообщение об ошибке (см. раздел 4).

**2) Команда D (DISPLAY)**

Эта команда выводит на консоль содержимое памяти в 16-ричном коде виде и в кодах ASCII.

Формат команд:

D

DS

DS,F

В первом случае содержимое отображается с текущего адреса отображения (инициализируется 100H) и выводится 16 предложений в виде:

AAAA BB...BB(16БАЙТ) CCCCCCCCCCCCCCCC

Где : AAAA - адрес отображения в 16-ричной форме;

BB - содержимое памяти в 16-ричной форме, начиная с адреса AAAA;

CC...- содержимое памяти в кодах ASCII. На месте содержимого не соответствующего кодам ASCII, отображаются ".". Заметим, что на консоль выводятся символы в верхнем регистре независимо от того, в каком регистре они находятся в памяти.

В каждой строке отображаются значения 16 ячеек памяти, и на консоль выводится 16 строк, отображающих содержимое 256 ячеек памяти.

Вторая форма подобна первой, но отображение начинается с адреса S. При вводе третьей формы команды отображается содержимое ячеек с адреса S по адрес F.

Во всех случаях адрес отображения после вывода на консоль устанавливается на следующую не отображенную ячейку, и дальнейший просмотр может осуществляться введением только D.

**3) Команда F (FILL)**

Эта команда имеет вид:

FS,F,C.

Где: S - начальный адрес;

F - конечный адрес;

C - 16-ричная константа.

Действие команды следующее: DDT записывает в адрес S константу C, увеличивая S на 1 и сравнивает S и F. Если S и F равны, действие завершается, если нет, повторяется. Таким образом команда F позволяет загрузить область памяти константами.

**4) Команда G(GO)**

Запуск выполнения программы осуществляется командой G, позволяющей вводить до двух контрольных точек. Команда G имеет следующие формы:

G

GS

GS,B

GS,B,C

G,B

G,B,S

Первая форма команды запускает выполнение программы без установки контрольных точек (возврат в монитор в этом случае может осуществиться нажатием RST7). Текущее значение программного счетчика может быть проверено с использованием команд X или XP. Вторая форма аналогична первой, за исключением того, что перед запуском программный счетчик устанавливается в S. Третья форма аналогична второй, но выполнение программы останавливается при достижении адреса B (контрольная точка). Команда в ячейке В не выполняется, если в этом месте установлена контрольная точка. Четвертая форма идентична третьей, но устанавливает две контрольные точки - B и C. После достижения этих точек, они очищаются. Последние две формы запускают программу с текущего программного счетчика с установкой (одной или двух контрольных точек, соответственно).

Выполнение начинается со стандартного адреса в реальном времени до следующей контрольной точки, т.е. от стандартного адреса до контрольной точки DDT не вмещается в программу. Таким образом, если после запуска программы, она не достигает контрольной точки, управление может передаваться DDT только по RST7. После достижения контрольной точки DDT останавливает выполнение и выводит на консоль.

$D

Где D - адрес останова.

В этой точке может быть проверено состояние машины командой X. Оператор должен устанавливать контрольные точки, отличающиеся от значения программного счетчика, иначе останов происходит сразу, без выполнения каких-либо команд.

Пример таких остановов:

G 400, 400 или G,1234 если текущее значение PC=1234H

**5) Команда I (INPUT)**

Команда I позволяет оператору ввести имя файла в блок управления файлом в ячейку 5CH (блок управления файлом создается CP/M для транзитных команд и помещается в ячейку см. интерфейс CM/M). Этот блок может быть использован проверяемой программой, как если бы он был послан CCP. Кроме того, это имя используется при чтении DDT файла, .HEX или .COM для отладки.

Форма команды:

I имя файла

I имя файла расширение

Если во второй форме используются расширения .COM или .HEX, тогда для чтения с диска машинных кодов может использоваться команда R.

**6) Команда L (LIST)**

Команда L выводит содержимое памяти с использованием мнемоники АССЕМБЛЕРА.

Формы команды:

L

LS

LS,F

Первая форма выдает на консоль двенадцать предложений в мнемокоде, начиная с текущего адреса L. Вторая форма сначала устанавливает адрес, а затем выводит 12 предложений. Третья выводит с адреса S по адрес F. Во всех трех случаях адрес L после окончания вывода устанавливается на первый невыведеный адрес. После достижения контрольных точек, адрес устанавливается по текущему значению программного счетчика (см. команды T и G). Длинный вывод может быть прерван клавишей возврата на шаг.

**7) Команда M (MOVE)**

Команда M позволяет пересылать блоки команд или данных из одного места памяти в другое.

Форма команды

MS,F,D

Где: S - начальный адрес пересылаемого блока;

F - конечный адрес пересылаемого блока;

D - начальный адрес, куда этот блок пересылается.

Данные пересылаются из S в D и оба адреса увеличиваются. Затем S сравнивается с F, если они равны, команда заканчивается, иначе действия повторяются.

**8) Команда R (READ)**

Команда R используется в объединении с командой I для загрузки в ТРА с диска файлов с расширением .COM или .HEX и подготовки их для отладки.

Форма команды

R

RB

Где B - необязательное смещение, которое может использоваться для загрузки программ в адрес, отличающийся от адреса трансляции.

Операция загрузки не должна портить системные параметры в ячейках 0000H...0 (первая ячейка памяти). Если B опущено, то устанавливается B=0000H. Команде R предшествует команда I, определяющая имя .COM или HEX файла. Адрес загрузки для каждой записи HEX файла содержится в ней самой, адрес загрузки файлов .COM - 100H.

Для повторного чтения программы для отладки команда R может использоваться любое количество раз, если отлаживаемая программа не разрушает FSB в 5CH. Напомним, что все файлы типа .COM содержат машинные коды в двоичном виде (созданные командами SAVE или LOAD), а файлы типа .HEX содержат машинные коды в INTEL-овском стандарте (генерируются, например командой ASM).

Повторим, что команда

DDT имя файла расширения,

инициирующая DDT, эквивалентна последовательности

DDT

-I имя файла

-R

Иногда при использовании команды R, DDT выводит индикатор ошибки "?" (файл не может быть открыт или ошибка в контрольной сумме формата HEX). В этом случае можно повторить команду R до правильного чтения.

Если чтение проведено успешно, на консоль выводится:

NEXT PC

NNNN PPPP

Где : NNNN - адрес команды программы;

PPPP - текущее значение PS (100H для .COM или из последней записи HEX)

**9) Команда S (SET)**

Эта команда осуществляет проверку и изменение памяти.

Форма команды:

SS

Где S - 16-ричный начальный адрес для проверки и изменения.

DDT выдает адрес и содержимое ячейки с этим адресом. Если оператор вводит возврат каретки, данные не изменяются, если вводится байт данных, а затем <CR>, этот байт записывается в память в обоих случаях. DDT продолжает последовательно перебирать адреса до тех пор, пока не будет введена "." или не будет введен неправильный байт (не 16-ричное число).

**10) Команда T (TRACE)**

Команда T позволяет проследить последовательно по шагам выполнение программы.

Формат команды:

T

TN

В первом случае отображается состояние ЦПЭ и выполняется один шаг программы, и она останавливается, выдавая

$HHHH

Где HHHH - адрес следующей команды.

Адрес для команды D устанавливается равным HL, а для L - значение HHHH. Состояние регистров ЦПЭ может быть затем проверено командой X.

Вторая форма команды аналогична первой, за исключением того, что выполняется N шагов (N-B 16-ричном виде от 0 до 65535). При трассировке можно остановить программу клавишей "ВОЗВРАТ НА ШАГ"

Формат отображения такой же, как в команде X. Отметим, что программа трассировки прекращается при обращении в CP/M и возобновляется при возвращении обратно. Таким образом, функции CP/M, по вводу-выводу такие, как операции с диском и т.п. выполняются в реальном времени и не вызывают проблем. Выполнение программ в режиме трассировки в 500 раз медленнее выполнения в реальном времени, т.к. после каждой команды DDT берет управление на себя. Программы обработки прерываний могут также выполняться в режиме отладки, однако уровень 7 занят для команд, использующих RST 7(G,T и U). Проверка программ в режиме трассировки должна осуществляться в режиме деблокировки прерывания, что может создать проблемы синхронизации при отладки программ обработки прерываний.

**11) Команда U (UNTRACE)**

Команда U аналогична команде T, за исключением того, что содержимое процессора после каждого шага не отображается. Количество шагов от 1 до 65535 (0FFFFH). Все условия для T справедливы для U.

**12) Команда X (Examine)**

Команда позволяет отобразить и изменить текущее состояние элементов ЦПЭ.

Форма команды:

X

XR

Где R - один из элементов ЦП 8080;

C - флажок переноса (0/1);

Z - флажок нуля (0/1);

M - флажок минус (0/1);

E - четность (0/1);

I - внутренний перенос (0/1);

A - аккумулятор (0-FF);

B - регистры BC (0-FFFF);

D - регистры DE (0-FFFF);

H - регистры HL (0-FFFF);

S - указатель стека (0-FFFF);

P - программный счетчик (0-FFFF).

В первом случае состояние отображается в форме:

CFZFMFEFT A=BB B=DDDD D=DDDD H=DDDD S=DDDD P=DDDD INST

Где F=0 или I - значение флажков;

BB - байт, A DDDD - два байта, содержащие значения регистров.

Поле "INST" содержит мнемокод команды, адрес которой содержится в программном счетчике.

Во втором случае можно проверить и изменить содержимое регистра или флажка, определяемого R. В этом случае после ввода команды, отображается ввод с консоли. Если с консоли вводится только <CR> содержимое регистра (флажка) не изменяется. Если вводится новое значение перед <CR> содержимое изменяется. Отметим, что регистры BC, DE и HL отображаются и изменяются попарно.

**13) Дополнительные замечания**

Организация DDT позволяет при отладке больших программ, не входящих в TPA, перекрывать некоторую его часть. Сам DDT состоит из двух частей: ядра и модуля ассемблера (дизассемблера). Ядро всегда загружается на место CCP, а ассемблер (дизассемблер) располагается над ним.

При загрузке адрес BDOS, содержащийся в ячейке 6H (адресное поле команды JMP в 5H) модифицируется DDT в базовый адрес ядра DDT, который, в свою очередь, содержит команду перехода в BDOS. Таким образом, программы, использующие эту ячейку как адрес конца памяти, определяют этот конец по базовому адресу DDT.

Модуль ассемблера (дизассемблера) располагается в TPA если адрес конца введенной программы больше, чем начальный адрес этого модуля, это означает, что он запорчен и в ответ на команды L и A выдается "?", а в командах T и U в поле "INST" выводится команда в 16-ричном виде.

**3. Дополнительные возможности SID**

Добавлены две команды:

C - запуск программы с установкой регистров BC и DE. Контрольные и проходные точки игнорируются. Если запускаемая программа заканчивается возвратом, осуществляется вывод в отладчик.

Форма команды:

CS

CS,B

CS,B,C

Первая форма у команды запускает программу с адреса S. Регистры WC и DE устанавливаются в "0". Вторая форма запускает программу с адреса S. Регистр DE устанавливается в "0", BC устанавливается в "B". Третья форма запускает с адреса S, BC устанавливается в "B", DE - в "C".

Команда P устанавливает проходные точки. Отличия проходных точек от контрольных следующие:

* они не сбрасываются при проходе через них. Останов в этой точке может быть задан после N-го прохода, где N может меняться в пределах от 1 до 255;
* при каждом проходе через эту точку выполнение программы прерывается, на консоль выводится количество оставшихся проходов, адрес точки и содержимое регистров и флажков.
* если количество оставшихся проходов >1, оно уменьшается на 1 и программа запускается дальше. При числе проходов = 1 программа останавливается как при достижении контрольной точки при этом проходная точка не сбрасывается, а количество проходов остаются равным 1. Может быть задано до 8-ми проходных точек.

Формы команды:

P

-P

PS

PS,B

-PS

Первая форма отображает все установленные проходные точки. Вторая форма сбрасывает все установленные проходные точки. Третья форма устанавливает точку в адрес S с числом = 1. Четвертая форма устанавливает точку в адрес S с числом проходов B. Пятая форма сбрасывает точку в адресе S.

Параметры:

Параметры в командах могут задаваться в шестнадцатеричном виде: как DDT и в десятичном. При задании параметров в десятичном виде перед ним ставится символ "#".

Параметр может представлять собой сумму или разность нескольких чисел.

Пример:

L#256+100

запускает команду L с адреса 200H;

L#512-100+10

запускает команду L с адреса 210H

В качестве параметра может использоваться содержимое стека. Для использования в качестве параметра содержимого стека необходимо ввести " " ("'"). Количество введенных символов " " определяет глубину стека, откуда берется параметр. Например, если содержимое стека будет таким, как приведено на рисунке:

\_\_\_\_\_\_\_\_\_\_

BBBB SP+10

~~~~~~~~~~

0650 SP+8

~~~~~~~~~~

3840 SP+6

~~~~~~~~~~

AF50 SP+4

~~~~~~~~~~

2100 SP+2

~~~~~~~~~~

1000 SP

~~~~~~~~~~

то ввод следующих команд будет означать:

L^ - запуск команды с адреса 1000;

D^^^ - запуск команды D с адреса AF50;

S^^^^+100 - запуск D с Адреса 750.

В качестве параметра могут быть использованы коды ASCII.

Например:

L "AВ"

Запускает L с адреса 4142. Команда D и S могут работать в режиме байта как в DDT и в режиме слова (двух байтов).

Так, команда

DW100,140

отображает содержимое памяти с адреса 100и 140 в виде двухбайтовых слов, при этом старший байт слова отображается первым, A младший - вторым (в памяти они хранятся наоборот).

Команда

SW300

позволяет изменять содержимое памяти в виде двухбайтовых слов. В качестве вводимых параметров в команде S могут использоваться символы кода ASCII, при этом перед ними нужно ввести символ ".

Например, команда

S 100 <CR>

0100 "ABCDEFGH <CR>

вводит символьную последовательность "ABCDEFGH" с адреса 100H.

3.1. Символьная отладка

В качестве параметров SID может использовать символьные метки вашей программы. Значения символьных меток загружаются из файла типа SYM, создаваемого макроассемблером MAC или редактором связей L80 при загрузке перемещаемых модулей, если указаны переключатели IN и IY.

Для загрузки символьных меток необходимо команде I указывать два имени - имя исполняемого файла и соответствующего ему файла с SYM. Например:

#I CONS.HEX CONS.SYM

Если исполняемый файл был загружен ранее, то для загрузки файла .SYM используйте команду

#I XX CONS.SYM

При символьной отладке символьные метки могут быть использованы в качестве параметров. При этом:

&XXX - двухбайтовое содержимое метки XXX;

=XXX - однобайтовое содержимое метки XXX.

Например:

#L.BREACK - запустить L с метки BREACK;

#P.COLD,5 - установить в метке .COLD проходную точку с 5-ю проходами;

#D& MEMORY - запустить команду D с адреса, находящегося в памяти по адресу MEMORY;

#XA

A=00=LOAD - загрузка в A содержимого ячейки LOAD.

В режиме символьной отладки для отображения всех меток может использоваться команда H без параметров.

Команда H с одним параметром указывает его значение в 16-ричном виде, десятичном виде и, если возможно, в виде кода ASCII.

Команда H с двумя параметрами выдает сумму и разность этих параметров.