# ОТЛАДОЧНАЯ СРЕДА МИКРОКОНТРОЛЛЕРОВ ТЕСЕЙ
# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Отладочная среда предназначена для разработки и отладки при помощи языка Ассемблер и компилятора **TESSA 0.1** прикладных программ для микроконтроллеров, построенных на основе микроконтроллерного ядра **ТЕСЕЙ**, в том числе для микроконтроллера **КР1878ВЕ1** (An15E03). Отладка программ осуществляется либо на персональном компьютере в режиме эмуляции, либо на специальном аппаратном отладочном модуле. Для выполнения программирования и отладки необходимы IBM-совместимый персональный компьютер и пакет программ **TESSA 0.1**.

## НАЗНАЧЕНИЕ И СОСТАВ ПРОГРАММ

Отладочная среда микроконтроллера (далее **ОСМ**) предназначена для:

- загрузки и пуска ***.SAV** программ в программный эмулятор или отладочный модуль;
- редактирования ОЗУ данных в программном эмуляторе или отладочном модуле;
- редактирования ОЗУ команд в программном эмуляторе или отладочном модуле;
- трассировки программ, работающих в программном эмуляторе или отладочном модуле.

## УСТАНОВКА И ЗАПУСК ПРОГРАММЫ

Для работы с программой необходимо на Вашем персональном компьютере (совместимом с IBM PC/AT) создать ее директорию (имя и тип диска принципиального значения не имеют) и записать в нее с прилагаемого к изданию диска следующие файлы:

– **MC_WIN.EXE** (управляюшая программа);

– ***.INI** (сохраняемые на диске установки);

– **CS_8x11.FNT** (шрифты).

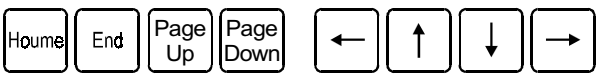
Запуск программы производится следующим образом:

**Path DIR.** > **MC_WIN.EXE** <**Enter**>

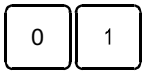
По завершении чтения текущих установок (***.INI**) и опознания подключенного оборудования программа начинает работу в текущем активном окне.

## ВВОД ДАННЫХ С КЛАВИАТУРЫ

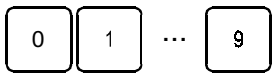
***Клавиши просмотра***




***Клавиши редактирования:***




– Двоичные числа;




– Десятичные числа;

0 1 … 9 A B … F – Шестнадцатиричные числа;

X – Маска числовой позиции;

Алфавито-цифровые клавиши – ASCII строки;

SP + - (Тип курсора: "Блок") – ASCII массив (рулон);

***Клавиши управления окнами:***

Ctrl + Home – Переключение активного окна;

Ctrl + End – Сворачивание окна в **Icon Box**;

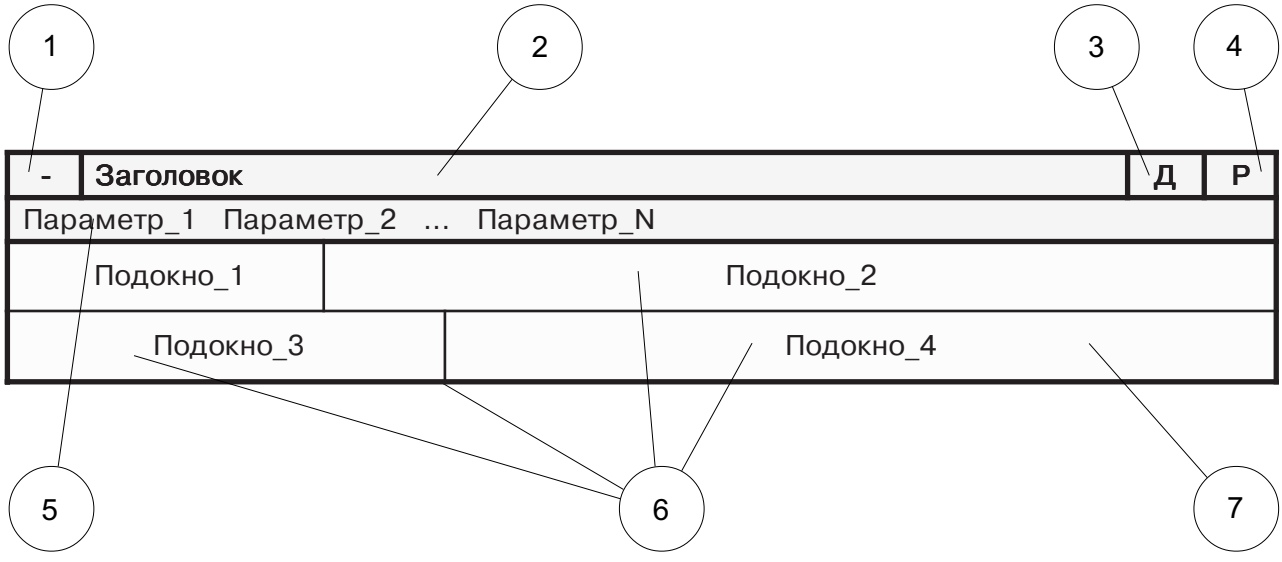
Tab, Shift + Tab – Переключение между подокнами (внутри активного окна);

Ctrl + ← – Расширение окна **DOS_Terminal** на весь экран;

Ctrl + → – Уменьшение размеров окна **DOS_Terminal**.

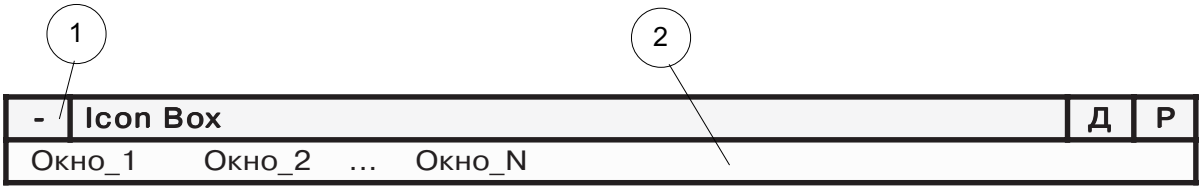
## ОСНОВНЫЕ СОГЛАШЕНИЯ

***Общий вид рабочего окна***



| Рабочее поле | Действие | Мышь | Клавиатура |
|---|---|---|---|
| 1 | Усыпление активного окна (АО) | Однократное нажатие | <Ctrl>+<End> |
| 2 | Перемещение АО на экране дисплея в новую позицию | Нажатие и удержание левой кнопки до завершения перемещения | - |
| 3 | Деактивизация окна; переключение управления на следующее окно | Однократное нажатие левой кнопки | <Ctrl>+<Home> |
| 4 | Изменение размера АО (вывод курсора за границу АО и изменение размера) | Нажатие и удержание левой кнопки до завершения масштабирования | - |
| 5 | Работа с параметрами АО (разворачивающиеся вниз подокна) | Однократное нажатие левой кнопки | <Alt> |
| 6 | Переход между подокнами в рабочей области АО | Однократное нажатие левой кнопки в требуемой области подокна | <Tab><br><Shift>+<Tab> |
| 7 | Перемещение курсора в области данных активного подокна | Однократное нажатие левой кнопки в новой строке подокна | <↑>, <↓>,<br><→>, <←>,<br><Home>, <PgUp>,<br><PgDwn>, <End> |

***Общий вид окна*** **"Icon Box"**




| Рабочее поле | Действие | Мышь | Клавиатура |
|---|---|---|---|
| 1 | Выход в DOS | Однократное нажатие левой кнопки | <Ctrl>+<End> |
| 2 | Перемещение курсора между иконками "спящих" окон в области требуемой иконки | Однократное нажатие левой кнопки | <Tab><br><Shift>+<Tab> |
| | Активизация указанного окна | Повторное нажатие левой кнопки | <Enter> |

## ОПИСАНИЕ РАБОЧИХ ОКОН

### *Окно установок* (Set Up)

| - | Icon Box | Д | Р |
|---|---|---|---|

**Default Pathnames**

Source Programm: C:\MCT\MIC\example.mic
Trace Data File: C:\MCT\TRS\
I/O Emulat. File: C:\MCT\ext_dev.exe

**Target System** — Emulator; **Communication Port** — COM 2

| IR0(Size) | IR1(Size) | ISP(Vol) | DSP(Vol) | RAM(Vol) |
|---|---|---|---|---|
| 8 bit | 14 bit | 8 * PC | 32 * 8 | 256 * 8 |

**Default Pathnames** (“Пути Файлов По Умолчанию”)**:**

**Source Programm** – имя директории (файла), которое будет устанавливаться по умолчанию для операций чтения/записи исполняемых файлов при работе с окном **Отладчик**;

**Trace Data File** – имя директории (файла), которое будет устанавливаться по умолчанию для операций чтения/записи файлов трассы при работе с окном **Просмотр трассы**;

**I/O Emulat. File** – программа в формате IBM/PC для эмуляции внешнего оборудования, подключенного к микроконтроллеру (для режима **Эмулятор**).

**Target System** (“Объектная ЭВМ”)**:**

**Test Envelope** – загрузка и исполнение (отладка) программ в режиме реального времени на аппаратуре отладочного модуля;

**Emulator** – исполнение (отладка) программ в режиме эмуляции на IBM/PC.

**Communication Port** (“Связь С Отладочным Модулем”)**:**

**COM 1** – Управление отладочным модулем по асинхронной линии связи, подключенной к **COM1** (IBM/PC).

**COM 2** – Управление отладочным модулем по асинхронной линии связи, подключенной к **COM2** (IBM/PC).

**IR0/IR1**(Size):

Размер индексных регистров (8 ¸ 14 бит). Действительно для режима **Эмулятор**.

**ISP/DSP**(Volume):

Размер стека команд (4/8/16 элементов), стека данных (16/32/64 элемента). Действительно для режима **Эмулятор**.

**RAM**(Size):

Размер ОЗУ в байтах (128/ 256/ 512 / 1К ... 16К). Действительно для режима **Эмулятор**.

### *Окно отладчика* (Debugger)

*Режим «Отладочный модуль»*

Параметр **Файл** (File)**:**

- загрузка исполняемого файла (***.SAV**) в память системы для отладки;
- загрузка программы (***.EXE**) эмуляции внешней среды микроконтроллера;
- сохранение на диске образа ОЗУ микроконтроллера .

Параметр **Трасса** (Trace)**:**

- установка количества повторений для указанных точек останова (**Breakpoint Table**);
- управление параметрами трассировки в **WAIT**-состоянии (шаг трассировки в единицах входной частоты, разрешение/запрет трассировки в WAIT-состоянии);
- выбор периферийных устройств (порты, таймеры) для трассировки.

Параметр **Поиск** ("Search")

- Поиск указанного текста в подокне **Символьная отладка**.

```
-  Debugger
File   Trace   Search
PC      Opcode   Assembler          ___I/O-Adr_____I/O-Data____
                                        00            FF
>0000   8012     JMP   012h
 0001   8011     JMP   011h   Addr  _____________Data_____________________
 0002   8011     JMP   011h   Seg   1   2   3   4   5   6   7   8
 0003   8015     JMP   015h   0040  F F F F F F F F F F F F F F F F
 0004   8039     JMP   039h   0048  F F F F F F F F F F F F F F F F
 0005   8011     JMP   011h   0050  F F F F F F F F F F F F F F F F
 0006   8011     JMP   011h   0058  F F F F F F F F F F F F F F F F
 0007   81F3     JMP   1F3h   0060  F F F F F F F F F F F F F F F F
 0008   8011     JMP   011h   ____________________________________________
 0009   8011     JMP   011h   0068  F F F F F F F F F F F F F F F F
 000A   8011     JMP   011h   0070  F F F F F F F F F F F F F F F F
 000B   8011     JMP   011h   0078  F F F F F F F F F F F F F F F F
 000C   8011     JMP   011h   0080  F F F F F F F F F F F F F F F F
 000D   8011     JMP   011h   0088  F F F F F F F F F F F F F F F F
 JMP    START   ;Точка старта программы
 JMP    ERROR   ;Переход на обработку ошибки
 JMP    ERROR   ;Переход на обработку ошибки
 JMP    I_TMRA  ;Переход на обработку опрерывания от таймера A
 JMP    I_TMRA  ;Переход на обработку опрерывания от таймера B
 JMP    ERROR   ;Переход на обработку ошибки
 JMP    ERROR   ;Переход на обработку ошибки
  Seg_A  Seg_B  Seg_C  Seg_D  IR0.%D6  IR1>%D7  Status  iSP  dSp
. 000    000    000    000    00 0000   0 0000   0000000  00   00
  F1>Step  F2>Step_Prc  F3>Go  F5>Set_BRK  F6>Rst_BRK  F8>PC=0
```

***Режим «Эмулятор»***

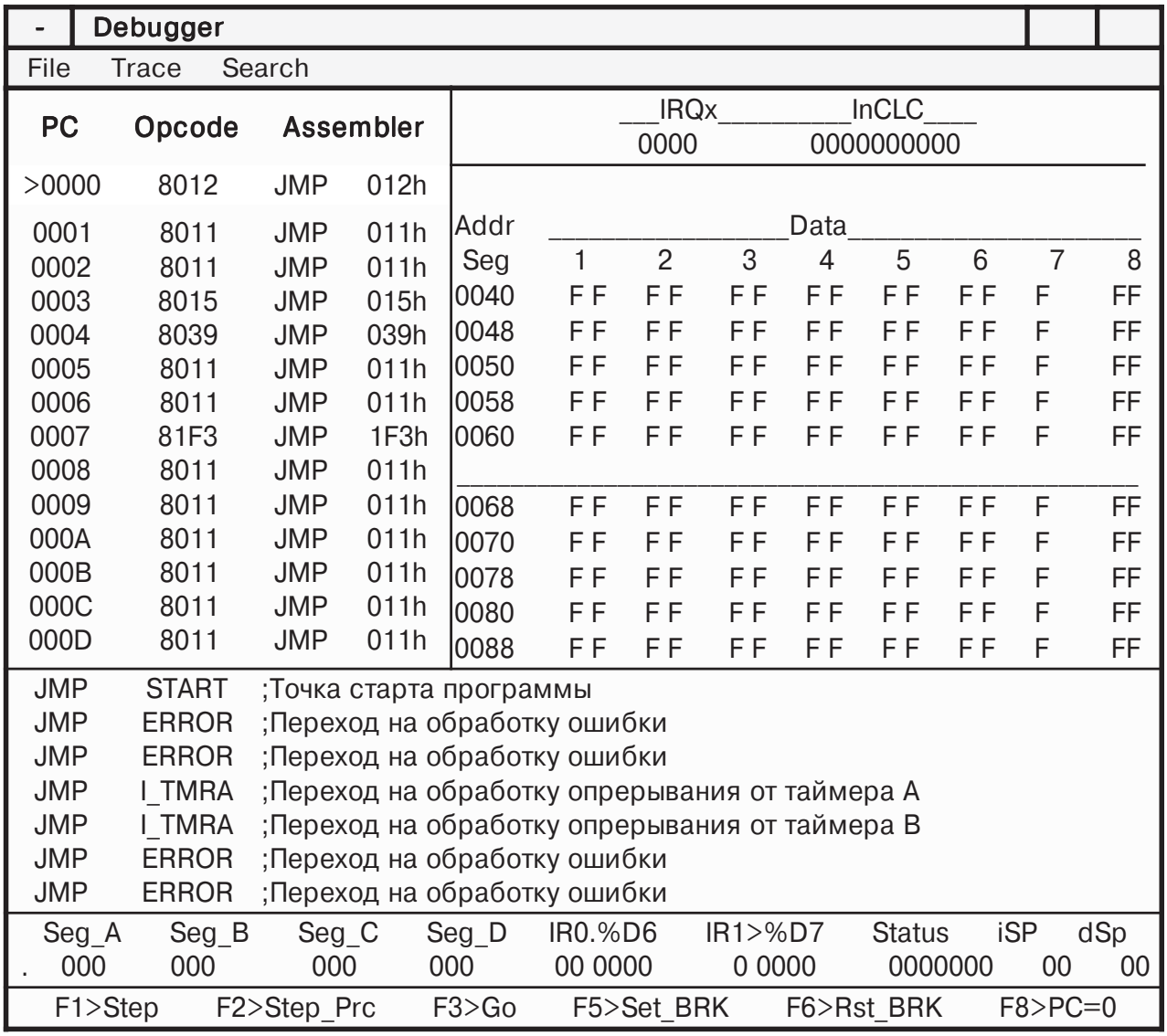
Подокно **Строка-подсказка** (назначение функциональных клавиш):

**F1>Step** – исполнение текущей команды (помеченной **>**);

**F2>Step_Prc** – исполнение текущей команды/процедуры (**JSR**, **IJSR**);

**F3>Go** – исполнение программы от текущей команды до точки останова;

**Ctrl+F3>** – исполнение программы до текущего положения курсора (допустимо только для программ эмулятора);




**F5>Set_BRK** – установка точки останова на просматриваемой (яркая подсветка) команде (точек останова может быть не более 4, ожидаемых по **ИЛИ**);

**F6>Rst_BRK** – сброс всех точек останова;

**F8>PC=0** – инициализация эмулятора/отладочного модуля (адрес текущей команды. (**PC**) = 0, очистка ОЗУ, ...)

Подокно **Рабочая программа**:

Состоит из трех столбцов – счетчика команд (**PC**), кода команды (**Opcode**), реассемблированного текста команды (**Assembler**). Команда, готовая к исполнению (**PC**) помечена символом **>**. Текущая просматриваемая (редактируемая) команда подсвечена ярким цветом. Для редактирования доступны поля: **PC**, **Opcode** (в 16-ричном представлении), тетрадная позиция помечна мигающим курсором в форме символа подчеркивания. При измененииисчетчика команд окно отображения программы перемещается на указанный участок программы.

Подокно **ОЗУ Данных, Обл.Ввода/Вывода**:

Состоит из двух областей - области ввода/вывода (адр.**0** ¸ **3Fh**), ОЗУ данных (адр.**40h** ¸ **3FFFh**). Чтение/запись регистров периферийных устройств (I/O) вынесено в отдельный режим из соображения корректности данной процедуры, т.к. при этом может быть либо сброшено прерывание, либо возникнуть ошибка для данного уст-

ройства (РУ таймера, последовательные интерфейсы, порты, ...). Область ОЗУ данных можно просматривать/корректировать (в шестнадцатеричном представлении) без каких-либо ограничений на любом шаге отладки программы. При останове программы, цветом выделяются строки, соответствующие положению сегментов **A**, **B**, **C**, **D**.

Подокно **Системные Регистры**:

Служит для просмотра содержимого базы сегментных регистров **A**, **B**, **C**, **D**, регистров косвенной адресации **IR0**/**IR1** (слева 2 старших бита режима, справа 14 разрядов адреса), регистра состояния (**A13**, **A12**, **OF**, **DC**, **IE**, **N**, **Z**, **C**), указателя стека инструкций (**iSP**) и указателя стека данных (**dSP**).

Подокно **Символьная отладка**:

Служит для отладки программы в виде исходного текста. Текущая команда подсвечена;

При работе в подокне с исходным текстом допустимы все команды для установки/сброса точек останова, пуска программы и т.д;

После задания модели поиска (исполнение команды **Search**), продолжить поиск возможно, используя клавиши <**Ctrl**>+<**PageUp**> и <**Ctrl**>+<**PageDown**> для нахождения требуемого текста соответственно выше текущей строки и ниже текущей строки.

Переключение между подокнами производится либо мышью, либо клавишами <**Tab**>, <**Shift**>+<**Tab**>.

### *Окно просмотра трассы*

Параметр **Файл** (File)**:**

- загрузка двоичного файла трассы в память системы для просмотра;
- сохранение двоичного образа трассы на диске (для последующего просмотра в системе);
- сохранение трассы на диске втекстовом виде (для вывода на принтер).

| - | Scan Tracer Data | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| File | | | | | | | | | | | | | | | |
| | | | Assembler | | Src | | Dst | | Wr | xSP | | Status | | | Port |
| PC | Opcode | | Instruction | | Addr | Dt | Addr | Dt | Dt | I | D | AADOINZC | I/O_Data | | Dt |
| *int | 8012 | jmp | 012h | | | | 0012 | | | 1F | | 00000000 | 00000000 | | |
| 0012 | 2000 | ldr | #0,000h | | | 00 | 0000 | | 00 | | | | | | 00 |
| 0013 | 2241 | ldr | #1,240h | | | 48 | 0001 | | 48 | | | | | | 00 |
| 0014 | 2042 | ldr | #2,040h | | | 08 | 0002 | | 08 | | | | | | 00 |
| 0015 | 5590 | movl | %C0,ACh | | | AC | 0040 | FF | AC | | | 00000100 | | | |
| 0016 | 42D1 | movl | %C1,16h | | | 16 | 0041 | FF | 16 | | | 00000000 | | | |
| 0017 | 0188 | sst | 8h | | | 08 | | | | | | | 00000000 | | |
| 0018 | 0600 | mov | %A0,%C0 | | 0040 | AC | 0028 | 00 | AC | | | 00001100 | | | |
| 0019 | 040B | mov | %B3,%A0 | | 0028 | AC | 0243 | FF | AC | | | 00001100 | | | |
| 001A | 0E11 | sub | %C1,%C0 | | 0040 | AC | 0041 | 16 | 6A | | | 00101001 | | | |
| 001B | 0A11 | cmp | %C1,%C0 | | 0040 | AC | 0041 | 6A | | | | 00111101 | | | |
| 001C | 028B | mtpr | #4,%B3 | | 0243 | AC | 0004 | | | | | | | | 00 |
| 001D | 0389 | mfpr | %B1,#4 | | 0004 | | 0241 | FF | AC | | | | | | 00 |

Окно **Просмотр Трассы** (Scan Tracer Data)**:**

Трасса представляет собой развернутый во времени процесс выполнения программы (включая циклы, подпрограммы, прерывания, ...). Для каждой команды на экране отображаются следующие параметры:

**PC** – текущий счетчик команд (*int-переход к программе обработки прерывания);

**Opcode** – код команды (в 16-ричном представлении);

**Assembler Instruction** – реассемблированный текст команды;

**Src Addr** – адрес операнда источника;

**Src Dt** – данные операнда источника;

**Dst Addr** – адрес операнда приемника;

**Dst Dt** – данные операнда приемника (до операции);

**Wr Dt** – записываемые данные (по адр.**DST**);

**xSP I** – указатель стека команд;

**xSP D** – указатель стека данных;

**Status** – регистр состояния (**A13**, **A12**, **OF**, **DC**, **IE**, **N**, **Z**, **C**);

**I/O_Data** – данные периферийных устройств (порты, таймеры);

**Port Dt** – данные порта (**A**, **B**, **C**, **D**).

Для команды **WAIT** данные периферийных устройств (**I/O_Data**) накапливаются в трассе. Для их просмора необходимо установить курсор на строку с командой **WAIT** и, нажав клавишу <**Enter**>, вызвать подокно с **I/O_Data**. Выход из данного режима осуществляется нажатием клавиши <**Esc**>.

## ОПИСАНИЕ РЕЖИМОВ РАБОТЫ

### *Рекомендации по работе в режиме “Эмулятор”*

Начать работу по отладке Вашей программы рекомендуется с окна **Установки** (**Set-Up**). Для удобства работы следует установить пути доступа к файлам указанным в данном окне. В результате Вам не понадобится вводить их в соответствующих окнах при загрузке исполняемых модулей, работе с трассой и т.д. Затем установите режим работы **Эмулятор** и настройте программы, описывающие конфигурацию микроконтроллера на требуемые размеры **IR0**/ **IR1**, объем стека команд / данных, размер ОЗУ.

Собственно работа по отладке проводится в окне **Отладчик** (**Debugger**). Для загрузки прикладной программы следует нажать клавишу <**Alt**> либо выбрать мышью параметр **Файл** и, установив курсор в позицию **Загрузка *.SAV файла**, нажать <**Enter**>. В появившемся окне следует набрать полный путь и имя исполняемого (***.SAV**) файла. Если всё было сделано правильно, в левой части окна появится текст прикладной программы.

Выполнять программу можно в двух основных режимах:

- пошаговом (исполнение текущей команды <**F1**> либо подпрограммы <**F2**>);

- пуск до точки останова (клавиша <**F3**>);

- пуск до текущего положения маркера (<ctrl>+<F3>).

Выбор (отмена) точек останова (**ТО**) производится клавишей <**F5**>. В каждый момент времени можно установить до четырёх **ТО**, которые будут срабатывать по **ИЛИ** (т.е.

останов процессора произойдёт по достижении любой из указанных **TO**). Многократное повторение **TO** можно задать (выбрав в строке параметров **Breakpoint Table**) в подокне **Таблица TO**. Количество повторов может быть задано в пределах от 1 до 254. Сброс всех выбранных **TO** можно осуществить, нажав клавишу <**F6**>.

В любой момент времени можно просмотреть содержимое ОЗУ (адреса **40h ¸ 3FFFh**), область ввода вывода (адреса **0 ¸ 3Fh**) и системные регистры микроконтроллера (базу сегментов **A**, **B**, **C**, **D**, индексные регистры **IR0**/**IR1**, указатели стека данных **DSP**, команд **ISP**). Текущее состояние памяти (адреса **0 ¸ 3FFFh**) можно сохранить в указанном Вами файле, выбрав в строке параметров опцию **Сохранение ОЗУ данных** (**Save RAM Data**).

Если Вы пустили микроконтроллер на выполнение программы (<**F2**>,<**F3**>), а процессор не выходит долгое время на **TO**, можно остановить его принудительно, нажав клавишу <**Esc**>.

Исполнение программы будет прервано, и маркер укажет на команду, на которой Вы остановили микроконтроллер. При желании можно продолжить отладку с данного места либо перезапустить процессор. Перезапуск Вашей программы (установка счетчика команд равным **0**) осуществляется нажатием клавиши <**F8**>.

В данном окне отображаются также запросы периферийных устройств на прерывание (**IRQx**: в 16-ричном представлении от **IRQ0** до **IRQ15**) и количество периодов входной частоты (**InCLC**: в десятичной нотации), прошедших с момента старта микроконтроллера. Так как процессор исполняет все команды за одинаковое время (1 команда : = 2 периода входной частоты, исключая **WAIT**), то легко подсчитать количество выполненных команд, а также время работы (зная входную частоту).

Также следует отметить, что в данном режиме (**Эмулятор**) постоянно идет заполнение трассы, включая пошаговое (клавиша <**F1**>) исполнение программы.

При отладке программы имеется возможность наблюдать за состоянием периферийных устройств (портов, таймеров, ...). Режимы трассировки выбираются опциями: **WAIT-I/O trace** и **WORK-I/O trace**. Трассировка ведется в двух форматах: 30-bit (команды **WAIT**, **JMP**, **JSR**, **JC**, **JE**, ...) и 8-bit (**LDR**, **MTPR**, **MFPR**, **PUSH**, **POP**). Для **WAIT** можно выбрать шаг сканирования в единицах входной частоты (1, 2, ..., 254), а также запрет / разрешение трассировки. Для остальных команд отображение состояния выбранных периферийных устройств ведется постоянно, синхронно с выполнением команд процессором.

Для ускорения работы программного эмулятора введен режим отключения трассировки программы, который включается при сворачивании окна **Trace Data**.

При запуске программы в режиме программного эмулятора контролируются так называемые исключительные ситуации в работе процессора. Если это случилось, то исполнение программы приостанавливается и на экране появляется сообщение, содержащее адрес команды, в которой произошло исключение (**PC:xxxx**), и один из нижеприведенных фрагментов текста, поясняющий ситуацию:

- недопустимый код команды;
- обращение по несуществующему адресу ОЗУ;
- обращение к **IR1** в режиме адресации к памяти команд в качестве приемника;
- переполнение **IR1** в режиме адресации к памяти команд при переходе через границу памяти команд;
- недопустимое значение адреса системного регистра при выполнении команд **LDR**, **MTPR**, **POP**;

- недопустимое сочетание операндов (например, **MTPR #0,%A1**);
- недопустимое использование **IR1** (например, **MTPR #0,%D7**, когда **IR1** в режиме адресации к памяти команд);
- недопустимые операции со стеком (если при прерывании по адресу вектора прерывания стоит команда работы со стеком **PUSH**, **POP**);
- переполнение стеков.

Для продожения работы следует нажать любую клавишу (для удаления сообщения с экрана) и продолжить отладку программы.

В микроконтроллерах, имеющих нестандартные устройства (электрически стираемые ПЗУ, ЖКИ ...), в эмулятор добавляются соответствующие окна, позволяющие просматривать и редактировать EEPROM, видеть эмуляцию экрана ЖКИ в процессе отладки программы.

### *Рекомендации по работе в режиме* "Отладочный модуль"

Подключите аппаратуру отладочного модуля (**ОМ**) к по следовательному порту (**COM1** или **COM2**) . Выберите в окне **Установки** опцию, соответствующую указанному порту, а также режим работы **Отладочный модуль**.

Практически всё сказанное выше справедливо для данного режима работы, за следующими исключениями:

Заполнение трассы производится только в режиме пуска с **ТО** (клавишы <**F3**>, <**F2**>), причём заполнение начинается с начала для каждого старта процессора. Если выполнение программы было прервано (клавиша <**Esc**>), то состояние процессора (текущая исполняемая команда, регистры,...) становится неопределённым и рекомендуется перезапустить микроконтроллер (клавиша <**F8**>).

Чтение оперативной памяти микроконтроллера (включая область **0 ¸ 3Fh**), отображаемой в двух подокнах (по 40 байт) производится после каждого останова (включая пошаговый режим). Это эквивалентно доступу самого процессора по чтению в данные области памяти. Поэтому следует быть осторожным, отображая область периферийных устройств (**0 ¸ 3Fh**), т.к. таким действием (чтением) можно нарушить работу программы (например, читая регистры таймеров, портов и т.д., можно сбросить установленные запросы на прерывание). Для уменьшения вероятности ошибки рекомендуется пользоваться подокном отображения указанной ячейки (байта) области ПУ в правом верхнем углу окна. Вывод туда производится с теми же ограничениями, хотя процессор в данном случае читает только данный байт данных, не затрагивая соседние регистры.

## ОТЛАДОЧНЫЙ МОДУЛЬ ТЕСЕЙ

Наиболее достоверную отладку прикладной программы микроконтроллеров **ТЕСЕЙ** обеспечивает использование программно-аппаратного отладочного модуля **ТЕСЕЙ** (**ОМ**).

**ОМ** построен на основе специального отладочного кристалла, содержащего все основные модули микропроцессорного ядра **ТЕСЕЙ**. В результате отладка программ производится практически в резидентном режиме, что и обеспечивает максимально возможное качество отладки. **ОМ** предназначен для отладки прикладных программ любых микроконтроллеров, построенных на основе ядра **ТЕСЕЙ**, в том числе микроконтроллера **КР1878ВЕ1**.

**ОМ** работает под управлением IBM-совместимого персонального компьютера, который так же является средством отображения результатов работы комплекса. Связь

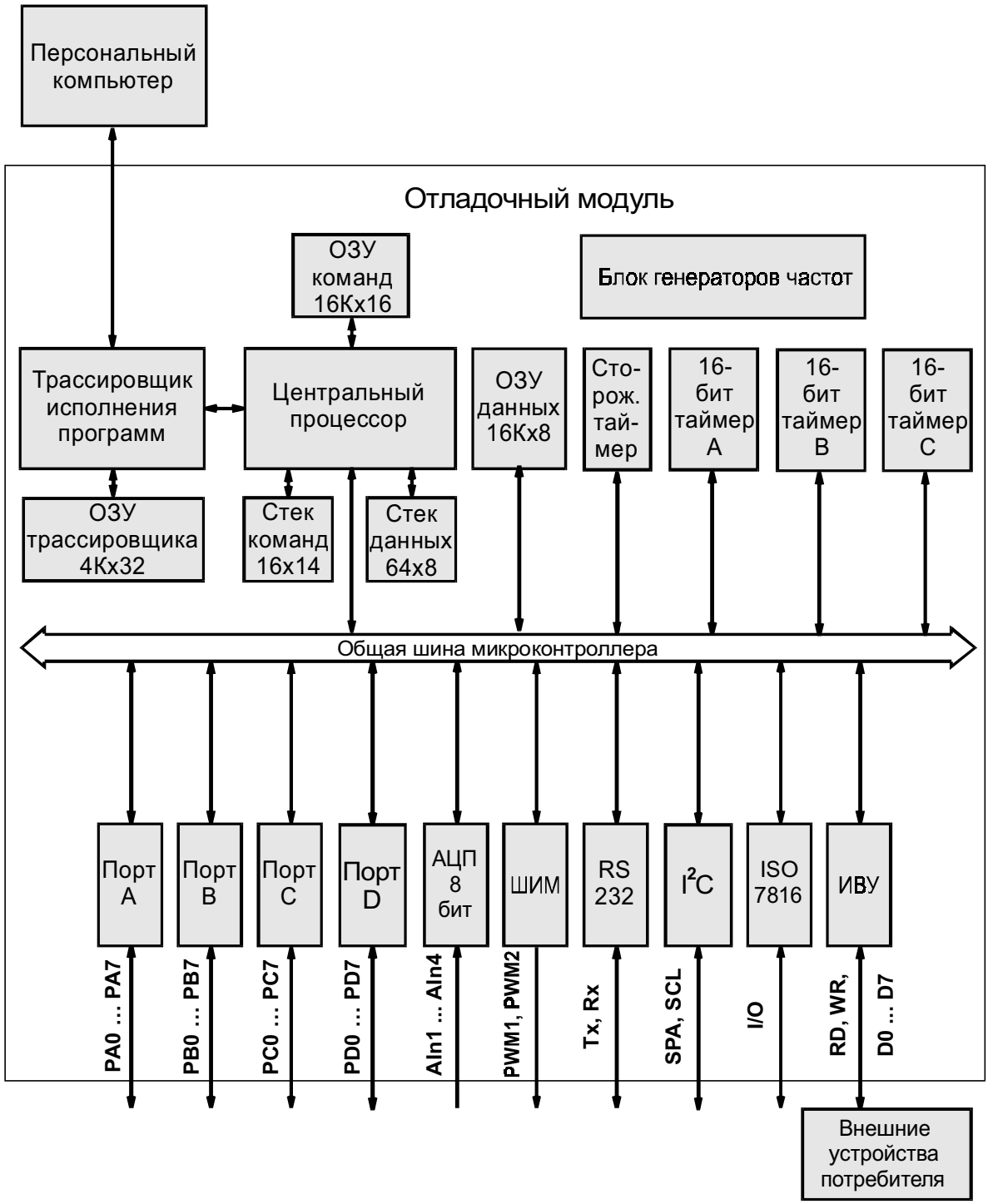



**Рис. 23. Структурная схема** ОМ

между компьютером и **ОМ** осуществляется по протоколу RS-232.

**ОМ** обеспечивает выполнение следующих функций:

- поддержка консольного режима между **ОМ** и ПЭВМ;
- загрузка из ПЭВМ программы пользователя в **ОМ**;
- пуск процессора **ОМ** для исполнения программы пользователя с заданного адреса;
- останов выполнения программы по заданным условиям;

- останов выполнения программы по команде из ПЭВМ;
- трассировка выполнения программы пользователя по заданным условиям;
- редактирование данных в ОЗУ данных и команд пользователя;
- выгрузка данных из ОЗУ трассировки в ПЭВМ для обработки.

Трассировщик **ОМ** содержит четыре вида защелок, срабатывающих по разным условиям.

*Защелка первого типа* ( 8 условий ) анализируют совпадение адреса команды процессора и введенного эталона. Срабатывание защелки происходит при достижении запрограммированного количества таких совпадений. Эти условия могут быть запущены в любом сочетании независимо друг от друга. При одновременной работе нескольких условий первого типа процессор останавливается при срабатывании одного из них. При этом процессор не выполняет ту команду, на которой произошло срабатывание защелки.

*Защелка второго типа* анализирует попарное совпадение адресов и данных операндов источника и приемника в фазе чтения с введенными эталонами. Срабатывание защелки происходит при достижении запрограммированного количества таких совпадений. Любые разряды данных и восемь младших разрядов адреса могут быть замаскированы, в этом случае в сравнении будут участвовать только незамаскированные разряды.

*Защелка третьего типа* анализирует попарное совпадение адреса и данных операнда приемника в фазе записи с введенными эталонами. Срабатывание защелки происходит при достижении запрограммированного количества таких совпадений. Любые разряды данных и восемь младших разрядов адреса могут быть замаскированы, в этом случае в сравнении будут участвовать только незамаскированные разряды.

*Защелка четвертого типа* анализируют совпадение данных на одном из четырех внешних портов и введенного эталона. Срабатывание защелки происходит при достижении запрограммированного количества таких совпадений. Любые разряды данных могут быть замаскированы, в этом случае в сравнении будут участвовать только незамаскированные разряды.

Все защелки могут быть запущены в любом сочетании независимо друг от друга. При одновременной работе нескольких защелок разного типа процессор останавливается при срабатывании одной из них.

Для задержки остановки процессора может быть запрограммирован регистр, содержимое которого говорит о количестве команд, которое будет выполнено после срабатывания защелки до момента остановки процессора.