ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

ДВК

КНИГА 7

1988 год

АННОТАЦИЯ

Данная книга является руководством по работе и эксплуатации тест - мониторной системы (ТМОС).

В главе "ОПИСАНИЕ СИСТЕМЫ" содержится информация о назначении, области применения, составе и функциях системы. Приведены сведения об оборудовании, определены обозначения и термины, которые используются при описании ТМОС.

Глава "ОПИСАНИЕ ТЕСТОВЫХ ПРОГРАММ" содержит описание тест - программ, действия оператора при загрузке и запуске тест - программ, команды, соответствующие каждой тест - программе и сообщения оператору (выводимые на экран или печать) при выполнении этих тест - программ.

В главе "ПОДСИСТЕМА МУЛЬТИПРОГРАММНОЙ ПРОВЕРКИ" перечислены функции мультипрограммного монитора, описаны его возможности и способы генерации тестов комплекса для проверки оборудования. Глава содержит общую информацию о подсистеме мультипрограммной проверки, описание состава подсистемы мультипрограммной проверки и процедур.

Глава "ПРОГРАММИРОВАНИЕ ТЕСТОВЫХ МОДУЛЕЙ" содержит описание структуры тестовых модулей, подробный разбор всех возможных типов интерфейсов модулей и макровызовов, с помощью которых тестовые модули могут обращаться к монитору за различными услугами. Перечислены общие стандарты по программированию, которых следует придерживаться для обеспечения работы с этими тестовыми модулями. Перед попыткой сконструировать новый тестовый модуль программист должен ознакомиться с документацией по ТМОС и иметь навыки работы на языке МАКРОАССЕМБЛЕР. Последующее изложение подразумевает, что программист понимает программные и технические особенности устройства для которого пишется модуль и уже составил предварительную схему программы.

ОПИСАНИЕ СИСТЕМЫ. РУКОВОДСТВО ОПЕРАТОРА.

1. НАЗНАЧЕНИЕ ТЕСТ-МОНИТОРНОЙ СИСТЕМЫ

Тест-мониторная операционная система (ТМОС) предназначена для эксплуатации на микро-ЭВМ с магистральной структурой (ПЭВМ), программно совместимой с "ЭЛЕКТРОНИКОЙ-60". ТМОС входит в состав базового программного обеспечения ПЭВМ, и предназначена для проверки работоспособности комплексов.

ТМОС обеспечивает следующие функции:

- проверку работоспособности отдельных устройств, входящих в состав комплекса;
- загрузку и запуск тестовых программ;
- обслуживание запросов тестовых программ;
- модификацию и создание тестовых программ;
- создание и корректировку текстовых файлов;
- выполнение последовательности команд командного файла;
- копирование и проверку магнитных носителей;
- генерацию рабочей версии тест-мониторной системы;
- пополнение существующей версии системы новыми программными компонентами.

2. УСЛОВИЯ ВЫПОЛНЕНИЯ ТЕСТ-МОНИТОРНОЙ СИСТЕМЫ

2.1. Технические средства

Для работы ТМОС необходима следующая минимальная конфигурация комплекса:

- процессор К1801ВМ1 или совместимые с ним К1801ВМ2, К1801ВМ3, К1801ВМ4;
- оперативная память емкостью не менее 56 кбайт;
- алфавитно-цифровой терминал;
- внешняя память на гибких магнитных дисках емкостью не менее 209 кбайт.

Дополнительно в состав технических средств могут входить:

- алфавитно-цифровое печатающее устройство;
- устройство формирования отображения графической информации на экране монитора;
- контроллеры телеграфных каналов (на 4-6 каналов);
- накопитель на жестких магнитных дисках типа винчестер.

2.2. Программные средства

Носитель тест-мониторной системы (см. приложение 1).

3. ОСНОВНЫЕ ПОНЯТИЯ

В данном разделе определены обозначения и термины, которые используются в документации ТМОС.

3.1. Символы и их функции

Допустимы все печатные символы русского и латинского

алфавитов, но некоторые символы при специфических условиях имеют специальное значение. Краткое описание функций символов приведено в табл.1.

Употребляемые в таблице термины определены ниже в соответствующих разделах.

Таблица 1

ФУНКЦИИ СИМВОЛОВ

Символ	Функция
1	2
ВК	Возврат каретки
ПС	Перевод строки
ЗБ	"Забой" - стирает последний символ
:	Ограничитель спецификации устройства (например, MY0:)
;	Начало строки комментария в командном файле
.	Разделитель трехсимвольного расширения имени файла
?	"Универсальный" символ - заменяет произвольный символ в спецификации файла (например, TEST.BI?)
*	"Универсальная" спецификация - заменяет группу символов в спецификации файла (например, TEST.*)
=	Разделитель входной и выходной спецификации в команде
<	Эквивалентен символу равно " = "
/	Ключ команды. Используется для модификации команды
@	Начало строки комментария в командном файле

Кроме того, в TMOC используются коды управляющих символов, приведенные в табл. 2. Управляющие символы вводятся одновременным нажатием клавиши <СУ> и клавиши символа.

Таблица 2

УПРАВЛЯЮЩИЕ СИМВОЛЫ

Символ	Функция
1	2
СУ/C	Прекратить выполняемые действия
СУ/U	Отменить вводимую строку, чтобы можно было ввести новую. Используется до нажатия клавиши <ВК>
СУ/S	Приостанов вывода на печать (на экран)
СУ/Q	Продолжение вывода после "СУ/S"
СУ/X	Продолжение выполнения командного файла
СУ/O	Подавление вывода текущего сообщения

1	2
	на экран терминала (в комплексном тесте)
	повторный ввод "СУ/О" возобновляет вывод на экран терминала "*"

3.2. Допустимые имена устройств

Каждому типу устройств в ТМОС соответствует драйвер.

В ТМОС приняты следующие мнемонические коды устройств:

LP - печатающее устройство;

DXN - накопитель на гибких магнитных дисках типа "ЭЛЕКТРОНИКА ГМД-7012";

MYN - накопитель на гибких магнитных дисках типа "ЭЛЕКТРОНИКА НГМД-6021", "ЭЛЕКТРОНИКА НГМД-6121";

MZN - накопитель на гибких магнитных дисках типа "СМ-5640", "ЕС-5088";

MTN - накопитель на магнитной ленте;

где N - номер устройства, к которому происходит обращение.

3.3. Имя и тип файла

Имя каждого файла состоит из шести символов имени программы и трех символов расширения (типа файла), разделяемых символом точка ".".

XXXXXX . XXX

Имя программы Расширение (тип файла)

Имя программы в системе формируется по следующим правилам:

1) Первый символ - буква, обозначает тип используемого процессора (см. приложение 2);

2) Для системных файлов:
- второй символ:
  M - для монитора
  D - для драйвера
- третий и четвертый символы определяют тип системного устройства (см. п. 3.2);
- пятый и шестой символы определяют номер версии программы;

3) для тестовых программ:
- второй и третий символы - буквы, обозначающие сокращенное наименование устройства;
- четвертый символ - буква, обозначающая символьный номер теста для данного устройства от A до Z;
- пятый символ - буква, обозначающая символьный номер версии для данного устройства от A до Z;
- шестой символ - цифра, обозначающая вариант, скорректированный по извещениям тестовой программы от 0 до 9.

4) Для текстовых и командных файлов имя программы произвольно.

Допускается использование следующих возможных расширений:

BIN - для программ в абсолютном формате;

BIC - для цепочечных программ в абсолютном формате;
SAV - для программ в формате отображения памяти;
OBJ - для программ в об'ектном формате;
TXT - для текстовых файлов;
CCC - текстовый формат командного файла; командный файл состоит из цепочки (последовательности) команд (см. раздел 8);
SYS - для системных файлов.

3.4. Справочник TMOC

Справочник TMOC, содержащийся в файле HELP.TXT, выводится на экран терминала или печатающее устройство по команде монитора H (H/L). Он содержит полный перечень команд указанных ниже программ, а также ключей и флагов программы обработки тестовых запросов.

Программы:
- монитор;
- обработка тестовых запросов;
- обработка файлов (UPD1);
- обработка файлов (UPD2);
- PATCH ;
- редактор текста (XTECO),

4. СОСТАВ ТЕСТ-МОНИТОРНОЙ СИСТЕМЫ

Таблица 3

ПЕРЕЧЕНЬ ФАЙЛОВ НА НОСИТЕЛЕ TMOC

Имя файла	Название программы
1	2
KMMY??.SYS	Монитор системы
KMMZ??.SYS	Монитор системы
KDMY??.SYS	Драйвер накопителя на гибком магнитном диске
KDMZ??.SYS	Драйвер накопителя на гибком магнитном диске
KDDX??.SYS	Драйвер накопителя на гибком магнитном диске "ЭЛЕКТРОНИКА ГМД-7012"
KDMT?.SYS	Драйвер накопителя на магнитной ленте
KSAA??.SYS	Программа обслуживания тестовых запросов
PATCH.BIN	Программа модификации файлов
KUDI??.SYS	Программа вывода каталога
UPD1.BIN	Обработка файлов 1
UPD2.BIC	Обработка файлов 2
XTECO.BIN	Редактор текста
SETUP.BIN	Программа построения таблиц связи
HELP.TXT	Справочник по системе
SYSTEM.CCC	Тестовая цепочка
SYSGEN.CCC	Файл генерации
TEST.TXT	Справочник по тестам

Кроме перечисленных файлов в состав тест-мониторной системы входят тестовые программы, набор которых определяется конфигурацией системы и может расширяться.

Описание тестов приведено в документе описание тестовых программ. Руководство оператора.

5. МОНИТОР СИСТЕМЫ

5.1. Назначение программы

Монитор - это программа, управляющая загрузкой и запуском системных, обслуживающих и тестовых программ.

Монитор позволяет:

- ввести команды с терминала;
- загрузить программу в оперативную память;
- запустить загруженную программу с указанного адреса;
- выполнить последовательность команд командного файла
- вывести на экран терминала или печать справочник TMOC.

5.2. Основные положения

В момент загрузки монитор занимает примерно 4к слов. Резидентная часть монитора составляет примерно 2к слов. Однако, младшие 0,5к слов монитора могут оказаться затертыми тестовой программой, и впоследствии они восстанавливаются монитором.

В программе монитора содержимое ячейки 1000 используется для информации о частоте сети. Если в 1000 ячейке "0", то принимается частота 50 ГЦ, любая другая величина определяет частоту 60 ГЦ. Состояние ячейки 1002 используется для инициации автоматического запуска командного файла SYSTEM.CCC в момент запуска монитора. Все параметры инициации монитора устанавливаются автоматически. Запуск командного файла происходит, если содержимое ячейки 1002 отлично от 0.

5.3. Выполнение программы

Считается, что пользователь хорошо знает логическую структуру периферийного устройства, используемого в качестве системного.

5.3.1. Загрузка монитора системы

Установите носитель с монитором TMOC в окно накопителя с номером 0.

Выполните процедуру начальной загрузки по одному из способов, зависящему от типа используемого загрузчика (см. приложение 3).

5.3.2. Запуск монитора системы

На экран терминала выводится имя монитора, об'ем имеющейся памяти (до 28к слов) и номер накопителя, с которого была произведена начальная загрузка.

Пример:

```
KMMYC1   TMOC   B2.00
Загрузка  с  устр. 0
24K  не  общая шина
```

Далее монитор просит ввести дату:

Дата (ДД - МММ - ГГ)

Дату вводят в следующем формате:

ДД - две цифры, определяющие число;

МММ - три символа, определяющие месяц (см. табл. 4);

ГГ - две цифры, определяющие год.

Таблица 4

МНЕМОНИЧЕСКИЕ ОБОЗНАЧЕНИЯ МЕСЯЦЕВ

Обозначение	Месяц
1	2
JAN	январь
FEB	февраль
MAR	март
APR	апрель
MAY	май
JUN	июнь
JUL	июль
AUG	август
SEP	сентябрь
OCT	октябрь
NOV	ноябрь
DEC	декабрь

По желанию можно дату не вводить, для этого достаточно нажать клавишу <ВК>.
В этом случае будет принята дата 01-JAN-70. После того, как дата воспринята, монитор выведет на экран адрес рестарта, который можно использовать для повторного запуска монитора после останова процессора:

```
РЕСТАРТ : 152010
TMOC В2.00 Введите "Н", "Н/L" для получения справки
```

После успешного выполнения всех вышеперечисленных операций монитор напечатает точку ".". Это означает, что монитор ждет ввода команд от оператора.

5.4. Команды монитора

В данном подразделе подробно описаны следующие команды:

R - загрузка и запуск программы;

L - загрузка программы;

S - запуск программы;

C - выполнение командного файла;

D - распечатка каталога системного носителя;

E - назначение другого накопителя в качестве системного устройства;

H - вывод справочника TMOC;

TEST - выполнение командного файла с именем SYSTEM.CCC.

ПРИМЕЧАНИЕ. Следует отметить, что команды монитора оперируют с теми файлами, которые размещены на системном носителе.

Для некоторых команд имеются ключи, которые представля-

ют собой набор символов, перед которым стоит знак "/". Ключи используются для модификации команд.

В описываемых ниже командах необязательные в команде поля заключены в квадратные скобки "[ ]". Все команды завершаются нажатием клавиши <ВК>.

5.4.1. Команда R используется для загрузки и запуска программы, хранящейся на системном устройстве. Команда R является комбинацией описанных ниже команд L и S.

Формат команды: R имя файла[.тип файла] [адрес]

где

имя файла – имя файла в абсолютном формате;

тип файла – .BIN или .BIC; по умолчанию тип файла принимается .BI? ; если на носителе имеются два файла с одинаковым именем и двумя расширениями .BIN и .BIC, то будет загружаться найденный первым файл с данным именем.

адрес – стартовый адрес программы; если адрес не указан в командной строке, то программа будет запущена либо по указанному в ней адресу передачи управления, либо по адресу 200(восьмеричное) – при отсутствии такового.

Чтобы показать, какая из нескольких возможно присутствовавших на носителе программ загружена, после загрузки программы, но до ее запуска, на экране терминала будет распечатано ее имя и расширение.

5.4.2. Команда L используется для загрузки программы в абсолютном формате с системного носителя в оперативную память.

Формат команды: L имя файла[.тип файла]

где тип файла – BIN или BIC

После загрузки на экране терминала распечатывается полное имя загруженной программы.

5.4.3. Команда S используется для запуска программы, которая была загружена ранее по команде L. Между командами L и S не должно вводиться никаких команд.

Формат команды: S [адрес]

где адрес – адрес запуска программы.

При отсутствии адреса запуска в командной строке, монитор запускает программу либо с указанного в ней адреса передачи управления, либо с адреса 200(восьмеричное) – при отсутствии такового.

5.4.4. Команда C используется для запуска командного файла.

Формат команды: C имя файла[/ключи]

где имя файла – имя командного файла типа .CCC

ключи – модификатор команды.

Описание ключей приведено в пункте 8.4.1.

Командный файл должен находиться на системном носителе. Подробное описание командного файла приводится в разделе 8.

5.4.5. Команда D используется для выдачи на экран тер-

минала или печатающее устройство списка всех файлов системного устройства.

Формат команды: D[/ключи]

/L - вывод каталога на печатающее устройство;

/F - вывод каталога в краткой форме.

Допускается одновременное использование ключей /L и /F.

Полная форма каталога:

ENTRY#	FILENAM.EXT	DATE	LENGTH	START
1	KMMYC1.SYS	Ø1-JAN-87	17	5Ø
2	KMMYB1.SYS	Ø1-JAN-87	3	71
3	KUDIBØ.SYS	Ø1-JAN-87	5	74
.	.....	.....	.	..
.	.....	.....	.	..
.	.....	.....	.	..

где

ENTRY - порядковый номер файловой записи;

FILENAM.EXT - спецификация файла (имя и тип файла);

DATE - дата создания файла;

LENGTH - длина файлов в блоках;

START - номер первого блока файла.

Краткая форма каталога:

ENTRY#	FILENAM.EXT
1	KMMYC1.SYS
2	KDMYB1.SYS
3	KUDIBØ.SYS
.	.....
.	.....
.	.....

Непрерывные файлы отмечаются в каталоге буквой "C", расположенной за датой.

При получении команды D монитор должен загрузить служебную программу KUDI??.SYS, ведущую каталог, которой в свою очередь для работы необходим драйвер KDMY??.SYS. Для того, чтобы команда вывода каталога работала, оба этих файла должны присутствовать на системном носителе, иначе монитор напечатает сообщения об ошибке.

5.4.6. Команда E позволяет заменить номер системного устройства на номер любого из однотипных устройств.

Формат команды: E N

где N - новое значение номера накопителя.

Например, если оператор произвел начальную загрузку системы с накопителя Ø, а затем возникла необходимость, чтобы монитор обращался к накопителю 1 как к системному, то он может это сделать не перезагружая систему при помощи команды E, набрав E 1.

5.4.7. Команда H предназначена для получения краткой информации по системе TMOC.

Формат команды: H[/L]

Содержимое файла под именем HELP.TXT (см. п. 3.4) выводится на экран терминала или на печатающее устройство.

Этот файл должен присутствовать на системном носителе.

Для вывода информации на печать существует ключ /L.

5.4.8. Команда TEST предназначена для запуска командного файла с именем SYSTEM.CCC, который должен находиться на системном устройстве.

Формат команды: TEST

5.5. Сообщения об ошибках

В процессе работы монитор распознает ряд ошибок и выдает о них сообщения на экран терминала. Все сообщения об ошибках начинаются с вопросительного знака "?".

? INVALID COMMAND
- неправильная команда. Пользователь ввел команду, которую монитор не может распознать. Необходимо проверить командную строку и ввести ее повторно.

? INVALID FILENAME
- неправильное имя файла. В заданной команде неверно указано имя файла, задан неверный формат имени (синтаксическая ошибка).

? BAD ADDR
- несуществующий адрес. В команде R (прогон) или S (запуск) был указан неверный адрес.

? CKERR
- ошибка контрольной суммы. Драйвер монитора по мере загрузки файла считает его контрольную сумму. Если вычисленная контрольная сумма не совпадает с записанной в файле, монитор выдает сообщение об ошибке контрольной суммы. Можно повторить загрузку, но, если ошибка повторяется, вероятно данный файл запорчен.

? NOT FOUND:имя файла
- ошибка поиска. Заданный монитором файл не обнаружен. Возникновение данной ошибки возможно либо при задании неверного имени файла, либо из-за отсутствия файла на системном носителе. Следует помнить, что оператор может читать файлы только с системного устройства.

? RD ERR
- ошибка чтения. При чтении с системного носителя возникла аппаратная ошибка. Повторите операцию, если ошибка повторится, то возможны либо неисправность устройства, либо содержимое носителя испорчено.

6. ОБРАБОТКА ФАЙЛОВ

6.1. Назначение и характеристики программ UPD1 и UPD2

Обработка файлов - это общее название двух служебных программ обработки файлов - UPD1 и UPD2.

Программа UPD1 предназначена для загрузки, модификации и удаления файлов. Она занимает 4к слов оперативной памяти в области старших адресов (но не выше 28к).

ПРИМЕЧАНИЕ. После завершения работы с программой UPD1 требуется перезагрузить монитор системы.

Служебная программа UPD2 предназначена для генерации и копирования версий системы, модификации, копирования, удаления файлов, корректировки двоичного текста программ и

других служебных функций. Она занимает 6к слов оперативной памяти в области младших адресов, начиная с адреса 1000.

6.2. Выполнение программ

Программа "Обработка файлов" использует оперативную память емкостью не менее:

- 8к слов для программы UPD1;
- 16к слов для программы UPD2.

6.2.1. Загрузка

1. Загрузите и запустите монитор TMOC (см. п. 5.3).
2. Подайте команду R UPDN <ВК> (N = 1,2)

Происходит загрузка и запуск программы "Обработка файлов".

6.2.2. Начальный запуск

После загрузки и запуска программы "Обработка файлов" на экран терминала выводится сообщение:

для UPD2:	для UPD1:
UPD2 TMOC B2.02	UPD1 TMOC B2.02
Рестарт: 103714	Рестарт: 003714

После повторного запуска на экран терминала выводится звездочка "*".
Звездочка означает готовность принимать команды с терминала.

6.3. Команды оператора

Все функции программ UPD1 и UPD2 реализуются посредством команд, заданных с клавиатуры терминала или из командного файла. Ниже приводятся все допустимые команды программ UPD1, UPD2 и их формат.

Программа "Обработка файлов" при задании в команде имени и типа формата файла допускает использование специальных знаков:
"?" (вопросительный знак), "*" (звездочка) и "%" (процент).

1) Использование вопросительного знака и процента эквивалентно понятию "любой символ в данной позиции".

```
П р и м е р:
  MY1:TECT.?IN ; обращение к
               ; файлам на MY1: с именем тест,
               ; тип формата которых оканчивается
               ; на IN
```

2) Использование звездочки эквивалентно понятию "любой символ, начиная с данной позиции".

```
П р и м е р:
  MY1:ABC.*    ; последовательное обращение
               ; ко всем файлам на MY1: с именем ABC.
```

6.3.1. Команда DIR (UPD2) предназначена для вывода на экран терминала или печатающее устройство каталога или списка файлов с указанного носителя или для сохранения каталога файлов.

Формат команды:

```
DIR Выхус:имя.тип/ключ=вхус:имя.тип/ключ <ВК>
```

где выхус – выходное устройство; указывается лишь в том случае, если требуется сохранить ка-

талог в виде файла с указанным именем на несистемном носителе.

вхус – устройство, каталог которого необходимо получить.

Если выходное устройство не указано, но задано имя выходного файла, то каталог в виде файла с указанным именем будет сохранен на системном носителе.

Если не указаны выходное устройство и имя выходного файла, но задан разделитель "=", то каталог будет выведен на системный носитель в виде файла с именем DIR.TXT, присвоенным по умолчанию.

Если указано лишь входное устройство, то его каталог будет выведен на экран терминала.

Ключи команды DIR:

/F – разрешение вывода краткой формы каталога;
/B – разрешение вывода числа свободных блоков;
/L – разрешение вывода каталога на печать;
/Q – запрет перемотки ленты на входном устройстве при поиске файла, если ключ указан для входного устройства;
– запрет перемотки ленты на выходном устройстве при записи каталога в виде файла,если ключ указан для выходного устройства(только для накопителя на магнитной ленте).

Допускается совместное использование ключей.

П р и м е р ы:

```
DIR                     Вывод на экран терминала каталога
                        всех файлов системного устройства.
DIR MY0:                Вывод на экран терминала каталога
DIR *.BIN/L             Вывод на печатающее устройство
                        списка всех файлов,  имеющих
                        тип файла .BIN, расположенных на
                        системном устройстве.
DIR MY1:*.BI?           Вывод на экран терминала списка
                        всех файлов устройства MY1,
                        тип файла которых начинается
                        с символов ".BI".
DIR ZTS???.BI?          Вывод на экран терминала системного
                        устройства списка всех файлов, имена
                        которых начинаются с символов "ZTS",
                        а тип файла – с символов ".BI".
DIR MY1:DIR.TXT=MY0:ZR????.*
                        Запись на MY1 списка файлов с устройства
                        MY0 с именами, начинающимися
                        с символов "ZR", и любыми типами файлов.
                        Список файлов на MY1 определяется в виде
                        файла с именем DIR.TXT.
DIR =MY1:               Вывод на системное устройство каталога
                        устройства MY1. Каталог выводится в
                        виде файла с именем DIR.TXT.
DIR MY0:/F/L            Вывод на печатающее устройство
                        каталога всех файлов в краткой форме.
```

6.3.2. Команда DEL (UPD1, UPD2) предназначена для удаления файла с носителя.

Формат команды: DEL устр:имя.тип/ключ <ВК>

По умолчанию используется системное устройство.

Эта команда может использоваться для удаления списка файлов или всех файлов на носителе.

П р и м е р :

DEL MY1: *.BIN

- удаляются с устройства все файлы, имеющие расширение .BIN.

Ключи команды DEL:

/N - запрет печати имен файлов по мере их удаления.

/Q - запрет перемотки ленты перед началом поиска указанного файла (только для накопителя на магнитной ленте).

6.3.3. Команда REN (UPD2) предназначена для изменения имени файла на внешнем устройстве; при этом изменяется только имя файла в каталоге, дата создания файла остается прежней.

Формат команды:

REN устр:новое имя.тип=устр:старое имя.тип <ВК>

Устройство должно быть одним и тем же. По умолчанию используется системное устройство.

П р и м е р :

REN TESOZY.BIN=TECT1.BIN

- изменить имя файла TECT1.BIN на TESOZY.BIN

6.3.4. Команда PIP (UPD2) предназначена для копирования файлов с одного внешнего устройства на другое.

Формат команды:

PIP выхус:имя.тип/ключ=вхус:имя.тип/ключ <ВК>

Ключи команды PIP:

/N - запрет печати имен файлов по мере копирования.

/Q - запрет перемотки ленты перед копированием (только для накопителя на магнитной ленте).

Имена файлов на входном и выходном устройствах могут быть различными.

ПРИМЕЧАНИЕ. В команде PIP запрещено автостирание, т.е. удаление с устройства файла, имя и расширение которого совпадают с именем и расширением копируемого файла. Поэтому, если предпринимается попытка скопировать такой файл, то печатается предупреждающее сообщение:

имя.тип ? FILE ALREADY EXISTS

"имя.тип ? Файл уже присутствует"

Копирование прерывается. Однако, если в качестве выходного устройства используется накопитель на магнитной ленте и в команде PIP указан ключ /Q для выходного устройства, то проверка на совпадение имен не производится и может произойти дублирование имени.

Двоичные файлы после копирования с помощью команды PIP желательно прочитать с помощью команды READ для проверки контрольных сумм. Проверку записи символьных файлов можно выполнить, распечатав их с помощью команд TYPE или PRINT.

6.3.5. Команда FILE (UPD2) предназначена для копирования файлов с одного устройства на другое.

Формат команды:

FILE выхус:/ключ=вхус:имя.тип/ключ <ВК>

Команда FILE аналогична команде PIP.

Ее отличие заключается в том, что имя копируемого файла выходного устройства обязательно совпадает с именем файла входного устройства, в то время как при копировании по команде PIP эти имена могут быть различными.

Команда COPY(UPD2) предназначена для копирования файлов или носителей. Копирование разрешается только между однотипными устройствами. Копирование может происходить в двух режимах:

- копирование всего содержимого носителя;
- пофайловое копирование.

Формат команды: COPY выхус:=вхус: <ВК>

После ввода команды программа UPD2 выдаст предупреждающее сообщение:

USER DATA ON OUTPUT DEVICE WILL BE DESTROYED!
PROCEED? (Y/N/CR=N)

"Информация на выходном устройстве будет потеряна!"
"Продолжать? (Да/нет/<ВК>=нет)"

Ответы:

<ВК> или N<ВК> - процедура прерывается и программа переходит в режим команд.

Y<ВК> - диалог будет продолжен и программа задаст вопрос:

FILE COPY OR IMAGE COPY ? (I/F)
"Режим копирования ? (I/F)"

Ответы:

I - копирование всего содержимого носителя будет происходить поблочно; после завершения процедуры копирования содержимое выходного устройства считывается для проверки контрольных сумм.

F - пофайловое копирование.

Указанную операцию копирования с однотипных устройств можно выполнить командами FILE и PIP, однако при копировании по команде PIP одноименные файлы не будут потеряны.

6.3.6. Команда CLR (UPD1, UPD2) предназначена для записи нулей в буфер программы UPD1 или UPD2 и для очистки программного буфера перед загрузкой очередной программы.

Формат команды: CLR <ВК>

Программа выходит в режим ожидания команды, на экране появляется звездочка "*".

6.3.7. Команда LOAD (UPD1, UPD2) выполняет загрузку файла в абсолютном двоичном формате в оперативную память устройства, указанного в команде.

Формат команды: LOAD устр:имя.тип/ключ <ВК>

Ключи команды LOAD:

/N - запрет печати верхней и нижней границ памяти и имени найденного файла (если применяются универсальные символы).

/Q - запрет перемотки ленты перед поиском заданного файла (только для накопителя на магнитной ленте).

После выполнения команды LOAD программа напечатает ад-

рес передачи управления, верхнюю и нижнюю границы загруженной программы:

XFR: АДР1 CORE: АДР2 АДР3

где АДР1 - адрес старта программы;
АДР2 - нижний адрес загрузки;
АДР3 - верхний адрес загрузки.

6.3.8. Команда DUMP (UPD1, UPD2) предназначена для записи содержимого программного буфера в файл на носителе. Размер буфера задается нижней и верхней границами памяти, которые установлены при выполнении команды LOAD и могут быть изменены командами HICORE и LOCORE.

Формат команды: DUMP устр:имя.тип/ключ <ВК>

По умолчанию используется системное устройство. Файлу будет присвоено имя, указанное в команде и дата создания, введенная при загрузке монитора.

Ключ команды:

/Q - запрет перемотки ленты на устройстве перед выводом файла (только для накопителя на магнитной ленте).

6.3.9. Команда XFR (UPD1, UPD2) предназначена для индикации и изменения адреса старта загруженной программы. Этот адрес определяется командой CORE.

Формат команды:

XFR <ВК>
АДР1 АДР2

где АДР1 - старый стартовый адрес;
АДР2 - новый стартовый адрес программы.

П р и м е р ы :

XFR <ВК>	индикация стартового адреса;
1000 <ВК>	адрес старта равен 1000.
XFR <ВК>	индикация и изменение стартового
1000 1100 <ВК>	адреса программы.
	1000 - прежний стартовый адрес,
	1100 - новый стартовый адрес.

6.3.10. Команда MOD (UPD1, UPD2) предназначена для индикации и изменения содержимого ячеек памяти, занятых программой.

Формат команды: MOD АДР <ВК>

где АДР - восьмеричный адрес памяти.

В ответ на команду MOD программа выводит на экран терминала содержимое ячейки с указанным адресом в восьмеричном виде: АДР ССССС

где ССССС - восьмеричное содержимое ячейки с адресом АДР.

Если требуется изменить содержимое ячейки, то следует ввести новое значение в восьмеричном виде:

АДР ССССС НННННН <ВК>

где АДР - адрес ячейки,
ССССС - старое содержимое ячейки,
НННННН - новое содержимое ячейки.

Если необходимо изменить содержимое нескольких ячеек с

последовательными адресами, то вместо клавиши <ВК> следует использовать клавишу <ПС>.

```
    П р и м е р :
    MOD 1000,XXXXXX YYYYYY <ПС>
        1002 XXXXXX YYYYYY <ПС>     изменить содержимое ячеек
        1004 XXXXXX YYYYYY <ПС>     памяти с 1000 по 1006
        1006 XXXXXX YYYYYY <ВК>
где     XXXXXX - старое содержимое ячеек;
        YYYYYY - новое содержимое ячеек.
```

6.3.11. Команда CORE (UPD1, UPD2) предназначена для индикации границ загрузки программы в памяти, загруженной по команде LOAD. После загрузки программы по команде LOAD адреса нижней и верхней границ программы печатаются по умолчанию.

```
     Формат команды:   CORE  <ВК>
     П р и м е р :
    CORE  <ВК>
    000000  014776  <ВК>
где   000000  -  нижний адрес загрузки;
      014776  -  верхний адрес загрузки.
```

6.3.12. Команда LOCORE (UPD1, UPD2) предназначена для индикации и изменения нижнего адреса загрузки программы в оперативной памяти. После выполнения команды LOAD значение этого адреса печатается по умолчанию.

```
      Формат команды:     LOCORE  <ВК>
        П р и м е р ы :
    LOCORE  <ВК>              индикация нижнего адреса
    000000  <ВК>              загрузки программы
    LOCORE  <ВК>              индикация и изменение нижнего
    000000  1000  <ВК>        адреса загрузки программы
```

6.3.13. Команда HICORE (UPD1, UPD2) предназначена для индикации и изменения верхнего адреса загрузки программы в оперативной памяти. После выполнения команды LOAD значение этого адреса печатается по умолчанию.

```
      Формат команды:  HICORE  <ВК>
        П р и м е р ы :
    HICORE  <ВК>              индикация верхнего адреса
    10000  <ВК>               загрузки программы;
    HICORE  <ВК>              индикация и изменение верхнего
    10000  11000  <ВК>        адреса загрузки программы.
```

6.3.14. Команда ZERO (UPD2) предназначена для инициализации носителя в формате системы, т.е. очищает разрядные шкалы (для устройств с произвольным доступом) и помещает на носитель пустой каталог.

```
      Формат команды:    ZERO УСТР:  <ВК>
      В Н И М А Н И Е !  после выполнения команды ZERO
                         вся информация на носителе
                         теряется безвозвратно!
```

При введении этой команды печатается предупреждающее сообщение:

```
    UZER DATA ON MYN WILL BE DESTROYED!
    PROCEED? (Y/N/CR=NO)
```

"Информация на MYN будет уничтожена!"
"Продолжать? (Да/нет/<ВК>=нет)"

Пользователь должен подтвердить, что необходима операция обнуления.

Если указанное устройство является системным, то будет выдано дополнительное предупреждающее сообщение о том, что ему может понадобиться дополнительный драйвер для устройства, с которого он будет восстанавливать инициированное системное устройство:

ZERO SYSTEM DEVICE
YOU MAY NEED AN ADDITIONAL DRIVER
PROCEED? (Y/N/CR=NO)
"Инициируется системное устройство"
"Может потребоваться дополнительный драйвер"
"Продолжать? (Да/нет/<ВК>=нет)"

Если системное устройство и устройство, с которого оно будет восстановлено, являются однотипными, то второй драйвер не нужен. Если устройства разнотипные, то на запрос PROCEED? следует ответить N<ВК>. После этого следует загрузить дополнительный драйвер с помощью команды DRIVER и снова использовать команду ZERO.

6.3.15. Команда INIT (UPD2) аналогична команде ZERO.

Формат команды: INIT устр: <ВК>

6.3.16. Команда DRIVER (UPD2) предназначена для загрузки с системного устройства в память драйверов внешних устройств.

Формат команды: DRIVER ДР1:[/ДР2:] <ВК>

где ДР1, ДР2 – двухсимвольный мнемонический код устройств, заканчивающийся символом ":". Код второго устройства отделяется от первого символом "/".

Можно загружать не более двух драйверов. Обычно эта команда не используется, так как драйверы загружаются автоматически в процессе исполнения других команд программ UPD1, UPD2, но при выполнении команды ZEROможет возникнуть ситуация, в которой потребуется загрузка в память дополнительного драйвера (см. описание команды ZERO).

Пример:

DRIVER DX:/MT: в память загружаются драйверы гибкого диска и магнитной ленты

6.3.17. Команда SAVM (UPD2) предназначена для записи на носитель монитора, загружаемого ранее по команде LOAD. Она используется для устройств с произвольным доступом (например, гибкие диски).

Формат команды: SAVM устр: <ВК>

Эта команда помещает в блок загрузчика системы соответствующий первичный загрузчик и записывает на носитель в заранее отведенную секцию файл монитора.

Пример:

*LOAD MY1:KMMYB1.SYS <ВК> загрузка монитора в оперативную память с устройства
*SAVM MY0: <ВК>

MY1 и запись его на устройство MY0.

6.3.18. Команда SAVE (UPD2) аналогична команде SAVM, но используется для устройств с последовательным доступом (например, магнитные ленты).

Формат команды: SAVE устр: <ВК>

6.3.19. Команда ASG (UPD2) предназначена для присвоения устройству логического номера.

Формат команды: ASG устр:=N <ВК>

где N - логический номер устройства.

После этого можно обращаться к устройству по его номеру.

Пример:

*ASG MY1:=1 <ВК> — устройству MY1 присваивается логическое имя 1;

PIP 1:=.BIN <ВК> — копировать все файлы с расширением BIN с системного устройства на устройство MY1, имеющее логическое имя 1.

6.3.20. Команда READ (UPD2) предназначена для проверки файлов и носителей.

Формат команды: READ устр:имя.тип/ключ <ВК>

Ключи команды READ:

/N - не печатать имя файла по мере его прочтения;

/Q - не производить перемотку ленты перед поиском указанных файлов (только для накопителей на магнитной ленте).

Каждый блок указанного файла читается в память, и для него вычисляется контрольная сумма, которая сравнивается с суммой, записанной в файле.

Пример:

READ MY1:*.* — прочитать все файлы с MY1

6.3.21. Команда EOT (UPD2) предназначена для записи метки "Конец ленты" в текущую позицию на магнитной ленте.

Формат команды: EOT устр: <ВК>

Процедура записи метки состоит в следующем: считывается файл, после которого пользователь хочет поместить метку, а затем используется команда EOT. Файлы, расположенные за меткой "Конец ленты", становятся недоступными.

Пример:

*READ MT0:ZQRADO.BIN <ВК> — считать с магнитной ленты файл с именем ZQRADO.BIN и после него поместить метку "Конец ленты"

*EOT MT0: <ВК>

По умолчанию принимается системное устройство.

6.3.22. Команда DO (UPD2) предназначена для выполнения командного файла, содержащего одну или несколько команд программы UPD2, за исключением команды EXIT.

Формат команды: DO имя.тип <ВК>

Файл должен быть расположен на системном устройстве. Он создается с помощью программы "Редактор текстов XTECO".

6.3.23. Команда TYPE (UPD2) предназначена для вывода текстового файла на экран терминала.

Формат команды: TYPE устр:имя.тип/ключ <ВК>

Ключи команды TYPE:

/Q - не производить перемотку ленты перед поиском указанного файла (для накопителя на магнитной ленте).

По умолчанию принимается системное устройство.

При просмотре текста можно приостановить вывод на экран терминала, нажав <СУ>/<S>, и продолжить, нажав <СУ>/<Q>.

6.3.24. Команда PRINT (UPD2) предназначена для вывода текстового файла на печатающее устройство.

Формат команды: PRINT устр:имя.Тип <ВК>

По умолчанию принимается системное устройство.

6.3.25. Команда BOOT (UPD1, UPD2) предназначена для загрузки и запуска монитора с устройства, не являющегося системным.

Формат команды: BOOT устр:

На устройстве должен быть установлен носитель, с которого возможна начальная загрузка. Процесс начальной загрузки состоит в чтении нулевого физического блока (блока загрузки) в первые 256 (десятичное) слов памяти и запуске с адреса 000000.

Устройство, с которого произведена начальная загрузка, становится системным.

6.3.26. Команда EXIT (UPD2) возвращает управление программе монитора.

Формат команды: EXIT <ВК>

6.4. Сообщения об ошибках программ UPD1 и UPD2

?RD ERR ON INPUT DEVICE
- неисправно устройство ввода.

?WR ERR ON OUTPUT DEVICE
- неисправно устройство вывода.

?RD ERR ON INPUT DEVICE DIRECTORY
- ошибка при обращении к каталогу устройства ввода.

?WR ERR ON OUTPUT DEVICE DIRECTORY
- ошибка при обращении к каталогу устройства вывода.

?RD ERR WHILE LOADING DRIVER FOR DEVICE
- ошибка устройства при загрузке драйвера устройства.

?RD ERR WHILE READIND [имя файла]
- ошибка произошла при чтении указанного файла.

? INVALID DEVICE
- в команде неверно указано устройство. Появление этой ошибки может быть вызвано рядом причин в зависимости от используемой команды:
- для команд DIR, REN, DEL, COPY, READ, ZERO, SAVE, SAVM устройство не имеет файловой структуры.
- для команд SAVE и EOT устройство не является магнитной лентой.
- для команд COPY и SAVM устройство не является

устройством с произвольным доступом.

- для команды COPY указанные в команде устройства не являются однотипными.

? INVALID FILNAME,

- неверное имя файла.Имя файла, указанное в последней команде, не является допустимым. В именах запрещается использование всех символов, кроме букв латинского алфавита и арабских цифр. Случаи применения символов "*", "%" и "?" описаны в подразделе 3.1.

? INVALID ADDRES

- ошибочный адрес. Адрес должен быть четным и находиться в пределах границ загрузки.

? INVALID COMMAND

- ошибочная или недопустимая команда.

? INVALID SWITCH

- недопустимый ключ. В указанной команде данный ключ не используется.

? INVALID NUMBER

- недопустимое число. Число не является восьмеричным.

? NON-EXISTENT FILE

- несуществующий файл.

? DIRECTORY FULL

- каталог устройства заполнен.

? SPECIFY DEVICE

- укажите устройство. В данной команде необходимо указать устройство.

? DEVICE FULL

- носитель заполнен. При записи была превзойдена емкость устройства вывода. Для устройств с произвольным доступом это означает, что для размещения всего файла не хватает физических блоков; для устройств с последовательным доступом был достигнут маркер конца ленты EOT при записи файла.

? NOT FOUND KDXX??.SYS

- не был обнаружен драйвер устройства "XX".

? UNEXPECTED END-OF-FILE

- рассматриваемый файл испорчен.

? SYNTAX ERROR

- синтаксическая ошибка. Последняя команда была введена неправильно.

? CHECKSUM ERR

- ошибка контрольной суммы при выполнении загрузки.

? OVERFLOW

- для существующей буферной области программа имеет слишком большие размеры. Для загрузки используйте программу UPD1.

? LOGICAL DEVICE NOT ASSIGNED

- в команде использовано логическое имя, которое не присваивалось устройству.

7. ПРОГРАММА PATCH

7.1. Назначение программы

Программа PATCH может быть использована для модификации

любого файла в двоичном формате (BIN или BIC). Ее можно использовать вместо последовательности команд LOAD-MOD-DUMP (загрузка - модификация - перепись содержимого памяти) в программах UPD1 и UPD2.

Программу PATCH следует использовать в двух случаях:

- для модификации файлов, которые не помещаются в буфере программы UPD2 или UPD1.
- для модификации мультипрограммных тестов комплексов, созданных конфигуратором-компоновщиком DXCL.

7.2. Работа программы

Работа программы проходит в два этапа:

1) На первом этапе строится входная таблица, содержащая перечень изменений, которые будут внесены в исправляемый файл. В таблицу вносят адреса, которые необходимо модифицировать и новое содержимое этих адресов. Таблицу можно использовать однократно или сохранить в виде файла на носителе для последующего использования.

2) На втором этапе двоичная информация в файле замещается двоичной информацией из таблицы, в результате чего получается модифицированный файл. Исходный файл сохраняется неизмененным.

7.3. Пуск программы

Для того, чтобы загрузить и запустить программу PATCH, необходимо ввести следующую команду монитора:

R PATCH <ВК>

Программа стартует и выведет на экран терминала свое имя и символ "*", означающий, что программа готова к приему команд.

В программу PATCH входят следующие команды:

BOOT - начальная загрузка системы с указанного устройства;
CLEAR - очистка "входной таблицы";
EXIT - возврат в монитор;
GETM - загрузка "карты загрузки" с заданного устройства;
GETP - загрузить "входную таблицу" с заданного устройства;
KILL - стереть адрес из "входной таблицы";
MOD - внести адрес во "входную таблицу";
PATCH - создать модифицированный файл;
SAVP - вывести "входную таблицу" на указанное устройство;
TYPE - вывести "входную таблицу" на экран терминала.

Для того, чтобы при помощи программы PATCH модифицировать файл, нужно выполнить две операции:

1) Необходимо построить "входную таблицу", содержащую все адреса файла, которые необходимо модифицировать, а также новое содержимое этих адресов. Число строк во "входной таблице" не должно превышать 50.

Для построения "входной таблицы" можно использовать следующие команды: CLEAR, GETM, MOD, TYPE, KILL, SAVP и GETP.

2) После того, как завершено построение "входной таблицы" для модификации в нужном файле ячеек, описанных во "входной таблице", необходимо использовать команду PATCH.

ПРИМЕЧАНИЕ. Важно отметить, что модифицируемый файл никогда полностью не загружается в память.

7.4. Команды программы PATCH

7.4.1. Команда CLEAR стирает все строки во "входной таблице".

Формат команды: CLEAR <ВК>

7.4.2. Команда GETM используется только при модификации мультипрограммных тестов комплексов. Команда GETM вызовет по заданному имени с заданного устройства "карту загрузки" и загрузит ее в память.

Для того, чтобы полностью использовать возможности команды MOD для модификации мультипрограммных тестов комплексов, необходимо иметь файл, так называемый "карта загрузки". "Карта загрузки" создается командой LINK в режиме конфигуратора-компоновщика DXCL. "Карта загрузки" представляет собой таблицу символов, генерируемую во время компоновки мультимультипрограммного примера комплекса.

Без "карты загрузки" команда MOD в качестве аргументов будет воспринимать только физические адреса, но если "карта загрузки" загружена (командой GETM), команда MOD будет воспринимать имена модулей (как мониторных так и дополнительных) в качестве аргументов.

Если у вас не будет соответствующей "карты загрузки", вам придется вручную высчитывать физический адрес соответствующий относительному адресу в модуле и вводить его в качестве аргумента команды MOD.

Формат команды: GETM устр:имя файла.MAP <ВК>

По умолчанию принимается системное устройство.

7.4.3. Команда MOD используется для записи во "входную таблицу" адреса и его желаемого содержимого.

Команда модификации MOD имеет два режима работы:

1) Работа с двоичными файлами, не относящимися к подсистеме проверки комплекса в мультипрограммном режиме работы;

2) Работа с двоичными файлами, относящимися к подсистеме проверки комплекса в режиме мультипрограммной работы.

7.4.3.1. Работа с двоичными файлами, не относящимися к подсистеме проверки комплекса в мультипрограммном режиме работы

MOD <АДРЕС> <ВК>

- внести адрес во входную таблицу;

<АДРЕС> - 16-разрядный физический адрес памяти.

После нажатия <ВК>, запрошенный адрес будет напечатан еще раз, а за ним будет выведена косая черта. Если этого адреса ранее не было во "входной таблице", за ним будут выведены шесть дефисов. Т.к. файла, который будет модифицироваться, нет в памяти, текущее значение указанного адреса узнать нельзя. Если адрес уже был введен во "входную таблицу", то после косой черты будет выведено содержимое этого адреса.

Пример 1:

```
*MOD 123456 <ВК>      ;оператор задал для модификации
 123456/------        ;адрес, не существующий во
                      ;"входной таблице";
```

```
*MOD 122222 <ВК>      ;оператор задал для модификации
 122222/000240        ;адрес,     существующий     во
                      ;"входной таблице".
```

После того, как вы ввели адрес во "входную таблицу", можно вводить то значение, которое вы хотите занести по этому адресу. После ввода значения, вы можете нажать либо <ВК>, либо <ПС>. В первом случае строка этого адреса закрывается, и на консоль выдается "*". При нажатии <ПС>, текущая строка таблицы будет закрыта и открыта новая строка для следующей адресуемой ячейки, т.е. <АДРЕС>+2. Затем можно вводить содержимое следующего адреса.

```
     П р и м е р 2:
*MOD 123456 <ВК>              ;оператор указал, что
 123456/------000240 <ВК>     ;содержимое 123456
*                             ;ячейки будет 240
*MOD 123456 <ВК>              ;оператор указал, что
 123456/000240 000137 <ПС>    ;нужно внести в таблицу
 123460/------ 000005 <ВК>    ;содержимое ячеек
*                             ;123456 и 123460
```

7.4.3.2. Работа с двоичными файлами, относящимися к подсистеме проверки комплекса в режиме мультипрограммной работы

При работе с файлами, относящимися к подсистеме проверки комплекса в режиме мультипрограммной работы, команда MOD имеет три различных формата:

```
        1)      MOD <АДРЕС> <ВК>
                - внести адрес во входную таблицу.
   <АДРЕС> -  18-разрядный физический адрес памяти.
```

Формат 1 команды MOD тот же, что и при работе с файлами, не относящимися к подсистеме мультипрограммной работы.

```
        2)      MOD MON <ИМЯ МОДУЛЯ><АДРЕС>
                - внести адрес во входную таблицу.
   <ИМЯ МОДУЛЯ> - имя мониторного модуля.
   <АДРЕС>       - относительный адрес памяти. Адрес
                   указывается относительно начального
                   адреса модуля.
```

Формат 2 используется для модификации ячеек в мониторных модулях.

```
     П р и м е р 1:
*MOD MON TEST1 24 <ВК>
 122224/------
```

```
        3)      MOD <ДОП.МОД.> <АДРЕС>
                - внести адрес во входную таблицу.
                  Адрес указывается относительно
                  начального адреса модуля.
  <ДОП.МОД.> - имя дополнительного модуля.
  <АДРЕС>     - относительный адрес памяти.
```

Формат 3 используется для модификации ячеек в дополнительных модулях.

```
     П р и м е р 2:
*MOD RXAE 10 <ВК>
 123444/------
```

Во всех трех форматах после нажатия <ВК> будет напечатан задаваемый таким образом физический адрес, за которым

выводится косая черта. Если этот адрес ранее в таблицу не вносился,то за косой чертой выводятся шесть дефисов "------". Если адрес уже вводился во "входную'таблицу", после косой черты будет напечатано ранее введенное его содержимое.

7.4.4. Команда KILL удаляет строку из входной таблицы.
Формат команды: KILL <АДРЕС> <ВК>

7.4.5. Команда BOOT производит начальную загрузку с указанного устройства.
Формат команды: BOOT устр: <ВК>

7.4.6. Команда EXIT передает управление монитору.
Формат команды: EXIT <ВК>

8. КОМАНДНЫЙ ФАЙЛ

Создание командного файла освобождает оператора от выполнения часто повторяющихся действий, таких как генерация системы на новом носителе или выполнения часто использующегося набора тестовых программ. Команды, которые в обычном режиме вводятся оператором, организуются при помощи редактора текста XTECO в текстовый файл и обрабатываются монитором.

По мере выполнения командного файла монитор выводит исполняемые команды и комментарии на экран терминала.

Прервать выполнение командного файла можно в любой момент одновременным нажатием клавиш <СУ> и <С> ,что обеспечит возврат в вызвавшую программу. Чтобы вернуться в монитор из вложенной цепочки, требуется двукратное нажатие указанных клавиш.

Командный файл может включать следующие команды и операторы:

- команды монитора;
- команды служебных программ;
- команды программы обработки тестовых запросов;
- условные операторы;
- операторы: GOTO, QOIET, PRINT, SMI, CMI, QUIT, WAIT;
- комментарии.

8.1. Команды монитора

В командном файле могут быть использованы следующие команды монитора: R, L, S, C, E

В отличие от выполнения под управлением оператора, команда R имеет ключ количества проходов: R имя файла/число
где "число" - количество проходов.

Команда C в командном файле может быть использована с одним ограничением. Уровень вложения не должен быть больше единицы, то есть команда C этого файла не может запустить третий командный файл.

8.2. Команды служебных программ

Командный файл может включать команды служебных программ UPD2, SETUP.

Диалог со служебными программами должен заканчиваться

командой EXIT - для того, чтобы завершить выполнение командного файла или разрешить дальнейшее его выполнение.

8.3. Команды программы обработки тестовых запросов

Командный файл представляет для программы обработки тестовых запросов файл косвенного управления. Все команды и ответы на запросы программы,которые требуются для выполнения тестов под управлением оператора, должны присутствовать в командном файле.

8.4. Условные операторы

8.4.1. Оператор IF

```
IF условие THEN
   [команды]
END
```

Условие представляет собой последовательность символов в кодах ASCII, которая используется в качестве ключа в командной строке при запуске командного файла. Монитор запоминает строку символов для последующего сравнения с условием.

Если указанное условие принимает значение "истина", то команды, заключенные между служебными словами IF и END будут выполнены. Если условие "ложно", то эти команды будут проигнорированы.

8.4.2. Оператор IFERR

```
IFERR THEN
   [команды]
END
```

Используется с тестовыми программами, работающими под управлением программы обработки тестовых запросов. Если при выполнении тестовой программы обнаружена ошибка, то будут выполнены команды,заключенные между служебными словами IFERR THEN и END.

8.4.3. Оператор IFLMD

```
IFLMD N THEN
  [команды]
END
```

В условном операторе этого типа используется байт типа носителя, расположенный по физическому адресу 41. Если тип носителя совпадает с указанным в условном операторе числом N, то будут выполнены команды, заключенные между служебными словами THEN и END.

8.5. Оператор GOTO используется для передачи управления внутри командного файла.

Формат оператора: GOTO метка

где метка - последовательность алфавитно-цифровых символов, ограниченная двоеточием.

Когда монитор встречает оператор GOTO, он ищет метку, указанную в поле оператора GOTO, и продолжает процесс выполнения команд, следующих за меткой. Метка может находиться в командном файле до и после оператора GOTO.

8.6. Оператор QUIET используется для управления выводом сообщений командного файла. Оператор используется как переключатель. Когда он встречается первый раз, весь вывод сообщений подавляется (за исключением сообщений об ошибках). Когда оператор встречается в следующий раз, вывод возобновляется.

8.7. Оператор PRINT обеспечивает вывод на экран терминала строки текста в случае, когда печать запрещена оператором QUIET.

Формат оператора: PRINT текст

Будет выведен текст, находящийся на одной строке с оператором PRINT.

8.8. Оператор SMI используется для разрешения режима вмешательства оператора в работу тестов, работающих под управлением программы обработки тестовых запросов.

8.9. Оператор CMI используется для запрещения режима вмешательства оператора в работу тестов, работающих под управлением программы обработки тестовых запросов. В начале исполнения командного файла принимается состояние CMI.

8.10. Оператор QUIT прекращает выполнение командного файла, и монитор возвращается в режим оператора.

8.11. Оператор WAIT. Когда встречается оператор WAIT, монитор прекращает выполнение командного файла и ждет до тех пор, когда оператор введет <СУ>/<X> (одновременное нажатие клавиш <СУ> и <X>).

8.12. Комментарии

В командном файле могут быть комментарии. Они будут выводиться на экран терминала по мере выполнения командного файла, если не установлен режим QUIET. Комментарии начинаются с символа ";" или "$".

";" – используется для обозначения начала комментария, который должен быть выведен на экран терминала в процессе выполнения командного файла. После вывода комментария выполнение командного файла продолжается;

"$" – обозначает начало комментария, выво-Которым происходит останов в выполнении командного файла. Продолжить выполнение командного файла можно одновременным нажатием клавиш <СУ> и <X>. Комментарий такого типа используется для того, чтобы оператор мог выполнить необходимые действия.

9. ПРОГРАММА ОБСЛУЖИВАНИЯ ТЕСТОВЫХ ЗАПРОСОВ (DRS)

9.1. Описание DRS

Программа обслуживания тестовых запросов (DRS) на этапе выполнения тестов является составной частью системы TMOC, управляющей выполнением совместимых с ней тестовых программ. Эта программа является расширением монитора системы, она автоматически загружается в память и запускается при запуске совместимых с ней тестовых программ.

Если не известно, какие именно из тестовых программ на конкретном носителе являются совместимыми с DRS, для получения списка таких программ можно использовать программу SETUP (см. раздел 10).

9.2. Запуск DRS

Процедура запуска DRS является относительно прямой. При помощи команды R пользователь загружает и запускает тестовую программу. Первой командой тестовой программы является EMT, передающая управление назад в монитор. Затем монитор загружает DRS с системного носителя. Файл DRS носит имя "KSAA??", где символы "??" означают соответственно номер версии и корректировки. Перед тем, как перейти в командный режим, DRS при помощи диспетчера памяти определяет об'ем памяти в системе и переходит в командный режим, выведя "DR>".

Если программа не выйдет на "DR>", то прежде всего необходимо выполнить тест диспетчера памяти.

9.3. Основные термины, используемые в DRS

Команды

- для диалога с пользователем DRS использует системный терминал. В распоряжении пользователя для управления работой DRS имеются 11 команд (эти команды описаны в следующем подразделе). В отличие от автономных тестовых программ функционирование DRS не зависит от адреса запуска или пульта процессора.

Модификаторы команд (ключи и флаги)

- пользователь может изменять функцию конкретной команды, снабдив ее ключом. Например, если не указано <ИНАЧЕ>, большинство команд воздействует на все устройства, которые могут проверяться тестовой программой. Для ограничения распространения действия команды до определенных устройств может быть использован ключ.

Устройство

- тестовая программа предназначена для тестирования определенного типа устройств, в состав которого может входить несколько подустройств. Каждое такое аппаратное средство рассматривается программой DRS как "тестируемое устройство" или просто "устройство". Программа DRS поддерживает до 64 устройств. Пользователь задает устройство номером. Первое устройство имеет номер 0. Устройства нумеруются в том порядке, в каком они указываются в "таблице аппаратных параметров".

Таблица аппаратных параметров

- совместимые с DRS тестовые программы сами не определяют состав оборудования в комплексе (т.е не определяют наличие оборудования при помощи тестов шины). Пользователь должен внести в тест информацию о составе проверяемых устройств. Эта информация хранится в

таблице, называемой "Таблица аппаратных параметров". Каждому тестируемому устройству соответствует одна таблица. Необходимая информация зависит от устройства. Существуют некоторые общие правила построения таблиц, описанные в подразделе 9.7. Тестовая программа выдает оператору запрос на ввод необходимой информации для каждого устройства, начиная с устройства 0. Важным моментом, который необходимо усвоить пользователю, является понятие "таблично управляемая тестовая программа" - вся информация об устройстве содержится в специфичной для этого устройства Таблице.

Таблица программных параметров

- пользователь может для управления функционированием конкретной тестовой программы выбрать некоторые операционные параметры. Эта информация помещается в таблицу, называемую "таблицей программных параметров". Для читателя,знакомого с TMOC, укажем, что эта Таблица заменяет регистр пульта процессора.

Проход

- проходом или шагом тестирования называется выполнение всех указанных тестов на всех проверяемых устройствах.

Тест

- тестовые программы подразделяются на независимые структурные единицы, называемые тестами. Пользователь может выбрать любое подмножество тестов, или выполнить все тесты в тестовой программе.

Сообщения об ошибках

- при обнаружении тестовой программой ошибки в тестируемом устройстве она обращается к DRS для выдачи оператору сообщения об ошибке. В сообщениях об ошибках имеется три уровня: заголовок, общий уровень и расширенный уровень.

Первый уровень - заголовок сообщения, содержит некоторую общую информацию об ошибке.

Пример:

```
ZNAME HRD  ERR  00002  ON UNIT5  TST 012  SUB 000  PC:013134
```

В заголовке дана следующая информация:

```
ZNAME      - имя тестовой программы;
HRD        - тип ошибки (аппаратная);
ERR 00002  - номер ошибки (2);
ON UNIT5   - номер устройства (5);
TST 012    - номер теста (12);
SUB 000    - номер подтеста (0);
PC:013134  - адрес, с которого произошел
             вызов DRS по ошибке.
```

ПРИМЕЧАНИЕ. Номер ошибки обозначает ее тип, а не общее количество ошибок.

Второй - общий уровень сообщений об ошибке, содержит также и простое описание ошибки. Расширенный уровень обычно используется для вспомогательной информации, такой как содержимое регистров в момент ошибки.

Пример:

```
ZNAME HRDERR 00002 ON UNIT5 TST 012 SUB 000 PC:013134
REGISTER FAILED TO CLEAR AFTER BUS RESET
"При сбросе по шине регистр не обнуляется"
```

```
CSR:    000000 CP: 010000  ERRREG: 000000
```

Первая строка - заголовок, вторая - основное сообщение, третья - расширение сообщения.

Сообщения об ошибках разделены на уровни для того, чтобы обеспечить оператору гибкость при определении, какие из сообщений об ошибках выводить на дисплей или печать (см. подраздел 9.6.).

9.4. Команды DRS

В DRS имеется 11 команд. Эти команды вводятся на запрос DRS: "DR>". Этот символ выдается в следующих случаях:

- при загрузке DRS;
- по окончании всех указанных операций тестирования;
- при обнаружении ошибок тестами, работающими под DRS;
- после останова по ошибке и после прерывания DRS при помощи <CУ>/<C>.

Ниже приведен краткий перечень команд и дано их назначение. Команды сгруппированы по функциям.

	Команды запуска
START	запуск тестовой программы и инициализация
RESTART	запуск тестовой программы без инициализации
CONTINUE	продолжать выполнение тестовой программы с того теста, где она была прервана нажатием <CУ>/<C>.
PROCEED	продолжение после останова по ошибке.
	Управление составом тестируемых устройств
DROP	вывести устройство из состава тестируемых
ADD	активизировать устройство для тестирования
DISPLAY	вывести содержимое Таблицы аппаратных средств.
	Флаги
GS	распечатка состояний всех флагов.
ZFLAGS	сбросить (установить в 0) все флаги.
	Статистика
PRINT	печать статистической информации.
	Выход из DRS
EXIT	возвращение управления монитору TMOC.

Ниже описано выполнение каждой команды. Функцию команд можно модифицировать при помощи описанных в следующем разделе ключей. Прежде чем пытаться использовать ключи, следует ознакомиться с командами. DRS распознает команды минимум по трем символам, необязательные для ввода символы заключены в квадратные скобки. Таким образом, команду запуска можно вводить и как "STA", и как "STAR", и как "START".

9.4.1. Команда START запускает тестовую программу с ее первоначального состояния. Это должна быть первая вводимая в DRS команда. Выполняются все действия по инициализации программы. Точный вид процесса инициализации для каждой конкретной тестовой программы можно найти в листингах и в описании тестовой программы. В пространство векторов перезагружаются так называемые "коды ловушек прерываний" (ловушки прерываний - это коды, позволяющие DRS обрабатывать непредвиденные прерывания и выдавать сообщения о них пользователю).

Формат команды: STA[RT] [список ключей]

Список ключей

- любая допустимая комбинация ключей (модификаторов) для данной команды.

При отсутствии ключей команда START выполняется следующим образом: все тесты выполняются на всех устройствах. Все флаги (см. подраздел 9.6.) будут сброшены. Тестирование начинается с первого теста для первого устройства и будет продолжаться до тех пор, пока от пользователя не поступит прерывание <СУ>/<С>, или не возникнет системная ошибка; после каждого прохода будет распечатываться сообщение "КОНЕЦ ПРОХОДА".

После введения команды START система запросит, не хотите ли вы изменить состав тестируемого оборудования.

Возможны следующие типы ответов и соответствующие им коды:

- O восьмеричное значение;
- D десятичное значение;
- B двоичное значение;
- A символьное значение ASCII;
- L логическая величина: Y(да), N(нет).

Если таблиц аппаратных параметров в системе нет, на этот вопрос необходимо ответить "Y"(да). Таблицы могут уже существовать в трех случаях:

1) если они были введены предыдущей командой запуска;

2) если они были созданы при помощи служебной программы SETUP (см. раздел 10);

3) таблицы созданы при написании программы.

Оператор может изменить уже существующие таблицы. Для этого необходимо ввести в десятичном виде число устройств, предназначенных для тестирования. Программа попросит вас ввести аппаратные параметры для каждого устройства (см. подраздел 9.7.).

Пример:

```
DR>STA
CHANGE HW(L) ? Y    (изменять аппаратные параметры?  да)
#UNITS (D) ? X      (количество устройств?           X )
[ответы на вопросы тестовой программы]
CHANGE SW(L) ? N    (изменять программные параметры? Нет)
```

Программа запросит у вас аппаратные параметры по "X" устройствам, где "X" - десятичное число от 1 до 64. После ввода аппаратных параметров вы получите запрос: не хотите ли вы изменить операционные параметры (таблицы программных параметров). Если вы не хотите изменять функционирование по умолчанию тестовой программы, то отвечать на все запросы относительно программных параметров не обязательно. Толкование специальных вопросов приводится в листингах и в описании по тестовой программе.

Если в системе отсутствуют таблицы аппаратных и программных параметров, тестирование не начнется, и будет выдано сообщение об ошибке.

Пример:

```
DR>STA
CHANGE HW(L) ? N    (изменять аппаратные параметры?  нет)
CHANGE SW(L) ? N    (изменять программные параметры? нет)
NO UNITS            (нет устройств)
```

```
DR>
```

9.4.2. Команда RESTART. По команде RESTART, так же, как и по команде START, производится запуск тестовой программы из начального состояния. Процесс инициализации по рестарту может отличаться от первоначального. Пространство векторов не изменяется. Имеется возможность изменения только программных параметров.

Формат команды: RES[TART] [список ключей]

Список ключей

- любая допустимая комбинация ключей для команды RESTART. Описание ключей дано в подразделе 9.5.

По умолчанию команда RESTART выполняется следующим образом: выполняются все тесты для всех устройств. Флаги сбрасываются. Тестирование будет производиться до тех пор, пока не поступит прерывание от пользователя посредством <СУ>/<С> или не возникнет системная ошибка. После каждого прохода будет распечатываться сообщение конца прохода.

Пример:

```
DR>RES
CHANGE SW(L) ? N    (изменять программные параметры? нет)
```

9.4.3. Команда CONTINUE используется для возобновления процесса тестирования после прерывания ее пользователем через <СУ>/<С> или после останова по ошибке. Тестовая программа будет вновь запущена с начала прерванного теста, а не с начала первого теста, как в случае команды RESTART. Устройство, находящееся в состоянии тестирования в момент прерывания тестовой программы, останется в этом состоянии. Будет предоставлена возможность изменить таблицу программных параметров. Таблицы аппаратных параметров изменить нельзя.

Формат команды: CON[TINUE] [список ключей]

Список ключей

- любая допустимая комбинация ключей для команды CONTINUE. Описание ключей дано в подразделе 9.5.

По умолчанию команда CONTINUE выполняется следующим образом: будет выполнено число проходов тестирования, равное указанному в последней команде START или RESTART. Все флаги останутся в тех состояниях, которые были определены ранее.

Пример:

```
^C
DR>CON
CHANGE SW (L) ? N
```

Для возобновления тестирования пользователь также может использовать команды START или RESTART, но при этом будет выполнена инициализация тестовой программы, и тестирование начнется с первого теста для первого устройства.

9.4.4. Команда PROCEED используется в DRS исключительно совместно с функцией останова по ошибке. В режиме останова по ошибке при получении сообщения об ошибке от тестовой программы DRS переходит в командный режим. Команда PROCEED отличается от команды CONTINUE тем, что по ней тестирование возобновляется с той точки, где было выдано сообщение об ошибке. Инициализация не производится, обращения к тестируе-

мому устройству не происходит и пространство векторов не изменяется.

Формат команды: PRO[CEED] [список ключей]

Список ключей

– любая допустимая комбинация ключей для команды PROCEED. Описание ключей дано в подразделе 9.5.

9.4.5. Команда DROP позволяет исключить из процесса тестирования одно или несколько устройств. Номера "отключенных" устройств необходимо указать в ключе устройств. Сначала все устройства являются активными для тестирования. Чтобы исключить устройство из процесса тестирования, его необходимо указать явно в тестовой программе или в поле ключа. Устройство считается активным, если его параметры занесены в таблицу устройств.

Формат команды: DRO[P] [/UNI[TS]:X]

где "X" – номера устройств, которые должны быть выведены из процесса тестирования. Описание ключа устройств UNITS приведено в подпункте 9.5.1.5. По умолчанию (когда ключ устройств не задан) по команде DROP из процесса тестирования исключаются все активные в текущий момент устройства.

9.4.6. Команда ADD предназначена для возобновления тестирования устройств, ранее выведенных из состава тестируемых. Номера вновь подключенных устройств указываются в поле ключа.

Формат команды: ADD[/UNI[TS]:X]

где X – номера устройств, переводимых в активные.

UNITS – ключ, который описан в подпункте 9.5.1.5.

По умолчанию (когда ключ устройств не указан) все ранее выведенные из состава тестируемых устройства переводятся в активное состояние.

9.4.7. Команда DISPLAY выводит на экран терминала содержимое таблиц аппаратных параметров для всех проверяемых устройств. Если устройство было выведено из состава тестируемых, это отмечается при распечатке.

Формат команды: DIS[PLAY][/UNI[TS]:X]

где X – номера устройств, для которых необходимо вывести содержимое таблиц аппаратных параметров.

UNITS – ключ, который описан в подпункте 9.5.1.5.

По умолчанию по команде DISPLAY информация обо всех устройствах, описанных в таблицах аппаратных параметров, выводится на экран терминала.

9.4.8. Команда FLAGS используется для определения состояний флагов DRS. Получив эту команду, DRS выведет на экран терминала состояния всех флагов.

Формат команды: FLA[GS]

П р и м е р :

```
DR>FLA
FLAGS SET
NONE
```

– установленных флагов нет.

```
DR>FLA
```

FLAGS SET
IER
LOE

– были установлены два флага: IER и LOE (эти флаги описаны в подразделе 9.6.).

9.4.9. Команда ZFLAGS. По команде ZFLAGS сбрасываются все флаги DRS.

Формат команды: ZFL[AGS]

9.4.10. Команда PRINT. По команде PRINT статистическая информация, накопленная во время прохождения тестовой программы, выводится на экран терминала. Следует отметить, что команда PRINT используется исключительно с тестовыми программами, поддерживающими статистические сообщения.

Формат команды: PRI[NT]

9.4.11. Команда EXIT предназначена исключительно для выхода из программы DRS и передачи управления монитору.

Формат команды: EXI[T]

9.5. Ключи программы DRS

Перед тем, как перейти к этому подразделу, следует подробнее ознакомиться с командами DRS.

Ключи являются модификаторами функций команд. Для каждой команды применяется своя группа ключей. Так многие команды DRS оперируют с устройствами. Обычно действие команд этого типа распространяется на все устройства, указанные при построении таблиц аппаратных параметров. Ключ дает пользователю возможность ограничить действие команды до некоторых выбранных им устройств.

Ключами являются:

/TESTS:список тестов	выполнить только указанные тесты
/PASS:XXXXX	выполнить XXXXX проходов (XXXXX – от 1 до 65536)
/FLAGS:список флагов	установить только указанные флаги = 1
/EOP:XXXXX	выдавать сообщения конца прохода после каждых XXXXX проходов (XXXXX – от 1 до 65536)
/UNITS:список устройств	действие команд распространяется только на указанные устройства

С каждой командой можно использовать не все ключи. В табл.5 указано, какие ключи можно использовать с каждой из команд.

Таблица 5

Команда	TESTS	PASS	FLAGS	EOP	UNITS
1	2	3	4	5	6
START	X	X	X	X	X
RESTART	X	X	X	X	X
CONTINUE		X	X	X	
PROCEED			X		
DROP					X
ADD					X
PRINT					
DISPLAY					X
FLAGS					
ZFLAGS					
EXIT					

9.5.1. Описание ключей

9.5.1.1. Ключ TESTS задает тесты, которые необходимо выполнить. По умолчанию в DRS выполняются все тесты.

Формат ключа: /TES[TS]:список тестов

Список тестов представляет собой последовательность номеров тестов, разделенных символом ":". Если номера тестов представляют собой упорядоченную последовательность, их можно задавать первым и последним номерами, разделенными символом "-". Например, если надо задать тесты 1, 2, 3 и 4, их номера можно ввести как 1:2:3:4 или же как 1-4. Номера тестов можно указать в любом порядке, но выполняться тесты будут всегда в порядке возрастания их номеров.

Примеры:

```
DR>START/TEST:5          ; будет выполнен тест 5
DR>START/TESTS:1:2       ; будут выполнены тесты 1 и 2
DR>RES/TES:1:5-9:15      ; будут выполнены тесты 1, 5,
                           6, 7, 8, 9, 15
```

9.5.1.2. Ключ PASS используется для задания числа проходов тестовой программы. Проходом называется один цикл выполнения всех указанных тестов на всех проверяемых устройствах. По умолчанию тестовая программа будет выполняться до тех пор, пока не будет приостановлена одновременным нажатием <CУ>/<C>.

Формат ключа: /PAS[S]:XXXXX

где XXXXX - десятичное число от 1 до 65536.

Примеры:

```
DR>STA/PASS:100  ; задано 100 проходов
DR>RES/PAS:1     ; будет выполнен 1 проход
```

9.5.1.3. Ключ FLAGS используется для задания флагов, управляющих функциями DRS. Эти флаги подробно описаны в подразделе 9.6. Если ключ FLAGS в команде не задан, то по умолчанию все флаги сброшены.

Формат ключа: /FLA[GS]: список флагов

где список флагов - это перечень флагов DRS, разделенных символом ":".

Примеры:

```
DR>STA/FLAGS:LOE
DR>RES/FLA:LOE:IER:BOE
```

9.5.1.4. Ключ EOP используется для выдачи пользователю сообщения о выполнении тестовых программ и конце проходов. В этих сообщениях указывается число законченных проходов и число обнаруженных ошибок. По умолчанию DRS печатает эти сообщения по окончании каждого прохода.

Формат ключа: /EOP:XXXXX

где XXXXX - десятичное число от 1 до 65536. Сообщение о конце проходов будет выдаваться после каждых XXXXX проходов.

Пример:

```
DR>RES/EOP:90   ; будет выдано сообщение о конце прохода
                  через каждые 90 проходов.
```

9.5.1.5. Ключ UNITS. При помощи ключа UNITS пользователь может указать номера устройств, предназначенных для тестирования. При отсутствии ключа UNITS действие любой команды DRS распространяется на все устройства.

Формат ключа: /UNI[TS]:список устройств

где список устройств - последовательность номеров устройств. Номера устройств представляют собой десятичные числа от 1 до 64.

Номера присваиваются устройствам в порядке их введения в таблицу. Первым устройством является устройство 1. Если номера проверяемых устройств представляют собой упорядоченную последовательность, они могут быть заданы первым и последним номерами, разделенными символом "-". Например, устройства 3, 4, 5, 6, 7 можно указывать как 3-7.

Примеры:

```
DR>DRO/UNITS:1          ; вывести устройство 1 из
                          процесса тестирования
DR>ADD/UNI:2,3          ; активизировать устройства 2 и 3 для проверки
DR>RES/UNI:5-9          ; перезапустить тестовую программу для тестирования устройств 5, 6, 7, 8, 9
```

9.5.2. Сочетания ключей

Пользователь по желанию может указывать в поле команды любое число допустимых ключей. Ключи указываются последовательно в командной строке.

Например, если пользователь хочет запустить тестовую программу согласно следующим ограничениям:

- тестировать только устройства 1-4
- выполнить тесты 1, 5 и 15
- выполнить 100 проходов
- выдавать сообщения конца прохода только через каждые 10 проходов,

то должна быть введена следующая командная строка:

STA/UNI:1-4/TES:1:5:15/PAS:100/EOP:10

9.6. Флаги программы DRS

При описании флагов предполагается, что пользователь подробно ознакомился с командами и ключами DRS.

Флаги используются для изменения работы тестовой программы. При запуске все флаги сбрасываются и будут оставаться в сброшенном состоянии до тех пор, пока не будут явно установлены ключом FLAGS. Флаги сбрасываются также по командам START или RESTART, если только в этих командах не указан ключ FLAGS. Кроме того, для сброса всех флагов может быть использована команда ZFLAGS. Никакие другие команды на состояния флагов не влияют.

В табл. 6 приведены флаги, используемые программой DRS,

Таблица 6

ФЛАГИ ПРОГРАММЫ DRS

Флаг	Функция
1	2
HOE	Останов по ошибке. Управление передается в командный режим DRS
LOE	Цикл на ошибке
IER	Запрет всех сообщений об ошибках
IBE	Запрет всех сообщений об ошибках, кроме сообщений первого уровня (сообщения первого уровня содержат тип ошибки, ее номер, значение программного счетчика, номер теста и номер устройства)
IXE	Запрет расширенных сообщений об ошибках
PRI	Направлять сообщения на печать. При наличии одного устройства флаг не используется
PNT	Печатать номер теста по мере его выполнения
BQE	"Звонок" по ошибке. При отсутствии "звонка" флаг не используется
UAM	Режим без ожидания (без вмешательства оператора)
ISR	Запрет статистических сообщений(не относится к тестовым программам, не поддерживающим статистические сообщения)
IDR	Запрет программного пропуска устройств
ADR	Выполнить программу автоматического пропуска устройств
LOT	Цикл на тесте
EVL	Выполнить оценку тестирования (для тестовых программ, которые обеспечивают оценку)

9.6.1. Флаг HOE в установленном состоянии вызывает останов по ошибке. Произойдет следующее:

1) Когда DRS получит тестовое сообщение об ошибке, будет распечатано сообщение (в случае, если нет запрета печати сообщений);

2) DRS перейдет в командный режим;

3) Тестовая программа "зависнет" в том месте, где в DRS поступил вызов по ошибке, а тестируемое устройство останется

в том состоянии, в каком оно находилось в момент вызова.

После того, как DRS вернулась в командный режим, пользователь может ввести команду PROCEED (продолжение) и возобновить выполнение тестовой программы с того места, в котором она была прервана. По желанию пользователь может ввести и другую команду.

9.6.2. Флаг LOE. При установленном флаге LOE DRS будет зацикливаться на ошибке. В режиме зацикливания DRS заставит тестовую программу вновь и вновь выполнять последовательность операций, вызвавшую ошибку. Зацикливание будет продолжаться и в том случае, если исчезнут причины, вызвавшие зацикливание. Это позволяет организовать цикл и на случайных сбоях. Чтобы выйти из цикла, пользователь должен ввести <СУ>/<С>, при этом DRS вернется в командный режим.

9.6.3. Флаг IER. При установленном флаге IER в DRS будет запрещен вывод на экран терминала всех сообщений об ошибках. Во время действия этого ключа оператору не будет выдаваться никаких сообщений, кроме системных ошибок, таких как ILL INT (запрещенное прерывание), и сообщений конца проходов. Это свойство обычно используется совместно с функцией зацикливания по ошибке. Оно ускоряет процесс тестирования.

9.6.4. Флаг IBE. Установка флага IBE вызывает запрет части сообщений об ошибках.

9.6.5. Флаг IXE. При установке флага IXE DRS запрещает распечатку только расширенных сообщений. Сообщения первого уровня и основные будут распечатываться.

9.6.6. Флаг PRI. При установленном флаге PRI все сообщения перенаправляются на печать.

9.6.7. Флаг PNT. При установленном флаге PNT по мере выполнения теста печатается его номер.

9.6.8. Флаг BOE. Когда установлен флаг BOE, DRS по ошибке выдает "СУ/G" ("звонок"). По этому символу терминал выдает звуковой сигнал при наличии синтезатора звука.При его отсутствии никаких действий не выполняется. Обычно это свойство используется совместно с функцией запрета распечатки сообщений.

9.6.9. Флаг UAM

Установка флага UAM вызывает запрет вмешательства оператора в процесс тестирования. Под "вмешательством оператора" понимается, что оператор может в любой момент выполнить необходимые ему действия. Использование флага UAM дает возможность оператору запустить тестовую программу и предоставить ей возможность работать автономно. При флаге UAM некоторые тестовые операции запрещены. Влияние флага UAM в каждом конкретном случае можно определить в листингах тестовой программы.

9.6.10. Флаг ISR. Установка флага ISR вызывает запрет печати тестовой программой статистических сообщений. Эта функция не является обязательной, не все тестовые программы поддерживают статистику. Для определения того, обладает ли данной функцией конкретная тестовая программа, необходимо смотреть в листингах и в описании по тестовой программе.

9.6.11. Флаг IDR. Установка флага IDR вызывает запрет пропуска \ устройств тестовой программой. Тестовая программа может пропустить устройство, если достигнут порог ошибки или обнаружена фатальная ошибка. Этот флаг дает пользователю возможность поддерживать устройство в активном состоянии, обычно это делается с целью прослеживания ошибки.

9.6.12. Флаг ADR. Установка флага ADR вызывает в тестовой программе выполнение подпрограммы автоматического пропуска устройств. Целью этой программы является проверка устройств на наличие и готовность. Если тестируемое устройство отсутствует или не готово, оно будет пропущено. Не во всех тестовых программах имеется возможность автоматического пропуска устройств. Для определения того, имеется ли у тестовой программы такая функция, необходимо смотреть в листингах и в описании по ней.

9.6.13. Флаг LOT. Установка LOT заставляет DRS вновь и вновь выполнять тест(ы), специфицированные ключом TESTS. Однако последовательности инициализации и конца проходов при этом не выполняются.

9.6.14. Флаг EVL. Установка флага EVL вызывает исполнение подпрограммы оценки тестирования. Эта функция не является обязательной, и наличие ее необходимо проверять в листингах по тестовой программе.

9.7. Построение таблиц

Как указывалось выше, в DRS для хранения информации об устройствах (такой, как адреса регистров, номера накопителей и приоритеты прерываний) используются Таблицы аппаратных параметров. Для хранения программных параметров для каждой конкретной тестовой программы (таких, как маски тестирования или информация о том, проводить или нет тестирование ПЗУ) используются аналогичные таблицы. Конкретный вид информации в разных тестовых программах различается, поэтому целью данного подраздела является – дать пользователю лишь некоторые основные сведения об этих таблицах.

Прежде всего, эти аблицы необходимо построить одним из следующих трех способов:

1) С использованием команды START;
2) С помощью программы SETUP;
3) При разработке тестовой программы.

Тестовая программа обычно загружается вместе с фиктивной таблицей, содержащей временную информацию об устройствах. В эту временную таблицу по умолчанию в некоторых случаях входят значения аппаратных параметров. Действительные значения этих параметров должны быть заданы пользователем.

Обычно это делается при запуске тестовой программы по команде START путем введения аппаратных параметров в соответствии с запросами тестовой программы.

Таблицы можно построить предварительно при помощи программы SETUP. SETUP является служебной программой, дающей пользователю возможность построить таблицы, не запуская тестовую программу. Таблицы хранятся на системном устройстве TMOC вместе с тестовой программой и загружаются в память вместе с ней при ее запуске. Тогда пользователь может начать тестирование, не строя таблицы.

Таблицы могут быть записаны программистом, разработавшим тестовую программу. Тогда эти таблицы являются частью загрузочного модуля тестовой программы,и их можно по желанию использовать в том виде, в каком они записаны, или изменять после команды START.

Таблица программных параметров (операционная таблица) может присутствовать не во всех тестовых программах. Если ее нет, то вам не будет выдан запрос,надо ли ее менять. Таблица является зарезервированной областью памяти, в которую занесены значения данных по умолчанию. Все связанные с таблицей вопросы имеют один и тот же формат:

ВОПРОС (ТИП) [ЗНАЧЕНИЕ ПО УМОЛЧАНИЮ] ?

Вопрос может иметь, например, такой вид:

"DRIVE NUMBER" (номер механизма)

ТИП - односимвольный код, обозначающий тип ответа, заключенный в круглые скобки.

Возможны следующие типы ответов и соответствующие им коды:

- O восьмеричное значение;
- D десятичное значение;
- B двоичное значение;
- A символьное значение ASCII;
- L логическая величина: Y(да), N(нет).

Вопросительный знак означает, что DRS готова к восприятию ответа. Если по какой-либо причине DRS воспринять ваш ответ не может, будет распечатано сообщение об ошибке, и пользователь получит повторный вопрос.

Отвечая на вопросы по устройствам, вы задаете значения в таблице, описывающей тестируемые устройства. Самым простым способом построения этой таблицы являются ответы на все вопросы по каждому тестируемому устройству. Если у вас имеется устройство с несколькими однотипными блоками, такое, как контроллер внешнего запоминающего устройства с несколькими внешними устройствами, или устройство связи с несколькими каналами, процедура становится утомительной, поскольку большинство вопросов повторяются.

Для иллюстрации более эффективного способа построения таблиц допустим, что тестируется некоторое фиктивное устройство под именем XVII. Предположим, что в это устройство входит контроллер со связанными с ним восемью блоками (подустройствами). Эти блоки описываются восьмеричными номерами от 0 до 7. Существует некоторый аппаратный параметр, изменяющийся от блока к блоку; назовем его "Q-фактор".Этот Q-фактор может принимать значения 0 или 1. Ниже приведена простая процедура построения таблицы для одного устройства XVII с

восемью блоками:

Вопросы DRS:	Ответы пользователя:	
#UNITS (D)?	8	количество устройств
UNIT 1		устройство N
CSR ADDRESS (o)?	160000	адрес РКС
SUB-DEVICE #(o)?	0	номер блока
Q-FACTOR (o) 0?	1	Q-фактор
UNIT 2		
CSR ADDRESS (o)?	160000	
SUB-DEVICE #(o)?	1	
Q-FACTOR (o) 1?	0	
UNIT 3		
CSR ADDRESS (o)?	160000	
SUB-DEVICE #(o)?	2	
Q-FACTOR (o) 0?	0	
CSR ADDRESS (o)?	160000	
SUB-DEVICE #(o)?	3	
Q-FACTOR (o) 0?	0	
UNIT 5		
CSR ADDRESS (o)?	160000	
SUB-DEVICE #(o)?	4	
Q-FACTOR (o) 0?	0	
UNIT 6		
CSR ADDRESS (o)?	160000	
SUB-DEVICE #(o)?	5	
Q-FACTOR (o) 0?	0	
UNIT 7		
CSR ADDRESS (o)?	160000	
SUB-DEVICE #(o)?	6	
Q-FACTOR (o) 0?	1	
UNIT 8		
CSR ADDRESS (o)?	160000	
SUB-DEVICE #(o)?	7	
Q-FACTOR (o) 1?	1	

Заметьте, что при введении ответа, не соответствующего значению Q-фактора по умолчанию, его начальное значение изменяется. Внимательно задавайте кратные блоки.

Как видно из вышеприведенного примера, аппаратные параметры существенно не изменяются от блока к блоку. Показанная процедура не очень эффективна. Однако в DRS имеется возможность заданий для таких устройств кратных спецификаций. Построим ту же таблицу, используя кратные спецификации:

1	# UNITS (D) ?	8	количество устройств
2	UNIT 1		устройство N
3	CSR ADDRESS (o) ?	160000	адрес РКС
4	SUB-DEVICE # (o) ?	0,1	номер подустройства
5	Q-FACTOR (o) 0 ?	1,0	Q-фактор
2	UNIT 3		
3	CSR ADDRESS (o) ?	160000	
4	SUB-DEVICE # (o) ?	2-5	
5	Q-FACTOR (o) 0 ?	0	
2	UNIT 7		
3	CSR ADDRESS (o) ?	160000	
4	SUB-DEVICE # (o) ?	6,7	
5	Q-FACTOR (o) 0 ?	1	

Как видно из вышеприведенного диалога, DRS построит максимальное число строк в таблице, возможное по введенной на вопрос информации. После первого запроса были построены две строки, поскольку были заданы два устройства и два значения Q-фактора. Поскольку адрес РКС был задан один раз, DRS считает, что он один и тот же для обоих блоков. После второго запроса были построены четыре строки, так как были специфицированные четыре блока. Если встречается выражение с символом "-", то DRS будет инкрементировать первое число до тех пор, пока не дойдет до второго. В данном случае были заданы устройства 2, 3, 4 и 5 (если бы блоки задавались адресами, то шаг инкрементации был бы равен 2, т.к. адрес должен быть четным). Для этих четырех строк в таблице адрес РКС и значения Q-фактора принимаются равными, соответственно, 160000 и 0, т.к. они были специфицированы только один раз. Два оставшихся блока специфицируются в третьем запросе.

Весь процесс, как показано ниже, можно было выполнить за один запрос:

```
1 # UNITS (D) ? 8                    количество устройств
2 UNIT 1                             устройство N
3 CSR ADDRESS (o) ?   160000         адрес РКС
4 SUB-DEVICE # (o) ?  0-7            номер блока
5 Q-FACTOR (o) 0 ?    0,1,0,,,,1,1   Q-фактор
```

Как видно из этого примера, пустые ответы (запятые, ограничивающие пустое поле), заставляют DRS повторять последнее значение.

9.8. Сообщения об ошибках

ERR HLT
- произошел останов по ошибке. Программа DRS возвращается в режим команд. Останов по ошибке произойдет в том случае, если оператор указал такой режим работы флагом "HOE".

ILL INTER NNN PC NNNNNN PS NNNNNN
- произошло непредвиденное прерывание по вектору NNN. При этом выдается на экран терминала содержимое счетчика команд PC и слова состояния процессора PS.

INSUFF MEM
- недостаточен об'ем памяти для построения таблицы. Оператор указал большое количество устройств.

INVAL SWITCH FOR CMND
- оператор указал несуществующий или неприменяемый ключ для предыдущей команды.

INVAL UNIT
- оператор задал несуществующее устройство.

LOOKUP ERROR FILNAM
- это сообщение исходит от монитора TMOC. Если имя файла "KSAA??.SYS", в этом случае тестовая программа обращается к программе DRS, а ее не существует на системном носителе. В случае любого другого имени файла это сообщение говорит о том, что тестовая программа пытается открыть файл, не существующий на системном устройстве.

LOOP CHNG
- в момент зацикливания по ошибке изменился режим цик-

ла.

NOT HALTED

- оператор пытался ввести команду PROCEED, когда не было останова по ошибке.

NO UNIT

- нет активизированных устройств, или устройства не были указаны флагом /UNITS, или все устройства "отключены" командой DROP.

PASS ABORTID FOR THIS UNIT

- проверка закончена для тестируемого устройства. Это сообщение имеет место после обнаружения ошибки. Следует обратиться к листингам соответствующих тестовых программ для выяснения причины отключения устройств от тестирования.

TRAP ERR AT NNNNNN

- выполнялась неопознанная команда TRAP. Команда TRAP используется для связи тестовой программы с программой DRS.

TST TOO BIG

- оператор задал большее количество подтестов, чем существует в тестовой программе.

10. ПРОГРАММА SETUP

10.1. Назначение программы

Предполагается, что изучая программу SETUP, пользователь уже имеет представление о программе обслуживания тестовых запросов DRS.

Программа SETUP предназначена для построения таблиц аппаратных и программных параметров до запуска тестовых программ и хранения этих таблиц совместно с соответствующими тестами.

ПРИМЕЧАНИЕ. Тестовые программы должны быть совместимы с программой обслуживания тестовых запросов - DRS.

Для запуска SETUP требуется такой же об'ем памяти, что и для программы DRS: 5,75 кслов. В системе должно быть как минимум 16 свободных кслов.

10.2. Запуск программы

Запуск программы SETUP выполняется по команде монитора:

R SETUP <BK>

или командами:

L SETUP <BK>
S <BK>

10.3. Команды программы SETUP

10.3.1. Команда SETUP предназначена для построения таблиц тестовой программы, работающей совместно с DRS. Эта команда сначала вызывает загрузку в память указанной тестовой программы, а затем выполняет подпрограмму построения таблиц в тестовой программе. Процедура построения таблиц описана в подразделе 9.7. Эта процедура аналогична процедуре запуска тестовой программы по команде START.

Формат команды:

SETUP выхуст:имя1.тип=вхуст:имя2.тип <ВК>

где имя1.тип - имя выходного файла;
имя2.тип - имя входного файла.

Имена входного и выходного файлов могут совпадать, но в случае, если имена выходного и входного устройств совпадают, будет выдано предупреждающее сообщение о том, что входной файл будет стерт:

DELETE INPUT FILE ?(Y/N/<ВК>=Y)
"Стирать входной файл ? (да/нет/<ВК>=да)"

Если пользователь отвечает Y<ВК> или <ВК>, то в случае совпадения устройств, входной файл будет стерт, а на его место после завершения работы будет записан выходной файл. Если ответить N<ВК>, то программа возвращается в режим команд.

10.3.2. Команда LIST используется для получения списка всех имеющихся на носителе тестовых программ, совместимых с DRS.

Формат команды: LIST уст:имя.BI?<ВК>

По умолчанию используется системное устройство.

ПРИМЕЧАНИЕ. Все файлы должны иметь расширение .BIN или .BIC, допускается использование универсальных символов; по умолчанию принимается "*.BI?".

10.3.3. Команда EXIT является завершающей в программе SETUP. При ее выполнении управление передается монитору.

Формат команды: EXIT<ВК>

После ее выполнения на экране монитора появится символ ".".

10.4. Сообщения об ошибках программы SETUP

?INVALID DEVICE
- в команде неверно указано устройство.

?INVALID NAME
- неверное имя файла. В именах запрещается использование всех символов, кроме букв латинского алфавита и арабских цифр. Случаи применения символов "*" и "?" описаны в разделе 3.1.

?INVALID ADDR
- ошибочный адрес. Адрес должен быть четным и находиться в пределах границ загрузки.

?INVALID COMMAND
- ошибочная или несуществующая команда.

?NON-EXISTENT FILE
- несуществующий файл. Имя файла не существует в каталоге устройства.

?CHECKSUM ERR
- ошибка контрольной суммы при выполнении загрузки файла.

?PROGRAM OVERFLOW
- для существующей буферной области программа имеет слишком большие размеры.

?LOGICAL DEVICE NOT ASSIGNED
- было использовано логическое имя, которое не присва-

ивалось устройству.

?NO DEVICE DEFAULTS ALLOWED

– не указан тип и номер устройства в команде, в которой это необходимо.

11. РЕДАКТОР ТЕКСТА

11.1. Назначение и характеристики программы

Программа редактор текста XTECO предназначена для создания новых и редактирования существующих текстовых файлов. Она используется для создания командных файлов, текстовых файлов и внесения комментариев в текстовые файлы.

Программа XTECO занимает 6к слов оперативной памяти. После загрузки в память программа располагается в области младших адресов, начиная с адреса 1000, и использует резидентную в памяти часть монитора системы. Она затирает ранее загруженную программу.

11.2. Выполнение программы

Загрузка и запуск программы выполняются по команде:

R XTECO <ВК>

После запуска программы на экран терминала будет выведено XTECO и адрес рестарта, а также символ "*". После этого необходимо задать режим редактирования. Существуют три команды задания режима редактирования: TEXT, TECO, EDIT (см. п. 11.4.).

После выполнения одной из трех команд программа XTECO войдет в режим редактирования и выдаст символ " " ".

Затем выполняется редактирование текста, используя команды из подраздела 11.6. По окончании редактирования пользователь должен набрать команду закрытия файлов EXIT.

11.3. Управляющие символы программы XTECO.

Для управления работой программы XTECO используются следующие управляющие символы:

<СУ>/<[>	ограничивает команду редактирования при задании нескольких команд редактирования в строке команд (печатает символ "$");
<СУ>/<[>/<СУ>/<[>	вызывает выполнение команды или строки команд (печатаются символы "$" "$");
<СУ>/<C>	переводит программу в режим команд, содержимое буфера программы теряется; повторное выполнение <СУ>/<C> переводит в режим монитора;
<СУ>/<U>	очищает буфер клавиатуры, используется для отмены набранной команды;
<СУ>/<S>	приостанавливает вывод на экран терминала;
<СУ>/<Q>	разрешает продолжение вывода на экран терминала;
<ЗБ>	стирает последний введенный символ.

ПРИМЕЧАНИЕ. Запись <СУ>/<[> означает одновременное

нажатие клавиш <СУ> и <[>.

11.4. Команды выбора режима редактирования

11.4.1. Команда TEXT используется для создания нового текстового файла.

Формат команды: TEXT уст:имя.тип <ВК>

где уст — имя устройства для записи текстового файла;
имя.тип — имя файла.

После ввода команды выводится напоминающее сообщение:

MAKE OUTPUT READY. TYPE <CR> WHEN READY

"Подготовить выходное устройство. Если готово, нажать <ВК>".

После ответа <ВК> программа переходит в режим редактирования. На экран терминала выводится символ " " ", после чего пользователь может вводить собственный текст.

11.4.2. Команда TECO используется для редактирования и создания текстового файла с тем же именем (только для устройств с произвольным доступом).

Формат команды: TECO уст:имя.тип <ВК>

где уст — имя устройства, на котором записан входной файл;
имя.тип — имя файла.

После ввода команды выводится напоминающее сообщение:

MAKE OUTPUT READY. TYPE <CR> WHEN READY

"Подготовить выходное устройство. Если готово, нажать <ВК>"

После ответа <ВК> программа переходит в режим редактирования. На экран терминала выводится символ " " ", после чего пользователь может редактировать исходный текст.

ПРИМЕЧАНИЕ. После выполнения команды закрытия файла входному файлу присваивается тип .BAK.

11.4.3. Команда EDIT используется для редактирования и создания текстового файла с другим именем (для устройств любого типа). Входное и выходное устройства могут быть различными (либо устройства различных типов, либо различные устройства одного типа).

Формат команды:

EDIT выхуст:вых.файл=вхуст:вх.файл <ВК>

где выхуст — имя выходного устройства, на которое записывается выходной файл;
вхуст — имя входного устройства, на котором записан входной файл.

После ввода команды выводится напоминающее сообщение:

MAKE OUTPUT READY. TYPE <CR> WHEN READY

"Подготовить выходное устройство. Если готово, нажать <ВК>"

После ответа <ВК> программа переходит в режим редактирования. На экран терминала выводится символ " " ", после чего пользователь может редактировать исходный текст.

11.5. Служебные команды программы XTECO

В XTECO существуют три команды, которые не имеют отношения к редактированию. Они служат для удобства пользователя.

11.5.1. Команда PRINT предназначена для вывода файла на печать.

Формат команды: PRINT уст:имя.тип <ВК>

11.5.2. Команда TYPE предназначена для вывода файла на экран терминала.

Формат команды: TYPE уст:имя.тип <ВК>

11.5.3. Команда EXIT предназначена для возврата управления монитору.

Формат команды: EXIT <ВК>

11.6. Команды редактирования текста используются для редактирования текста после его вызова в оперативную память по одной из команд: TEXT, TECO, EDIT.

11.6.1. Команда J предназначена для перемещения маркера в начало буфера текста.

Формат команды: J <СУ>/<[><СУ>/<[>

11.6.2. Команда ZJ предназначена для перемещения маркера в конец буфера текста.

Формат команды: ZJ <СУ>/<[><СУ>/<[>

11.6.3. Команда C предназначена для посимвольного перемещения маркера в буфере текста.

Формат команды: NC <СУ>/<[><СУ>/<[>

где N - необязательный аргумент, который определяет направление перемещения и число символов, на которое происходит перемещение. N - десятичное число, которое положительно при перемещении маркера вперед и отрицательно при перемещении маркера назад. Если оно не указано, то принимается значение, равное единице.

Эта команда не распознает строки, последовательность символов "ВК" и "ПС" ею рассматриваются как два символа.

11.6.4. Команда L предназначена для построчного перемещения маркера в буфере текста.

Формат команды: NL <СУ>/<[><СУ>/<[>

где N - необязательный аргумент, который определяет направление перемещения и число строк, на которое происходит перемещение. Это десятичное число положительно при перемещении маркера вперед и отрицательно при перемещении назад. По умолчанию N = 1.

Строка текста есть последовательность символов, ограниченная символами "ВК" и "ПС". Эти символы генерируются при нажатии клавиши <ВК>. После выполнения команды маркер всегда располагается в начале строки.

11.6.5. Команда T предназначена для вывода строк текста на экран терминала,

Строка текста есть последовательность символов, ограниченная символами "ВК" и "ПС".

Формат команды: NT <СУ>/<[><СУ>/<[>

где N - необязательный аргумент, который определяет направление и число выводимых строк. Это десятичное число положительно, если строки расположены за маркером, и отрицательно, если строки предшествуют маркеру. По умолчанию N = 1.

Если маркер расположен внутри строки текста, то вывод начнется от того места, где расположен маркер, или закончится на нем.

Существуют две специальные формы команды T:

OT <СУ>/<[><СУ>/<[> - распечатывает текст от начала строки до текущего положения маркера;

NT <СУ>/<[><СУ>/<[> - распечатывает весь текст, находящийся за маркером.

11.6.6. Команда D предназначена для удаления символов из буфера текста.

Формат команды: ND <СУ>/<[><СУ>/<[>

где N - необязательный аргумент, определяющий число символов, которое необходимо удалить, и их положение относительно маркера. Это десятичное число положительно, если удаляемые символы расположены за маркером, и отрицательно, если они расположены перед ним. По умолчанию N = 1.

11.6.7. Команда K предназначена для удаления строк текста из его буфера.

Строка текста - это последовательность символов, ограниченная символами "ВК" и "ПС". Стирание строки включает в себя удаление из текста не только символов строки, но и ограничивающих ее символов "ВК" и "ПС". Если маркер расположен внутри строки, то будет стерта не вся строка.

Формат команды: NK <СУ>/<[><СУ>/<[>

где N - необязательный аргумент, определяющий число строк, которое необходимо удалить, и их положение относительно маркера. Это десятичное число, положительно, если удаляемые строки расположены за маркером, и отрицательно, если строки расположены перед маркером. По умолчанию N = 1.

11.6.8. Команда I является основной командой, используемой при вводе текста. Она предназначена для вставки нового текста.

Формат команды: I текст <СУ>/<[><СУ>/<[>

Текст вставляется после текущего положения маркера. Маркер по окончании выполнения команды будет расположен после нового текста. Включаемый текст не должен содержать символы "СУ/[", "СУ/C", "СУ/Q", "СУ/U", "СУ/S" .

11.6.9. Команда A. На устройстве ввода весь текст записан блоками постоянной длины. Считывание текста начинается с

первого блока. Команда A предназначена для считывания следующего блока текста с устройства ввода и добавления его к содержимому буфера текста.

Формат команды: NA <СУ>/<[><СУ>/<[>

где N - необязательный числовой аргумент, указывающий количество добавляемых блоков. Если буфер переполнен, то часть содержимого выводится в выходной файл. N - десятичное положительное число.

11.6.10. Команда S предназначена для поиска строки текста, содержащей заданную комбинацию символов, находящуюся в буфере текста.

Формат команды: S<ТЕКСТ> <СУ>/<[><СУ>/<[>

По этой команде будет происходить поиск строки, содержашей последовательность символов "текст". Поиск начинается с символа, определяемого текущим положением маркера и продолжается до конца буфера текста. Маркер устанавливается за найденной последовательностью или за всем текстом, если последовательность не найдена. Если указанная последовательность не обнаружена, то печатается сообщение:

NOT FOUND: ТЕКСТ

"Не найдена последовательность символов <текст>"

11.6.11. Команда N предназначена для поиска указанной строки текста во входном файле.

Формат команды: N<ТЕКСТ> <СУ>/<[><СУ>/<[>

Команда N выполняется аналогично команде S, но если указанная последовательность не обнаружена в буфере текста, то программа XTECO перепишет текст из буфера текста в выходной файл, загрузит следующую часть текста из входного файла и начнет поиск указанной последовательности в буфере текста. Процесс будет продолжаться до тех пор, пока указанная последовательность символов не будет обнаружена или не будет исчерпан входной файл.

Если последовательность символов так и не найдена, печатается сообщение:

NOT FOUND: ТЕКСТ

"Не найдена последовательность символов <текст>"

и маркер устанавливается в конце входного файла.

ПРИМЕЧАНИЕ. Если в результате неудачного поиска тот раздел текста, с которого начался поиск, окажется переписанным в выходной файл, то вернуться в него уже нельзя. В этом случае пользователь должен выйти из режима редактирования и вновь войти в него, задав тот файл, который ранее был выходным, в качестве входного. При этом выходной файл будет закрыт, а отредактированная часть текста сохранится.

11.6.12. Команда EX предназначена для завершения всех операций редактирования и закрытия выходного файла.

Формат команды: EX <СУ>/<[><СУ>/<[>

Если режим редактирования был инициализирован командой XECO, то входной файл получает расширение .BAK, а выходному файлу будет присвоено первоначальное имя входного файла. Программа XTECO выйдет из режима редактирования. При этом на экране терминала вместо символа " " " появится символ "*".

11.7. Комбинирование команд редактора

Пользователь может задать несколько команд редактирования в одной командной строке. При этом команды отделяются одна от другой одновременным нажатием клавиш <СУ>/<[>, а вся строка завершается двукратным нажатием клавиш <СУ>/<[><СУ>/<[>.

Пример:

```
"N MY0: <СУ>/<[><СУ>/<[>
"OT <СУ>/<[><СУ>/<[>
"T <СУ>/<[><СУ>/<[>
```

Эту последовательность команд можно ввести одновременно в командной строке:

```
"N MY0:<СУ>/<[>OT<СУ>/<[>T<СУ>/<[><СУ>/<[>
```

В обоих случаях XTECO осуществит сквозной поиск MY0:; распечатывает символы от начала строки, где была обнаружена эта последовательность, до текущего положения маркера, а затем распечатывает символы от текущего положения маркера до конца строки.

11.8. Сообщения об ошибках

? NOT FOUND<ASCII STRING>
- не найдена строка в коде ASCII.

? INVALID DEVICE
- в команде неверно указано устройство.

? INVALID NAME
- неверно указано имя файла. В именах запрещается использование всех символов, кроме букв латинского алфавита и арабских цифр. Случаи применения символов "*" и "?" описаны выше.

? INVALID ADDR
- ошибочный адрес. Адрес должен быть четным и находиться в пределах границ загрузки.

? INVALID COMMAND
- ошибочная или недопустимая команда.

? NON-EXISTENT FILE
- несуществующий файл. Имя файла не существует в каталоге устройства.

? DELETE OLD FILE
- уничтожить старый файл. Перед выполнением данной команды, создающей новый файл, необходимо уничтожить старый файл с тем же именем.

? RD/WT DEV ERR
- ошибка устройства ввода-вывода.

? CHECKSUM ERR
- ошибка контрольной суммы при выполнении загрузки.

? NO DEVICE DEFAULTS ALLOWED
- не указан тип и номер устройства. В данной команде не действует соглашение "по умолчанию".

12. КОПИРОВАНИЕ ТМОС

Рекомендуется проводить проверку ЭВМ и периферийных устройств с помощью рабочей копии ТМОС.

Для получения рабочей копии необходимо выполнить следующее:

1) Загрузить и запустить программу UPD2 (см.п. 6.2.)

2) Установить носитель для записи рабочей копии устройство, одноименное тому, с которого копируется ТМОС.

3) Подать команду: COPY выхуст:=вхуст: <ВК>

где выхуст: - имя устройства для записи;
вхуст: - имя устройства для считывания.

После ввода команды будет выдано предупреждающее сообщение:

USER DATA ON OUTPUT DEVICE WILL BE DESTROYED!
PROCEED? (Y/N/CR=N)

"Информация на выходном устройстве будет потеряна!"
"Продолжать?" (Да/нет/<ВК>=нет)

В ответ на это нужно набрать: Y <ВК>

Программа задаст вопрос: "FILE COPY OR IMAGE COPY? (I/F)"
"Режим копирования? (I/F)"

На этот вопрос нужно ответить: I <ВК>

После этого будет выполнено копирование дисков.

Рабочую копию ТМОС можно также получить с помощью командного файла SYSGEN.CCC, входящего в состав тест-мониторной системы (см. приложение 4).

13. ГЕНЕРАЦИЯ МИНИМАЛЬНОЙ КОНФИГУРАЦИИ

Для проверки, наладки и поиска неисправностей микро-ЭВМ допустима генерация ТМОС минимальной конфигурации - монитор и требуемые тест-программы.

13.1. Генерация монитора системы

1) Загрузите и запустите программу "обработка файлов":

R UPD2 <вк>

2) Установите носитель для генерации монитора в устройство и подготовьте его для записи.

3) Подайте последовательность команд:

ZERO уст1: <ВК>	инициализация носителя в формате системы;
LOAD уст0:имя.тип <ВК>	загрузка монитора, предназначенного для генерации,в оперативную память;
SAVM уст1: <ВК>	помещение на носитель с произвольным доступом загружаемого монитора;
или	
SAVE уст1: <ВК>	помещение на носитель с последовательным доступом загружаемого монитора.

где уст1 - имя устройства с носителем для генерации монитора;
уст0 - имя устройства для считывания монитора;
имя.тип - имя файла монитора.

13.2. Генерация тестовых программ

После генерации монитора подайте команду:

FILE уст1:=уст0:имя.тип <ВК>

где уст0 - имя устройства для считывания тестовой

программы;
уст1 - имя устройства для записи тестовой программы;
имя.тип - имя файла тестовой программы.

14. ОСНОВНЫЕ ПОНЯТИЯ О ТЕСТОВЫХ ПРОГРАММАХ

14.1. Назначение тестовых программ

Тестовые программы осуществляют проверку правильности функционирования проверяемых устройств ЭВМ.

Назначение и функции каждой конкретной тестовой программы описаны в документе "Описание тестовых программ. Руководство оператора".

14.2. Требуемая оперативная память

Конкретная тестовая программа требует для своей работы оперативную память емкостью не менее:

4к+м - для загрузки программой "Монитор системы";
8К+м - для загрузки программой "Обработка файлов" UPD1;
16к+м - для загрузки программой "Обработка файлов" UPD2,

где М - емкость оперативной памяти, необходимая для выполнения конкретной тестовой программы.

Монитор допускает выполнение последовательности команд при работе с тестовыми программами без вмешательства оператора путем выполнения командного файла.

14.3. Печать наименований тестовых программ

Вывод наименований тестовых программ и, соответствующих имен файлов выполняет после ее загрузки и запуска программа "Обработка файлов" UPD2 (см. раздел 6) по одной из команд:

TYPE уст:TEST.TXT <ВК> для вывода списка на экран терминала;
PRINT уст:TEST.TXT <ВК> для вывода на печать;

где уст: - имя устройства, на котором находится файл TEST.TXT с справочником по тестам.

Список тестов:

ZDSAA0 - тест дисплея символьного;
ZLPAA0 - тест параллельного интерфейса (печати);
ZDVAA0 - тест быстродействия;
ZMYAA0 - тест КМД и НГМД
ZTKAA0 - тест КТЛК на 6 каналов, имеющий адреса 176500 - 176556;
ZTKBA0 - тест КТЛК на 6 каналов, имеющий адреса 176560 - 176636;
ZTKCA0 - тест КТЛК на 12 каналов, имеющий адреса 176500 - 176636;
ZKBDA0 - тест КГД.

Для ВМ2:

BPCAA0 - основной тест команд;
BINAA0 - тест прерываний;
BKMAA0 - тест памяти.

Для ВМ3:

CDMAA1 - тест диспетчера памяти;
CINAA0 - тест прерываний;

CKMAA0 - тест памяти;
CPCBA0 - тест базовых команд процесора;
CPCCA0 - тест арифметических команд;
CKMBA0 - тест памяти короткий;
CKMCA0 - тест памяти длинный;
CPCAA0 - основной тест команд.

14.4. Выполнение тестовой программы

1) Установить носитель с ТМОС в устройство с номером 0;
2) Загрузить и запустить монитор системы (см. п.5.3);
3) Загрузить в оперативную память ЭВМ требуемую тестовую программу по команде монитора: L имя <ВК>

где имя - имя файла требуемой тестовой программы (см.п.14.3).

ПРИМЕЧАНИЕ. Если в тесте требуется задать режим работы, то следует отключить клавишу <ПРОГР>. Задайте режим работы. Включите клавишу <ПРОГР> и введите с клавиатуры терминала символ "Р" латинского алфавита.

Запустите тестовую программу по команде монитора:

S <ВК>

14.5. Выполнение тестовой цепочки

1) Установить носитель с ТМОС в устройство с номером 0;
2) Загрузить и запустить монитор системы (см. п.5.3.);
3) Запустить командный файл по команде монитора:

С имя <ВК>

где имя - имя командного файла с типом формата .CCC

Командный файл должен находиться на системном носителе.

Для запуска командного файла с именем SYSTEM.CCC используется команда монитора: TEST <ВК>

ПРИЛОЖЕНИЕ 1

НОСИТЕЛИ ТЕСТ-МОНИТОРНОЙ СИСТЕМЫ

У1.00032-01 МД 01	–	для накопителя типа "ЭЛЕКТРОНИКА НГМД-6021";
У1.00032-01 МД 02	–	для накопителя типа "ЭЛЕКТРОНИКА НГМД-6121";
У1.00032-01 МД 03	–	для накопителей типа "СМ-5640", "ЕС-5088".

ПРИЛОЖЕНИЕ 2

ОБОЗНАЧЕНИЕ ТИПА ИСПОЛЬЗУЕМОГО ПРОЦЕССОРА

Первый символ в названии программы – буква, обозначает тип используемого процессора, а именно:

Z – для всех типов процессоров;
K – системная программа
A – для процессора К1801ВМ1;
B – для процессора К1801ВМ2;
C – для процессора К1801ВМ3;
D – для процессора К1801ВМ4.

ПРИЛОЖЕНИЕ 3

ПРОЦЕДУРЫ НАЧАЛЬНОЙ ЗАГРУЗКИ

1. Загрузка с пульта без аппаратного загрузчика:

```
@172140/XXXXXX  37   < ПС >
@172142/XXXXXX  N    < ВК >
```

где N – номер окна привода накопителя (0 или 1), куда помещен носитель с монитором ТМОС. После нажатия клавиши <ВК> загорится лампочка на приводе накопителя, с которого производилась загрузка системы. Через одну-две секунды необходимо нажать клавишу <G> .

2. Загрузка с пульта с аппаратным загрузчиком:

```
@B
@MYN
```

где N – номер окна накопителя 0 или 1), куда помещен носитель с монитором ТМОС.

3. Загрузка по команде программы UPD2:

1) Загрузить программу UPD2 по команде монитора:

```
.R UPD2
```

2) После того, как на экране терминала высветится звездочка "*", ввести команду:

```
* BOOT уст: <ВК>
```

где уст: – имя устройства, с которого производится загрузка.

КОПИРОВАНИЕ ТМОС С ПОМОЩЬЮ КОМАНДНОГО ФАЙЛА

1) Установить гибкий диск с системой ТМОС в накопитель 0;
2) Установить гибкий диск для записи копии ТМОС в нокопитель 1;
3) Загрузить монитор системы согласно п. 5.3.
4) Подать с терминала следующую команду: .C SYSGEN/уст

где уст - имя устройства MY или MZ.

На экран терминала последовательно вызываются и выполняются команды файла SYSGEN.CCC. После того, как все содержимое диска, находящегося в накопителе 0, будет переписано на диск, находящийся в накопителе 1, на экран выдается распечатка каталога диска с копией ТМОС.

Процесс копирования системы ТМОС можно считать законченным после выдачи на экран точки ".".

П р и м е р :

```
IF MY THEN
R UPD2
COPY MY1:=MY0:
Y
I
DIR MY1:
EXIT
END
IF MZ THEN
COPY MZ1:=MZ0:
Y
I
DIR MZ1:
EXIT
END
```

ОПИСАНИЕ ТЕСТОВЫХ ПРОГРАММ. РУКОВОДСТВО ОПЕРАТОРА.

1. УСЛОВИЯ ВЫПОЛНЕНИЯ ТЕСТ-ПРОГРАММ

1.1. Технические средства

Для выполнения тест-программ необходима следующая минимальная конфигурация комплекса:

- процессор К1801ВМ1 или совместимые с ним К1801ВМ2, К1801ВМ3, К1801ВМ4;
- оперативная память емкостью не менее 56 кбайт;
- алфавитно-цифровой терминал;
- внешняя память на гибких магнитных дисках емкостью не менее 209 кбайт.

Дополнительно в состав технических средств могут входить:

- алфавитно-цифровое печатающее устройство;
- устройство формирования отображения графической информации на экране монитора;
- контроллеры телеграфных каналов (на 4-6 каналов);
- накопитель на жестких магнитных дисках типа винчестер.

1.2. Программные средства

Носитель тест-мониторной системы (см. приложение 1).

2. ОСНОВНОЙ ТЕСТ КОМАНД

2.1. Назначение программы

Тест-программа предназначена для проверки основных базовых команд микро-ЭВМ.

Состав базовой системы команд приведен в приложении 2;

Расширение базовой системы команд для ПЭВМ с микропроцессором К1801ВМ2 приведено в приложении 3;

Расширение базовой системы команд для ПЭВМ с микропроцессором К1801ВМ3 приведено в приложении 4.

2.2. Описание функций программы

Тест-программа состоит из 82 малых тестов, пронумерованных от 0 до 122.

При выполнении каждой проверяемой команды ее значение при заданных тестовых наборах сравнивается с эталонными. При несовпадении полученных значений с эталонными выдается сообщение об ошибке.

При правильном выполнении проверяемых команд тест-программа циклится, в противном случае происходит останов по ошибке.

Тесты по их назначению можно об'единить в следующие группы:

1) Тесты 0-15 проверяют выполнение команд ветвления JMP, JSR, MARK;
2) Тесты 16-31 проверяют выполнение байтовых команд с регистровым методом адресации;
3) Тесты 32-53 проверяют выполнение команд, оперирующих со словами, с регистровым методом адресации;
4) Тесты 54-62 проверяют все методы адресации, кроме регис-

трового, на командах MOVB, MOV, INCB, INC;

5) Тесты 63-76 проверяют выполнение байтовых команд со всеми методами адресации, кроме регистрового;

6) Тесты 77-122 проверяют выполнение команд, оперирующих со словами, со всеми методами адресации, кроме регистрового.

Соответствие между номером малого теста, начальным адресом и проверяемой командой, выполняемой функцией, приведено в табл.1.

Таблица 1

СООТВЕТСТВИЕ НОМЕРОВ ТЕСТОВ ВЫПОЛНЯЕМЫМ ФУНКЦИЯМ

Номер теста	Выполняемая функция	Номер теста	Выполняемая функция
1	2	3	4
	Команды ветвления JMP, JSR, MARK		
0	N=Z=V=C=0	7	BRANCH
1	N=1	10	JMP(1-ый метод адресации
2	N=V=1	11	JMP(2-ой, 3-ий методы адресации
3	N=V=C=1		
4	N=Z=V=C=1	12	JMP(4-ый, 5-ый методы адресации
5	N=Z=V=C=1, NVC=1		
6	N=Z=V=C=0	13	JMP(6-ой, 7-ой методы адресации
		14	JSR, MARK
		15	выборка регистров
	Байтовые команды с регистровыми методами адресации		
16	CLRB, MOVW	24	ROLB
17	CMPB, BISB	25	RORB
20	BICB, BITB	26	ASLB
21	INCB, DECB	27	ASRB
22	COMB	30	ADCB
23	NECB	31	SBCB
	Словные команды с регистровыми методами адресации		
32	TST, CLR, MOV	43	ASR
33	CMP, BIS	44	ADC
34	BIC, BIT	45	SBC
35	INC, DEC	46	SXT
36	COM	47	SWAP
37	NEC	50	XOR
40	ROL	51	ADD
41	ROR	52	SVB
42	ASL	53	MTPS, MFPS
	Проверка методов адресации, кроме регистрового, на командах MOVB, MOV; INCB, INC		

1	2	3	4
54	метод 0, 1, регистровый, косвенно-регистровый	60	метод 5, косвенно-автодекрементный
55	метод 2, автоинкрементный	61	метод 6, индексный
56	метод 3, косвенно-автоинкрементный	62	метод 7, косвенно-индексный
57	метод 4, автодекрементный		
	Проверка байтовых команд со всеми методами адресации, кроме регистрового		
63	TSTB, CLRB, MOVB	71	ROLB
64	CMPB, BISB	72	RORB
65	BICB, BITB	73	ASLB
66	INCB, DECB	74	ASRB
67	COMB	75	ADCB
70	NECB	76	SBCB
	Проверка словных команд со всеми методами адресации, кроме регистрового		
77	TST, CLR, MOV	111	ADC
100	CMP, BIS	112	SBC
101	BIC, BIT	113	SXT
102	INC, DEC	114	SWAB
103	COM	115	XOR
104	NEC	116	ADD
105	ROL	117	SUB
106	ROR	120	SOB
107	ASL	121	MTPS, MFPS
110	ASR	122	особые случаи выполнения байтовых команд

2.3. Режим работы

П У С К О В Ы Е А Д Р Е С А

000200 - выполнение тест-программы с теста 0;

000530 - выполнение тест-программы с выбранного малого теста. При этом номер теста заносится в ячейку 404.

П Е Р Е К Л Ю Ч А Т Е Л И

Установка кода 020000 в ячейку 000420 запрещает печать.

2.4. Сообщения оператору

П Р Е Д У С М О Т Р Е Н Н А Я П Е Ч А Т Ь

(печать при успешном тестировании)

После первого прохода и через каждые 256 проходов тест программы, при нормальной работе теста, печатается:

К П Р О Х О Д

П Е Ч А Т Ь П О О Ш И Б К Е

При восстановлении питания после его нарушения печатается:

П И Т А Н И Е

2.5. Останова

П Р Е Д У С М О Т Р Е Н Н Ы Е О С Т А Н О В Ы

Останов по адресу 000544 предназначен для задания в ячейку 000404 номера малого теста, начиная с которого должна выполняться тест-программа.

О С Т А Н О В Ы П О О Ш И Б К Е

Останов по адресу 000620 - 016702 означает, что произошла ошибка в текущем малом тесте или нарушена последовательность выполнения тестов. Число проходов тест-программы содержится в ячейке 000406, номер следующего теста - в ячейке 000404, номер ошибки - в ячейке 000402. Номера ошибок приведены в табл.2.

Останов по адресу 017076 - 017372 означает, что неправильно установился признак при выполнении проверяемой команды. В этом случае R6 указывает на ячейку стека, содержащую адрес команды, следующей за проверяемой.

Останов по адресу ловушки 000000 - 000776 означает, что произошло непредусмотренное прерывание. Адрес останова минус 2 является адресом вектора, по которому произошло прерывание, а R6 указывает на ячейку стека, содержащую адрес следующей невыполненной команды прерванной тест-программы.

Для возобновления выполнения тест-программы после устранения причины останова запустить ее с адреса 000200 или 000530 (см. п.2.3).

Таблица 2

НОМЕРА ОШИБОК ТЕСТОВ

N тестов	Номера ошибок восьмеричное	Типы ошибок
1	2	3
0-6	1 - 7	ошибки ветвления или НПТ
7	10	нарушен порядок выполнения тестов
	11,12,13	BR не выполнила ветвления
10	15,20	нет перехода: по JMP, JMP (R0) или НПТ
	16,17	JMP изменила: РСП, R0
11	21,24,25,27	нет перехода по: JMP (R0)+,
		JMP *(R0)+, или НПТ
	22	JMP изменила (РСП)
	23,26	(R0) не увеличился на 2
12	30,33,34,37	нет перехода по: JMP -(R0), JMP *-(R0)
	31,35	R0 не уменьшился на 2
13	40,43-46	нет перехода по: JMP -6(R3), JMP 0(R3)
		JMP *0(R3), JMP *4(R0), JMP *0(R0)

1	2	3
14	47	нет перехода по JSR PC,2252
	50,60,63	RTS не восстановила (УС)
	51,56	JSR изменила (РПС)
	52	указатель стека не уменьшился на 2
	53	адрес возврата не равен 2224
	54	нет перехода по RTS PC
	55	нет перехода по JSR R4,2520
	57	MARK не восстановила (R5)
	61,62	нет перехода по JSR R0,(R1)
	64	адрес возврата (R4) не равен 237
	65	нет перехода по RTS R5
	66	адрес возврата (R0) не равен 2462
	67	нет перехода по RTS R0
15	70	неправильная выборка регистра или НПТ
16	71	нарушен порядок тестов
17	72,73,74	ошибка: BISB #377,R2 или CMPB R0,R2
		CMPB R2,R0 или НПТ
20	75	нарушен порядок тестов
21	76,77	ошибка BIC R0,R3 или BITB R0,R3
		BISB R0,R3
	100	нарушен порядок тестов
	101	ошибки: INCB R4
22	102	нарушен порядок тестов
	103,104	COMB R3
23	105,106	ошибка NECB R0
24	107-110	ошибка ROLB R1
25	111,112	ошибка ROLB R2
26	113	нарушен порядок тестов
	114	ошибка ASLB R3
27	115,116	ошибка ASRB R3
30	117	нарушен порядок тестов
	120,121	ошибка ADCB R0
31	122	нарушен порядок тестов
	123-125	ошибка SBCB R1
32	126	нарушен порядок тестов
	127	ошибка MOV # -1,R4 или TST R1

1	2	3
33	130-132	ошибка CMP R0,R2 CMP #77,R2 CMP R2 #77
34	133-135	ошибка BIC (R0),R3 или BIT (R0),R3
		BIS #1525252,R3 BIC R0,R0
35	136	нарушен порядок тестов
	137	ошибка INC R4
36	140	нарушен порядок тестов
	141,142	ошибка COM R3
37	143,144	ошибка NEG R0 или НПТ
40	145,146	ошибка ROL R1 или НПТ
41	147,150	ошибка ROR R2 или НПТ
42	151	нарушен порядок тестов
	152	ошибка ASL R3
43	153	ошибка ASR R4
	154	ошибка ASR R3 или НПТ
44	155	нарушен порядок тестов
	156,157	ошибка ADC R0
45	160	нарушен порядок тестов
	161,162,163	ошибка SBC R1
46	164,165	ошибка SXT R2 или НПТ
47	166,167	ошибка SWAP R3 или НПТ
50	170	ошибка XOR R2,R4 или НПТ
51	171	ошибка ADD R1,R1
	172	ошибка ADD R0,R0
	173	ошибка ADD R4,R1
	174	ошибка ADD R1,R4 или НПТ
52	175	нарушен порядок тестов
	176	ошибка SUB R2,R3
	177	ошибка SUB R4,R3
	200	ошибка SUB R3,R2
53	201,202	ошибка MFPS R1 или НПТ
54	203,204	ошибка метода 0
	205,206	ошибка метода 1 или НПТ
55	207,210	ошибка метода 2 или НПТ

1	2	3
56	211,212	ошибка метода 3 или НПТ
57	213,214,215, 216,217	ошибка метода 4 или НПТ
60	220,221,222	ошибка метода 5 или НПТ
61	223,224	ошибка метода 6 или НПТ
62	225,226	ошибка метода 7 или НПТ
63	227	ошибка MOVB #200,0(R2) или
		MOVB (R2)+,-(R1)
	230	ошибка MOVB (R2)+,-(R1) или НПТ
64	231	ошибка CMPB R4,(R2)
	232,233	ошибка CMPB (R1),(R2)
65	234	нарушен порядок тестов
	235	ошибка BITB -1(R1),(R3)
	236	ошибка BISB -(R1),(R3)
	237,240	ошибка BICB *-(R0),*0(R0)
66	241	ошибка INCB -1(R4)
	242	ошибка DECB -(R4) или DECB 0(R4) или НПТ
67	243	нарушен порядок тестов
	244,245	ошибка COMB (R3)
70	246,247	ошибка NEGB (R0) или НПТ
71	250,251	ошибка ROLB (R1) или НПТ
72	252,253	ошибка RORB (R2) или НПТ
73	24	нарушен порядок тестов
	255	ошибка ASLB (R3)
74	256	ошибка ASRB (R4)
	257	ошибка ASRB (R3) или НПТ
75	260	нарушен порядок тестов
	261,262	ошибка ADCB (R0)
76	263	нарушен порядок тестов
	264,265,266	ошибка SBCB (R1)
77	267	нарушен порядок тестов
100	270	ошибка CMP (R2)+, # -1
	271	(R2) не увеличился на 2
	272	ошибка BIS *1777776(R4),*(R2)+ или

1	2	3
		!CMP *-(R0),# -1
	!273	!(R2) не увеличился на 2
	!274	!ошибка CMP (R0)+,(R0)+ или
		!BIS *-(R0),*2(R0) или НПТ
!101	!275	!ошибка BIC (R0)+,(R3) или BIT -(R0),(R3)
	!276	!ошибка BIS 0(R0),(R3)
	!277	!ошибка BIC 177776(R2),(R3)
	!300	!ошибка BIC *-(R0),R0 или
		!BIT *177776(R4),(R3) или BIC 0(SP),-(R3)
	!301	!ошибка BIT *-(R0),(SP)+ или НПТ
!102	!302	!нарушен порядок тестов
	!303	!ошибка INC (R4)
!103	!304	!нарушен порядок тестов
	!305	!ошибка COM 0(R3)
	!306	!ошибка COM (R3)+
!104	!307	!ошибка NEG -(R4)
	!310	!ошибка NEG (R4) или НПТ
!105	!311	!ошибка ROL (R1)+ или ROL -(R1)
	!312	!ошибка ROL (R2) или НПТ
!106	!313,314	!ошибка ROR (R2) или НПТ
!107	!315	!нарушен порядок тестов
	!316	!ошибка ASL (R3)
!110	!317	!ошибка ASR (R4)
	!320	!ошибка ASR (R3) или НПТ
!111	!321	!нарушен порядок тестов
	!322,323	!ошибка ADC (R0)
!112	!324	!нарушен порядок тестов
	!325,326,327	!ошибка SBC (R1)
!113	!330,331	!ошибка SXT (R2) или НПТ
!114	!332	!ошибка SWAB (R3)
	!333	!ошибка SWAB 0(R3) или НПТ
!115	!334	!ошибка XOR R1,-(R0) или НПТ
!116	!335	!ошибка ADD (R0),(R1)
	!336	!ошибка ADD R0,(R0)+
	!337	!ошибка ADD -(R0),(R1)
	!340	!ошибка ADD *-(R0),0(SP)
	!341	!ошибка ADD #137777,TEMP 2

1	2	3
117	342,343	ошибка SUB (R2),(R3)
	344	ошибка SUB (R3),(R2)
	345	ошибка SUB #77777 , TEMP или НПТ
120	346	переход при (R0) = 0
	347	нет перехода при (R0), не равном 0
	350	число переходов не равно 12
	351	нет перехода при (R4), не равном 0
		или НПТ
121	352	ошибка MFPS (R1)
	353	ошибка MFPS TEMP1 или MTPS #377 или НПТ
122	354,355	MOVB не расширила знак
	356	(УС) не увеличился на 2
	357	в стеке не 377
	360	(УС) не уменьшился на 2
	361	ошибка MOVB -(SP),TEMP+1
	362	ошибка COMB TEMP+1 или НПТ
	364	NZVC не равно 0000
	365	NZVC не равно 0001
	366	NZVC не равно 0010
	367	NZVC не равно 0011
	370	NZVC не равно 0100
	371	NZVC не равно 0101
	372	NZVC не равно 0111
	373	NZVC не равно 1000
	374	NZVC не равно 1001
	375	NZVC не равно 1010
	376	NZVC не равно 1011
	377	NZVC не равно 1111

2.6. Время

Время выполнения первого прохода тест-программы - 1с, каждых последующих 256 проходов - 5с.

2.7. Выполнение программы

З А Г Р У З К А Т Е С Т - П Р О Г Р А М М Ы

Перед загрузкой тест программы необходимо загрузить тест-мониторную систему согласно документа "Описание системы" Руководство оператора".

Загрузить тест-программу "Основной тест команд" оперативную память с носителя TMOC по команде монитора:

L CPCAA0 <ВК>

В Ы П О Л Н Е Н И Е П Р О Г Р А М М Ы

Установить режим работы тест-программы (см. п.2.3.), написав в ячейку 000420 один из кодов:

000000 для разрешения печати или

020000 для запрещения печати.

Запустить тест-программу с адреса 000200, либо по команде монитора: S 200 <ВК>,

либо по команде пультового терминала 2006, предварительно выключив и включив (дважды нажав) клавишу <ПРОГР>.

Если заданный по умолчанию режим печати не нужно менять то вместо команд L CPCAA0 <ВК> и S 200 <ВК> рекомендуется выполнить команду монитора: R CPCAA0 <ВК>

При правильном выполнении основных команд после первого прохода (через 1с) и через каждые 256 проходов (через 5с) печатается:

" К П Р О Х О Д "

Для останова выполнения тест-программы отключить клавишу <ПРОГР>.

При неправильной работе процессора происходит останов (см. п. 2.5.).

Для выполнения тест-программы с малого теста N необходимо занести в ячейку 000404 номер теста N и запустить ее с адреса 000530. После останова по адресу 000544 занести в R7 начальный адрес выбранного малого теста и продолжить выполнение тест-программы, набрав на терминале "Р" (буква Р на нижнем регистре клавиатуры).

ПРИМЕЧАНИЕ. При запуске тест-программы с адреса 000530 в случае несоответствия номера малого теста его начальному адресу происходит останов по нарушению последовательности выполнения тестов. Адреса остановов 000620 - 016702.

3. ТЕСТ ПРЕРЫВАНИЙ

3.1. Назначение программы

Тест прерываний проверяет все операции и команды, вызывающие прерывания, а также правильность обработки возникающих прерываний.

3.2. Загрузка и запуск

Программа теста прерываний загружается и запускается командой монитора: R CINAA0 <ВК>

3.3. Описание функций программы

Программа состоит из 73 малых тестов, пронумерованных от нуля до 114.

Первые четыре теста непосредственно не выполняют проверку правильности обработки прерываний. Однако без этих тестов нельзя делать выводы о правильной работе блока прерывания.

Тесты прерываний выполняют проверку:

- режима автоуменьшения и автоувеличения регистра R6 для слов и байтов - тест 1;
- передачи байтов из/в регистр R6 - тест 2;
- выполнения операций с четными и нечетными байтами - тест 3;
- выполнения команд управления кодами условий слова состояния процессора - тест 4;
- правильности обработки возникшего прерывания при выполнении резервной команды (уменьшение указателя стека на 4, сохранение в стеке счетчика команд и слова состояния процессора, выборка новых значений счетчика

команд и слова состояния из вектора прерывания)- тесты 5-11;
- правильности обработки прерывания по команде TRAP (код 104400) - тесты 13-16;
- возникновения прерываний по всем кодам команды TRAP (104400 - 104777) - тест 17;
- правильности обработки прерывания по команде IOT (код 000004) - тесты 20-24;
- правильности обработки прерывания по команде EMT (код 104000) - тесты 25-31;
- возникновения прерываний по всем кодам команды EMT (104400 - 104377) - тест 32;
- правильности обработки прерывания по команде BPT (код 000003) - тесты 33-37;
- правильности обработки прерываний при использовании команд с запрещенными режимами адресации (JMP RN, JSR RN,RN N = 0,1...6)-тесты 40-44 и 45-50 соответственно;
- выполнения и обработки прерывания по неправильной адресации - тесты 52-56;
- выполнения и обработки прерываний по переполнению стека - тесты 60-71;
- выполнения и обработки прерываний по Т-биту - тесты 72-74;
- выполнения команд RTT и RTI - тесты 75-77;
- выполнения и обработки прерываний по нечетному адресу в R7 - тест 100;
- выполнения команд RTS и IOT - тест 101;
- сохранения Т-бита в стеке - тест 102;
- возникновения прерываний при обращении к несуществующей памяти - тест 103;
- обработки прерывания по переполнению стека, вызванного обработкой прерывания от видеотерминала - тест 104;
- обработки вложенных прерываний - тест 105;
- приоритета прерываний - тест 106;
- выполнения команды RESET - тесты 107-110;
- выполнения прерываний от видеотерминала - тест 111;
- выполнения прерывания на команды WAIT - тест 112;
- выполнения прерываний диспетчера памяти - тест 113 (только для микро-ЭВМ с диспетчером памяти);
- выполнения прерываний по всем резервным командам- тест 114.

3.4. Сообщения оператору

В конце прохода теста прерываний выводится сообщение:

КОНЕЦ ПРОХОДА ТЕСТА ПРЕРЫВАНИЙ

и происходит переход на начало теста, если программа работает не в режиме цепочки тест-мониторной системы. В противном случае происходит возврат в монитор по адресу, содержащемуся в 42 ячейке.

При обнаружении ошибки любым из тестов происходит останов. Более подробная информация об ошибке, вызвавшей останов, содержится в тексте программы.

Если необходимо зациклить тест, обнаруживший ошибку, следует команду HALT заменить командой NOP (по адресу команды HALT занести код 240), а следующую команду заменить ко-

мандой безусловного перехода, код которой содержится в комментарии к данной команде останова.

4. ТЕСТ ПАМЯТИ

4.1. Назначение программы

Тест-программа предназначена для проверки оперативной памяти емкостью от 8к до 248к байт.

4.2. Описание программы

Тест-программа состоит из 12 тестов, пронумерованных от 0 до 13 (восьмеричное).

Тест 0. Тест выбора банка
Тест 1. Тест движения нулей и единиц
Тест 2. Тест записи/считывания по байтам
Тест 3. Тест адреса А
Тест 4. Тест адреса В
Тест 5. Тест перестановки байтов
Тест 6. Тест сохранения данных
Тест 7. Тест смещения кодов по диагонали
Тест 10. Тест галопирования
Тест 11. Тест длинного галопирования
Тест 12. Тест максимальных помех
Тест 13. Тест восстановления

Каждый тест последовательно проверяет банки памяти, начиная с первого банка (банка, имеющего наименьший номер среди проверяемых) и до последнего банка (банка, имеющего наибольший номер среди проверяемых). После выполнения всех тестов тест-программа перемещается и проверяются ячейки нулевого банка, в которых располагалась программа. Затем тест-программа перемещается на прежнее место и проверка повторяется, т.е. программа работает циклически, причем перед выполнением каждого теста все ячейки проверяемой памяти очищаются.

Пределы проверяемой памяти устанавливаются автоматически или вручную (см. п.4.4.).

Если при проверке памяти обнаружена ошибка, то выдается сообщение об ошибке (см. табл.4).

ПРИМЕЧАНИЕ 1. Банк определяет память емкостью 20000 байт. Нулевой банк содержит ячейки с 000000 по 017777, первый с 020000 по 037777 и т.д.

ПРИМЕЧАНИЕ 2. Тест-программа (000430-007776) при КР09=0 перемещается на длину, равную разности между верхним пределом проверяемой памяти и адресом - 2 первой ячейки первого проверяемого банка.

4.3. Описание тестов

Т Е С Т 0 - тест выбора банка

В первую ячейку первого проверяемого банка записывается код 177777, считывается и проверяется содержимое первых ячеек остальных банков.

Затем ячейка, в которую был записан код 177777, очищается, а в первую ячейку следующего банка записывается код 177777. После этого считывается и проверяется содержимое первых ячеек остальных банков и так до тех пор, пока не бу-

дет проверен последний банк.

Если содержимое одной из проверяемых ячеек окажется равным 177777, то выдается сообщение об ошибке 6.

Если содержимое одной из проверяемых ячеек окажется отличным от 177777 и 000000, то выдается сообщение об ошибке 7.

ПРИМЕЧАНИЕ 1. Первая ячейка проверяемой памяти емкостью 8 кбайт - 010266, 16 кбайт - 010310, 24к байт - 010332 и т.д.

Т Е С Т 1 - тест движения нулей и единиц

В первую ячейку первого проверяемого банка записывается код 000001. Затем этот код считывается и проверяется. При правильном выполнении записи/считывания этот код сдвигается на один разряд влево и вновь записывается в ту же самую ячейку банка и весь процесс повторяется до тех пор, пока "1" не пройдет через все разряды проверяемой ячейки. Такая проверка осуществляется для всех ячеек проверяемых банков.

Аналогично тест проверяет поразрядную запись/считывание при движении "0". В этом случае в первую ячейку первого банка записывается код 177776.

В случае ошибки при записи/считывании будет выдаваться сообщение об ошибке 13.

Т Е С Т 2 - тест записи/считывания по байтам

В каждую ячейку проверяемой памяти, начиная с первой ячейки первого проверяемого банка, побайтно записываются коды 377 - в младший байт и 000 - в старший байт. Затем эти коды считываются, начиная со старшего адреса памяти, и проверяются.

Аналогичная проверка осуществляется на кодах 000 для младшего байта и 377 для старшего байта.

В случае возникновения ошибки при считывании выдается сообщение об ошибке 15.

Т Е С Т 3 - тест адресов А

Во все ячейки проверяемой памяти записывается код 000377. Затем в первую ячейку первого проверяемого банка записывается код 177400 (обратный код 000377), считывается и проверяется содержимое остальных ячеек этого банка. Далее в следующую ячейку первого банка записывается код 177400, считывается и проверяется содержимое остальных ячеек банка, причем, предварительно восстанавливается содержимое первой ячейки. Таким образом проверяются все ячейки проверяемых банков.

Если содержимое одной из проверяемых ячеек окажется отличным от 000377, то выдается сообщение об ошибке 17.

Если содержимое одной из проверяемых ячеек окажется равным 177400, то выдается сообщение об ошибке 20.

Если произошла ошибка при записи/считывании кода 177400, то выдается сообщение об ошибке 21.

Т Е С Т 4 - тест адресный

В каждую ячейку проверяемой памяти, начиная с первой ячейки первого проверяемого банка, записывается содержимое, равное ее адресу. Затем содержимое всех ячеек, начиная со старшего адреса, считывается и проверяется. Далее в каждую ячейку проверяемой памяти, начиная с первой ячейки первого банка, записывается содержимое, равное ее инвертированному адресу, и начинается проверка, начиная со старшего адреса.

В случае несовпадения содержимого ячейки с ее адресом

или с ее инвертированным адресом, выдается сообщение об ошибке 23.

Т Е С Т 5 - тест перестановки байтов

Во все ячейки проверяемой памяти записывается код 000377 и выполняется следующее:

1) Содержимое каждой ячейки проверяемой памяти, начиная со старшего адреса - 2, считывается и проверяется, затем записывается код 177400, считывается и проверяется;

2) Содержимое каждой ячейки проверяемой памяти, начиная с младшего адреса, считывается и проверяется, затем записывается код 000377 и вновь считывается и проверяется;

3) Повторяется п. 1), только проверка начинается с младшего адреса;

4) Повторяется п. 2), только проверка начинается со старшего адреса - 2.

В случае несовпадения считанного и записанного кода выдается сообщение об ошибке 25, 26 или 27.

Т Е С Т 6 - тест сохранения данных

Во все ячейки проверяемой памяти записывается код 000377, считывается и проверяется. После этого в первую ячейку первого проверяемого банка переставляются байты 2048, а затем содержимое ячеек вновь считывается и проверяется. Далее аналогично переставляются байты в ячейке следующего банка, соответствующей первой ячейке первого банка и т.д. до тех пор, пока не будут проверены все банки.

Затем во все ячейки проверяемой памяти записывается код 177400 и весь процесс повторяется.

Если при записи/считывании кодов 000377 или 177400 произойдет ошибка, то будет выдаваться сообщение об ошибке 31.

Т Е С Т 7 - тест смещения кодов по диагонали

Начиная с первой ячейки первого проверяемого банка и далее через каждые 200 (восьмеричное) ячеек банка, записывается код 177400, а в остальные ячейки этого банка записывается код 000377. После этого считывается и проверяется содержимое всех ячеек первого банка. Далее записанные коды смещаются к старшему адресу первого банка на одну ячейку 64 раза. После каждого смещения на одну ячейку считывается и проверяется содержимое всех ячеек первого банка. Таким образом проверяются все банки проверяемой памяти.

Затем, начиная с первой ячейки первого проверяемого банка и далее через каждые 200 (восьмеричное) ячеек банка, записывается 177400 И весь процесс повторяется.

Если при проверке окажется, что записанные и считанные коды не совпадают, то будет выдаваться сообщение об ошибке 33.

Т Е С Т 10 - тест галопирования

Во все ячейки проверяемой памяти записывается код 000377.

Назовем одну ячейку "А", а другую "В". Первоначально "А" и "В" совпадают с первой ячейкой первого проверяемого банка.

Далее выполняется следующее:

1) "В" = "А" ;

2) переставливаются байты в ячейке "А";

3) считывается и проверяется содержимое ячеек "А" и "В";

4) "B" = "B" + 200(восьмеричное);

5) повторяются п. 3) и 4) до тех пор, пока "B" не станет больше старшего адреса проверяемого банка;

6) "A" = "A" + 2;

7) повторяются п. 1) - 6) до тех пор, пока "A" не достигнет старшего адреса проверяемого банка;

8) повторяются п. 1) - 7) для каждого проверяемого банка, причем, в начале проверки каждого банка ячейка "B" ячейка "B" будет первой ячейкой того банка, который проверяется.

После выполнения п. 1) - 8) во все ячейки проверяемой памяти записывается код 177400 и п. 1) - 8) повторяются.

Если при записи/считывании кодов 000377 или 177400 произошла ошибка, то будет выдаваться сообщение об ошибке 35,36 или 37.

Т Е С Т 11 - тест длинного галопирования

Основное отличие данного теста от теста галопирования заключается в том, что вместо выбора ячеек "B" с шагом 200(восьмеричное), выполняется шаг выбора, равный 2 ("B" = "B" + 2).

Во все ячейки проверяемой памяти записывается код 000377.

Назовем одну ячейку "A", а другую - "B". Первоначально "A" и "B" совпадают с первой ячейкои первого проверяемого банка.

Далее выполняется следующее:

1) "B" = "A";

2) переставляются байты в ячейке "A";

3) считывается и проверяется содержимое ячеек "A" и "B";

4) "B" = "B" + 2;

5) если КР07=1, то п. 3) и 4) повторяются до тех пор, пока "B" не достигает старшего адреса проверяемого банка. Затем к "A" прибавляется 2 и п. 1) - 4) повторяются до тех пор, пока "A" не достигнет старшего адреса неового проверяемого банка. Таким образом проверяются все банки, причем, в начале проверка каждого банка "B" указывает на первую ячейку проверяемого банка.

Если КР07=0, то проверка происходит аналогично, только в каждом банке будет проверяться до 200(восьмеричное) ячеек.

Затем во все ячейки проверяемой памяти вместо кода 000377 записывается код 177400 и п. 1) - 5) повторяются.

Сообщения об ошибках, выдаваемые в данном тесте, аналогичны сообщениям в тесте 10.

Т Е С Т 12 - тест максимальных помех

Код 177400 записывается по тем адресам проверяемой памяти, для 1-го и 8-го разрядов которых результат выполнения логической операции "исключающее или" равно 1, и код 000377 в остальные ячейки проверяемои памяти. Затем, начиная с младшего адреса, считывается и поозеряется содержимое каждой ячеики, инвертируется и вновь считывается. Аналогичный процесс повторяется, начиная со старшего адреса.

Далее код 177400 записывается по тем адресам, для 8-го и 13-го разрядов которых результат выполнения логической операции "исключающее или" равен 1, и код 000377 - в осталь-

ные ячейки проверяемой памяти. Затем происходит считывание, как было указано выше.

После этого код 177400 записывается по тем адресам, для 3-го и 9-го разрядов которых результат выполнения логической операции "исключающее или" равен 1, и код 000377 - в остальные ячейки проверяемой памяти и вновь считывается.

Далее весь процесс повторяется, только вместо кода 177400 записывается код 000377, а вместо кода 000377 - код 177400. Если произошла ошибка при записи/считывании кода 000377 или 177400, то будет выдаваться сообщение об ошибке 42 или 43.

Т Е С Т 13 - тест восстановления

В первую половину проверяемого банка записывается код 010247 команда "MOV R2,-(R7), а во вторую половину - код 177667. R2 содержит код 005141 команда "COM-(R1)", а R1 первоначально равен старшему адресу проверяемого банка.

После записи кодов происходит их считывание и проверка. Затем управление передается первой проверяемой ячейке:

1) Выполняется команда MOV R2,-(R7), по которой содержимое ячейки заменяется на содержимое R2, и управление передается этой же ячейке;

2) Выполняется команда COM-(R1), по которой код 177667, записанный по адресу R1, заменяется на код 000110 - код команды JMP(R0). Далее управление передается следующей ячейке и процесс повторяется до тех пор, пока не будут выполнены все команды первой половины банка, по которым коды, записанные во вторую половину банка, заменяются на коды 000110.

После этого управление передается первой ячейке второй половины банка, выполняется команда JMP(R0), в (R0) - адрес возврата к началу теста. Аналогично проверяются остальные банки проверяемой памяти.

Если произошла ошибка при записи/считывании кода команды MOV R2,-(R7), то выдается сообщение об ошибке 45, если при записи/считывании кода 177667, то выдается сообщение об ошибке 47.

4.4. Режим работы

П У С К О В Ы Е А Д Р Е С А

000200 - выполнение тест-программы с теста 0.

У С Т А Н О В К А Р Е Ж И М А

Для установки режима работы используются соответствующие разряды ячейки 000176 (см. табл.3).

Таблица 3

КР	Режим работы
1	2
15 (100000)	останов по ошибке
14 (040000)	зацикливание теста, номер которого установлен в разрядах 03-00
13 (020000)	запрещение печати по ошибке

1	2
12 (010000)	проверка памяти емкостью более 56 кбайт
10 (002000)	останов после выполнения очередного теста
09 (001000)	запрещение перемещения тест-программы
08 (000400)	печать 1-го отказавшего разряда в каждом банке
07 (000200)	выполнение варианта длинного галопирования в тесте 11
06 (000100)	запрещение автоматического определения пределов памяти
05 (000040)	запрещение печати "К ПРОХОД#XX"
04 (000020)	запрещение печати "ГЛП", "ТСТ13 БАНК XX" и причины ошибки
03 - 00	номер теста

Нажатие клавиш <СУ>/<С> во время работы тест-программы вызывает печать "^C" после окончания выполнения текущего теста, перемещение тест-программы в банк 00 и останов по адресу 006212.

4.5. Сообщения оператору

Предусмотренная печать по нормальной работе теста

1) После запуска тест-программы с адреса 000200 печатается:

```
ПАМЯТЬ
XXXXXX - YYYYYY
ГЛП
```

где XXXXXX - нижний предел проверяемой памяти;
YYYYYY - верхний предел проверяемой памяти;
"ГЛП" печатается только в том случае, если выбран вариант длинного галопирования (КР07=1).

2) После проверки тестами всех банков проверяемой памяти 00, 01, 02 и т.д. печатается:

```
ТСТ13 БАНК NN
ТСТ13 БАНК NM
.............
ТСТ13 БАНК MM
```

3) Перед перемещением тест-программы печатается:

```
ПЕРЕМ
```

а после перемещения тест-программы и проверки банка 00 печатается: ТСТ13 БАНК XX

4) После окончания очередного прохода тест-программы печатается: К ПРОХОД # XX

где XX - номер прохода.

П Е Ч А Т Ь П О О Ш И Б К Е

Если при проверке памяти обнаружена ошибка, то на экран терминала выдается сообщение об ошибке. Номера ошибок и возможные причины их возникновения приведены в табл.4. В квадратных скобках указано наименование подпрограммы, обнаружившей ошибку.

В случае возникновения ошибок с номерами 1, 3-5, 11, 12, 14, 16, 22, 24, 30, 32, 34, 40, 41, 44, 52 печатается:

ОШИБ # XX

где XX - номер ошибки.

В случае возникновения ошибки с номером 52 дополнительно печатается: НЕТ ДП

В остальных случаях выдается сообщение, состоящее из шести восьмеричных чисел:

1-ое - адрес отказавшей ячейки памяти;
2-ое - ожидаемые данные;
3-е - полученные данные;
4-ое - содержимое счетчика команд;
6-ое - содержимое счетчика проходов теста.

Если произошла ошибка адреса (см. табл.4), то перед сообщением об ошибке печатается:

ОШИБ А

После печати сообщения об ошибке может печататься причина ошибки, состоящая из 3-х чисел:

1-ое - десятичный номер отказавшего разряда (0-15);
2-ое - номер банка памяти, в котором произошел отказ;
3-е - число отказов.

Таблица 4

Номер ошибки	Причина возникновения ошибки
1	2
1	[TSTTPT] ошибка данных в ячейке 00000 - 000043 R0-содержит правильные данные, R1 - адрес отказавшей ячейки.
3	[TSTSIZ] ошибка оператора. Выбрана память емкостью более 56 кбайт, а КР12 не установлен (см.табл.3). Установить КР12=1 и повторно запустить тест-программу с адреса 000200.
4	[TSTSIZ] ошибка оператора. Нижний предел проверяемой памяти больше верхнего. Установить правильно пределы проверяемой памяти в ячейках 000322-000330 и запустить тест-программу с адреса 000200.
5	[TST0] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен 000.
6	[TST0] ошибка адреса.

1	2
	Напечатанные на экране терминала ожидаемые данные являются адресом. Этот адрес был выбран, когда эти же данные записывались в отказавшую ячейку памяти.
7	[TST0] ошибка адреса и данных. Идентична предыдущей ошибке, только данные, записанные в отказавшую ячейку, являются неправильными.
10	[TST0] ошибка данных. Сравните ожидаемые и полученные данные.
11	[TST0] обращение к несуществующему адресу.
12	[TST1] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен 001.
13	[TST1] ошибка данных. Сравните ожидаемые и полученные данные.
14	[TST2] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен 002.
15	[TST2] ошибка адреса и данных. Сравните ожидаемые и полученные данные.
16	[TST3] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен 003.
17	[TST3] ошибка адреса. Сравните ожидаемые и полученные данные.
20	[TST3] ошибка адреса. Для этой ошибки напечатанные на экране терминала данные являются адресом. Этот адрес был выбран при записи этих же данных в отказавшую ячейку.
21	[TST3] ошибка адреса. Идентична предыдущей ошибке, только другие данные.
22	[TST4] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен 004.
23	[TST4] ошибка адреса. Если число проходов данного теста равно 0, то отказавшая ячейка и полученные данные являются двойными адресами.

1	2
24	[TST5] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен 005.
25	[TST5] ошибка данных. Ошибка записи или считывания данных.
26	[TST5] ошибка данных. Идентична ошибке 25.
27	[TST5] ошибка данных. Идентична ошибке 25.
30	[TST6] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен 006.
31	[TST6] ошибка записи/считывания. Если содержимое счетчика проходов теста равно 0, то ошибка записи или считывания кода 000377. Если равно 1, то содержимое ячейки, в которой переставлялись байты, изменилось. Если равно 2, то ошибка записи или считывания кода 177440.
32	[TST7] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен 007.
33	[TST7] ошибка записи/считывания. Если содержимое счетчика проходов теста равно 0, то ошибка записи или считывания кода 000377. Если равно 1, то ошибка считывания кода 000377. Если больше 1 и четное число, то ошибка записи в отказавшую ячейку. Если больше 1 и нечетное число, то ошибка считывания данных из отказавшей ячейки.
34	[TST10] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен 010.
35	[TST10] ошибка данных. Ошибка записи или считывания кода 000377.
36	[TST10] или [TST11] ошибка данных. Сравните ожидаемые и полученные данные. Номер теста который обнаружил ошибку, указан в ячейке 000404.
37	[TST10] ошибка данных. Идентична ошибке 36.
40	[TST11] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен

1	2
	011.
41	[TST12] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должнен быть равен 012.
42	[TST12] ошибка данных. Если содержимое счетчика проходов теста равно 1, то сравните ожидаемые и полученные данные для определения отказавших разрядов. Если равно 2, то данные в отказавшую ячейку были записаны и считаны правильно, но изменились при считывании в направлении от старшего адреса к младшему адресу проверяемой памяти. Если равно 3, то идентичная ошибка, только считывание происходит в направлении от младшего адреса к старшему.
43	[TST12] ошибка данных. Идентична ошибке 42, только другие данные.
44	[TST13] нарушена последовательность выполнения тестов. Младший байт ячейки 000404 должен быть равен 013.
45	[TST13] ошибка данных. Сравните ожидаемые и полученные данные.
46	[TST13] ошибка данных, обнаруженная перед работой теста 13. Тест 13 прекращает работу.
47	[TST13] ошибка данных. Идентична ошибке 45, только другие данные.
52	[NOMM] ошибка оператора. Выбрана проверка памяти емкостью более 56 кбайт, а диспетчера памяти нет.

4.6. Остановы

Предусмотренные остановы приведены в табл.5.

Таблица 5

Адрес останова	Причина останова и действия оператора
1	2
001664	Конец выполнения текущего теста (КР10=1). Нажать клавишу с буквой Р на нижнем регистре

Остановы по ошибке приведены в табл.6.

Таблица 6

Адрес останова	Причина останова и действия оператора
1	2
000000	Непредусмотренное прерывание.
000776	Адрес останова минус 2 является адресом вектора, по которому произошло прерывание, а R6 указывает на ячейку стека, содержащую адрес следующей невыполненной команды прерванной тест-программы.
006124	Останов по ошибке (КР15=1). Для продолжения тест-программы нажать "Р".
006210	Останов по ошибкам 1,3-5,11,12,14,16,22,24,30,32,34,40,41,44,52 или было нажато <СУ>/<С>. После устранения ошибки для продолжения тест-программы нажать "Р".

В случае останова тест-программы после перемещения неперемещенный адрес останова можно вычислить следующим образом: из адреса останова в перемещенной тест-программе вычесть содержимое ячейки 000346 и прибавить 500.

4.7. Время

Время выполнения прохода тест-программы для КР=000000 10-15 мин.

В случае выбора варианта длинного галопирования (КР07=1) время выполнения тест-программы увеличивается в 25 раз.

4.8. Загрузка и выполнение программы

Переменной КР, определяющей режим работы теста, соответствует ячейка 000176.

Загрузить тест-программу "Тест памяти" в оперативную память с носителя ТМОС по команде: L СКМСА0 <ВК>

Запустить тест-программу с адреса 000200, установив предварительно на КР необходимый режим работы (см. табл.3, по умолчанию КР=000000): S 200 <ВК>

ПРИМЕЧАНИЕ. Режим работы тест-программы (см.п.4.4.) можно изменять во время ее работы, переустанавливая переключатели КР.

Для прекращения работы тест-программы отжать клавишу <ПУЛЬТ>.

ПРИМЕЧАНИЕ. Для перемещения тест-программы в банк 00 или останова в конце работы текущего теста нажать <СУ>/<С>.

Для проверки области памяти, задаваемой оператором, необходимо перед запуском тест-программы записать в ячейки:

000324 - младший адрес выбранной области памяти,

000330 - старший адрес выбранной области памяти

и запустить тест-программу с адреса 000200, установив КР06=1

и КР09=1.

5. ТЕСТ ДИСПЛЕЯ СИМВОЛЬНОГО

5.1. Назначение программы

Тест-программа "Тест дисплея символьного" предназначена для визуальной проверки работы дисплея символьного типа 15ИЭ-00-013.

Тест-программа состоит из пяти тестов, пронумерованных с 1 до 5. Тесты с 1 по 3 обеспечивают выход на экран дисплея последовательности символов. Тесты 4 и 5 обеспечивают проверку правильности ввода символов с терминала.

Т Е С Т 1

Заполняет символами "Е" все позиции экрана дисплея.

Т Е С Т 2

Выводит на экран диплея символы с кодами 40-176.

Вид изображения представляет собой последовательность символов, коды которых в строке постоянны, а в столбце возрастают на 1.

Т Е С Т 3

Выводит на экран дисплея символы с кодами 40-176.

Вид изображения предсталяет собой последовательность символов, коды которых возрастают на 1 в строке слева направо, а в столбце сверху вниз. После кода 176 формируется код 40.

ПРИМЕЧАНИЕ. Для визуальной проверки изображения, полученного на экране дисплея при выполнении тестов 1, 2 или 3, можно приостановить вывод символов на экран по команде <СУ>/<S>. После сравнения возобновить вывод по команде <СУ>/<Q>.

Т Е С Т 4. Тест восьмеричного эквивалента

Проверяет соответствие символа его восьмеричному коду. После ввода символа с терминала на экране должны высвечиваться символ и его восьмеричный код.

Т Е С Т Э Х О

Проверяет правильность ввода символа с терминала. После ввода символа с терминала на экран должен высвечиваться соответствующий символ.

5.2. Выполнение программы

Загрузка программы

Загрузить тест-программу "Тест дисплея символьного" по командам монитора:

```
L  ZDSAA0  <ВК>
S  <АДР>   <ВК>
     или
R  ZDSAA0  <ВК>
```

где <АДР> - пусковые адреса:

000200 - адрес начального запуска (выполняются тесты 1, 2, 3).

000214 - адрес запуска теста восьмеричного эквивалента (выполняется тест 4).

000220 - адрес запуска теста ЭХО (выполняется тест 5).

Выполнение программы

Для выполнения тестов 1, 2, 3 запустить тест-программу с адреса 000200 или по команде монитора S. В ответ на экран терминала выводится наименование тест-программы, а перед выполнением каждого теста - наименование теста.

Для выполнения теста восьмеричного эквивалента запустить тест-программу с адреса 000214. На экран терминала выводится наименование теста, и ожидается ввод символа с терминала. Для выполнения теста ЭХО запустить тест-программу с адреса 000200. На экран терминала выводится наименование теста, и ожидается ввод символа с терминала.

Время выполнения одного прохода тест-программы при запуске с адреса 000200 - 2 мин.

5.3. Команды оператора

Команда <СУ>/<S>

Одновременное нажатие служебной клавиши <СУ> и клавиши с буквой "S" запрещает вывод символов на экран дисплея при выполнении тестов 1, 2, 3.

Команда <СУ>/<Q>

Одновременное нажатие служебной клавиши <СУ> и клавиши с буквой "Q" возобновляет вывод символов на экран дисплея при выполнении тестов 1, 2, 3.

Для прекращения выполнения тест-программы отключите клавишу <ПУЛЬТ>.

На экран терминала выводится содержимое счетчика команд в момент останова.

Для продолжения выполнения тест-программы введите с терминала символ "P".

5.4. Сообщения оператору

Тест ДС

Начало выполнения тест-программы после запуска с адреса 000200.

<Заполнение экрана> - начало выполнения теста 1.

<Один символ на строке> - начало выполнения теста 2.

<Сдвиг символов> - начало выполнения теста 3.

Тест восьмеричного эквивалента

Нажмите клавишу,

должны высвечиваться символ и его восьмеричный код.

Начало выполнения теста 4:

Введите с терминала символ на русском или латинском регистре. На экране дисплея должно высветиться:

X = YYY,

где X - введенный с терминала символ;

YYY - восьмеричный код введенного с терминала символа.

Тест ЭХО

Начало выполнения теста 5:

Введите с терминала символ на русском или латинском регистре. На экране дисплея должен высветиться введенный с терминала символ.

ПРОХ = K

где K - номер прохода.

Конец выполнения прохода тест-программы при запуске с

адреса 000200.

6. ТЕСТ ДИСПЕТЧЕРА ПАМЯТИ

6.1. Назначение программы

Тест диспетчера памяти (ТДП) проверяет регистры диспетчера памяти, правильность формирования физического адреса из виртуального адреса с использованием соответствующих значений PAR (регистра адреса страницы) и PDR (регистра описания страницы). Затем следует проверка обработки прерываний и команд процессора MFPI и MTPI.

6.2. Загрузка и выполнение теста

Тест диспетчера памяти загружается и запускается командой монитора:

```
R CDMAA1  <ВК>
    или
L CDMAA1  <ВК>
S  <ВК>
```

6.3. Управление режимом работы

Режим выполнения программы определяется содержимым программного регистра переключателей РП (адрес 000176).

Назначение разрядов регистра переключателей:

РП15 = 1 – останов по ошибке;
РП14 = 1 – зацикливание текущего теста;
РП13 = 1 – запрещение вывода сообщений об ошибках;
РП12 = 1 – запрещение прерываний по слежению;
РП11 = 1 – запрещение итераций;
РП10 = 1 – вывод * при обнаружении ошибки;
РП9 = 1 – зацикливание по ошибке;
РП8 = 1 – зацикливание теста, номер которого набран на РП (7:0).

Если программа выполняется в командном режиме (с использованием цепочного файла) тест-мониторной системы, то во время выполнения программы нельзя изменять содержимое программного регистра переключателя.

Если тест диспетчера памяти выполняется не в командном режиме, то во время первого (после загрузки) прохода после вывода названия программы на экран терминала выдается сообщение: РП = XXXXXX НОВ =
где XXXXXX – текущее восьмеричное содержимое ячейки 176.
и программа ожидает ввода с терминала нового значения регистра переключателей.
Возможны следующие ответы:

YYYYYY <ВК> – YYYYYY – восьмеричное число, состоящее не более чем из 6 цифр, которые будут загружены как новое значение РП;
<ВК> – оставляет исходное значение РП;
<СУ>/<U> – спец.режим/U. Используется, когда новое значение РП было набрано с ошибкой (до нажатия клавиши <ВК>) для отмены введенного нового значения,

После отмены ввода нужно набрать новое значение.

<СУ>/<С> – спец.режим/С. Используется для предварительного вывода на экран терминала номера прохода и номера текущего теста, а также требования повторного ввода нового значения для РП. Программа перейдет к выполнению подпрограммы "Конец прохода", и следующий проход будет выполняться с новым значением РП.

В ответ на любой введенный оператором знак или цифру, отличающиеся от приведенных, после нажатия <ВК> будет выведено: ?

Программа будет реагировать на него, как на <СУ>/<U>, после чего необходимо ввести правильное значение РП.

Если программа выполняется не в режиме цепочки тест-мониторной системы, то при необходимости заменить содержимое программного РП во время выполнения программы оператор должен ввести с клавиатуры терминала <СУ>/<G>.

В ответ на нажатие <СУ>/<G> на экран терминала будет выведено требование нового значения РП, упомянутое выше.

6.4. Описание функций

6.4.1. Характеристика тестов программы

Первый проход программы выполняется без итераций. Если номер прохода нечетный и не меньше 3, то программа выполняется с прерываниями по Т-биту, которые можно запретить, установив РП12 в 1. В конце каждого прохода программы на терминал выдается номер выполненного прохода и общее количество ошибок (десятичные числа).

Тест диспетчера памяти состоит из 52-х (восьмеричное) тестов, каждый из которых выполняет отдельные функции.

Тесты 1-3 – проверка установки и сброса битов режима и приоритета PSW (слова состояния процессора), адресации байтов PSW.

Тест 4 – проверка установки указателей стека режимов "пользователя" и "системы".

Тест 5 – проверка возможности обращения к регистрам SR0, SR1 и SR2 (регистры состояния 0,1 и 2) диспетчера памяти.

Тесты 6,7 – проверка возможности обращения к регистрам PAR/PDR режима "система".

Тесты 10,11 – проверка возможности обращения к PAR/PDR режима "пользователя".

Тест 12 – проверка установки и сброса битов (15:13) SR0 и проверка того, что содержимое SR2 блокируется, когда любой из битов (15:13) SR0 установлен в 1.

Тест 13 – проверка правильности выборки адреса SR0.

Тест 14 – проверка того, что SR1 всегда отвечает нулями.

Тесты 15,16 – проверка vгтановки и сброса битов в "сис-

темных" и "пользовательских" PAR и PDR.

Тесты 17,20 – проверка правильности выборки адресов "системных" и "пользовательских" PAR и PDR.

Тест 21 – проверка правильности выборки адресов PAR и PDR.

Тест 22 – проверка того, что команда RESET не изменяет содержимое PAR/PDR.

Тесты 23,24 – проверка того, что адрес выбираемой команды и адрес источника не перемещаются в режиме диагностики.

Тесты 25,26 – проверка перемещения и логики сумматора.

Тест 27 – проверка чтения и записи в режиме перемещения.

Тесты 30-32 – проверка установки и сброса W-бита в PDR.

Тесты 33,34 – проверка защиты памяти при помощи ключей доступа.

Тест 35 – проверка возникновения прерывания при установке в PSW запрещенного режима 01.

Тесты 36,37 – проверка логики сравнения длины страниц и логики анализа ошибок диспетчера памяти.

Тесты 40,41 – проверка заполнения регистров диспетчера памяти SR0 и SR2 при возникновении прерывания.

Тест 42 – проверка того, что при работе в режиме "пользователя", вектор прерывания выбирается из области режима "системы".

Тест 43 – проверка того, что когда RTI выполняется в режиме "пользователя", биты режима и приоритета в PSW не изменяются.

Тест 44 – проверка того, что при обращении к нечетному адресу не обрабатывается ошибка диспетчера памяти.

Тест 45 – проверка того, что PC (счетчик команд) и PSW сохраняются в стеке для диспетчера памяти во время обработки ошибки нечетного адреса.

Тесты 46-52 – проверка выполнения команд MFPI, MTPI и MFPD, MTPD.

6.4.2. Сообщения оператору

Если РП13=0, то при обнаружении ошибки выводится информация об ошибке, состоящая из нескольких строк, содержащая сообщение об ошибке, заголовок выводимых данных, данные.

Каждое сообщение об ошибке обязательно выводит номер теста, в котором обнаружена ошибка и адрес вызова ошибки.

Пример сообщения об ошибке для случая, когда биты регистра диспетчера памяти установлены неверно:

```
Адрес   Запис.  Прочит. Прочит.-(двоич.) N теста  PC ошибки
регист. (восьм) (восьм)
177572  040000  060000  0110000000000000 000012   022060
```

Из сообщения следует, что ошибка обнаружена тестом 12 при проверке регистра SR0 (адрес 177572). При выполнении команды с адресом 22060 ошибка произошла при попытке записи в SR0 кода 040000.

Если РП15=1, процессор остановится после выдачи сообще-

ния об ошибке. Для продолжения выполнения программы необходимо нажать клавишу <Р> на клавиатуре терминала.

Если РП9=0, то после обработки ошибки программа перейдет к команде, следующей за вызовом ошибки.

Если РП9=1, то произойдет зацикливание части теста, обнаружившего ошибку.

Если произойдет непредусмотренное прерывание по ошибке канала (вектор 4), то на экран терминала будет выведео сообщение: НЕПРЕДВИДЕННОЕ ПРЕРЫВАНИЕ ПРОЦЕССОРА ПО 4 ВЕКТОРУ

Если произойдет непредусмотренное прерывание диспетчера памяти (вектор 250), то выводится сообщение:
НЕПРЕДВИДЕННОЕ ПРЕРЫВАНИЕ ДИСПЕТЧЕРА ПАМЯТИ ПО ВЕКТОРУ 250

Если же возникнет второе прерывание до того, как было выведено сообщение о первом прерывании (по 4 вектору или по вектору 250), произойдет останов.

7. ТЕСТ БЫСТРОДЕЙСТВИЯ

Для измерения быстродействия микро-ЭВМ необходимо после загрузки ТМОС выполнить следующие действия:

1) Ввести с клавиатуры терминала команду:

R ZDVAA0 <ВК>

В случае успешной загрузки и запуска программы на экран терминала выдается имя программы и сообщение:

ВКЛЮЧИТЕ ТАЙМЕР

2) Поднять клавишу "ТАЙМЕР" на передней панели блока питания. После включения таймера на экран терминала выводится сообщение:

```
ТЕСТ БЫСТРОДЕЙСТВИЯ
КОМАНДА СЛОЖЕНИЯ РЕГИСТР-РЕГИСТР .
БЫСТРОДЕЙСТВИЕ (ТЫС.ОП./СЕК) XXXXX
КОМАНДА СЛОЖЕНИЯ ПАМЯТЬ-РЕГИСТР
БЫСТРОДЕЙСТВИЕ (ТЫС.ОП./СЕК) XXXHXX
КОМАНДА УМНОЖЕНИЯ РЕГИСТР-РЕГИСТР      (ДЛЯ МИКРО-ЭВМ
БЫСТРОДЕЙСТВИЕ (ТЫС.ОП./СЕК) YYYYYY     "ЭЛЕКТРОНИКА
ВЫКЛЮЧИТЕ ТАЙМЕР, ЗАТЕМ НАЖМИТЕ <Р>       МС 1201.02")
```

где XXXXXX- значение быстродействия микро-ЭВМ на данном типе команд.

3) Опустить клавишу "ТАЙМЕР" на передней панели блока питания;

4) Ввести с клавиатуры терминала директиву "Р". На экран терминала выводится символ приглашения монитора ".".

ПРИМЕЧАНИЕ. Погрешность измерения быстродействия +3%.

8. ТЕСТ КМД

8.1. Назначение программы

Тестовая программа ZMYAA0 предназначена для проверки работоспособности контроллера КМД и накопителя НГМД.

Она позволяет контролировать правильность выполнения всех команд контроллера по всем диапазонам значений параметров команды и обеспечивает отработку следующих режимов:

Установка параметров эмер устройства, диапазон доро-

жек, сторона диска, размер сектора), выбор привода, форматирование дисков, выполнение функционального теста, запись на диск константы или псевдо-случайных чисел, чтение с диска, запись и чтение случайных секторов, последовательное чтение с перезаписью, последовательная запись секторов с чтением и сравнением.

Все тесты могут выполняться циклически.

Программа ZMYAA0 работает в диалоговом режиме. Она обеспечивает сбор статистики с выдачей ее, по желанию оператора, на экран терминала или на печатающее устройство.

8.2. Загрузка и выполнение программы

Для загрузки программы ZMYAA0 необходимо ввести команду монитора: R ZMYAA0

Для выхода из программы и возврата в систему набрать команду: 0 <ВК>

8.3. Способ задания режимов

Необходимый режим работы теста задается с помощью меню, высвечиваемого на экране терминала. В программе используются меню двух уровней.

Меню первого уровня высвечивается на экране после загрузки и начала выполнения программы TESTMY:

```
        Т Е С Т      К М Д    И    Н Г М Д
            ---> установка параметров
                 форматирование
                 функциональный тест
                 запись
                 чтение
                 выбор привода
```

В режиме "установка параметров" по умолчанию в программе приняты следующие значения параметров:

- номер устройства - 0;
- диапазон дорожек от 0 до 39;
- число сторон диска - 2;
- размер сектора - 512 байт.

После выбора режимов записи или чтения на экран терминала выводится меню второго уровня:

После выбора режима записи:

```
  ---> последовательная запись секторов
       запись случайных секторов
       последовательная запись секторов с чтением и
       сравнением
```

После выбора режима чтения:

```
  ---> последовательное чтение секторов
       чтение случайных секторов
       последовательное чтение с перезаписью
```

Выбор требуемого режима осуществляется с помощью клавиш <!> ("стрелка вниз") и <!> ("стрелка вверх") п утем подвода указателя меню ---> в нужную строку и нажатия клавиши <ВК>.

Зацикливание выбранного режима осуществляется путем подвода указателя в нужную строку меню, нажатия клавиши < --> > ("стрелка вправо") и <ВК>. При этом после нажатия клавиши < --> > на экране появится символ "+" . После этого выбранный тест будет выполняться циклически. Выход из цикли-

ческого режима осуществляется путем одновременного нажатия клавиш <СУ> и <С>.

8.4. Возврат из теста заданного режима

Для возврата в меню второго уровня после выполнения тестов режима чтения или записи, в меню 1-го уровня из меню 2-го уровня, или в тест-мониторную систему из меню 1-го уровня, необходимо последовательно нажать клавиши <М> и <ВК> т.е. дать команду: М <ВК>

Для возврата в систему из меню любого уровня необходимо последовательно нажать клавиши <О> и <ВК>. т.е. дать команду: О <ВК>

8.5. Отображение результатов выполнения теста

В результате тестирования накопителя и контроллера в режиме записи и чтения соответствующие тесты создают диагностическую карту (К) и таблицу сбоев (Т) (рис. 1.).

Оператор имеет возможность вывести на экран терминала или распечатать карту, таблицу сбоев и содержимое последней дорожки заданного диапазона дорожек.

Список команд, с помощью которых эти данные можно вывести на экран терминала после завершения работы соответствующего теста и вывода одного из двух сообщений:

<ОШИБОК НЕТ>
- при успешном завершении теста, или
<ТЕСТ ОБНАРУЖИЛ ОШИБКИ>
- при обнаружении сбоев,

приведен ниже:

К<ВК> - карта на экран
КП<ВК> - карта на печать
Т<ВК> - таблица на экран
ТП<ВК> - таблица на печать
Д<ВК> - содержимое дорожки на экран
ДП<ВК> - содержимое дорожки на печать
М<ВК> - возврат в меню

8.5.1. Диагностическая карта отображает состояние всех секторов диска после работы теста. Каждая позиция карты соответствует одному сектору.

На карте по горизонтали выведены номера дорожек, по вертикали - номера секторов, начиная с первого. Штриховая линия отделяет данные верхней и нижней сторон диска. Нижней стороне соответствует нижняя часть экрана, верхней стороне - верхняя часть экрана.

В каждой позиции карты записывается:

. - если к сектору обращений не было;
О - если сектор сбойный и оценивается как "плохой";
С - если сектор сбойный и оценивается как "удовлетворительный", (количество сбоев меньше половины количества обращений к сектору);
+ - если сектор оценивается как "хороший" (сбоев при обращении к сектору не было).

Из приведенной карты видно, что сектора 2-ой на дорожках с 30 по 39, 6-ой на дорожках с 9 по 13 и с 24 по 37 верхней стороны диска, с 31 по 39, 6-ой на дорожках с 18 по

21 и с 28 по 39 нижней стороны диска, определены как "плохие", а к секторам на дорожках с 40 по 79 обращений не было, остальные сектора диска определены как "хорошие".

8.5.2. Таблица сбоев дает более полную информацию о сбойных секторах по сравнению с диагностической картой. В таблице приводится число обращений к сектору, число сбоев при обращении к сектору, содержимое регистра состояния и ошибок, полученное при последнем сбойном обращении к сектору. Содержимое регистра состояния и ошибок дается в восьмеричном представлении, возможные значения этого регистра приведены в приложении 5. Для выдачи таблицы сбоев на экран необходимо дать команду Т <ВК>, для выдачи таблицы на печать - ТП <ВК>.

Пример таблицы сбоев приведен на рис. 1.

Дорож.	Сектор	Сторона	Рег.сост.ош.	Число обр.	Число сбоев
32	4	верх	2201	1	1
33	4	верх	2201	1	1

рис. 1

8.5.3. Вывод содержимого дорожки можно осуществлять после выполнения тестов: последовательное чтение секторов и последовательное чтение с перезаписью. При этом возможен просмотр содержимого последней прочитанной дорожки. Для приостановки вывода содержимого дорожки на экран надо одновременно нажать клавиши <СУ> и <S>, для продолжения вывода нажать одновременно клавиши <СУ> и <Q>, для прекращения вывода - клавиши <СУ> и <С>.

Для вывода содержимого дорожки на экран необходимо дать команду Д <ВК>, а для вывода на печать - ДП <ВК>.

8.6. Описание режимов работы теста

8.6.1. Выбор номера устройства

Режим предназначен для изменения номера устройства. После выбора этого режима на экране появится:

НОМЕР УСТРОЙСТВА N:

где N - номер устройства.

Введите номер устройства (0,1,2 или 3) и нажмите клавишу <ВК>.

После этого программа будет настроена на новый номер устройства и произойдет возврат в меню.

8.6.2. Установка параметров

Режим предназначен для установки параметров (номер устройства, диапазон дорожек, число сторон диска) работы теста. После выбора этого режима на экране последовательно будут появляться названия параметров и их последние установленные значения. Для изменения параметра необходимо задать новое значение и нажать клавишу <ВК>. Если изменять параметр не нужно, то нажмите клавишу <ВК>, не задавая нового значения.

Возврат в меню произойдет после установки всех параметров.

8.6.3. Форматирование

Режим обеспечивает форматирование дисков стандартной и нестандартной разметками.

Стандартная разметка определяется следующими параметрами:

диапазон дорожек от 0 до 39
число сторон диска - 2
размер сектора - 512 байт
код форматирования - 100

При изменении хотя бы одного параметра разметка считается нестандартной.

После выбора режима форматирования сначала экран очищается, а затем на нем появляется надпись:

```
ФОРМАТИРОВАНИЕ
УСТРОЙСТВО (0,1,2,3) 0:
```

Необходимо задать номер устройства (0,1,2,3), в котором находится диск, предназначенный для форматирования, и нажать клавишу <ВК>.

В ответ на экране появится следующая надпись:

```
РАЗМЕТКА СТАНДАРТНАЯ (ДОРОЖКИ:0-39;СЕКТОР:512;КОД:100) (Д,Н)
```

Если стандартная, нужно набрать Д <ВК>, в противном случае Н <ВК>.

При выборе нестандартной разметки оператор должен установить параметры разметки. На экране поочередно высвечиваются названия параметров и их значения.

```
ДОРОЖКИ ОТ 0:
        ДО 39:
СТОРОНА(0-НИЗ,1-ВЕРХ,2-ОБЕ)2:
СЕКТОР(1-256 БАЙТ,2-512, 3-1024) 2:
КОД 100:
```

Для задания новых значений необходимо набрать новое значение и нажать клавишу <ВК>, если значение изменять не нужно, нажать клавишу <ВК>.

При разметке диска на экран дважды выдаются номера дорожек и сторона записи и чтения. В случае обнаружения ошибок на экран выдается сообщение об ошибке и форматирование продолжается. После окончания форматирования на экран выводится сообщение:

```
ПОВТОРИТЬ ПРИ ТЕХ ЖЕ ПАРАМЕТРАХ (Д,Н,К)
```

В ответ на которое необходимо нажать требуемую клавишу и клавишу <ВК>.

Д - повторить разметку с теми же параметрами;
Н - повторить разметку с другими параметрами;
К - конец форматирования, возврат в меню.

8.6.4. Функциональный тест

В данном режиме обеспечивается проверка правильности выполнения команд и системы прерываний контроллера. Во время работы теста выполняемая команда контроллера отображается на экране. Если при выполнении команды контроллера была обнаружена ошибка, то для анализа ошибки на экран выдается содержимое регистра состояния и ошибок в восьмеричной системе счисления. Например:

Чтение. Ошибка. ESR=10600

Возможные значения регистра состояния и ошибок приведены в приложении 5.

Функциональный тест может выполняться циклически. Для выхода из цикла надо одновременно нажать клавиши <СУ> и <С>. По останову теста сообщается номер прохода.

8.6.5. В режиме записи может выполняться последовательная запись секторов, запись в случайно выбранные секторы, последовательная запись секторов с чтением и сравнением. Выбор конкретного режима осуществляется из меню второго уровня после выбора из основного меню режима записи.

8.6.5.1. В режиме последовательной записи секторов обеспечивается запись константы или последовательности псевдослучайных чисел по всем дорожкам заданного диапазона на заданных сторонах диска.

После выбора этого режима на экране появится сообщение:

КОНСТАНТА (ВК - П/С ЧИСЛА)

в ответ на которое оператор при выборе записи константы вводит константу в восьмеричном виде. Ввод заканчивается нажатием клавиши <ВК>. Для отработки записи псевдо-случайных чисел оператору достаточно нажать клавишу <ВК>.

Во время выполнения этого теста на экран выдается номер дорожки и сторона (верх, низ), на которую осуществляется запись. Последовательная запись секторов может выполняться циклически. Для выхода из цикла нужно одновременно нажать клавиши <СУ> и <С>.

После выполнения теста можно посмотреть диагностическую карту и таблицу сбоев.

8.6.5.2. Запись в случайно выбранные сектора

Этот режим аналогичен предыдущему (последовательной записи секторов). Он обеспечивает запись константы или последовательности псевдо-случайных чисел на псевдо-случайные сектора в заданном диапазоне дорожек и на заданные стороны диска.

Режим записи псевдо-случайных чисел всегда выполняется циклически: после записи в очередной выбранный сектор выбирается следующий сектор и т.д. для останова надо одновременно нажать клавиши <СУ> и <С>. Во время выполнения теста на экране высвечивается номер дорожки и сторона, в сектор которой производится запись.

После выполнения теста можно посмотреть диагностическую карту и таблицу сбоев.

8.6.5.3. Режим последовательной записи секторов с чтением и сравнением позволяет проверять правильность выполнения команд записи и чтения контроллером НГМД.

После выбора этого режима на экране появится сообщение:

КОНСТАНТА (ВК - П/С ЧИСЛА)

в ответ на которое оператор при выборе режима записи константы вводит константу в восьмеричном виде. Ввод константы заканчивается нажатием клавиши <ВК>. Для отработки записи псевдо-случайных чисел оператору достаточно нажать клавишу <ВК>.

Во время выполнения команд записи или чтения на экране терминала высвечиваются номера дорожек и сторона диска.

Тест-программа отработки данного режима выполняется следующим образом: сектор, номер которого задан в команде, сначала расписывается константой или псевдо-случайными числами, а затем содержимое этого сектора читается в память и прочитанная информация сравнивается с записываемой. В случае несовпадения номер сектора запоминается, а после просмотра всех секторов дорожки заданной стороны выдается сообщение в виде:

<НОМЕР ДОРОЖКИ> ВЕРХ/НИЗ ОШИБКИ В СЕКТОРАХ: I,J,..K

где I, J, K - номера секторов, в которых записываемые и считываемые данные не совпадают. Например:

15 низ ошибки в секторах: 1; 2; 10

В данном режиме последовательная запись секторов с чтением и сравнением может выполняться циклически. Для выхода из цикла достаточно одновременно нажать клавиши <СУ> и <С>.

8.6.6. В режиме чтения может выполняться последовательное чтение секторов, чтение с случайно выбранных секторов и последовательное чтение секторов с перезаписью. Выбор конкретного режима осуществляется из меню старого уровня после выбора из основного меню режима чтения.

8.6.6.1. Режим последовательного чтения секторов обеспечивает чтение секторов в заданном диапазоне дорожек и на заданных сторонах. Во время выполнения этого режима на экране терминала высвечивается номер дорожки и сторона диска, с которой идет считывание информации.

Режим последовательного чтения секторов может выполняться циклически. Для выхода из цикла необходимо одновременно нажать клавиши <СУ> и <С>. После останова теста на экране высвечивается номер последнего прохода теста.

По окончании выполнения программы данного режима можно посмотреть диагностическую карту, таблицу сбоев и содержимое последней прочитанной дорожки одной из сторон: нижней, если при установке параметров была задана нижняя сторона, или верхней.

8.6.6.2. Чтение с случайно выбранных секторов

Данный режим позволяет считывать в память данные с случайно выбранных секторов в заданном диапазоне дорожек на заданных сторонах.

Этот режим всегда выполняется циклически. Для выхода из цикла необходимо одновременно нажать клавиши <СУ> и <С>.

Во время выполнения программы на экране высвечивается номер дорожки и сторона, с которой выполняется режим чтения.

После отработки даного режима можно посмотреть диагностическую карту и таблицу сбоев.

8.6.6.3. Режим последовательного чтения с перезаписью позволяет читать и вновь писать один сектор или все сектора в заданном диапазоне дорожек на заданных сторонах диска.

После выбора этого режима на экран терминала выдается сообщение:

СЕКТОР (ВК - ПО ВСЕМ СЕКТОРАМ)

В ответ на которое оператор, нажав клавишу <ВК>, задает

режим чтения с перезаписью всех секторов в заданном диапазоне дорожек на заданных сторонах. Если необходимо прочитать с перезаписью один конкретный сектор, оператор вводит номер сектора в десятичном виде и нажимает клавишу <ВК>.

Режим последовательного чтения с перезаписью может выполняться циклически. Для выхода из цикла необходимо одновременно нажать клавиши <СУ> и <С>.

После отработки данного режима можно вывести на экран терминала диагностическую карту, таблицу сбоев или содержимое последней прочитанной и перезаписанной дорожки.

9. ТЕСТ КТЛК

9.1. Назначение теста КТЛК

Тестовые программы ZTKAA0, ZTKBA0, ZTKCA0 (далее по тексту: тест КТЛК) предназначены для проверки работоспособности платы КТЛК (на 4, 6 или 12 каналов). Последовательность проверки каналов определяется коммутатором ЩИМ.847.118, который перед проверкой вставляется в раз'ем КТЛК.

Тест КТЛК выполняет следующие работы:

- проверку начальной установки;
- поочередную проверку работы регистров состояния;
- поочередную проверку прерываний от передатчика каждого канала;
- проверку передачи и приема информации регистрами данных с прерыванием от приемника при заданной коммутации каналов.

Тест КТЛК обеспечивает проверку КТЛК в любом сочетании следующих режимов:

- печать диагностических сообщений по ходу выполнения проверок канала;
- останов после проверки одного канала;
- останов после проверки всего устройства;
- повторение проверки любого числа каналов до 177777 раз.

Тест КТЛК позволяет устанавливать параметры режимов проверки в любом сочетании до начала проверки КТЛК и изменять их во время выполнения теста с помощью команд, указанных в в п. 9.6.

Тест КТЛК выдает следующие сообщения оператору:

- о неправильных действиях оператора;
- о сбоях и неисправностях устройства;
- справочные данные.

По умолчанию приняты следующие режимы работы теста: оперативная печать сообщений результатов контроля, останов после проверки канала и после прохождения теста, возможность задания изменения адресов и векторов КТЛК.

В таблице 7 приведены распределения адресов регистров и векторов по каналам:

Таблица 7

Канал	Регистры передатчика RS	Регистры передатчика RD	Регистры приемника RS	Регистры приемника RD	Вектора прерываний передатчика	Вектора прерываний приемника
1	2				3	
TK0	176564	176566	176560	176562	324	320
TK1	176574	176576	176570	176572	334	330
TK2	176604	176606	176600	176602	344	340
TK3	176614	176616	176610	176612	354	350
TK4	176624	176626	176620	176622	364	360
TK5	176634	176636	176630	176632	374	370
TK6	176504	176506	176500	176502	244	240
TK7	176514	176516	176510	176512	254	250
TK8	176524	176526	176520	176522	264	260
TK9	176534	176536	176530	176532	274	270
TK10	176544	176546	176540	176542	304	300
TK11	176554	176556	176550	176552	314	310

9.2. Загрузка и начальный этап выполнения программы

Для загрузки и запуска программы тест КТЛК на клавиатуре терминала необходимо набрать одну из следующих команд монитора:

.R ZTKAA0<ВК> - для КТЛК6, имеющего адреса РГ 176500-176556,
.R ZTKBA0<ВК> - для КТЛК6, имеющего адреса РГ 176560-176636,
.R ZTKCA0<ВК> - для КТЛК12,имеющего адреса РГ 176500-176636.

Если КТЛК имеет адреса, отличные от указанных, то их необходимо ввести в командном режиме во время инициации теста после появления на экране терминала сообщения:

```
**************************************
*     "ТЕСТ ТЕЛЕГРАФНЫХ КАНАЛОВ"     *
************************************** ,
```

означающего начало работы программы, и запроса:

"КОНТРОЛЬ ? (Y,N) " ,

означающего выбор режима работы теста, в ответ на который необходимо ввести "Y" или "N".

При вводе "Y" будет проводиться контроль на работоспособность устройства. Тест повториться пять раз, и на экран терминала будут выданы сообщения о ходе тестирования и о результате проверки устройства:

```
.........................................
.          НАЧАЛЬНАЯ  УСТАНОВКА         .
.........................................
.........................................
.ПРОВЕРКА ВЕКТОРОВ ПРЕРЫВАНИЯ ПЕРЕДАТЧИКОВ .
.........................................
.........................................
.       ПЕРЕСЫЛКА КОДОВ С ПРЕРЫВАНИЕМ    .
.........................................
```

" СУММАРНОЕ ЧИСЛО СБОЕВ - <ЧИСЛО>"
" УСТРОЙСТВО ИСПРАВНО / УСТРОЙСТВО НЕИСПРАВНО"
" ПРОВЕРКА ОКОНЧЕНА"
" ПОВТОРИТЬ ПРОВЕРКУ КТЛК ? (Y,N)"

При ответе "N" будет выполняться диагностический контроль устройства в режиме принудительного диалога при заданных оператором условиях.

9.3. Изменение режима работы теста

В тесте КТЛК допускается изменение режима работы теста, принятого по умолчанию. Изменение режима выполняется в ответ на запрос, выданный ЭВМ.

Оператор должен ввести "Y", если соответствующий режим необходимо изменить и "N", если режим, установленный по умолчанию, необходимо оставить.

Указанные ниже режимы вводятся в ЭВМ в ответ на запрос:
ИЗМЕНИТЬ НАЗНАЧЕНИЯ РЕЖИМА РАБОТЫ ТЕСТА КТЛК (Y,N) ?
"Y" - если режимы, заданные по умолчанию сохраняются;
"N" - если требуется изменение хотя бы одного из режимов:
- печать сообщений (Y,N),
- останов после проверки канала (Y,N),
- останов после прохождения теста (Y,N),
- изменение адресов и векторов КТЛК (Y,N),
- необходимость задания признака повторений выполнения теста КТЛК (Y,N),
- необходимость установки нового числа каналов (Y,N).

При вводе признака изменения адресов и векторов прерываний КТЛК на экране терминала появится запрос:
ВВЕСТИ НОВЫЕ АДРЕСА РЕГИСТРОВ И ВЕКТОРОВ ? (Y,N),
в ответ на который необходимо набрать "Y", если необходимо ввести новые адреса, и "N", если адреса и вектора прерываний, заданные по умолчанию, остаются без изменений.

При вводе "Y" на экран терминала поочередно будут выведены имена каналов КТЛК в виде:
КАНАЛ! TKN! , (N=0,1,2,...11)

В ответ на каждое сообщение-запрос необходимо ввести, разделяя запятыми, адреса регистров канала состояния передатчика (RSI), данных передатчика (RDI), состояния приемника (RSP), данных приемника (RDP), адрес вектора прерываний передатчика (VI) и адрес вектора прерываний приемника (VP),или нажать клавишу <ВК>, если адреса регистров и векторов прерываний соответствующего канала не изменяются.

Установка нового числа каналов производится в ответ на сообщение-запрос:
ЧИСЛО КАНАЛОВ В УСТРОЙСТВЕ = 6, НАДО? <ЧИСЛО КАНАЛОВ>

9.4. Изменение порядка проверки каналов

В тесте КТЛК допускается изменение порядка проверки и коммутации каналов. По умолчанию принята следующая коммутация каналов:

Каналы:	Коммутация каналов
передающие	0,1,2,3,4,5,6,7,8,9,10,11,12,13,
принимающие	0,1,2,3,4,5,6,7,8,9,10,11,12,13,

Изменение порядка проверки каналов выполняется в ответ на сообщение-запрос от ЭВМ:

ОСТАВИТЬ ПОРЯДОК ПРОВЕРКИ ?

Необходимо ввести "Y" (оставить) или "N" (изменить).

При вводе "Y" необходимо сначала ввести число проверяемых каналов в ответ на сообщение:

ЧИСЛО ПРОВЕРЯЕМЫХ КАНАЛОВ = ? ,

а затем, разделяя запятой, номера передающих и принимающих каналов в ответ на запрос:

КАНАЛЫ	! КОММУТАЦИЯ КАНАЛОВ !
ПЕРЕДАЮЩИЕ	<НОМЕРА КАНАЛОВ>
КАНАЛЫ	
ПРИНИМАЮЩИЕ	<НОМЕРА КАНАЛОВ>

Ввод номера последнего передающего или принимающего канала заканчивается символом "," и нажатием клавиши <ВК>.

9.5. Порядок проверки каналов

Проверка каналов начинается после задания режимов работы теста и порядка проверки каналов.

Сначала осуществляется начальная установка всех переменных, таблиц сбоев, регистров состояния всех внешних устройств, а затем выполняется последовательная проверка каждого канала.

Каждый канал проверяется по следующему алгоритму:

- предварительное считывание содержимого регистра данных приемника;
- проверка нулевого и второго разрядов регистра передатчика;
- проверка разрядов готовности приемника и предатчика;
- проверка вектора прерываний передатчика;
- проверка блокировки передатчика;
- проверка шестого разряда (разрешение прерывания) регистра состояния приемника;
- дальнейшая проверка каналов по прерыванию.

Окончательная проверка работоспособности каждого канала осуществляется на восьмибитных кодах путем сравнения пересылаемого и принимаемого кодов.

При несовпадении кодов содержимое соответствующей строки таблицы состояния и ошибок канала увеличивается на единицу.

Если число ошибок по каждому каналу превышает минимально допустимое число сбоев, канал считается неисправным.

Работа на прерывание по каждому каналу проверяется при блокировке прерывания всех остальных каналов.

Во время проверки каналов на экран терминала могут выдаваться сообщения:

НАЧАЛЬНАЯ УСТАНОВКА
КАНАЛ <НОМЕР КАНАЛА>
ПРОВЕРКА ВЕКТОРОВ ПРЕРЫВАНИЙ ПЕРЕДАТЧИКА
ПО УМОЛЧАНИЮ ЧИСЛО ПОВТОРЕНИЙ ТЕСТА : 5
ПО ЗАДАНИЮ ЧИСЛО ПОВТОРЕНИЙ ТЕСТА ?
ПЕРЕСЫЛКА КОДОВ С ПРЕРЫВАНИЕМ

и.т.д.

9.6. Команды оператора

Заданный режим и порядок выполнения теста КТЛК оператор может изменить командами. Вход в командный режим происходит после нажатия клавиши <СУ> и клавиши <С>. Тест КТЛК отвечает запросом:

"КОМАНДА?"

Оператор должен ввести одну из команд: R, P, H, K, S, N

Ниже приводится краткое описание функций по каждой команде.

- "R" - выход на начало теста и выдача запроса:
  "КОНТРОЛЬ ? (Y,N)"

Далее проводится описанная выше работа.

- "P" - выход на запрос:
  "ИЗМЕНЕНИЕ ПОРЯДКА ПРОВЕРКИ (Y,N)"

Затем тест КТЛК выполнит обнуление служебных таблиц и проведет заново проверку КТЛК в соответствии с новым порядком проверки.

- "H" - по команде распечатывается таблица команд, используемых оператором. Выход в командный режим осуществляется нажатием клавиш: <СУ> и <С> . Второе нажатие клавиш <СУ> и <С> приводит к выходу программы на конец теста, при этом выдается сообщение:
  "ПОВТОРИТЬ ПРОВЕРКУ КТЛК ? (Y,N)"
- "K" - установка критерия определения нормальной задержки
- "S" - выдача справки о всех начальных назначениях в тесте КТЛК
- "N" - установка режима работы теста КТЛК, предназначенного для зацикливания операций при регулировке по частным проверкам, затем следует запрос:
  "КАНАЛ ?" ,

в ответ на который необходимо ввести:

< НОМЕР ПРОВЕРЯЕМОГО КАНАЛА >

В ответ на сообщение:

"C - ПЕРЕСЫЛКА КОДА"
"B - ПРЕРЫВАНИЕ ПО ВЕКТОРУ ПЕРЕДАТЧИКА"
"P - РАБОТА КАНАЛА"

следует ввести символ требуемого режима работы теста КТЛК.

Режимы зацикливания по команде "N" позволяют выполнять непрерывно одну из операций теста КТЛК как без ошибки, так и с ошибкой:

- "C" - циклическая пересылка кода на регистр данных передатчика с ЭВМ, далее - на регистр данных приемника с регистра данных передатчика, без ожидания готовности передатчика, причем код пересылки указывается по запросу:
  " КОД = "

Требуемый ответ: < ЛЮБОЕ ЧИСЛО > и < ВК >, при этом установится циклическое повторение передачи назначенного кода.

Выход из цикла по команде "T".

- "B" - циклическое задание условий для возникновения прерывания. После проведения прерывания

организуется возврат в программу независимо от результата обработки прерывания.

- "Р" - циклическая работа двух каналов по передаче кода с прерыванием от приемника.

9.7. Сообщения оператору

В процессе выполнения тест КТЛК выдает различные сообщения, характеризующие его работу.

Ниже приведен список таких сообщений.

Сообщения по ходу выполнения проверки:

"ПРОВЕРКА НАЧАЛЬНОЙ УСТАНОВКИ"
"ПРОВЕРКА ВЕКТОРОВ ПРЕРЫВАНИЙ ПЕРЕДАТЧИКА"
"ПРОВЕРКА ПЕРЕСЫЛКИ КОДОВ"
"ПРОВЕРЕН КАНАЛ НОМЕР <НОМЕР КАНАЛА>",
"НЕИСПРАВЕН КАНАЛ НОМЕР <НОМЕР КАНАЛА>",
"ТАБЛИЦА СБОЕВ"
"СУММАРНОЕ КОЛИЧЕСТВО СБОЕВ"
"ОШИБОЧНЫЙ ВЕКТОР ПРЕРЫВАНИЯ ПРИЕМНИКА <НОМЕР> ВМЕСТО <НОМЕР>"
"ОШИБОЧНЫЙ ВЕКТОР ПРЕРЫВАНИЯ ПЕРЕДАТЧИКА <НОМЕР> ВМЕСТО <НОМЕР>"
"СБОЙ ПО КРИТЕРИЮ ВЕКТОР ПРЕРЫВАНИЯ <НОМЕР> "
"СБОЙ - ПРЕРЫВАНИЕ"
"ВЕКТОР ПРЕРЫВАНИЯ <НОМЕР> "

Сообщения при ошибках оператора:

"ОШИБКА:ПРОВЕРЯЕМЫХ ТК БОЛЬШЕ ИМЕЮЩИХСЯ"
"ОШИБКА ОПЕРАТОРА ПРИ НАЗНАЧЕНИИ ЧИСЛА ТК"

Сообщения при неисправностях КТЛК:

"ОШ.ПЕРЕДАТЧИК - РАЗРЯД 6 РЕГИСТРА СОСТОЯНИЯ"
"ОШ.ПРИЕМНИК - РАЗРЯД 6 РЕГИСТРА СОСТОЯНИЯ"
"ОШ.СВЯЗИ - ОБРЫВ ЛИНИИ ПРЕРЫВАНИЯ"
"ОШ.СВЯЗИ - ОТКАЗ ЛИНИЙ СВЯЗИ"
"ОШ.ПРИЕМНИК - РАЗРЯД 7 РЕГИСТРА СОСТОЯНИЯ"
"ОШ.ПЕРЕДАТЧИК - РАЗРЯД 7 РЕГИСТРА СОСТОЯНИЯ"
"ОШ.ПРИЕМНИК - ГОТОВ ДО ОБРАЩЕНИЯ К ПЕРЕДАТЧИКУ"
"ОШ.ПЕРЕДАТЧИК - НЕПРАВИЛЬНЫЙ ВЕКТОР ПРЕРЫВАНИЯ"
"ОШ.ПРИЕМНИК - НЕПРАВИЛЬНЫЙ ВЕКТОР ПРЕРЫВАНИЯ"
"ОШ.ПРИЕМНИК - РАЗРЯД 0 РГ.СОСТ.- ОТСУТ.СТОП-БИТ"
"ОШ.ПРИЕМНИК - РАЗРЯД 12 РГ.СОСТ.-ПЕРЕПОЛНЕНИЕ"
"ОШ.ПРИЕМНИК - РАЗРЯД 15 РГ.СОСТ.-ОШИБКА ПАРИТЕТА"
"ОШ.ПЕРЕДАТЧИК - РАЗРЯД 0 РГ.СОСТ.-"РАЗРЫВ" ЛИНИИ"
"ОШ.ПЕРЕДАТЧИК - РАЗРЯД 2 РГ.СОСТ.РЕЖИМ ПРОВЕРКИ"
"ОШ.ПЕРЕДАТЧИК - НЕСРАВНЕНИЕ КОДОВ В ШЛЕЙФЕ"
"НЕИСПРАВНОСТЬ - СБОЯ ПРЕРЫВАНИЕ"
"ОШИБКА ОБРАЩЕНИЯ К КАНАЛУ"
"РЕЗЕРВНАЯ КОМАНДА"
"НАРУШЕНИЕ ПИТАНИЯ"

Сообщения по окончании проверки по тесту КТЛК:

"ПРОВЕРКА ТК ОКОНЧЕНА"
"СУММАРНАЯ ТАБЛИЦА СБОЕВ"
"СУММАРНОЕ КОЛИЧЕСТВО СБОЕВ"
"УСТРОЙСТВО ИСПРАВНО"
"УСТРОЙСТВО НЕИСПРАВНО "

По окончании проверки кроме таблиц распечатываются ошибки, обнаруженные в каждом канале и приведшие к неисправ-

ности канала.

10. ТЕСТ КГД

10.1. Назначение программы

Тестовая программа ZKGDA0 (далее по тексту: тест КГД) используется для проверки функционирования контроллера графического дисплея.

Программа ZKGDA0 обеспечивает выполнение следующих функций:

- проверку регистров платы КГД;
- запись и чтение регистров, запуск в цикле записи и чтения регистров платы КГД;
- проверку памяти устройства методами "бегущего ноля" и "бегущей единицы";
- проверку памяти адресным тестом;
- общую проверку устройства;
- заполнение и стирание графического экрана;
- переключение дисплея в символьняй и графический режимы.

Тест КГД выполняет останов при проверке графической памяти и при зацикливании записи или чтения регистров.

Кроме этого, тест КГД выводит на экран терминала информационные сообщения и сообщения о сбоях и ошибках, обнаруженных при проверке контроллера.

10.2. Загрузка и выполнение программы

Для загрузки программы и передачи на нее управления необходимо на клавиатуре терминала набрать команду монитора:

R ZKGDA0 <ВК>

после чего на экране появится сообщение:

**********ТЕСТ КГД**********

Список команд теста:

полная проверка устройства	<O>
проверка регистров	<R>
проверка памяти	<Q> останов по пробелу
адресный тест памяти	<A> останов по пробелу
очистка графического экрана	<S>
заполнение графического экрана	<W>
включить символьный режим	<B>
включить графический режим	<G>
возврат в меню	<M>
конец	<K>

В левой части списка приведено название команд, в правой части - односимвольные коды, которые необходимо ввести для инициации команды.

Вслед за приглашением к работе (символ "--->") на экран терминала выдается сообщение: КОМАНДА(<M>-МЕНЮ) --->

Для выполнения требуемой команды на клавиатуре терминала необходимо нажать соответствующую клавишу. После выполнения очередной команды программа выходит на точку, означающую приглашение к работе.

10.3. Команды и сообщения оператору

10.3.1. Проверочный режим

10.3.1.1. По команде проверки регистров ("R") выполняется проверка регистров платы КГД, состояния, адреса, данных.

При успешной проверке на экран терминала выдается сообщение: РЕГИСТРЫ ПРОВЕРЕНЫ **ОШИБОК НЕТ**

При обнаружении ошибок на экран терминала выдается таблица ошибок регистров:

Регистр	Разряды	Ош зап	Ош чт	Ош чмз	Нет уст	Нет сбр	Ош нет
состояния	14	***	***				
	15			***	***	***	+
данных	0–15	***	***	***			+
адреса	0–13				***	***	+

где *** – количество ошибок (от 0 до 100), обнаруженных при проверке;

\+ – сообщение об отсутствии ошибок при проверке указанного регистра (рст, рег дан, рег адр).

ош зап, ош чт, ош чмз – ошибка записи, чтения, чтения-модификации-записи;

нет уст – нет устройства;

нет сбр – нет сброса;

ош нет – нет ошибок.

10.3.1.2. Адресный тест памяти ("A").

Выполняется проверка памяти платы КГД адресным тестом (по адресу памяти записывается его графическое значение). В начале адресного теста проверяются регистры контроллера, при обнаружении ошибок в которых на экран терминала выдается сообщение:

ОШИБКИ ПРИ ПРОВЕРКЕ РЕГИСТРОВ; ПРОВЕРКА ПАМЯТИ НЕ ПРОИЗВОДИТСЯ

и программа выходит в режим ожидания следующей команды. При этом на экране терминала выдается приглашение к работе:

КОМАНДА(<M>–МЕНЮ)--->

При отсутствии регистровых ошибок сначала на экране терминала появляется сообщение: SWT АДРЕСНЫЙ ТЕСТ
а затем выполняется проверка графической памяти платы КГД адресным тестом: включается графический режим терминала, на экране видна мигающая горизонтальная полоса, движущаяся сверху вниз. При успешной проверке включается символьный режим и на экран терминала выдается сообщение:

ОШИБОК НЕТ

При обнаружении ошибок по каждой ошибке выдается сообщение:

БАНК ББ
ПО АДРЕСУ АААААА ЗАПИСАНО XXXXXXXX ПРОЧИТАНО YYYYYYYY

где ББ – номер проверяемой страницы памяти (размер страницы 400 ячеек);

АААААА – адрес, при обращении к которому обнаружена

ошибка;
XXXXXXXX - записываемое двоичное число;
YYYYYYYY - прочитанное двоичное число.

При работе адресного теста допускается останов процессора. Для этого на клавиатуре терминала необходимо нажать клавишу <ПРОБЕЛ>. При этом на экран терминала выдается сообщение: ОСТАНОВ ПРОВЕРКИ ПО ПРОБЕЛУ

10.3.1.3. Проверка памяти ("Q").

Производится комплексная проверка графической памяти методами "бегущей единицы", "бегущего ноля" и адресным тестом.

В начале теста памяти проверяются регистры КГД. При обнаружении ошибок на экране появляется сообщение, аналогичное адресному тесту, и программа переходит в режим ожидания команды с клавиатуры терминала.

При успешной проверке регистров выполняется проверка памяти КГД. При этом на экране сначала появляется сообщение:

ПРОВЕРКА ПАМЯТИ (БЕГУЩАЯ "1")

Затем включается графический режим и производится проверка памяти методом "бегущей единицы" (на экране видны "бегущие точки"). При отсутствии ошибок выдается сообщение:

ОШИБОК НЕТ

ПРОВЕРКА ПАМЯТИ (БЕГУЩИЙ "0")

Далее, аналогично тесту "бегущая единица" происходит проверка памяти методом "бегущего ноля".

В случае обнаружения ошибок при проверке "бегущей единицей" или "бегущим нолем" по каждой ошибке выдается сообщение:

ПО АДРЕСУ AAAAAA ЗАПИСАНО XXXXXXXX ПРОЧИТАНО YYYYYYYY

Вслед за тестами "бегущая 1" и "бегущий 0" выполняется проверка графической памяти адресным тестом.

По окончании проверки адресным тестом программа выходит в режим ожидания команды.

10.3.1.4. Полная (общая) проверка устройства ("O") выполняется в последовательности, указанной в п.п.10.3.1.5.

При этом на экране терминала выдаются сообщения о ходе выполнения программы (см.табл.8). При обнаружении ошибок после выдачи сообщения, дальнейшая проверка КГД прекращается, и программа выходит в режим ожидания ввода.
Затем происходит проверка регистров КГД. При обнаружении ошибок на экран терминала выдается таблица ошибок регистров и сообщение:

ОШИБКИ ПРИ ПРОВЕРКЕ РЕГИСТРОВ;ДАЛЬНЕЙШАЯ ПРОВЕРКА НЕ ПРОИЗВОДИТСЯ

Программа после этого переходит в режим ожидания следующей команды.

В случае успешной проверки регистров на экран терминала выдается сообщение:

РЕГИСТРЫ ПРОВЕРЕНЫ **ОШИБОК НЕТ**

и затем выполняется проверка памяти КГД методами "бегущей 1", "бегущего 0" и адресным тестом аналогично предыдущему тесту.

По окончании проверки памяти включается графический режим и экран терминала заполняется точками (темный фон сменя-

ется светлым снизу вверх). Затем включается символьный режим для выдачи сообщений:

ПЕРЕКЛЮЧЕНИЕ РЕГИСТРА СОСТОЯНИЯ:
ГРАФИЧЕСКИЙ РЕЖИМ

После этого включается графический режим монитора. Весь экран должен быть заполнен точками (светлый фон).

Вслед за сообщением:

СИМВОЛЬНО-ГРАФИЧЕСКИЙ РЕЖИМ

на заполненном экране (светлый фон) должны просматриваться символы предыдущих сообщений. При появлении на экране терминала сообщения:

НА ЭКРАНЕ ТЕРМИНАЛА ДОЛЖНЫ БЫТЬ ВИДНЫ ОДНИ СИМВОЛЫ - ПРЕДЫДУЩИЕ СООБЩЕНИЯ ДАННОГО ТЕСТА.

При появлении сообщения:

ОЧИСТКА ГРАФИЧЕСКОГО ЭКРАНА

включается графический режим, на экране происходит постепенная замена светлого фона на темный (снизу вверх). Если при этом была обнаружена ошибка, то выдается сообщение:

ОШИБКА ПРИ СТИРАНИИ ЭКРАНА

и очистка графического экрана прекращается. Затем выдается сообщение:

ЗАПОЛНЕНИЕ ГРАФИЧЕСКОГО ЭКРАНА

Включается графический режим, на экране происходит постепенная замена темного фона на светлый (снизу вверх). При обнаружении ошибки на экран выдается сообщение:

ОШИБКА ПРИ ЗАПОЛНЕНИИ ЭКРАНА

и заполнение экрана прекращается.

10.3.1.5. Сообщения оператору и действия программы приведены в таблице 8.

Таблица 8

Вид сообщения. Общая проверка	Действие программы ZKGDA0 Начало выполнения команды "Общая проверка"
1	2
Ошибки при проверке регистров: дальнейшая проверка регистров	ожидание следующей команды оператора
Регистры проверены: ** ошибок нет **	дальнейшая проверка КГД тестами: "бегущая 1", "бегущий 0", адресным тестом
Переключение регистра состояния: графический режим	включение графического режима и заполнение экрана точками (светлый тон)
Символьно-графический режим	включение символьно-графического режима. На экране терминала появляются символы - предыдущие сообщения

1	2
Символьный режим	отключение графического режима. Отображение на экране только символов
Ошибка при стирании экрана	прекращение очистки графического экрана
Заполнение графического экрана	включение графического режима. Постепенная замена темного фона на светлый
Ошибка при заполнении экрана	прекращение заполнения экрана
Проверка окончена ** ошибок нет **	окончание проверки графического ОЗУ. Переход в режим ожидания ввода следующей команды
При проверке обнаружены ошибки	прекращение работы. Ожидание ввода следующей команды

10.4. Пультовой режим

В тесте предусматривается специальный пультовой режим. Этот режим дает возможность чтения регистров и запись в них чисел. Чтение регистров выполняется в следующей последовательности:

Сначала нажимается клавиша </>, затем одна из клавиш:

<S> - регистр состояния;
<A> - регистр адреса;
<D> - регистр данных.

При этом на экран терминала выдаются соответственно сообщения:

RST ******
RAD ******

где ****** - содержимое соответственно:

регистра состояния;
регистра адреса;
регистра данных.

Запись в регистры КГД осуществляется следующим образом: набирается вводимое восьмеричное число, затем "S", "A" или "D" (при наборе любого другого символа программа выходит в режим ожидания команды). При нажатии клавиши <S> число записывается в регистр состояния, <A> - в регистр адреса, <D> - в регистр данных.

Для включения режима записи или чтения в цикле необходимо нажать клавишу <Z>, а затем проделать действия, описанные выше. Если при первой попытке записи или чтения произошел сбой, на экран выдается сообщение:

ЗАВИСАНИЕ

Для останова процесса циклической записи или чтения необходимо нажать любую клавишу. Программа выйдет в режим ожидания следующей команды.

ПРИЛОЖЕНИЕ 1

НОСИТЕЛИ ТЕСТ-МОНИТОРНОЙ СИСТЕМЫ

У1.00032-01 Мд 01 – для накопителя типа "ЭЛЕКТРОНИКА НГМД-6021";

У1.00032-01 Мд 02 – для накопителя типа "ЭЛЕКТРОНИКА НГМД-6121";

У1.00032-01 Мд 03 – для накопителей типа "СМ-5640", "ЕС-5088".

ПРИЛОЖЕНИЕ 2

БАЗОВАЯ СИСТЕМА КОМАНД

Таблица

Команда		Признак				Наименование команды
мнем.	код	N	Z	V	C	
1		2				3
HALT	000000					останов
WALT	000001					ожидание
RTI	000002					возврат из прерывания
BPT	000003					прерывание для отладки
IOT	000004					прерывание для ввода-вывода
RESET	000005					сброс
RTT	000006					возврат из прерывания
JMP	0001DD					безусловная передача
RTS	00020R					возврат из подпрограммы
JSR	004RDD					обращение к подпрограмме
EMT	104000 – 104377					командное прерывание
TRAP	104400 – 104777					командное прерывание
NOP	000240					нет операции
CLC	000241	–	–	–	0	очистка C

1	2	3
CLV !000242	!- !-!0! -	!очистка V
CLZ !000244	!- !0!-! -	!очистка Z
CLN !000250	!0 !-!-! -	!очистка N
SEC !000261	!- !-!-! 1	!установка C
SEV !000262	!- !-!1! -	!установка V
SEZ !000264	!- !1!-! -	!установка Z
SEN !000270	!1 !-!-! -	!установка N
SCC !000277	!1 !1!1! 1	!установка N, Z, V, C
CCC !000257	!0 !0!0! 0	!очистка N, Z, V, C
SWAB !0003DD	!+ !+!0! 0	!перестановка байтов
CLR(B) !*050DD	!0 !1!0! 0	!очистка
COM(B) !*051DD	!+ !+!0! 1	!инвертирование
INC(B) !*052DD	!+ !+!+! -	!прибавление единицы
DEC(B) !*053DD	!+ !+!+! -	!вычитание единицы
NEG(B) !*054DD	!+ !+!+! +	!инвертирование и увеличение на единицу
ADC(B) !*055DD	!+ !+!+! +	!прибавление переноса
SBC(B) !*056DD	!+ !+!+! +	!вычитание переноса
TST(B) !*057DD	!+ !+!0! 0	!проверка
ROR(B) !*060DD	!+ !+!+! +	!циклический сдвиг вправо
ROL(B) !*061DD	!+ !+!+! +	!циклический сдвиг влево
ASR(B) !*062DD	!+ !+!+! +	!арифметический сдвиг вправо
ASL(B) !*063DD	!+ !+!+! +	!арифметический сдвиг влево
MARK !0064NN	!- !-!-! -	!восстановление SP
SXT !0067DD	!- !+!0! -	!расширение знака
MTPS !1064SS	!+ !+!+! +	!запись ССП
MFPS !1067DD	!+ !+!0! -	!чтение ССП

1	2	3
MOV(B) *1SSDD	+!+!0!	- пересылка
CMP(B) *2SSDD	+!+!+!	+ сравнение
BIT(B) *3SSDD	+!+!0!	- проверка разрядов
BIC(B) *4SSDD	+!+!0!	- очистка разрядов
BIS 05SSDD	+!+!0!	- логическое "ИЛИ"
XOR 074RDD	+!+!0!	- исключающее "ИЛИ"
ADD 06SSDD	+!+!+!	+ сложение
SUB 16SSDD	+!+!+!	+ вычитание
BR 0004XXX		ветвление безусловное
BNE 0010XXX	Z=0	ветвление, если не равно нулю
BEQ 0014XXX	Z=1	ветвление, если равно нулю
BGE 0020XXX	N&V=0	ветвление, если > или = 0
BLT 0024XXX	N&V-1	ветвление, если < 0
BGT 0030XXX	ZV(N&V)=0	ветвление, если > 0
BLE 0034XXX	ZV(N&V)=1	ветвление, если < или = 0
SOB 077ROO	Z=0	вычитание единицы и ветвление
BPL 1000XXX	N=0	ветвление, если плюс
BMI 1004XXX	N=1	ветвление, если минус
BHI 1010XXX	ZVC=0	ветвление, если больше
BLOS 1014XXX	ZVC=1	ветвление, если меньше
BVC 1020XXX	V=0	ветвление, если нет переполнения
BVC 1024XXX	V=1	ветвление, если переполнение
BHIS 1030XXX	C=0	ветвление, если нет переноса
BLO 1034XXX	C=1	ветвление, если перенос

ПРИМЕЧАНИЕ.

* - имеет значение :
0 - для команд с полными словами
1 - для байтовых команд
+ - признак изменяется по результату АЛУ-операций
0 - признак очищается
1 - признак устанавливается

ПРИЛОЖЕНИЕ 3

РАСШИРЕНИЕ СИСТЕМЫ ДЛЯ МИКРОПРОЦЕССОРА К1801ВМ2

Таблица

Команда		Признак	Наименование команды
мнем.	код	N Z V C	
1		2	3
MUL	070RSS	+ + 0 +	умножение
DIV	071RSS	+ + + +	деление
ASH	072RSS	+ + + +	арифметический сдвиг
ASHC	073RSS	+ + + +	арифметический сдвиг двойного слова
FADD	07500R	+ + 0 0	сложение с плавающей запятой
FSUB	07501R	+ + 0 0	вычитание с плавающей запятой
FMUL	07502R	+ + 0 0	умножение с плавающей запятой
FDIV	07503R	+ + 0 0	деление с плавающей запятой

ПРИМЕЧАНИЕ.

+ - признак изменяется по результату АЛУ-операций
0 - признак очищается

ПРИЛОЖЕНИЕ 4

РАСШИРЕНИЕ СИСТЕМЫ ДЛЯ МИКРОПРОЦЕССОРА К1801ВМ3

Таблица

Команда: мнемоника	Команда: код	Наименование команды
1		2
MUL	070RSS	умножение
DIV	071RSS	деление
ASH	072RSS	сдвиг на "N" разрядов одного слова
ASHC	073RSS	сдвиг на "N" разрядов двойного слова
MFPD MFPI	1065SS 0065SS	засылка слова (D – данные, I – инструкция) в стек текущей моды по адресу предварительной моды
MTPD MTPI	1066 0066	засылка слова из стека текущей моды по адресу предварительной моды

ПРИЛОЖЕНИЕ 5

ВОЗМОЖНЫЕ ЗНАЧЕНИЯ КОДА ОШИБОК

Значение кода ошибки выдается на регистр данных в ответ на команду КМД "Чтение регистра состояния и ошибок".

ЗНАЧЕНИЯ КОДА ОШИБОК

Таблица

Восьмерич. значение кода ошибок	Вид ошибки
1	2
1	ошибка контрольной суммы зоны данных или ошибка при попытке выполнить запись на диск, если установлен запрет записи
2	ошибка контрольной суммы зоны заголовка
10	нет установки на нулевую дорожку
20	ошибка поиска дорожки
40	сектор не найден за 5 оборотов диска
4000	запись или чтение по несуществующему

1	2
	адресу ЦП
10000	не найден адресный маркер
20000	не найден маркер данных
40000	неверно задан формат
100000	ошибка контроллера
100001	ошибка ОЗУ контроллера
100002	ошибка ПЗУ контроллера
100003	ошибка ПДП
100004	зависание на магистрали контроллера
100005	ошибка ПРЦ
100006	ошибка внутреннего регистра 177100
100007	ошибка внутреннего регистра 177102
100010	ошибка внутреннего регистра 177130
100011	ошибка внутреннего регистра 177132
100012	сбой питания

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

ВК - возврат каретки;
КМД - контроллер гибкого магнитного диска с удвоенной плотностью записи;
НГМД - накопитель на гибком магнитном диске;
ПДП - прямой доступ в память;
ПРЦ - процессор КМД;
ЦП - центральный процессор;
КГД - контроллер графического дисплея;
ОЗУ - оперативное запоминающее устройство;
ПЭВМ - персональная ЭВМ;
НПТ - нет питания.

ПОДСИСТЕМА МУЛЬТИПРОГРАММНОЙ ПРОВЕРКИ. РУКОВОДСТВО ОПЕРАТОРА.

1. НАЗНАЧЕНИЕ И УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММ

Подсистема мультипрограммной проверки предназначена для проверки системного взаимодействия устройств персональной микро-ЭВМ (ПЭВМ). Проверка ПЭВМ осуществляется в мультипрограммном режиме при максимальной загрузке системной магистрали. Тестовые модули выполняются асинхронно с перемещением теста по оперативной памяти и циклическим перемещением буферов записи.

1.1. Понятие комплексного теста

Мультипрограммный монитор компонуется программой конфигуратор/компоновщик (DXCL) с определенным количеством тестовых модулей в файл абсолютного формата, называемый тестом комплекса. Скомпонованный тест комплекса представляет собой программный модуль, размещаемый при выполнении в оперативной памяти. Максимальное количество компонуемых тестовых модулей ограничивается размерами оперативной памяти. Для проверки с помощью большого количества тестовых модулей рекомендуется создать несколько тестов комплекса.

1.2. Назначение составных частей подсистемы мультипрограммной проверки

Мультипрограммный монитор предназначен для обслуживания выполнения теста комплекса. Монитор осуществляет запуск и снятие отдельных тестовых модулей, перемещение теста по памяти и циклическое перемещение буферов записи.

Основным назначением программы конфигуратор/компоновщик (в дальнейшем программа DXCL) является генерация теста комплекса и его настройка на конфигурацию оборудования. Программа DXCL компонует в единый программный модуль модули мультипрограммного монитора и тестовые модули.

Модули делятся на тестовые и мониторные. Структура тестовых модулей описана в документе "Программирование тестовых модулей. Руководство программиста. Тестовые модули на носителе записаны в об'ектном коде.

Мониторные модули представляют собой подпрограммы мультипрограммных мониторов в об'ектном коде, об'единенные в библиотеки.

Все модули в об'ектном коде получены путем трансляции исходного текста модуля в операционной системе ОСДВК.

1.3. Структура и назначение тестовых модулей

Тестовые модули предназначены для проверки оборудования. При создании теста комплекса программой DXCL они компонуются с мультипрограммным монитором. Между тестовыми модулями и мультипрограммным монитором существует двусторонняя связь. Тестовый модуль обращается к мультипрограммному монитору через макровызовы монитора, кодированные командой TRAP. Информацию о буферных областях, общих рабочих ячейках, флагах, стартовых адресах тестовых модулей и так далее мультипрограммный монитор получает из интерфейса модуля. В тесте комп-

лекса тестовые модули связаны с мультипрограммным монитором и взаимно независимы.

1.4. Условия применения подсистемы мультипрограммной проверки

Проверка системного взаимодействия устройств, входящих в состав ПЭВМ, осуществляется после автономной проверки отдельных устройств, если все устройства признаны годными.

Все обслуживающие средства подсистемы мультипрограммной проверки предназначены только для работы с тестовыми модулями и не могут быть использованы в других операционных системах.

Программа DXCL и тест комплекса загружаются в оперативную память с носителя TMOC и запускаются согласно п.3.4.

1.5. Требуемое оборудование

Подсистема мультипрограммной проверки для своей работы требует следующее минимальное оборудование:

- Процессор К1801ВМ1 или программно совместимые с ним процессоры К1801ВМ2, К1801ВМ3;
- оперативную память емкостью не менее 16 кслов;
- устройство внешней памяти с носителем TMOC;
- алфавитно-цифровой терминал;

В качестве носителей используются гибкие магнитные диски, магнитная лента.

2. ОПИСАНИЕ СОСТАВНЫХ ЧАСТЕЙ

2.1. Функции подпрограмм теста комплекса

В функции мультипрограммного монитора входит обслуживание теста комплекса при его выполнении, которое заключается в следующем:

- запуск отдельных тестовых модулей в тесте комплекса;
- обслуживание запросов теста комплекса на выполнение служебных функций (печать сообщений тестового модуля, выдача сообщений об ошибках и т.д.);
- снятие с выполнения отдельных тестовых модулей;
- перемещение буфера записи тестовых модулей по оперативной памяти.

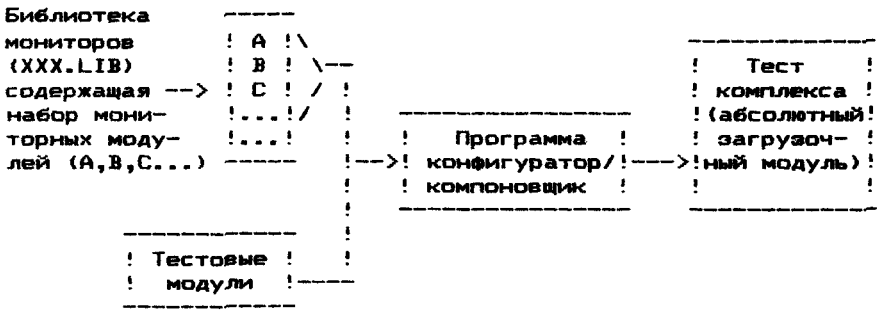
В функции программы DXCL входит компоновка теста комплекса и настройка его на конкретную конфигурацию оборудования.

В функции тестовых модулей входит проверка работоспособности отдельных устройств, входящих в состав ПЭВМ.

2.2. Состав подсистемы мультипрограммной проверки

Подсистема мультипрограммной проверки используется для создания независимых тестов комплекса, включающих модуль монитора и модули тестов устройств, выбранных пользователем. В состав ПЭВМ может входить несколько тестов комплекса.

На рис.1 приведена структура подсистемы мультипрограммной проверки и конструкция типового теста комплекса.



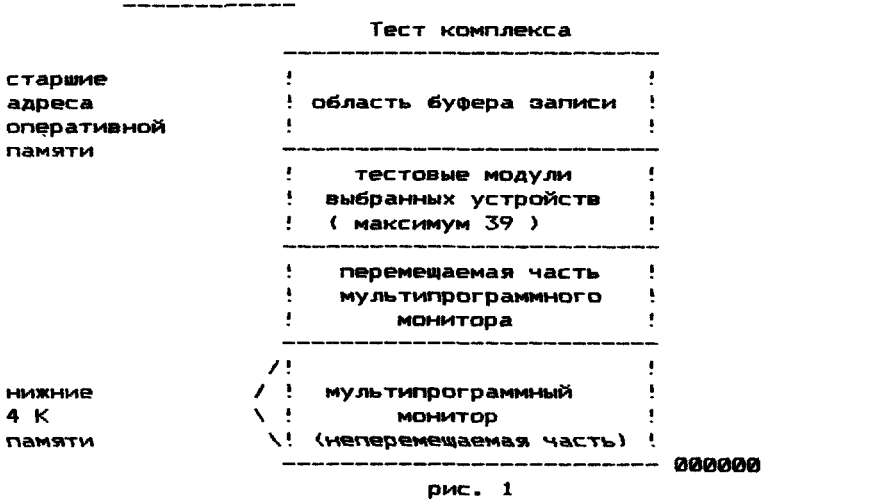


рис. 1

Подсистема мультипрограммной проверки содержит три основные части:

1) Библиотеку модулей мультипрограммных мониторов;
2) Набор тестовых модулей (программ в об'ектном коде);
3) Программу DXCL.

Пользователь выбирает необходимый монитор, тестовые модули и с помощью программы конфигуратор/компоновщик компонует требуемый тест комплекса.

Тест комплекса после компоновки имеет абсолютный загрузочный формат и может быть самостоятельно загружен абсолютным загрузчиком или любым монитором системы TMOC.

После загрузки теста комплекса неперемещаемая его часть будет всегда находиться в нижних 4K словах памяти (с адреса 000000).

Область за неперемещаемой частью мультипрограммного монитора содержит максимум 39 тестовых модулей и перемещаемую часть мультипрограммного монитора. Оставшаяся свободная область памяти используется для буфера записи, перемещения в ней тестовых модулей и перемещаемой части мультипрограммного монитора.

2.3. Тестовый модуль устройства

Каждый тестовый модуль устройства - это программа, которая предназначена для проверки отдельного внешнего устройства, устройства управления, процессора или дополнительного оборудования. Эта программа взаимосвязана с мультипрограммным монитором. Мультипрограммный монитор обеспечивает различное обслуживание модуля, выполнение процедур, которые могут потребоваться любому тестовому модулю.

Для связи с мультипрограммным монитором тестовому модулю необходимы определенные программные связки, которые содержатся в интерфейсе модуля и в каждом модуле. Тестовые модули разделены на семь основных типов, в зависимости от типа проверяемой функции или устройства. Каждый тип модуля требует соответствующие средства связи, которые определяются интерфейсом (заголовком) модуля.

ПРИМЕЧАНИЕ. При написании тестового модуля программист должен определить все глобальные и внутренние метки и значения литералов. После этого программист должен определить тип модуля, который нужно написать. Выбрав тип тестового модуля, программист указывает соответствующий интерфейс.

2.4. Классификация тестовых модулей

2.4.1. Фоновый модуль (BKMOD)

Модули BKMOD выполняются в так называемом фоновом режиме. Все модули этого типа выполняются с установленным битом "T" (бит прерывания по слежению, т.е. четвертый разряд слова состояния процессора установлен в "1"). В этом случае прерывания возникают после выполнения каждой команды модуля. Кроме обработки прерывания по биту "T", это позволит монитору управлять рядом инструкций, выполняемых фоновым модулем с более высоким приоритетом. При написании фонового модуля метки "START" и "RESTART" должны быть размещены по одному и тому же адресу. Фоновые модули используются в тех случаях, когда проверяются устройства или функции, не вызывающие прерываний. Например, модуль, написанный для процессора, проверяет все основные команды микро-ЭВМ.

2.4.2. Специальный фоновый модуль (SBKMOD)

Модули SBKMOD работают только в фоновом режиме и являются первой группой модулей, которые запускаются перед неперезапускаемыми фоновыми модулями. Модуль этого типа запускается только один раз после каждого перемещения теста комплекса в памяти.

Примером применения специального фонового модуля может служить мультипроцессорная система с переключателем шины, где выполнение функций модуля требуется после каждого перемещения теста комплекса в оперативной памяти.

2.4.3. Одиночный фоновый модуль (NBKMOD)

Модули NBKMOD выполняются в фоновом режиме. Модули этого типа запускаются за специальными фоновыми модулями только один раз. После успешного завершения модуля он никогда не запускается вновь. Если тест комплекса был снят с выполнения и запущен вновь, модули вновь выполняются только один раз.

Запускаются эти модули за специальными фоновыми модулями.

2.4.4. Модуль ввода-вывода (IOMOD)

Модули IOMOD функционируют в режиме ввода-вывода. Модули управляются по прерыванию и способны выполнить операции ввода-вывода. Тестовые модули этого типа связаны с устройствами, которые не осуществляют внепроцессорных передач и не имеют регистров счета слов. Ввод-вывод для таких устройств выполняется программным путем через регистры данных. К таким устройствам относятся, например, гибкий диск, печатающее устройство. Модуль ввода-вывода, запустив операцию, ожидает прерывания от внешнего устройства, чтобы продолжить выполнение. Если прерывание потеряно и не приходит, то модуль "зависает" до тех пор, пока не устанавливается специальный код для обнаружения этого факта. Эти модули перемещаются по оперативной памяти без ограничения.

2.4.5. Расширенный модуль ввода-вывода (IOMODX)

Модули IOMODX используются для проверки устройств с внепроцессорными передачами данных и обеспечивают дополнительные возможности, не требуемые для простых модулей ввода-вывода. Вот некоторые из этих дополнительных возможностей:

- использование буфера записи;
- возможность изменения размера буферов чтения и записи;
- обращение к монитору для проверки передаваемых данных;
- перевод виртуального 16-битного адреса в 22-битный.

Примером модуля этого типа является модуль проверки накопителей на магнитных дисках.

Подробную информацию об интерфейсе расширенного модуля ввода-вывода описывает [1].

2.4.6. Частично перемещаемый расширенный модуль ввода-вывода (IOMODP)

Модуль IOMODP из-за аппаратных ограничений может выполняться при перемещении только на определенные фиксированные границы оперативной памяти, например, кратные 32 ксловам.

Подробную информацию об интерфейсе частично перемещаемого расширенного модуля ввода-вывода описывает [1].

2.4.7. Ограниченный модуль ввода-вывода (IOMODR)

Модули IOMODR вообще не могут выполняться после перемещения из-за ограничения на оборудование. Такой тип модуля выполняется, если тест комплекса находится в нижнем банке памяти (размещен с адреса 0). Примером может быть модуль, проверяющий тестер общей шины.

Подробную информацию об интерфейсе ограниченного модуля ввода-вывода описывает [1].

2.5. Мультипрограммные мониторы

Поскольку имеется возможность сгенерировать несколько различных программ мультипрограммного монитора, пользователь выбирает подходящий мультипрограммный монитор в зависимости от конфигурации оборудования, подлежащего проверке. Необходимый мультипрограммный монитор может быть создан в процессе генерации из мониторных модулей, находящихся в библиотеке.

В зависимости от требуемых характеристик предусмотрена классификация программ монитора по типу и именам (т.е. A, C и E). Характеристики мультипрограммных мониторов приводятся в приложении 1.

Такая классификация мультипрограммных мониторов не только позволяет использовать подходящие программы монитора для данной конфигурации оборудования, но и выбрать мультипрограммный монитор, наиболее удовлетворяющий требованиям оборудования вычислительного комплекса в соответствии с функциональными потребностями получаемого теста комплекса и более эффективным использованием области памяти мультипрограммного монитора.

При этом способе генерации базовые программные компоненты теста комплекса (мультипрограммный монитор и выбранные модули) зависят от конфигурации системы. Компоновка базовых программных средств с мониторной частью теста комплекса (модули монитора) зависит от типа процессора, который должен быть проверен (К1801ВМ1, К1801ВМ2, К1801ВМ3), и его дополнительного оборудования.

2.6. Функции мультипрограммного монитора

По существу, мониторная часть программы теста комплекса управляет его запуском, устанавливает связь с его резидентными выбранными модулями (через интерпретатор команд TRAP), используется для управления выбранным модулем (через очередь приоритета обслуживания) и обеспечивает управление использованием памяти (при перемещении теста комплекса и циклическом сдвиге буфера записи).

2.6.1. Инициализация теста комплекса

Когда тест комплекса загружен, его мониторная часть обеспечивает инициализацию проверяемого оборудования с помощью выполнения двух программ: запуска и инициализации.

2.6.1.1. Программа запуска определяет условия выполнения теста комплекса посредством следующих действий:

- устанавливает вектор отказа питания;
- устанавливает управление программными и аппаратными прерываниями;
- устанавливает программный регистр переключателей (SWR) и значения слов состояний модулей;
- определяет характеристики оборудования системы и размер оперативной памяти.

Характеристики оборудования включают:

- тип процессора;
- дополнительное оборудование процессора, памяти, быстродействующей буферной памяти (в дальнейшем ББП);
- размер оперативной памяти (размер теста комплекса содержится в нулевой ячейке);
- должна ли записываться в память информация для установления в памяти верных контрольных разрядов (для случая, если установлен паритетный контроль информации (PARITY) или контроль информации с исправлением ошибок (ECC)).

Программа запуска завершает работу следующими действия-

ми теста комплекса:

- выводит идентификационное сообщение;
- выводит сообщение о размерах системы;
- выводит на консоль указатель командного режима (CMD>).

2.6.1.2. Программа инициализации устанавливает в начальное состояние определенные программные и аппаратные компоненты и выполняет запуск мультипрограммного монитора следующим образом:

- инициализирует очередь управления и очередь печати;
- инициализирует программы обработки ошибок логических функций (если они существуют);
- инициализирует выбранный тестовый модуль и его обслуживающие механизмы (т.е. устанавливает различные указатели и флаги);
- разрешает ввод команд с клавиатуры.

Когда программа теста комплекса инициализирована, выполняется мониторная часть программы, с клавиатуры разрешен ввод команд, что подтверждается выводом на терминал указателя командного режима (указатель командного режима - (CMD>). В командном режиме (CMD>) мультипрограммный монитор принимает и выполняет более 20 команд пользователя (см. п.п. 3.9.3.1.). Некоторые из команд (такие как запуск модуля, запрещение/разрешение варианта выполнения и модификация параметров модуля) разрешены только в командном режиме (CMD>). Другие команды, такие как разрешение и отмена выборки модуля и определение формата печати данных, могут быть введены как в командном режиме, так и в режиме, следующем за командой запуска, называемом режимом выполнения (BSY>).

Когда команда вводится с клавиатуры, каждый символ команды записывается в буфер ввода клавиатуры и содержится в нем до тех пор, пока не будет введен символ возврата каретки <ВК> и опознан монитором. Когда это произойдет, монитор (через планировщик задач) передает управление программе обработки вводимых с клавиатуры команд, чтобы выполнить пять следующих программ мультипрограммного монитора:

1) Программу обработки команды (OMDPRC);
2) Программу копирования команды (OMDCPY);
3) Программу декодирования команды (OMDDEC);
4) Программу обслуживания команды (OMDSRV);
5) Программу сброса команды (OMDRST).

2.7. Управление выбранными модулями

Мониторное управление тестовыми резидентными модулями в основном связано с выполнением двух задач:

1) Приоритетное планирование выполнения тестовых модулей;
2) Установление информационных связей с каждым тестовым модулем.

2.7.1. Приоритетное планирование

Как часть теста комплекса, каждый выбранный тестовый модуль устройства представляет тест, который должен выполняться во время работы теста комплекса. Если рассмотреть тестовый модуь с точки зрения функциональных и обрабатывающих способов выполнения операций (фоновые модули или модули вво-

да-вывода), то многие модули имеют общие черты. В этом смысле все модули в соответствии с общими функциональными и обрабатывающими режимами делятся на семь типов: фоновые модули (SBKMOD, NBKMOD, BKMOD), которые не проверяют работающие по прерыванию устройства, и модули ввода-вывода (IOMOD, IOMODX, IOMODP, IOMODR), которые проверяют управляемые по прерыванию устройства. Тип каждого модуля определяется через биты слова состояния, находящегося в интерфейсе тестового модуля. Используя биты слова состояния модуля, мультипрограммный монитор во время инициализации определяет приоритетное планирование с помощью предварительного описания каждого резидентного тестового модуля. Таким образом, порядок выполнения модуля определяется типом модуля. Ввод команды RUN или RUNL в командном режиме (CMD>) вызывает запуск выбранных модулей в порядке, заданном приоритетным планированием, и вывод на консоль указателя режима выполнения (BSY>).

Порядок запуска следующий:

1) Модули SBKMOD запускаются первыми. Каждый модуль будет отдельно выполняться один раз до перемещения теста комплекса в оперативной памяти и один раз при каждом следующем перемещении, если перемещение разрешено;

2) Модули NBKMOD запускаются во вторую очередь, за модулями SBKMOD. Каждый модуль выполняется отдельно один раз и никогда больше;

3) Все модули ввода-вывода IOMOD, X, P, R запускаются в третью очередь. Они выполняются одновременно и управляются прерываниями от проверяемых устройств;

4) Модули BKMOD будут запускаться последними, и каждый модуль выполняется отдельно. Поскольку эти модули имеют низший приоритет, они могут быть выполнены только тогда, когда никакой другой тип не будет занимать времени центрального процессора.

2.7.2. Связи модуля

Когда выбранные модули выполняются, связь с монитором устанавливается через программные связки. Они содержатся в основной части каждого модуля (TRAP вызовы), а также, в некоторых случаях, в параметрических ячейках интерфейсов модулей. Выбранным модулям разрешено использовать 19 вызовов. Некоторые вызовы используют все модули, другие используются только с модулями определенного типа. В основном, когда запрос модуля вызывает монитор, монитор отвечает служебной информацией или параметрами, такими как: параметры обслуживающего буфера, ошибка, сообщаемая программе или сообщение подпрограмме вывода.

2.8. Управление использованием памяти

Когда выполняется тест комплекса, ответственность за эффективное управление средствами памяти в тесте комплекса несет один мультипрограммный монитор.

Управление включает в себя:

- управление памятью;
- вычисление размера и расположения буферов записи по запросам тестовых модулей для ранее определенной области буфера записи;

- управление перемещением перемещаемой части теста комплекса и мультипрограммного монитора (если позволяет оборудование);
- генерацию шаблона максимальных помех в свободной области оперативной памяти для более эффективной проверки доступных ресурсов памяти.

2.8.1. Диспетчер памяти

Первоначально мультипрограммный монитор определяет, имеется ли диспетчер памяти. Если он имеется, то выполняются следующие функции управления:

- использование диспетчера памяти может быть разрешено или запрещено с помощью клавиатурных команд KTON и KTOFF;
- все регистры адреса страницы (PARS) и регистры-описатели страницы (PDR) устанавливаются;
- если вызовы CDATA& или DATCK& вызывают прерывание, передавая управление мультипрограммному монитору, то процессор будет переключен из режима ядра (KERNEL) в режим пользователя (USER).

Мониторное управление паритетом, коррекцией памяти ББП заключается в управлении ими с помощью клавиатурных команд и обработки ошибок.

2.8.2. Управление 22-битной адресацией

Мультипрограммный монитор управляет 22-битной адресацией с помощью клавиатурных команд MON и MOFF (см. п.3.10), заполняет регистры блока преобразования адреса и инициирует биты 22-битного расширенного адреса для области буфера записи.

2.8.3. Управление буфером записи

Тестовые модули (такие как IOMODX и IOMODP) могут вызывать мультипрограммный монитор, чтобы получить буфер записи (начальный адрес и размер). Из этого буфера тестовый модуль выполняет запись на устройство, с которым он работает. Мультипрограммный монитор управляет назначением адреса буфера записи и его размером. Он циклически перемещает адрес буфера (если перемещение буфера было разрешено по умолчанию или была использована команда RDTON). Это позволяет проверить работоспособность устройства с внепроцессорными запросами передачи данных во всей доступной оперативной памяти.

2.8.4. Область буфера записи

Для систем, имеющих не более 256к байт памяти, вся область памяти, не занятая тестом комплекса, предназначена для размещения и циклического сдвига буфера записи. Мультипрограммный монитор по запросам тестового модуля назначает конкретный адрес и размер буфера записи. Назначив эти параметры, монитор управляет перемещением буфера записи, изменяя при перемещении адресный параметр.

В системах, имеющих более 256к байт памяти, используется 22-битная адресация оперативной памяти. Память в этом случае разделяется на непрерывные сегменты об'емом по 248К слов. При выполнении теста комплекса область буфера записи

назначается в том же сегменте, где в данный момент находится тест комплекса. Это связано с ограничениями оборудования на адресацию. На рис.2 приведены три примера сегментации, изображающие отношения, существующие между текущим размещением теста комплекса и границами области буфера записи.

Несколько примеров для буфера записи

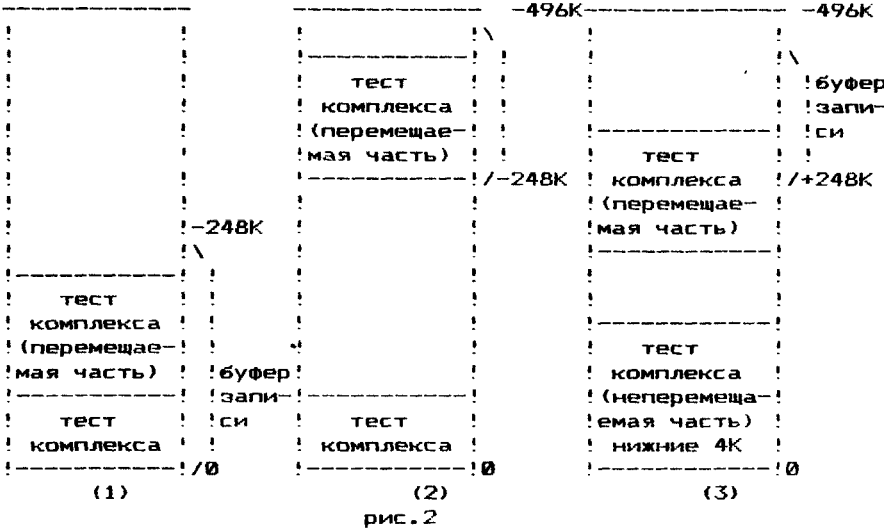


рис.2

На рис.2:

1) показана граница области буфера записи, когда перемещаемая часть теста комплекса находится в младшем сегменте 248K байт памяти. Заметим, если тест комплекса не будет перемещаться в этом сегменте (находится на прежнем месте), то пределы оперативной памяти, в которых перемещается буфер записи не будут меняться;

2) показана граница области буфера записи, когда перемещаемая часть теста комплекса перемещена в другой сегмент 248K байт памяти. Заметим, что, если тест комплекса перемещается, но в пределах этого сегмента 248K байт, то границы перемещения буфера записи остаются без изменения;

3) показаны границы области буфера записи, когда перемещаемая часть теста комплекса переходит через границу 248K байт памяти. В этом примере начало области буфера записи находится выше перемещаемой части теста комплекса (т.е. выше границы 248K байт), а конец ниже теста комплекса и ниже границы 248K байт. Размер сегмента по-прежнему не превышает 248к байт памяти.

Перемещение буфера записи (если оно разрешено) позволяет перемещать первоначально назначенный буфер записи с предварительно указанной начальной позиции, т.е. с первой свободной ячейки, находящейся за концом перемещаемой части теста комплекса. Впоследствии область буфера записи передвигается под управлением монитора. Как только буфер записи достигнет конечных старших адресов области, мультипрограммный

монитор переместит его в младшие адреса, и перемещение буфера будет продолжено. Если перемещение буфера запрещено вследствие недостатка памяти или клавиатурных команд, то используется буфер записи, расположенный в начальной позиции за концом перемещаемой части теста комплекса.

Как только расширенный модуль ввода-вывода запрашивает пространство для буфера записи, мультипрограммный монитор определяет, достаточно ли места в оперативной памяти для запрашиваемого буфера. Если места для буфера достаточно, то мультипрограммный монитор разрешает модулю использовать буфер запрошенного размера, если места для буфера недостаточно, то мультипрограммный монитор вычисляет допустимый размер буфера записи и записывает его в интерфейс модуля. Мультипрограммный монитор передает в интерфейс тестового модуля начальный адрес буфера записи.

Этот начальный адрес буфера записи является адресом первой свободной ячейки, расположенной непосредственно за перемещаемой частью теста комплекса и модифицируется при перемещении буфера записи.

Этот адрес мультипрограммный монитор устанавливает в начальное значение, равное базовому:

- при первом запуске теста комплекса в фазе инициализации;
- непосредственно вслед за перемещением теста комплекса в оперативной памяти.

Указатель буфера может изменяться только тогда, когда перемещение буфера записи разрешено, и все запросы, связанные с использованием перемещаемого буфера записи, запрещены. Базовый адрес буфера записи используется как первое значение указателя буфера записи.

Для получения нового начального адреса буфера записи текущее положение указателя необходимо увеличить, чтобы определить первую свободную ячейку за концом первого запрашиваемого буферного пространства. Если имеется достаточно оперативной памяти, новый адрес начала буфера будет равен:

новый адрес начала буфера = текущее значение начала буфера + размер буфера + 1

В противном случае указателю будет присвоен адрес, который соответствует доступному пространству. В этом случае каждый последующий запрос будет продвигать указатель через область буфера до достижения конца области памяти, после чего указатель буфера будет установлен на нижнюю часть области памяти для буфера записи, затем процесс продвижения будет продолжен. Как показано на рис.2, в зависимости от перемещения нижняя часть области буфера может снова быть в конце теста комплекса (рис.2 (1) и (2)) или ниже теста комплекса (3). В этих случаях продвижение будет продолжаться до последней свободной ячейки, в конце концов достигнув нижней границы перемещаемой части теста комплекса. В этой точке, поскольку сам тест комплекса не может использоваться как область буфера записи, адрес указателя будет установлен в конец теста комплекса, и процесс перемещения продолжится.

2.8.5. Перемещение теста комплекса

Когда программа теста комплекса загружается в память,

программа разделена на две секции:

1) Фиксированная часть мультипрограммного монитора, которая размещается в нижних 4К памяти и никогда не перемещается;

2) Перемещаемая часть теста комплекса, которая включает перемещаемую часть мультипрограммного монитора и все тестовые модули. Перемещаемая часть теста комплекса первоначально размещается непосредственно за неперемещаемой частью мультипрограммного монитора и может последовательно перемещаться в оставшейся части оперативной памяти (см. рис.2). Непрерывное перемещение будет возможно только если:

- система содержит диспетчер памяти (ДП);
- использование диспетчера памяти разрешено командой KTON;
- процесс перемещения не запрещен (т.е. тест комплекса был запущен командой RUN, а не командой RUNL).

Процессом перемещения управляет мультипрограммный монитор, который позволяет выполнять циклический сдвиг перемещаемой части теста комплекса в основной оперативной памяти с помощью процедур перемещения, чтобы проверить работоспособность теста комплекса во всей оперативной памяти (кроме 4К, где расположена неперемещаемая часть монитора).

Во время процесса каждая новая операция перемещения инициализируется мультипрограммным монитором, когда все модули ввода-вывода завершили первый шаг, или когда все модули BKMOD (если нет модулей ввода-вывода) завершили шаг. Однако, поскольку модули имеют разное время выполнения, некоторые модули завершают шаг и будут вновь перезапущены. Поэтому все модули должны быть остановлены перед следующей итерацией, в этой точке они будут вновь запущены, когда перемещение будет завершено.

Когда операция перемещения завершена, выводится сообщение: RELOCATED TO XXXXXX (перемещен в XXXXXX)
где XXXXXX - указывает новый физический начальный адрес перемещаемой части теста комплекса.

Выполняется два типа операции перемещения:

1) Перемещение на постоянную величину;

2) Перемещение на величину, задаваемую генератором случайных чисел (случайное перемещение).

Бит 8 программного регистра переключателей (SW) управляет способами перемещения теста комплекса:

- SW08=1, запрещает случайные перемещения теста комплекса;
- SW08=0, разрешены оба типа перемещения, первым выполняется перемещение на постоянную величину один раз по всей памяти. После завершения перемещения теста комплекса на постоянную величину будет непрерывно продолжаться процесс случайного перемещения.

2.8.6. Перемещение теста комплекса на постоянную величину

Перемещаемая часть теста комплекса выполняется при начальном запуске с базового адреса, определенного по умолчанию, или с базового адреса, заданного в команде RUN или RUNL. При перемещении теста комплекса с постоянным шагом эти базо-

вые адреса изменяются на постоянную величину (обычно 4K). Тест комплекса перемещается в оперативной памяти до тех пор, пока в ней достаточно места для перемещения, и верхняя граница памяти еще не достигнута. Затем мультипрограммный монитор перемещает тест комплекса к начальным базовым адресам, и на этом один цикл перемещения завершается и перемещение на постоянную величину будет продолжено.

2.8.7. Перемещение теста комплекса на случайную величину

Если цикл перемещения на постоянную величину закончен и разрешено перемещение на случайную величину, то выполнение теста комплекса будет продолжено с перемещением его на случайную величину, вычисляемую монитором.

В начале случайного перемещения тест комплекса запускается с начального базового адреса в нижней части памяти. Затем тест комплекса перемещается в самый старший возможный адрес, по которому может быть размещен тест комплекса. Перемещение продолжается, используя случайные числа, генерируемые мультипрограммным монитором.

2.8.8. Генерация шаблона максимальных помех в оперативной памяти

Область памяти, не содержащая тест комплекса, определяется как свободная область памяти, обслуживающая буфер записи. Когда вводится команда запуска теста комплекса (т.е. RUN или RUNL), в память автоматически записывается шаблон для того, чтобы эффективно проверять работоспособность общей шины при операциях ввода-вывода по передаче данных.

Если перемещение теста комплекса разрешено, во время каждого успешного перемещения тест комплекса накладывается на шаблон. Однако при перемещении теста комплекса в нижнюю область памяти шаблон максимальных помех вновь перезаписывается.

2.9. Обработка прерываний

Обработка прерываний мультипрограммным монитором включает обработку аппаратных и программных прерываний. Употребление программного прерывания обеспечивает доступ к выбранным для обработки программам через TRAP инструкции, когда тестовый модуль устройства требует обслуживания мультипрограммным монитором, чтобы, например, обслужить буфер или сообщить об ошибке.

Аппаратные прерывания используются, чтобы обеспечить доступ к специальным обрабатывающим программам мультипрограммного монитора, когда процессор обнаруживает ошибки при выполнении инструкции.

2.9.1. Программное прерывание

Когда тестовый модуль использует макровызов для связи с мультипрограммным монитором (GWBUF&, GETPAS и др.), закодированный инструкцией TRAP, на этой инструкции тестового модуля происходит прерывание по вектору 34. Управление передается в мультипрограммный монитор подпрограмме обслуживания прерываний. Код прерывания идентифицируется, а содержимое регистров тестового модуля, действительный адрес вызова сох-

раняются. Монитор передает управление программе, соответствующей коду вызова, и выполняет некоторые операции:

- запрос выполняется, и управление передается назад в тестовый модуль (т.е. OTOA%, RAND% и др.);
- запрос выполняется и/или ставится в очередь на вывод (т.е. CDATA%, DATER% и др.);
- запрос ставится в очередь управления для последующего обслуживания (т.е. PIRG%, BREAK% и др.);
- запрос выполняется и управление возвращается подпрограмме планировщика мультипрограммного монитора (т.е. END%, EXIT%).

2.9.2. Аппаратные прерывания

Аппаратные прерывания вызываются внутренними ошибками, связанными с центральным процессором и выбранной памятью. Ошибки разделяются на:

- системные ошибки;
- ошибки паритета (основной или ББП) и некорректируемые ошибки памяти;
- ошибки диспетчера памяти (ДП).

2.9.3. Системные ошибки

Системные ошибки:

- вызывают прерывание по вектору 04 (несуществующая память, нечетный адрес и др.);
- вызывают прерывание по вектору 10 (резервная инструкция).

Мультипрограммный монитор сохраняет содержимое PC, PSW и SP, инициализирует систему и выводит сообщение о системной ошибке.

Возможны следующие варианты:

- выводится сообщение об ошибке;
- если ошибка обнаружена в тестовом модуле, после четырех ошибок тестовый модуль снимается;
- если при выполнении всего теста комплекса обнаружено большое количество системных ошибок, выполнение теста комплекса заканчивается и система возвращается в командный режим (CMD>).

Исключая последний случай (т.е. возврат в CMD>), после обработки системной ошибки тест комплекса перезапускается следующим способом: все модули, завершившие шаг, будут вновь инициализированы сначала (с метки RESTART), а оставшиеся модули инициализируются с метки START.

2.9.4. Ошибки четности и двойные ошибки расширенного корректирующего кода

Если ошибка четности памяти, ошибка четности ББП или двойная ошибка корректирующего кода вызывает прерывание по вектору 114, система инициализируется, и содержимое соответствующих регистров выводится вместе с соответствующими сообщениями об ошибке. Тем не менее, если имеют место десять таких ошибок, выполнение завершается и система возвращается в режим CMD>.В противном случае тест комплекса перезапускается.

2.9.5. Ошибки диспетчера памяти (ДП)

Если ошибка диспетчера памяти вызывает прерывание по вектору 250, система инициализируется, и содержимое имеющихся основных регистров (SR0 и SR2, SR1 и SR3) выводится вместе с соответствующим сообщением об ошибке. Если имеет место ошибка, выполнение завершается и система возвращается в режим CMD>.

2.10. Программа конфигуратор/компоновщик DXCL

Программа конфигуратор/компоновщик TMOC DXCL используется для создания теста комплекса. Вначале выполняется процесс конфигурации (через построение таблицы компоновки), следующим выполняется процесс генерации (через выполнение команды LINK), в результате создается индивидуальный модуль теста комплекса. Пользователь указывает модуль мультипрограммного монитора, тестовые модули, записывает их параметры в таблицу конфигурации (С-таблицу) и, используя команду компоновки, получает модуль теста комплекса.

2.10.1. Процесс конфигурации облегчает процесс генерации. Загрузив программу DXCL, пользователь выполняет процесс конфигурации, установив режим конфигурации, и создает таблицу компоновки (С-таблицу). Во время ее создания имя желаемого монитора вводится в таблицу. Кроме того, отдельно вводится имя каждого необходимого модуля с некоторыми параметрами, такими как адреса устройства и вектора, уровень приоритета. Таблица компоновки максимально содержит 40 входов, каждый из которых содержит по 11 слов (один вход используется для монитора и 39 для тестовых модулей). Когда создание таблицы компоновки завершено, информация, необходимая для процесса компоновки, имеется, пользователь выходит из режима конфигурации и начинает генерацию.

2.10.2. Процесс генерации

Когда создание таблицы компоновки завершено, пользователь начинает процесс генерации с помощью команды LINK. По существу, результат процесса генерации – создание теста комплекса, при котором рассматривается таблица компоновки и из нее выбирается информация, указанная пользователем, подходящий модуль монитора (из библиотеки монитора) и выбранные тестовые модули.

Когда каждый модуль будет выбран, он отдельно обрабатывается и выводится по блокам как часть теста комплекса.

3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

Этот раздел содержит информацию, необходимую для того, чтобы выполнить генерацию теста комплекса для проверки оборудования из выбранного пользователем мультипрограммного монитора и тестовых модулей для проверки устройств и запустить его для проверки ПЭВМ.

Предварительно необходимо уточнить конфигурацию проверяемой системы (тип процессора, размер памяти, типы внешних устройств, их вектора, адреса, уровни приоритетов и т.д.), так как при построении таблицы компоновки теста комплекса

нужно указать параметры проверяемого оборудования и режимы проверки.

3.1. Подготовка к генерации теста комплекса

Перед генерацией теста комплекса необходимо ознакомиться с функциями мультипрограммных мониторов и выбрать из них тот, который удовлетворяет требованиям пользователя. Необходимо ознакомиться с описаниями и текстами программ тестовых модулей для выбора режима проверки каждого устройства. Необходимо ознакомиться с алгоритмом проверки в каждом тестовом модуле и сообщениями тестового модуля. Это знакомство позволит понять, что происходит на устройствах во время выполнения теста комплекса и ориентироваться в сообщениях, выводимых тестом комплекса. Следует заметить, что если в вычислительном комплексе используются базовые адреса подключения устройств, то можно воспользоваться существующими параметрами устройств. Эти параметры описаны в документации на каждый тестовый модуль и их можно найти в тексте программы каждого тестового модуля (в интерфейсе тестового модуля).

Для каждого проверяемого устройства должны быть установлены следующие параметры:

- DVA адрес устройства;
- VCT адрес вектора;
- BR1 уровень приоритета 1 на системной магистрали;
- BR2 уровень приоритета 2 на системной магистрали;
- DVC счетчик устройств;
- SR1-SR4 программные регистры переключателей.

При генерации теста комплекса необходимо помнить, что каждый тестовый модуль предназначен для проверки устройств, подключенных к одному контроллеру. Если в конфигурацию комплекса включено несколько одноименных контроллеров, то для проверки их в режиме одновременной работы необходимо включить в тест комплекса такое же количество одноименных тестовых модулей. В каждый тестовый модуль нужно ввести параметры контроллера, для проверки которого предназначен этот модуль и количество устройств, подлежаших проверке под управлением этого контроллера.

3.2. Процесс генерации

Программа DXCL компонует выбранный пользователем мультипрограммный монитор с тестовыми модулями и создает тест комплекса в абсолютном загрузочном формате. Мониторные модули в об'ектном коде располагаются в библиотеке мониторов (файл XXXXXX.LIB). Когда пользователь указывает версию мультипрограммного монитора для включения в тест комплекса. Программа конфигуратор/компоновщик выбирает из библиотеки модулей мониторов только те модули, которые должны войти в состав выбранной версии.

Тестовые модули устройств в перемещаемом об'ектном формате расположены на носителе и не могут быть использованы для автономной проверки устройств (файлы YYYYYY.OBJ). Пользователь назначает имя теста комплекса, используя расширения BIN или BIC. Построенный тест комплекса пригоден для проверки вычислительного комплекса той конфигурации оборудования, которая была задана пользователем в процессе построения таб-

лицы компоновки. При изменениях состава вычислительного комплекса и подключении дополнительных устройств необходимо вновь построить новый тест комплекса с учетом изменившейся конфигурации оборудования.

3.3. Команды и ключи программы DXCL

Команды программы DXCL приведены в табл.1.

Таблица 1

Формат команды	Описание
1	2
BOOT устр.вв.:	загружает монитор TMOC с указанного устройства
CHECK устр.:имя файла.рас	проверка правильности формата об'ектного файла и контрольной суммы
CNF	вход в режим конфигурации
EXIT	возврат в монитор ТЕДОС
GETC устр:имя файла.рас	ввод в память таблицы компоновки с указанного устройства
LINK устр0:имя файла.рас <устр1:имя файла.EXT	компоновка теста комплекса с указанного устройства и вывод загрузочного модуля на устройство с файловой структурой
MOD <АДР>	открывает содержимое ячейки для модификации
PRINTC	вывод таблицы компоновки на печатающее устройство
PRINTM устр:имя файла.рас	вывод карты загрузки на печатающее устройство
SAVC устр0:имя файла.рас	сохранение таблицы компоновки на указанном устройстве
SAVM устр0:имя файла.рас	сохранение карты загрузки на указанном устройстве
TYPEC	вывод таблицы компоновки на консольный терминал
TYPEM устр1:имя файла.рас	вывод карты загрузки на консольный терминал
BR1 <ЧИСЛО>	ввод уровня приоритета 1 старшего байта
BR2 <ЧИСЛО>	ввод уровня приоритета 2 младшего байта
CL	очистка таблицы компоновки
DVA <АДР>	ввод адреса устройства (базовый адрес устройства)
DVC <ЧИСЛО>	ввод счетчика устройств (число выбранных устройств)
EX	выход из режима конфигурации
KI	исключение текущего входа из таблицы компоновки
MDL	вывод строки параметров заголовка текущего входа модуля в таблице компоновки

1	2
MDL <ИМЯ МОДУЛЯ>	ввод имени модуля в таблице компоновки
MON <ИМЯ МОДУЛЯ>	ввод имени мультипрограммного монитора в таблице компоновки
NXT	вывод строки параметров заголовка следующего входа модуля в таблице компоновки
POINT <ИМЯ МОДУЛЯ>	вывод строки параметров входа заданного модуля в таблице компоновки. Передвигает указатель таблицы компоновки
SR1 <ЧИСЛО>	ввод значения в 1-ый программный регистр переключателей
SR2 <ЧИСЛО>	ввод значения во 2-ой программный регистр переключателей
SR3 <ЧИСЛО>	ввод значения в 3-ий программный регистр переключателей
SR4 <ЧИСЛО>	ввод значения в 4-ый программный регистр переключателей
VCT <АДР>	ввод адреса вектора устройств

Ключи программы DXCL приведены в табл.2.

Таблица 2

Ключ	Описание
1	2
/MLP	во время компоновки выводит карту загрузки на печатающее устройство
/MP	во время компоновки выводит карту загрузки на консольный терминал
/NP	в режиме конфигурации запрещает подсказку оператору

ПРИМЕЧАНИЕ. Команда модификации (MOD АДР) аналогична команде модификации теста комплекса.

3.4. Загрузка и процедура запуска программы DXCL

Программа DXCL, размещенная на системном носителе, поддерживаемом монитором системы, может быть загружена с помощью монитора системы.

После того, как монитор системы загружен, он идентифицирует себя, запрашивает ввод даты, выводит адрес рестарта, точку ".", и ожидает команду оператора. После точки "." оператор печатает имя файла конфигуратора (.R <ИМЯ ФАЙЛА> без расширения), выбранная программа загружается и самозапускается.

3.4.1. Процедура запуска программы DXCL

После загрузки программа DXCL идентифицирует себя, выводит адрес и подсказку.

Пример:

```
.R DXCL                    ;загрузка-старт программы
                           ;DXCL
HUKC-7 DO-MMM-YY- TMOC конф/комп ВЕР84
                           ;программа идентифицирована
РЕСТАРТ:006570             ;адрес рестарта
нужна справка ?            ;нужна подсказка?
Y<ВК> или только <CR>      ;при вводе <ВК> подсказка
                           ;не печатается
*                          ;знак режима ввода команд с
                           ;клавиатуры
```

В этом примере оператор может воспользоваться подсказкой или нет (нажав <ВК>). Если нужна подсказка, он вводит Y<ВК>, и все имеющиеся команды и ключи будут выведены прежде, чем появится звездочка "*" - указатель режима ввода команд клавиатуры. Когда выводится звездочка "*", программа DXCL готова принимать клавиатурные команды, необходимые для генерации теста комплекса.

Пользователь может перезапустить программу с адреса рестарта, выведенного в сообщении.

3.5. Классификация команд программы DXCL

Некоторые команды программы DXCL могут быть использованы только в режиме конфигурации в то время, как остальные команды могут выполняться в обоих режимах.

Для успешного создания теста комплекса необходимо знание клавиатурных команд.

Для облегчения изучения клавиатурных команд их подразделяют на четыре типа: 1, 2, 3, и 4. Первые три типа предназначены для обслуживания процесса построения теста комплекса. Тип 4 содержит всего одну команду, предназначенную для модификации ячеек оперативной памяти.

Команды каждого типа и ключи (/MLP и т.д.), могут быть использованы для изменения режима работы определенных команд (CNF и команды LINK).

Тип 1: команды режима конфигурации

Вход в режим конфигурации по команде CNF разрешает пользователю с помощью следующих команд в группе построить таблицу компоновки (С-таблицу).

```
CNF                        ;вход в режим конфигурации
MON <ИМЯ МОДУЛЯ>           ;ввод имени мультипрограммного
                           ;монитора в таблицу компоновки
MDL                        ;печать строки параметров текущего
                           ;входа модуля
MDL <ИМЯ МОДУЛЯ>           ;ввод имени модуля в таблицу
                           ;компоновки
DVA <ADDR>                 ;ввод адреса устройства
VCT <ADDR>                 ;ввод адреса вектора
BR1 <ЧИСЛО>                ;ввод уровней приоритета
BR2 <ЧИСОО>                ;устройства на системной магистрали
DVC <ЧИСЛО>                ;ввод счетчика устройств
                           ;количество проверяемых устройств
```

```
SR1-SR4 <ЧИСЛО>          ;ввод значений в программные
                         ;регистры переключателей
KI                       ;уничтожение текущего входа
POINT <ИМЯ МОДУЛЯ>       ;вывод строки параметров текущего
                         ;входа модуля
NXT                      ;вывод строки параметров
                         ;следующего входа модуля
CL                       ;очистка таблицы компоновки
EX                       ;выход из режима конфигурации
```

При установке режима конфигурации командой CNF на экран терминала автоматически выводятся команды с подсказками, благодаря которым пользователь последовательно создает таблицу компоновки.

Последовательность подсказок может быть запрещена ключом /NP ("нет подсказки") с командой CNF.

Тип 2: команды компоновки

При вводе режима генерации командой LINK модули мультипрограммного монитора и тестовые модули вызываются с устройства ввода, компонуются и выводятся блок за блоком на устройство вывода.

Формат команды:

```
LINK устр0:имя файла.рас<устр1:AAAAAA.LIB
;компоновка и вывод теста комплекса на заданное
;устройство
;AAAAAA.LIB - имя библиотеки мультипрограммного
;монитора
;уст0, уст1 - физические имена устройств
```

Модули с устройства ввода (устр1) компонуются и передаются на устройство вывода блок за блоком. Аргумент "имя файла" состоит из шести символов; аргумент "расширение" - из трех символов.

Тип 3: команды, управляющие вводом-выводом

Эти команды могут быть использованы в или вне режима конфигурации и используются для управления вводом и выводом.

```
TYPEC                          ;вывод таблицы компоновки на
                               ;терминал
PRINTC                         ;вывод таблицы компоновки на
                               ;печатающее устройство
SAVC устр0:имя файла.рас       ;вывод таблицы компоновки на
                               ;указанное устройство
GETC устр1:имя файла.рас       ;ввод в оперативную память
                               ;таблицы компоновки
SAVM устр0:имя файла.рас       ;вывод карты загрузки на
                               ;устройство
TYPEM устр1:имя файла.рас      ;поиск и вывод на консоль карты
                               ;загрузки с устройства
PRINTM                         ;поиск и вывод на устройство
                               ;печати карты загрузки
CHECK устр1:имя файла.рас      ;проверка об'ектного формата
                               ;модуля
                               ;и контрольной суммы
EXIT                           ;выход в монитор системы
BOOT устр:                     ;перзагрузка монитора системы
```

Тип 4: общая вспомогательная команда и управляющий символ "СУ/С"

Команда MOD ADDR может использоваться в или вне режима конфигурации для изменения по желанию пользователя заданной ячейки.

Формат команды:

MOD ADDR ;открывает ячейку для модификации

"СУ/С" (<СУ>/<С>) - управляющий символ клавиатуры.

Ввод с клавиатуры этого символа приводит к прекращению любой операции.

3.5.1. Командные ключи

Существует три командных ключа. Ключи /MLP и /MP используются для расширения и изменения работы команды компоновки (LINK). Ключ (/NP) используется для изменения работы команды режима конфигурации (CNF).

Вывод карты загрузки на печатающее устройство

Ключ /MLP можно добавить к команде LINK, чтобы вывести карту загрузки на печатающее устройство в процессе компоновки.

П р и м е р :

```
*LINK MY0:TEST1.BIN<MY0:XMON??.LIB/MLP <ВК>
```

Вывод карты загрузки на консольный терминал

Ключ /MP можно добавить к команде LINK для вывода карты загрузки на консольный терминал.

П р и м е р :

```
*LINK MY0:TEST1.BIN<MY0:XMONC0.LIB/MP <ВК>
```

Режим работы без подсказки

Ключ /NP можно добавить к команде CNF для запрещения запросов параметров при создании таблицы компоновки.

П р и м е р :

```
*CNF/NP <ВК>
```

3.5.2. Команды режима конфигурации

Рассмотрим команды режима конфигурции, примеры использования команд и действия оператора.

К этим командам относятся: CNF, MON, MDL, DVA, VCT, BR1, BR2, DVC.

Команда CNF

Формат команды:

```
*CNF <ВК>
*CNF/NP <ВК>
```

Команда CNF используется для входа в режим конфигурации и создания таблицы компоновки. Если таблица компоновки пустая, то программа конфигуратор/компоновщик напечатает:

*МОНИТОР:

Пользователь должен ввести имя мультипрограммного монитора, который он выбирает при генерации теста комплекса и закончить нажатием клавиши <ВК>. Когда имя мультипрограммного монитора принято, программа DXCL выводит "*". Разрешенные имена мониторов: A, C, E.

Пользователь набирает команду ввода имени тестового модуля:

*MDL <ИМЯ МОДУЛЯ> <ВК>

Если команда CNF была введена без ключа /NP, то команда MDL запрашивает девять последующих параметров для ввода их в тестовый модуль устройства.

Когда последовательность параметров исчерпана, выводится итоговая строка параметров модуля.

Примеры:

```
                                ;монитора
        MDL WXYZ <ВК>           ;ввод имени тестового модуля
        DVA 177540 <ВК>         ;ввод адреса устройства
        VCT 230 <ВК>            ;ввод вектора устройства
        BR1 5 <ВК>              ;ввод первого приоритета
        BR2 5 <ВК>              ;ввод второго приоритета
        DVC 2 <ВК>              ;ввод счетчика проверяемых
                                ;устройств
        SR1 <ВК>                ;ввод значений программных
        SR2 4000 <ВК>           ;регистров переключателей
        SR3 <ВК>                ;тестового модуля для выбора
        SR4 <ВК>                ;режима проверки
 WXYZ DVA=177540 VCT=000230 BR1=000240 BR2=000240
 DVC=000003 SR1=000000 SR2=004000 SR3=000000 SR4=000000
```

ПРИМЕЧАНИЕ. Если пользователь хочет прервать последовательность ввода параметров, он должен нажать клавиши <СУ>/<С>. Параметры, введенные до <СУ>/<С>, сохраняются. Далее пользователь может вводить любое другое имя тестового модуля.

Если команда CNF введена с ключом /NP, то последовательность подсказок подавляется.

Пример:

```
        *CNF/NP <ВК>            ;в примере пользователь
        *MONITOR:B <ВК>         ;сам вводит каждую
        *MDL QRST <ВК>          ;команду с параметрами,
        *DVA 177530 <ВК>        ;когда на экран выведена
        *VCT 230 <ВК>           ;звездочка "*"
```

и т.д.

```
                                ;это позволяет изменять по
                                ;отдельности каждый параметр
```

ПРИМЕЧАНИЕ. Если после использования команды CNF с ключом /NP пользователь вновь желает получить последовательность подсказок параметров, он может ввести команду CNF без ключа, не выходя из режима конфигурации. Если пользователь не желает изменять параметр, назначенный по умолчанию в тестовом модуле, то на запрос параметра он вводит <ВК> или <ПС>.

Команда MON

Формат команды:

*MON <ИМЯ> <ВК>

Команда MON используется для изменения имени мультипрограммного монитора в таблице компоновки. Разрешенные имена мониторов: A, C, E.

Команда MDL

Формат команды:

1) MDL <ИМЯ МОДУЛЯ> <ВК>
2) MDL <ВК>

Команда MDL используется для ввода заданного имени тестового модуля и его параметров в таблицу компоновки и вывода содежимого текущего входа модуля.

Формат 1 команды MDL используется для ввода имени тестового модуля и его параметров в первую свободную строку таблицы компоновки.

Тестовым модулям присваивается имя, состоящее из четырех символов, в следующем формате:

DKAA

DK - имя устройства, проверяемого тестовым модулем, два символа;

A - код тестового модуля (1 символ). Один или более тестовых модулей могут существовать для одного и того же устройства, каждому тестовому модулю, если их несколько, соответствует свой код;

A - версия модуля (1 символ). Если версия неизвестна, то символ "?" может использоваться, как четвертая характеристика имени тестового модуля. Символ "?" обозначает слово "любой".

Во время работы команды LINK все символы "?" в таблице компоновки заменяются буквами-версиями.

Последовательность подсказок для ввода параметров в текущий вход тестового модуля в команде MDL приведена в описании команды CNF.

Формат 2 команды MDL используется для вывода итоговой строки параметров текущего входа модуля в следующем формате:

```
DKAA DVA-000000 VCT-000000 BR1-000000 BR2-000000
     DVC-000000 SR1-000000 SR2-000000 SR3-000000
     SR4-000000
```

Команда DVA

Формат команды:

DVA <АДРЕС> <ВК>

Команда DVA используется для ввода адреса устройства в текущий вход таблицы компоновки, если в тестовом модуле не указан адрес устройства или если устройство имеет адрес, отличный от базового.

Вводимый адрес должен быть четным и восьмеричным.

Пример:

```
*MDL DKAA <ВК>
*DVA 177600 <ВК>
```

Напечатав после этого: *MDL <ВК>

получим следующее содержимое текущего входа тестового модуля:

```
DKAA DVA-177600 VCT-000000 BR1-000000 BR2-000000
     DVC-000000 SR1-000000 SR2-000000 SR3-000000
     SR4-000000
```

Команда VCT

Формат команды:

VCT <АДРЕС> <ВК>

Команда VCT используется для ввода адреса вектора устройства в текущий вход таблицы компоновки, если адрес не указан в тестовом модуле, или если вектор находится по нестандартному адресу.

Адрес должен быть восьмеричным, четным и не превышать 774(8).

Пример:

```
*VCT 200 <ВК>
*MDL <ВК>
KAA    DVA-177600 VCT-000200 .......
```

В этом примере итоговая строка приведена неполностью.

Команды BR1 и BR2

Формат команд:

*BR1 <ЧИСЛО> <ВК>

*BR2 <ЧИСЛО> <ВК>

Команды BR1 и BR2 используются для ввода первого (BR1) и второго (BR2) уровней приоритетов в текущий вход тестового модуля. Эти параметры вводятся, если уровни не заданы в тестовом модуле или если устройство имеет нестандартные уровни приоритетов.

Пример:

```
*BR1 6  <ВК>
*BR2 4  <ВК>
```

Вводимое восьмеричное число, не должно превышать 7. Напечатав MDL <ВК>, получим следующее содержимое текущего входа модуля:

```
DKAA DVA=17600 VCT=000200 BR1=000300 BR2=000200
```

В этом примере итоговая строка неполная. При выводе содержимого текущего входа тестового модуля указываются уровни приоритетов BR1 и BR2, преобразованные в эквивалентные слова состояния процессора.

Команда DVC

Формат команды:

*DVC <ЧИСЛО> <ВК>

Команда DVC вводит в текущий вход таблицы компоновки максимальное число проверяемых модулем устройств. Например, чтобы указать, что устройство управления дисками имеет два дисковода, следует напечатать: *DVC 2 <ВК>

Вводимое число должно быть десятичным. Указанное число устройств должно существовать и не превышать 16; адреса устройств должны последовательно располагаться за первым адресом устройства. Промежутки в адресах не разрешаются, кроме случая некоторых устройств, где второе и третье устройства имеют об'единенный адрес. В этом случае подразумевается, что адреса устройств идут по порядку без пропусков.

Пример:

```
*DVC 5 <ВК>          ;ввод счетчика проверяемых устройств
*MDL <ВК>            ;печать текущего входа модуля
WXYZ DVA=177600      VCT=000200 BR1=000300 BR2=000200
     DVC=000037
```

В примере итоговая строка неполная. Десятичное число 5 преобразуется в позиционный код 37.

Соответствие разрядов номерам устройств приведено на рис.3.

рис.3

Тестирование многочисленных устройств всегда последовательно. Таким образом, бит-схема может иметь последовательные единичные биты, которые приравниваются номерам устройств по принципу один к одному.

Подобным же образом последовательно выбираются адреса каждого устройства (по возрастанию или по убыванию) из выбранных с помощью позиционного кода.

Команды SR1-SR4

Формат команды:

```
*SR1 <ЧИСЛО>  <ВК>
*SR2 <ЧИСЛО>  <ВК>
*SR3 <ЧИСЛО>  <ВК>
*SR4 <ЧИСЛО>  <ВК>
```

Команды SR1-SR4 используются для ввода в текущий вход таблицы компоновки входных параметров, определяющих режимы работы тестового модуля. При необходимости они используются для модификации выполнения тестового модуля.

Команды SR1-SR4 используются раздельно для того, чтобы ввести восьмеричные значения в программные регистры переключателей текущего входа модуля. Значение и смысл SR1-SR4 указаны в [1].

```
*SR2 4000 <ВК>      ;устанавливает бит 11 в программном
                    ;регистре 2 переключателей
*MDL       <ВК>     ;вывод текущего входа модуля
DKAA DVA=177600     VCT=000200 BR1=000300 BR2=000200
     DVC=000037     SR1=000000 SR2=004000 SR3=000000
     SR4=000000
```

В примере бит 11 в программном регистре переключателей указывает, что печатающее устройство имеет 132 символа в строке (вместо 80).

Команда KI

Формат команды:

```
*KI  <ВК>
```

Команда KI используется для исключения текущего входа модуля из таблицы компоновки.

Когда текущий вход модуля уничтожен по команде KI, и если содержимое таблицы компоновки выводится по команде PRINT или TYPEC, то для исключенного входа модуля (с помощью команды KI) выводится сообщение <EMPTY>(пусто).

Команда POINT

Формат команды:

```
*POINT <ИМЯ МОДУЛЯ>  <ВК>
```

Команда POINT используется для поиска имени тестового модуля в таблице компоновки и вывода строки параметров этого модуля. После выполнения команды указатель таблицы компоновки устанавливается на входе найденного модуля.

Если имя модуля не найдено, выводится сообщение:

```
?INVALID NAME (неверное имя)
```

Если указанное имя модуля обнаружено, выводится строка его входных параметров.

Пример:

```
*POINT DKAA <ВК>
DKAA DVA=177600.....
```

Команда NXT

Формат команды:

```
*NXT  <ВК>
```

Команда NXT используется для вывода строки параметров следующего входа модуля. Указатель таблицы компоновки уста-устанавливается на следующем входе.

Если следующий вход модуля найден, выводится содержимое его заголовка, если не найден, то печатается "*".

Пример:

```
*NXT <ВК>
DKAA DVA-177600  и т.д.
```

Команда CL

Формат команды:

```
*CL  <ВК>
*MONITOR:<ИМЯ>  <ВК>
```

Команда CL очищает таблицу компоновки для создания новой таблицы. После очистки таблицы компоновки программа DXCL запрашивает имя мультипрограммного монитора как в команде CNF. Старая таблица компоновки уничтожена. Можно создать таблицу компоновки.

Команда EX

Формат команды:

```
*EX  <ВК>
```

Команда EX используется для выхода из режима конфигурации, когда запрещено построение таблицы компоновки. Повторный вход осуществляется через команду CNF. Если вход вновь выполнен, то наличие правильного входа мультипрограммного монитора исключает запрос имени мультипрограммного монитора. Таким образом, программа просто указывает первый вход модуля в таблице компоновки и выводит командную подсказку "*".

```
*EX <ВК>              ;выход из режима компоновки
*CNF <ВК>             ;перезапуск режима компоновки
*                     ;вывод командной подсказки
```

3.5.3. LINK - команда процесса компоновки

После выхода из режима конфигурации следует процесс компоновки, который выполняется по команде LINK.

Формат команды:

LINK устр0:[имя файла.рас]<устр1:[имя файла.рас]

Если устр0 или устр1 пропущены, то по умолчанию используется системное устройство.

Формат команды LINK для устройств с бесфайловой структурой (перфолента):

LINK устр0<устр1

Формат требует, чтобы были заданы только устройства ввода-вывода.

Во время выполнения первой фазы устанавливается перфолента с библиотекой мультипрограммного монитора, тестовые модули будут запрашиваться последовательностью подсказок.

Формат команды для устройств с файловой структурой (диски):

LINK устр0:имя файла.рас<устр1:LIBNAM.LIB

Формат требует, чтобы вместе с устройствами ввода-вывода было задано имя файла библиотеки мультипрограммного монитора (LIBNAM.LIB) для ввода, и задано имя файла вывода, назначенное пользователем (имя файла.BIN или имя файла.BIC). Расширение .BIC назначается для обеспечения цепочечного режима выполнения нескольких тестов комплекса.

В обоих случаях команда LINK вводит библиотеку (устр1), компонует модули, используя текущую таблицу компоновки, и выводит тест комплекса в формате загрузки на устройство вывода (устр0).

В следующих двух примерах показано использование команды LINK. Следует отметить, что если в течение работы команда LINK выводит заданный файл на устройство вывода, а там уже существует файл с таким же именем, то выводится следующее сообщение:

DELETE OLD?(Y <BK> OR JUST <BK>)
(исключить старый?)

Если вводится положительный ответ (Y <BK>), старый файл уничтожается, команда LINK начинает работу и выводит новый файл. Если ответ оператора отрицательный (<BK>), то старый файл не уничтожается, и команда LINK не выполняется. Выводится сообщение:

(?USE NEW FILE NAME)
(используйте новое имя файла)

и выводится звездочка "*".

Пример выполнения команды LINK
на устройстве с бесфайловой структурой

В примере перфолента с об'ектными модулями вводится с перфоленточного считывающего устройства (PR), тест комплекса выводится на перфоленту перфоратором (PP). В примере монитор компонуется с одним тестовым модулем WXYZ.

```
*LINK PP:<PR: <BK>                    ;формат команды LINK
SYS SIZE: 160000 <BK>                 ;размер оперативной памяти
                                      ;28к слов или больше
MAKE OUTPUT READY.WRITE ENABLE        ;подготовьте устройство
                                      ;вывода. Разрешите запись
TYPE (CR) WHEN READY <BK>             ;введите <BK>, когда
                                      ;готово
PASS 1                                ;фаза 1
ANYMORE MONITOR PAPER TYPES, CASSETTES, ETC.? (YES, NO)
                                      ;это лента мультипрограм-
                                      ;много монитора?(да, нет)
YES <BK>                              ;да
RELOAD INPUT WITH NEXT PAPER TAPE, CASSETTE, ETC
                                      ;вновь установите
                                      ;следующую ленту
TYPE <BK> WHEN READY <BK>             ;введите <BK>, когда
                                      ;готово
ANYMORE MONITOR PAPER TAPES, CASSETTES, ETC.? (YES, NO)
                                      ;это лента мультипрограм-
                                      ;много монитора?(да, нет)
NO <BK>                               ;нет
WXYZ SHOULD BE NEXT!                  ;модуль WXYZ должен быть
                                      ;следующим
TYPE (CR) WHEN READY <BK>             ;введите <BK>, когда
                                      ;готово
TRANSFER ADDRESS:002200               ;стартовый адрес теста
                                      ;комплекса
LOW LIMIT:000000                      ;начальный адрес теста
                                      ;комплекса
HIGH LIMIT:045302                     ;конечный адрес теста
                                      ;комплекса
PASS 2                                ;фаза 2
INPUT TAPES, CASSETTES, ETC.IN SAME SEQUENCE AS IN PASS1
                                      ;введите ленты в той же
```

```
                                    ;последовательности, как в
                                    ;фазе 1
TYPE (CR) WHEN READY <BK>           ;введите <BK>, когда
                                    ;готово
ANY MORE MONITOR PAPER TAPE, CASSETTES, ETC.? (YES, NO)
                                    ;это лента мультипрограм-
                                    ;много монитора?(да, нет)
YES <BK>                            ;да
RELOAD INPUT WITH NEXT PAPER TAPE, CASSETTE, ETC
                                    ;вновь установите
                                    ;следующую ленту
TYPE (CR) WHEN READY <BK>           ;введите <BK>, когда
                                    ;готово
WXYZ SHOULD BE NEXT!                ;модуль WXYZ должен быть
                                    ;следующим
TYPE (CR) WHEN READY <BK>           ;введите <BK>, когда
                                    ;готово
LINK DONE                           ;генерация завершена
```

Как только команда LINK введена, программа запрашивает необходимый размер памяти (т.е. фактический размер, в котором будет работать построенный тест комплекса).

В ответ оператор может ввести одно из следующих восьмеричных чисел:

Размер памяти в словах	Ввести:
4к	20000
8к	40000
12к	60000
16к	100000
20к	120000
24к	140000
28к	160000

Программа печатает PASS 1 (фаза 1) и начинает процесс генерации, в котором лента мультипрограммного монитора и тестовых модулей запрашиваются в той же последовательности, как и в таблице компоновки. В фазе 1 (PASS 1) программа неполностью считывает запрашиваемые ленты и определяет структуру теста комплекса. В фазе 2 (PASS 2) выполняется полное чтение лент, генерация и вывод сгенерированного теста комплекса.

Если команда LINK используется с одним из ключей /MP (вывод карты загрузки на консольный терминал) или /MLP (вывод на печатающее устройство), то адреса границ для теста комплекса (т.е. TRANSFER ADDRESS,LOW LIMIT,HIGH LIMIT) не печатаются в ходе фазы 1 (PASS 1).

3.5.3.1. Устройство с файловой структурой

В данном подпункте показан пример, в котором об'ектные модули выбираются с нулевого диска, компонуются в соответствии с таблицей компоновки. Затем выводится тест комплекса.

```
*LINK MY0:TEST.BIN<MY0:X?????.LIB  <BK>
                                    ;ввод команды LINK
SYS SIZE:160000 <BK>                ;память, необходимая для
                                    ;теста комплекса
MAKE OUTPUT READY.WRITE ENABLE      ;подготовьте устройство
                                    ;вывода. Разрешите запись
```

```
TYPE (CR) WHEN READY <BK>          ;введите <BK>, когда
                                   ;готово
PASS 1                             ;поиск всех модулей
TRANSFER ADDRESS:002200            ;стартовый адрес
LOW LIMIT:000000                   ;нижний адрес
HIGH LIMIT:063514                  ;верхний адрес
PASS 2                             ;вывод теста комплекса
LINK DONE                          ;процесс генерации
                                   ;завершен
```

Если команда LINK использовалась вместе с ключом /MP или ключом /MLP, то граничные адреса не будут распечатаны в фазе 1 (PASS 1).

3.5.4. Команды управления вводом-выводом используются в или вне режима конфигурации и позволяют:

1) сохранять и вновь вводить таблицу компоновки и карту загрузки;
2) управлять возвратом в монитор системы;
3) перезагружать монитор системы.

Команда TYPEC

Формат команды:

*TYPEC <BK>

Команда TYPEC используется для вывода содержимого текущей таблицы компоновки из оперативной памяти на консольный терминал.

П р и м е р :

*TYPEC <BK>

Команда PRINTC

Формат команды:

*PRINTC <BK>

Команда PRINTC используется для вывода содержимого текущей таблицы компоновки из оперативной памяти на печатающее устройство.

П р и м е р :

```
*PRINTC <BK>          ;вывод таблицы компоновки на
                      ;печатающее устройство.
```

Команда SAVC

Формат команды:

1) SAVC <УСТР.ВЫВ>:<ИМЯ ФАЙЛА.РАС> <BK>
2) SAVC <УСТР.ВЫВ.>: <BK>

Команда SAVC используется для сохранения копии текущей таблицы компоновки на заданном устройстве вывода.

Формат 1 команды SAVC используется для сохранения копии текущей таблицы компоновки на устройстве с файловой структурой (диск). Обязательным аргументом данного формата является имя файла. Если не указано устройство вывода в этом формате, то используется системное устройство (по умолчанию).

П р и м е р :

*SAVC MY0:CNF1.CNF <BK>

В данном примере таблица компоновки выводится на диск DK0 под заданным пользователем именем CNF1.CNF.

Если в процессе работы команды SAVC (формат 1) на указанном носителе уже существует файл с таким же именем, то выводится сообщение:

DELETE OLD (Y <BK> OR JUST <BK>)

(уничтожить старый файл?)

Если ответ положительный (Y <ВК>), то старый файл уничтожается, команда выполняется и выводится новый файл.

Если ответ отрицательный (<ВК>), то старый файл не уничтожается и команда не выполняется.

На консоль выводится сообщение:

?USE NEW FILE NAME

(используйте новое имя файла)

и печатается точка ".".

Формат 2 команды SAVC используется для сохранения копии текущей таблицы компоновки на устройстве с бесфайловой структурой (перфолента).

П р и м е р :

*SAVC PP: <ВК>

В данном примере таблица компоновки выводится через перфоратор (PP:) на перфоленту.

Команда GETC

Формат команды:

1) GETC <УСТР.ВВОДА>:<ИМЯ ФАЙЛА.РАС> <ВК>

2) GETC <УСТР.ВВОДА>: <ВК>

Команда GETC используется для ввода в оперативную память ранее записанной таблицы компоновки с указанного устройства для модификации или повторного использования.

Формат 1 команды GETC используется для восстановления копии таблицы компоновки с устройства с файловой структурой (диск). Обязательным аргументом для данного формата является имя файла. Если не указано устройство ввода, то используется системное устройство (по умолчанию).

П р и м е р :

*GETC MY0:CNF1.CNF <ВК>

В этом примере таблица компоновки размещена на диске MY0 под именем CNF1.CNF. По команде GETC таблица компоновки вызывается в оперативную память с MY0:.

Формат 2 команды GETC используется для восстановления копии таблицы компоновки с устройства с бесфайловой структурой (перфоввод).

П р и м е р :

*GETC PR: <ВК>

В этом примере копия таблицы компоновки вводится с перфоленты (PR:).

Команда SAVM

Формат команды:

1) SAVM <УСТР.ВЫВ>:<ИМЯ ФАЙЛА.РАС> <ВК>

2) SAVM <УСТР.ВЫВ>: <ВК>

Команда SAVM используется для сохранения карты загрузки, созданной командой LINK на носителе с файловой (диск) или бесфайловой (перфолента) структурой. Команда SAVM может быть введена непосредственно после сообщения "LINK DONE" (компоновка завершена).

Формат 1 команды SAVM используется для сохранения карты загрузки на устройстве с файловой структурой (диск и т.д.). Обязательным аргументом данного формата является имя файла. Если в команде SAVM (формат 1) не указано устройство вывода, то используется системное устройство (по умолчанию).

П р и м е р :

*SAVM MY1:LMP1.MAP <ВК>

В этом примере карта загрузки сохраняется на MY1 под именем LMP1.MAP.

Если в процессе работы команды SAVM на заданном устройстве уже существует файл с указанным именем, то выводится сообщение:

DELETE OLD? (Y <ВК> OR JUST <ВК>)
(уничтожить старый файл?)

Если ответ положительный (Y <ВК>), то старый файл уничтожается, команда выполняется и выводится новый файл. Если ответ отрицательный (<ВК>), то старый файл не уничтожается и команда не выполняется. Выводится сообщение:

?USE NEW FILE NAME
(используйте новое имя файла)

и печатается точка ".".

Формат 2 команды SAVM используется для сохранения карты загрузки на устройстве с бесфайловой структурой.

Пример:

*SAVM PP: <ВК>

В этом примере карта загрузки выводится на перфоленту.

Команда TYPEM

Формат команды:

1) TYPEM <УСТР.ВВОДА>:<ИМЯ ФАЙЛА.РАС> <ВК>
2) TYPEM <УСТР.ВВОДА>: <ВК>

Команда TYPEM используется для вывода на консоль карты загрузки ранее созданной на файловом (диск) или бесфайловом (перфолента) носителе.

Формат 1 команды TYPEM используется для вывода на консоль находящейся на носителе с файловой структурой копии карты загрузки.

Обязательным аргументом для этого формата является имя файла. Если в этом формате не указано устройство ввода, то используется системное устройство (по умолчанию).

Пример:

*TYPEM MY1:LMP1.MAP <ВК>

В этом примере карта загрузки находится на MY1 под именем LMP1.MAP, вызывается в оперативную память и выводится на консоль.

Формат 2 команды TYPEM используется для вывода на консоль карты загрузки, находящейся на носителе с бесфайловой структурой (перфолента).

Пример:

*TYPEM PR: <ВК>

В этом примере карта загрузки вводится с перфоленты в оперативную память (PR:) и выводится на консоль.

Команды PRINTM

Формат команды:

1) PRINTM <УСТР.ВВОДА>:<ИМЯ ФАЙЛА.РАС> <ВК>
2) PRINTM <USTR.ВВОДА>: <ВК>

Команда PRINTM используется для вывода карты загрузки на печатающее устройство с носителя с файловой или бесфайловой структурой.

Формат 1 команды PRINTMиспользуется для вывода на печатающее устройство карты загрузки, находящейся на носителе с файловой структурой. Обязательным аргументом для этого фор-

мата является имя файла. Если в формате 1 команды PRINTM не указано устройство ввода, то используется системное устройство (по умолчанию).

Пример:

*PRINTM MY0:LMP1.MAP <ВК>

В этом примере карта загрузки, находящаяся на MY0 под именем LMP1.MAP вызывается в оперативную память и выводится на печатающее устройство.

Формат 2 команды PRINTM используется для вывода на печатающее устройство карты загрузки, находящейся на носителе с бесфайловой структурой (перфолента).

Пример:

*PRINTM PR: <ВК>

В этом примере карта загрузки вводится с перфоленты в оперативную память через считывающее устройство (PR) и выводится на печатающее устройство.

Команда CHECK

Формат команды:

1) CHECK <УСТР.ВВОДА>:<ИМЯ ФАЙЛА.РАС> <ВК>
2) CHECK <УСТР.ВВОДА>: <ВК>

Команда CHECK используется для проверки правильности формата об'ектного файла и контрольной суммы.

Формат 1 команды CHECK используется для проверки формата об'ектного файла и контрольной суммы, находящихся на носителе с файловой структурой (диск), об'ектного файла и контрольной суммы. Обязательным аргументом в этом формате является имя файла. Если не указано устройство ввода, то используется системное устройство (по умолчанию).

Пример:

*CHECK MY0:XRKA60.OBJ <ВК>

В этом примере с MY0 в оперативную память вводится файл с именем XRKA60.OBJ для контроля.

Формат 2 команды CHECK используется для проверки формата об'ектного файла и контрольной суммы, находящегося на носителе с бесфайловой структурой (перфолента).

Пример:

*CHECK PR: <ВК>

В этом примере с перфоленты в оперативную память вводится модуль для контроля.

По мере выполнения проверки печатается имя каждого модуля, находящегося в файле.

Команда EXIT

Формат команды:

*EXIT <ВК>

Команда EXIT используется для выхода из программы конфигуратор/компоновщик и возврата в монитор ТМОС. Эта команда не убирает из оперативной памяти программу конфигуратор/компоновщик и не перезагружает монитор ТМОС.

Пример:

*EXIT <ВК>

В данном примере осуществляется возвращение к монитору ТМОС.

Команда BOOT

Формат команды:

BOOT <УСТР.ВВОДА>: <ВК>

Команда BOOT используется для загрузки монитора TMOC с указанного устройства.

П р и м е р :

```
*BOOT MT0:  <ВК>
```

В данном примере монитор TMOC загружается с магнитной ленты MT0:.

3.5.5. Команда MOD ADR - команда модификации

Данная команда может использоваться в или вне режима конфигурации для модификации заданных ячеек в программе.

Формат команды:

```
MOD ADR                 ;выводит содержимое заданной
                        ;ячейки
```

Пример демонстрирует использование команды модификации:

```
*MOD 4000 <ВК>          ;открывает ячейку 4000
программа отвечает:
004000/123456           ;содержимое ячейки 4000
                        ;величина 123456
```

Оператор:

1) закрывает ячейку 4000 нажатием <ВК>;
2) вводит новую величину и закрывает ячейку 4000 нажатием <ВК>;
3) вводит новую величину и открывает следующее слово, напечатав <ПС>;
4) закрывает ячейку 4000 и открывает следующее слово, напечатав <ПС>.

3.6. Сообщения об ошибках программы DXCL

Программа DXCL в процессе построения теста комплекса выявляет ошибки, допущенные пользователем, проверяет формат используемых тестовых модулей и т.д. и выводит на консоль сообщения об ошибках. В подразделе приводятся сообщения об ошибках с пояснениями.

```
?INVALID COMMAND                !с клавиатуры введена невер-
 неверная команда               !ная команда. Следует ввести
                                !исправленную команду;
?INVALID NAME                   !используется неверное имя в
 неверное имя                   !формате команды (специаль-
                                !ные символы запрещены).
                                !Следует указать исправлен-
                                !ное имя;
?NUMBER TOO BIG                 !в ответе на запрос програм-
 число слишком большое          !мы с клавиатуры введено
                                !значение параметра больше
                                !допустимого. Например,адрес
                                !вектора не может превышать
                                !774. Необходимо указать до-
                                !пустимое значение параметра
?INVALID SWITCH                 !указан недопустимый ключ
 неверный ключ                  !для данной команды, или ко-
                                !манда не может использо-
                                !ваться с ключем. Необходимо
                                !указать верный ключ;
?CHECKSUM ERROR                 !ошибка контрольной суммы
 ошибка контрольной суммы       !встретилась при чтении бло-
```

```
                                    !ка в двоичном формате.
                                    !Некорректируемая ошибка;
?FILENAMEXT?NON-EXISTANT FILE       !на носителе нет файла с
 несуществующий файл                !указанным в команде именем.
                                    !Необходимо указать имя
                                    !существующего файла;
?END-OF-MEDIUM                      !обнаружен физический конец
 конец носителя                     !носителя данных, или ненай-
                                    !ден блок в файле, так как
                                    !раньше был обнаружен приз-
                                    !нак "Конец файла";
?PROGRAM OVERLOW                    !размер файла на носителе
 переполнение                       !превысил размер буфера вво-
                                    !да;
?NOT IN CNF MODE                    !была попытка использовать
 не в CNF режиме                    !команды режима конфигурации
                                    !(DVA, VCT, BR1 и др.) не в
                                    !режиме конфигурации. Для
                                    !того, чтобы войти в режим
                                    !конфигурации, используйте
                                    !команду CNF <BK>;
?MUST BE OCTAL                      !с клавиатуры на запрос
 должно быть в восьмеричном         !программы введено не
 коде                               !восьмеричное число;
?NO ROOM FOR A DRIVER               !драйвер устройства, задан-
 нет места для драйвера             !ного в команде не вмещается
                                    !в буфер драйвера;
?CNF TABLE FULL                     !число входов в таблице ком-
 CNF таблица переполнена            !поновки превысило допусти-
                                    !мое (40);
?COR EXCD                           !созданный в результате ком-
 превышен размер памяти             !поновки тест комплекса пре-
                                    !вышает размер оперативной
                                    !памяти. Следует уменьшить
                                    !количество тестовых модулей
                                    !при компоновке;
?SYMBOL TABLE OVERFLOW              !на первом шаге компоновки
 таблица символов                   !переполнена таблица симво-
 переполнена                        !лов из-за недостаточного
                                    !размера оперативной памяти.
                                    !Необходимо использовать вы-
                                    !числительный комплекс с
                                    !большим размером оператив-
                                    !ной памяти;
?USE NEW FILE NAME                  !файл с указанным в команде
 используйте новое имя файла        !именем уже существует на
                                    !носителе. Необходимо ис-
                                    !пользовать новое имя файла
                                    !или удалить старый файл;
?DEVICE FULL                        !на указанном в команде уст-
 устройство переполнено             !ройстве нет места для раз-
                                    !мещения файла. Нужно иск-
                                    !чить ненужные файлы на уст-
                                    !ройстве;
?RD ERROR                           !ошибка чтения файла или
```

ошибка чтения	!блока с устройства, указан- !ного в команде;
?WT ERROR ошибка записи	!ошибка возникает при попыт- !ке записать файл на указан- !ное в команде устройство !вывода. Нужно проверить !разрешена ли запись на уст- !ройстве вывода;
ERR01 ошибка в таблице символов	!ошибка в таблице символов. !Обнаружена в процессе ком- !поновки;
ERR02 ошибка поиска в RLD	!не найден символ в блоке !информации перемещаемых ло- !кальных символов;
ERR03 нет PC модифицируемой команды в RLD	!блок информации перемеща- !мых локальных символов не !содержит счетчика инструк- !ций для модифицируемой ко- !манды;
ERR04 GSD блок отсутствует	!об'ектный модуль не начина- !ется с блока, содержащего !информацию о глобальных !символах (GSD);
ERR05 отсутствует имя модуля в GSD	!первый вход в GSD не явля- !ется именем об'ектного мо- !дуля;
ERR06 отсутствует имя секции	!отсутствует имя секции в !каталоге перемещаемых сим- !волов (RLD);
ERR07 нет имени модуля в таблице символов	!имя модуля отсутствует в !таблице символов. Вероятная !причина - неверно указано !имя модуля. Имя модуля не !соответствует имени файла;
ERR09 ошибка указателя таблицы переходов	!величина указателя таблицы !переходов превышает допус- !тимое значение (то есть код !байта CSD слишком большой). !Это программная ошибка;
ERR12 ошибка в модуле загрузочного формата	!ошибка обнаружена при запи- !си на устройство вывода, !скомпонованного тестом ком- !плекса в формате абсолютной !загрузки.

3.7. Генерация теста комплекса

Рассматривается краткое содержание процессов конфигурации и компоновки с точки зрения:

1) Построения и/или модификации таблицы компоновки;
2) Выполнения команды компоновки;
3) Генерации теста комплекса.

Текст содержит ссылки на команды компоновки, но не предусматривает подробного описания их формата и употребления.

3.7.1. Таблица компоновки (C-таблица)

Таблица компоновки используется для облегчения процесса компоновки с помощью команды LINK. Таблица компоновки обслуживает максимально 40 входов (т.е. 39 выбранных модулей и один мультипрограммный монитор), для каждого из которых отводится 11 служебных слов.

Построение таблицы компоновки может начинаться, когда загружена программа конфигуратор/компоновщик и установлен режим конфигурации с помощью команды CNF <ВК>. Формат расположения модуля в таблице конфигурации приведен на рис.4.

ФОРМАТ ВХОДА МОДУЛЯ В ТАБЛИЦЕ КОНФИГУРАЦИИ

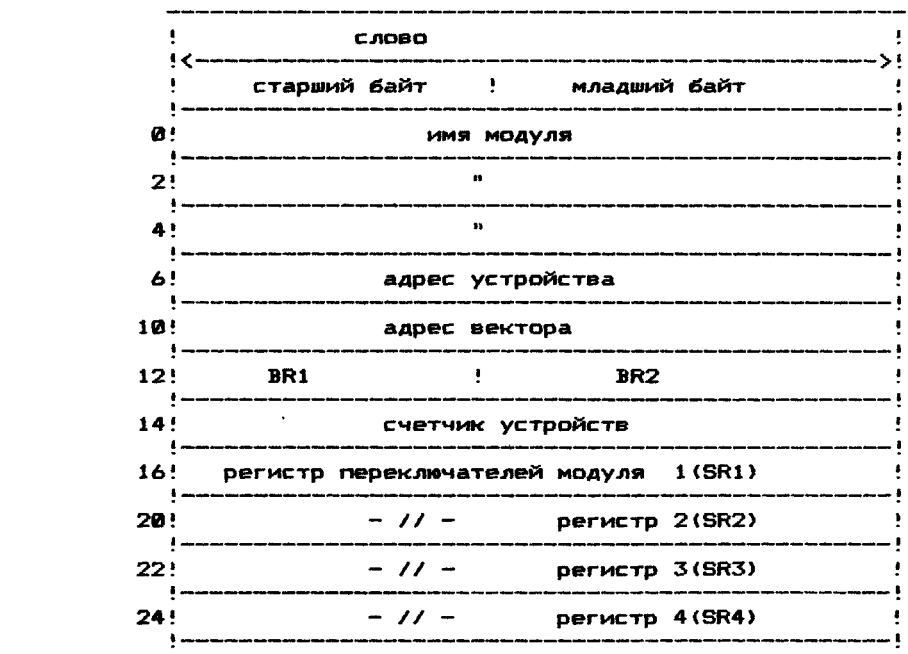


рис. 4

Если к моменту ввода команды CNF таблица компоновки пустая, программа запрашивает имя мультипрограммного монитора, и пользователь может создавать новую таблицу. Если таблица компоновки уже создана, (предполагается, что пользователь собирается модифицировать существующие входы), то имя мультипрограммного монитора не будет запрашиваться. Следовательно, если таблица компоновки непустая и необходимо создать новую, она должна быть предварительно очищена.

Для очистки таблицы компоновки необходимов вести с клавиатуры команду CL <ВК>.

3.7.1.1. Указание текущего входа в таблице компоновки

В процессе построения таблицы компоновки программа DXCL устанавливает указатель таблицы, помечающий группу данных, относящихся к модулю, включенному в таблицу. Положение указателя входа может изменяться.

Указатель текущего входа соответствует положению указателя после ввода очередной команды MDL. Таким образом, параметры команды MDL (DVA, DVC, и др.) относятся к текущему входу, то есть к модулю, имя которого указано в команде MDL. Команды NXT и POINT могут изменять положение указателя текущего входа, что облегчает корректировку таблицы компоновки. Содержимое текущего входа выводится на консольный терминал, позволяя ознакомиться с назначенными для модуля параметрами. Команда NXT устанавливает указатель на следующий вход, если он существует. Команда POINT устанавливает указатель на вход с именем, указанным в команде.

3.7.1.2. Вывод и хранение таблицы компоновки

Команды TYPEC и PRINTC используются для вывода содержимого таблицы компоновки на системный терминал и соответственно на печатающее устройство. Эти команды могут использоваться в любом режиме работы программы DXCL. Таблица компоновки может быть сохранена (выведена) на устройство вывода с указанием имени файла по команде SAVC и затем вызвана в оперативную память по команде GETC. Эти команды выполняются в любом режиме.

3.8. Процесс компоновки (команда LINK)

Команда LINK, используя построенную таблицу компоновки, об'ектные модули мультипрограммного монитора и тестовые модули выполняет компоновку теста комплекса. Результат компоновки - тест комплекса в абсолютном двоичном формате, пригодный для последующей загрузки в оперативную память и выполнения.

Процесс компоновки состоит из двух фаз. В течение фазы1 (PASS 1) в память считывается только часть каждого тестового модуля и мультипрограммного монитора для анализа и присвоения значения глобальным ссылкам. В течение второй фазы (PASS 2) в память считывается оставшаяся часть каждого модуля, присваиваются абсолютные адреса, и создается тест комплекса в формате абсолютной загрузки. Созданный тест комплекса переписывается поблочно на устройство вывода.

3.8.1. Выполнение фазы 1 компоновки

Когда введена команда LINK, начинается выполнение фазы 1. Выполнение этой фазы сопровождается выводом на экран терминала сообщения PASS 1. В течение фазы 1 DXCL анализирует таблицу компоновки, чтобы определить,какой мультипрограммный монитор был затребован пользователем.

DXCL находит библиотеку мониторных модулей и определяет, какие мониторные модули необходимы для построения указанного мультипрограммного монитора. Как только мониторные модули будут считаны, фаза 1 выполняется для каждого из них.

Если библиотека мониторных модулей находится на устройстве с каталогом (магнитный диск, магнитная лента), DXCL находит указанные тестовые модули.

Таким образом, когда мониторные модули обработаны, DXCL вновь обращается к таблице компоновки за именами тестовых модулей, чтобы обработать их.

Важно отметить, что в фазе 1 считывается небольшая

часть модуля.

Когда фаза 1 завершена, на консольном терминале (/MP) печатается адресный ряд модулей в тесте комплекса, и начинается выполнение фазы 2.

3.8.2. Выполнение фазы 2 компоновки

Начало выполнения фазы 2 сопровождается выводом на консоль сообщения PASS 2. Используя информацию, содержащуюся в таблице компоновки и полученную при выполнении фазы 1 из первого просмотра модулей, в фазе 2 окончательно определяется структура теста комплекса, и он выводится на устройство вывода.

В начале фазы 2 в память из библиотеки мультипрограммного монитора блок за блоком передается каждый из мониторных модулей и затем блок за блоком после обработки выводится на устройство вывода как часть теста комплекса. Этот процесс будет продолжен для всех необходимых мониторных модулей.

Обработка тестовых модулей выполняется аналогично.

Во время обработки обработанный модуль выводится блок за блоком на устройство вывода как часть теста комплекса. По окончании обработки и вывода на консольный терминал выводится сообщение LINK DONE (компоновка завершена).

Пример генерации теста комплекса приведен в приложении 3.

3.9. Тест комплекса, полученный в результате компоновки, представляет собой именованный двоичный файл в формате абсолютной загрузки. Имя теста может содержать до 6 символов и иметь расширение BIN или BIC. Расширение BIC используется для включения теста комплекса в цепочку, выполняемую под управлением монитора системы.

С носителя тест комплекса может быть вызван в оперативную память монитором системы или другими программами, способными загружать с носителя программу в абсолютном загрузочном формате.

3.9.1. Загрузка и запуск теста комплекса

В зависимости от того, на каком носителе находится тест комплекса, для его загрузки и запуска используется монитор системы, обслуживающий этот вид носителя.

Когда тест комплекса загружен в оперативную память, он может быть запущен с адреса 200(восьмеричное). Адрес повторного запуска теста комплекса 1000(восьмеричное).

При запуске тест комплекса определяет размер памяти в проверяемом вычислительном комплексе и другие характерные особенности оборудования.

3.9.1.1. Загрузка через монитор системы

Если сгенерированный тест комплекса записан на носитель, где располагается TMOC, он может быть загружен и запущен с помощью монитора системы, поддерживающего этот носитель.

Пример:

```
.L KTST1  <ВК>          ;загрузка программы, запуск
.S <ВК>                 ;командой S<ВК>
.R KTST1  <ВК>          ;загрузка программы и запуск
```

```
                                        ;ее с адреса 0200
```

Отметим, что загрузку теста комплекса всегда лучше проводить через монитор системы. Только монитор системы передает тесту комплекса параметры, определяющие имя устройства, с которого была выполнена загрузка. Тест комплекса, используя эту информацию, защищает носитель ТМОС от случайного разрушения, если будет предпринята попытка выполнить тестирование этого носителя.

3.9.2. Управление тестом комплекса

Для внешнего управления выполнением теста комплекса используются 22 типа клавиатурных команд.

Все команды могут вводиться в командном режиме (CMD>). Некоторые из них могут быть также введены в режиме выполнения (BSY>). Некоторые команды (RUN, MOD, и т.д.) могут быть введены только в командном режиме. Управление выполнением теста комплекса выполняется с помощью программного регистра переключателей (SWR).

Разряды программного регистра переключателей управляют следующими функциями:

```
SR00=0          !запрещает распечатку единичного знака (еди-
                !ничной характеристики) "NULL" сообщение
                !("пусто");
SR00=1          !разрешает распечатку единичной характеристики
                !"NULL";
SR08=0          !разрешает перемещение теста комплекса по всей
                !памяти, включая перемещение на случайную
                !величину;
SR08=1          !разрешает перемещение теста комплекса по па-
                !мяти с постоянным шагом. Запрещает случайное
                !перемещение;
SR09=0          !разрешает вывод сообщения "RELOCATED TO";
SR09=1          !запрещает вывод сообщения "RELOCATED TO";
SR10=0          !сообщается только о первых трех ошибках дан-
                !ных, встречающихся в передаваемом блоке;
SR10=1          !сообщаются все ошибки данных;
SR12=0          !запрещает вывод сообщения "END OF PASS";
SR12=1          !разрешает вывод сообщения "END OF PASS";
SR13=1          !запрещает печать ошибок и сообщений модуля;
SR14=0          !после 20-й ошибки модуль снимается с выпол-
                !нения выводится сообщение "MODULE DROPPED"
                !(модуль снят;
SR14=1          !модуль не снимается с выполнения после 20-ти
                !ошибок;
SR15=1          !снимает с выполнения модуль после одной ошиб-
                !ки и печатает сообщение "MODULE DROPPED"
                !(модуль снят).
```

3.9.3. Классификация и характеристика клавиатурных команд мультипрограммного монитора

3.9.3.1. Команды, разрешенные в командном режиме (SMD>) и режиме выполнения (BSY>) (см. п.п. 2.7.1.)

```
MAP             !вывод справочной таблицы для всех тесто-
                !вых модулей в тесте комплекса;
```

```
MAP <ИМЯ МОДУЛЯ>   !вывод справочной таблицы указанного
                   !тестового модуля;
SEL                !выбор всех модулей для выполнения;
SEL <ИМЯ МОДУЛЯ>   !выбор модуля с указанным именем выполне-
                   !ния;
DES                !запрещение выполнения всех модулей;
DES <ИМЯ МОДУЛЯ>   !запрещение выполнения указанного модуля;
PON                !разрешить контроль информации при обраще-
                   !нии к памяти;
POFF               !запретить контроль информации при обраще-
                   !нии к памяти;
ROTON              !разрешение циклического перемещения буфе-
                   !ра записи;
ROTOFF             !запрещение циклического перемещения буфе-
                   !ра записи;
LPON               !вывод сообщения на устройство печати и
                   !консольный терминал;
LPOFF              !запрещение вывода всех консольных сообще-
                   !ний на устройство печати;
CON                !разрешение использования ББП;
COFF               !запрещение использования ББП;
EXAM               !вывод содержимого ячейки ранее открытой
                   !командой EXAM <АДР>;
EXAM <АДР>         !вывод содержимого указанной ячейки;
EXAM <ИМЯ МОДУЛЯ>  !вывод содержимого ячейки в указанном мо-
                   !дуле <АДР>. Адрес указывает смещение
                   !ячейки относительно начала модуля;
SUM                !вывод итогового сообщения выполнения тес-
                   !та комплекса для всех модулей;
SUM <ИМЯ МОДУЛЯ>   !вывод итогового сообщения для модуля с
                   !указанным именем;
SWR                !вывод содержимого программного регистра
                   !переключателей;
SWR <ЧИСЛО>        !замена содержимого программного регистра
                   !переключателей.
```

3.9.3.2. Команды, разрешенные в командном режиме (CMD>)

```
RUN                !запуск теста комплекса;
RUN <АДР>          !запуск теста комплекса с предваритель-
                   !ным перемещением на указанный адрес;
RUNL               !запуск теста комплекса с запретом
                   !перемещения по памяти;
RUNL <АДР>         !перемещение теста комплекса по указан-
                   !ному адресу, запрет дальнейших переме-
                   !щений и запуск;
MOD                !открывает содержимое последней модифи-
                   !цированной ячейки;
MOD <АДР>          !вывод содержимого ячейки с указанным
                   !адресом;
MOD <ИМЯ МОДУЛЯ><АДР>!вывод содержимого ячейки указанного
                   !модуля. Адрес указывает смещение ячей-
                   !ки относительно начала модуля;
KTON               !разрешение использования диспетчера
                   !памяти;
KTOFF              !запрещение использования диспетчера
```

```
                    !памяти;
MON                 !разрешение использования 22-разрядной
                    !адресации общей шины;
MOFF                !запрещение использования 22-разрядной
                    !адресации общей шины.
```

3.10. Клавиатурные команды мультипрограммного монитора

Для обслуживания выполнения теста комплекса используются клавиатурные команды.

Команды вводятся с клавиатуры. Они интерпретируются и выполняются мультипрограммным монитором. Неверно выведенные командные строки отвергаются и сопровождаются сообщением об ошибке.

3.10.1. Используемые символы для командных строк

При вводе клавиатурных команд могут использоваться алфавитные (от A до Z) и цифровые (от 0 до 9) символы клавиатуры.

Дополнительно используются следующие символы: "Пробел", ",", "ПС", "ВК", "ЗБ", "СУ/C", "СУ/U", "СУ/O" для форматирования, редактирования командных строк и управления ими.

Пробел (код 040)

- ввод с клавиатуры символа "Пробел" передвигает маркер экрана на один символ вправо;

Перевод строки (<ПС> код 012)

- ввод с клавиатуры символа "ПС" переводит маркер экрана на следующую строку;

Возврат каретки (<ВК> код 015)

- ввод с клавиатуры символа "ВК" передвигает маркер экрана в начало следующей строки;

Стирание (<ЗБ> код 177)

- ввод с клавиатуры символа "ЗБ" исключает последний введенный символ текущей строки. Ввод с клавиатуры символа "ЗБ" N раз исключает N последних введенных символов;

<СУ>/<C> (код 003)

- ввод с клавиатуры <СУ>/<C> прекращает выполнение теста комплекса и возвращает его в командный режим (CMD>);

<СУ>/<U> (код 025)

- ввод с клавиатуры <СУ>/<U> отменяет введенную командную строку, не изменяя режим работы, переводит маркер экрана в начало следующей строки и позволяет вновь ввести командную строку;

<СУ>/<O> (код 017)

- ввод с клавиатуры <СУ>/<O> подавляет вывод текущего сообщения на экран терминала.

3.10.2. Сообщения об ошибках при вводе команд с клавиатуры

Существует восемь основных сообщений об ошибках при вводе с клавиатуры и четыре добавочных сообщения, которые могут появляться только во время использования команды RUN и RUNL. Эти сообщения об ошибках выводятся на консольный терминал.

Основные сообщения об ошибках с пояснениями:

INVALID ADDRESS неверный адрес	!введен несуществующий адрес. Адрес больше 1 бит или он не разрешен мультипрограммным монитором;
INVALID COMMAND неверная команда	!используется недопустимая команда. Сообщение сопровождается печатью неверно выведенной команды (INVALID COMMAND-MAPP);
INVALID COMMAND IN RUN MODE неверная команда в режиме выполнения	!введена недопустимая команда в режиме выполнения (BSY>), например RUN, RUNL, MOD и другие, которые должны вводиться только в командном режиме (CMD>);
INVALID MODULE NAME неверное имя модуля	!введенное имя модуля содержит менее или более пяти символов, или оно не распознается мультипрограммным монитором;
INVALID OR MISSING ARGUMENT неверный или отсутствующий аргумент	!указан неверный аргумент или он отсутствует (например, (MOD<ИМЯ МОДУЛЯ> с отсутствующим адресом)
MUST BE EVEN ADDRESS адрес должен быть четным	!введен нечетный адрес для адресного аргумента;
NOT AN OCTAL NUMBER не восьмеричное число	!введен не восьмеричный аргумент т.е. был употреблен символ, запрещенный для представления числа в восьмеричном коде;
NUMBER TOO LARGE очень большое число	!указано числовое значение вводимого аргумента, превышающее максимально допустимые 16 бит (т.е. восьмеричное 177777);

Сообщения об ошибках команд RUN и RUNL:

ADDRESS-CK-BUT-EXEROISER-WON'T адрес верный, но нет места	!не достаточно места для размещения теста комплекса между указанным в команде адресом и максимальным адресом оперативной памяти;
MUST-HAVE-KT-ON должен быть разрешен диспетчер памяти	!адресный аргумент, указанный в команде, требует использования диспетчера памяти (ранее была использована команда KTOFF);
NO-MODULES-SELECTED нет выбранных модулей	!команда введена, а выбранных для выполнения модулей нет;
MAP BOX MUST BE ON должно быть разрешено использование оборудования для 22-разрядной адресации общей шины	!задан адресный аргумент больше 96K (600000) и 22-битная адресация запрещена (ранее была использована команда MOFF).

3.10.3. Описание клавиатурных команд

Клавиатурные команды мультипрограммного монитора предназначены для управления выполнением теста комплекса.

Команда COFF

Формат команды: COFF <ВК>

Команда COFF запрещает использование ББП. Когда тест комплекса запускается, использование ББП разрешено.С помощью команды COFF запрещается использование ББП. Использование вновь разрешается командой CON.

Команда CON

Формат команды: CON <ВК>

Команда CON используется для повторного разрешения ББП.

Когда тест комплекса запускается, использование ББП разрешено. С помощью команды COFF запрещается использование ББП и вновь разрешается с помощью команды CON.

Команда DES

Формат команды: DES <ВК>
DES <ИМЯ МОДУЛЯ> <ВК>

Команда DES запрещает выполнение одного или всех модулей.

Когда тест комплекса загружается, все модули выбираются для выполнения. Пользователь может запретить выполнение одного или всех модулей с помощью команды DES. Команда DES, обычно, используется совместно с командой SEL, разрешающей выполнение указанного или всех модулей.

П р и м е р 1 :

Разрешено выполнение всех модулей, кроме одного.

```
SEL  <ВК>
DES <ИМЯ МОДУЛЯ> <ВК>
```

П р и м е р 2 :

Запрещено выполнение всех модулей, кроме одного.

```
DES  <ВК>
SEL <ИМЯ МОДУЛЯ> <ВК>
```

ПРИМЕЧАНИЕ. Аргумент имени модуля должен состоять из пяти символов (CPA60).

Команда EXAM

Формат команды:

1) EXAM <ИМЯ МОДУЛЯ> <АДРЕС> <ВК>
2) EXAM <АДРЕС> <ВК>
3) EXAM <ВК>

Команда EXAM осуществляет контроль содержимого ячейки во время работы теста комплекса в режиме выполнения (BSY>).

Адрес этой ячейки определяется форматом вводимой команды.

Использование формата 3 позволяет выводить содержимое ранее открытой ячейки командой EXAM <АДРЕС>.

Формат 2 позволяет выводить содержимое ячейки, виртуальный адрес которой указан в команде.

В формате 1 выводится содержимое ячейки модуля, указанного в команде.

Аргумент адреса, указанный в формате 1, содержит величину смещения ячейки относительно начала модуля.

ПРИМЕЧАНИЕ. Максимальный размер адресного аргумента 16 бит.

П р и м е р ы :

Формат 3:

```
EXAM 053772  <ВК>
053772/001000
EXAM  <ВК>
053772/002345                    ;содержимое изменилось в
```

```
                                        ;процессе работы
 Формат 2:
EXAM 053776  <ВК>
053776/00005
 Формат 1:
EXAM LPAE0 36  <ВК>             ;выводит 36-е слово
053774/0000004                  ;модуля LPAE0
```

Команды KTOFF и KTON

```
Формат команд:
  KTOFF  <ВК>
  KTON   <ВК>
```

В командном режиме (CMD>) пользователь может запретить использование диспетчера памяти и вновь разрешить его использование.

Когда тест комплекса запускается, использование диспетчера памяти разрешено и указатель его состояния (KTSTAT) установлен.

Команда KTOFF запрещает использование диспетчера памяти и очищает указатель его состояния (KTSTAT).

Команда KTON разрешает использование диспетчера памяти, если он был запрещен командой KTOFF.

ПРИМЕЧАНИЕ. Команды KTON и KTOFF могут быть введены только в командном режиме.

Команды LPOFF и LPON

```
Формат команд:
 LPOFF  <ВК>
 LPON   <ВК>
```

В процессе работы пользователь может разрешить или запретить вывод всех сообщений на устройство печати.

Команда LPOFF запрещает вывод всех последующих сообщений на устройство печати, а разрешает вывод только на консольный терминал.

Команда LPON разрешает вывод всех сообщений на устройство печати.

Команда MAP

```
  Формат команды:
 1) MAP <ИМЯ МОДУЛЯ>  <ВК>
 2) MAP  <ВК>
```

В процессе работы на печатающее устройство или терминал может быть выведена справочная таблица о текущем состоянии всех тестовых модулей в тесте комплекса или одного указанного модуля.

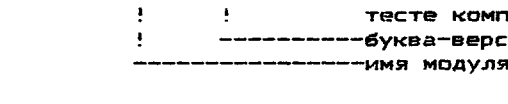
Для этой цели используется команда MAP.

Каждая строка справочной таблицы имеет вид:

<ИМЯ МОДУЛЯ> AT VA:<АДРЕС> STAT <СЛОВО СОСТОЯНИЯ МОДУЛЯ>

где <ИМЯ МОДУЛЯ>

- состоит из пяти символов, каждый из которых определяет:



```
AT VA: <АДРЕС>
   - виртуальный адрес, определяющий первое слово в модуле
     (т.е. нулевое слово интерфейса модуля);
STAT <СЛОВО СОСТОЯНИЯ МОДУЛЯ>
   - за исключением  битов 11, 13 и 14 все биты 16-битного
     слова  состояния (00-15) используются для определения
     типа модуля (т.е. фоновый, ввода-вывода и т.д.). Биты
     11,  13  и  14  используются для определения текущего
     состояния модуля (т.е. выбран, снят, выполняется).
    П р и м е р :
      Формат 2:
      MAP  <ВК>
    Монитор выводит таблицу тестовых модулей:
  RKAD0 AT VA:021544 STAT:150000      ;IOMODX модуль
                                      ;RKAD0 выбран
  TCAD0 AT VA:034700 STAT:130000      ;IOMODX модуль
                                      ;TCAD0 выбран
  CPAD0 AT VA:042346 STAT:40020       ;BKMOD модуль
                                      ;CPAD0 выбран
      Формат 1:
     MAP TAAC0 <ВК>                   ;справочная таблица
                                      ;модуля TAAC0
    Монитор выводит сообщение:
  TAAC0 AT VA:037460 STAT:140000      ;IOMOD модуль TAAC0
                                      ;выбран
                          Команда MOD
     Формат команды:
        1) MOD <ИМЯ МОДУЛЯ> <АДРЕС>  <ВК>
        2) MOD <АДРЕС>  <ВК>
        3) MOD  <ВК>
```

Команда MOD используется для проверки и/или модификации содержимого выбранных ячеек памяти.

Команда MOD дает возможность выводить и/или модифицировать содержимое ячеек с абсолютными и относительными адресами (т.е. относительно начального адреса указанного модуля). Когда задаются относительные адреса, мультипрограммный монитор отвечает печатью эквивалентных абсолютных адресов.

П Р И М Е Ч А Н И Я :

1. Команда MOD вводится только в командном режиме (CMD>).
2. Все указанные адреса должны быть меньше 32K.
3. Все указанные адреса должны быть четные.

```
    П р и м е р ы :
      Формат 3:
      MOD  <ВК>              ;выводится последняя модифициро-
                             ;ванная ячейка 002000/000254
      Формат 2:
    MOD 4000  <ВК>           ;выводится содержимое
                             ;ячейки 4000
    Монитор отвечает:
      004000/123456
```

Возможные действия оператора:

1) Закрыть ячейку 4000, нажав <ВК>;
2) Ввести новое число и закрыть ячейку, нажав <ВК>;
3) Ввести новое число и открыть следующее слово, нажав <ПС>.

```
      Формат 1:
```

```
MOD DCAA0 20  <BK>        ;открывает относительную
                          ;ячейку 20 в модуле DCAA0
Монитор отвечает:
  012020/140000           ;абсолютный адрес относительной
                          ;ячейки - 012020, содержимое
                          ;этой ячейки 140000
```

Действия оператора аналогичны действиям оператора, приведенным в примере формата 2.

Команды MOFF и MON

Формат команд:

```
  MOFF  <BK>
  MON   <BK>
```

В процессе работы пользователь может запретить или разрешить использование 22-разрядной системной магистрали.

Когда тест комплекса запускается, использование 22-разрядной адресации системной магистрали разрешено. Оно может быть запрещено командой MOFF и вновь разрешено командой MON.

Команды MON и MOFF могут использоваться только в командном режиме (SMD>).

Команды POFF и PON

Формат команд:

```
  POFF  <BK>
  PON   <BK>
```

Во время запуска теста разрешается использование паритетного контроля памяти. Использование паритетного контроля памяти может быть запрещено командой POFF и вновь разрешено командой PON.

Команды ROTOFF и ROTON

Формат команд:

```
 ROTOFF  <BK>
 ROTON   <BK>
```

Во время запуска теста допускается запрещение или разрешение циклического перемещения буфера записи.

Команда ROTOFF запрещает, а команда ROTON вновь разрешает циклическое перемещение этого буфера. По умолчанию циклическое перемещение буфера записи разрешено.

Команды RUN и RUNL

Формат команд:

```
1) RUN <АДРЕС>  <BK>        1) RUNL <АДРЕС> <BK>
2) RUN <BK>                 2) RUNL <BK>
```

Команда RUN используется для входа в режим выполнения (BSY>) и запуска теста комплекса с предварительным перемещением на указанный адрес или без перемещения.

Команда RUN позволяет циклическое перемещение программы теста комплекса (неперемещаемая часть теста комплекса всегда размещается в нижних 4К памяти в диапазоне 0-17776(8) и никогда не перемещается), если для перемещения имеется оперативная память необходимых размеров, и использование диспетчера памяти разрешено (KTON).

По команде RUN вводится режим выполнения (BSY>) и затем выбираются и запускаются модули в следующем порядке:

1) Запускаются специальные фоновые тестовые модули (SBKMOD);
2) Запускаются фоновые одиночные тестовые модули (NBKMOD);
3) Выполняется первый итерационный шаг фоновых тестовых модулей (BKMOD);

4) Запускаются тестовые модули ввода-вывода (IOMOD, X, P, AND R);
5) В заключении запускаются фоновые тестовые модули (BKMOD).

Во время выполнения теста комплекса осуществляется циклическое перемещение буфера записи. Оно разрешается при запуске теста комплекса, может быть запрещено командой ROTOFF и вновь разрешено командой ROTON.

Предварительное перемещение теста комплекса по указанному пользователем (формат 1) адресу осуществляется, если позволяет размер оперативной памяти, диспетчер памяти существует и разрешено его использование (по умолчанию или командой KTON).

Размер оперативной памяти с указанного адреса до конца должен быть достаточным для размещения перемещаемой части теста комплекса.

Перемещение в область с указанным адресом выполняется до запуска тестовых модулей.

После запуска теста комплекса, указанные тестовые модули выполняются. Тестовые модули снимаются с выполнения в следующих случаях:

1) Введен <СУ>/<С> - мультипрограммный монитор прекращает выполнение тестовых модулей, возвращает перемещаемую часть теста комплекса на исходную позицию в памяти и переводит тест комплекса в командный режим (CMD>);
2) После обнаружения ошибок модулем снимаются с выполнения все тестовые модули. Мультипрограммный монитор возвращает перемещаемую часть теста комплекса в исходную область памяти (если необходимо) и возвращает систему в командный режим (CMD>);
3) В случае фатальной ошибки (обнаружено много системных ошибок) мультипрограммный монитор прекращает выполнение тестовых модулей, возвращает перемещаемую часть теста комплекса в исходную область памяти (если необходимо) и возвращает систему в командный режим (CMD>).

Команда RUNL аналогична команде RUN, но в отличие от команды RUN, запрещает циклическое перемещение теста комплекса в оперативной памяти.

По этой команде (формат 1) устанавливается режим выполнения (BSY>). Перемещаемая часть теста комплекса перемещается по адресу, указанному пользователем и запускается. Дальнейшие перемещения запрещаются.

Формат 2 команды RUNL устанавливает режим выполнения (BSY>) и запускает тест комплекса без предварительного и последующего перемещения.

Перемещаемая часть теста комплекса может быть предварительно перемещена по адресу, указанному пользователем (формат 1), если имеется достаточная оперативная память и диспетчер памяти существует и разрешен (по умолчанию или командой KTON). Последующее перемещение теста комплекса запрещено.

Размер оперативной памяти с указанного адреса до конца должен быть достаточным для размещения перемещаемой части теста комплекса.

Перемещение в область с указанным адресом выполняется до запуска тестовых модулей.

После загрузки теста комплекса указанные модули выпол-

няются. Тестовые модули снимаются с выполнения аналогично команде RUN.

Для получения <итога работы> тестовых модулей используется команда SUM.

Примеры:

Формат 2:

```
RUNL  <ВК>              ;тест комплекса запускается без
                        ;предварительного перемещения с
                        ;запретом последующего переме-
                        ;щения
```

Формат 1:

```
RUNL 360000  <ВК>       ;тест комплекса размещается с
                        ;адреса 360000 и запускается.
                        ;Последующее перемещение запре-
                        ;щается
```

Монитор отвечает:

```
RELOCATED TO 360000
(перемещен в 360000)
```

Формат 2:

```
RUN  <ВК>               ;тест комплекса запускается
                        ;без предварительного пере-
                        ;мещения
```

Формат 1:

```
RUN 360000  <ВК>        ;тест комплекса размещается
                        ;с адреса 360000 и запуска-
                        ;ется
```

Монитор отвечает:

```
RELOCATED TO 360000
(перемещен в 360000)
```

Команда SEL

Формат команды:

1) SEL <ИМЯ МОДУЛЯ> <ВК>
2) SEL <ВК>

Команда SEL используется для выборки указанного тестового модуля или для выборки всех модулей.

Когда тест комплекса запускается, то по умолчанию все модули назначены для выполнения. По желанию пользователь может запретить выполнение одного или всех модулей с помощью команды DES. Поэтому команда SEL, в основном, всегда используется вместе с командой DES.

Пример:

```
DES                     ;запрещает выполнение всех
                        ;модулей
SEL <ИМЯ МОДУЛЯ>        ;выбирает модуль по имени
```

ПРИМЕЧАНИЕ. Аргумент имени модуля должен состоять из пяти символов (CPAG0).

Пример:

Формат 1:

```
SEL  <ВК>               ;выбираются для выполнения
                        ;все модули
```

Формат 2:

```
SEL DCAA 0 <ВК>         ;для выполнения выбирается
                        ;модуль DCAA0
```

Команда SUM

Формат команды:

1) SUM <ИМЯ МОДУЛЯ> <ВК>
2) SUM <ВК>

Команда SUM используется для вывода итогового сообщения для всех модулей или указанного модуля в следующем порядке: имя модуля, текущее состояние модуля, десятичное число шагов, ошибки оборудования, программные ошибки, системные ошибки, сбои питания.

Если указан только один модуль, последние два сообщения не выводятся.

Команда SUM вводится в режиме выполнения (BSY>).

Итоговое сообщение выводится в следующем формате:

<ИМЯ МОДУЛЯ> AT VA:<АДРЕС> STAT (СЛОВО СОСТОЯНИЯ)
PASS (ЧИСЛО) HRDERRS (ЧИСЛО) SFTERRS (ЧИСЛО)
SYSTEM ERROR (ЧИСЛО) POWER FAILS (ЧИСЛО)

где

<ИМЯ МОДУЛЯ>
- состоит из пяти символов, каждый из которых определяет следующее:

AT VA:<АДРЕС>
- виртуальный адрес, определяющий первое слово в модуле (т.е. нулевое слово интерфейса модуля);

STAT <СЛОВО СОСТОЯНИЯ>
- за исключением битов 11, 13 и 14 все биты 16-битного слова состояния (00-15) используются для определения состояния модуля (т.е. выбранный, снятый, выполняющийся);
- все очищенные биты (000000) за исключением 11, 13 и 14 указывают, что модуль является специальным фоновым модулем (SBKMOD);
- установленный 04 бит (000020) указывает, что модуль является фоновым модулем (BKMOD);
- установленный бит 11 (004000) указывает на то, что модуль выполняется;
- установленный 09 бит (001000) указывает, что модуль является фоновым одиночным модулем (NBKMOD);
- установленные 10 и 15 биты (102000) указывают, что модуль является частично-ограниченным модулем ввода-вывода (IOMODP);
- установленные 10, 12 и 15 биты (104000) указывают, что модуль является ограниченным модулем ввода-вывода (IOMODR);
- установленные биты 12 и 15 (110000) указывают, что модуль является расширенным модулем ввода-вывода (IOMODX);
- установленный бит 13 (020000) указывает на то, что модуль снят с выполнения;
- установленный бит 14 (040000) указывает на то, что модуль выбран для выполнения;

<ЧИСЛО>
- максимальное выводимое число, содержит пять десятичных цифр.

ПРИМЕЧАНИЕ. Аргумент имени модуля должен состоять из пяти символов.

П р и м е р :

Формат 1:

```
      SUM RKAF0 <BK>
    Монитор отвечает:
      RKAF AT VA:054524 STAT 150000 PASS#000000
      HRDERRS 000000 SFTERRS 000000
        Формат 2:
         SUM  <BK>
    Монитор отвечает:
  SUMMARY AT RUNTIME:000:02:52  ;время          указывается,
    (итог на время)             ;если в системе имеется
                                ;таймер и в тесте комплекса
                                ;выполняется тест таймера
    LPAE0 AT VA:053734 STAT 150000 PASS#00000
    HRDERRS 00000
              .
              .
              .
    TCAF0 ATAVA:055310 STAT 150000 PASS#00000
    HRDERRS 00000 SFTERRS 00000
    SYSTEM ERRORS:000000 POWER FAILS:000000
                          Команда SWR
    Формат команды:
     1) SWR <ЧИСЛО>  <BK>
     2) SWR  <BK>
```

Команда SWR используется для вывода содержимого программного регистра переключателей и/или изменения его содержимого.

По команде SWR (формат 2) выводится текущее содержимое программного регистра переключателей.

Команда SWR <ЧИСЛО> (формат 1) заменяет содержание прогаммного регистра переключателей числом и распечатывает его.

```
      П р и м е р ы :
         Формат 1:
         SWR  <BK>
    Монитор отвечает:
         SWR/112000
         Формат 2:
     SWR 053401  <BK>
    Монитор отвечает:
         SWR/053401
```

4. ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ

4.1. Выполнение теста комплекса в командном файле

При выполнении теста комплекса в командном файле все команды клавиатуры выполняются также, как и в обычном режиме.

Тест комплекса периодически передает управление монитору системы. Однако, передача управления выполняется только тогда, когда тест комплекса находится в нулевом банке памяти (перемещен в 0), и все тестовые модули завершили итерационный шаг. При каждом возврате в монитор системы из счетчика проходов теста комплекса вычитается единица. Как только счетчик проходов будет исчерпан, монитор системы начинает выполнение следующего теста в командном файле. Таким образом,

при наличии в вычислительном комплексе диспетчера памяти тест комплекса будет передавать управление монитору столько раз, сколько раз он окажется в нулевом банке памяти.

Если во время выполнения теста комплекса в командном файле с клавиатуры вводится <СУ>/<C>, тест комплекса переходит в командный режим (CMD>). Это не влияет на счетчик проходов. Только при последующем запуске теста комплекса с помощью команды RUN из счетчика проходов вычитается единица. Как только счетчик проходов будет исчерпан, выполняется следующий тест цепочки.

4.2. Изменения режимов работы теста комплекса

В данном подразделе описываются изменения режимов работы.

4.2.1. Модификация мультипрограммного монитора

При перемещении теста комплекса в оперативной памяти константа перемещения по умолчанию указана в ячейке 1044. Константа перемещения имеет формат регистра адреса страницы. Для изменения шага перемещения необходимо изменить константу в ячейке 1044. Значение 200 соответствует шагу перемещения 4K слов. Значение 400 - шагу 8K слов и т.д.

Ячейка 1110 содержит максимально допустимое значение некорректируемых ошибок оборудования, при достижении которого тестовый модуль снимается с выполнения. По умолчанию установлено значение 20(десятичное). Можно изменить это значение.

Ячейка 1112 содержит максимально допустимое значение корректируемых ошибок, при достижении которого тестовый модуль снимается с выполнения. По умолчанию установлено значение 40(десятичное). Можно изменить это значение.

Ячейка 1114 содержит константу, определяющую максимальный временной интервал выполнения итерационного шага тестового модуля. Тестовый модуль, не завершивший итерационный шаг в этот интервал, снимается с выполнения. По умолчанию установлено 15 минут. Можно изменить это значение.

4.2.2. Установка специального режима для проверки оперативной памяти с корректирующим кодом (ECC)

Команда POFF отключает коррекцию одиночных ошибок и запрещает прерывания по ошибкам. Команда PON включает коррекцию одиночных ошибок и разрешает прерывания по ошибкам.

Если необходимо разрешить коррекцию одиночных ошибок, но запретить прерывания, то необходимо изменить содержимое ячеек:

PONOF+226 на 000001
PONOF+272 на 000000

Если необходимо запретить коррекцию, но разрешить прерывания по ошибкам, то необходимо изменить содержимое ячеек:

PONOF+226 на 000000
PONOF+272 на 000003

Для восстановления режима необходимо ввести последовательно команды POFF и PON.

Если в вычислительном комплексе используется 22-разрядная адресация оперативной памяти, и в тесте комплекса ис-

пользуется мультипрограммный монитор E, то для установления специального режима проверки оперативной памяти необходимо выполнить коррекцию ячеек ICSR00+126 и ICSR00+132. При этом проверяемый модуль памяти не должен быть установлен в странице ввода-вывода.

Если необходимо разрешить коррекцию и запретить прерывания по ошибкам, то нужно изменить содержимое ячеек:

ICSR00+126 на 000001
ICSR00+132 на 000000

Если необходимо запретить коррекцию одиночных ошибок и разрешить прерывания по ошибкам, то нужно изменить содержимое ячеек:

ICSR00+126 на 000000
ICSR00+132 на 000003

Начальные адреса PONOF и ICSR00 указываются в карте загрузки теста комплекса, построенной программой DXCL.

4.2.3. Модификация тестового модуля

Пользователь может модифицировать любую ячейку тестового модуля с помощью команды MOD, а также может провести замену параметров в интерфейсе тестового модуля (т.е. заменить адреса и вектора устройств, заменить уровень приоритета шины и т.д.). Эти изменения сопровождаются заменой определенных слов тестового модуля, содержащихся в интерфейсе модуля.

Описание слов интерфейса

Слово 4 (ADDR): адрес устройства на общей шине

- слово 4(ADDR) интерфейса модуля должно указывать адрес устройства на общей шине для первого устройства в группе однотипных устройств, которое будет тестироваться. Если необходим более чем один адрес, то параметр ADDR указывает первый в непрерывной группе адресов.

  Пример:

```
CMD> MOD WXYZ0 6  <BK>
52346/000000 172460  <BK>            ;адрес устройства
CMD>
```

Слово 10 (VECTOR): адрес вектора прерывания устройства

- в слове 10 (VECTOR) интерфейса модуля должен быть указан адрес вектора для первого устройства в группе однотипных устройств, которое будет тестироваться. Если необходим более чем один адрес, то параметр VECTOR указывает первый адрес в непрерывной группе.

  Пример:

```
CMD> MOD WXYZ0 10  <BK>
52350/000000 230  <BK>               ;вектор устройства
CMD>
```

Слово 12 (BR1, BR2): уровень приоритета прерывания на общей шине

- в слове 12 (BR1, BR2) интерфейса модуля указываются уровни приоритета, в четном байте - BR1, в нечетном байте - BR2. Обычно указывается только параметр BR1. Параметр BR2 задается для тех устройств, которые способны вызывать прерывания по двум уровням.

  Пример:

```
CMD> MOD WXYZ0 12  <BK>
52352/000000 300 <BK>
```

```
CMD>
```

Слово 14 (DVID1): позиционный счетчик устройств

- слово 14 (DVID1) интерфейса модуля указывает общее число работающих устройств, которые должны быть проверены (до 16), посредством установки соответствующих бит в "1". Каждый установленный бит в слове является позиционным номером проверяемого устройства. Единственный способ проверить непоследовательно расположенные устройства - использование оператора MOD.

  П р и м е р :

```
CMD> MOD WXYZ0 1  <BK>
52352/000000 5  <BK>              ;проверка устройств 0 и 2
CMD>
```

Слова 16-24 (SR1-SR4): регистры переключателей модуля

- в словах с 16 по 24 интерфейса модуля (SR1, SR2, SR3, SR4) размещены четыре 16-битных регистра переключателей, имеющихся у каждого модуля для выборки режима работы. Пользователь может, изменяя содержимое программных регистров переключателя, указать выполнение специфических подпрограмм в модуле или задать дополнительные параметры.

Слово 36 (ICONT): константа итерации

- слово 36 (ICONT) интерфейса модуля указывает число циклов работы модуля, которое будет выполняться до вывода сообщения END-OF-PASS (конец шага) и может быть изменено по усмотрению пользователя.

  П р и м е р :

```
CMD> MOD WXYZ0 36  <BK>
52376/004000 100  <BK>            ;счетчик предусматривает
                                  ;64 десятичных шага
CMD>
```

Слово 140 (WBUFRO): запрашиваемый модулем размер буфера записи

- определяет число слов, передаваемое из памяти на устройство за одну операцию записи или контроль записи в расширенных модулях ввода-вывода. Изменение значения в этой ячейке существенно изменяет время выполнения теста и загрузку общей шины передачами данных. Обычно в этой ячейке установлено значение 2000(8). Для полноценной проверки вычислительного комплекса необходимо выполнить модификацию содержимого ячеек WBUFRO во всех расширенных модулях ввода-вывода несколько раз для проверки взаимодействия устройств в режиме передач данных различной длины. Рекомендуемые значения:
  - 4008(8)
  - 5000(8)
  - 7000(8)
  - 10000(8)
  - 177400(8)

Количество ошибок, обнаруженных тестовыми модулями при одновременной работе на различных длинах передач, не должно отличаться от числа ошибок, обнаруженных при длине 400(8).

5. СООБЩЕНИЯ ОПЕРАТОРУ

Существует три типа сообщений программы:

1) Сообщения об ошибках клавиатурных команд - указывают на неправильную клавиатурную команду;

2) Сообщения о выполнении теста комплекса - указывают на местонахождение и/или окончание нормально выполненного теста комплекса или тестового модуля;

3) Сообщения об ошибках теста комплекса - указывают на ошибки, обнаруженные программой, и связанные с ними устройства.

5.1. Сообщения теста комплекса

Существует пять типов сообщений теста комплекса:

1) Сообщения о завершении прохода тестовых модулей;
2) Сообщения о снятии тестового модуля;
3) Дополнительные сообщения выполняемых модулей;
4) Сообщения о перемещении теста комплекса;
5) Сообщения о сбое питания.

Описание сообщений теста комплекса:

1) END PASS (конец шага)

Сообщение "END PASS" - необязательное сообщение. Его вывод разрешен, когда установлен 12-ый бит программного регистра переключателей (SR12=1): указывает на завершение шага очередного тестового модуля.

Когда тест комплекса запускается, вывод этого сообщения запрещен (SR12=0).

После вывода сообщения "END PASS" происходит дальнейшее выполнение заданного модуля, за исключением случая, когда шаг завершается для фонового модуля. В этом случае мультипрограммный монитор запускает выполнение следующего фонового модуля.

Сообщение "END PASS" имеет следующий вид:

```
CPAF0 END PASS # 00034.RUNTIME:000:11:37
                       PSTIME:000:00:37
```

где CPAF0 - имя модуля;
END PASS # NNNNN - десятичный номер завершенного шага;
RUNTIME/PSTIME (час:мин:сек) - определяет общее рабочее время выполнения шага;

2) MODULE DROPPED (модуль снят)

Это сообщение вызывается с помощью макровызова "END¤", или оно может быть послано мультипрограммным монитором (после анализа регистра переключателей). Сразу после вывода сообщения модуль снимается с выполнения и может быть запущен вновь по команде RUN или RUNL после прекращения работы теста комплекса при вводе с клавиатуры <CY>/<C>.

Сообщение "MODULE DROPPED" выводится:

- мультипрограммным монитором, если количество допущенных системных ошибок для модуля больше допустимого (т.е. больше четырех);
- с помощью вызова END в случае, когда возникают условия, которые тестовый модуль определяет как ненормальные (проверяемое устройство недоступно и т.д.);
- мультипрограммным монитором в случае ошибки (напечатана она или нет), если бит 15 регистра переключателей установлен (SR15=1);

- мультипрограммным монитором в случае 20-й ошибки оборудования и 40-й программной ошибки, если бит 14 регистра переключателей обнулен (SR14=0). Если бит 14 установлен (SR14=1), сообщение не выводится и модуль не снимается.

Сообщение "MODULE DROPPED"(модуль снят) имеет следующий вид:

```
                CPAF0 DROPEED AT APC XXXXXX
```

где

CPAF0 - имя снятого модуля;
APC XXXXXX - счетчик адреса в тексте модуля;

3) Сообщения в коде КОИ 7

Дополнительно мультипрограммный монитор представляет возможность каждому модулю вывести сообщение в коде КОИ 7. Такая возможность может быть использована тестовым модулем для вывода условий состояния и/или статистики.

Формат сообщений в коде КОИ 7 следующий:

```
  LPAA0 PA XXXXXXXX APC YYYYYY PASS#NNNNN
```

где

LPAA0 - имя модуля;
PA XXXXXXXX - 22-битный физический адрес модуля LPAA0;
APC YYYYYY - 18-битный счетчик адреса в тексте модуля;
PASS#NNNNN - десятичный номер завершенного шага.

Аналогичное сообщение теста комплекса с тестовой информацией:

```
          RKAA0 PA XXXXXXXX APC YYYYYY PASS#NNNNN
DATA TRANSFERS:XXXXXX   ;количество переданных слов
SOFT ERRORS:YYYYYY      ;количество корректируемых ошибок
HARD ERRORS:ZZZZZZ      ;количество некорректируемых ошибок
```

4) RELOCATED TO (перемещен в...)

Сообщение выводится, когда тест комплекса перемещается в памяти (см. команды RUN и RUNL) в следующем формате:

```
          RELOCATED TO XXXXXX00
          (перемещен в XXXXXX00)
```

где

XXXXXX00 - 22-битовый физический адрес, по которому перемещен тест комплекса;

5) POWER FAILURE (сбой питания)

Это сообщение выводится при повторном запуске после отключения питания, режим работы при этом сохраняется (BSY> или CMD>) и выводится сообщение:

```
           POWER FAILURE OCCURRED
           (обнаружен сбой сети)
```

Несмотря на то, что это аварийная ошибка, она рассматривается как нормальное сообщение теста комплекса (программная ошибка).

5.2. Сообщения об ошибках, обнаруженных тестом комплекса

Тест комплекса выводит 10 типов сообщений об ошибках:

1) Сообщение о системной ошибке (SYSTEM ERROR);
2) Сообщение о программной (корректируемой) ошибке (SOFT ERROR);
3) Сообщение об ошибке оборудования (HARD ERROR);
4) Расширенное сообщение о программной ошибке;

5) Расширенное сообщение об ошибке оборудования;
6) Сообщение об ошибке данных;
7) Сообщение мультипрограммного монитора об ошибке данных;
8) Сообщение об ошибке диспетчера памяти;
9) Сообщение об ошибке паритета памяти;
10) Сообщение о неверном векторе прерывания.

5.2.1. Сообщение о системной ошибке

Сообщение о системной ошибке выводится каждый раз, когда происходит прерывание по ошибке шины (вектор 4) или прерывание по резервной инструкции (вектор 10).

Формат сообщения:

```
                ****SYSTEM ERROR****
 VECTOR          PC+       ADDR       PSW       SP        ERCT
 AAAAAA        BBBBBB    CCCCCC     DDDDDD    EEEEEE    FFFFFF
 AT   GGGGGG         HHHHHH
```

где

AAAAAA – равно 000004 при прерывании по ошибке шины и 000010 при прерывании по резервной инструкции;
BBBBBB – значение счетчика команд, занесенное в стек в момент ошибки;
CCCCCC – действительный физический адрес ошибки;
DDDDDD – слово состояния процессора в момент ошибки;
EEEEEE – содержимое (виртуальный адрес) регистра указателя стека;
FFFFFF – десятичный счетчик системных ошибок;
GGGGGG – имя тестового модуля, если ошибка обнаружена в тестовом модуле;
NNNNNN – адрес ошибки в тексте тестового модуля, если ошибка обнаружена в модуле.

Если возникла системная ошибка, а тест комплекса был в режиме команд, то после вывода сообщения он будет снова в режиме команд (CMD>).

Если тест комплекса находился в режиме выполнения или в режиме цепочки, когда возникла системная ошибка, то режим будет вновь восстановлен. Однако, счетчик шагов и счетчик ошибок данных не будут очищены.

5.2.2. Сообщение о программных ошибках и ошибках оборудования (SOFT ERR и HARD ERR)

С позиции операционной системы программные ошибки относятся к числу корректируемых ошибок. Ошибки оборудования – некорректируемые ошибки.

Сообщения, выводимые тестом комплекса, идентичны и отличаются только указателем типа ошибки:

SOFT ERR – указатель программной ошибки;
HARD ERR – указатель ошибки оборудования.

Пример сообщения об ошибке оборудования:

```
 ABCD0  PA  XXXXXXXX APC  YYYYYY PASS#NNNNNN NARD ERR#NNNNNN
 CSRA  AAAAAA CSRC  CCCCCC  STATC SSSSSS  ERRTYP NNNNNN
```

где

ABCD0 – имя модуля, обнаружившего ошибку;
PA XXXXXXXX – 22-х битный физический адрес макровызова вывода сообщения об ошибке;

APC YYYYYY — адрес макровызова вывода сообщения в тексте тестового модуля;
PASS#NNNNNN — десятичный номер шага, в котором была обнаружена ошибка;
HARD ERR#NNNNNN — десятичное общее число обнаруженных ошибок;
CSRA AAAAAA — адрес регистра команд и состояний;
CSRC SSSSSS — содержимое регистра команд и состояний;
STATC SSSSSS — содержимое регистра состояний устройства;
ERRTYP NNNNN — восьмеричный код, определяющий тип ошибки (описан в документации на тестовый модуль). Рекомендуемые коды ошибок см. в приложении 2.

5.2.2.1. Поиск макровызова сообщения об ошибке

Для того, чтобы найти место в программе, где расположен макровызов сообщения об ошибке, необходимо в тексте тестового модуля найти адрес, указанный в поле APC (APC YYYYYY).

5.2.3. Расширенное сообщение о программных ошибках и ошибках оборудования

Расширенное сообщение о программных ошибках и ошибках оборудования выводит дополнительно к информации, указываемой в обычном сообщении о программных ошибках и ошибках оборудования, строку, в которой указано содержимое регистров проверяемого устройства.

Формат сообщения:

```
ABCD0 PA XXXXXXXX APC YYYYYY PASS#GGGGG HARD ERR#NNNNN
CSRA AAAAAA CSRC CCCCCC STATC SSSSSS ERRTYP NNNNN
XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XX XX XXXXXX
```

Поиск макровызова сообщения об ошибке выполняется аналогично тому, как это описано для сообщений о программной ошибке и ошибке оборудования.

5.2.4. Сообщение об ошибке данных

За исключением расширенных модулей ввода-вывода (IOMODX) все тестовые модули сообщают об ошибке при передаче данных с помощью макровызова "DATER#" (ошибка данных).

Формат сообщения при использовании макровызова "DATER#":

```
CSRA AAAAAA S/B BBBBBB WAS WWWWWW WRADR DDDDDD RDADR EEEEEE
```

где

DCAA0 — имя тестового модуля;
PA XXXXXXXX — 22-битный физический адрес макровызова;
APC YYYYYY — адрес макровызова DATER# в тексте тестового модуля;
PASS#NNNNNN — десятичный номер прохода;
CSRA AAAAAA — адрес регистра команд и состояний устройства, на котором обнаружена ошибка;
S/B — эталонные данные (ожидаемые);
WAS WWWWWW — данные, полученные в результате проверки (ошибочные данные);
WRADR DDDDDD — адрес эталонных данных;
RDADR EEEEEE — адрес ошибочных данных.

Для поиска в тексте тестового модуля макровызова "DATER#" используется значение APS YYYYYY. Из текста тесто-

вого модуля необходимо выяснить, в результате какой процедуры проверки произошла ошибка данных.

5.2.5. Сообщение мультипрограммного монитора об ошибке данных

Для обнаружения ошибок во время передачи данных расширенным модулем ввода-вывода (IOMODX) используется макровызов монитора "CDATA%". Этот вызов используется потому, что тестовые модули непосредственно не связаны с буфером записи. Так как назначением и перемещением буфера управляет мультипрограммный монитор, то проверка записанных и считанных данных выполняется непосредственно мультипрограммным монитором.

Сообщение мультипрограммного монитора об ошибке данных включает в свой состав ранее рассмотренное сообщение об ошибке данных, однако, имеются следующие отличия:

- все ошибки данных, обнаруженные в блоке размером 256 слов, считаются одной ошибкой (ERR#00001);
- счетчик ошибок в блоке данных выводится в конце сообщения;
- количество сообщений об ошибках данных в блоке зависит от значения бита 10 в программном регистре переключателей. Если SR10=0, то сообщается только о трех ошибках данных. Если SR10=1, то сообщается о всех ошибках данных.

Формат сообщения:

```
RKAF0 PA XXXXXXXX APC YYYYYY PASS#NNNNN ERR#NNNNN DATA ERRO
 CSRA AAAAAA S/B BBBBBB WAS WWWWWW WRADR DDDDDD RDADR EEEEEE
 RKAF0 HAD NNNN ERRORS OUT OF 256 WORDS READ
```

(Модуль RKAF0 обнаружил NNNN ошибок из 256 считанных слов). Все остальные поля сообщения аналогичны полям сообщения об ошибке данных.

5.2.6. Сообщение об ошибках диспетчера памяти

При ошибках диспетчера памяти возникает прерывание по вектору 250. Мультипрограммный монитор выводит следующее сообщение:

```
          ***  KT  TRAP  ***
          SR0    SR2
        CCCCCC   CCCCCC
          SR1    SR3
        CCCCCC   CCCCCC
```

где

SR0, SR1, SR2, SR3 - наименования регистров состояния диспетчера памяти;

CCCCCC - их содержимое.

5.2.7. Сообщение об ошибках памяти

При ошибках ББП или оперативной памяти с корректирующим кодом возникает прерывание по вектору 114. Регистр команд и состояний содержит информацию об ошибке.

Формат сообщения:

```
          ****  TRAP  THRU VEC.114  ****
           CSR      CONTENT
          AAAAAA    BBBBBB
```

где

CSR - идентификатор регистра команд и состояний;
AAAAAA - адрес регистра команд и состояний (CSR);
CONTENT
BBBBBB - содержимое регистра команд и состояний.

5.2.8. Сообщение о неверном векторе прерывания

Сообщение о неверном векторе прерывания указывает на то, что адрес вектора, по которому произошло прерывание, неверно назначен. Программа обработки прерывания не может находиться по адресу, указанному в векторе. Эта ошибка не связана с операциями, выполняемыми тестом комплекса. Однако, тестовый модуль, содержащий неверный адрес, не выведет сообщение END PASS (конец шага) и через некоторое время будет снят, если в тесте комплекса выполняется тест таймера (т.е. ведется счет системного времени).

Формат сообщения:

BAD VECTOR: XXX
(плохой вектор XXX)

Когда тестовый модуль, содержащий ложный адрес вектора найден, необходимо изменить слово 10 интерфейса модуля, назначив верное значение вектора прерывания. Информацию о назначении векторов прерывания нужно извлечь из технической документации на вычислительный комплекс.

ХАРАКТЕРИСТИКИ МУЛЬТИПРОГРАММНЫХ МОНИТОРОВ

Библиотека мультипрограммных мониторов XXXXXX.LIB содержит набор мониторных модулей, из которых программа DXCL строит рабочую версию мультипрограммного монитора. Входным параметром программы DXCL для построения рабочей версии является однобуквенное имя мультипрограммного монитора (A, C, E).

В данной версии реализована возможность построения мультипрограммных мониторов трех типов:

1) Минимальная версия (A);
2) Средняя версия (C);
3) Максимальная версия (E).

Версия E реализует все возможности версий C и A. Версия C реализует все возможности версии A.

В таблице приведены характеристики мультипрограммных мониторов:

Таблица

Выполняемые команды и операции	Имена мониторов		
	A	C	E
1		2	
COFF		*	*
CON		*	*
DES	*	*	*
EXAM		*	*
KTOFF		*	*
KTON		*	*
LPOFF		*	*
LPON		*	*
MAP		*	*
MOD	*	*	*
MOFF			*
MON			*
POFF		*	*
PON		*	*
ROTOFF		*	*
ROTON		*	*
RUN	*	*	*
RUNL		*	*
SEL	*	*	*
SUM	*	*	*
SWR	*	*	*
Прямой доступ 28K слов	*	*	*
Прямой доступ 124K слов		*	*
Прямой доступ 1024K слов			*
Отсчет времени		*	*
Перемещение		*	*

Символом "*" отмечены существующие команды и операции для версии мультипрограммного монитора.

Для ПЭВМ без диспетчера памяти рекомендуется мультипрограммный монитор A. Для вычислительных комплексов с диспетчером памяти и оперативной памятью 256К байт рекомендуется мультипрограммный монитор C, а для комплексов с диспетчером памяти и с оперативной памятью свыше 1024К байт - монитор E.

ПРИЛОЖЕНИЕ 2

КОДЫ ОШИБОК, ФОРМИРУЕМЫЕ ТЕСТОВЫМ МОДУЛЕМ

В сообщениях об ошибках по макровызовам "HRDER8" и "SOFER8" присутствует параметр "ERRTYP" (тип ошибки). Ячейка "ERRTYP" - это байт 106 интерфейса тестового модуля. Тестовый модуль перед использованием вызовов "HRDER8" и "SOFER8" должен занести в ячейку ERRTYP код, идентифицирующий ошибку.

Коды ошибок приводятся в таблице.

Таблица

Код ошибки	Причина ошибки
1	2
0	не определена
1	ошибка данных
2	потеря данных
3	контроллер не готов
4	блок не обнаружен
5	блок пропущен
6	устройство в автономном режиме или не готово
7	ошибка выборки
10	несуществующая память
11	появляется ложное прерывание
12	преждевременно обнаружен конец файла
13	ошибка перемотки (слишком длительная перемотка)
14	неверное число прерываний
15	неверный адрес вектора
16	состояние "занято" на устройстве сохраняется слишком долго
17	неизвестная ошибка приемного устройства
20	неизвестная ошибка передающего устройства
21	выход из границы
22	ошибка кадрирования
23	устройство не реагирует на прерывание
24	ошибка при блокировке/переключении
25	бит в регистре не изменяет состояния за назначенный интервал времени
26	аналого-цифровое преобразование неверно
27	ошибка разрешения прерывания
30	неизвестная ошибка во время передачи данных

1	2
31	среднеквадратичное отклонение или максимальное значение шума при аналого-цифровом преобразовании превысило предел
32	ошибка внепроцессорной передачи
33	устройство не в рабочем режиме
34	устройство не инициализируется
35	ошибка при заполнении буфера
36	не выполняется функция чтения
37	не выполняется функция записи
40	бит передачи при считывании не установлен
41	ошибка в пересылке последних данных
42	бит активности в регистре должен быть установлен, а не сброшен
43	обнаружена ошибка циклической суммы
44	флаг не должен быть установлен
45	математические действия с плавающей запятой привели к неверным результатам
46	ошибка синхронизации при аналого-цифровом преобразовании
47	контроллер не должен сбрасываться
50	изменения в наборе данных
51	неверное обращение
52	микрокод не загружен

Так как данная таблица кодов не может отразить все многообразие ошибочных ситуаций на устройствах, то данные в ней коды имеют справочный характер. В документации на каждый тестовый модуль должны быть описаны коды ошибок, применяемых в данном тестовом модуле и связанные с ними ошибочные ситуации на устройстве.

ПРИЛОЖЕНИЕ 3

ПРИМЕР ГЕНЕРАЦИИ ТЕСТА КОМПЛЕКСА

```
  R DXCL                ;загрузка и старт
  -                     ;программы DXCL
HUXC - ? DD-MMM-YY TMOS конф/комп BEP84
нужна справка ? (Y <BK> или только <BK>)
      <BK>              ;если да, то Y <BK>, если нет - <BK>
  *CNF/NP <BK>          ;запуск режима конфигуратора с
  -                     ;запретом подсказок
  MONITOR: E <BK>       ;ввод имени монитора
  *MDL CPAA <BK>        ;ввод модуля CPAA
  -
  *MDL CPBA <BK>        ;ввод модуля CPBA
  -
  *MDL FPAA <BK>        ;ввод модуля FPAA
  -
  *SR1 1 <BK>           ;изменение значения FPAA SR1
  -
```

```
*MDL KWAA <BK>          ;ввод модуля KWAA
-
*SR1 1 <BK>             ;изменение значения KWAA SR1
-
*MDL RKAA <BK>          ;ввод модуля RKAA
-
*MDL RKBA <BK>          ;ввод модуля RKBA
-
*MDL TMAA <BK>          ;ввод модуля TMAA
-
*DVC 2 <BK>             ;изменение значения TMAA DVC
-
*MDL RXAA <BK>          ;ввод модуля RXAA
-
*MDL DHAA <BK>          ;ввод модуля DHAA
-
*DVA 160200 <BK>        ;изменение значения DHAA DVA
-
*VCT 300 <BK>           ;изменение значения DHAA VCT
-
*EX <BK>                ;выход из режима конфигуратора
-
*LINK DK0:EXAMP0.BIN<DK0:XMONC0.LIB/MPL <BK>
-                       ;скомпоновать пример EXAMP0 и
                        ;вывести его на DK0 и на печать
SYS SIZE: 160000        ;необходимый размер памяти для
                        ;примера
MAKE OUTPUT READY. WRITE ENABLE
                        ;подготовить устройство вывода DK0
                        ;к работе. Отключить защиту записи
TYPE <BK>, WHEN READY;нажми <BK>, когда готово
PASS 1
TRANFER ADDRESS: 002200
LOW LIMIT: 000000
PASS 2
LINK DONE
*SAVC DK0:CEXAMP.CNF ;вывести карту памяти на DK0
-
DONE
*EXIT                   ;выход в монитор TMOC
-
-
-
```

ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

1. Тест-мониторная система. Программирование тестовых модулей. Руководство программиста.

ПРОГРАММИРОВАНИЕ ТЕСТОВЫХ МОДУЛЕЙ. РУКОВОДСТВО ПРОГРАММИСТА.

1. НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

Тестовые модули (или модули) компонуются в общий файл с мультипрограммным монитором, и полученный тест комплекса используется для проверки функционирования вычислительных комплексов.

Тест комплекса предназначен для использования в качестве теста вычислительного комплекса и для проверки возможности одновременной работы отдельных устройств. Проверка комплекса производится в мультипрограммном режиме при максимальной загрузке общей шины асинхронно с перемещением теста комплекса по памяти и циклическим перемещением буферов записи тестовых модулей. Тест комплекса предназначен для проверки взаимодействия устройств в комплексе и обнаружения ошибок(сбоев), вызванных взаимодействием. Отдельные тестовые модули пред - назначены для проверки отдельных устройств.

Тест комплекса может быть запущен с помощью монитора системы ТМОС с носителя, на котором он располагается.

С помощью теста комплекса возможна эффективная проверка вычислительных комплексов самых больших конфигураций и,кроме того, возможна проверка устройств, подсоединенных к нестандартным адресам общей шины и имеющим нестандартный вектор.

2. ХАРАКТЕРИСТИКИ ПРОГРАММЫ

Средства мультипрограммной проверки включают в себя:

1) мультипрограммные мониторы;
2) тестовые модули;
3) программу DXCL;
4) документацию по тестовой диагностической системе.

Мультипрограммный монитор, тестовые модули и программа DXCL используются для компоновки теста комплекса, который имеет формат абсолютной загрузки. При компоновке программа DXCL компонует выбранный мультипрограммный монитор и необходимые тестовые модули. Данный документ не имеет непосредственного отношения к процессу компоновки. Описание и команды программы DXCL приводятся в документе [1]. Краткие сведения о мультипрограммном мониторе приведены в приложении 1.

Тестовые модули устройств выполнены в виде отдельных необходимые модули в соответствии с конфигурацией вычислительного комплекса и размером оперативной памяти. Модульность предос - тавляет возможность легкой корректировки и модификации.

2.1. Определение тестовых модулей и их классификация

Тестовый модуль - это программа, предназначенная для проверки отдельного внешнего устройства, устройства управления, процессора или дополнительного оборудования. Тестовый модуль связан с мультипрограммным монитором, обеспечивающим обслу - живание модуля, выполнение процедур, которые могут потребо - ваться любому тестовому модулю.

Для связи с мультипрограммным монитором тестовому модулю

необходимы определенные программные средства.

Тестовые модули, используемые в тесте комплекса, разделены на семь различных групп в зависимости от типа проверяемой функции или устройства. Каждый тип модулей требует соответствующие средства связи, которые определяются интерфейсом модуля.

ПРИМЕЧАНИЕ. При написании модуля программист должен определить все глобальные и внутренние метки и значения литералов. Список общих (обязательных) меток и литералов приведен в приложении 2.

После этого программист должен определить тип модуля и его интерфейс. Описание интерфейсов для разных типов модулей дано в приложении 3.

2.2. Фоновые модули выполняются в фоновом режиме.Все модули этого типа выполняются с установленным битом "Т" слова состояния процессора (бит прерывания по слежению). В этом случае прерывания возникают после выполнения каждой команды модуля.

Кроме обработки прерываний по биту "Т", это позволяет мультипрограммному монитору управлять рядом инструкций, выполняемых фоновым модулем с более высоким приоритетом.

При написании фонового модуля метки "START" и "RESTART" должны быть размещены по одному и тому же адресу. Фоновые модули используются в тех случаях, когда проверяются устройства или функции, не вызывающие прерываний. Напрмер, модуль, написанный для процессора, проверяет все основные команды процессора.

2.3. Фоновый одиночный модуль выполняется в фоновом режиме, но с неустановленным битом "Т". Модуль этого типа выполняется только один раз. После его успешного завершения он никогда вновь не запускается, кроме тех случаев, когда тест комплекса снимается и запускается еще раз. Запускаются эти модули за специальными фоновыми модулями.

Примером являются модули для проверки временных соотношений и ошибок паритета оперативной памяти. Эти модули выполняются до запуска других модулей в тесте комплекса.

2.4. Специальный фоновый модуль работает в специальном фоновом режиме.Модули этого типа запускаются только один раз после каждого перемещения теста комплекса в памяти. Модули этого типа запускаются перед фоновыми одиночными модулями.

В качестве примера можно привести мультипроцессорную систему с переключателем шины, где выполнение функций переключателя шины требуется после каждого перемещения теста комплекса в оперативной памяти.

2.5. Модули ввода-вывода функционируют в режиме ввода-вывода. Модули управляются по прерыванию от проверяемых устройств и способны выполнять операции ввода- вывода. Тестовые модули этого типа связаны с устройствами, которые не осуществляют внепроцессорных передач и не имеют регистров счета слов. Ввод-вывод для таких устройств выполняется программным путемчерез регистр данных.

Напрмер, к таким устройствам относятся накопитель на гибком диске, печатающее устройство. Модуль ввода-вывода,запустив операцию,ожидает прерывания от внешнего устройства,чтобы продолжить выполнение. Если прерывание потеряно и не происходит, то модуль "зависает" до тех пор, пока не устанавливается специальный признак для обнаружения этого факта. Единственным способом отметить, что устройство "зависло",является установка 12-го бита программного регистра переключателей, чтобы заставить вывести на печать ENDPAS или напечатать SUM <ВК> для получения итога работы RUN SUMMARY.

2.5.1. Расширенные модули ввода-вывода используются для устройств с внепроцессорными передачами данных и обеспечивают дополнительные возможности, не требуемые для простых модулей ввода-вывода. Некоторые из этих дополнительных возможностей:

- использование буфера записи мультипрограммного монитора;
- возможность изменения размера буфера записи мультипрограммного монитора;
- обращение к мультипрограммному монитору для проверки передаваемых данных;
- перевод 16-битного адреса в 18-битный, или 18-битного адреса в 22-битный для некоторых типов процессоров.

2.5.2. Частично перемещаемый расширенный модуль ввода-вывода из-за аппаратных ограничений может перемещаться только в определенных фиксированных границах оперативной памяти. Например, внутри памяти об'емом 32К слов.

2.6. Ограниченные модули ввода-вывода не могут быть перемещены в памяти из-за ограничений на оборудование. Такой тип модуля запускается в том случае, если тест комплекса находится в нижнем разделе памяти (перемещен на 0).

3. ОБРАЩЕНИЕ К ТЕСТОВОМУ МОДУЛЮ

Модуль, включенный в тест комплекса, выполняется под управлением мультипрограммного монитора. Это управление определяется типом тестового модуля.

Мультипрограммный монитор вызывает тестовые модули в соответствующем порядке и запускает их на выполнение. В процессе работы теста комплекса модуль неразрывно связан с мультипрограммным монитором, который управляет интерфейсами между модулем и мультипрограммным монитором.

4. ВХОДНЫЕ И ВЫХОДНЫЕ ДАННЫЕ И СООБЩЕНИЯ

Входные данные тестового модуля делятся на:

1) данные, задаваемые в интерфейсе тестового модуля во время построения таблицы компоновки теста комплекса или указанные по умолчанию;
2) данные, получаемые по макровызовам от мультипрограммного монитора.

Выходные данные тестового модуля делятся на:

1) данные, передаваемые мультипрограммному монитору с помощью макровызовов;
2) сообщения, передаваемые мультипрограммному монитору, для вывода на экран терминала или печатающее устройство.

Выходные данные,передаваемые мультипрограммному монитору, определяют режим обслуживания тестового модуля мультипрограммным монитором. Так как мультипрограммный монитор выполняет обслуживание модуля, эта связь пред'являет требования к организации программы тестового модуля. В данном разделе рассматриваются способы обмена входными и выходными данными и требования, пред'являемые к структуре тестового модуля.

Для обеспечения программных связей модуля с мультипрог - раммным монитором необходимо задать в интерфейсе модуля не - которые аргументы. Детально эти аргументы описаны в п.п. 4.1. - 4.8.

4.1. Имя модуля

Формат имени модуля:

/ABCD /

Имя модуля указывается начиная с ячейки MODNAM. Аргумент состоит из пяти символов, заключенных в косые черты. Последний символ должен оставаться незаполненным (пробел) и заполняться программой DXCL во время компоновки для идентификации нескольких копий одного модуля, если он включен в тест комплекса несколько раз.

Первые обязательные четыре символа должны отвечать следующим соглашениям:

AB - любые два буквенных мнемонических символа, которые идентифицируют оборудование и этот модуль. Например, для накопителя "ЭЛЕКТРОНИКА НГМД-6121" это будет "MY";

C - буквенный символ для различия двух или более неодинаковых модулей для одного и того же устройства. Последовательность A, B, C и т.д. должна быть использована для каждого нового (добавленного) модуля. Например, модуль MYB будет дополнительным модулем MY для НГМД "ЭЛЕКТРОНИКА НГМД-6121".

D - буквенный символ, определяющий новую версию модуля. Последовательность A, B, C и т.д. должна быть использована для каждой новой версии модуля. Например, MYBC указывает, что это третья версия модуля MY.

4.2. Адрес устройства

В четвертое слово интерфейса модуля ADDR: (см.приложение 3) помещается адрес тестируемого устройства или дополнительного оборудования на системной магистрали. Если устройство использует более одного адреса, то ADDR: должен указывать первый адрес в группе непрерывных адресов.

Если адреса неизвестны до процесса компоновки, то в ячейку ADDR: следует занести 1. В этом случае, если оператор при компоновке не устанавливает действительный адрес, это влечет за собой системную ошибку - нечетный адрес, что уменьшает возможность появления ложных ошибок.

Слово ADDR: должно быть использовано для получения адреса,

необходимого тестовому модулю для доступа к регистрам устройства.

П р и м е р 1:

```
MOV     ADDR,R5          ;получение первого адреса
          :
MOV     #10000,2(R5)     ;установка бита принудительной
                         ;очистки в регистре команд
MOV     #1024,4(R5)      ;загрузка регистра счета байтов
MOV     WBUFPA,6(R5)     ;загрузка регистра адреса памяти
MOV     CMD,2(R5)        ;загрузка регистра команд
```

П р и м е р 2:

```
MOV     ADDR,R5          ;получение первого адреса
MOV     R5,MTS           ;сохранение адреса регистра состояния
TST     (R5)+            ;определение доступности регистра
MOV     R5,MTC           ;сохранение адреса регистра команд
TST     (R5)+            ;определение доступности регистра
MOV     R5,MTBC          ;сохранение адреса регистра счета
TST     (R5)+            ;определение доступности регистра
MOV     R5,MTMA          ;сохранение адреса регистра адреса шины
MOV     #100,@MTC        ;установка бита в регистре команд
MOV     WBUFPA,@MTMA     ;загрузка регистра адреса шины
```

Способ,описанный в первом примере,требует меньшего об'ема памяти и более эффективен для использования, но требует сохранения и восстановления регистра R5, если необходимо обратиться к регистрам в процедуре, обслуживающей прерывание. Этот метод не рекомендуется и должен быть отвергнут. Второй метод менее эффективен,но исключает необходимость сохранения и восстановления регистров в процедуре, обслуживающей прерывания. Это делает программу более легкой при отладке и сопровождении, поэтому рекомендуется второй метод.

4.3. Адрес вектора прерывания- VECTOR назначается устройству или дополнительному оборудованию. Если назначено больше одного вектора, то VECTOR указывает адрес первого вектора в группе. Этот адрес будет помещен в пятое слово интерфейса модуля. Для фонового модуля в ячейку VECTOR: должен быть записан 0 (ноль).

Если не назначено ни одного вектора или он неизвестен до процесса компоновки, то в ячейку VECTOR: следует занести 1. В этом случае, если оператор не устанавливает вектор при компоновке, будет генерироваться системная ошибка - нечетный адрес. Это минимизирует возможность разрушения этим модулем некоторых других модулей.Параметр VECTOR должен быть использован совместно с параметрами BR1 и BR2 для установки векторов прерывания (см. подраздел 4.4.).

4.4. Уровни приоритетов прерывания на общей шине

Для фонового модуля уровни приоритетов прерывания BR1 и BR2 всегда должны иметь значения, равные 0 (нулю). Эти параметры определяют приоритетные уровни запросов от устройств, работающих по прерываниям. Обычно используется только пара-

метр BR1. Параметр BR2 указывается только в том случае, если устройство имеет возможность выставлять запросы прерывания на двух различных уровнях. Указанные значения помещаются в шестое слово интерфейса модуля.BR1 помещается в младший байт, BR2 - в старший. Для загрузки BR1 и BR2 используется команда MOVB.

П р и м е р:

```
MOV     VECTOR,R0       ;получение адреса вектора
MOV     #MTINTR,(R0)+   ;адрес программы обработки прерываний
MOVB    BR1,(R0)+       ;установка уровня BR1
TSTB    (R0)+           ;установка четного адреса
```

4.5. Счетчик устройств DVC определяется в процессе компоновки. Он указывает число активных устройств, которые нужно проверить с помощью тестового модуля. DVC, содержащий восьмеричный код количества устройств, преобразуется и записывается в ячейку DVID1 так, что каждый бит представляет одно устройство. Если значение DVC определено как 0 (ноль), DVID1 будет соответственно равно 1. Это означает, что тестовый модуль будет проверять устройство 0 (ноль). Может быть выбрано до 16 отдельных устройств. Если во время компоновки должно быть проверено 3 устройства, тогда число 3 должно быть введено в ответ на запрос DVC при построении таблицы компоновки. Когда программа DXCL печатает DVC тестового модуля, он будет содержать число 7 - одна битовая позиция представляет каждое устройство. Это - число, которое помещается в ячейке DVID1 (*седьмое слово интерфейса модуля*).

П р и м е р:

Если в DVC помещено 3, то в DVID1 будет:

```
  0-й бит = 1       ;устройство 0 выбрано
  1-й бит = 1       ;устройство 1 выбрано
  2-й бит = 1       ;устройство 2 выбрано
```

Следовательно, три устройства выбраны и в DVID1 находится число 7. Непосредственно перед запуском теста комплекса ячейка DVID1 может быть модифицирована для того,чтобы задать комбинацию устройств, которые должны быть проверены. Модуль будет использовать DVID1 для определения того, какие устройства должны быть проверены.

Существует единственный способ проверить непоследовательные устройства - это модификация DVID1 после компоновки и сразу перед выполнением теста комплекса, используя команду MOD. Команда MOD описана в [1].

Ячейка DVID1 не должна модифицироваться ни одной из команд в модуле. Перед использованием содержимое DVID1 должно быть перемещено в рабочую ячейку.

Необходимость в модификации DVID1 возникает при снятии с проверки устройства с неисправимой ошибкой.При этом значение DVID1, установленное при компановке, не должно изменяться, а должна модифицироваться рабочая ячейка.

П р и м е р :

Считаем, что тестовый модуль проверяет 8 устройств.

```
      MOV     DVID1,TDVD    ;сохранение выбранного параметра
      MOV     #1,MSK        ;установка маски устройства
```

```
      MOV     #-1,R1          ;установка R1
KC1:  INC     R1              ;генерация номера
      BIT     MSK,TDVD        ;выбранный бит установлен?
      BNE     1$              ;если да, то переход
      CMP     #10,R1          ;проверено 8 устройств
      BEQ     OUT             ;если да, то выход
1$:   MOVB    R1,CMD+1        ;выбор устройства
        :
        :
  проверка выбранного устройства
        :
        :
      ASL     MSK             ;сдвиг выбранного бита
      BR      KC1             ;переход к проверке следующего
                              ;устройства
```

Если в процедуре проверки тестовый модуль определяет, что выбранное устройство отключено, то модуль снимает устройство с проверки:

```
      BIC     MSK,TDVD        ;снятие плохого устройства
      BNE     1$              ;переход, если какой-либо
                              ;выбранный бит установлен
      END$                    ;снятие модуля
1$:                           ;продолжение
```

4.6. Виртуальный адрес буфера чтения RBUFVA, связанный с расширенным и частично перемещаемым расширенным модулем ввода-вывода, является адресом первой ячейки буфера чтения, размещенного в модуле. Область буфера чтения содержит данные, которые были считаны с устройства. Метка для обозначения первой ячейки этого буфера обычно BUFIN. Этот буфер размещается в конце тестового модуля. Мультипрограммный монитор использует эту метку при проверке данных CDATA$ (проверка данных) и GETPA$ (получение физического адреса буфера).

4.7. Размер буфера чтения RBUFSZ, связанный с простым и частично перемещаемым расширенным модулем ввода-вывода, содержит действительный размер буфера чтения в словах. Требуемый размер буфера чтения 256 слов. Мультипрограммный монитор использует этот параметр как счетчик числа слов,которые нужно проверить при макровызове CDATA$.

Значение RBUFSZ подразумевается восьмеричным. Если значение в исходной программе задается в десятичном виде, то за указанным десятичным числом должна следовать точка <.>.

4.8. Запрашиваемый размер буфера записи WBUFRQ,связанный с простым и частично перемещаемым расширенным модулем ввода-вывода, содержит требуемый размер буфера записи в словах. Если требуемый буфер записи больше, чем допустимый, то наи - больший допустимый размер запоминается в ячейке WBUFSZ: ин - терфейса модуля. Тестовый модуль должен использовать размер буфера из ячейки WBUFSZ:, а не из ячейки WBUFRQ. Стандартный размер WBUFRQ - 1024 слова (1K).

4.9. Слово состояния модуля STAT размещается в тринадцатом слове интерфейса тестового модуля. Для определения типа

тестового модуля необходимо задать в слове состояния различные активные биты. Слово состояния модуля STAT содержит информационные биты о состоянии тестового модуля. Значения 10-15 битов слова состояния представлены ниже:

```
 бит 15 = 1    модуль ввода-вывода
        = 0    фоновый модуль
 бит 14 = 1    модуль выбран
        = 0    модуль не выбран
 бит 13 = 1    модуль был запущен, но снят монитором
        = 0    модуль не был снят
 бит 12 = 1    расширенный модуль ввода-вывода
        = 0    нет
 бит 11 = 1    модуль находится в активном состоянии
        = 0    модуль находится в неактивном состоянии
 бит 10 = 1    частично перемещаемый модуль ввода-вывода
        = 0    нет
```

Первая половина слова состояния указывает состояние процессора, когда выполняется этот тестовый модуль. В фоновых тестовых модулях бит 4 установлен, чтобы позволить мультипрограммному монитору непосредственно установить бит T в слове состояния процессора. Бит 4 = 0 для фоновых одиночных модулей ввода-вывода.

В зависимости от типа модуля устанавливаются следующие разряды слова состояния STAT:

```
STAT= 100000  для модуля ввода-вывода
STAT= 112000  для частично перемещаемого модуля ввода-вывода
STAT= 102000  для ограниченного модуля ввода-вывода
STAT= 110000  для расширенного модуля ввода-вывода
STAT= 000020  для фонового модуля
STAT= 001000  для фонового одиночного модуля
STAT= 000000  для специального фонового модуля
```

ПРИМЕЧАНИЕ. При выполнении фоновых тестовых модулей 11-й разряд (разряд активности) будет всегда равен нулю, если слово состояния этих модулей выводится с помощью команд клавиатуры SUM, MAP, EXAM.

4.10. Программные регистры переключателей SR1-SR4 размещаются с восьмого по двенадцатое слово интерфейса каждого тестового модуля. Эти регистры могут быть использованы как программные переключатели общего назначения, например, для того, чтобы однозначно определить режим проверки устройства или указать специфическую программу в тестовом модуле. Эти слова не должны модифицироваться тестовым модулем. Любое состояние SR1-SR4 должно быть указано оператором перед запуском тестового модуля. Все значения битов SR1-SR4 должны быть определены в документации на тестовый модуль. Это позволяет оператору правильно модифицировать SR1-SR4 для задания режима работы тестового модуля.

П р и м е р :

```
 SR1 определен: бит 2 = 0  ;распечатка ошибок,
                           ;запаздывания данных
                бит 2 = 1  ;распечатка не производится
                BIT    #BIT2,SR1
                BNE    6#
                MSGN#,BEGIN,DLTERR
```

В этом примере SR1 используется для указания действий, выполняемых тестовым модулем. В одном случаепроисходит вывод сообщений об ошибках, в другом - нет.

4.11. Метки

4.11.1. Метка"START" указывает первую выполняемую команду в тестовом модуле.Адрес"START" нужно указывать в слове"INIT". Мультипрограммный монитор использует содержимое этой ячейки как начальный адрес модуля.

4.11.2. Метка "RESTRT" указывает в тестовом модуле адрес перезапуска.Мультипрограммный монитор передает управление по этому адресу после завершения шага при выполнении макровызова"ENDIT#". Метка "RESTRT" помещается в начале программы подпрограммой инициализации модуля. В фоновых модулях метки "START" и "RESTRT" должны относиться к одной и той же инс - трукции программы.

4.12. Макровызовы. При программировании тестовых модулей разрешается пользоваться множеством макровызовов. Некоторые из них необходимы и должны быть использованы, другие просто облегчают процедуру программирования. Эти макровызовы применяются для обращения к процедурам определения размеров буферов, проверки данных и т.п., вывода различных сообщений об ошибках.

4.12.1. Макровызов - получение буфера записи (GWBUF#)
Формат строки:

GWBUF#,BEGIN

Макровызов GWBUF# используется в простом и частично перемещаемом расширенном модуле ввода-вывода для получения из мультипрограммного монитора размера буфера записи.Мультипрограммный монитор использует буфер записи при обмене: опреативная память - внешнее устройство. Информация никогда не должна записываться в буфер записи тестовым модулем, так как это может разрушить тест комплекса. Мультипрограммный монитор определяет размер памяти для того, чтобы определить размер свободной области памяти, которая может быть использована как буфер записи. Затем мультипрограммный монитор сравнивает его с требуемым в интерфейсе тестового модуля размером буфера записи (WBUFRQ). Если требуемый размер буфера записи больше,чем свободная область, то мультипрограммный монитор записывает допустимый размер в ячейку "WBUFSZ" интерфейса тестового модуля. Если свободная область памяти больше требуемой,в ячейку "WBUFSZ" будет записана величина, заданная в "WBUFRQ" (требуемый буфер записи).

ПРИМЕЧАНИЕ. Важно, чтобы программа тестового модуля использовала размер буфера записи из ячейки "WBUFSZ", а не значение требуемого буфера записи из ячейки "WBUFRQ" для получения действительного размера буфера записи.

Макровызов "GWBUF#" помещает также физический начальный

адрес буфера записи в ячейку "WBUFFA" и биты расширенного адреса в ячейку "WBUFEA". Биты расширенного адреса (16 и 17) помещаются в биты позиций 4 и 5 так, как расположены биты *расширенного адреса "EA" в регистрах большинства внешних* устройств. Ячейки "WBUFSZ" и "WBUFRQ" в интерфейсе тестового модуля не должны модифицироваться тестовым модулем. Этот макровызов должен использоваться перед каждым новым циклом передачи данных (таких как GWBUF, WRITE, READ, CHECKDATA и т.д.).

Мультипрограммный монитор изменяет буфер записи только после того, как все расширенные модули ввода-вывода использовали его. Следовательно, необходимо внимательно относиться к тем случаям, когда "GWBUF¤" не используется в рекомендованном виде. Значение "WBUFSZ" - положительное число и часто должно быть преобразовано в отрицательное перед использованием.

П р и м е р :

```
GWBUF¤,BEGIN            ;получение информации о буфере
                        ;записи
MOV   WBUFSZ,WCNT       ;получение размера буфера записи
NEG   WCNT              ;преобразование в отрицательное
                        ;число
MOV   WCNT,@RKBA        ;загрузка регистра счетчика слов RK
MOV   WBUFPA,@RKBA      ;загрузка адреса шины RK
BIS   WBUFEA,FUNCT      ;загрузка битов расширенного адреса
                        ;в ячейку FUNCT
```

В исходном тексте модуля макровызов "GWBUF¤" следует указывать в следующем виде:

GWBUF¤,BEGIN.

4.12.2. Макровызов - получение физического адреса(GETPA¤)

Формат строки:

GETPA¤,BEGIN,ADR

Макровызов GETPA¤ используется для преобразования 16- битового виртуального адреса в 22-битовый физический адрес. "ADR" - аргумент, который содержит виртуальный адрес. Мультипрограммный монитор берет этот виртуальный адрес, преобразует его в 18-битовый адрес и помещает его в две ячейки, следующие за ячейкой ADR. Эти три ячейки зарезервированы в интерфейсах расширенного и частично перемещаемого расширенного модуля ввода-вывода.

ПРИМЕЧАНИЕ. Когда используется любой другой тип модулей, следует убедиться, что указанные три ячейки зарезервированы в следующем порядке:

ADR - 16-разрядный виртуальный адрес;
PA - 16-разрядный физический адрес;
EA - в битах 4 и 5 данной ячейки содержатся 16-ый и 17-ый биты расширенного адреса.

Макровызов"GETPA¤" используется для получения физического адреса буфера чтения, который модуль должен загрузить в регистр устройства перед тем, как выдать команду чтения. Этот вызов может быть использован в любом типе модуля, где бы ни понадобился физический адрес. Физический адрес,эквивалентный виртуальному адресу, будет изменяться только между выводом сообщения "END OF PASS" ("Конец шага") и повторным запуском

тестового модуля. Таким образом, макровызов "GETPA%" используется только в последовательности команд, выполняющих старт или перезапуск модуля.

В примере показано использование "GETPA%" на MACRO-11.

П р и м е р :

```
GETPA%,BEGIN,RBUFVA     ;получение физического адреса из
                        ;содержимого RBUFVA
MOV     RBUFPA,@RKBA    ;загрузка адреса буфера в регистр
                        ;адреса устройства
BIS     RBUFEA,FUNCT    ;загрузка битов расширенного адреса
                        ;в ячейку FUNCT
```

4.12.3. Макровызов - запрос к мультипрограммному монитору для сравнения данных (CDATA%)

Формат строки:

CDATA%,BEGIN,ADR,ERRET

Макровызов CDATA% испоьзуется для обращения к мультипрограммному монитору при необходимости сравнения данных. Этот макровызов используется только в расширенном и частично перемещаемом расширенном модуле ввода-вывода. Аргумент ADR, используемый с макровызовом, должен быть меткой ячейки, которая содержит младшие 16 разрядов адреса сравниваемых данных. За этой ячейкой следует слово, содержащее биты расширенного адреса (EA), сдвинутые в позиции 4 и 5, а затем третье слово, содержащее размер буфера.

Макровызов "CDATA%" сравнивает содержимое буфера, указанного в аргументе "ADR" с буфером записи. Информация о буфере записи помещается в интерфейсе модуля, начиная с "WBUFPA" и должна быть предварительно установлена макровызовом"GWBUF%".

В качестве аргумента "ADR" с макровызовом "GWBUF%" наиболее часто используется "RBUFPA". Расширенный модуль ввода-вывода имеет в своем интерфейсе достаточно места для требуемых ячеек появляется в интерфейсе в следующем порядке:

```
RBUFPA:                 ;физичецкий адрес буфера чтения
RBUFEA:                 ;биты "EA" буфера чтения
RBUFSZ:                 ;размер буфера чтения
WBUFPA:                 ;физический адрес буфера записи
WBUFEA:                 ;биты "EA" буфера записи
WBUFSZ:                 ;допустимый размер буфера записи
```

Так, в исходном тексте модуля вызов следует указывать в следующем виде:

CDATA%,BEGIN,RBUFPA,ERRET

Мультипрограммный монитор будет автоматически обрабатывать все сообщения об ошибках данных, связанных с этим макровызовом "CDATA%".

Аргумент "ERRET" задает мультипрограммному монитору значения счетчика команд для возврата к указанной ячейке в случае ошибки данных. Если ошибок нет, то управление передается следующей за макровызовом "CDATA%" команде.Если при проверке содержимого буфера обнаружена ошибка, мультипрограммный монитор выдает сообщение об этой ошибке. Сообщение об ошибке аналогично сообщению по макровызову "DATER%".

4.12.3.1. Сообщение об ошибке при сравнении данных

Если мультипрограмминый монитор обнаруживает ошибку во вре-

мя проверки буфера, он сообщает об этом. Сообщение выглядит также, как для ошибки данных DATER¤, но включен один дополнительный аргумент "WORD#" и сообщается общее число переданных слов и общее число обнаруженных ошибок.

Значение аргумента "WORD#" определяет положение в буфере слова, в котором обнаружена ошибка. "WORD#" выврдится в десятичном виде. В макровызове "CDATA¤" все ошибки в данной передаче отмечаются в счетчике ошибок модуля как одна ошибка. Этот счетчик не увеличивается до тех пор, пока не будет выдано сообщение об ошибке данных.

4.12.4. Макровызов – ошибка данных (DATER¤)

Формат строки:

```
DATER¤,BEGIN
```

Макровызов DATER¤ используется для сообщения об ошибках сравнения данных в памяти. Макровызов "DATER¤" обычно никогда не используется в расширенных модулях ввода-вывода. В этих модулях используется "CDATA¤" при обращении к мультипрограммному монитору для сравнения данных. Перед использованием макровызова"DATER¤" модуль должен загрузить следующие слова:

CSRA – адрес текущего регистра команд и состояния;
SBADR – адрес эталонных данных или данных, которые должны быть верными;
WASADR – адрес данных, которые были получены в действительности;
ASB – эталонные данные;
AWAS – действительно полученные данные.

4.12.4.1. Сообщение об ошибке данных

Формат сообщения:

```
ABCD0 PA XXXXXXXX APC YYYYYY PASS#NNNNN,ERR#NNNNN,
CSRA AAAAA S/B BBBBBB WAS WWWWWW WRADR DDDDDD RDADR EEEEEE
```

где

ABCD0 – имя модуля, обнаружившего ошибку;
PA XXXXXXXX – действительный 22-битовый физический адрес вызова DATER¤;
APC YYYYYY – адрес макровызова"DATER¤"в тексте программы;
PASS#NNNNN – номер шага (десятичный), в течение которого имела место ошибка;
ERR #NNNNN – общее число(десятичное) обнаруженных ошибок;
CSRA AAAAAA – адрес регистра команд и состояний (РКС) устройства, где обнаружена ошибка;
S/B BBBBBB – данные, которые должны быть(хорошие данные);
WAS WWWWWW – данные, которые были получены(плохие);
WRADR DDDDDD – адрес эталонных данных;
RDADR EEEEEE – адрес полученных неверных данных.

4.12.5. Макровызов – ошибка оборудования (HRDER¤)

Формат строки:

```
HRDER¤,BEGIN,NULL;COMMENT
```

Макровызов HRDER¤ используется для обращения к мультипрограммному монитору,чтобы вывести на экран терминала сообщение об ошибке оборудования. Аргумент<COMMENT> позволяет включить дополнительную текстовую информацию об ошибке. При использовании этого макровызова в интерфейс модуля должна быть заг-

ружена следующая информация:

```
 CSRA      - адрес регистра команд;
 ACSR      - содержимое регистра команд;
 ASTAT     - содержимое регистра состояния (если необходимо);
 ERRTYP    - код конкретного типа ошибки (коды ошибок при-
             ведены в приложении 4).
```

В ячейки нельзя загружать никакую иную информацию,так как при инициализации модуля содержимое этих ячеек обнуляется, а после обнаружения каждой ошибки модифицируется. После вывода сообщения мультипрограммный монитор передает управление команде, следующей непосредственно за макровызовом "HRDER#".

ПРИМЕЧАНИЕ. Если ячейки не загружаются до первого обращения к макровызову "HRDER#", то при выводе сообщения об ошибке их содержимое будет представлено нулями.

П р и м е р :

Предположим, что регистр R4 содержит адрес регистра команд и состояния устройства(РКС),а регистр R5 содержит адрес регистра состояния:

```
 MOV    R4,CSRA       ;загрузка адреса РКС
 MOV    (R4),ACSR     ;загрузка содержимого РКС
 MOV    (R5),ASTAT    ;загрузка содержимого регистра состояния
 MOV    #11,ERRTYP    ;неразрешенное прерывание или не уста-
                      ;новлена готовность
 HRDER#,BEGIN,NULL ;
 B       RESTRT
```

4.12.5.1. Сообщение об ошибке оборудования

Формат сообщения:

```
 ABCD0 PA XXXXXXXX APC YYYYYY PASS#NNNNN HARD ERR#NNNNN
 CSRA AAAAAA CSRC CCCCCC STATC SSSSSS ERRTYP#XXX
```

где:

```
ABCD0           - имя модуля, обнаружившего ошибку;
PA XXXXXXXX     - действительный 22-разрядный физический
                  адрес вызова HRDER#;
APC YYYYYY      - адрес вызова HRDER на листинге;
PASS#NNNNN      - номер шага (десятичный), где произошла
                  ошибка;
HARD ERR#NNNNN  - общее количество (десятичное) обнаружен-
                  ных ошибок;
CSRA AAAAAA     - адрес регистра команд и состояния (РКС)
                  устройства, на котором произошла ошибка,
                  0 - если его нет;
CSRC CCCCCC     - содержимое регистра команд и состояния
                  (РКС), 0 - если его нет;
STATC SSSSSS    - содержимое регистра состояния, если он
                  есть;
ERRTYP#XXX      - восьмеричный код ошибки.
```

4.12.6. Макровызов - расширенное сообщение об ошибке оборудования (HRDER#)

Формат строки:

```
HRDER#,BEGIN,ADR;COMMENT
```

Макровызов HRDER# используется для обращения к мультипрограммному монитору,чтобы выдать стандартное сообщение"HRDER#",

а также содержимое всех ячеек, адреса которых содержатся в таблице, указанной аргументом "ADR". Содержимое всех ячеек в таблице печатается до тех пор, пока не будет обнаружен ограничитель (177777). После того, как сообщение выведено,управление передается модулю - ячейке, следующей сразу же за макровызовом "HRDER$".

П р и м е р :

Программа вывода содержимого всех регистров для диска будет выглядеть так:

```
HRDER$,BEGIN,TABLE      ;вывод содержимого всех регистров
                        ;контроллера кассетного диска
где:
TABLE:  177400          ;адреса регистров DK
        177402
        177404
        177406
        177410
        177412
        177416
        177777          ;ограничитель таблицы
```

ПРИМЕЧАНИЕ. Перед каждым макровызовом "HRDER$" слова CSRA, ACSR, ERRTYP и ASTAT должны быть загружены.

4.12.6.1. Пример расширенного сообщения об ошибке оборудования

Формат сообщения:

```
ABCD@ PC XXXXXX APC YYYYYY PASS#NNNNNN HARD ERR#NNNNNN
CSRA  AAAAA  CSRC  CCCCCC  STATC SSSSSS ERRTYP #NNNNNN
XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX
```

Первые две строчки вывода расширенного сообщения об ошибке имеют то же значение, что и при выводе ошибки оборудования. Третья и остальные добавочные строки, содержат до восьми восьмеричных значений в строке. Они выводятся для того,чтобы предоставить дополнительную информацию о природе ошибки.Если таблица содержит регистры устройства,регистры в таблице должны быть в том же порядке, что и действительные регистры устройства.

4.12.7. Макровызов - сообщения о программной ошибке (SOFER$)

Формат строки:

```
SOFER$,BEGIN,ADR;COMMENT
```

Макровызов SOFER$ используется для обращения к мультипрограммному монитору, чтобы вывести сообщения о программных ошибках, которые могут подвергаться коррекции. Все ранее сказанное о макровызове "HRDER$" остается справедливым для макровызова "SOFER$", включая правила использования аргументов. Сообщение о программной ошибке сопровождается выводом SOFT ERR #NNNNNN вместо HARD ERR#NNNNNN.

4.12.8. Макровызов - вызов сообщения или группы сообщений (MSGN$)

Формат строки:

```
MSGN$,BEGIN,ADR
```

Макровызов MSGN$,ADR обеспечивает средства для вывода сообщений в коде КОИ-7. Апостроф "'" служит для разделения сообщений, а знак процента "%" интерпретируется мультипрог - раммным монитором как возврат каретки <ВК> и перевод строки <ПС>. ADR является указателем таблицы сообщений. Таблица должна ограничиваться записью признака конца таблицы - 177777.

Макровызов "MSGN$"(в отличие от макровызова "MSG$")дополнительно выводит строку, идентифицирующую название модуля.

П р и м е р :

```
MSGN$,BEGIN,SOFT
SOFT:   MES2     ;указатель сообщения
        177777   ;ограничитель
MES2:   .ASCIZ   ;% программная ошибка %
```

В исходном тексте модуля макровызов "MSGN$", ADR следует указывать в следующем виде:

```
MSGN$,BEGIN,ADR
```

4.12.9. Макровызов - вывод одного сообщения (MSG$)

Формат строки:

```
MSG$,BEGIN,ADR
```

Макровызов MSG$,ADR используется для обращения к мульти - программному монитору для вывода одного сообщения, представленного в коде КОИ-7. Символ апостроф "'" используется в качестве ограничителя сообщения, а символ процент "%" интерпретируется мультипрограммным монитором как возврат каретки и перевод строки. Аргумент "ADR" является адресом сообщения. Так как при выполнении этого макровызова не выводится название модуля, для идентификации модуля, его название должно быть включено в сообщение.

П р и м е р :

```
MSG$,BEGIN,TEXT
TEXT:    .ASCIZ     ;%DKA - слишком много ошибок%
```

4.12.10. Макровызов - временный возврат в монитор (BREAK$)

Формат строки:

```
BREAK$,BEGIN
BREAK$,BEGIN
```

Макровызов BREAK$ используется для временной передачи управления мультипрограммному монитору. Этот макровызов позволяет мультипрограммному монитору перейти к обработке запро - сов в очереди, пока текущий модуль ожидает некоторого асинхронного события, которое должно случиться для того, чтобы модуль мог выполняться дальше.

При выполнении вызова монитор получает адрес возврата в модуль. Этот адрес расположен сразу после макровызова "BREAK$". Монитор сохраняет регистры общего назначения в интерфейсе модуля в ячейках SVR0-SVR6, затем проверяет очередь ожидающих запросов. Когда все предыдущие запросы обслужены, монитор восстанавливает регистры и возвращает управление в модуль, на команду, следующую за макровызовом "BREAK$".

ПРИМЕЧАНИЕ. Никакие циклы ожидания для определения задержек ответа любого вида не разрешаются без макровызова "BREAK$". Заметим, что при написании исходного текста модуля

макровызовов "BREAK¤" необходимо писать дважды.

4.12.10.1. Пример цикла ожидания, использующего макровызов "BREAK¤"

В примере цикла ожидания тестовый модуль контролирует момент появления готовности дисковода. Макровызов "BREAK¤" используется для того, чтобы на время ожидания обеспечить возврат в мультипрограммный монитор для обработки возникших за это время запросов от других модулей. Использование макровызова"BREAK¤" уменьшает время реакции системы на обработку запросов, ожидающих в очереди.

```
WAIT:   MOV     #177777,CLK     ;установка таймера
1¤:     BREAK¤,BEGIN            ;возврат в мультипрограммный
                                ;монитор
                                ;для обслуживания других мо-
        BREAK¤,BEGIN            ;дулей и выполнение модуля
        BIT     #BIT6,@RKDS     ;дисковод готов?
        BNE     2¤              ;да, продолжайте
        DEC     CLK             ;нет, ждите, пытайтесь снова
        BNE     1¤              ;снова ждать
        JSR     PC,DROP         ;время вышло, снять дисковод
```

В исходном тексте модуля макровызов "BREAK¤" следует указать в следующем виде:

```
1¤:     BREAK¤,BEGIN            ;возврат в мультипрограммный
                                ;монитор
        BREAK¤,BEGIN            ;для выполнения других модулей
```

4.12.11. Макровызов – выход в мультипрограммный монитор (EXIT¤)

Формат строки:

EXIT¤,BEGIN

Макровызов EXIT¤ необходим для ожидания прерывания от устройства. Макровызов "EXIT¤" ставится в модуле за командой, которая запускает операцию на устройстве ввода-вывода, и ее выполнение должно закончиться прерыванием центрального процессора. Важно помнить, что перед макровызовом "EXIT¤" необходима некоторая команда, чтобы сгенерировать последующее прерывание. Если это не сделано, модуль никогда не получит управления снова и в действительности прекратит выполнение. Из-за этого макровызов "EXIT¤" никогда не используется при написании тестового модуля для непрерывно работающих устройств. Другим важным пунктом является то, что макровызов "EXIT¤" не является возвратом из прерывания(RTI) и не может использоваться для возврата из программы обработки прерывания. Когда макровызов "EXIT¤" выполнен, мультипрограммный монитор сохраняет регистры общего назначения в словах SVR0-SVR6, расположенных в интерфейсе модуля (см. приложение 3). Затем мультипрограммный монитор передает управление следующему модулю, ожидающему обслуживания.

Пример:

```
MOV     WB,TMWC         ;загрузка регистра счета слов
MOV     WBUFRA,TMBA     ;загрузка регистра адреса памяти
BIS     WRCMD,TMCS      ;установка команды записи и
                        ;разрешение прерывания
```

```
EXITØ,BEGIN                ;возврат в мультипрограммный монитор
```

ПРИМЕЧАНИЕ. Если нет системных часов и не обнаруживается прерывание после макровызова "EXITØ", модуль зависает, и все другие модули остановятся после конца текущего прохода.

В исходном тексте модуля вызов "EXITØ" следует указывать в следующем виде:

```
EXITØ,BEGIN
```

4.12.12. Макровызов - возврат из прерывания (PIRQØ)

Формат строки:

```
PIRQØ,BEGIN,ADR
```

Когда тестовый модуль запускает операцию на устройстве, он возвращает управление мультипрограммному монитору по макровызову "EXITØ" и ожидает прерывания от устройства. Мультипрограммный монитор выполняет запуск следующего модуля или выполняет свои подпрограммы. Когда приходит прерывание от устройства, выполняемая программа прерывается. В этом случае необходимо обслужить это прерывание в модуле и вновь вернуться в прерванную программу.

Макровызов "PIRQØ" используется для выхода из программы, обслуживающей прерывание, и для выполнения обработки прерывания на более низком приоритетном уровне. Этот макровызов помещается первым в процедуре обработки прерывания, когда это возможно. Это увеличивает пропускную способность системы путем откладывания обслуживания некритических прерываний и последующей их обработки на более низком приоритетном уровне. Например, когда модуль DK получает прерывание, известно, что ни один из регистров не нуждается в немедленном обслуживании и не находится в процессе изменения. Следовательно, можно использовать "PIRQØ" и проверку на ошибки делать на более низком приоритетном уровне. С другой стороны, некоторые устройства связи могут нуждаться в считывании буфера немедленно, в противном случае это может привести к потере данных. В таких случаях сначала выполняются действия по предотвращению потерь данных и только затем выполняется макровызов "PIRQØ" или команда RTI (в том случае, если ожидаются еще прерывания).

Когда выполнен "PIRQØ", содержимое регистров общего назначения сохраняется в ячейках SVR0-SVR6, расположенных в интерфейсе модуля, мультипрограммный монитор сохраняет запрос макровызова "PIRQØ" в очереди, организованной по принципу "Первым пришел- первым обслужен" с приоритетом 7 и выполняет команду RTI. Эти действия передают управление команде программы, которая выполнялась и была прервана прерыванием от устройства. Запрос макровызова "PIRQØ", поставленный в очередь, обрабатывается позднее. Необходимо, чтобы программы обработки прерываний были как можно более короткими для предотвращения временного зацикливания других модулей, возможных временных ошибок и ошибок потери данных. Когда запрос макровызова "PIRQØ", стоящий в очереди монитора, обслужен, регистры общего назначения восстанавливаются из интерфейса модуля (ячейки SVR0-SVR6). Затем монитор передает управление назад модулю, выдавшему запрос "PIRQØ", на ячейку, определенную аргументом ADR.

ПРИМЕЧАНИЕ. Если необходимо использовать регистры общего

назначения в процедуре обслуживания прерывания перед использованием запроса"PIRQ#", регистры должны быть предварительно сохранены.Это необходимо потому, что эти регистры прерывания. Эти регистры должны быть восстановлены перед использованием "PIRQ#" или RTI.

```
        П р и м е р :
NTRUPT:PIRQ#,BEGIN,1#       ;процедура обработки
                            ;прерывания. Требование отложить
                            ;обработку прерывания в модуле
1#      JSR   R5,ERRORS     ;переход на проверку ошибок
```

В исходном тексте необходимо указывать:

```
 NTRUPT:
 ************************************************************
  PIRQ#,BEGIN,1#          ;в очередь для продолжения с 1#
 ************************************************************
  1#      JSR     R5,ERRORS
```

4.12.13. Макровызов – конец итераций тестового модуля (ENDIT#)

Формат строки:

```
ENDIT#,BEGIN
```

С помощью макровызова"ENDIT#" в мультипрограммный монитор передается информация о том, что закончилась одна итерация выполнения (шаг) модуля. Мультипрограммный монитор затем сравнивает содержимое ячеек "ICONT" и "ICOUNT",предварительно увеличив содержимое ячейки "ICOUNT". Если их содержимое оказывается равным, то мультипрограммный монитор сообщает на консоль о конце прохода и запускает модуль снова с адреса, определенного меткой "RESTRT". Если содержимое этих ячеек не равно, мультипрограммный монитор запускает модуль по адресу ячейки, которая следует непосредственно за макровызовом "ENDIT#". Программа модуля никогда не должна модифицировать содержимое ячеек "ICONT" и ICOUNT".

ПРИМЕЧАНИЕ. Перед использованием макровызова "ENDIT#" необходимо запретить прерывания от устройства, проверяемого модулем. Если прерывание произойдет во время выполнения макровызова "ENDIT#", то тест комплекса может быть перемещен в памяти в это время, и адрес возврата из прерывания окажется неверным.

В тексте программы макровызов "ENDIT#" выглядит следующим образом:

```
 PASS:  ENDIT#,BEGIN  ;сигнал о конце итерации, модуль
                      ;начинает проверку на конец прохода
```

4.12.14. Макровызов – прекратить выполнение тестового модуля (END#)

Формат строки:

```
END#,BEGIN
```

Макровызов END# предписывает мультипрограммному монитору прекратить выполнение модуля в тесте комплекса. Его следует использовать, если модуль определил фатальную ошибку. Вызов прекращает выполнение модуля. Например, устройство,выбранное для проверки,было отключено. Когда этот вызов выполнен,мультипрограммный монитор останавливает модуль путем установки

бита 13 слова состояния STAT, что предотвращает какую-либо повторную передачу управления этому модулю. Важно, чтобы вы выключили модуль до выполнения макровызова "END#",то есть запретили все прерывания от устройства. Мультипрограммный монитор выводит следующее сообщение для информирования оператора, что модуль был снят.

Формат сообщения:

```
ABCD0 DROPPED AT APC YYYYYY
ABCD0 снят при APC YYYYYY
```

где:

APC YYYYYY - адрес вызова END# в тексте модуля.

4.12.15. Макровызов - преобразование восьмеричного кода в КОИ-7 (OTOA#)

Формат строки:

```
OTOA#,BEGIN,NUM,ADR
```

Макровызов OTOA# преобразует одно восьмеричное число в шесть символов КОИ-7. Это может быть полезным перед выводом сообщения, использующего "MSGN#". Перед макровызовом "OTOA#, NUM,ADR должно быть предусмотрено шесть байтов памяти с начальным адресом "ADR" для запоминания результата преобразования. Аргумент NUM представляет собой адрес ячейки, в которой хранится преобразуемое число.

Пример:

```
OTOA#,BEGIN,NUM,ADR          ;вызов
.....................
.....................
NUM:    000122               ;преобразуемое число
ADR:    .BLKW   3            ;зарезервированное место для
Результат выполнения:
        ADR+0   060
           +1   060
           +2   060
           +3   061
           +4   062
           +5   062
```

4.12.16. Макровызов - преобразование двоичного числа в десятичное (BTOD#)

Формат строки:

```
BTOD#,BEGIN,NUM,ADR
```

Макровызов BTOD# преобразует двоичное число в его десятичный эквивалент, представленный пятью символами КОИ-7.Это может быть полезным перед выводом сообщения, использующего "MSGN#". Перед вызовом модуль должен заслать число, которое нужно преобразовать, в ячейку NUM. Для хранения результата должны быть обеспечены пять байтов, начиная с метки "ADR".

Пример:

```
BTOD#,BEGIN,NUM,ADR          ;вызов
.....................
.....................
ADR:    BLKW 3               ;резервирование 5 байтов
NUM:    000010
результат выполнения:
        ADR+0   060          ;код 0
```

```
+1    060          ;код 0
+2    060          ;код 0
+3    060          ;код 0
+4    070          ;код 8(10)
```

4.12.17. Макровызов – формирование случайного числа (RAND8)

Формат строки:

RAND8,BEGIN

С помощью макровызова RAND8,BEGIN выполняется обращение к мультипрограммному монитору для получения случайного числа. Это число вычисляется мультипрограммным монитором и засыла – ется в ячейку RANNUM, расположенную в интерфейсе модуля.

П р и м е р :

```
RAND8,BEGIN              ;вызов
MOV     RANNUM,BLOCK     ;засылка полученного числа
                         ;в ячейку BLOCK
```

4.13. Состав тестового модуля

Для написания тестового модуля используются команды процессора К1801ВМ1, К1801ВМ2 или К1801ВМ3. Используемый язык программирования – МАКРОАССЕМБЛЕР.

В большинстве случаев тестовый модуль может быть разделен на три части-подпрограммы: инициализация, обработка прерываний, обслуживание устройства.

4.13.1. Подпрограмма инициализации-это последовательность команд, с помощью которых инициализируется тестовый модуль и проверяемые устройства.

4.13.2. Подпрограмма обслуживания устройств является основной частью. Она состоит из связанных подпрограмм.Подпрограмма устанавливает режим, инициализирует устройство или проверяемую функцию и выдает команду для устройства ввода-вывода.

4.13.3. Подпрограмма обработки прерываний используется для обработки прерываний. Возможно использование макровызова "PIRQ8" для постановки запроса на обработку прерывания в очередь запросов на обслуживание прерываний. Необходимо,чтобы время в процедуре обслуживания прерывания при приоритете процессора было минимальным.Это должно быть сделано для предотвращения "зависания" других устройств и для правильного использования макровызова "PIRQ8".

4.13.4. Ограничения на подпрограммы тестового модуля

Благодаря связи "мультипрограммный монитор – модуль", действуют следующие ограничения для модулей теста комплекса:

1) модуль должен быть способен выполняться на всех процессорах типа к1801вм;

2) модуль не должен содержать:

- инструкций HALT в режимах, отличных от отладки;
- инструкций WAIT;
- вызовов EMT;
- вызовов пользовательских TRAP;

- команд, модифицирующих слово состояния процессора.

3) модули ввода-вывода не должны выполнять циклы ожидания, не содержащие макровызова "BREAK#";

4) если программа обработки прерываний начинается не с макровызова "PIRQ#", и перед макровызовом "PIRQ#" выполняются операции с регистрами общего назначения, то программа обработки прерываний должна сначала сохранить содержимое регистров общего назначения. Непосредственно перед макровызовом "PIRQ#" содержимое регистров должно быть восстановлено. При выполнении сегмента программы следующего за макровызовом "PIRQ#" содержимое регистров общего назначения запоминать не нужно;

5) при выходе из программы обработки прерываний нельзя изменять содержимое стека (используйте для этих целей макровызов "PIRQ#");

6) никакие изменения интерфейса модуля не могут быть сделаны самим модулем, за исключением, конечно, использования стека.

4.13.5. Стандарты по программированию необходимы для того, чтобы помочь обеспечить подобие, постоянство и единообразие всех тестовых модулей.

Эти стандарты следующие:

1) тест модуля должен содержать:
- краткое описание;
- определение итерационного шага;
- требования к оборудованию и мат. обеспечению;
- время выполнения;
- требования конфигурации;
- алгоритм работы;
- нестандартные сообщения;
- любую другую информацию, которая относится к тестовому модулю.

2) тип модуля, необходимые аргументы. Все константы и переменные должны быть определены в интерфейсе модуля;

3) метка "START" должна быть на строке, следующей за интерфейсом модуля, и указывать на первую строку кода;

4) каждая строка программы должна иметь комментарий;

5) все подпрограммы должны содержать стандартный заголовок для подпрограммы. Он будет включать функцию или обращение к подпрограмме, текущие параметры (если есть), используемые регистры, внешние параметры, процедуры обработки ошибок;

6) все регистры общего назначения (R0-R6), которые должны изменяться в подпрограмме, должны быть сохранены сразу после входа в подпрограмму и восстановлены непосредственно перед выходом из нее. Не следует использовать регистры для передачи параметров;

7) рекурсия подпрограмм не рекомендуется;

8) не рекомендуется использовать множественные (повторные) точки входа, их следует избегать. В случае использования, многократные точки входа должны быть в начале подпрограмм, а уже оттуда должен следовать переход к нужным ячейкам;

9) должен быть только один выход из подпрограммы.

4.13.6. Трансляция программы тестового модуля

Последующее обсуждение подразумевает, что программист знает,как редактировать и транслировать программы, используя MACRO, в операционных системах ОС ДВК, ФОДОС-2.

Транслированный об'ектный файл будет использован программой DXCL в качестве части загрузочного модуля системного теста комплекса. Для получения тестового модуля в об'ектном виде требуется выполнить следующие шаги:

1) используя программу редактор, создается исходный файл модуля,имеющий расширение .MAC;
2) используя транслятор MACRO, транслируется исходная программа;
3) после трансляции будут существовать следующие файлы:
   - исходный файл с расширением .MAC;
   - об'ектный файл с расширением .OBJ;
   - файл текста программы с расширением .LST (или подобным);
4) используя программу PIP (обмена между внешними устройствами):
   - вывести листинг на построчно-печатающее устройство;
   - вывести об'ектный файл на внешний носитель (например, на магнитный диск).

Программа DXCL при компоновке теста комплекса будет осуществлять ввод тестовых модулей с НГМД или другого носителя. Единственное ограничение - все модули, которые нужно об'единить, должны быть на одном носителе. На этом же носителе должна быть расположена библиотека модулей мультипрограммного монитора.

4.13.7. Выявление ошибок в программе тестового модуля

Данный пункт призван помочь программисту при проверке тестового модуля. В нем обсуждаются некоторые вопросы, возникающие при написании первых модулей:

1) используйте параметр ADDR для установки первого адреса устройства;
2) используйте параметры VECTOR, BR1, BR2 для установки векторной области;
3) используйте байтовую команду MOVB для установки BR1 и BR2;
4) перед макровызовом "EXIT$" выполните последовательность инструкций, запускающих устройство с разрешением прерывания;
5) для устройств, не работающих в режиме прерываний, нельзя использовать макровызов "EXIT$" и "PIRQ$";
6) используйте команду RTI для возврата из подпрограммы обработки прерывания только тогда, когда макровызов "PIRQ$" не используется;
7) макровызов "PIRQ$" следует использовать для выхода из подпрограммы обработки прерывания;
8) не должн быть вызовов мультипрограммного монитора в подпрограмме обработки прерывания;
9) модуль должен сохранять и восстанавливать все регистры, которые необходимы в подпрограмме обработки прерывания;
10) перед макровызовами "HRDNR$" и "SOFER$" нужно загрузить слова CSRA, ACSR, ASTAT и ERRTYP;
11) перед макровызовом "DATER$" нужно загрузить слова CSRA,

ASB, AWAS, WASADR и SBADR;
12) модуль должен восстанавливаться и продолжать выполнение после нефатальной ошибки;
13) если обнаружена ошибка оборудования, которая делает невозможным продолжение выполнения модуля, необходимо использовать макровызов "END¤" для снятия модуля;
14) можно использовать макровызов "ENDIT¤" для сообщения о том, что модуль закончил шаг;
15) не должно быть циклов ожидания в модуле без использования макровызова "BREAK¤";
16) необходимо позаботиться о разрешении всех прерываний во время работы модуля;
17) запрещается обращаться к абсолютной памяти;
18) не рекомендуется использовать команды HALT, WAIT, TRAP и EMT;
19) следует правильно использовать модификацию "DVID1" для проверки множества устройств;
20) если устройство имеет возможность работать с расширенной памятью, нужно правильно использовать биты для установки битов расширенной памяти;
21) нельзя разрешать прерывания из модуля перед вызовом монитора (исключение - макровызов "EXIT¤");
22) следует использовать метку "START" для обозначения первой выполняемой инструкции модуля;
23) следует использовать метку "RESTRT" для обозначения адреса возврата из макровызова "ENDIT¤";
24) в фоновых модулях метки "START" и "RESTRT" должны относиться к одной ячейке.

4.13.8. Проверка работоспособности тестового модуля

4.13.8.1. Самостоятельное выполнение тестового модуля

Модуль должен быть способен выполняться самостоятельно в любых условиях:
1) При проверке на надежность модуль должен быть проверен непрерывно в течение времени, определяемого ОТК.
2) Если модуль способен выполняться более, чем на одном устройстве (или линии, или канале и т.п.), то абсолютное минимальное число устройств, на которых он должен проверяться, устанавливается совместно с ОТК.
3) Все сообщения об ошибках и распечатки должны быть проверены на точность;
4) Все режимы, задаваемые с помощью SR1-SR4, должны быть проверены;
5) Различные комбинации модификации DVID1 должны быть проверены на правильность выполнения;
6) В системе с диспетчером памяти модуль должен быть выполнен во всех банках памяти;
7) Модуль должен быть способен восстанавливаться после отказа питания.

4.13.8.2. Выполнение тестового модуля в тесте комплекса

Модуль должен быть способен выполняться одновременно с другими тестовыми модулями в следующих режимах:
1) при проверке на надежность модуль должен выполняться в

тесте комплекса не менее восьми часов (лучше 24);

2) при максимально-допустимом наборе внешних устройств и проверяемых функций (наибольшее число внешних устройств);
3) модуль в скомпонованном тесте комплекса должен быть обязательно проверен на совместную работу с модулем для проверки дисков;
4) модуль должен быть проверен на совместную работу со всеми ранее разработанными модулями.

МУЛЬТИПРОГРАММНЫЕ МОНИТОРЫ

1. Характеристики мультипрограммных мониторов

Библиотека мультипрограммных мониторов XXXXXX.LIB содержит набор мониторных модулей, из которых программа DXCL строит рабочую версию мультипрограммного монитора. Входным параметром программы DXCL для построения рабочей версии является однобуквенное имя мультипрограммного монитора(A,C,E).

В данной версии реализована возможность построения мультипрограммных мониторов трех типов:

1) минимальная версия (A);
2) средняя версия (C);
3) максимальная версия (E).

Следует отметить, что версия E реализует все возможности версий C и A. Версия C реализует все возможности версии A.

Команды и операции, выполняемые мультипрограммным монитором, описаны в документе [1].

В таблице приведены характеристики мультипрограммных мониторов:

Таблица

Выполняемые команды и операции	Имена мониторов		
	A	C	E
1	2		
COFF		*	*
CON		*	*
DES	*	*	*
EXAM		*	*
FILL	*	*	*
KTOFF		*	*
KTON		*	*
LPOFF		*	*
LPON		*	*
MAP		*	*
MOD	*	*	*
MOFF			*
MON			*
POFF		*	*
PON		*	*
ROTOFF		*	*
ROTON		*	*
RUN	*	*	*
RUNL		*	*
SEL	*	*	*
SUM	*	*	*
SWR	*	*	*
Прямой доступ 28K слов	*	*	*
Прямой доступ 124K слов		*	*
Прямой доступ 1024K слов			*

1	2			
Отсчет времени		*	*	
Перемещение		*	*	

Символом "*" отмечены существующие команды и операции для версии мультипрограммного монитора.

1.1. Изменения режимов работы теста комплекса

В данном подразделе приложения описываются способы модификации теста комплекса для изменения режимов работы.

1.2. Модификация мультипрограммного монитора

При перемещении теста комплекса в оперативной памяти константа перемещения по умолчанию указана в ячейке 1044. Константа перемещения имеет формат регистра адреса страницы. Для изменения шага перемещения необходимо изменить константу в ячейке 1044. Значение 200 соответствует шагу перемещения 4К слов. Значение 400 - шагу 8К слов и т.д.

Ячейка 1110 содержит максимально допустимое значение не - корректируемых ошибок оборудования, при достижении которого модуль снимается с выполнения. По умолчанию установлено значение 20. Можно изменить это значение.

Ячейка 1112 содержит максимально допустимое значение корректируемых ошибок, при достижении которого тестовый модуль снимается с выполнения. По умолчанию установлено значение 40. Можно изменить это значение.

Ячейка 1114 содержит константу, определяющую максимальный временной интервал выполнения итерационного шага тестового модуля. Тестовый модуль, не завершивший итерационный шаг в этот интервал, снимается с выполнения. По умолчанию установлено 15 минут. Можно изменить это значение.

1.3. Установка специального режима для проверки оперативной памяти с корректирующим кодом (ECC)

Команда POFF отключает коррекцию и запрещает прерывания по ошибкам. Команда RON включает коррекцию и разрешает пре - рывания по ошибкам.

Если необходимо разрешить коррекцию, но запретить преры - вания, то необходимо изменить содержимое ячеек:

PONOF+226 на 000001
PONOF+272 на 000000

Если необходимо запретить коррекцию, но разрешить преры - вание по ошибкам, то необходимо изменить содержимое ячеек:

PONOF+226 на 000000
PONOF+272 на 000003

Для восстановления режима необходимо ввести последова- тельно команды POFF и PON.

Если в вычислительном комплексе используется 22-разрядная адресация оперативной памяти и в тесте комплекса использу - ется мультипрограммный монитор E, то для установки специаль- ного режима проверки оперативной памяти необходимо выполнить коррекцию ячеек ICSR00+126 и ICSR00+132. При этом проверяе -

мый модуль памяти не должен быть установлен в странице ввода-вывода.

Если необходимо разрешить коррекцию и запретить прерывания по ошибкам, то нужно изменить содержимое ячеек:

ICSR00+126 на 000001
ICSR00+132 на 000000

Если необходимо запретить коррекцию и разрешить прерывания по ошибкам, то нужно изменить содержимое ячеек:

ICSR00+126 на 000000
ICSR00+132 на 000003

Начальные адреса PONOF и ICSR00 указываются в карте загрузки теста комплекса, построенной программой DXCL.

1.4. Выполнение теста комплекса в командном файле

При выполнении теста комплекса в командном файле все команды клавиатуры выполняются так же, как в обычном режиме.

Тест комплекса периодически передает управление монитору системы. Однако, передача управления выполняется только тогда, когда тест комплекса находится в нулевом банке памяти (перемещен в 0),и все тестовые модули завершили итерационный шаг.При каждом возврате в монитор системы вычитается единица из счетчика проходов теста комплекса в цепочке. Как только счетчик проходов будет исчерпан, монитор системы начинает выполнение следующего теста в командном файле. Таким образом, при наличии в вычислительном комплексе диспетчера памяти тест комплекса будет передавать управление монитору системы столько раз, сколько раз он окажется в нулевом банке памяти.

Если во время выполнения теста комплекса в командном файле с клавиатуры ввести <СУ>/<С>, то тест комплекса переходит в командный режим (CMD>). Это не влияет на счетчик проходов. Только при последующем запуске теста комплекса с помощью команды RUN из счетчика проходов вычитается единица. Как только счетчик проходов будет исчерпан, выполняется следующий тест цепочки.

ПРИЛОЖЕНИЕ 2

ОПИСАНИЕ ГЛОБАЛЬНЫХ МЕТОК И ЛИТЕРАЛОВ

При написании модулей необходимо в первую очередь определить используемые глобальные метки и литералы.

Список обязательных определений, необходимых для работы любого тестового модуля:

Имя	Значение	Имя	Значение
BIT0	= 000001	MSGS8	= 104402
BIT1	= 000002	NULL	= 000000
BIT2	= 000004	OTOA8	= 104420
BIT3	= 000010	OPEN	= 000000
BIT4	= 000020	PC	= %000007
BIT5	= 000040	PIRQ8	= 000004
BIT6	= 000100	PRTY	= 000000
BIT7	= 000200	PRTY0	= 000000
BIT8	= 000400	PRTY1	= 000040
BIT9	= 001000	PRTY2	= 000100
BIT10	= 002000	PRTY3	= 000140

```
BIT11   = 004000        PRTY4   = 000200
BIT12   = 010000        PRTY5   = 000240
BIT13   = 020000        PRTY6   = 000300
BIT14   = 040000        PRTY7   = 000340
BIT15   = 100000        PS      = 177776
BREAK$  = 104407        PSW     = 177776
BTOD$   = 104421        RAND$   = 104417
CDATA$  = 104412        R0      = %000000
DATCK$  = 104411        R1      = %000001
DATER$  = 104404        R2      = %000002
ENDIT$  = 104413        R3      = %000003
END$    = 104410        R4      = %000004
EXIT$   = 104400        R5      = %000005
GETPA$  = 104415        R6      = %000006
GWBUF$  = 104414        R7      = %000007
HRDER$  = 104405        SP      = %000006
MAP22$  = 104416        SOFER$  = 104406
MSGN$   = 104403        SPSIZ   = 000040
MSG$    = 104401
```

ПРИЛОЖЕНИЕ 3

ИНТЕРФЕЙСЫ ТЕСТОВЫХ МОДУЛЕЙ

Интерфейс модуля необходим мультипрограммному монитору для управления операциями модуля. Интерфейс модуля содержит 122 или 150 (в случае расширенного модуля ввода-вывода) слов,которые разделены на отдельные поля следующими метками:

Байт - 00 BEGIN: - первая метка интерфейса, указывающая начало модуля. Относится к логической ячейке 0;

Байт - 00 MODNAM: - следует за меткой BEGIN. Указывает на поля, содержащие имя модуля ASCII. Поле занимает пять байтов;

```
MODNAM: .ASCII /ABCD / ; имя модуля
```

Байт - 05 XFLAG: - резервирует один байт. Используется для перемещения расширенного модуля ввода-вывода по всему полю памяти;

```
XFLAG:   .BYTE OPEN
```

Байт - 06 ADDR: - резервирует одно слово, содержащее восьмеричный адрес первого регистра проверяемого устройства;

```
ADDR:   XXXXXX+0
```

Байт - 10 VECTOR: - резервирует одно слово, содержащее восьмеричный адрес назначенного устройству вектора;

```
VECTOR:  XXX+0
```

Байт - 12 BR1: - резервирует один байт для указания первого уровня приоритета;

```
BR1:     .BYTE PRTYX+0
```

Байт - 13 BR2: - указывает второй уровень приоритета, если проверяемое устройство разрешает прерывания по двум уровням;

```
BR2:     .BYTE  PRTYX+0
```

Байт - 14 DVID1: - резервирует одно слово, содержащее

счетчик устройств. Используется для указания количества и номеров устройств, которые нужно проверить. Для каждого устройства устанавливается один бит;

П р и м е р :

Если драйвер МЛ управляет восемью устройствами, будут установлены биты 0 - 7 DVID1. Бит 0 указывает устройство 0, бит 7 - устройство 7.

DVID1: N15+N14+...+N0

где N0 - N15 - позиционные номера устройств;

Байт 16 -24 SR1:-SR4: - резервирует четыре слова, для внутренних регистров переключателей. Каждый бит этого слова используется для изменения режима работы тесто - вого модуля;

SR1: OPEN
SR2: OPEN
SR3: OPEN
SR4: OPEN

Байт - 26 STAT: - слово состояния модуля;

STAT: XXXXXX

Байт - 30 INIT: - резервирует слово, содержащее стар - товый адрес модуля;

INIT: START

Байт - 32 SPOINT: - резервирует одно слово, содержащее адрес загрузки указателя стека при первоначальном запуске модуля;

SPOINT: MODSP

где MODSP - адрес начала области стека модуля;

Байт - 34 PASONT: - резервирует слово,содержащее счетчик проходов;

PASCNT: 0

Байт - 36 ICONT: - число итераций за один проход;

ICONT: 512
ICOUNT: 0

Байт - 42 SOFCNT: - счетчик программных ошибок;

SOFCNT: 0

Байт - 44 HRDCNT: - счетчик ошибок оборудования;

HRDCNT: 0

Байт - 46 SOFPAS: - счетчик программных ошибок за проход;

SOFPAS: 0

Байт - 50 HRDPAS: - счетчик ошибок оборудования за проход;

HRDPAS: 0

Байт - 52 SYSCNT: - счетчик системных ошибок;

SYSCNT: 0

Байт - 54 RANNUM: - случайное число, полученное по макровызову RUND%;

RUNNUM: 0
байт - 56 CONFIG:

Байт - 56 RES1: - зарезервировано для использования мультипрограммным монитором;

CONFIG:

RES1: 0

Байт - 60 RES2: - зарезервировано для использования мультипрограммным монитором;
RES2: 0

Байт 62 - 76 SVR0-SVR6 - эти метки резервируют семь слов для сохранения регистров модуля и указателя стека при передаче модулем управления монитору;
Байт - 62 SVR0: OPEN
. .
. .
Байт - 76 SVR6: OPEN

Байт - 100 CSRA: - резервирует одно слово, содержащее адрес регистра команд и состояний устройства, где произошла ошибка;
CSRA: OPEN

Байт - 102 SBADR/ACSR: - резервирует одно слово, содержащее в случае ошибки данных адрес правильных данных, а в случае сообщения "ERROR8" содержимое регистров команд и состояний устройства с ошибкой;
SBADR:
ACSR: OPEN

Байт - 104 WASADR/ASTAT: - резервирует одно слово. Очищается после распечатки ошибок. В случае ошибки данных содержит адрес неправильных данных. В сообщении ERROR содержит содержимое регистров состояний устройства с ошибкой (если он есть);
WASADR:
ASTAT: OPEN

Байт - 106 ERRTYP: - идентификатор типа ошибки (заносится из модуля);

Байт - 106 ASB: - резервирует слово, содержащее правильные данные (эталон). Слово очищается после распечатки ошибки;
ERRTYP:
ASB: OPEN

Байт - 110 AWAS: - резервирует слово, содержащее плохие данные. Очищается после распечатки ошибок;
AWAS: OPEN

Байт - 112 RSTRT: - резервирует слово, содержащее адрес повторного запуска модуля;
RSTRT: RESTRT

Байт - 114 WDTO: OPEN - число слов, переданных в память за итерацию, заполняется тестовым модулем;

Байт - 120 INTR: OPEN - число прерываний за итерацию, заполняется тестовым модулем;

Байт - 122 IDNUM: NUM - идентификационный номер модуля;

Дополнительные слова используются только в интерфейсах

расширенных модулей ввода-вывода.

Байт - 124 RBUFVA: - виртуальный адрес буфера чтения модуля;
RBUFVA: BUFIN

Байт - 126 RBUFPA: - содержит младшие 16 бит физического адреса буфера чтения;
RBUFPA: OPEN

Байт - 130 RBUFEA: - содержит биты расширенного адреса буфера чтения, сдвинутые в позиции 4 и 5;
RBUFEA: OPEN

Байт - 132 RBUFSZ: - содержит размер буфера чтения в словах (обычно 256.);
RBUFSZ: XXX
где XXX - десятичное число слов;

Байт - 134 WBUFPA: - содержит младшие 16 бит физического адреса буфера записи, назначенные мультипрограммным монитором при выполнении макровызова GWBUF;
WBUFPA: OPEN

Байт - 136 WBUFEA: - биты расширенного адреса буфера записи, сдвинутые в позиции 4 и 5;
WBUFEA: OPEN

Байт - 140 WBUFRQ: - указывается запрашиваемый модулем размер буфера записи в словах (обычно 1024.);
WBUFRQ: XXXX. (1024.)
где XXXX - десятичное число слов;

Байт - 142 WBUFSZ: - размер буфера записи, представленного модулю мультипрограммным монитором (в словах);
WBUFSZ: OPEN

Байт - 144 CDERCT: - счетчик ошибок при сравнении данных;
CDERCT: OPEN

Байт - 146 CDWDCT: - счетчик сравниваемых слов.
CDWDCT: OPEN

Здесь кончается специальная область интерфейса для расширенных модулей ввода-вывода.

Заключительная часть интерфейсов всех модулей:

Байт - 150 FREE: - резервируется одно слово для возможного использования мультипрограммным монитором в будущем;
FREE: OPEN

после этого указываются директивы:

```
.REPT SPSIZ - размер стека модуля SPSIZ=000040
.NLIST
.WORD 0
.LIST
.ENDR
```

MODSP: - стек тестового модуля.

Заметим, что в каждом модуле резервируются 32 слова для стека модуля. Когда модуль выполняется, он работает со своим собственным стеком.

ИНТЕРФЕЙС ФОНОВЫХ МОДУЛЕЙ ВСЕХ ТИПОВ
И МОДУЛЕЙ ВВОДА-ВЫВОДА

```
BEGIN:
MODNAM: .ASCII /ABCD / ;имя модуля
XFLAG:  .BYTE OPEN
ADDR:   XXXXXX+0
VECTOR: XXX+0
BR1:    .BYTE  PRTYX+0
BR2:    .BYTE  PRTYX+0
DVID1: N15+N14+...+N0
SR1:    OPEN
SR2:    OPEN
SR3:    OPEN
SR4:    OPEN
STAT:   XXXXXX
INIT:   START
SPOINT: MODSP
PASCNT: 0
ICONT: 512.
ICOUNT: 0
SOFCNT: 0
HRDCNT: 0
SOFPAS: 0
HRDPAS: 0
SYSCNT: 0
RUNNUM: 0
CONFIG:
RES1: 0
RES2: 0
CSRA: OPEN
SBADR:
ACSR:  OPEN
WASADR:
ASTAT: OPEN
ERRTYP:
ASB:   OPEN
AWAS: OPEN
RSTRT: RESTRT
WDTO: OPEN
WDFR: OPEN
INTR: OPEN
IDNUM: NUM
FREE:  OPEN
    после этого указываются директивы:
.REPT SPSIZ - размер стека тестового модуля SPSIZ=000040
.NLIST
.WORD 0
.LIST
.ENDR
MODSP:
      - стек тестового модуля.
```

ИНТЕРФЕЙС РАСШИРЕННЫХ МОДУЛЕЙ ВВОДА-ВЫВОДА

```
BEGIN:
MODNAM: .ASCII /ABCD / ;имя модуля
XFLAG:  .BYTE CPEN
ADDR:   XXXXXX+0
VECTOR: XXX+0
BR1:    .BYTE  PRTYX+0
BR2:    .BYTE  PRTYX+0
DVID1: N15+N14+...+N0
SR1:    OPEN
SR2:    OPEN
SR3:    OPEN
SR4:    OPEN
STAT:   XXXXXX
INIT:   START
SPOINT: MODSP
PASCNT: 0
ICONT: 512.
ICOUNT: 0
SOFCNT: 0
HRDCNT: 0
SOFPAS: 0
HRDPAS: 0
SYSCNT: 0
RUNNUM: 0
CONFIG:
RES1: 0
RES2: 0
CSRA: OPEN
SBADR:
ACSR:  OPEN
WASADR:
ASTAT: OPEN
ERRTYP:
ASB:   OPEN
AWAS: OPEN
RSTRT: RESTRT
WDTO: OPEN
WDFR: OPEN
INTR: OPEN
IDNUM: NUM
RBUFVA: BUFIN
RBUFRA: OPEN
RBUFEA: OPEN
RBUFSZ: XXX
WBUFRA: OPEN
WBUFEA: OPEN
WBUFRQ: XXXX.   (1024.)
WBUFSZ: OPEN
CDERCT: OPEN
CDWDCT: OPEN
FREE:  OPEN
.REPT SPSIZ
.NLIST
.WORD 0
.LIST
```

```
.ENDR
MODSP:
```

ПРИЛОЖЕНИЕ 4

КОДЫ ОШИБОК, ФОРМИРУЕМЫЕ ТЕСТОВЫМ МОДУЛЕМ

В сообщениях об ошибках по макровызовам"HRDR#" и "SOFER#" присутствует параметр"ERRTYP"(тип ошибки). Ячейка "ERRTYP" – это слово 106 интерфейса модуля. Модуль перед использованием макровызовов"HRDER#"и"SOFER#"должен занести в ячейку"ERRTYP" код, идентифицирующий ошибку.

В таблице приводятся следующие коды ошибок:

Таблица

Код ошибки	Причина ошибки
0	не определена
1	ошибка данных
2	потеря данных
3	контроллер не готов
4	блок не обнаружен
5	блок пропущен
6	устройство в автономном режиме или не готово
7	ошибка выборки
10	несуществующая память
11	появляется ложное прерывание
12	преждевременно обнаружен конец файла
13	ошибка перемотки(слишком длительная перемотка)
14	неверное число прерываний
15	неверный адрес вектора
16	состояние "Занято" на устройстве сохраняется слишком долго
17	неизвестная ошибка приемного устройства
20	неизвестная ошибка передающего устройства
21	выход за границы
22	ошибка кадрирования
23	устройство не реагирует на прерывание
24	ошибка при блокировке/переключении
25	бит в регистре не изменяет состояния за назначенный интервал времени
26	аналого-цифровое преобразование неверно
27	ошибка разрешения прерывания
30	неизвестная ошибка во время передачи данных
31	среднеквадратичное отклонение или максимальное значение шума при аналого-цифровом преобразовании превысило предел
32	ошибка внепроцессорной передачи
33	устройство не в рабочем режиме
34	устройство не инициализируется
35	ошибка при заполнении буфера
36	не выполняется функция чтения
37	не выполняется функция записи

1	2
40	бит передачи при считывании не установлен
41	ошибка в пересылке последних данных
42	бит активности в регистре должен быть установлен, а не сброшен
43	обнаружена ошибка циклической суммы
44	флаг не должен быть установлен
45	математические действия с плавающей запятой привели к неверным результатам
46	ошибка синхронизации при аналого-цифровом преобразовании
47	контроллер не должен сбрасываться
50	изменения в наборе данных
51	неверное обращение
52	микрокод не загружен

Так как данная таблица кодов не может отразить все многообразие ошибочных ситуаций на устройствах, то данные в ней коды имеют справочный характер. В документации на каждый модуль должны быть описаны коды ошибок, применяемых в данном модуле и связанные с ними ошибочные ситуации на устройстве.

ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

1. Подсистема мультипрограммной проверки. Руководство оператора.

СОДЕРЖАНИЕ

ОПИСАНИЕ СИСТЕМЫ. РУКОВОДСТВО ОПЕРАТОРА.

ОПИСАНИЕ ТЕСТОВЫХ ПРОГРАММ. РУКОВОДСТВО ОПЕРАТОРА.

ПОДСИСТЕМА МУЛЬТИПРОГРАММНОЙ ПРОВЕРКИ. РУКОВОДСТВО ОПЕРАТОРА.

ПРОГРАММИРОВАНИЕ ТЕСТОВЫХ МОДУЛЕЙ. РУКОВОДСТВО ПРОГРАММИСТА.

Подписано в печать 31.05.88. Заказ 510. Тираж 20 000. Бесплатно.

Тип. им. Котлякова издательства „Финансы и статистика" Государственного комитета по делам издательств, полиграфии и книжной торговли.
195273, Ленинград, ул. Руставели, 13